Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Secure authentication in motion: A novel online payment framework for drive-thru Internet

Jun Song, Fan Yang, Lizhe Wang*

School of Computer Science, China University of Geosciences, Wuhan, 430074, PR China

HIGHLIGHTS

- We propose a novel secure payment framework for the drive-thru Internet.
- We present an adaptive authentication scheme in online and offline scenarios.
- Utilizing a new certificateless public key scheme will derive a novel property.
- A traceable batch authentication will reduce the load of key computation and management.
- It provides a comprehensive evaluation to show the security and feasibility of the proposed scheme.

ARTICLE INFO

Article history:
Received 1 September 2015
Received in revised form
29 December 2015
Accepted 16 June 2016
Available online xxxx

Keywords:
Drive-thru Internet
Online payment
Auditing
Encrypted signature

ABSTRACT

The security and privacy issues have been well investigated in typical vehicle ad hoc networks. However, considering the drive-thru Internet properties, in particular for a secure and in-motion payment services case, merely implementing the existing online payment schemes may be either infeasible or inefficient. In this paper, we propose an advanced online payment framework, which integrates three main features, including the novel pairing-free certificateless encryption, signature and semi-honest RSU-aided verification, and the CA-aided tracking and batch auditing, and providing following properties independently, e.g., achieving a higher trust level and supporting primary security services, introducing a semi-honest RSU to indicate more practicality, and optimizing the verifying and auditing efficiency for a large number of authentication requests case. Performance evaluations such as security analysis, efficiency analysis, and simulation evaluation show the security and feasibility of the proposed framework.

1. Introduction

With the advancement of vehicular ad hoc networks (VANETs) in recent years, increasing numbers of researchers and engineers have developed much new concepts and innovative ideas into the intelligent transportation services, *e.g., Toyota Safety Sense, BroadR-Reach Automated Compliance*, and *Mercedes-Benz Companion*, emerging as a promising approach to ensure a high quality of life. The primary purpose of vehicular networks is to enable vehicular communication applications, such as increasing driving safety, efficiency, and convenience [1,2]. However, people might prefer to get Internet services via driving vehicles quickly and easily, and to fully experience the pleasures of activities, such as online shopping, downloading software, and uploading video or audio, the so-called drive-thru Internet [3]. To meet these demands, it really needs to provide not only a *large-scale high-quality* deployments of wireless infrastructures, i.e., a stable and reliable communication environment, but a set of effective security mechanisms to secure vehicular communication.

Over the past several years, there have been much research on achieving an efficient message authentication [4,5] or establishing a secure communication channel [6,7] in typical VANETs. However, from the security and privacy perspectives, vehicular networks have brought many new challenges owing to network congestion and performance degradation issues, particularly when vehicular nodes are on the status of *intermittent* or *short-lived* communication connectivity [8,9], such as in a typical drive-thru Internet scenario. More exactly, compared with general wireless ad hoc networks, the drive-thru Internet in nature not only is an improvement of network property and user quantity, but also has some new features involved, i.e., fast-moving nature, intermittent network connectivity, and high contention environments, etc. In addition, it should be noted that, when a large number of fast-moving vehicle nodes compete for communication simultaneously, they

* Corresponding author.
E-mail address: lizhe.wang@gmail.com (L. Wang).

http://dx.doi.org/10.1016/j.future.2016.06.011
0167-739X/© 2016 Elsevier B.V. All rights reserved.

may have fallen into to a kind of *volatile* or *vulnerable* communications environment [8]. From the point of view of authentication protocol, if the access requests from vehicle nodes cannot submit to an authentication server or a secure gateway multiple times, those web-based secure services may be inefficient in such a case.

Although the security and privacy issues have been well investigated in typical vehicular ad hoc networks scenarios [1,2,10], most of them tend to focus on network environment with properties of good stability and high reliability. Generally, vehicular applications need security assurance to authenticate entities and trustworthy information exchange via an insecure network. Similar to the previous work [4,11,12], both authentication and identification are the fundamental mechanisms in securing VANETs. For the case of online payment over drive-thru Internet, it is essential to address higher security demands and goals for electronic transactions. Due to its in-motion payment nature, more security properties, i.e., confidentiality, integrity, authenticity, and non-repudiation, should be the most essential security concerns that must be provided. Considering these properties, existing online payment solutions, i.e., proposed for general static wireless networks, may be infeasible for the drive-thru Internet scenario.

