2012 Fourth International Conference on Intelligent Networking and Collaborative Systems

An Improved Binary Authentication Tree Algorithm for Vehicular Networks

Huaqun Wang[1,2]
1. School of Information Engineering Dalian Ocean University, Dalian, China
2. Dept. of Computer Engineering and Maths, UNESCO Chair in Data Privacy, Universitat Rovira i Virgili
E-mail: wanghuaqun8@gmail.com

Bo Qin[3,4]
3. Dept. of Computer Engineering and Maths, UNESCO Chair in Data Privacy, Universitat Rovira i Virgili
4. Xi'an University of Technology, Xi'an, China
Email: bo.qin@urv.cat

Josep Domingo-Ferrer
Dept. of Computer Engineering and Maths, UNESCO Chair in Data Privacy, Universitat Rovira i Virgili
Email: josep.domingo@urv.cat

***Abstract*—Vehicular networks are gaining popularity because vehicular communications are able to help minimize accidents, improve traffic conditions, etc. To avoid malicious attacks and potential abuse, employing digital signatures is widely recognized as the most efficient approach for vehicular networks. However, when the number of signatures received by a roadside unit (RSU) becomes large, a scalability problem emerges immediately: it can be difficult for the RSU to sequentially verify every received signature within 100-300ms as required by the current Dedicated Short Range Communications (DSRC) broadcast protocol. Jiang *et al.* proposed a robust and efficient signature scheme for vehicular-to-infrastructure communications, called binary authentication tree. In this paper, we show that their binary authentication algorithm is insecure to at least two attacks. The first attack shows that the original binary authentication algorithm is universally forgeable under chosen-message attacks, *i.e.* the attacker can forge other vehicles' authentication on any message under chosen-message attack. The second attack shows that the original binary authentication algorithm is universally forgeable, *i.e.*, the attacker can forge other vehicles' authentication on any message at will. Although Jiang et *al.*'s scheme is insecure, it can be repaired. Using the binary authentication tree model, we repair their scheme in order to make it provably secure and efficient.**

***Keywords*-vehicular networks; authentication; cryptanalysis; bilinear pairings**

I. Introduction

Vehicular Ad-hoc Networks (VANETs) have attracted extensive attention in recent years for their promise of revolutionizing the human driving modes and transportation systems. Smart vehicles have embedded computers, Global Positioning System (GPS), short-range wireless network interfaces, and potentially wireless access to the internet. With these equipments, vehicles can communicate with each other (V2V: Vehicle-to-Vehicle) or with roadside units (RSU) which are connected to the internet (V2I: Vehicle-to-Infrastructure). Vehicular communication over the wireless medium employs the Dedicated Short Range Communications protocol (DSRC, [1]). According to the DSRC protocol, each vehicle in a VANETs broadcasts a traffic safety message every 100-300ms, which keeps other vehicles updated about the sending vehicle's driving-related information, such as location, speed, turning intention, and driving status.

The security and efficiency in VANETs face many challenges due to the open broadcasting of wireless communications and the high-speed mobility of the vehicles. It is obvious that any malicious behavior of the user, such as injecting beacons with false information, or modifying and replaying the disseminated messages, could be fatal to the other users. Furthermore, privacy must be achieved in the sense that the vehicle related privacy information should be protected so that an attacker can be prevented from collecting vehicle messages, tracking locations, and inferring sensitive data. Meanwhile, the authorities should be able to trace the identities of message senders in case of a traffic dispute. To satisfy the security and efficiency requirements, it is a prerequisite to develop a suite of elaborate protocols to achieve security, privacy, and efficient message authentication before vehicular networks can be practically deployed.

Strong authentication is desirable to validate each message sent by the On Board Units (OBUs). In IEEE Standard 1609.2 [2], strong authentication was achieved by using a signature scheme. However, classic signature schemes that sequentially verify the messages may fail to satisfy the real-time requirement in vehicular communications. Robustness and efficiency are the two basic requirements for strong authentication in VANETs [3–5]. Message authentication, integrity and non-repudiation, as well as privacy preservation are identified as primary requirements. Another important issue is verification performance. According to the DSRC protocol, an RSU may communicate with hundreds of OBUs and each OBU will periodically transmit a safety or traffic message (beacon) to the nearest RSU via a common DSRC channel. The beaconing rate ρ typically ranges from 3 to 10 beacons per second, with $\rho = 10$ currently considered as necessary for safety applications. Therefore, even in a normal traffic scenario, it is a very rigorous requirement for any RSU using classic signature schemes to verify a mass of messages in real-time. The delay caused by verifying a

978-0-7695-4808-1/12 $26.00 © 2012 IEEE
DOI 10.1109/iNCoS.2012.27

IEEE computer society

bulk of signatures may radically impede the transmission throughput and impair the system scalability. A possible promising approach to improve the verification efficiency is to employ batch verification [6–11], which permits verifying a large number of signatures simultaneously instead of sequentially. This decreases the number of time-consuming operations, especially when authenticating a large number of signatures.

