APPA: Aggregate Privacy-Preserving Authentication in Vehicular Ad Hoc Networks

Lei Zhang[1], Qianhong Wu[2,3], Bo Qin[2,4], and Josep Domingo-Ferrer[2]

[1] Software Engineering Institute, East China Normal University, Shanghai, China
leizhang@sei.ecnu.edu.cn
[2] UNESCO Chair in Data Privacy, Dept. of Comp. Eng. and Maths
Universitat Rovira i Virgili, Tarragona, Catalonia
{qianhong.wu,bo.qin,josep.domingo}@urv.cat
[3] Key Lab. of Aerospace Information Security and Trusted Computing
Ministry of Education, School of Computer, Wuhan University, China
[4] Dept. of Maths, School of Science, Xi'an University of Technology, China

Abstract. Most security- and privacy-preserving protocols in vehicular *ad hoc* networks (VANETs) heavily rely on time-consuming cryptographic operations which produce a huge volume of cryptographic data. These data are usually employed for many kinds of decisions, which poses the challenge of processing the received cryptographic data fast enough to avoid unaffordable reaction delay. To meet that challenge, we propose a vehicular authentication protocol referred to as APPA. It guarantees trustworthiness of vehicular communications and privacy of vehicles, and enables vehicles to react to vehicular reports containing cryptographic data within a very short delay. Moreover, using our protocol, the seemingly random cryptographic data can be securely and substantially compressed so that the storage space of a vehicle can be greatly saved. Finally, our protocol does not heavily rely on roadside units (RSUs) and it can work to some extent even if the VANET infrastructure is incomplete. These features distinguish our proposal from others and make it attractive in various secure VANET scenarios.

Keywords: Traffic Security, VANETs, Privacy, Protocol Design, Data Compression.

1 Introduction

With the fast development of mobile networks and information processing technologies, vehicular *ad hoc* networks (VANETs) have attracted in recent years particular attention in both industry and academia. A VANET mainly consists of vehicles and properly distributed roadside units (RSUs), both equipped with onboard sensory, processing, and wireless communication modules. The DSRC standard [1] is suggested to support wireless communication in VANETs. In addition to messages routed to/from RSUs, vehicles can also broadcast safety messages concerning accidents, dangerous road conditions, sudden braking, lane changing, etc., in a one-hop or multi-hop fashion. With these mechanisms, VANETs are

X. Lai, J. Zhou, and H. Li (Eds.): ISC 2011, LNCS 7001, pp. 293–308, 2011.

expected to enhance the driving experience by improving road safety and traffic efficiency, as well as supporting value-added applications such as automatic toll collection, infotainment, context-oriented personalized services and so on.

Despite the great potential benefits of VANETs, many challenges arise around this technology, especially in what regards security and privacy. VANETs aim at a safer driving environment by allowing vehicle-to-vehicle (V2V) and vehicle-to-RSU (V2R) communications. However, selfish vehicles can also exploit this mechanism to send fraudulent messages for their own profit. Malicious vehicles may impersonate innocent ones to launch attacks without being caught. To address the security requirements in VANETs, digital signatures are usually employed so that the receiving vehicles can verify that these messages have been originated by authentic sources and have not been modified during transmission. Driving privacy is another critical concern in VANETs. Although employing signatures can mitigate the security challenge in VANETs, the presence of signatures in vehicle-generated messages might allow attackers to identify who generated those messages. Since vehicular messages contain speed, location, direction, time and other driving information, a lot of private information about the driver can be inferred, which jeopardizes vehicle and driver privacy.

To address security and privacy concerns, a large body of proposals have striven to secure VANETs (*e.g.*, [5,15,17,18,23,25]). Most proposals heavily rely on time-consuming cryptographic operations which produce a huge volume of cryptographic data. These data are usually employed for many kinds of decisions. Indeed, according to DSRC [1], a vehicle will broadcast a (signed) message to nearby vehicles and/or RSUs every few hundreds of milliseconds. Hence, a vehicle or an RSU may receive hundreds of messages in a short period of time, all of those should be verified in real time. Otherwise, the delay caused by verification of a bulk of signatures may radically impair transmission throughput and system scalability. This poses the challenge of processing the received cryptographic data fast enough to avoid an unaffordable reaction delay, which might result in traffic jams and even accidents.

Furthermore, if some signed messages are later found to be false and have misguided other vehicles into accidents, the message generators and endorsers should be traceable. This implies that the signed vehicle-generated messages have to be stored by the receiving vehicles. However, vehicular messages, especially their appended cryptographic data (being almost random), grow linearly with time. Hence, It's preferable if the vehicular communication protocol can securely compress those cryptographic data.

1.1 Our Results

We propose a security- and privacy-preserving protocol referred to as APPA. The APPA protocol is built on our new notion of one-time identity-based aggregate signature (OTIBAS). This notion incorporates the desirable properties of identity-based cryptography, aggregate signature and one-time signature. In an OTIBAS scheme, a user can compute a signature on a message only if the user has obtained a secret key from a trusted authority, where the secret key is

associated with the user's identity. The signature can be verified by anyone who knows the user's identity. Since the user's identity works as her public key, no certificate is needed on the public key, unlike in the conventional PKI setting, which avoids the certificate management overhead. As a one-time signature, the signer cannot use one secret key (corresponding to one identity of the user) to generate no more than one signature, and accordingly, the user's identity should change for every signature and can be viewed as a one-time pseudonym. This feature is well matched to the anonymity requirement of vehicles in VANETs. As an aggregate signature scheme, the signatures on n messages by n signers can be aggregated into one signature which can be verified as if it had been generated by a single signer. This feature caters for the requirements of fast response in VANETs and can be used to save storage space in a vehicle. By exploiting bilinear pairings, we propose an efficient OTIBAS scheme. The security of the proposed OTIBAS scheme is formally proven under the computational co-Diffie-Hellman (co-CDH) assumption. With our provably secure OTIBAS scheme and the *multiplicative secret sharing* technique introduced by Kiltz and Pietrzak [6], we realize the APPA protocol which is efficient and practical to secure vehicular communications.