To address above concerns, in this paper, we propose an advanced secure and efficient online payment framework especially for a drive-thru Internet applications. The contributions of this paper are threefold. First, inspired by *Lite-CA-based* public key cryptosystem [13], we propose a new pairing-free certificateless encryption scheme, which is not only to reduce the certificate management complexity but to achieve a higher trust level as well, *e.g.*, to achieve an explicit authentication property. Second, based on the proposed encryption and signature scheme, we introduce an RSU-aided online verification process, especially considering a more practical security property, i.e., the semi-honest RSUs, and thus appropriate for the secure online payment applications in drive-thru Internet case. Third, with the purpose to enhance the security of proposed framework, we present a CA-aided tracking and batch auditing scheme to improve the verifying and auditing efficiency in such a case, *e.g*, a large number of authentication requests. Besides that, a comprehensive performance based on drive-thru Internet scenario, including security analysis, efficiency analysis, and simulation and numerical analysis, is presented to show the security and feasibility of the proposed framework.

The rest of this paper is organized as follows. Section 2 presents the background and the related work, and overviews the related cryptographic requirements. Section 3 introduces the system model and security goals. Then, we present a formal definition and design of the proposed verifiably encrypted signature scheme without pairing in Section 4. Section 5 describes the detailed description of the secure online payment framework, including the system setup and different algorithms involved. Security analysis and performance evaluation are presented in Sections 6 and 7, respectively. Section 8 concludes the paper.

2. Background and related work

2.1. Related work

Security and privacy are always hot topics in VANETs [1,2,10]. [11] investigated the methods of providing security services and preserving privacy in VANET, especially to address two fundamental issues, *e.g.*, certificate revocation and conditional privacy preservation. [14] presented an RSU-aided messages authentication scheme to meet the needs of the messages authentication. In this scheme, RSU is trustworthy and hard to be compromised. Recent work [15] introduced a batch authentication scheme by utilizing pairing-based computation, to achieve the verification of lots of messages. Besides that, a recent work [16], proposed a stored-value card to provide an added-value service of payment in VANET. This work focused on a specific wireless network scenario. None of these solutions provides the online payment in case of drive-thru Internet, particularly for scenario where a lot of fast-moving vehicle nodes compete for communication simultaneously. Table 1 shows a quantitative comparison between the other relevant schemes and our schemes in terms of functions and features.

For the case of authenticated key agreement frameworks, so far there are many different categories of public key cryptosystem (PKCs), such as CA-based PKC [17], identity-based PKC (IBC) [18–20], certificateless (CL)-PKC [21,22], and *lite*-CA based PKC [13], and so on. Generally, in CA-based PKC schemes exist a most common issue, that is, *the complexity of certificates management*. Previous solutions of VANETs mainly adopt the ID-based authenticated key exchange (AKE) scheme [18,19,23], which reduces the management workloads of public key certificates compared with the CA-based PKC scheme. However, its *key escrow problem* still exists. [21] first introduces a certificateless (CL) public key cryptosystem, which usually exists *the impersonation attack issue* [13]. In addition, [13] introduced a *Lite-CA-Based* PKC scheme, which can be viewed as a *variations* originated from certificateless PKC (CL-PKC). This scheme achieves the highest trust level, i.e., detecting the impersonation attack easily, and provides efficient public key certificate management.

In this paper, we also propose a new certificateless public key encryption scheme, including two main properties: to resist the impersonation attack by introducing an explicit authentication mechanism, and to relieve the certificates management difficulty by utilizing a CL-PKC method as well. According to the existing results from [13], we discuss features and functions of various public key cryptosystems, and further show much more advantages of our proposed scheme, as shown in Table 2.

2.2. Security threats

The possible threats for an online payment framework include:

- **Message forging/cheating**: An adversary can send fake messages so as to either cheat on its identity, *e.g.*, Sybil attack, or disperse fake information, *e.g.*, forged payment receipts.
- **Message tampering**: An adversary may tamper the received messages and broadcast them to other nodes.
- **Message dropping**: An adversary may drop the messages to conduct a black-hole attack.
- **message congestion**: An adversary sends irrelevant bulk messages to take up the communication channel or to consume the service resources.
- **Message detour attack**: An adversary may take indirect or detour paths intentionally to increase services cost.
- **Message replay attack**: An adversary replays the expired messages in order to disturb the network.

3. System model and design goals

In this section, we present the system model and security goals towards an advanced online payment framework for drive-thru Internet.