To address the aforesaid security and performance issues, Jiang *et al.* proposed a robust and efficient signature scheme for V2I communications, called binary authentication tree (BAT, [12]). They claimed that the scheme features the following notable properties. (1) Robustness: The BAT scheme can resist attacks with bogus messages, since each RSU can quickly distinguish the bogus messages from all the authentic ones; hence, the scheme can efficiently tolerate, to a large extent, message flooding attacks. (2) Efficiency: The BAT scheme efficiently eliminates the performance bottleneck due to significantly reduced computational overhead. Therefore, the authors of BAT claimed that their scheme can meet the security and efficiency requirements for V2I communications with low message transmission overhead, identity privacy, and traceability. They claimed that the proposed BAT scheme is the first one to include evaluated theoretical boundaries of verification complexity for the batch verification of identity-based signatures under adverse attacks, which can be used to guide the balance between security and performance. However, we found that BAT is vulnerable by cryptanalysis. Two attack methods are given. In order to design a robust and efficient signature scheme for V2I communications, we improved Jiang *et al.*'s BAT scheme and gave a formal security proof.

The remainder of the paper is organized as follows. In Section II, preliminaries related to the proposed research are given, including the application model, the pairing concept and the batch verification concept. In Section III, the review of Jiang *et al.*'s scheme was given. In Section IV, two attack methods are presented. In Section V, we gave our improved BAT scheme and the corresponding security proof. The conclusions are given in Section VI.

II. Preliminaries

A. Application Scenario Model

In this section, we first give the description of the application scenario model, followed by the introduction of identity-based cryptography and the bilinear pairings, which are the foundation of the proposed BAT scheme. At last, we give the concept of batch verification. The notations throughout this paper are listed in Table I.

As shown in Figure 1, we consider the representative Vehicle-to- Infrastructure communications architecture, which includes:

Table I
Notations and descriptions

Notations	Descriptions
s	The private master key of the TA
P_{pub}	The public key of the TA
ID_i	The real identity of the Vehicle V_i
PID_i	The pseudo identity of the Vehicle V_i
SK_i	A private key of the Vehicle V_i
$\|\|$	Message concatenation operation
$h(\cdot)$	A one-way hash function
$H(\cdot)$	A MapToPoint hash function
$E_K(\cdot)$	Symmetric encryption with key K
$D_K(\cdot)$	Symmetric decryption with key K
V_i	The i-th vehicle
M_i	A message sent by vehicle V_i
α_i	A signature sent by vehicle V_i
$\bigoplus$	Bitwise XOR operation

1) RSU: An RSU serves as a gateway connecting the vehicles within its transmission range to the Internet.
2) Vehicles: A vehicle periodically exchanges messages with the RSU within its range. Each vehicle is equipped with sensing and processing units, called On-Board Units (OBUs).
3) TA (Trusted Authority): The TA server, as the key distribution center, is responsible for generating and assigning related parameters for the vehicles and RSUs, and identifying a malicious identity for any dispute events.
4) SP (Service Provider): The SP or Application Server is responsible for collecting the traffic-related information.
5) VRS (Vehicle Registration Site): The VRS is responsible for registrating the vehicles.

An RSU may communicate with hundreds of OBUs at the same time within its communication range, using the DSRC broadcast protocol, the designated protocol for vehicular networks. Each vehicle uses its private keys to sign messages and then sends them to its neighboring RSU, while each RSU is in charge of authenticating the received messages.

B. Identity-based Cryptography and Bilinear Pairings

Identity-based cryptography (IBC) is a type of public-key cryptography in which the public key of a user is his or her unique identity information. As an important IBC scheme, the pairing-based IBC scheme can offer lower transmission cost compared with the traditional RSA-based schemes, due to the smaller signature overhead. We briefly introduce the bilinear pairing as follows.

Let $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively, be a cyclic additive group generated by P and a cyclic multiplicative group with the same prime order q, *i.e.*, $|\mathcal{G}_1| = |\mathcal{G}_2| = q$. Let $e : \mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_2$ be a bilinear map, which satisfies the following properties:

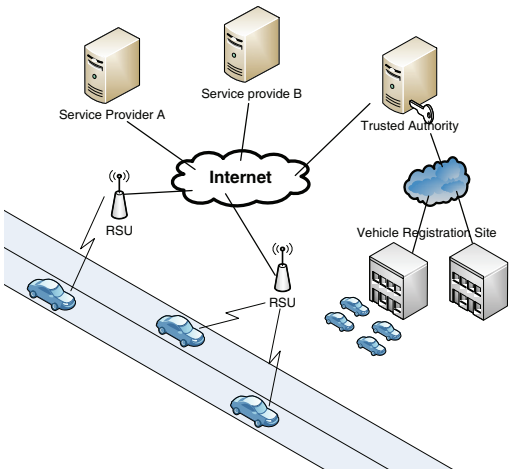

Figure 1. Application scenario model

1) Bilinearity: $\forall Q, R, S \in \mathcal{G}_1$ and $a, b \in Z$, $e(R, Q + S) = e(Q + S, R) = e(Q, R)e(S, R)$.
2) Non-degeneracy: $\exists Q, R \in \mathcal{G}_1$ such that $e(Q, R) \neq 1_{\mathcal{G}_2}$.
3) Computability: $\forall Q, R \in \mathcal{G}_1$, there is an efficient algorithm to calculate $e(Q, R)$.

