# Adaptive Privacy-Preserving Authentication in Vehicular Networks (Invited Paper)

Kewei Sha[1], Yong Xi[1], Weisong Shi[1], Loren Schwiebert[1], and Tao Zhang[2]

[1]Department of Computer Science, Wayne State University
[2]Telcordia Technologies, Inc.

***Abstract*— Vehicular networks have attracted extensive attentions in recent years for their promises in improving safety and enabling other value-added services. Most previous work focuses on designing the media access and physical layer protocols. Privacy issues in vehicular systems have not been well addressed. We argue that privacy is a *user-specific* concept, and a good privacy protection mechanism should allow users to select the degrees of privacy they wish to have. To address this requirement, we propose an adaptive privacy-preserving authentication mechanism that can trade off the privacy degree with computational and communication overheads (resource usage). This mechanism, to our knowledge, is the first effort on adaptive privacy-preserving authentication. We present analytical and preliminary simulation results to show that the proposed protocol is not only adaptive but also scalable.**

## I. Introduction

About half of the 43,000 deaths that occur each year on U.S. highways result from vehicles leaving the road or traveling unsafely through intersections. Traffic delays waste more than a 40-hour workweek for peak-time travelers [1]. Fortunately, with the development of micro-electronic technologies and wireless communications, it is possible to install an On-Board-Unit (OBU), which integrates the technologies of wireless communications, micro-sensors, embedded systems, and Global Positioning System (GPS), on vehicles. With these devices, vehicles can communicate with each other or with roadside units (RSU) connected to Internet. Thus, vehicles, RSUs and the backbone network form a vehicle infrastructure integration (VII) system [1]. VII can be used to collect traffic and road information from vehicles, and deliver road services including road warning and traffic information to the users in the vehicles. Thus, a great attention has been put into designing and implementing similar systems in the past several years [2], [3].

Current research in VII mainly focuses on vehicular communications. Significant progresses have been made in media access (MAC) layer protocols [4] and physical layer protocols [5]. However, issues about security and privacy, which will play a critical role in the acceptance of the VII system, have not been well studied. Vehicles and the networks need to authenticate each other. Several previous efforts have been made to protect user privacy in the authentication process, but most of them use a policy that places trust on the RSUs or the authentication servers in the network. That is, these trusted RSUs or authentication servers can track the locations and activities of vehicles and their drivers. Concerns about security and privacy may prevent vehicle owners from joining this system. We argue that we need to provide vehicle owners better privacy through anonymity, i.e., no one can trace their activities based on the information provided for the authentication purpose. In this paper, we analyze security and privacy requirements and challenges with the assumption that there is zero-trust of authentication servers. Among these requirements and challenges, we observe that privacy is treated as a one-size-fits-all concept in previous research efforts. However, we argue that privacy is a *user-specific* concept in the sense that different users may have varying privacy requirements. Moreover, a higher privacy requirement usually results in more computational or communication overhead. A trade-off should be made between the privacy degree and resource usage to meet overall system design goals such as scalability and real-time response. Thus, we propose an adaptive group-based authentication protocol that is able to trade off the degree of privacy with resource usage. Both analytic results and preliminary simulation results show that the protocol provides promising performance in a real system. In summary, the main contributions of this paper are: (1) We analyze the system design requirements from the view of security and privacy and define the challenges to achieve these requirements; (2) We propose and evaluate an adaptive privacy-preserving authentication protocol; (3) We introduce the concept of adaptive privacy and discuss the trade-off between the degree of privacy and resource usage.

The rest of the paper is organized as follows. We analyze the requirements and challenges of security and privacy design in VII in Section II. The application scenario of the authentication protocol in a VII is described in Section III. In section IV, a privacy-preserving authentication protocol for the zero-trust model is proposed and the performance evaluation of the proposed protocol is done in Section V. Finally, related work is discussed in Section VI and conclusion is drawn in Section VII.

## II. Requirements and Challenges

VII can improve driving safety. However, due to the extremely large system scale, the fast movement of vehicles, and the broadcast nature of wireless communications, there are several requirements and challenges in designing and

deploying such a system. The challenges related to security and privacy include the following:

