ABAKA: An Anonymous Batch Authenticated and Key Agreement Scheme for Value-Added Services in Vehicular Ad Hoc Networks

Jiun-Long Huang, Lo-Yao Yeh, and Hung-Yu Chien

***Abstract*—In this paper, we introduce an anonymous batch authenticated and key agreement (ABAKA) scheme to authenticate multiple requests sent from different vehicles and establish different session keys for different vehicles at the same time. In vehicular ad hoc networks (VANETs), the speed of a vehicle is changed from 10 to 40 m/s (36–144 km/h); therefore, the need for efficient authentication is inevitable. Compared with the current key agreement scheme, ABAKA can efficiently authenticate multiple requests by one verification operation and negotiate a session key with each vehicle by one broadcast message. Elliptic curve cryptography is adopted to reduce the verification delay and transmission overhead. The security of ABAKA is based on the elliptic curve discrete logarithm problem, which is an unsolved NP-complete problem. To deal with the invalid request problem, which may cause the batch verification fail, a detection algorithm has been proposed. Moreover, we demonstrate the efficiency merits of ABAKA through performance evaluations in terms of verification delay, transmission overhead, and cost for rebatch verifications, respectively. Simulation results show that both the message delay and message loss rate of ABAKA are less than that of the existing elliptic curve digital signature algorithm (ECDSA)-based scheme.**

***Index Terms*—Authentication, batch verification, conditional privacy, elliptic curve cryptographic.**

I. Introduction

VEHICULAR ad hoc networks (VANETs) have appealed to many research interests from academia and deployment efforts from industries. Ranging from road safety and traffic management to value-added services, VANETs have been regarded as a killer application in the near feature. It is anticipated that vehicles equipped with wireless communication devices can communicate with each other and with roadside units (RSUs) located at critical points such as intersections or gas stations. Vehicles are expected to communicate by means of the Dedicated Short-Range Communication protocol (DSRC) standard [1], which applies the IEEE 802.11p standard for wireless communication. To offer communication with participants out of radio range, the messages could be forwarded by other vehicles (multihop communication).

The creation of VANETs is to enhance road safety and improve drivers' driving experiences, which are called safety-related applications. Onboard units (OBUs) equipped in vehicles periodically broadcast routine traffic-related messages with information such as position, current time, direction, speed, acceleration/deceleration, and traffic events. With that information, the driver can get a better understanding of their driving environment. In addition, value-added applications, which are called nonsafety applications, can also be envisioned to offer various entertaining services to drivers and passengers. Convenient value-added services include Internet access, navigation, instant messenger, toll payment service, and electronic advertizements [2], [3].

Along with the growth of VANETs, several challenges are emerging, such as security and privacy issues. Prior to realizing enjoyable value-added applications into practice in VANETs, we have to deal with security and privacy issues [4]–[8]. Fundamentally, we must guarantee identity authentication and data integrity. In value-added applications, confidentiality is also required. In addition, the requirement of privacy preservation must be reached in terms of user-related private information, including user identity and user location. Although several studies [3], [4], [7]–[10] have addressed the aforementioned issues, most of them are designed for safety-related applications to ensure message verification and integrity. It is obvious that attractive value-added services play an important role in raising the interests of consumers to take in VANETs. On the other hand, due to the speed of vehicles varying from 36 to 140 km/h [11], there is a unique stringent time requirement in vehicular communication [4], [7]. According to the DSRC standard [1], a vehicle sends a safety-related message to its neighboring RSU every 100–300 ms, which means that an RSU has to verify some 600 safety-related message/s if there are roughly 180 vehicles kept within the communication range of the RSU [7]. In other words, the security scheme for value-added applications should not pose a heavy burden on RSUs. Therefore, the burden may gather at a single authentication server, which incurs a bottleneck problem. Obviously, it is critical to develop an efficient and secure authentication scheme before value-added applications can take effect.

To tackle the aforementioned problems, including security, efficiency, and scalability problems, we proposed an anonymous batch authentication and key agreement (ABAKA)

Manuscript received February 12, 2010; revised May 28, 2010 and August 15, 2010; accepted October 15, 2010. Date of publication October 25, 2010; date of current version January 20, 2011. This work was supported in part by the National Science Council of Taiwan under Contract NSC 98-2219-E-002-021. The review of this paper was coordinated by Prof. Y. Zhang.

J.-L. Huang and L.-Y. Yeh are with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: jlhuang@cs.nctu.edu.tw; lyyeh@cs.nctu.edu.tw).

H.-Y. Chien is with the Department of Information Management, National Chi Nan University, Nantou 545, Taiwan (e-mail:hychien@ncnu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2010.2089544

scheme to build a secure environment for value-added services in VANETs. To avoid bottleneck problems, ABAKA is inspired by the concept of batch verification [7] to simultaneously authenticate multiple requests sent from different vehicles using elliptic curve cryptography (ECC), which is adopted by the IEEE Trial-Use standard [11]. Meanwhile, multiple session keys for different vehicles can also be negotiated at the same time. To the best of our knowledge, this is the first study that provides batch authenticated and key agreements for value-added applications in VANETs. ABAKA enjoys the following unparalleled features: 1) Multiple vehicles can be authenticated at the same time rather than one after the other. It is an appealing solution to elaborately solve the possible bottleneck problems. 2) Not only can batch authentication be achieved but batch key agreement can also be accomplished. Depending on different key agreement parameters sent from the requesting vehicles, ABAKA could negotiate a distinct session key with each vehicle to ensure the confidentiality of subsequent messages. 3) By creating distinct pseudoidentities and the corresponding private keys, the privacy regarding the real identity of a vehicle and private information is guaranteed. 4) The real identities of the vehicles can be uniquely revealed by the service provider (SP) under specific conditions. 5) Due to the advantage of tamper-proof devices in vehicles, the efforts on the storage cost and the transmission overhead can be significantly alleviated.

The remainder of this paper is organized as follows: Section II briefly introduces our system model, the preliminary, and the design objectives. In Section III, the proposed ABAKA scheme is presented in detail. Section IV gives a security analysis of the proposed scheme. The performance evaluation is shown in Section V. Section VI surveys the related works. Finally, we conclude in Section VII.

II. System Model and Preliminaries

A. System Model

We introduce a two-layer vehicular network model for value-added applications, as shown in Fig. 1. The lower layer is composed of vehicles and RSUs. The communications, either intervehicle communication (IVC) or roadside-to-vehicle communication, are based on the dedicated short-range communications (DSRC) standard [1]. According to the DSRC standard, the communication range of an RSU is adjustable, and therefore, it can be larger than that of the vehicles, meaning that some vehicles can hear messages sent from the RSU while the RSU may not hear messages sent from the vehicles. The upper layer comprises various SPs and a trust authority (TA). The SPs have made a contract with the TA because the SPs will set up the system parameters in vehicles with the aid of the TA. The SPs can be connected with RSUs through secure channels, such as the transport layer security protocol by wired or wireless connections. The SPs provide various services, such as multimedia streams, instant messenger, and navigation services, and RSUs serve as gateways to deliver data to the requesting vehicles. According to the current VANET security standard [11], before messages are sent, OBUs should sign the messages with their private keys issued by the TA to ensure the integrity

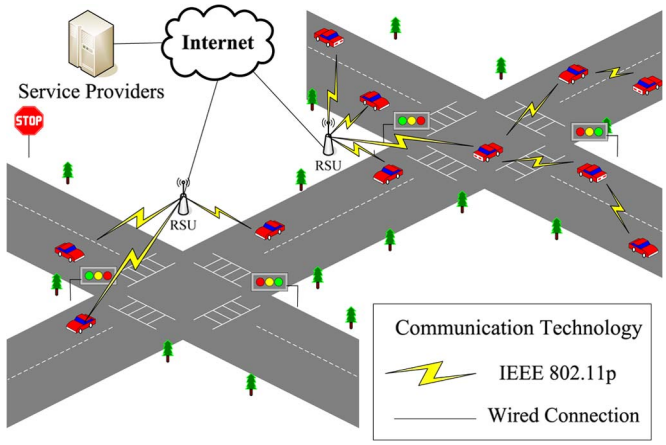

Fig. 1. Network model for value-added service.

of the messages. Then, each RSU is responsible for checking the integrity and forwarding the valid messages to the SPs.

In this paper, ABAKA aims to address the security between the SPs and vehicles. We assume that the TA is trusted and will never be compromised, which is often assumed in VANET schemes [6]–[8]. In addition, the SPs will not be compromised in the system initialization phase.[1]

B. ECC Preliminaries

An elliptic curve is a cubic equation of the form $\mathrm{y}^2 + a\mathrm{xy} + b\mathrm{y} = \mathrm{x}^3 + c\mathrm{x}^2 + d\mathrm{x} + e$, where a, b, c, d, and e are real numbers. In an ECC system, the elliptic curve equation is defined as the form of $\mathrm{E}_p(a,b) : \mathrm{y}^2 = \mathrm{x}^3 + a\mathrm{x} + b \ (\mathrm{mod}p)$ over a prime finite field F_p, where $a, b \in \mathrm{F}_p$, $p > 3$, and $4a^3 + 27b^2 \neq 0 \ (\mathrm{mod}p)$ [13]. Given an integer $s \in \mathrm{F}_p$ and a point $P \in \mathrm{E}_p(a,b)$, the point multiplication sP over $\mathrm{E}_p(a,b)$ can be defined as $sP = \underbrace{P + P + \cdots + P}_{s}$. In general, the security of ECC depends on the difficulties of the following problems [14], [15].

Definition 1: Given two points P and Q over $\mathrm{E}_p(a,b)$, the elliptic curve discrete logarithm problem (ECDLP) finds an integer $s \in \mathrm{F}_p$ such that $Q = sP$.

Definition 2: Given three points P, sP, and tP over $\mathrm{E}_p(a,b)$ for $s, t \in \mathrm{F}_p$, the computational Diffie–Hellman problem finds the point $(s \cdot t)P$ over $\mathrm{E}_p(a,b)$.

Up to now, there is no polynomial algorithm that is able to solve any of the aforementioned problems [14], [15]. Compared with the counterpart scheme [7], the proposed scheme exploits the point multiplication over ECC, instead of bilinear pairing, to reduce the computational cost.

C. Security Objectives

For value-added applications in VANETs, a secure system should meet four security objectives.