Such a bilinear map e can be constructed by the modified Weil [13] or Tate pairings [14] on elliptic curves. A group with such a map e is called a bilinear group, on which the Computational Diffie-Hellman (CDH) problem is assumed hard while the Decisional Diffie-Hellman (DDH) problem is easy to solve [15]. For instance, given unknown $a, b, c \in \mathcal{Z}_q$ and $P, aP, bP, cP \in \mathcal{G}_1$, it is recognized that there exists an efficient algorithm to determine whether $ab = c \bmod q$ by verifying $e(aP, bP) = e(P, cP)$ in polynomial time (DDH problem), while there exist no efficient algorithms to compute $abP \in \mathcal{G}_1$ with non-negligible probability within polynomial time (CDH problem).

C. Batch Verification

Batch verification can reduce large computational cost when multiple signatures are verified together. When a collection of signatures pass the batch verifications, the verifier accepts all the signatures as valid. Otherwise, the collection is rejected. The idea of batch cryptography was introduced by Fiat [8, 16]. In 1994, Naccache *et al.* [17] proposed the first DSA batch verification scheme. The authors introduced batch verification to verify several DSA signatures at once, which is much more efficient than sequential verification of individual DSA signatures. The pre-proceedings version of [17] paper included an additional interactive batch verifier. Lim and Lee [18] showed that this version is not secure since any attacker can easily forge multiple individual signatures to make a false batch verification valid. Bellare *et al.* [19] proposed a small exponents test technique to overcome this security problem. Current batch verifications are efficient at verifying many signatures done by one signer. Boyd *et al.* have attacked and repaired some batch verification schemes [20]. H. C. Jung *et al.* developed two batch verification algorithms using sparse exponents of small weights [21]. Ferrara *et al.* introduce new batch verifiers for a wide variety of regular, identity-based, group, ring and aggregate signature schemes [22]. We make use of the pairing-based batch verifiers concept from [22].

$PSetup$ is an algorithm that, on input the security parameter 1^τ, outputs the parameters $(q, P, \mathcal{G}_1, \mathcal{G}_2, e)$, where $\mathcal{G}_1, \mathcal{G}_2$ are of prime order q. Pairing-based verification equations are represented by a generic pairing based claim X corresponding to a Boolean relation of the following form:

$$\prod_{i=1}^{k} e(f_i, h_i)^{c_i} \stackrel{?}{=} A$$

for $k \in poly(\tau)$ and $f_i, h_i \in \mathcal{G}_1$, and $c_i \in \mathcal{Z}_q^*$, for each $i = 1, \cdots, k$. A pairing-based verifier Verify for a generic pairing-based claim is a probabilistic poly(τ)-time algorithm which on input the representation $\{A, f_1, \cdots, f_k, h_1, \cdots, h_k, c_1, \cdots, c_k\}$ of a claim X, outputs *accept* if X holds and *reject* otherwise. We define a batch verifier for pairing-based claims.

Definition 1 (Pairing-based Batch Verifier): Let $PSetup(1^\tau) \rightarrow (q, P, \mathcal{G}_1, \mathcal{G}_2, e)$. For each $j \in [1, \eta]$, where $\eta \in poly(\tau)$, let $X(j)$ be a generic pairing-based claim and let Verify be a pairing-based verifier. We define a pairing-based batch verifier for Verify as a probabilistic $poly(\tau)$-time algorithm which returns:

1) *accept* if $X(j)$ holds for all $j \in [1, \eta]$;
2) *reject* if $X(j)$ does not hold for some $j \in [1, \eta]$ except with negligible probability.

III. Review of Jiang et al.'s Scheme

Based on a new data structure called BAT, Jiang *et al.* propose a robust and efficient signature scheme for vehicular communications. Jiang *et al.*'s binary authentication tree algorithm consists of the basic signature scheme and the binary authentication tree algorithm. The binary authentication tree algorithm is based on the basic signature scheme. We briefly describe them as follows.

A. Basic Signature Scheme

It mainly consists of four algorithms: setup, extract, sign and verify. There are three parties:the TA(i.e., trusted authority), the vehicle (signer), the RSU (verifier).

Setup: TA generates the following parameters $(\mathcal{G}_1, \mathcal{G}_2, g, e, H, h)$. $\mathcal{G}_1$ and $\mathcal{G}_2$ are a cyclic additive group and a cyclic multiplicative group with the same order q. g is a generator of the cyclic multiplicative group $\mathcal{Z}_q^*$. $e : \mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_2$ is a bilinear map. H is a MapToPoint hash function and h is a one-way function. TA randomly picks

$s \in \mathcal{Z}_q^*$ as its secret master key and computes $P_{pub} = sP$ as its public key.