The APPA protocol exhibits a number of attractive features. The proposal preserves privacy for honest vehicles. However, if a vehicle authenticates a bogus message, then it will be caught for penalty. This mechanism guarantees trustworthiness of vehicular communications and it is a deterrent to malicious vehicles. Furthermore, our protocol enables vehicles to react to vehicular messages within a very short delay and the seemingly random cryptographic data can be securely and substantially compressed: any number of signatures can be compressed into a single group element of about 21 bytes without degrading security. Finally, our protocol does not heavily rely on RSUs, a part of the VANET infrastructure, and it can work to some extent even if the VANET infrastructure is incomplete, *i.e.*, still under construction or partially destroyed by, *e.g.*, an earthquake. These features facilitate deployment in various environments to secure VANETs.

1.2 Related Work

A large number of papers deal with the security and privacy challenges in VANETs [5,15,17,18,23,25]. Among them, pseudonym-based authentication is a popular research line. In [3,17,18], digital signatures for message integrity, authentication and non-repudiation are combined with short-lived anonymous certificates to guarantee the security requirements for the vehicular nodes. However, in this approach, each vehicle needs to pre-load a huge pool of anonymous certificates to achieve privacy of vehicles, and the trusted authority also needs to maintain all the anonymous certificates of all the vehicles, which incurs a heavy burden of certificate management.

To relieve from the certificate management burden, several proposals [12,13] exploit group signatures [4] and present conditional privacy-preserving vehicular authentication protocols. Subsequent efforts have been made to improve the trustworthiness of vehicle-generated messages [23] or to achieve robustness

and scalability in VANETs [27]. However, due to the difficulty of dealing with revoked and compromised signers in group signatures (an open question in cryptography), these systems may degrade in performance as the number of revoked/compromised vehicles grows with time.

Another approach to avoid certificate management is to exploit identity-based cryptography [20]. In [25], Zhang *et al.* designed an efficient conditional privacy-preserving protocol for vehicular communications using identity-based cryptography. Their protocol requires the master secret key of the system to be stored in an *idealized* tamper-proof device embedded into vehicles. This device is assumed secure against any attempt of compromise in any circumstance and an attacker cannot extract any data stored in the device. This assumption seems too strong to be met in practice. For instance, without probing into the device, an attacker might collect some side-channel information leaked by cryptographic operations in the device. This kind of attacks are known as side-channel attacks [11,21] and they seem attractive for organized criminals: if the tamper-proof device is compromised, the criminals obtain full control of the system.

Compared to the intensive attention received by security and privacy issues, few efforts have been made to aggregate vehicle-generated messages and cryptographic witnesses to save response time and storage space for vehicles. Picconi *et al.* proposed a PKI-based authentication scheme in VANETs [16]. Their scheme focuses on aggregating messages, rather than aggregating cryptographic witnesses; aggregating the latter is more challenging, because they are almost random. As noted by themselves, their solution suffers from some limitations. In [28], Zhu *et al.* apply short signatures in the PKI setting to aggregate *emergency messages* in VANETs. More recently, Wasef *et al.* [22] employed aggregation technologies to enable each vehicle to simultaneously verify signatures and their certificates in the PKI scenario. Since these schemes are implemented in the PKI setting, certificate management remains a problem. Furthermore, although some general privacy ideas have been discussed in these proposals, it is unclear how they can technically achieve reasonable vehicle privacy.

1.3 Paper Organization

The rest of the paper is organized in the following way. In Section 2, we describe the system architecture and the goals of our design. Section 3 proposes an identity-based aggregate one-time signature scheme as the building block. Section 4 proposes our APPA protocol. We evaluate the new protocol in Section 5. Finally, Section 6 concludes the paper.

2 System Architecture and Design Goals

2.1 System Architecture

As shown in Figure 1, the system architecture in our proposal consists of a trusted authority (TA), a number of RSUs and numerous vehicles.

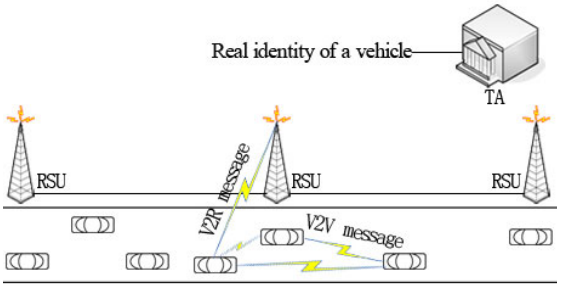

Fig. 1. System architecture

- TA: It generates the system parameters. When an authenticated message is found false, TA might be asked to trace the vehicle having generated that cheating message.
- RSUs: They are distributed along the road side as part of the infrastructure of VANETs. RSUs are equipped with on-board sensory, processing, and wireless communication modules. In our system, they are mainly used to collect/forward data from vehicles or distribute global broadcast from TA to vehicles. We enforce minimum security-related operations on RSUs so that the system can work even if the infrastructure is incomplete at the early stage of VANETs or it is destroyed by some disaster.
- Vehicles: They are equipped with on-board sensory, processing, tamper-proof devices and wireless communication modules. Vehicles move along the roads, and periodically exchange messages with nearby vehicles and RSUs within their communication range.