1) Mutual authentication: The communication parties should be authenticated to guard against the

[1]The TA can adopt Kerberos scheme [12] to guarantee the genuineness of the SPs in the system initialization phase.

TABLE I
NOTATIONS

Notation	Descriptions
V_i	The ith vehicle
RSU	A roadside unit
SP	A service provider
$RVID$	The real identity of a vehicle
SID	The identity of the service provider
G	A cyclic additive group
P	The generator of the cyclic group G
q	The order of the group G
v	The private secret of the tamper-proof device
PK_{SP}	The SP's public key preloaded in each vehicle
RK_{SP}	The SP's private key
PWD	A password to activate a tamper-proof device
ID_i	A pseudo identity of the vehicle V_i
ID_i^j	A part of the ID_i, where j=1 or 2 and ID_i=(ID_i^1‖ ID_i^2)
mrk	A master private key
CRK_i	A corresponding private key of the vehicle V_i
T	The timestamp
$\triangle T$	The predefined endurable transmission delay
KP	The key parameter used for key agreement
$h(.)$	A collision-free one-way hash function h: $\{0, 1\}^* \rightarrow \mathbb{Z}_q^*$
$H(.)$	A MapToPoint hash function such as H: $\{0, 1\}^* \rightarrow G$
‖	Message concatenation operation
$\oplus$	Exclusive-OR operation

impersonation attack. The previous prominent works [5]–[9] are designed for safety-related applications focusing on the message authentication. For value-added applications, the mutual authentication between the vehicles and SPs is essential.

2) Session key establishment: To ensure data confidentiality, session key establishment is indispensable for value-added applications in VANETs. With the session key, the SP can build a secure communication path with the requesting vehicle for subsequent communications for various value-added services such as multimedia streams.
3) Privacy preservation: The identities of vehicles should be hidden from a message receiver during the authentication process to keep the senders' personal information private.
4) Low transmission overhead and fast verification: Due to the stringent time requirement in VANETs, the security scheme should consider the efficiency into account. The lower the transmission overhead, the better; many requests should be verified as soon as possible.

III. Anonymous Batch Authentication and Key Agreement Scheme

In this section, we propose a novel ABAKA scheme for value-added applications in VANETs. ABAKA consists of the following three phases: 1) the system initiation phase; 2) the pseudoidentity generation phase; and 3) the batch authentication and key agreement phase. A new vehicle first performs the system initiation phase to preload the system parameters. Then, the pseudoidentity generation phase is used to generate the pseudoidentity and corresponding private key for privacy issue. Finally, the batch authentication and key agreement phase is executed when the vehicle wants to access services provided by SPs. The notations throughout this paper are listed in Table I.

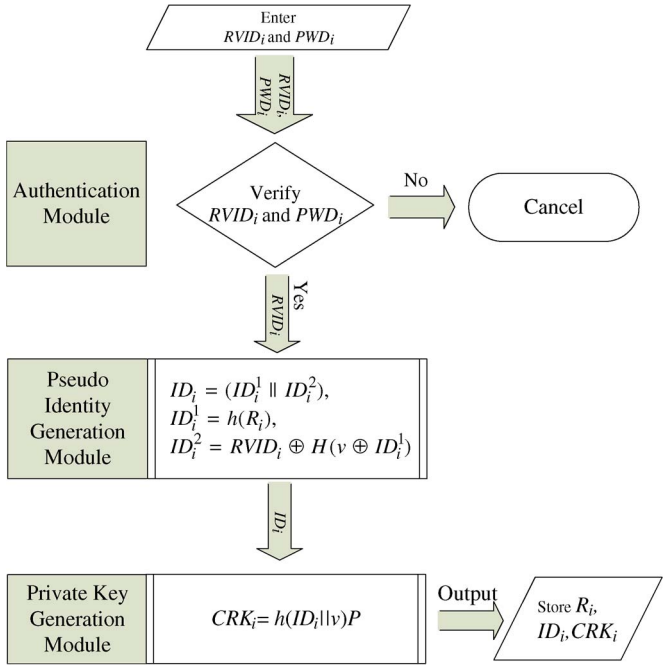

Fig. 2. Procedures of tamper-proof device.

A. System Initiation

First, we assume that each vehicle is equipped with a tamper-proof device, which is secure against any compromise attempt in any circumstance. Note that the use of a tamper-proof device is recommended by the current VANET security standard [11] and several well-known VANET schemes [3], [7] to reduce the risk of vehicles compromised by adversaries. Due to tamper-proof devices on vehicles, an adversary can attain no data stored in the device [3], [7], [10]. Initially, the SP sets up three system parameters for vehicles that have made purchase contracts with the SP.

1) Let G be a cyclic additive group generated by P with the order q.
2) The SP randomly chooses $v \in \mathbb{Z}_q^*$ as the private secret, and v will be loaded in the vehicles' tamper-proof devices.
3) Each vehicle is preloaded with the public parameters $\{G, q, P, PK_{\rm SP}, h(), H()\}$, and $\{v, mrk = h(v\|SID)\}$ are preloaded in the tamper-proof device.

B. Pseudoidentity Generation

For privacy preservation, similar to [7], the tamper-proof device is responsible for generating random pseudoidentities and corresponding private keys CRK based on ECC [13]. The tamper-proof device consists of three modules: 1) authentication module; 2) pseudoidentity generation module; and 3) corresponding private key generation module. Fig. 2 shows the procedures of the tamper-proof device.

1) *Authentication module*: The authentication module is used to ensure the validity of the user. A user inputs its unique real vehicle identity $RVID_i \in G$ and $PWD_i \in \{0,1\}^*$ to pass the verification of the authentication

module. If both $RVID_i$ and corresponding PWD_i are valid, $RVID_i$ is delivered to the next module, i.e., the pseudoidentity generation module; otherwise, the tamper-proof device refuses to activate itself. Ensured by the authentication module, an adversary cannot get any information, even though the tamper-proof device is compromised by the adversary.

2) *Pseudoidentity generation module*: The pseudoidentity generation module takes charge of generating a random pseudoidentity ID_i for the purpose of anonymity. Each ID_i is composed of two parts ID_i^1 and ID_i^2. Upon receiving $RVID_i$, the pseudoidentity generation module chooses a random number $w_i \in \mathbb{Z}_q^*$ to create a point $R_i \in G$ such that $R_i = (x_i, y_i) = w_i P$. Note that the coordinates (x, y) of each point are in the finite field F_q so that both x and y are integers [13], [16].[2] Then, let $ID_i^1 = h(R_i)$ and $ID_i^2 = RVID_i \oplus H(v \| ID_i^1)$, which allows only the SP to reveal the real identity $RVID_i$ of V_i. Finally, the tamper-proof device sends ID_i to the next module, i.e., the private key generation module.

3) *Private key generation module*: This module manages the generation of the corresponding private key based on the pseudoidentity ID_i. The corresponding private key CRK_i is set to $h(ID_i \| v)P$.

In the end, V_i can store a list of random point R_i and pseudoidentities $ID_i = (ID_i^1 \| ID_i^2)$ with its corresponding private keys CRK_i. Note that the generation of pseudoidentities and private keys can be finished offline with no delay.

C. Batch Authentication and Key Agreement

In this phase, there are three kinds of procedures: 1) the request procedure; 2) the vehicle verification and key agreement procedures; and 3) the mutual authentication and key agreement procedures. First, the request procedures are initiated by a vehicle when the vehicle wants to access some services provided by an SP. Next, the vehicle verification and key agreement procedures are performed by the SP to check the validity of the requesting vehicle and to negotiate a session key for the confidentiality of subsequent communications. Finally, the mutual authentication and key agreement procedures are executed by the requesting vehicle to check the validity of the SP. After performing the three procedures, a session key shared by both the requesting vehicles and the SP is generated to secure the subsequent communications.

We first introduce the request procedures launched by vehicles. Next, we elaborate on the vehicle verification and key agreement procedures in terms of dealing with a single request and multiple requests. Finally, the mutual authentication and key agreement procedures are discussed to ensure the validity of the SP.

1) Request Procedures: With the tamper-proof device, V_i obtains a random point $R_i \in G$, a pseudoidentity ID_i, and the corresponding private key CRK_i, as well as the master private key mrk. To issue a request, V_i executes the following procedures.

1) To ensure the freshness, V_i first generates $t_i = h(T_i)$, where T_i denotes the current timestamp. Note that it is assumed that each vehicle can perform time synchronization using the tamper-proof device [3].
2) With the aforementioned values, V_i can calculate $M_i = R_i + t_i CRK_i$ and $F_i = (mrk \cdot x_i)P$, where x_i is the x coordinate of point R_i.
3) Finally, according to the request packet format shown in Fig. 3, V_i delivers the request packet $\langle ID_i, M_i, F_i, T_i \rangle$ to the SP with the help of the neighboring RSU.

Pseudo Identity (*ID*)	Material Message (*M*)	Verification Message (*F*)	Timestamp (*T*)
40 bytes	20 bytes	20 bytes	4 bytes

Fig. 3. Request packet format.

Notice that, although there is no explicit key agreement parameters transmitted in the request packet, ABAKA takes the advantage of the random point R_i as the Diffie–Hellman key agreement parameter to save the transmission overhead. The request message packet defined in Fig. 3 is composed of the following: The pseudoidentity ID is in the first field, the second field is the material message M, the verification message F is in the third field, and the last field stores the current timestamp T for withstanding replay attacks. Here, ABAKA adopts SHA-1 as the underlying hash algorithm and uses the MNT curve [17] with 160-bit prime order q.

2) Vehicle Verification and Key Agreement Procedures: Based on the system model described in Section II-A, the SP is responsible for authenticating and negotiating a session key with each requesting vehicle. In some situations, numerous requests may crowd in the SP at the same period. To mitigate possible bottleneck problems, we propose a batch verification and key agreement scheme. For ease of presentation, we introduce the verification of a single request, followed by the presentation on the batch verification of multiple requests.

1) Single-request authentication: Given the request $\langle ID_i, M_i, F_i, T_i \rangle$ from V_i, the SP performs five steps.