- **Adaptive privacy:** Mobile users may be concerned with two types of privacies: location and identity privacy (when users/vehicles communicate with the network or with each other) the privacy about the service usage pattern (when a user/vehicle requests services from service providers). Furthermore, privacy is a *user-specific* concept; some users are more serious about their privacy than others. Thus, we argue that the VII should support multiple privacy degrees, and each user should be allowed to choose his own privacy degree. The authentication protocol should support the trade-off between the privacy degree and resource utilization according to the user's specific privacy requirements.
- **Scalability:** VII is designed for nation-wide applications which may involve millions of vehicles and a large number of service providers. As a result, scalability is a key challenge for the design of this system. During traffic congestion, there may be simultaneously a large number of authentication requests delivered to the authentication server. The problem of how to avoid the clogging caused by the burst of the authentication messages should be analyzed and tackled.
- **Real-time response:** VII is designed to collect road condition data as well as provide mobile services to moving vehicles. Both information collection and service delivery have real-time requirements, especially when a vehicle needs immediate help. Because authentication needs to be performed before data can be collected and the service can be delivered, the authentication process has a strict real-time requirement. Furthermore, the fast movement of the vehicles and the small radio coverage range of the roadside units also require that authentication be finished in a very short time. This suggests that the authentication protocols should be light-weight.
- **Data security:** Collected data should be consistent with the raw data on the road. Faked data should be filtered and data modification during transmission should also be prevented. The broadcast nature of wireless communications makes eavesdropping easier, thus, technologies are needed to prevent this kind of attacks. In addition, only authenticated OBUs can use the provided services and OBUs should only access services provided by legitimate service providers. It will be a challenge to detect faked data and locate an attack, especially in the case of anonymous authentication and data reporting.
- **High availability:** Customers of this system may request authentication at anytime and anywhere when they are on the road. Availability is a critical design issue and an important metrics to evaluate the quality of VII. Secure protocols are essential to prevent the attacks that interrupt these services, especially distributed denial of service (DDoS) attacks. Moreover, load balancing algorithms from traditional distributed system research should be applied to balance the load and relief the clogging.
- **Service differentiation:** Various services will be provided by both private service providers (e.g., automakers and other private companies offering services to the vehicles) and public service providers (e.g., government agencies). Those services need to be differentiated based on the priorities of services and the prices that customers have paid. However, there is a dilemma between service customization and user anonymity. On one hand, a good resource allocation algorithm should provide customized services for each individual. On the other hand, differentiating services based on specific customer requirements will violate the anonymity requirement of the system.

In this paper, we intend to address the first three requirements in an authentication protocol, i.e., adaptive privacy, scalability, and timeliness. Other issues will be the objectives of our future work.

## III. Problem Statement

In this section, we use a typical authentication scenario, as depicted in Figure 1, to illustrate the problems we want to solve in this paper. A typical authentication process involves the basic components in VII: the vehicles, the RSUs, and the remote authentication servers. All components can communicate with each other over wireless media or via the Internet. In this paper, we focus on the authentication and secure communication between the vehicles and the RSUs.

VII may support two types of servers: the public servers controlled by government agencies such as the federal or local Departments of Transportation (DoT) and the private servers controlled by the private service providers. Mobile users may want to use different trust policies depending whether they are communicating with a public or private server (or application). These trust policies include 1) *the full-trust* in which the users trust both types of servers, 2) *the partial-trust* in which the users trust the private or public only, and 3) *the zero-trust* in which the users trust neither of these two types of servers. Most previous researches, such as [6], [7], take the partial-trust policy that trusts some public servers. With these approaches, the authentication requests are sent to some anonymity sever first. Then, the anonymity server sends the anonymized or aggregated requests to other service servers. Thus, anonymity is achieved at the anonymity server level. However, we argue that higher degree anonymity is needed from the perspective of mobile users, who do not want the network operators or others to track their daily activities. In the partial-trust model, the trusted servers have the authentication information, e.g., identity of the mobile user, which can be used to easily track the activities of each individual mobile user based on the spatial-temporal analysis such as MTT algorithm [8]. In this paper, we focus on the

zero-trust model, i.e., the users will trust no servers in the network.

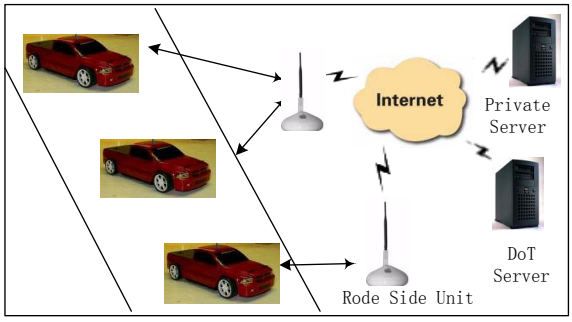



Fig. 1. A typical authentication scenario.

We further observe that the previous efforts in providing privacy treat privacy as a one-size-fits-all requirement. That is, they assume that all users have the same privacy requirements. In reality, however, privacy is a user specific concept. Different users may have different concerns about the privacy. For example, some users may care more about their privacy than receiving certain services, some simply do not wish to reveal their location information unless for emergence reasons, and others may not even be so concerned about location privacy at all. Thus, each user should be allowed to select the degree of privacy that fits his/her own requirements. Furthermore, the degrees of the privacy are usually closely related with the resource usage, e.g., the computational overhead of encryption and decryption, and communication overhead for supporting privacy protocols. Usually, high privacy degree will result in more resource usage. Thus, a trade-off should be made between the degree of privacy and resource usage.