Extract: When a legitimate signer V_i registers with the TA, it submits her unique identity $ID_i \in \{0,1\}^z$ to the TA, where $z \in \mathcal{Z}_q^*$. TA picks $\{w, v_{i,1}, v_{i,2}, \cdots, v_{i,z}\} \in \mathcal{Z}_q^*$ and computes

$$PK_i^* = \{g^{v_{i,1}}, \cdots, g^{v_{i,z}}\},$$
$$K_{TV_{i,k}} = g^{v_{i,k}w},$$
$$PID_{i,k} = E_{K_{TV_{i,k}}}(g^{v_{i,k}} \bigoplus ID_i),$$
$$PID_i^* = \{PID_{i,k} | k = 1, 2, \cdots, z\},$$

where $K_{TV_{i,k}}$ is the secret key. TA uses the pseudo identity in PID_i^* to derive the corresponding signature key $SK_i^* = \{SK_{i,k} | k = 1, 2, \cdots, z\}$ as $SK_{i,k} = sH(PID_{i,k})$.

Finally, TA delivers the corresponding security parameters $\{\mathcal{G}_1, \mathcal{G}_2, q, P, P_{pub}\}$ and $\{PID_i^*, SK_i^*, PK_i^*\}$ to V_i.

Sign: To sign a message $m \in \{0,1\}^*$, V_i picks a random pseudo identity $PID_{i,k} \in PID_i^*$ and picks a random $r_i \in \mathcal{Z}_q^*$ and computes:

$$E_i = r_i P,$$
$$F_i = r_i P_{pub} + h(M_i, E_i) SK_{i,k}$$

where $M_i = \{PID_i || g^{v_{i,k}} || H(PID_i) || m_i\}$, $g^{v_{i,k}}$ is associated with $SK_{i,k}$. Let $\alpha_i = (E_i, F_i)$. V_i sends (M_i, α_i) to the verifier.

Verify: Upon receiving the message-signature pair (M_i, α_i), the RSU verifies the validity of the signature (M_i, α_i) by checking if

$$e(F_i, P) = e(E_i + h(M_i, E_i) H(PID_{i,k}), P_{pub})$$

B. Binary Authentication Algorithm

Due to the time-consuming pairing operation, verifying the messages sequentially would result in a performance bottleneck for each RSU and impair the system scalability. Jiang *et al.* introduce an alternative verification algorithm for increased efficiency and robustness, based on the following novel BAT data structure. Without loss of generality, assuming that there are $n = 2^h$ vehicles $\{V_1, \cdots, V_n\}$ and the corresponding signatures $\{\alpha_1, \cdots, \alpha_n\}$, a binary authentication tree can be constructed as follows:

1) Each leaf node (h, v) in BAT is associated with the signature $\alpha_{i+1} = (E_{i+1}, F_{i+1})$ of vehicle V_{v+1}, where (h, v) is the v-th node at the h-level;
2) Each inner node (l, v), for $l \leq h-1$, is associated with an aggregate signature $\alpha_{(l,v)} = \{\alpha_{k_1}, \alpha_{k_1+1}, \cdots, \alpha_{k_2}\}$ of all the signatures in the leaf nodes of the subtree rooted at (l, v), where $k_1 = 2^{h-l}v$ and $k_2 = 2^{h-l}(v+1)-1$. The root node is an aggregate signature $\alpha_{(0,0)} = \{\alpha_1, \cdots, \alpha_n\}$ associated to all signatures at the leaf nodes.

For the 2^{h-l} message-signature pairs, $\{(M_{k_1}, \alpha_{k_1}), (M_{k_1+1}, \alpha_{k_1+1}), \cdots, (M_{k_2}, \alpha_{k_2})\}$, where $\alpha_i = (E_i, F_i)$ for $i = k_1, k_1+1, \cdots, k_2$ is the basic signature scheme. All the signatures $\{\alpha_{k_1}, \alpha_{k_1+1}, \cdots, \alpha_{k_2}\}$ can be verified by checking if the following equation holds:

$$e(\sum_{i=k_1}^{k_2} F_i, P) = e(\sum_{i=k_1}^{k_2} (E_i + h(M_i, E_i) H(PID_i)), P_{pub})$$

where $k_1 = 2^{h-l}v$ and $k_2 = 2^{h-l}(v+1) - 1$.

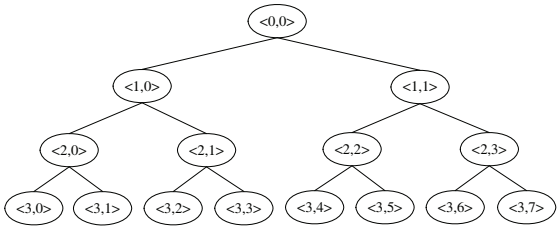

Figure 2. Binary authentication tree

For instance, as shown in Figure 2, leaf node $< 3, 0 >$ is associated with the signature α_1 of vehicle V_1, while the inner node $< 2, 3 >$ is associated with the aggregate signature $\alpha_{<2,2>} = \{\alpha_5, \alpha_6\}$ for vehicles V_5 and V_6. The root node $< 0, 0 >$ is associated with the whole signatures $\alpha_{<0,0>} = \{\alpha_1, \cdots, \alpha_8\}$.

Thus, the group-based authentication can noticeably reduce the computational cost, especially when verifying a large number of aggregate signatures. The computation cost to verify k signatures mainly consists of k multiplications, k one-way hashes, and 2 pairing operations.