2.2 Design Goals

Our APPA protocol has the following design goals.

- **Global security.** The vehicle-generated messages should be authenticated to guarantee that they are from real sources and have not been tampered with during transmission. If a vehicle authenticates a bogus message, it must be caught and it must be unable to deny authorship on the cheating message.
- **Individual privacy.** If a vehicle behaves honestly and follows the APPA protocol, its privacy should be guaranteed against attackers who can eavesdrop communication in VANETs.
- **Easy deployment.** The protocol should allow vehicles to quickly react to received messages. It is preferable if there is no heavy overhead incurred by certificate management and the storage requirement of digital signatures is minimized. The protocol should work to some extent in case that the infrastructure is incomplete.

3 The Building Block: OTIBAS

3.1 Modeling OTIBAS

An OTIBAS scheme consists of the following efficient algorithms: Setup, Extract, Sign, Aggregate, and Verify. Setup takes as input a security parameter and outputs the global system parameters including TA's public key. Extract takes as input TA's master secret key and a signer's identity, and outputs a private key for the signer. Sign takes as input a signer's private key and any message, and outputs a signature on the message. The constraint here is that a private key corresponding to a specific identity can be used to generate only one signature. Aggregate takes as input n message-signature pairs generated in the Sign procedure, and outputs an aggregate signature. Verify takes as input the n messages, the aggregate signature, the n identities corresponding to the n message-signature pairs and TA's public key, and it outputs a bit 1 or 0 to represent whether the original n message-signature pairs are valid or not.

An OTIBAS scheme should be correct in the sense that, if each party honestly follows the scheme, then Verify always outputs 1. An OTIBAS scheme should be also secure. Informally (a more formal definition can be found in Section 3.3), an OTIBAS scheme is said to be secure if any polynomial-time attacker not requesting a private key corresponding to an identity ID^* cannot forge a signature corresponding to ID^* that is aggregated so that Verify outputs 1.

3.2 An OTIBAS Scheme

Our OTIBAS scheme is implemented in bilinear groups [7,14]. Bilinear groups have been widely employed to build versatile cryptosystems [8,24,26].

Let $\mathbb{G}_1, \mathbb{G}_2$ be two cyclic groups of prime order q and $\mathbb{G}_T$ be a multiplicative cyclic group of the same order. Let g_1 denote a generator of $\mathbb{G}_1$, g_2 a generator of $\mathbb{G}_2$, ψ a computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, with $\psi(g_2) = g_1$. A map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is called a bilinear map if $\hat{e}(g_1, g_2) \neq 1$ and $\hat{e}(g_1^{\alpha}, g_2^{\beta}) = \hat{e}(g_1, g_2)^{\alpha\beta}$ for all $\alpha, \beta \in Z_q^*$. By exploiting bilinear groups, what follows implements our OTIBAS scheme.

Setup: TA runs this algorithm to generate the system parameters as follows:

1. Choose $q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \psi, \hat{e}$.
2. Pick $\kappa \in \mathbb{Z}_q^*$ as its master secret key, and compute $y = g_2^{\kappa}$ as its master public key.
3. Select cryptographic hash functions $H_0(\cdot) : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $H_1(\cdot) : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.
4. Publish the system parameter $\Psi = (\hat{e}, q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, y, H_0(\cdot), H_1(\cdot))$.

Extract: Taking as input κ and a signer's identity ID_i, this algorithm outputs the private key for the signer as follows:

1. Compute $id_{i,0} = H_0(ID_i, 0), id_{i,1} = H_0(ID_i, 1)$.
2. Compute $s_{i,0} = id_{i,0}^{\kappa}, s_{i,1} = id_{i,1}^{\kappa}$.
3. Set $s_i = (s_{i,0}, s_{i,1})$ as the private key of the signer.

Sign: To sign a message m_i, a signer with identity ID_i and private key $s_i = (s_{i,0}, s_{i,1})$ computes

$$h_i = H_1(m_i, ID_i), \sigma_i = s_{i,0} s_{i,1}^{h_i}.$$

The signer outputs σ_i as the signature on message m_i. Notice that a signer needs different temporary identities to sign multiple messages, as implied by the name of one-time identity based aggregate signature.

Aggregate: This publicly computable algorithm aggregates n signatures into a single signature. For a set of n users with identities $\{ID_1, \cdots, ID_n\}$, and corresponding message-signature pairs $\{(m_1, \sigma_1), \cdots, (m_n, \sigma_n)\}$, this algorithm outputs

$$\Omega = \prod_{i=1}^{n} \sigma_i$$

as the resulting aggregate signature.

Verify: To verify an aggregate signature Ω on messages $\{m_1, ..., m_n\}$ under identities $\{ID_1, ..., ID_n\}$, the verifier performs the following steps:

1. For $1 \leq i \leq n$, compute $h_i = H_1(m_i, ID_i)$ and $id_{i,0} = H_0(ID_i, 0)$, $id_{i,1} = H_0(ID_i, 1)$.
2. Check
$$\hat{e}(\Omega, g_2) = \hat{e}(\prod_{i=1}^{n} id_{i,0} id_{i,1}^{h_i}, y).$$
Output 1 if the equation holds; else output 0.