In summary, security and privacy are critical success factors in VII. The design of the privacy protection mechanisms should allow users to decide the degrees of privacy that fit their specific requirements and should achieve a proper balance between privacy protection and resource usage.

## IV. Privacy-preserving Authentication

In this section, we solve the problem raised in the previous section by proposing an adaptive group-based authentication protocol.

### A. Rationale for Adaptation

A higher level of privacy requirement usually results in more overheads. The overheads include 1) the computational overheads to encrypt, digitally sign, and decrypt messages; and 2) the communication overhead to transport authentication messages and encrypted (or digitally signed) user data. These authentication overheads also affect the performance of the system, e.g., both communication and computation take time to finish, thus, more communication and computation

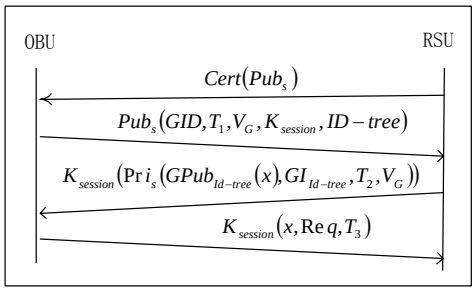



Fig. 2. The message flow of the authentication protocol.

make it more difficult to meet the delay requirements of the real-time applications and to achieve high system scalability. Thus we propose an adaptive protocol to tradeoff the level of privacy and the authentication overhead.

Adaptive privacy is aimed at satisfying mobile users' privacy requirements and reducing communication and computational overheads. The communication overhead is determined by the amount of encrypted data that is transported. The computational overhead is determined by the amount of the encrypted data and the number of the verified common secrets shared among all valid members in the group. We can calculate the number of messages to be delivered and the number of common secrets to be verified based on the mobile user's privacy requirements. And these numbers can be used to set the values of the parameters in the privacy protection protocol. Thus, different privacy requirements can be mapped accordingly into resource usage cases. With adaptation, a user may still get a lower privacy degree when there is insufficient resources to support higher privacy degrees. High privacy degrees can be achieved when sufficient resources are available at the expense of increased resource usage and potentially longer delays. We use a tuple of $< P, PL >$ to describe the mobile user's privacy requirements, which means that the privacy degree $PL$ can be expected with probability $P$. For example, if a mobile user requires $< 0.99, 79 >$, he/she expects a privacy degree of 79 with $99\%$ guarantee. Later on, if the mobile user (or device) finds that the authentication process consumes too much resource or is too slow, he (or the mobile device) may reduce the privacy requirement to $< 0.99, 50 >$ (see analysis in Section IV-C). Thus, the privacy requirements can be satisfied and the communication and computation overhead can be reduced as much as possible.

### B. Anonymous authentication

To support privacy in the context of the zero-trust model, we propose a group-based anonymous authentication protocol as depicted in Figure 2. In the group-based protocol, the authentication requester only need to be verified that it is a valid member of a group, and the authentication server treats every member in the group the same, because the shared com-

mon information among all the group members make them indistinguishable from the authentication server's view. To support mutual authentication and to prevent server probing, each member in a group should maintain a set of necessary information, $GID$, the ID of the group it is in; $GPub$ and $GI$, a set of public keys $Pub_1, Pub_2; \ldots, Pub_{|G|}$ for all members in the group it belongs to and the corresponding index of each key in the group, where $|G|$ denotes the size of the group $G$; $V_G$, the version of the group; $Pri_i$, the private key of the $i^{th}$ member itself. In our design, the set of public keys, $GPub$, for the whole group are organized and stored as an ordered list. To provide flexible subgroup organization, we build a complete binary tree over the ordered list, in which the public keys are leaves in the tree and internal nodes are IDs to identify each subtree. Then, each subtree root can be efficiently expressed with a binary number. Assigning each pair of keys a key index, we can mark the encryption using this key with its key index, so others can find the appropriate key based on the key index to decrypt it.

The authentication protocol consists of five steps and the details of each step are described as follows. First, the RSU broadcasts its ID along with its public key $Pub_s$ signed by the Certificate Authority (CA). Using the public key of the Certificate Authority, an OBU can verify the authenticity of RSU's identity. Then, the RSU's public key is stored temporally and locally by the OBU.