1) For freshness, we assume that the receiving time is $T_{\mathrm{SP\_now}}$. The SP checks whether $\Delta T \geq T_{\mathrm{SP\_now}} - T_i$ is valid, where ΔT is the predefined endurable transmission delay. If yes, then go to step 2; otherwise, the SP ceases this connection.
2) To ensure the legitimacy of this request, the SP calculates $CRK_i = h(ID_i \| v)P$, depending on the pseudoidentity ID_i, public parameters, and his own private secret v, and computes $t_i = h(T_i)$. With CRK_i and t_i, the SP acquires the point $\hat{R}_i = M_i - t_i CRK_i$, where $\hat{R}_i = (\hat{x}_i, \hat{y}_i)$ and verifies whether $F_i \stackrel{?}{=} (h(v \| SID) \cdot \hat{x}_i)P$ is held or not. If so, then go to the next step; otherwise, this connection is terminated.
3) For mutual authentication, the SP picks a random number $z \in \mathbb{Z}_q^*$ and computes a point $R_{\mathrm{SP}} \in G$ such that $R_{\mathrm{SP}} = zP$. Next, the SP also signs $\{R_{\mathrm{SP}}, T_{\mathrm{SP}}\}$, where

[2]The points on the y-axis are not recommended to be chosen in our scheme.

Material Message (R)	Timestamp (T)	Signature (σ)
20 bytes	4 bytes	56 bytes

Fig. 4. Response packet format.

T_{SP} denotes the SP's current timestamp, to produce an elliptic curve digital signature algorithm (ECDSA) signature σ_{SP}. Note that the point R_{SP} also serves as the Diffie–Hellman key parameter.

4) Therefore, the SP can negotiate the session key $SK_{\mathrm{SP}i} = z\hat{R}_i = zw_iP$ to protect the subsequent communications.
5) Finally, based on the response packet format shown in Fig. 4, the SP sends the values $\langle R_{\mathrm{SP}}, T_{\mathrm{SP}}, \sigma_{\mathrm{SP}}\rangle$ back to the requesting vehicle.

Batch verification and key agreement: Given n distinct requests denoted as $\langle ID_1, M_1, F_1, T_1\rangle, \langle ID_2, M_2, F_2, T_2\rangle, \ldots, \langle ID_n, M_n, F_n, T_n\rangle$ sent from $V_1, V_2, \ldots, V_n$, respectively. Similar to the verification of a single request, six steps are performed by the SP.

1) To withstand replay attacks, we assume that the receiving time is $T_{\mathrm{SP\_now}}$. The SP checks whether $\Delta T \geq T_{\mathrm{SP\_now}} - T_i$ is valid, where ΔT is the predefined endurable transmission delay. If yes, then go on; otherwise, the SP ceases this connection.
2) To ensure the validity, the SP calculates $CRK_i = h(ID_i\|v)P$, depending on the pseudoidentity ID_i (for $1 \leq i \leq n$), public parameters, and his own private secret v, and computes $t_i = h(T_i)$. With the CRK_i and t_i, the SP individually extracts the random point $\hat{R}_i = M_i - t_iCRK_i$, where $\hat{R}_i = (\hat{x}_i, \hat{y}_i)$ for $1 \leq i \leq n$. Up until now, the steps are the same as that of single-request verification. The steps given here are designed for batch verification and key agreement.
3) To verify a batch of requests, the SP accumulates $\sum_{i=1}^{n} F_i = \sum_{i=1}^{n}(mrk \cdot x_i)P = h(v\|SID)(\sum_{i=1}^{n} x_i)P$ and computes $(\sum_{i=1}^{n} \hat{x}_i)$ to verify whether $\sum_{i=1}^{n} F_i \stackrel{?}{=} h(v\|SID)(\sum_{i=1}^{n} \hat{x}_i)P$ is valid or not. If so, then go to the next step; otherwise, this connection is terminated.
4) For mutual authentication, the SP picks a random number $z \in \mathbb{Z}_q^*$ and produces a point $R_{\mathrm{SP}} \in G$ such that $R_{\mathrm{SP}} = zP$. By the SP's private key RK_{SP}, the SP also signs $\{R_{\mathrm{SP}}, T_{\mathrm{SP}}\}$ to generate ECDSA signature σ_{SP}. Because of the ECDSA signature, the SP can generate only a single message to broadcast for a batch of the requesting vehicles. Each vehicle can verify the signature by the SP's public key PK_{SP} to assure the validity of the SP and the integrity of the message.
5) The SP negotiates the session keys $SK_{\mathrm{SP}i} = z\hat{R}_i = zw_iP$ with V_i, where $1 \leq i \leq n$, to protect the subsequent communications. Note that the session keys are distinct because of the different w_i's sent from different vehicles.
6) Finally, following the response packet format, the SP broadcastly sends $\langle R_{\mathrm{SP}}, T_{\mathrm{SP}}, \sigma_{\mathrm{SP}}\rangle$ back to the vehicles.

To be precise, the response packet format, as shown in Fig. 4, consists of the material message, timestamp, and ECDSA signature.[3]

3) Mutual Authentication and Key Agreement Procedures: Given the response packet $\langle R_{\mathrm{SP}}, T_{\mathrm{SP}}, \sigma_{\mathrm{SP}}\rangle$ sent from the SP, V_i carries out two steps to mutually authenticate the validity of the SP and to negotiate a session key for the confidentiality of the subsequent communications.

1) For freshness, V_i checks whether $\Delta T \geq T_{V\_\mathrm{now}} - T_{\mathrm{SP}}$ is valid, where ΔT is the predefined endurable transmission delay, and T_{V_now} is V_i's receiving time. If not, this session is dropped; otherwise, V_i first verifies the signature σ_{SP} to ensure the integrity of the message. If σ_{SP} is legal, V_i goes to the next step; otherwise, this connection is terminated.
2) For key agreement, V_i computes the session key $SK_{\mathrm{SP}i} = w_iR_{\mathrm{SP}} = w_izP$ to encrypt the messages in the subsequent communications.

The details of the proposed ABAKA scheme are shown in Fig. 5. Note that the key confirmation can be checked in the following communications to reduce transmission cost. For example, the first encrypted message can be $E_{SK_{\mathrm{SP}i}}(R_i, R_{\mathrm{SP}}, Msg)$, which can be used to withstand parallel session attacks [18].

D. Discussion

1) Reliability Analysis: In the section, we discuss the reliability of ABAKA. Due to batch verification, ABAKA enjoys several advantages such as lower verification delay and transmission overhead. However, the expense of the batch verification is that, once an invalid request exists in a batch of requests, the batch verification may lose its efficacy. Note that the invalid request could come from a variety of reasons such as packet loss, wireless channel interference, or the involvement of malicious attackers. This problem commonly accompanies other batch-based verification schemes [7], [19]. To deal with this problem, we carefully analyze what happens if the problem occurs.

First, we develop a probabilistic model for characterizing the risk of some requesting vehicles suffering from packet loss or sending bogus messages to pass the batch authentication based on the following assumptions.

1) According to [5], the average packet loss ratio is almost lower than 0.07%, whereas the velocity of vehicles is changing from 10 to 40 m/s (36–144 km/h). On the other hand, if an attacker plans to send a bogus message at will, RSUs can rule out the bogus message if the signature verification of the message fails. A possible case is that an attacker uses his valid private key issued by TA to sign a bogus message designed to pass the SP's batch authentication. In this case, RSUs will forward the bogus message to the aimed SP. However, once the attacker is detected by the SP, the SP can inform the TA to revoke the

[3]In ABAKA, we adopt ECDSA-224, which is also recommended by the current VANET standard [11].

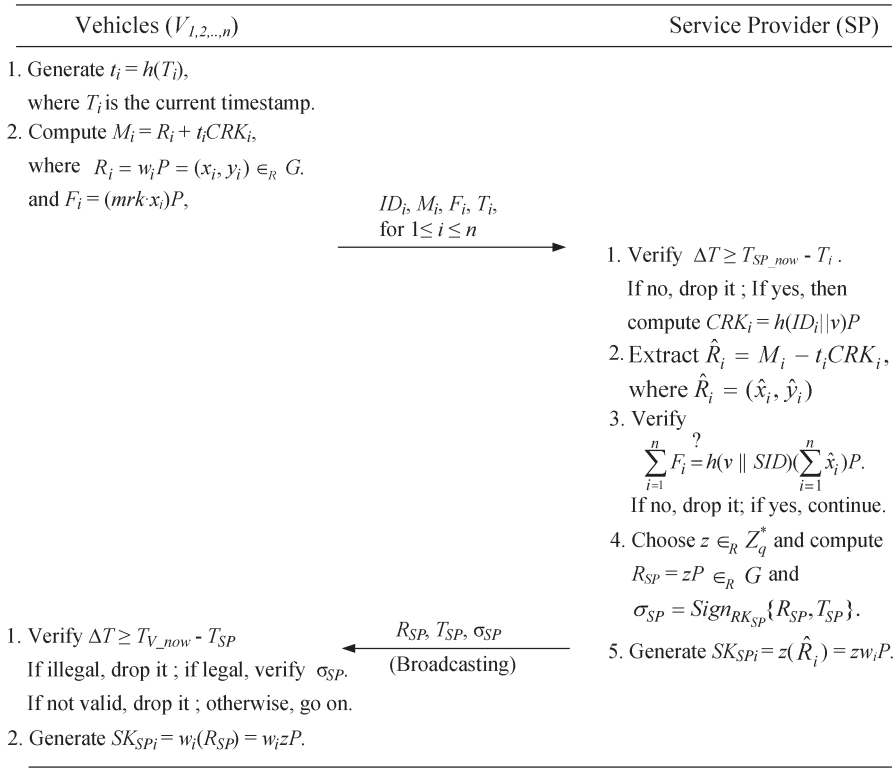

Fig. 5. ABAKA scheme.

attacker's certificate, which can prevent the attacker from sending a bogus message in the future. Using the two aforementioned protection mechanisms, we assume that at most 1% registered vehicles[4] can be compromised and send an invalid message passing the signature verification of RSUs to the SP in a batch period. While the number of registered vehicles (denoted as N_{Reg}) is assumed to be 10^4, the largest number of compromised vehicles (which is denoted as N_C) is $N_{\text{Reg}} \times 1\% = 10^4 \times 1\% = 100$. For ease of analysis, we assume that a vehicle at most sends a request in a batch period.

2) In a period, the number of requests that an SP can process in a batch authentication is defined as N_B. Then, when one or more malicious requests are within N_B, the other requests in the same batch are needed to re-authenticate, which is referred to as rebatch authentication in this paper.