3.3 Correctness and Security

The correctness of the OTIBAS scheme in Section 3.2 follows from a direct verification.

In general, the security of an OTIBAS scheme is modeled via the following EUF-OTIBAS-CMA (existential universal forgery under adaptive chosen-message attack) game which is based on the security model of Gentry-Ramzan [8] and takes place between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. The game has the following three stages:

Initialize: $\mathcal{C}$ runs the Setup algorithm to obtain a master secret key and the system parameters. $\mathcal{C}$ then sends the system parameters to $\mathcal{A}$ while keeping secret the master secret key.

Attack: $\mathcal{A}$ can perform a polynomially bounded number of the following types of queries in an adaptive manner.

– Extract *queries*: $\mathcal{A}$ can request the private key of an entity with identity ID_i. In response, $\mathcal{C}$ outputs the private key of the entity.

– Sign *queries*: $\mathcal{A}$ can request an entity's (whose identity is ID_i) signature on a message M_i. On receiving a query on (M_i, ID_i), $\mathcal{C}$ generates a valid signature σ_i on message M_i under identity ID_i, and replies with σ_i.

Forgery: $\mathcal{A}$ outputs a set of n identities $L^*_{ID} = \{ID^*_1, ..., ID^*_n\}$, a set of n messages $L^*_M = \{M^*_1, ..., M^*_n\}$ and an aggregate signature σ^*.

We say that $\mathcal{A}$ wins the above game, iff

1. σ^* is a valid aggregate signature on messages $\{M^*_1, ..., M^*_n\}$ under identities $\{ID^*_1, ..., ID^*_n\}$.
2. At least one of the identities, without loss of generality say $ID^*_1 \in L^*_{ID}$, was not submitted in the Extract queries, and the signature of ID^*_1 can be queried at most only once, and (M^*_1, ID^*_1) was never queried during the Sign queries.

We can now define the security of an OTIBAS scheme in terms of the above game.

Definition 1. *An OTIBAS scheme is secure,* i.e.*, secure against existential forgery under adaptive chosen-message attack, iff the success probability of any polynomially bounded adversary in the above EUF-OTIBAS-CMA game is negligible.*

We next recall the co-CDH assumption on which the security of the OTIBAS scheme in Section 3.2 is based.

Definition 2 (co-CDH Assumption). *The co-CDH assumption in two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q equipped with bilinearity states that, given (g^a_1, g^b_2) for randomly chosen $a, b \in Z^*_q$, it is hard for any polynomial-time algorithm to compute g^{ab}_1.*

Regarding the security of our OTIBAS scheme, we have the following claim.

Theorem 1. *Assume there exists an adversary such that: i) it has an advantage ϵ in forging a signature of the OTIBAS scheme of Section 3.2 in an attack modeled by the above EUF-OTIBAS-CMA game, within a time span τ; ii) it can make at most q_{H_i} times $H_i(\cdot)$ $(i = 0, 1)$ queries, q_E times* Extract *queries, q_S times* Sign *queries. Then there exists an adversary who can solve the co-CDH problem with probability $\epsilon' \geq \frac{1}{e(q_E+q_S+n+1)}\epsilon$ within time $\tau' = \tau + \Theta(4q_{H_1}+q_S)\tau_{G_1}$, where τ_{G_1} is the time to compute a point exponentiation in $\mathbb{G}_1$ and n is the size of the aggregating set.*

Due to page limitation, the proof will be presented in the full version of this paper.

4 The APPA Protocol

4.1 Intuition behind Our Protocol

Our APPA protocol is built on the above OTIBAS scheme. The OTIBAS notion incorporates the desirable features of identity-based cryptography [20], aggregate

signature [8] and one-time signature. In an OTIBAS scheme, any identity string can be the public key of a signer and the signer can only generate a signature after obtaining the private key corresponding to the identity of the signer. This guarantees the security of the VANETs deployment and eliminates the need for an extra certificate for each identity. Since OTIBAS allows a signer to compute only one signature under one identity, the signer's temporary identity changes for each signature and anonymity is naturally achieved for vehicles/signers. However, since TA knows the secret used by the signer, TA can trace a misbehaving vehicle with the signatures generated by the vehicle. Furthermore, OTIBAS can aggregate n signatures on n (distinct or not) messages from n signers into a single signature. This greatly reduces the time to verify n signatures in order and speeds up the reaction of the verifying vehicles to received messages. Hence, the APPA protocol meets our design goals very well.

Our protocol also requires each vehicle to be equipped with a *practical* tamper-proof device. This is used to allow each vehicle to locally generate its temporary identities (as pseudonyms), without frequently contacting TA, and also to relieve from the reliance on RSUs. However, unlike the protocol in [25] storing the system master secret key in the *idealized* tamper-proof device (assumed secure against any attempt of compromise in any circumstance), we just require the tamper-proof device to store a secret identity of the vehicle and some auxiliary secret information, to enable the vehicle to generate its one-time identity (as a pseudonym) and the private key corresponding to this one-time identity. The secret identity is computed by TA from the real identity of the vehicle and TA's secret. That is, the secret identity in the tamper-proof device of each vehicle is different. If a vehicle does not renew its secret identity, it may leave the opportunity to an attacker to recover this secret. In our protocol, each vehicle can update its secret identity information in its tamper-proof device to counter possible side-channel attacks. This is very different from the protocol in [25] in which the secret in the tamper-proof device cannot be updated.