Second, when the OBU wants to send data or request a service from the server, an authentication request will be send to the RSU. The request message includes several items, $K_{session}$, a session key randomly generated by the OBU to be used in the rest of authentication, $GID$, the ID of the group, $V_G$, the group version, $T_1$, a timestamp of the message to prevent reply attack, and $ID-tree$, the root of a sub-tree of the group key tree. The sub-tree is required to include the ID of the OBU and its depth can be decided by the mobile user according to his privacy requirements. For example, if the mobile user wants very high privacy, he can set the value of $ID-tree$ as the root of the group key tree, thus the maximum privacy can be expected. If the user does not care about the privacy, he can set the value of $ID-tree$ as his own ID; However, this may jeopardize other users' privacy. For example, if the RSUs know that user A is at location 1, another user in the same group, authenticated at location 2, cannot be A. For this reason, it is not allowed to use a user's own ID for authentication. In most cases, the depth of the sub-tree should be larger than a certain number and the mobile user can preset a range of acceptable valid depth based on his privacy requirements. Then, in each authentication, the mobile user can randomly pick a sub-tree with the depth in the range. All these information is encrypted by the RSU's public key, $Pub_s$, and sent to the RSU.

After the RSU receives and decrypts the authentication request message, it will first check the validity of the group version, $V_G$. If the group version is inconsistent, a group updating is necessary to keep the key consistency. Then, it validates the timestamp, $T_1$, to prevent reply attack. Next, the RSU creates a challenge to test the validity of the membership of the OBU based on $GID$ and $ID-tree$. This challenge is a set of encrypted values of a random value $x$, each of which is generated by encrypting the value $x$ by one public key in the sub-tree with root $ID-tree$ specified by the OBU. Thus, the challenge is denoted as $GPub_{ID-tree}(x) = Pub_{ID-tree}(x), Pub_i(x), ...,$ where $i$ specifies the other leaves in the tree. The challenge is also used as the common secret set used by the OBU to verify the validity of the RSU. The RSU builds the challenge message that includes the challenge itself, the corresponding group of key index $GI_{ID-tree}$, a timestamp $T_2$ and the correct group version $V_G$. After the RSU signs the challenge message, it encrypts the message using the session key received from the OBU and sends resulting message back to the OBU.

Forth, the OBU receives the challenge message from the RSU and decrypts it to get the challenge using the session key and the RSU's public key. The OBU also checks the validity of the timestamp and the group version $V_G$. If it receives a new $V_G$, a group updating is processed. To complete the process, the OBU need to solve the challenge by picking out the encrypted value with corresponding key index, $I_{OBU}$, and decrypting it use its private key, $Pri_{OBU}$. Thus, it gets the answer of the challenge, $x = Decrypt(Pub_{OBU}(x), Pri_{OBU})$. Then, a group updating request is sent to the RSU. In the case of the same $V_G$, the OBU still need to find the answer to the challenge. Next, it checks the common secret to prevent from the RSU's active probing, because the RSU may encrypt different random values for different keys in the group to identify the ID of the OBU based on different answers. To verify that, the OBU calculates a common secret set locally by using keys in the key tree rooted by $ID-tree$ to encrypt $|x|$. Then it compares the two sets of common secret. If those two reflect the same, the RSU is trustable; otherwise, malicious. After the verification, the answer for the challenge together with the type of the request, $Req$, and $T_3$, are sent from the OBU to the RSU, which are also encrypted using the session key.

Finally, the RSU decrypts the answer message to get the value of $x$. If this recovered $x$ is the same as it encrypted, the OBU is authenticated; otherwise, the authentication fails. A successful authentication results in a data reporting, a service downloading, or a group information updating.

In the proposed protocol, we assume that the mobile user's privacy requirements have already been mapped to the protocol parameters, which is precalculated and stored locally. The communication overhead is determined by the depth of the sub-tree, which constraints the number of encrypted and transferred data. The computational overhead of the server is also decided by the depth of the sub-tree, because the RSU have to encrypt the random value using all keys in the sub-tree. The computational overhead of the OBU is decided by

both the depth of the sub-tree and the probability to check each common secret.

### C. Probabilistic Verification

Because the verification of common secret involves the asymmetric key encryption, which needs heavy computational resources, we tradeoff the privacy degree and the computational overhead by verifying each item in the common secret set with a certain probability. For example, if the common secret set includes 100 encrypted values, and a random subset of 20 encrypted values is verified, a privacy degree of 79 can be provided with probability 0.99. As a result, about 30 encrypted values are verified in average, and the computational overhead is reduced significantly. Next, we analyze the effect of the probabilistic verification for attack scenarios with and without server active probing.