IV. Cryptanalysis of Jiang et al.'s Binary Authentication Algorithm

We now show that Jiang *et al.*'s binary authentication algorithm cannot resist the forgery attack. Of course, this algorithm was not proven secure in the original paper. We think that resisting forgery is the most fundamental security requirement for a batch authentication algorithm. Loosely speaking, secure signature schemes should prevent an adversary from generating valid signature on "unauthentic" documents (*i.e.*, documents that were not approved by the legitimate signer). According to the concept of unforgeable signature [23], we give the concept of unforgeable batch verification signature.

Definition 2: For a probabilistic oracle machine, M, we denote $Q_M^O(x)$ the set of queries made by M on input x and access to oracle O. As usual, $M^O(x)$ denotes the output of the corresponding computation. We stress that $Q_M^O(x)$ and $M^O(x)$ are dependent random variables that represent two aspects of the same probabilistic computation. A batch verification signature scheme is *unforgeable* if for every probabilistic polynomial-time oracle machine M, every positive polynomial p, and all sufficient large n, it holds that

$Pr[V_{v_1,\cdots,v_k}((\alpha_1,\beta_1),(\alpha_2,\beta_2),\cdots,(\alpha_k,\beta_k))=1 \ \& \ \exists i \in \{1,\cdots,k\}, V_{v_i}((\alpha_i,\beta_i)) \neq 1$, where $(s_i,v_i) \leftarrow G(1^\tau)$, $(\alpha_i,\beta_i) \leftarrow M^{S_{s_i}}, i \in \{1,\cdots,k\}] < \frac{1}{p(x)}$ where the probability is taken over the coin tosses of algorithms G,S,V, as well as over the coin tosses of machine M.

A batch verification signature is secure (or unforgeable) if every feasible chosen message attack succeeds with at most negligible probability. According to the attack model, two attack methods are given as follows. Finally, we analyze the feasibility of the two attack methods.

A. Attack 1

We will show that universal forgery is possible in the batch verification signature. Assume the attacked inner node is (l,v). Consider the adversary A queries for the private key of some vehicle $ID_j, k_1 \leq j \leq k_2$. Let the forged identity is $ID_k, k \in \{k_1,\cdots,j-1,j+1,\cdots,k_2\}$. Assume that A in the training phase queries for the signature of vehicle ID_k on message $\hat{M}_k$, $k \in \{k_1,\cdots,j-1,j+1,\cdots,k_2\}$. The signature is of the form $\hat{E}_k = \hat{r}_k P$, $\hat{F}_k = \hat{r}_k P_{pub} + h(\hat{M}_k,\hat{E}_k)SK_k$, for $k \in \{k_1,\cdots,j-1,j+1,\cdots,k_2\}$ and a random unknown $\hat{r}_k \in \mathcal{Z}_q^*$.

The adversary does the following steps to forge a batch verification signature on messages $\{M_{k_1},\cdots,M_{j-1},M_{j+1},\cdots,M_{k_2}\}$.

1) Pick random $r_k \in \mathcal{Z}_q^*$ and set $E_k = r_k P$;
2) Set $h_k = H(M_k,E_k)$, $\hat{h}_k = H(\hat{M}_k,\hat{E}_k)$;
3) Set $E_k^* = h_k \hat{h}_k^{-1} \hat{E}_k$;
4) Set $F_k^* = h_k \hat{h}_k^{-1} \hat{F}_k$;
5) Set $F_k = F_k^* + r_k P_{pub}$

The signature on M_k by vehicle ID_k is $\sigma_k = (E_k,F_k)$, $k \in \{k_1,\cdots,j-1,j+1,\cdots,k_2\}$.

Making use of the private key SK_j, the adversary A signs a random message M_j as follows.

1) Pick random $r_j \in \mathcal{Z}_q^*$ and set $E_j = r_j P + \sum_{k \neq j} E_k^*$;
2) Set $F_j = r_j P_{pub} + h(M_j,E_j)SK_j$.

Then the batch signature $\{(E_{k_1},F_{k_1}),(E_{k_1+1},F_{k_1+1}),\cdots,(E_{k_2},F_{k_2})\}$ can pass the batch verification but almost certainly none of the signatures is correct.

The correctness of the forged batch verification signature can be easily proven as follows:

$$\begin{aligned}
& e(\sum_{i=k_1}^{k_2} F_i, P) \\
= \ & e(\sum_{i \neq j}\{F_i^* + r_i P_{pub}\} + r_j P_{pub} + h(M_j,E_j)SK_j, P) \\
= \ & e(\sum_{i \neq j}\{h_i \hat{h}_i^{-1}(s\hat{E}_i + \hat{h}_i SK_i) + r_i P_{pub}\} + r_j P_{pub} \\
& + h(M_j,E_j)SK_j, P) \\
= \ & e(\sum_{i \neq j}\{h_i \hat{h}_i^{-1}\hat{E}_i + h_i H(PID_i) + r_i P\} + r_j P + \\
& h(M_j,E_j)H(PID_j), P_{pub}) \\
= \ & e((\sum_{i \neq j} E_i^* + r_j P) + \sum_{i \neq j} E_i + \sum_{i=k_1}^{k_2} H(M_i,E_i)H(PID_i), \\
& P_{pub}) \\
= \ & e(\sum_{i=k_1}^{k_2}(E_i + h(M_i,E_i)H(PID_i)), P_{pub})
\end{aligned}$$

Thus, this forged batch verification signature can pass the verification. The adversary can forge batch verification signatures on any message. This flaw occurs because any signer is able to cancel out the components of any other user in the batch verification. The basic signature scheme is secure but it is not secure when batch verified. Thus, we have proposed a universal forgery on Jiang *et al.*'s binary authentication algorithm.