Let $Pr\{i\}$ represent the probability that exactly i invalid requests sent from N_C are being sent to the SP. The probability follows the hypergeometric distribution $\mathcal{H}(i, N_{\text{Reg}}, N_C, N_B)$ as follows:

$$Pr\{X = i\} = \frac{\binom{N_{\text{Reg}}-N_C}{N_B-i}\binom{N_C}{i}}{\binom{N_{\text{Reg}}}{N_B}}, \quad i = 0, 1, \ldots, 100$$

That is, in a period, there are N_B requests to be authenticated, i invalid requests sent from N_C, and $N_B - i$ valid requests sent from $N_{\text{Reg}} - N_C$. Let A be the event that rebatch verification is required to successfully verify all valid requests N_B. Then, $Pr\{A\}$ can be represented as

$$\begin{aligned} Pr\{A\} &= Pr\{i = 1\} + \ldots + Pr\{i = 100\} \\ &= \frac{\binom{N_{\text{Reg}}-N_C}{N_B-1}\binom{N_C}{1}}{\binom{N_{\text{Reg}}}{N_B}} + \ldots + \frac{\binom{N_{\text{Reg}}-N_C}{N_B-100}\binom{N_C}{100}}{\binom{N_{\text{Reg}}}{N_B}} \\ &= \frac{\sum_{i=1}^{100}\binom{N_{\text{Reg}}-N_C}{N_B-i}\binom{N_C}{i}}{\binom{N_{\text{Reg}}}{N_B}}. \end{aligned}$$

That is, there is at least an invalid request in a batch, which leads to the failure of batch verification. Hence, rebatch verifications are required. Then, we demonstrate the relationship between the number of compromised vehicles and that of requests in a batch in Fig. 6. In Fig. 6, the number of compromised vehicles is assumed to be 0–100, and batch verification can simultaneously authenticate 0–100 requests. We can observe that the probability of rebatch verification is at most about 0.42 while only one invalid request $(i = 1)$ in a batch and dramatically drops to 0.18 while there are two invalid requests $(i = 2)$ in the batch. The probability is almost negligible: approximately lower than 0.06 while there are more than two invalid requests $(i \ge 3)$ in a batch. To tackle the invalid request problem, we further propose a detection algorithm to find the invalid request in the next section. Based on the proposed detection algorithm, we discuss the cost of a rebatch verification in Section V-C. Moreover, we examine the expected verification

[4]In [6], the attacker can compromise at most 0.2% entities subordinated by the TA.

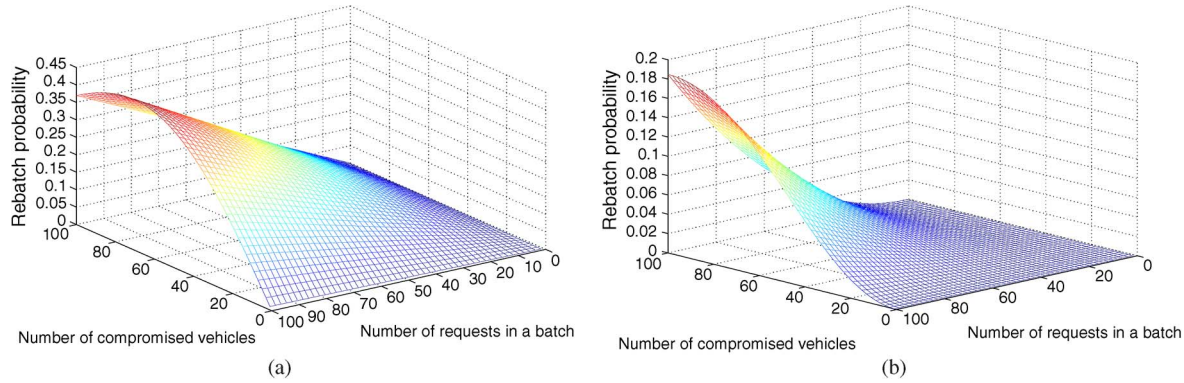

Fig. 6. Rebatch probability in ABAKA under different N_C's and different N_B's, where $1 \leq N_C \leq 100$, and $1 \leq N_B \leq 100$. (a) $i = 1$. (b) $i = 2$.

delay, including the original verification cost and the expected rebatch verification cost of Section V-D. The result shows that ABAKA can enjoy the more efficient than conventional ECDSA and other batch-based schemes, even if the invalid request problem exists.

2) Invalid Request Detection: In Section III-D1, we discuss the probability of rebatch authentication. In this section, we provide a detection algorithm to detect invalid requests. The concept of the detection algorithm is based on the "divide-and-conquer" approach [20]. When failing to verify a batch of requests, the SP can divide the batch into several subbatches and then separately check the validity of each subbatch. If the number of requests in a subbatch is left to be only one and the request still remains invalid, then the SP determines that this request in the subbatch is invalid. The detection algorithm with binary divisions is shown in Algorithm 1.

Algorithm 1: Detection algorithm

- Data: The SP received a batch of requests $BR = \{Req_1, Req_2, \ldots, Req_n\}$
- Result: Output the invalid requests if there are invalid requests in RB; otherwise, return $Ture$.

```
DetAlg(BR):
  begin
if    BatchVerify(BR) then
        return True;
      else if Num(BR) == 1 then
        return ID_i ∈ BR as an invalid request;
      else
        set BR_Front = {Req_1, Req_2, ..., Req_⌈n/2⌉};
        set BR_Rear = {Req_⌈n/2⌉+1, Req_⌈n/2⌉+2, ..., Req_n};
          DetAlg(BR_Front);
          DetAlg(BR_Rear);
      end if
  end
```

IV. Security Analysis

As mentioned in Section II-C, we analyze the security objectives of the proposed ABAKA as follows.

1) Mutual authentication: ABAKA achieves mutual authentication between the SP and requesting vehicles based on the ECDLP and ECDSA certificates. To be authenticated by the SP, a requesting vehicle V_i must be able to produce the corresponding private key CRK_i to conceal the random point R_i into M_i and to generate the valid verification message $F_i = (mrk \cdot x_i)P$, where x_i is the x coordinate of point R_i. Without knowing the corresponding private key CRK_i, it is computationally infeasible to forge a valid pair (M_i, F_i). By only knowing M_i and t_i, it is still difficult to obtain R_i and CRK_i. That is, $M_i = R_i + t_i CRK_i$ is a Diophantine equation, which is also the security fundamental of the IBV scheme [7]. Without knowing R_i and mrk, attackers cannot counterfeit the verification message F_i as well, which is based on ECDLP. Notice that, even if there are insider attackers who are legitimate users trying to impersonate other users, the insider attack is also withstood by the difficulty of ECDLP. Since the insider attacker neither realizes the random point R_i nor acquires the corresponding private key CRK_i, the insider attackers are not able to forge a valid F_i. On the other side, the authenticity of the SP is guaranteed by ECDSA, which is also adopted as the current standard [11] in VANETs.
2) Session key establishment: For confidentiality of subsequent communications between the SP and requesting vehicles, ABAKA provides the capability of negotiating session keys with vehicles. We exploit the concept of the ECDH key exchange protocol [13] to establish the session keys. The random points $R_i = w_i P$ and $R_{\mathrm{SP}} = zP$ serve as the exchange key parameters chosen by the requesting vehicles and SP, respectively. Only the legitimate SP and vehicles are able to compute the session key $SK_{\mathrm{SP}i} = z(R_i) = zw_i P = w_i zP = w_i(R_{\mathrm{SP}})$, which also relies on the difficulty of ECDLP. Moreover, the session key enjoys the perfect forward secrecy, where, even if a long-term secret is compromised, the previous session keys still remain confidential. In summary, ABAKA can securely negotiate a session key shared between the SP and each vehicle.
3) Privacy preservation: The privacy of each vehicle can be well protected by the pseudoidentities $ID_i = (ID_i^1 \| ID_i^2)$, where $ID_i^1 = h(R_i)$, and $ID_i^2 = RVID \oplus H(v \oplus ID_i^1)$. ID_i^1 and ID_i^2 are made up of a one-way hash function and XOR operation without leaking any

TABLE II
COMPARISONS OF VERIFICATION DELAY (IN MILLISECONDS)

	Authenticate a single request		Authenticate *n* requests	
	Vehicle auth.	SP auth. with key agree.	Vehicles auth.	SP auth. with key agree.
ABAKA	$3T_{mul}$	$7T_{mul}$	$(2n+1)T_{mul}$	$(n+6)T_{mul}$
IBV	$3T_{par} + T_{mtp} + T_{mul}$	N./A.	$3T_{par} + nT_{mtp} + nT_{mul}$	N./A.
BLS	$4T_{par} + 2T_{mtp}$	N./A.	$(2n+2)T_{par} + 2nT_{mtp}$	N./A.
ECDSA-AKA	$4T_{mul}$	$4T_{mul}$	$4nT_{mul}$	$4nT_{mul}$

identity information. Moreover, ABAKA achieves the conditional privacy, meaning that the SP should be able to realize who is accessing the services by computing $ID_i^2 \oplus H(v \oplus ID_i^1) = RVID \oplus H(v \oplus ID_i^1) \oplus H(v \oplus ID_i^1) = RVID$. Therefore, the requirement of conditional privacy preservation is met.

4) Low transmission overhead and fast verification: In terms of transmission overhead, a requesting vehicle does not need the signature and corresponding public key certificate in each request message. Although the SP computes a signature and broadcastly sends to all requesting vehicles in the response packet, we can preload the SP's public key into each vehicle to mitigate the transmission overhead. Note that the SP's public key is fixed, so we can preload it in the system initiation phase [3]. Moreover, we can only broadcast the signature to a few RSUs where there are requesting vehicles, instead of sending it to all vehicles. As for fast verification, ABAKA adopts the concept of batch verification to simultaneously authenticate a batch of requests. The more requests come, the more performance advantages of our scheme emerge, which is demonstrated in Section V-A.

Proposition 1: Batch verification is successful if and only if all individual requesters are valid.