Without server probing, probability verification will not affect the privacy because the same value is encrypted in all common secrets. We analyze the privacy in this case. In the proposed protocol, the OBU may choose a different sub-tree in each authentication, making it difficult for the RSU to link these trees to figure out the identity of the OBU by using the spatial-temporal analysis. The RSU has to guess the ID of the OBU. The privacy that can be expected is determined by the number of nodes in the tree, denoted as $|T|$ and bounded by the depth of the tree, $d$, where $2^d \leq |T| \leq 2^{d+1}$. The probability of successfully guessing the vehicle's identity is the multiplicative inverse of the number of nodes in the sub-tree. For a sub-tree with $|T|$ nodes, the RSU can only have $1/|T|$ probability to identify who is talking with him even when the authenticator knows which vehicles are in this group. The maximum privacy can be expected when the RSU is not active probing the identity of the OBU.

If the RSU wants to identify the OBU exactly, it can use a different random value for encryption with each member's public key. Then, the RSU can know the identity of the OBU based on the value returned by the OBU. We call it probing by the RSU. In such a case, OBU can easily detect the encrypting its decrypted value with another member's public key and comparing the encrypted value with the received value for that member. So, let's assume that the RSU is trying to decrease the anonymity degree of the OBU by using the same number for a subgroup of $s$ slots, which is not distinguishable to the RSU. In this case, if the OBU only verifies $m$ slots where $m < s$, the probing may not be detected by the OBU. However, we will show that the probability $Pr$ of the probing not being detected is very small. Assume that there are $|T|$ members in the sub-tree. Then, the probability of probing not being detected is (please see Appendix for derivation)

$$Pr = \frac{C_{s-1}^{m}}{C_{|T|-1}^{m}} = \frac{(s-1)(s-2)\cdots(s-m)}{(|T|-1)(|T|-2)\cdots(|T|-m)} \quad (1)$$

We define the $anonymityreductionfactor\ r$ as the ratio of the size of the subgroup to the size of the original group:

$$r = \frac{s}{|T|} \quad (2)$$

We have:

$$\frac{s-i}{|T|-i} < \frac{s}{|T|} \text{ for } i > 0 \quad (3)$$

So,

$$Pr < \left(\frac{s}{|T|}\right)^{m} = r^m \quad (4)$$

This means that the probability of successful probing by the RSU decreases exponentially as the number of verifications done by the OBU increases. The rate of decreasing is proportional to the anonymity reduction factor. So the RSU can not reduce the anonymity significantly. Otherwise, the probing by the RSU is easily detected. For example, let $m = 20$, $Pr = 0.01$, then $r$ must be greater than $0.79$. In this case, if $|T|$ is 100, then the OBU can be confident that it can detect the reduction of the anonymity up to 79 with probability $0.99$ even only $20$ slots are verified.

The above analysis shows that if we use a probabilistic anonymity definition, it is more flexible in choosing the protocol parameters. So we define the privacy as a tuple, $< P, PL >$, where $PL$ is the privacy degree and $P$ is probability that any reduction to the anonymity degree can be detected. We have shown that there is a mapping between the expected anonymity, which is $< P, PL >$, and the two parameters in the protocol, the number of the nodes in the tree, $|T|$, and the number of verifications done by the OBU, $m$. Thus, based on these two parameters we can estimate how much privacy can be expected by the mobile user. Based on the expected privacy requirements, the mobile user can set up the correlation between the two parameters. Based on Formula (4), we can deduce the relationship between these four parameters as listed below:

$$P = 1 - Pr > 1 - \left(\frac{PL}{|T|}\right)^{m} \quad (5)$$

where the left part of the formula specifies the expected probability, $P$, of the expected privacy degree, $|PL|$. And the right part of the formula shows the probability of expected privacy degree based on the calculation of the system parameters. Given specific values of $m$ and $|T|$, high anonymity degree can be achieved with low probability and low anonymity degree can be achieved with high probability. The relationship between $P$ and $PL$ is depicted in Figure 3 with fixed value of $|T|$ and $m$.

In Figure 3, the x-axis is the desired probability to detect the reduction while the y-axis is the corresponding maximal reduction the RSU can achieve. For a certain anonymity degree, a further reduction can be achieved with even higher probability of being detected. However, we want to have a strong guarantee for the anonymity degree we choose. So it is desired that there will only a slim change for further

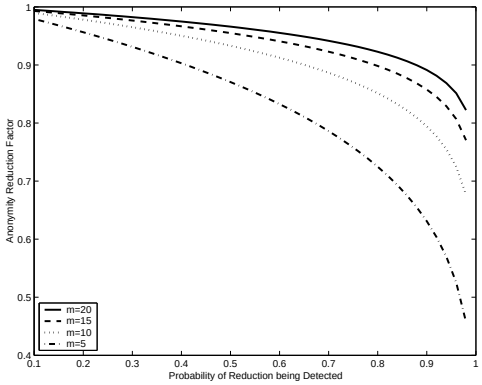



Fig. 3. The relationship between probability of detecting reduction and anonymity reduction factor.

reduction. In Figure 3, this is shown as the right part of a curve. For all values of $m$, the Anonymity Reduction Factor drops rapidly when $p$ is greater than 0.9. Based on this, we suggest to use $P \geq 0.9$. As shown by the previous example, significant computation cost can still be saved for this choice of $P$.