B. Attack 2

This attack shows that any vehicle $PID_i \in \{PID_{k_1}, PID_{k_1+1},\cdots,PID_{k_2}\}$ can impersonate other vehicles to generate a batch verification signature $\alpha_{(l,v)} = \{\alpha_{k_1}, \alpha_{k_1+1},\cdots, \alpha_{k_2}\}$. The signature can pass the batch verification. We next detail how the forgery works. Suppose that the attacker is PID_{k_1}, without loss of generality.

For $i = \{k_1+1, k_1+2,\cdots,k_2\}$, PID_{k_1} randomly picks $M_i \in \{0,1\}^*, E_i, F_i \in \mathcal{G}_1$, $r \in \mathcal{Z}_q^*$, $M_{k_1} = \{0,1\}^*$, and computes:

$$E = rP, \ \ E_{k_1} = E - \sum_{i=k_1+1}^{k_2}(E_i + h(M_i,E_i)H(PID_i))$$

$$F = rP_{pub} + h(M_{k_1},E_{k_1})SK_{k_1}, \ \ F_{k_1} = F - \sum_{i=k_1+1}^{k_2} F_i$$

Let $\alpha_i = (E_i,F_i), i = k_1, k_1+1,\cdots,k_2$, then $\alpha_{(l,v)} = \{\alpha_{k_1}, \alpha_{k_1+1},\cdots,\alpha_{k_2}\}$ are signatures on messages $\{M_{k_1}, M_{k_1+1},\cdots, M_{k_2}\}$.

The correctness of the forged batch verification signature can be easily proven as follows:

$$\begin{aligned}
& e(\sum_{i=k_1}^{k_2} F_i, P) = e(F,P) \\
= \ & e(rP_{pub} + h(M_{k_1},E_{k_1})SK_{k_1}, P) \\
= \ & e(E + h(M_{k_1},E_{k_1})H(PID_{k_1}), P_{pub})
\end{aligned}$$

$$
\begin{aligned}
&= e(E_{k_1} + \sum_{i=k_1+1}^{k_2} (E_i + h(M_i, E_i)H(PID_i)) \\
&+h(M_{k_1}, E_{k_1}) \times H(PID_{k_1}), P_{pub}) \\
= \quad & e(\sum_{i=k_1}^{k_2} (E_i + h(M_i, E_i)H(PID_i)), P_{pub})
\end{aligned}
$$

The above attack shows that it is easy for a vehicle to forge other vehicles' signatures on any message. The forged signatures can pass the verification of the binary authentication algorithm . Thus we have proposed another universal forgery on the Jiang *et al.*'s binary authentication algorithm.

C. Feasibility Analysis of the Two Attacks

According to the security model of the batch verification signature, our two attacks can succeed against Jiang *et al.*'s binary authentication algorithm. First, we analyze the efficiency of the two attacks. Second, we analyze the success probability of both attacks when the original approach was fixed by adding some computation overhead. We analyze the two attacks for attacking the inner node (l, v). Supposed that there are $n = 2^h$ vehicles, and $k_1 = 2^{h-l}v$, $k_2 = 2^{h-l}(v+1) - 1$, $l < h$.

The two attacks can succeed in the unforgeability model of the batch verification signature, *i.e.*, the attacker can get batch verification signature on a message that has not been queried to the Sign oracle. For the first attack, the attacker needs $4(k_2 - k_1) + 3$ scalar multiplication operations and $2(k_2 - k_1) + 1$ point addition operations in the group $\mathcal{G}_1$, and $(k_2 - k_1)$ sign queries to the Sign oracle. For the second attack method, the attacker needs $k_2 - k_1 + 2$ scalar multiplication operations, $2(k_2 - k_1) + 1$ point addition operations and 2 inverse operations in the group $\mathcal{G}_1$. According to elliptic curve cryptography, the most expensive operation is the pairing. The scalar multiplication, point addition, and inverse operation have less computation cost than the pairing. Hence, our two attack methods are computationally feasible.

When the original approach was fixed by adding some computation overhead, we analyze our two attack methods. Specifically, the verifier can choose several levels of signatures to verify. Or, the verifier chooses several nodes of signatures to verify. First, when the verifier can choose several levels of signatures to verify, according to the BAT algorithm, the verifier only needs to choose the deepest level to verify. When the deepest level passes the verification, other levels also can pass the verification. So, we only consider the chosen deepest level. Suppose that the chosen deepest level is the t-level, where t is chosen randomly. When $t < l$, the above two attacks can succeed. The probability of success is not less than $1 - \frac{h-l}{h} = \frac{l}{h}$. In order to reduce the computation overhead, the verifier would be willing to choose small $t < h$. Otherwise, the batch verification will become meaningless. Specifically, when $l = h - 1$, the probability of success of the two attacks is 1. Second, when the verifier chooses several random nodes of signatures to verify, according to the BAT algorithm, the number of children of the inner node (l, v) is $2^{h-l+1} - 2$. Then, the above probability of success of the above two attacks is $1 - \frac{2^{h-l+1}-2}{2^h} = 1 + \frac{1}{2^{h-1}} - \frac{1}{2^{l-1}}$. When $l = h - 1$, the probability of success is $1 - \frac{1}{2^{h-1}}$. So, our attack methods can succeed with a large probability.