4.2 The Concrete Protocol

The APPA protocol consists of the following five stages.

[System Setup]

At this stage, TA initializes the system-wide parameters as follows:

1. Generate $q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \psi, \hat{e}$, where $q-1$ has a large prime factor (say $q-1 = 2q'$ for a prime q').
2. Pick $\kappa \in Z_q^*$ as its master secret key, and compute $y = g^\kappa$ as its master public key.
3. Select cryptographic hash functions $H_0(\cdot) : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_1(\cdot) : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ and $H_2^{key}(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^l$, where $H_2^{key}(\cdot)$ is a keyed hash.
4. Choose Λ as a hash key of $H_2^{key}(\cdot)$.
5. Pre-load the system parameters $\Psi = (\hat{e}, q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, y, H_0(\cdot), H_1(\cdot), H_2^{key}(\cdot))$ in each vehicle and RSU.

The TA also maintains a member list ML which is kept secret. We will define this list later.

[Vehicle Join]

Each vehicle is equipped with a tamper-proof device. Before a vehicle $\mathcal{V}_i$ joins a VANET, its tamper-proof device should be initialized. The tamper-proof device of $\mathcal{V}_i$ is preloaded with the system parameters Ψ and two secret values (α_i, β_i), where α_i and β_i satisfy $\kappa = \alpha_i \beta_i$. Furthermore, an internal pseudo-identity (IPID) and a hash key λ_i are also preloaded to the tamper-proof device. In the following, we show how to set IPID and λ_i.

Assume that each vehicle is associated with a real identity, *e.g.*, the driving license number. The real identity is used by TA to generate IPIDs for the vehicle. Suppose the real identity of a vehicle $\mathcal{V}_i$ is $ID_{\mathcal{V}_i}$. To generate an IPID for $\mathcal{V}_i$, TA concatenates the real identity and a validity period VP_i, *e.g.*, "01.01.2011-01.02.2011", which is associated with and used to compute the IPID $IPID_{\mathcal{V}_i} = H_2^{\Lambda}(ID_{\mathcal{V}_i}||VP_i)$. TA chooses a hash key λ_i, and stores $IPID_{\mathcal{V}_i}, \lambda_i$ in the tamper-proof device. $(ID_{\mathcal{V}_i}, VP_i, \lambda_i)$ is added to the member list ML.

We notice that, if a vehicle does not update its IPID, its privacy is exposed to potential side-channel attacks. Hence, we suggest each IPID to have a limited life time, and we require the vehicle to periodically renew $(IPID_{\mathcal{V}_i}, \lambda_i)$ before the current IPID expires.

[Vehicle Sign]

At this stage, with the help of the embedded tamper-proof device, a vehicle $\mathcal{V}_i$ computes a signature on a message. This stage has four steps: *Generate public pseudo-identity*, *Extract one-time signing key*, *Sign message* and *Randomize secret values*. The description of each step is as follows:

Generate public pseudo-identity: At this step, $\mathcal{V}_i$ uses its internal pseudo-identity $IPID_{\mathcal{V}_i}$ to generate its public pseudo-identity (PPID). It generates a PPID $PPID_{i,j}$ by computing $PPID_{i,j} = H_2^{\lambda_i}(IPID_{\mathcal{V}_i}, \tau)$, where τ is a timestamp.

Extract one-time signing key: At this step, $\mathcal{V}_i$ generates the private signing key corresponding to the PPID $PPID_{i,j}$. Assuming that the current secret values of $\mathcal{V}_i$ are $\alpha_{i,j}$ and $\beta_{i,j}$, $\mathcal{V}_i$ generates the private key associated with the PPID $PPID_{i,j}$ as follows:

1. Compute $pid_{i,j,0} = H_0(PPID_{i,j}, 0), pid_{i,j,1} = H_0(PPID_{i,j}, 1)$.
2. Compute $s_{i,j,0} = pid_{i,j,0}^{\alpha_{i,j}} pid_{i,j,0}^{\beta_{i,j}}, s_{i,j,1} = pid_{i,j,1}^{\alpha_{i,j}} pid_{i,j,1}^{\beta_{i,j}}$.
3. Set $s_{i,j} = (s_{i,j,0}, s_{i,j,1})$ as the one-time signing key of the vehicle.

Sign message: At this step, $\mathcal{V}_i$ computes and outputs the signature $\sigma_{i,j} = s_{i,j,0} s_{i,j,1}^{h_i}$, where $h_i = H_1(m_i, PPID_{i,j})$.

Randomize secret values: To counteract potential side channel attacks, the vehicle's secret values in the tamper-proof device should be updated. The *multiplicative secret sharing* technique introduced by Kiltz and Pietrzak [6] is employed to achieve the goal. Assuming that the current secret values of the tamper-proof device are $\alpha_{i,j}, \beta_{i,j}$, the device generates the new secret values as follows:

1. Choose a random $r \in \mathbb{Z}_q^*$.
2. Compute $\alpha_{i,j+1} = r\alpha_{i,j}$ and $\beta_{i,j+1} = r^{-1}\beta_{i,j}$.
3. Set $(\alpha_{i,j+1}, \beta_{i,j+1})$ as the new secret values.