If we determine the value of $P$ and $PL$, we can find the relationship between the $|T|$ and $m$,

$$\left(\frac{PL}{|T|}\right)^m < 1 - P \tag{6}$$

To achieve a given anonymity degree $PL$ and a probability $P$ to detect any reduction, the OBU can either choose a large $|T|$ or a large $m$. The relationship is shown in Figure 4.

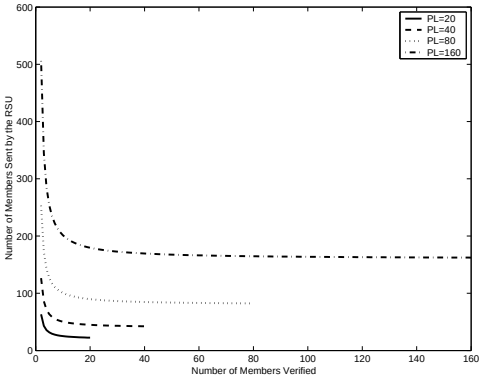



Fig. 4. The relationship between the number of members verified and the number of members sent. We use $P = 0.9$ for all cases.

In Figure 4, the x-axis is the number of members being verified while the y-axis is the corresponding total members that should be requested from the RSU. It shows that when the OBU only verifies 20 members, the number of members being sent approaches the minimum required values in all four cases. It also shows that the increased number of members being sent is exponential to the decreased number of members being verified for small values of $m$. So it is generally preferable to use a reasonably large $m$. It also shows the effectiveness of the probabilistic verification and discourages the RSU from reducing the anonymity degree since the reduction can easily be detected by the vehicle with much lower cost.

The above analysis assumes that the vehicle requires a certain anonymity degree and investigates the trade-off between privacy degree and the communication and computational costs. It is our conjecture that, to reduce both communication cost and computational cost, the anonymity degree has to be reduced. The reduced anonymity degree can be specified with a reduced $P$, a reduced $PL$, or both.

### *D. Group Management*

In the group-based authentication approach, group management is a challenge especially in a large system. Group management includes two stages: initialization and dynamic group management. Note that, for each group, we limit the size of the group to be between $n_1$ and $n_2$, and we use a central server to manage the group information.

In the stage of group initialization, the server first estimate the needed number of the groups, $N_g$, based on the estimated number of the total potential vehicles, $N_v$, and the minimum and maximum numbers of vehicles in one group, $n_1$ and $n_2$. Thus, the $N_g$ should satisfy $N_g = \frac{1}{2}(\frac{N_v}{n_1} + \frac{N_v}{n_2})$. For each group member, we build a pair of public/private keys, assign an index for that key pair, and maintain a group version. These members are kept as dummy members before they are assigned to new vehicles, which is usually done by the second level key distribution servers. Thus, the central server has no idea about which key is already issued, although there are a lot of dummy keys in each group, assuming that the key distribution server and the central server are not cooperating. When they cooperate, we can delay the function of the whole system until all groups have enough keys distributed. After initialization, all the keys in the group are organized to a complete binary tree, whose breadth-first travel results in an ordered sequence of the corresponding key index.

The dynamic group management is conducted in the following ways. When the keys are revoked, the previous member which hold the key is no long valid. Thus, the central server will replace the invalid key using a new key pair and update the group version. When new member joins in, assuming the central server has the information about the number of dummy keys in each group, it will be put to a group with most dummy keys, and the server will find the first dumb or empty slot in the key tree based on a breadth-first search and distribute the corresponding private key, the key index, the group version, and the public key of the whole group, to the new joined member. The updated group information should be distributed to the authentication server and other members in the group, which is a challenge in a large distributed system such as vehicular networks. Fortunately, we can assume that the membership updating

is not so frequently, and we can also take advantage of some existing approaches proposed in distributed system research to keep the consistency among different replicas of the group information. A push approach is used to update the group information to the cashed servers. The updating of group information to the group member is integrated with the process of the authentication. A group information updating is conducted if these two versions do not match and the membership is verified during authentication.