According to the above analysis, our two attacks are feasible from the points of view of efficiency and success probability.

V. Our Improved BAT scheme

In light of the aforementioned weaknesses, we have improved Jiang *et al.*'s signature scheme for vehicular networks using a binary authentication tree. The vehicle's signature and binary authentication tree are the same as in Jiang *et al.*'s scheme. The difference lies in the phase of verification.

Consider 2^{h-l} message-signature pairs, $\{(M_{k_1}, \alpha_{k_1}), (M_{k_1+1}, \quad \alpha_{k_1+1}), \cdots, (M_{k_2}, \alpha_{k_2})\}$, where $\alpha_i = (E_i, F_i)$ for $i = k_1, k_1 + 1, \cdots, k_2$ is the basic signature scheme. All the signatures $\{\alpha_{k_1}, \alpha_{k_1+1}, \cdots, \alpha_{k_2}\}$ can be verified using the following steps:

1) Choose random vector $a_{k_1}, a_{k_1+1}, \cdots, a_{k_2} \in \{\mathcal{Z}_q^*\}^{2^{h-l}}$;
2) Check whether the following equation holds:

$$
\begin{aligned}
& e(\sum_{i=k_1}^{k_2} a_i F_i, P) \\
= \quad & e(\sum_{i=k_1}^{k_2} (a_i(E_i + h(M_i, E_i)H(PID_i))), P_{pub})
\end{aligned}
$$

where $k_1 = 2^{h-l}v$ and $k_2 = 2^{h-l}(v+1) - 1$.

A. Security Analysis

The improved scheme is the same as the original scheme except the verification phase. Thus, it can also offer some conventional security properties for vehicular communications, such as identity privacy and identity traceability. Although Jiang *et al.* claimed message integrity, they did not give any formal proof. Schemes without formal proofs may have security weaknesses. Our attacks show this point. We give the formal proof of unforgeability.

Theorem 1: Let $\mathcal{F}$ be a forger who performs an existential forgery attack against our scheme with adaptively chosen message and given identity, within a time bound T with probability ϵ in the random oracle model. The forger $\mathcal{F}$ queries the oracles Hash, Extract, Sign at most q_h, q_E, q_S-times, respectively. If $\epsilon \geq \frac{10(q_h+1)(q_h+q_S)}{2^k}$, then the CDH problem can be solved with probability $\geq \frac{1}{9}$ and within run time $\leq \frac{23(q_h+q_E)T}{\epsilon}$.

Due to the constraint of page limit, we omit the proof here.

Table II
PERFORMANCE COMPARISON OF SIGNATURE SCHEMES

	n authentic signatures	n signatures with $k \geq 1$ fake signatures
Improved Scheme	$2C_{par} + 3nC_{mul}$	$((k+1)\log_2(n/k) + 4k - 2)$ $C_{par} + (2(n-1)k + n)C_{mul}$
BAT	$2C_{par} + nC_{mul}$	$((k+1)\log_2(n/k) + 4k$ $-2)C_{par} + nC_{mul}$
Basic	$(2n+2)C_{par}$	$(2n+2)C_{par}$
ECDSA	$4nC_{mul}$	$4nC_{mul}$

B. Performance analysis

Let C_{mul} denote the time cost to perform one point multiplication over an elliptic curve, and C_{par} the time of a pairing operation. Since these operations dominate the verification overhead, we neglect all the other light-weight operations. Table II gives the comparison of four signature schemes in terms of verification overhead for n authentic single signatures and n signatures with k bogus signatures, respectively.

The improved scheme has the same pairing computation times as the BAT scheme. The performance analysis of the schemes Basic, ECDSA and BAT can be found in Jiang *et al.*'s paper [12]. The differences between our improved scheme and BAT lie in the computation of C_{mul}. In the case of n authentic signatures, our improved scheme needs $3nC_{mul}$ although BAT needs nC_{mul}. In the case of n signatures with k fake signatures, for every fake signature, our scheme can identify it with $\log_2 n$ verifications at most and needs $(2^{\log_2 n+1} - 1)C_{mul} = (2n-1)C_{mul}$ more operations than BAT. Thus, it needs at most $(2n-1)kC_{mul}$ more operations than BAT.

From Table II, we know that our improved scheme is more efficient than Basic and ECDSA. At the same time, it is less efficient than BAT. However, since BAT is insecure, our improved scheme is an attractive option. Note also that, since ECDSA is not an identity-based signature scheme, extra operations are needed to verify the public key certificate. Thus, the overall message verification time for ECDSA will be larger.