[Message Verification and Signature Storage]

Suppose that a vehicle or an RSU receive n message-signature pairs $\{(m_1, \sigma_1), \cdots, (m_n, \sigma_n)\}$ from n vehicles with public pseudo identities $\{PPID_{1,j_1}, ..., PPID_{n,j_n}\}$, respectively. The verifier computes the aggregate signature $\Omega = \prod_{i=1}^{n} \sigma_i$. To verify the aggregate signature Ω, the verifier performs the following steps:

1. For $1 \leq i \leq n$, compute $h_i = H_1(m_i, PPID_{i,j_i}), pid_{i,j_i,0} = H_0(PPID_{i,j_i}, 0)$, $pid_{i,j_i,1} = H_0(PPID_{i,j_i}, 1)$.
2. Output 1 if $\hat{e}(\Omega, g_2) = \hat{e}(\prod_{i=1}^{n} pid_{i,j_i,0} pid_{i,j_i,1}^{h_i}, y)$. Else output 0.

After verifying the aggregate signature, a vehicle or an RSU may store

$$(m_1||...||m_n; PPID_{1,j_1}||...||PPID_{n,j_n}; \Omega)$$

in its local database. One may notice that the last field is of constant size, (the length of one group element, which is argued to be 21 bytes in Section 5.2).

[Trace]

If a message has passed the verification procedure but is found false, TA should be able to trace the real identity of the message originator. Assume the false message is m and the corresponding public pseudo-identity is $PPID_{\mathcal{V}_{?,?}}$. To recover the real identity corresponding to $PPID_{\mathcal{V}_{?,?}}$, TA extracts the timestamp τ from m. According to τ, TA may know the valid period of the internal pseudo-identity of the message sender. To find who is the real sender, TA tests $H_2^{\lambda_i}(H_2^{\Lambda}(ID_{\mathcal{V}_i}||VP_i), \tau) \stackrel{?}{=} PPID_{\mathcal{V}_{?,?}}$, where $(ID_{\mathcal{V}_i}, VP_i, \lambda_i)$ is on the member list ML. If the equation holds, TA outputs $ID_{\mathcal{V}_i}$.

5 Evaluation

5.1 Security and Privacy

It is easy to see that our protocol satisfies the *global security* requirement of Section 2.2, by noting that: i) the employed IDAOTS scheme is shown to be secure; ii) the real identity of a vehicle can be traced as in the Trace stage. Below we analyze whether our protocol satisfies the *individual privacy* requirement. In our protocol, the only information that can be used by an eavesdropper to trace a vehicle is the public pseudo-identities. However, it is hard for anyone (except TA and the vehicle itself) to know the vehicle's internal pseudo-identity IPID: computing IPID implies finding the inverse of PPID (*i.e.*, the keyed hash output), which is impossible since the keyed hash is one-way. Furthermore, since the keyed hash outputs are computationally indistinguishable, that is, it is hard

for an attacker to determine whether two different public pseudo-identities (*i.e.*, two outputs of the keyed hash) are computed from the inputs of the same internal pseudo-identity, different time stamps and a secret key or the inputs of different internal pseudo-identities, different time stamps and a secret key. This implies that our protocol is unlinkable in the sense that an attacker cannot know whether two pseudonyms are linked with the same vehicle or not. Therefore, the *privacy* goal is achieved.

It is also worth noticing that our protocol is resistant to side-channel attacks. In our protocol, there are two kinds of secrets stored in the tamper-proof device. The first one is $(IPID_{\mathcal{V}_i}, \lambda_i)$ which is related to a vehicle $\mathcal{V}_i$'s privacy. If $\mathcal{V}_i$ does not renew $(IPID_{\mathcal{V}_i}, \lambda_i)$, it may leave the opportunity to an attacker to recover this secret, so that the attacker can trace $\mathcal{V}_i$. However, one may notice that, in practice, the attacker can only launch a side-channel attack occasionally. Therefore, in most cases, before the attacker collects enough side-channel information to recover the current secret $(IPID_{\mathcal{V}_i}, \lambda_i)$, the vehicle has already updated it, noting that APPA suggests a vehicle to renew $(IPID_{\mathcal{V}_i}, \lambda_i)$ periodically. Furthermore, in the worst case, even if the attacker recovers $(IPID_{\mathcal{V}_i}, \lambda_i)$, the attacker can only track the vehicle for a short period of time, *i.e.*, before $(IPID_{\mathcal{V}_i}, \lambda_i)$ is renewed. The second kind of secrets are the secret values $(\alpha_{i,j}, \beta_{i,j})$. In our protocol, these secret values are used once and then a random value is chosen to blind them (the *multiplicative secret sharing* technique). This technique is introduced by Kiltz and Pietrzak [6], and, can be used to convert a scheme to be leakage resilient one. With this technique, the attacker cannot collect enough information about these secret values. Because the precondition of side-channel attacks is that the same secret value be involved in a large number of cryptographic operations so that enough side-channel information about the secret value can be collected to perform a statistical analysis.

5.2 Transmission and Storage Overhead

Table 1 compares the transmission and storage overhead incurred by the security and privacy mechanisms with up-to-date protocols in the literature. For fairness, we consider protocols which, as our APPA protocol, do not require pre-storing a large number of anonymous certificates/pseudo identities.