### *E. Intrusion Isolation and Key Revocation*

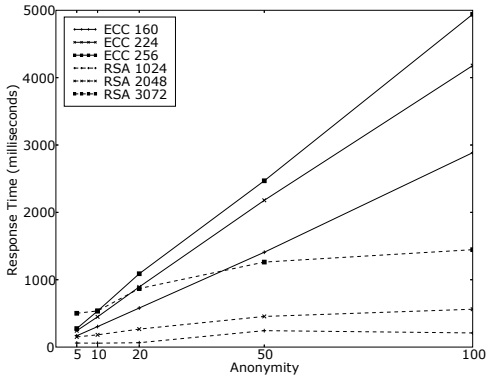
Intrusion detection is necessary to identify an attack. When an attack is detected, we could leverage several communication techniques such as DSRC [5] and GPS to locate the attacker based on the communication between the attack and the server. When the attacker is located, several follow-up actions can be taken. For example, a message can be sent to a pre-installed device in the vehicle to forbid its wireless communication; under proper situations, a message can be sent to the attack source to disable the vehicle; or a policeman can be sent to that location to catch the attacker;.

It is not sufficient to only isolate the attack because the attacker may try to authenticate himself using the same key at a different location or on another vehicle. Thus, key revocation is necessary. In our design, key revocation is integrated seamlessly with the group management and the process of authentication. When an attacker is reported, the public key of the attacker will be removed from the group. When the attacker uses the revoked key, the RSU will send it the challenges encrypted by the valid public keys in the group. Since the attacker's public key has already been removed from that set, the attacker will not be able to decrypt the challenge. In this way, the key is automatically revoked and other members in the group will not be affected.

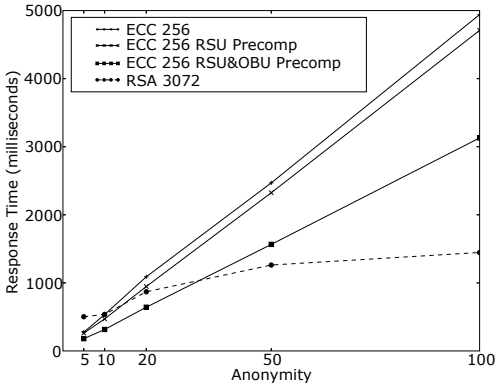
## V. Performance Evaluation

We implemented the proposed protocol with the Crypto++ [9] package. Since the number of common secret is proportional to the size of the sub-tree, to minimize communication delay, we use a public key cryptography with the most security per bit - Elliptical Curve Cryptography (ECC). For comparison, we also evaluate the performance of the proposed protocol implemented using the widely used RSA cryptography. The result is shown in Figure 5, where the x-axis is the number of users (privacy degree) in a tree.

The time spent on the authentication using ECC is linear to the privacy degree. This is expected since in this case cryptography computation dominates and is proportional to the number of members in the group. RSA-based authentication performs better than ECC-based schemes. The reason is that most computation involved in the authentication is public key encryption. ECC public key encryption is considerably slower than comparable RSA public key encryption with the same security strength. This is shown in Crypto++ benchmarks [9]. However, RSA-based authentication generates much longer packets. To roughly estimate the communication cost, the number of DSRC packets required for transmitting the common secret for two different anonymity degrees and different cryptographic key sizes are shown in Table I. A 3072 bit RSA-based signature for a group with 100 members uses about 98 DSRC packets. So it is more likely to suffer from packet losses on wireless communication channels.




(a) Response time comparison between ECC and RSA




(b) ECC performance improvement through pre-computation

Fig. 5. Response time evaluation.

TABLE I
SIZE OF THE COMMON SECRET IN DSRC PACKETS FOR 100 MEMBERS.

| Public Key Cryptography | Number of Packets for 100 Members | Number of Packets for 50 Members |
|---|---|---|
| ECC 160 | 21 | 10 |
| ECC 224 | 30 | 15 |
| ECC 256 | 34 | 17 |
| RSA 1024 | 32 | 16 |
| RSA 2048 | 64 | 32 |
| RSA 3072 | 98 | 49 |

Figure 5(b) shows the performance improvement of 256 bit ECC through pre-computation. For comparison, we evaluated two configurations of pre-computations. One is to pre-

compute at only the authentication server. The other is to precompute at both the server and the OBU. It can be clearly seen that the computation on the OBU is the bottleneck of the protocol. As we showed in Section IV-C, verifying only 20 slots is enough for providing a higher level of anonymity with high probability. Our probabilistic verification not only provides flexibility but also favors vehicle side implementation.

These preliminary results show that the proposed protocol can meet the design goals of achieving trade-offs between the privacy degree and overheads and being responsive to different system conditions. We are currently carrying out further evaluations taking into consideration more system factors, such as mobility, wireless channel contention, storage requirements, and so on.

## VI. Related Work

A lot of work has been done to build vehicular networks, most focusing on the design of MAC layer protocols [4], routing protocols [10] based on DSRC [5] and potential applications [11]. However, few efforts address security and privacy issues. Several related efforts are listed as follows.