Proof: ($\Longrightarrow$) If batch verification is successful, then all individual requests are valid. Because batch verification is successful, $\sum_{i=1}^{n} F_i = \sum_{i=1}^{n} (mrk \cdot x_i)P = h(v\|SID)(\sum_{i=1}^{n} \hat{x}_i)P$ is held. Using the tamper-proof device, the value of $mrk = h(v\|SID)$ can be derived by only the SP and P is the public generator of the cyclic additive group G, meaning that it is not easy to forge. Then, it represents that $\sum_{i=1}^{n} \hat{x}_i$ is valid. Each $\hat{x}_i$ is the x coordinate of the point $\hat{R}_i$ derived from M_i sent by V_i. The bits of each $\hat{x}_i$ are at least 224 bits, meaning that the probability that an attacker can guess a correct $\hat{x}_i$ with corresponding M_i is extremely low, i.e., by more than $1/2^{224}$. Note that, because each $\hat{x}_i$ is the x coordinate of the point $\hat{R}_i$, the value of $\hat{x}_i$ cannot be tampered with at will. As a result, it is reasonable to infer that each individual request is valid if batch verification is successful.

($\Longleftarrow$) If all individual requests are valid, then batch verification is successful. As long as each individual request is valid, each tuple (ID_i, M_i, F_i, t_i) can be correctly verified by the formula $F_i \stackrel{?}{=} (h(v\|SID) \cdot \hat{x}_i)P$, where $\hat{x}_i$ is the x coordinate of the point $\hat{R}_i$, and $\hat{R}_i = M_i - t_i \cdot h(ID_i\|v)P$. Then, we can accumulate all individual F_i into $\sum_{i=1}^{n} F_i = h(v\|SID)(\sum_{i=1}^{n} x_i)P$. As a result, the formula $\sum_{i=1}^{n} F_i = h(v\|SID)(\sum_{i=1}^{n} x_i)P \stackrel{?}{=} h(v\|SID)(\sum_{i=1}^{n} \hat{x}_i)P$ will hold since each x_i is the same as $\hat{x}_i$. Note that $\hat{x}_i$ is extracted by the SP from the receiving M_i. ■

V. PERFORMANCE EVALUATIONS

In this section, we first evaluate the performance of ABAKA in terms of the verification delay, transmission overhead, and verification cost for rebatch verifications by analytical analysis. In [8], an authentication and key establishment protocol with the ECDSA signature scheme, which is referred to as the ECDSA-AKA scheme in this paper, has been proposed. Here, we compare ABAKA with some related protocols, such as IBV [7], BLS [21], [22], and ECDSA-AKA [8]. Note that ECDSA is the current standard signature algorithm adopted by IEEE 1609.2 [11],[5] whereas IBV and IBS are notable batch-based verification schemes. Next, we further verify the efficiency and applicability of the proposed ABAKA in real-world environment using ns-2 [23]. In addition, to fully estimate the road environment and vehicular traffic, a well-known mobility model generation tool called TraNS [24] is adopted in the simulation.

A. Verification Delay

First, we define the time complexity of the cryptographic operations required in ABAKA and other schemes. Let T_{mul} denote the time to perform one point multiplication over an elliptic curve, T_{par} be the time to execute a pairing operation, and T_{mtp} represent the time of a MaptoPoint hash operation. Since the three operations dominate the speed of verification, we only consider the three operations and neglect the other operations such as additive and one-way hash function. Here, we adopt the experiment in [25] for an MNT curve [17] of embedding degree $k = 6$ and 160-bit q. The implementation was executed on an Intel Pentium IV 3.0-GHz Machine. The following results are obtained: T_{mul} is 0.6 ms, and T_{par} is 4.5 ms. T_{mtp} takes the same time as T_{mul}.

Table II shows the verification delay of all schemes in terms of authenticating a single request and n requests. Notice that IBV and BLS are designed for the message verification without mutual authentication and key agreement. For fairness, we compare the verification delay of the ABAKA, IBV, BLS, and ECDSA-AKA schemes in the one-way authentication case.[6] Then, we discuss the verification delay of the ABAKA and ECDSA-AKA schemes since both schemes provide the functionality of mutual authentication and key agreement. Fig. 7(a) shows the effect on the verification delay of all schemes in the one-way authentication case while the number of requests

[5]In the DSRC standard, ECDSA is used for message verification. Considering mutual authentication and key agreement, ECDSA-AKA [8] can be employed.

[6]Message verification can be regarded as one-way authentication since the signature can be used to manifest the identity of the user. However, one-way authentication does not provide the functionality of the session key agreement.

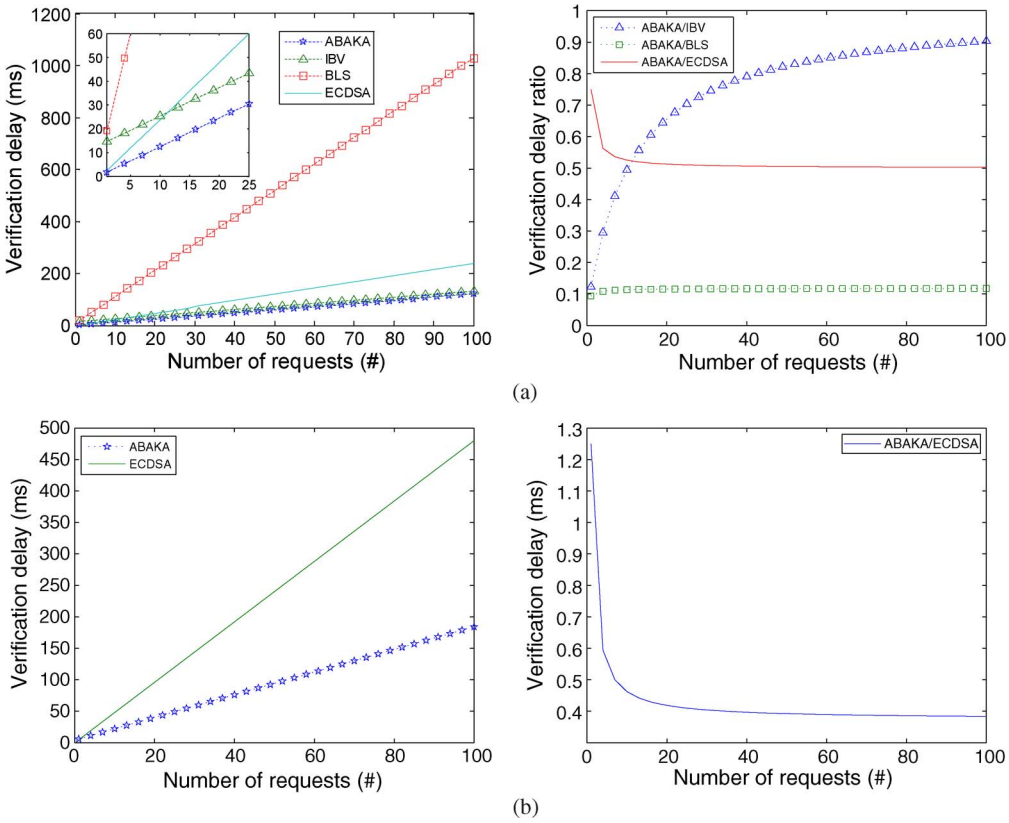

Fig. 7. Verification delay versus number of requests. (a) One-way authentication case. (b) Mutual authentication case.

TABLE III
COMPARISONS OF TRANSMISSION OVERHEAD (IN BYTES)

	Send a single request		Send n requests	
	V to SP	SP to V	V to SP	SP to V
ABAKA	84 bytes	80 bytes	$84n$ bytes	$N_{RSU} \times 80$ bytes
IBV	63 bytes	N./A.	$63n$ bytes	N./A.
BLS	146 bytes	N./A.	$146n$ bytes	N./A.
ECDSA-AKA	167 bytes	167 bytes	$167n$ bytes	$167n$ bytes

N_{RSU} is the number of roadside units (RSUs) where the requests sent from.

increases. For clarity, we zoom in the number of request ranging from 0 to 25 in the embedded small figure. Furthermore, we also show the ratio of the verification delay for comparison. Fig. 7(b) focuses on the mutual authentication case to compare the verification delay of ABAKA and ECDSA-AKA. From Fig. 7(a), we can observe that ABAKA holds significant advantages, compared with the other schemes. The ratio of verification delay shows that ABAKA is almost constantly 89% faster than BLS. ABAKA is 48% faster than ECDSA when the number of requests is larger than 10. It is worth mentioning that IBV can verify a batch of numerous messages almost as fast as ABAKA since the verification delays of both schemes have fewer relationships with the number of requests.

B. Transmission Overhead

In this section, we analyze the transmission overhead of ABAKA, compared with that of the IBV, BLS, and ECDSA-AKA schemes. The transmission overhead consists of two aspects: the transmission overhead incurred by delivering the packet from requesting vehicles to the SP (V to SP) and from the SP to the requesting vehicles (SP to V). Table III lists the total transmission overhead of all schemes in terms of sending a single request and n requests. The packet size of ABAKA, as defined in Figs. 3 and 4, costs 84 and 80 bytes, respectively. Note that ABAKA utilizes the advantage of broadcasting to design the response message.[7] Therefore, the transmission overhead from SP to V in ABAKA can be much lower than the traditional scheme. The packet of IBV consists of a 21-byte signature and a 42-byte pseudoidentity. The packets of BLS and ECDSA comprise a signature and a 125-byte certificate, but BLS adopted a short signature, cutting down the signature size from 42 to 21 bytes. In Fig. 8, we also discuss the transmission overhead of the ABAKA and ECDSA-AKA schemes [8] while a number of requests pour in. From Fig. 8, we can see that the ratio of the transmission overhead sharply drops from 138% to 31% when the number of requests is more than 10. More

[7]According to [1] and [7], there could be roughly 180 vehicles in the communication range of an RSU in a high-density traffic scenario. It is highly possible that several requests from the same RSU can be verified in the same batch. Thus, we assume $N_{\mathrm{RSU}} = 5$, where N_{RSU} is the number of RSUs, which are where the requests were sent from, in the succeeding analysis.