According to [2], the length of a point in group $\mathbb{G}_1$ is 171 bits (about 21 bytes). In addition, the length of an identity is 20 bytes. For our protocol, it is easy to see that the length of a signature and an IPID is about $21 + 20$ bytes. However, the signatures in our protocol can be aggregated into a single point in $\mathbb{G}_1$. Hence, the length of the aggregate signature will not increase with the number of messages received and the total overhead is $21 + 20n$ bytes. This length is about 1/2 of that in [25]. Furthermore, the transmission and storage overhead of our protocol is much lower than that in [12] and [27].

5.3 Impact of Signature Verification on Response Time

Table 2 compares the computational overhead of signature verification with the same protocols considered in Table 1. It is sufficient to only consider the most

Table 1. Comparison of protocols in terms of Transmission and Storage Overhead

	Send/Store a single message	Store n messages
Protocol 1 [12]‡	192 bytes	$192n$ bytes
Protocol 2 [25]†	$21 + 41$ bytes	$21 + 41n$ bytes
Protocol 3 [27]‡	368 bytes	$368n$ bytes
Our Protocol†	$21 + 20$ bytes	$21 + 20n$ bytes

†: Identity-based signature based
‡: Group signature based

Table 2. Comparison of protocols in terms of Computational Overhead

	Verify a single signature	Verify n signatures
Protocol 1 [12]	$5\tau_{bp} + 12\tau_{pe}$	$5n\tau_{bp} + 12n\tau_{pe}$
Protocol 2 [25]	$3\tau_{bp} + \tau_{pe} + \tau_{mh}$	$3\tau_{bp} + n\tau_{pe} + n\tau_{mh}$
Protocol 3 [27]	$2\tau_{bp} + 14\tau_{pe}$	$2\tau_{bp} + \frac{14n}{4.8}\tau_{pe}$
Our Protocol	$2\tau_{bp} + \tau_{pe} + 2\tau_{mh}$	$2\tau_{bp} + n\tau_{pe} + 2n\tau_{mh}$

costly operations, *i.e.*, pairing, point exponentiation and map-to-point hash (*e.g.*, $H_0(\cdot)$) operations.

The processing time for one bilinear pairing operation is about $\tau_{bp} = 1.87$ ms, the time for one point exponentiation operation is about $\tau_{pe} = 0.49$ ms and the computational cost of one map-to-point hash is about 0.22 ms [9,10]. In Figure 2, one may notice that the signature verification cost in our protocol is much more efficient than that in [12] and slightly more efficient than that in [27]. Further, when the number of signatures to be verified is small, our protocol is more efficient than that in [25]. When the number of signatures to be verified grows, our protocol has comparable efficiency with that in [25]. However, the signature length in our protocol is about 1/2 of that in [25]. Furthermore, unlike the protocol in [25], our protocol is resistant to side-channel attacks.

5.4 Tracing Efficiency

As shown in Section 4, to find the real identity of a message sender, TA tests

$$H_2^{\lambda_i}(H_2^{\Lambda}(ID_{\mathcal{V}_i}||VP_i), \tau) \stackrel{?}{=} PPID_{\mathcal{V}_{?,?}},$$

where $(ID_{\mathcal{V}_i}, VP_i, \lambda_i)$ is on the list ML. If the equation holds, TA outputs $ID_{\mathcal{V}_i}$. The cost of recovering the real identity of a sender may seem very high, because there may be millions of tuples in ML. However, we notice that TA does not need to test all the tuples on ML. Instead, TA only needs to test the tuples that contain VPs that match τ. Therefore, for a VANET with 1 million vehicles, TA only needs to test 0.5 million times in average. We use the popular keyed SHA-1 to estimate the tracing efficiency of our protocol. It only takes 0.001 ms on

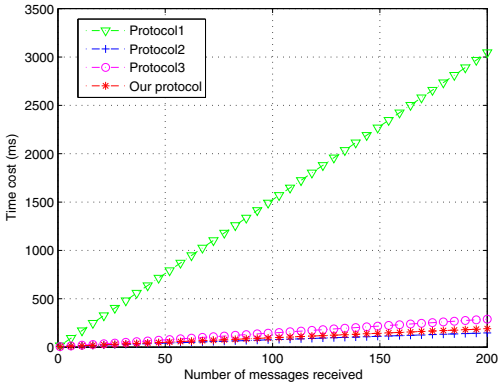

Fig. 2. Verification cost

average for TA to perform a SHA-1 hash operation [19]. Therefore, for a VANET with 1 million vehicles, our protocol only needs about one second to find the real identity of the message originator.

6 Conclusion

In this paper, we proposed the APPA protocol to secure VANETs. The protocol allows aggregate privacy-preserving authenticated vehicular communications. The protocol guarantees trustworthiness of vehicle-generated messages and privacy of vehicles. With APPA, vehicles can react to received messages within a very short delay. The digital signatures for the authentication purpose are securely and substantially compressed. The APPA protocol does not heavily rely on roadside units, which implies that the protocol can work even if the VANET infrastructure is incomplete. These features seem desirable and allow our protocol to be deployed in various secure VANET scenarios.