Attack models and requirements, with some general approaches to prevent these attacks, are described in [12] in detail. The protocol proposed in this paper can be used to prohibit most of these attacks. [13] addresses some security issues in vehicular networks focusing on system design, but lacks in-depth analysis of privacy protection. Security and privacy of smart vehicles are studied in [7]. This work proposes to use electronic license plates and tamper-proof GPSes to preserve security and privacy, which can be used in our design to strengthen the security and privacy; so their work complements our protocol. Raya and Hubaux also explore the security issues in vehicular ad hoc networks [14]. They analyze attack models and some concrete attacks, then propose a set of security protocols for vehicular ad hoc networks. They also design a key changing algorithm to preserve anonymity and minimize the storage costs of the public keys. However, we use group-based authentication protocol to preserve anonymity which is different from them.

Our authentication protocol is close to a previous group-based approach proposed in [15]. We share the same goal of achieving anonymous authentication based on group information. However, we differ in protocol design. In particular, this paper presents a protocol to support adaptive privacy by making a trade-off between the privacy and the resource usage. Furthermore, our protocol design is closely integrated with the system design of VII, while theirs is more general. $K$-anonymity for location privacy is proposed in [6], which anonymizes users at the authentication server so that it is suitable to be applied in the *partial trust* case that the RSU is trustable. Our protocol provides adaptive privacy without the requirement of trusting RSUs, i.e., our proposed protocol supports the *zero-trust* model. Ren *el al.* propose a privacy preserving authentication in [16] that uses blind signature and one-way hash chain to keep privacy. However, their approach does not support adaptive privacy.

## VII. Summary and Future Work

In this paper, we analyze the requirements and challenges of providing security and privacy in VII, and identify the importance of achieving adaptive privacy in the *zero-trust* model. Then, an adaptive, group-based, privacy-preserving authentication protocol is proposed to tradeoff the privacy and the resource usage. Both analytic results and preliminary simulation results show the feasibility of our protocol.

Next, we will extend our work in two-fold. First, we will give a comprehensive system level performance evaluation to our protocol by considering more system factors. Second, we will try to further reduce the overhead in the authentication caused by the asymmetric key encryption and decryption.

## References


[1] "Vehicle infrastructure integration." [Online]. Available: http://www.its.dot.gov/vii/docs/vii_factsheet.pdf

[2] R. Bishop, "A survey of intelligent vehicle applications worldwide," in *in Proc. of IEEE intelligent Vehicles Symposium 2000*, Oct. 2000.

[3] ITSA and DoT, "National intelligent transportation systems program plan: A ten-year vision," 2002, a Report from Intelligent Transportation Society of America and Departemnt of Transportation.

[4] T. Mak, K. Laberteaux, and R. Sengupta, "A multi-channel vanet providing concurrent safety and commercial services," in *in Proc. of the 2nd ACM international workshop on Vehicular ad hoc networks*, Sept. 2005.

[5] "Dedicated short range communications (dsrc) home." [Online]. Available: http://grouper.ieee.org/groups/scc32/dsrc/

[6] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *in Proc. of the 25th International Conference on Distributed Computing Systems*, June 2005.

[7] J.Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security and Privacy*, vol. 4, no. 3, pp. 49–55, 2004.

[8] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, 1979.

[9] "Crypto++ library 5.2.1." [Online]. Available: http://www.eskimo.com/~weidai/cryptlib.html

[10] T. Munaka, T. Yamamoto, and T. Watanabe, "A reliable advanced-join system for data multicasting in its networks," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, vol. 6, no. 4, pp. 424–437, Dec. 2005.

[11] "Trafnet: Real-time seattle area traffic conditions over the internet." [Online]. Available: http://www.its.washington.edu/trafnet/

[12] A. Aijaz *et al.*, "Attacks on inter-vehicle communication systems - an analysis, Tech. Rep. Technical Report, Jan. 2005.

[13] M. Zarki, S. Mehrotra, G. Tsudik, and N. Venkatasubramanian, "Security issues in a future vehicular network," in *in Proc. of EuroWireless 2002*, Feb. 2002.

[14] M. Raya and J. Hubaux, "The security of vehicular ad hoc networks," in *in Proc. of the 3rd ACM workshop on Security of ad hoc and sensor networks*, Nov. 2005.

[15] S. Schechter, T. Parnell, and A. Hartemink, "Anonymous authentication of membership in dynamic groups," in *in Proc. of the third international conference on financial data security and digital commerce*, Jan. 1999.

[16] K. Ren *et al.*, "A novel privacy preserving authentication and access control scheme for pervasive computing environments," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 4, pp. 1373 – 1384, 2006.