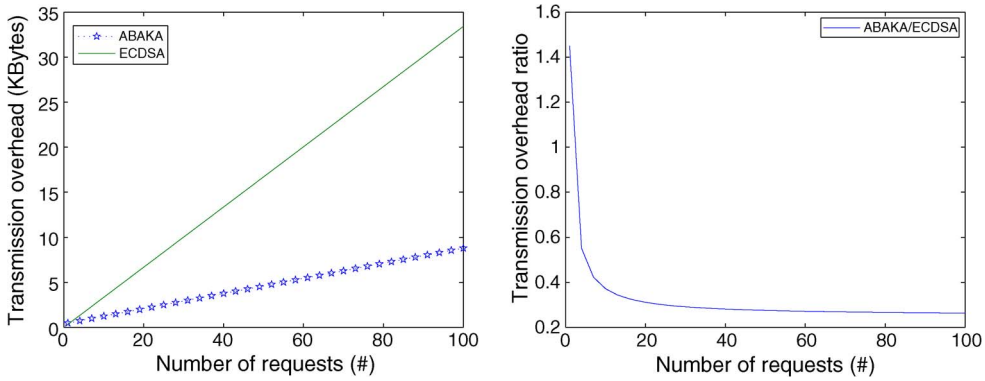

Fig. 8. Transmission overhead versus the number of requests (in mutual authentication case).

TABLE IV
VERIFICATION COST FOR REBATCH VERIFICATION

	First batch verification	Rebatch verification
ABAKA	$(2n+1)T_{mul}$	$1T_{mul}$
IBV	$3T_{par}$ + nT_{mtp}+ nT_{mul}	$3T_{par}$
BLS	$(2n+2)T_{par}$ + $2nT_{mtp}$	$(n+1)T_{par}$

precisely, as long as the number of requests is more than 13, the transmission overhead of ABAKA is 68.9% lower than that of ECDSA-AKA.

C. Verification Cost for Rebatch Verifications

According to the reliability analysis described in Section III-D1, the probability of only one invalid request in a batch is the most significant. Here, we further analyze the cost for a vehicle if rebatch verifications are required. We elaborately analyze the verification delay for a vehicle in the three batch-based schemes ABAKA, IBV, and BLS in Table IV. In ABAKA, the SP has to calculate requesting vehicles' corresponding private key CRK_i and t_iCRK_i to derive random point R_i in the first batch verification, which takes 2 T_{mul}. However, in rebatch verification, the SP only spends 1 T_{mul} for verification.[8] In IBV and BLS, some operations can be omitted in the rebatch verification. In the following analysis, we assume that the number of requests in a batch is 100. To be precise, we provide the verification cost of the worst case and average case. Although the ECDSA scheme is not required to perform the rebatch verification, we also show the verification cost of ECDSA in two cases to examine the value of the batch-based schemes.

1) Worst case: According to the proposed detection algorithm in Section III-D2, the worst case means that a valid request is always with the invalid request in the same batch until the last batch division. A batch of requests can be divided at most $\lceil \log_2 n \rceil$ times, where n is the number of requests in a batch. Let $T_{\text{first\_ver.}}$ denote the time to perform the verification for the first time and $T_{\text{rebatch\_ver.}}$ denote the time to perform the verification in a rebatch verification. As a result, the total verification delay for a valid request in the worst case is

$$T_{\text{worst}} = 1 \times T_{\text{first\_ver.}} + 2 \times \lceil \log_2 n \rceil \times T_{\text{rebatch\_ver.}}$$

2) Average case: The average case is the total verification delay over all possible cases divided by the number of possible cases. Then, the total verification delay for a valid request in the average case is

$$T_{\text{Avg}} = 1 \times T_{\text{first\_ver.}} + \frac{1}{\lceil \log_2 n \rceil + 1} \sum_{i=1}^{\lceil log_2 n \rceil} (T_{\text{first\_ver.}} + 2 \times T_{\text{rebatch\_ver.}}).$$

Fig. 9(a) shows the verification delay for rebatch verifications in the worst case while the number of requests in a batch is changed from one to 100 requests. Along with the verification delay for rebatch verifications, the comparison of the ratio of verification delay for rebatch verifications is also represented. In addition, the average case is demonstrated in Fig. 9(b). From Fig. 9, we can observe that ABAKA outperforms the other schemes (even ECDSA). Note that ECDSA is not batch-based verification without additional rebatch verification delay. In Fig. 9(a), ABAKA is almost constantly faster than BLS by 94%, outperforms IBV by at least 60%, and gains about 41% faster than ECDSA, whereas the number of requests is more than 28. As compared with BLS and ECDSA, ABAKA enjoys more advantages while more requests are issued. In the average case, the advantage of ABAKA over ECDSA is more significant. It is worth mentioning that IBV also enjoys the advantages (faster than ECDSA) in the average case while the number of requests is more than 94. The reason is that both ABAKA and IBV take the constant time to verify a batch of requests, but ECDSA and BLS do not. Moreover, to measure the effectiveness of the batch-based schemes, we compare the ratio of the verification delay of the three batch-based schemes with the ECDSA scheme in Fig. 10. Compared with ECDSA, only ABAKA can perform better than ECDSA no matter how many numbers of requests appear. When the number of requests is up to 90, IBV outperforms ECDSA. Unfortunately, BLR did not have the performance advantage. As a result, ABAKA should be more suitable than other batch-based schemes in VANETs.

[8]Similar to [7], we are only concerned with the cost of three dominant operations, i.e., T_{mul}, T_{par}, and T_{mtp}.

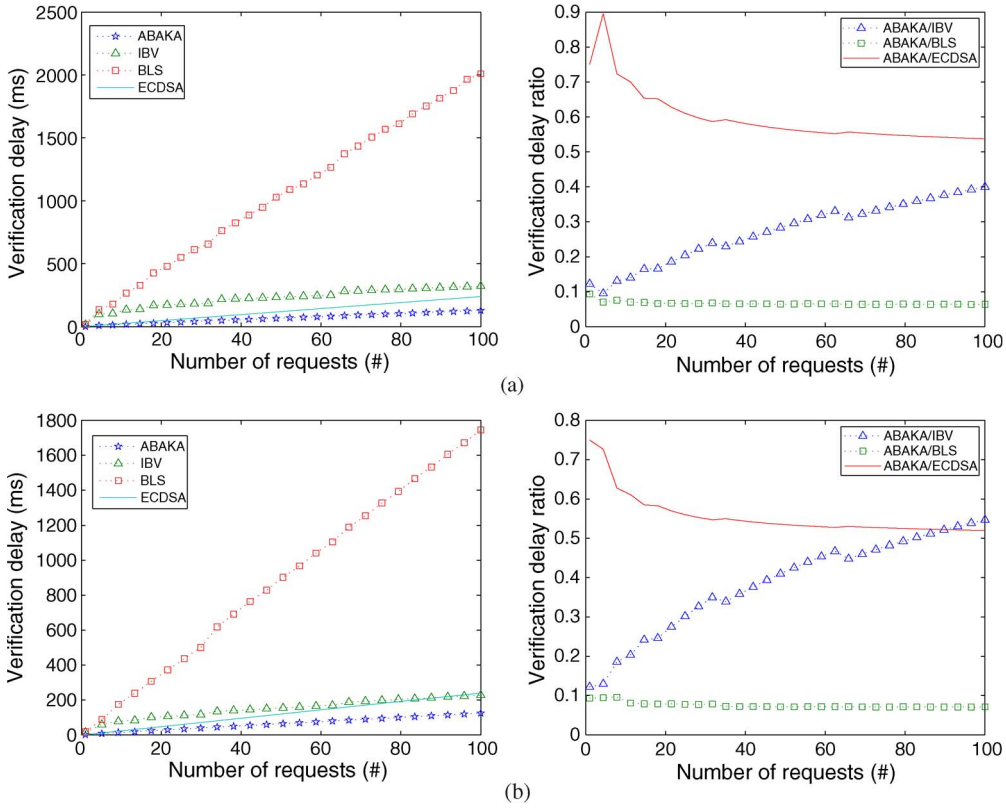

Fig. 9. Verification delay for rebatch verifications versus the number of requests. (a) Verification delay for rebatch verifications in the worst case. (b) Verification delay for rebatch verifications in the average case.

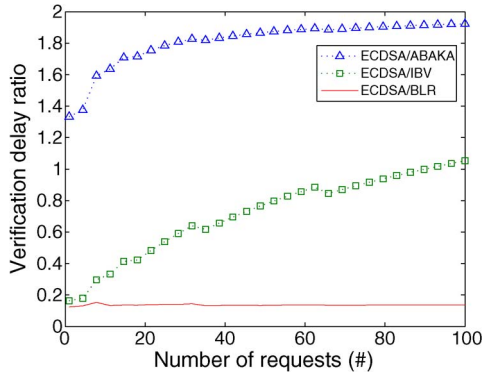

Fig. 10. Verification delay ratio compared with the ECDSA scheme versus the number of requests.

D. Expected Verification Delay

Based on the reliability analysis described in Section III-D1 and verification cost for rebatch verifications described in Section V-C, we further examine the expected verification delay composed of the original verification cost and the expected verification cost for rebatch verifications. Here, we consider the worst case of rebatch verifications. Let $T_{\text{orignal\_ver.}}$ denote the time to perform unidirectional verification in the proposed scheme with no invalid request, Pr_{rebatch} denote the probability of performing rebatch verification, and $T_{\text{rebatch\_cost}}$ represent the extra verification cost for rebatch verifications. Thus, the expected verification delay, which is denoted as T_{expected}, can be formulated as

$$T_{\text{expected}} = T_{\text{orignal\_ver.}} + Pr_{\text{rebatch}} \times T_{\text{rebatch\_cost}}.$$

In Fig. 11, the relationship between the expected verification delay and the different number of compromised vehicles is presented, whereas the number of requests in a batch is set to 100. With the results of the reliability analysis, we examine the two most possible cases; there are one $(i = 1)$ or two $(i = 2)$ invalid requests sent from compromised vehicles in a batch of requests. We can observe that the variation of the number of compromised vehicles only slightly affects the expected verification delay for ABAKA because the rebatch verification cost for ABAKA is relatively less than other batch-based verification schemes. Compared with ECDSA without the rebatch verification cost, ABAKA also keeps the superior expected verification delay. Note that the rebatch verification cost for the $(i = 2)$ case can be derived from Section V-C by assuming that the two invalid requests are separately distributed in the front part and the rear part of requests, which is the worst case. To sum up, it is anticipated that ABAKA could effectively ease the verification burden of SPs.