VI. CONCLUSIONS

In this paper, we point out an inherent design flaw in the binary authentication algorithm of [12]. We describe two attacks. In both of them, any inner node can impersonate other vehicles to generate any message's aggregate signature, *i.e.*, there exists a universal forgery attack on the binary authentication algorithm of [12] and therefore this scheme is insecure. Making use of the binary authentication tree model, we repair that flawed scheme and obtain a scheme that is provably secure and efficient in the random oracle model.

ACKNOWLEDGMENTS

The authors are with the UNESCO Chair in Data Privacy, but the views expressed in this article do not necessarily reflect the position of UNESCO nor commit that organization. This work was partly supported by the Spanish Government through projects TSI2007-65406-C03-01 "E-AEGIS", TIN2011-27076-C03-01 "CO-PRIVACY" and CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", by the Government of Catalonia under grant 2009 SGR 1135, by the European Commission under FP7 project "D-wB", by Natural Science Foundation of Liaoning Province (No. 20102042), by China Post-doctor Science Fund (No. 20110490061) and by Program for Liaoning Excellent Talents in University (No. LJQ2011078). The third author is partly supported as an ICREA Acadmia Researcher by the Government of Catalonia.

REFERENCES

[1] Dedicated Short Range Communications (DSRC), [On-line] http://grouper.ieee.org/groups/scc32/dsrc/index.html

[2] IEEE Standard 1609. 2 - IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages, July, 2006.

[3] W. Franz, C. Wagner, C. Maihofer, and H. Hartenstein, "Fleetnet: platform for inter-vehicle communications, " in Proc. 1st Intl. Workshop on Intelligent Transportation, Hamburg, Germany, 2004.

[4] "NoW: Network on Wheels Project, " [On-line] http://www.network-onwheels.de, 2007.

[5] "US Vehicle Safety Communication Consortium, " [On-line] http://wwwnrd.nhtsa.dot.gov/pdf/nrd-12/CAMP3/pages/VSCC.htm

[6] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures, " in Advances in Cryptology-CRYPTO 2004, LNCS 3152, pp. 41-55, 2004.

[7] C. Zhang, R. Lu, X. Lin, P. H. Ho, and X. Shen, "An efficient identitybased batch verification scheme for vehicular sensor networks, " in Proc. IEEE INFOCOM'08, 2008.

[8] A. Fiat, "Batch RSA, " in Advances in Cryptology-CRYPTO'89, LNCS 435, pp. 175-185, 1990.

[9] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps, " in Advances in Cryptology-EUROCRYPT 2003, LNCS 2656, pp. 416-432, 2003.

[10] H. Yoon, J. H. Cheon, and Y. Kim, "Batch verification with ID-based signatures, " in Proc. Information Security and Cryptology-ICISC 2004, LNCS 3506, pp. 233-248, 2004.

[11] J. Camenisch, S. Hohenberger, and M. Pedersen, "Batch verification of short signatures, " in Advances in

Cryptology-EUROCRYPT 2007, LNCS 4514, pp. 246-263, 2007.

[12] Y. Jiang, M. Shi, X. Shen, and C. Lin, "BAT: a robust signature scheme for vehicular networks using binary authentication tree", IEEE Transactions on Wireless Communications, vol. 8, no. 4, pp. 1974-1983, 2009.

[13] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing, " in Advances in Cryptology-CRYPTO 2001, LNCS 2139, pp. 213-229, 2001.

[14] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction, " IEICE Transactions Fundamentals, vol. 5, pp. 1234-1243, 2001.

[15] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing, " in Advances in Cryptology-ASIACRYPT 2001, LNCS 2248, pp. 514-532, 2001.

[16] A. Fiat, "Batch RSA", Journal of Cryptology, vol. 10, no. 2, pp. 75-88, 1997.

[17] D. Naccache, D. M'Rihi, D. Raphaeli, and S. Vaudenay, "Can DSA be improved? Complexity trade-offs with the digital signature standard", in Advances in Cryptology-EUROCRYPT 1994, LNCS 950, pp. 77-85, 1994.

[18] C. H. Lim, and P. J. Lee, "Security of interactive DSA batch verification, " Electronics Letters, vol. 30, no. 19, pp. 1592-1593, 1994.

[19] M. Bellare and J. A. Garay, "Fast batch verification for modular exponentiation and digital signatures, " in Advances in Cryptology-EUROCRYPT 1998, LNCS 1403, pp. 236-250, 1998.

[20] C. Boyd and C. Pavlovski, "Attacking and repairing batch verification schemes", in Advances in Cryptology-ASIACRYPT 2000, LNCS 1976, pp. 58-71, 2000.

[21] H. C. Jung and H. L. Dong, "Use of sparse and/or complex exponents in batch verification of exponentiations", IEEE Transactions on Computers, vol. 55, no. 12, pp. 1536-1542, 2006.

[22] A. L. Ferrara, M. Green, S. Hohenberger, and M.O. Pedersen, "Practical short signature batch verification", CT-RSA 2009, LNCS 5473, pp. 309-324, 2009.

[23] O. Goldreich, Foundations of Cryptography, Volume II, Basic applications, Cambridge University Press, 2004.

[24] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures, Journal of Cryptology, vol. 13, no. 3, pp. 361-396, Springer-Verlag, 2000.