Acknowledgments and disclaimer. This work is partly supported by the Nature Science Foundation of China through projects 61021004, 11061130539, 60970114, 60970115, 60970116 and 61003214, by the European commission under FP7 project "DwB", by the Spanish Government under projects TSI2007-65406-C03-01 "E-AEGIS", TIN2009-11689 "RIPUP", "eVerification" TSI-020100-2009-720, SeCloud TSI-020302-2010-153 and CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", by the Fundamental Research Funds for the Central Universities of China through projects 3103004, 78210044, by the Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No. 2010JK727), by the Shanghai Natural Science Foundation under Grant No. 11ZR1411200 and by the Beijing Municipal Natural Science Foundation to Project 4112052. The forth author is partially supported as an ICREA-Acadèmia

researcher by the Catalan Government. The views of the author with the UNESCO Chair in Data Privacy do not necessarily reflect the position of UNESCO nor commit that organization.

References

1. Dedicated Short Range Communications (DRSC) home, `http://www.leearmstrong.com/Dsrc/DSRCHomeset.htm`
2. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
3. Calandriello, G., Papadimitratos, P., Hubaux, J.-P., Lioy, A.: Efficient and robust pseudonymous authentication in vanet. In: ACM VANET 2007, pp. 19–28. ACM Press, New York (2007)
4. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
5. Daza, V., Domingo-Ferrer, J., Sebé, F., Viejo, A.: Trustworthy privacy-preserving car-generated announcements in vehicular ad hoc networks. IEEE Transactions on Vehicular Technology 58(4), 1876–1886 (2009)
6. Kiltz, E., Pietrzak, K.: Leakage resilient elGamal encryption. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 595–612. Springer, Heidelberg (2010)
7. Frey, G., Rück, H.-G.: A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. Mathematics of Computation 62(206), 865–874 (1994)
8. Gentry, C., Ramzan, Z.: Identity-Based Aggregate Signatures. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 257–273. Springer, Heidelberg (2006)
9. Icart, T.: How to hash into elliptic curves. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 303–316. Springer, Heidelberg (2009)
10. Jiang, Y., Shi, M., Shen, X., Lin, C.: BAT: A robust signature scheme for vehicular networks using binary authentication trees. IEEE Transactions on Wireless Communications 8(4), 1974–1983 (2009)
11. Kocher, P.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)
12. Lin, X., Sun, X., Ho, P., Shen, X.: GSIS: A secure and privacy preserving protocol for vehicular communications. IEEE Transactions on Vehicular Technology 56(6), 3442–3456 (2007)
13. Lu, R., Lin, X., Zhu, H., Ho, P., Shen, X.: ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications. In: IEEE INFOCOM 2008, pp. 1229–1237. IEEE Computer Society Press, Los Alamitos (2008)
14. Menezes, A., Okamoto, T., Vanstone, S.A.: Reducing elliptic curves logarithms to logarithms in a finite field. IEEE Transactions on Information Theory 39(5), 1639–1646 (1993)
15. Papadimitratos, P., Gligor, V., Hubaux, J.: Securing vehicular communications - Assumptions, requirements, and principles. In: ESCAR 2006 (2006)
16. Picconi, F., Ravi, N., Gruteser, M., Iftode, L.: Probabilistic validation of aggregated data in vehicular ad hoc networks. In: ACM VANET 2006, pp. 76–85. ACM Press, New York (2006)

17. Raya, M., Hubaux, J.: The security of vehicular ad hoc networks. In: ACM SASN 2005, pp. 11–21. ACM Press, New York (2005)
18. Raya, M., Hubaux, J.: Securing vehicular ad hoc networks. Journal of Computer Security 15(1), 39–68 (2007)
19. Satizábal, C., Martínez-Peláez, R., Forné, J., Rico-Novella, F.: Reducing the computational cost of certification path validation in mobile payment. In: López, J., Samarati, P., Ferrer, J.L. (eds.) EuroPKI 2007. LNCS, vol. 4582, pp. 280–296. Springer, Heidelberg (2007)
20. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
21. Standaert, F., Malkin, T., Yung, M.: A unified framework for the analysis of side-channel key recovery attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 443–461. Springer, Heidelberg (2009)
22. Wasef, A., Shen, X.: ASIC: Aggregate signatures and certificates verification scheme for vehicular networks, `http://www.engine.lib.uwaterloo.ca`
23. Wu, Q., Domingo-Ferrer, J., Gonzalez-Nicolas, U.: Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications. IEEE Transactions on Vehicular Technology 59(2), 559–573 (2010)
24. Wu, Q., Mu, Y., Susilo, W., Qin, B., Domingo-Ferrer, J.: Asymmetric group key agreement. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 153–170. Springer, Heidelberg (2009)
25. Zhang, C., Lu, R., Lin, X., Ho, P., Shen, X.: An efficient identity-based batch verification scheme for vehicular sensor networks. In: IEEE INFOCOM 2008, pp. 246–250. IEEE Computer Society Press, Los Alamitos (2008)
26. Zhang, L., Wu, Q., Qin, B., Domingo-Ferrer, J.: Identity-based authenticated asymmetric group key agreement protocol. In: Thai, M.T., Sahni, S. (eds.) COCOON 2010. LNCS, vol. 6196, pp. 510–519. Springer, Heidelberg (2010)
27. Zhang, L., Wu, Q., Solanas, A., Domingo-Ferrer, J.: A scalable robust authentication protocol for secure vehicular communications. IEEE Transactions on Vehicular Technology 59(4), 1606–1617 (2010)
28. Zhu, H., Lin, X., Lu, R., Ho, P., Shen, X.: AEMA: An aggregated emergency message authentication scheme for enhancing the security of vehicular ad hoc networks. In: IEEE ICC 2008, pp. 1436–1440. IEEE Computer Society Press, Los Alamitos (2008)