E. Simulation Evaluation

In this section, we adopt the ns-2 simulator [23] to properly estimate the real-world road environment and vehicular traffic. To genuinely generate the mobility of the real-world vehicles,

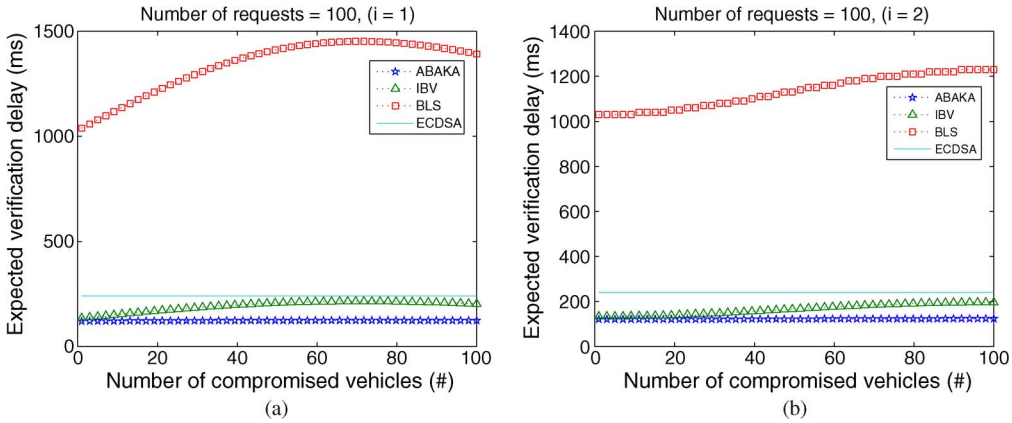

Fig. 11. Expected verification delay versus the number of compromised vehicles. (a) $i = 1$. (b) $i = 2$.

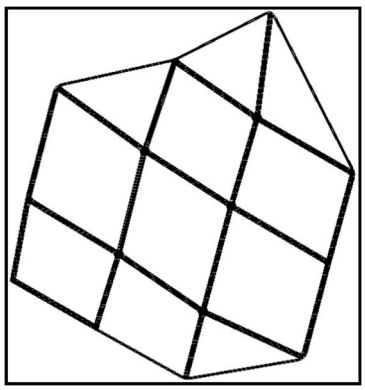

Fig. 12. City street map.

we use the mobility model generation tool named TraNS, which was introduced by [24]. TraNS can take advantage of the publicly available The Topologically Integrated Geographic Encoding and Referencing (TIGER) database from the U.S. Census Bureau, where the street maps of cities/towns in the United States are offered. Our simulation adopts the map shown in Fig. 12, which corresponds to a part of Manhattan Island in New York City. At first, vehicles are randomly scattered on the roads and then move toward randomly selected intersections in the map. Vehicles are driving along the roads with a random speed between 1 and 40 m/s. Road-speed limit is also implemented in every street. All possible cryptographic operations in the simulation are considered to have the same simulation delay. We assume that some 20% of the vehicles are requesting services, which is a value used to calculate the verification delay. In this simulation, we are interested in the performance of ABAKA and ECDSA-AKA since only the two schemes can provide mutual authentication and key agreement. All the simulation parameters are listed in Table V.

The average message delay (which is denoted as $avgD$) and average loss ratio (which is denoted as $avgLR$) are considered in this simulation and can be expressed as follows:

$$avgD = \frac{1}{N_A \cdot M^n_{\text{sent}} \cdot SP^n} \times \sum_{n\ inA} \sum_{m=1}^{M^n_{\text{sent}}} \sum_{s=1}^{SP^n} \left(T^{n\_m\_s}_{\text{trans}} + T^{n\_m\_s}_{v-auth} \cdot L^s + T^{n\_m\_s}_{\text{SP}-\text{auth}} \right)$$

TABLE V
SIMULATION CONFIGURATION

City simulation area	$1000m \times 1000m$
Communication range	250 m
Simulation time	100 s
Wireless Protocol	802.11a
Channel bandwidth	6 *Mb*s
Pause time	0 s
Packet size for ECDSA message	167 bytes
Packet size for ABAKA message	84 bytes

where A is the sample area in this simulation, N_A is the number of vehicles in A, M^n_{sent} is the number of request messages sent by vehicle n, and SP^n is the number of SPs where vehicle n has registered. For simplicity, we assume that $SP^n = 1$ in this simulation. $T^{n\_m\_s}_{\text{trans}}$ is the time that vehicle n transmits messages m to SP s; $T^{n\_m\_s}_{v-auth}$ is the time that SP s authenticates vehicle n, which is triggered by message m; and $T^{n\_m\_s}_{\text{SP}-\text{auth}}$ is the time that vehicle n authenticates SP s, which is triggered by message m. n_m_s represents the message m sent by vehicle n and received by SP s, and L^s is the length of the queue in SP s, i.e.,

$$avgLR = \frac{1}{N_A} \cdot \sum_{n=1}^{N_A} \frac{M^n_{\text{consumed}}}{\sum_{s=1}^{SP^n} M^n_{\text{arrived}}}$$

where M^n_{consumed} means the number of messages consumed by vehicle n in the application layer, and M^n_{arrived} represents the number of messages received by SP s in the application layer.

1) Impact of Vehicle Density: In the first set of simulations, we investigate the impact of vehicle density. Fig. 13 shows the simulation results on the average message delay and the average message loss rate. In general, the more vehicles that appear, the more advantages ABAKA holds. In Fig. 13(a), ABAKA outperforms ECDSA-AKA between 31% and 34%. As one can see, the curve tendency of message delay corresponds to the analytical results analyzed in Section V-A and Fig. 7(b). Note that the analytic results do not include the transmission delay. With regard to the message loss ratio, both ABAKA and ECDSA-AKA increase the message loss ratio while the number

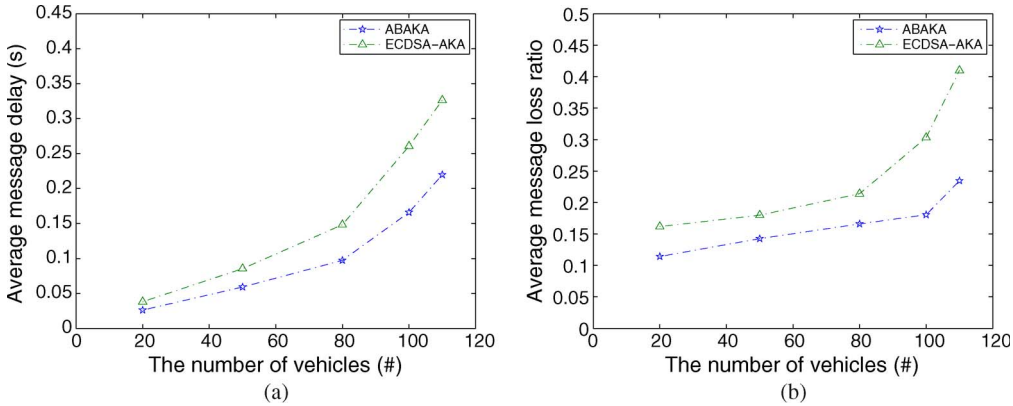

Fig. 13. Impact of vehicle density. (a) Average delay versus the number of vehicles. (b) Average loss rte versus the number of vehicles.

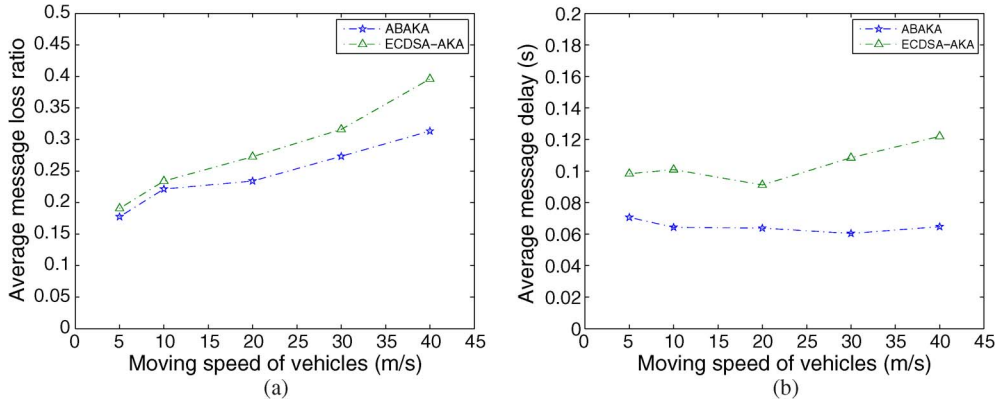

Fig. 14. Impact of vehicles' moving speed. (a) Average delay versus moving speed of vehicles. (b) Average loss rate versus moving speed of vehicles.

of vehicles soars. The increasing ratio of ABAKA is between 5% and 24%, and that of ECDSA-AKA is between 11% and 30%. As compared with ECDSA-AKA, ABAKA performs better in terms of message loss ratio, which has reached 38%, while the vehicle density is greater than 100.

2) Impact of Vehicle Moving Speed: In the second set of simulations, the average speed of the vehicles is changed from 5 to 40 m/s (36–144 km/h). In this simulation, we assume that the number of vehicles is 50. The simulation results on the average message delay and average message loss rate are shown in Fig. 14. As one can see, the average message delay of ABAKA is slightly affected by the speed of the vehicles. However, the average message delay of ECDSA-AKA is significantly increased as the speed is more than 20 m/s. It can be seen that ABAKA yields less message delay than ECDSA-AKA does at every speed. In terms of the message loss ratio, we can see that both schemes are significantly affected by the speed, particularly for ECDSA-AKA. The different speeds of vehicles will trigger different times for handoff procedures, which are the process of transferring an ongoing session from one RSU to another. The handoff procedures may incur a higher message loss ratio while data transmission is ongoing. From Fig. 14(b), it is obvious that ABAKA has a lower message loss ratio while the speed is up to 20 m/s because while the packet size of ABAKA is shorter than that of ECDSA-AKA, the period of packet transmission of ABAKA is also shorter than that of ECDSA-AKA. Therefore, we infer that the packet transmission of ECDSA-AKA has a higher probability of being interrupted while the vehicles are moving fast.

VI. Related Work

Recently, several related studies have been proposed, addressing the security and privacy preservation issues for safety-related applications in VANETs [3], [5]–[9]. In 2007, Raya and Hubaux [3] proposed a security scheme to achieve both message authentication and user anonymity. In Raya and Hubaux's scheme [3], each vehicle is preloaded with a large number of anonymous public and private key pairs, together with the corresponding public key certificates. The conventional public-key-based scheme is adopted. The short lift time of each public/private key pair is used to ensure privacy, and different pseudoidentities are used with corresponding public key certificates. However, the disadvantage of Raya and Hubaux's scheme is the need for a large storage capacity to store numerous public/private key pairs. In the same year, Lin *et al.* [9] devised a Group Signature and Identity-based Signature (GSIS) scheme based on the group signature scheme to sign each message. With no identity information included in the transmitted messages, their approach can keep the identities a secret. Not only the storage costs of public/private key pairs but the bandwidth consumption as well can be reduced. The cost of their scheme is

TABLE VI
FUNCTIONALITIES OF VANET SCHEMES

Scheme	Goal	Cryptographic primitive	Key agreement	Comp. & comm. cost
[3]	Safety-related message verification	Traditional PKI	No	High
[9]	Safety-related message verification	Group signature	No	Medium
[6]	Safety-related message verification	Localized group signature	Yes	Medium
[5]	Safety-related message verification	One-way hash function	No	Low
[7]	Safety-related message verification	Batch ID-based crypto.	No	Low
[8]	Safety-related message verification	Symmetric crypto. and ECDSA	Yes	Low
[4]	Vehicles and SPs key agreement	Blind signature and RSA	Yes	High

the overhead of maintaining the group-signature-based scheme. In 2008, Lu *et al.* [6] proposed the efficient conditional privacy preservation (ECPP) scheme, which divides privacy into three levels. First, an RSU issues anonymous certificates to vehicles, and only the RSU could realize which vehicle possesses the specific anonymous certificate. Then, the privacy of the identities of the vehicles is guaranteed by the anonymous certificates in IVC. Third, ECPP can allow a trusted authority (TA) to extract the real identity of a vehicle to achieve the requirement of the conditional privacy. Although those schemes have cleverly solved privacy issues in VANETs, unfortunately, they were not concerned with the scalability issues that may affect the performance of VANETs.

To deal with the scalability issues, Lin *et al.* [5] proposed a time-efficient and secure vehicular communications (TSVC) based on TESLA [26] to address the scalability issue. In TSVC, a vehicle first broadcasts a commitment of hash chain to its neighbors. By the use of the elements of the hash chain, the neighbors can authenticate this vehicle's following messages. Owing to the rapid verification of MAC, TSVC can greatly alleviate the message LR. However, the weakness of TSVC is not robust enough. The larger the dynamics of traffic becomes, the more Loss Ratio TSVC has. Zhang *et al.* [7] also came up with an identity-based batch verification (IBV) scheme for vehicular sensor networks based on pairing-based cryptography. When verifying a batch of message signatures, the verification speed of IBV is much faster than that of other public-key infrastructure (PKI)-based schemes. Both conditional privacy and lower verification delays are accomplished. Nevertheless, it is clear that the verification speed of a pairing is slower than that of a multiplication operation [8], and we found that IBV may suffer from replay attacks. Moreover, Zhang *et al.* [8] offered an RSU-aided message authentication scheme for vehicular communications named RAISE. Instead of PKI-based message signatures to improve the efficiency and scalability for IVC, RAISE uses a symmetric key to generate the symmetric MAC for message verifications. The expense of RAISE is that each RSU needs to maintain two tables, i.e., the ID-key table and trace evidence table, for message verification and traceability. To achieve the goal of traceability, the number of records in the trace evidence table may become huge. It is obvious that we have to carefully strike a balance between the scalability and maintaining cost.

Until now, a few studies have been devoted to developing the security mechanisms for value-added applications in VANETs. Li *et al.* [4] proposed a secure and efficient communication scheme with a privacy preservation (SECSPP) scheme to support the value-added applications. In SECSPP, a vehicle needs to acquire a blind signature for privacy preservation before the vehicle accesses the desired services from the near RSU. An SP is responsible for verifying the validity of signatures. However, these studies did not take the scalability into consideration, which may incur a bottleneck problem in the SP once a large number of requests flows out. For clarity, the functionalities of the aforementioned schemes are summarized in Table VI.

VII. CONCLUSION

We have proposed a novel ABAKA scheme for value-added services in VANETs. With ABAKA, an SP can simultaneously authenticate multiple requests and establish different session keys with vehicles. ABAKA considers not only scalability and security issues but privacy preservation as well. To deal with the invalid request problem, a detection algorithm has also been proposed. In the analytical analysis, we have elaborately evaluated ABAKA with current standard ECDSA schemes and other batch-based schemes in terms of verification delay and transmission overhead, as well as the verification cost for rebatch verifications. Moreover, the efficiency and practicality to the real-world applications have been verified by the simulation analysis. To sum up, ABAKA is a suitable scheme for value-added services in VANETs. In the future, we will further take the features of VANETs, such as the mobility model and predicable routing, to design novel schemes to gain more efficiency.

REFERENCES

[1] *Dedicated Short Range Communications (DSRC)*. [Online]. Available: http://grouper.ieee.org/groups/scc32/dsrc/index.html
[2] S. Lee, G. Pan, J. Park, M. Gerla, and S. Lu, "Secure incentives for commercial ad dissemination in vehicular networks," in *Proc. ACM Int. Symp. MobiHoc*, 2007, pp. 150–159.
[3] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *J. Comput. Secur.*, vol. 15, no. 1, pp. 39–68, Jan. 2007.
[4] C.-T. Li, M.-S. Hwang, and Y.-P. Chu, "A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks," *Comput. Commun.*, vol. 31, no. 12, pp. 2803–2814, Jul. 2008.
[5] X. Lin, X. Sun, X. Wang, C. Zhang, P.-H. Ho, and X. Shen, "TSVC: Timed efficient and secure vehicular communications with privacy preserving," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4987–4998, Dec. 2008.
[6] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. IEEE INFOCOM*, 2008, pp. 1229–1237.
[7] C. Zhang, R. Lu, X. Lin, P. H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE INFOCOM*, 2008, pp. 816–824.
[8] C. Zhang, X. Lin, R. Lu, P. H. Ho, and X. Shen, "An efficient message authentication scheme for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 57, no. 6, pp. 3357–3368, Nov. 2008.

[9] X. Lin, X. Sun, P. H. Ho, and X. Shen, "GSIS: A secure and privacy-preserving protocol for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3442–3456, Nov. 2007.
[10] M. Raya, P. Papadimitratos, and J. P. Hubaux, "Securing vehicular communications," *IEEE Wireless Commun. Mag.—Special Issue on Inter-Vehicular Communications*, vol. 13, no. 5, pp. 8–15, Oct. 2006.
[11] *IEEE Trial-Use Standard for Wireless Access in Vehicular Environment-Security Services for Applications and Management Messages*, IEEE Std. 1609.2-2006, Jul. 2006.
[12] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," *IEEE Commun. Mag.*, vol. 32, no. 9, pp. 33–38, Sep. 1994.
[13] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Berlin, Germany: Springer-Verlag, 2004, ser. LNCS.
[14] F. Li, X. Xin, and Y. Hu, "Identity-based broadcast signcryption," *Comput. Standard Interfaces*, vol. 30, no. 1/2, pp. 89–94, Jan. 2008.
[15] J.-H. Yang and C.-C. Chang, "An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Comput. Secur.*, vol. 28, no. 3/4, pp. 138–143, May/Jun. 2009.
[16] *Standards for Efficient Cryptography (sec 1)*. [Online]. Available: http://www.secg.org/download/aid-780/sec1-v2.pdf
[17] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction," *IEICE Trans. Fundam.*, vol. E84-A, no. 5, pp. 1234–1239, 2001.
[18] C.-L. Hsu, "Security of Chien *et al.*'s remote user authentication scheme using smart cards," *Comput. Standard Interfaces*, vol. 26, no. 3, pp. 167–169, May 2004.
[19] J. Camenisch, S. Hohenberger, and M. O. Pedersen, "Batch verification of short signatures," in *Proc. EUROCRYPT*, 2007, pp. 246–263.
[20] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, U.K.: MIT Press, 2001.
[21] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. EUROCRYPT*, 2003, pp. 416–432.
[22] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," in *Proc. Asiacrypt*, 2001, vol. 2248, pp. 514–532.
[23] *The Network Simulator-ns-2*. [Online]. Available: http://nsnam.isi.edu/nsnam/index.php
[24] M. Piorkowski, M. Raya, A. L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "TraNS: Realistic joint traffic and network simulator for VANETs," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 12, no. 1, pp. 31–33, Jan. 2008.
[25] M. Scott, *Efficient Implementation of Cryptographic Pairings*. [Online]. Available: http://ecrypt-ss07.rhul.ac.uk/slides/thursday/mscottsamos07.pdf
[26] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA broadcast authentication protocol," *RSA Crypto.*, vol. 5, no. 2, pp. 2–13, 2002.

Jiun-Long Huang received the B.S. and M.S. degrees from National Chiao Tung University, Hsinchu, Taiwan, in 1997 and 1999, respectively, and the Ph.D. degree from National Taiwan University, Taipei, Taiwan, in 2003.

He is currently an Assistant Professor with the Department of Computer Science, National Chiao Tung University. His research interests include mobile computing, mobile data management, wireless networks, and Internet technology.

Lo-Yao Yeh received the B.S. degree in information management from Da Yeh University, Changhua, Taiwan, in 2003 and the M.S. degree from National Chi Nan University, Nantou, Taiwan, in 2005. He is currently working toward the Ph.D. degree with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan.

He was a Visiting Scholar with the University of California, Berkeley. His current research interests include network security, overlay network security, and sensor networks.

Hung-Yu Chien received the B.S. degree in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 1988, the M.S. degree in computer and information engineering from National Taiwan University, Taipei, Taiwan, in 1990, and the Ph.D. degree in applied mathematics from National Chung Hsing University, Taichung, Taiwan, in 2002.

During 1992–1995, he was an Assistant Researcher with TL, MOTC, Taiwan. He has also been the Director of the Computer Center, Nan-Kei College. From September 2003 to September 2006, he was an Associate Professor with ChaoYang University of Technology. He is currently a Professor and Department Head with the Department of Information Management, National Chi Nan University, Nantou, Taiwan. His research interests include cryptography, networking, and network security.

Dr. Chien is a member of the Chinese Association for Information Security.