3.1. System architecture

As shown in Fig. 1, the proposed secure online payment framework under consideration consists of five network entities: a root *certificate authority* (CA), a few of stationary *roadside units*

Please cite this article in press as: J. Song, et al., Secure authentication in motion: A novel online payment framework for drive-thru Internet, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.06.011

Table 1
Comparison of functions and features via various schemes.

Scheme	Functions	Features
Lin (2008) [11]	Cert. revocation and privacy	RSU is trustworthy
Zhang (2008) [14]	RSU msg. authentication	RSU is trustworthy
Lee (2013) [15]	Batch msg. authentication	Pairing-based computation
Chen (2013) [16]	Added-value payment service	Typical wireless network
Our scheme	Batch msg. authentication	Pairing-free and semi-honest RSU

Table 2
Comparison of various PKC schemes.

Scheme	Features	Mode	Level	Key gen.
CA-based PKC [17]	Complicated cert. management	Explicit	III	U/U
ID-based PKC [18,19,23]	Key escrow problem	Implicit	I	A/U
CL-PKC [21,22]	Impersonation attack	Implicit	II	U&A/U
Lite-CA-based PKC [13]	Efficient and robust	Explicit	III	U/U&A
Our CL-PKC	Efficient and robust	Explicit	III	U&A/U&A

Mode: Authentication mode.
Level: Trust level [13].
Key gen.: Private key generation/Public key generation.
A: Authority U: User U&A: User and Authority.

(RSUs), a number of moving vehicles equipped with *on-board units* (OBUs), an *Internet services provider* (ISP), and a *secure payment gateway* (SPG).

- **CA**: Generally, we can regard CA as an integration of the top-level authority (TA) and the key management center (KMC). CA is in charge of the registration, issuing and verifying certificates. Here it is noted that the proposed framework uses a novel certificateless scheme to avoid intricate certification management and key escrow problems. Besides these, CA can provide the features of tracking suspicious vehicles and batch auditing signatures in our proposed framework. Furthermore, we can assume that a CA should be fully trusted by all entities and be practically secure against adversaries.
- **RSU**: In this work, one RSU is connected by wired links to other RSUs, ISP, SPG, and CA; meanwhile, it has a wireless access point (AP) for all OBUs in a special communication range. Generally, RSUs are trusted in most of existing VANET solutions. However, in our study, an RSU is assumed to be one semi-honest entity which can be compromised by adversaries. It is indeed necessary in reality for a secure payment system to meet the higher security requirements.
- **OBU**: A vehicle equipped with OBU can communicate with other vehicles or with RSUs in VANET. Those pseudonymous certificates issued by CA or self-generated are installed in OBUs to provide the security services. In our framework, a vehicle node should be viewed as a dishonest entity due to the security consideration. In addition, we assume that all transmitted messages are divided into two types, the ordinary messages, i.e., local traffic information, and the confidential messages, i.e., bills, receipts, and accounts, etc. Thus we use two types of encryption methods to protect them respectively.
- **ISP**: In our study, ISPs mainly refer to the profit-making commercial organizations or the privately-owned enterprises, specified to online agencies, which provide web services for e-commerce, including selling, negotiating, ordering, etc. Usually, we assume that the ISP is never to be trusted by all entities in a drive-thru Internet scenario. Therefore, in this proposed framework, we introduce two main security features from the third parties, i.e., the RSU-aid verification and the CA-based traceability, to prevent the possible commercial cheating.
- **SPG**: The SPGs serve as the secure payment gateway to provide the online financial services. SPG has the responsibility to protect the payment details and, if needed, to support the verification of the bills of payment from customers or merchants. In addition, we assume that SPGs are fully trusted by all entities as well.

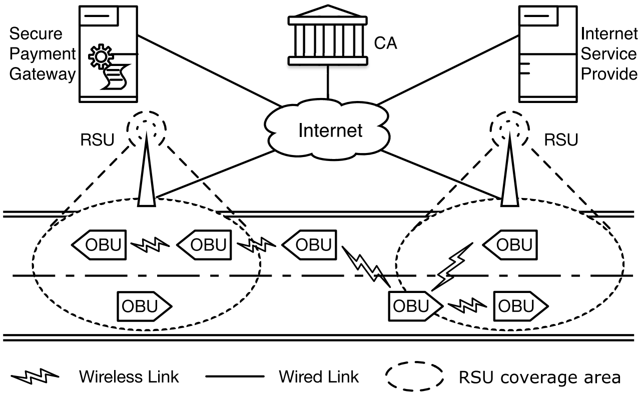

Fig. 1. An online payment system model.

In our study, we propose an online payment framework towards the properties of the secure and efficient authentication via a drive-thru Internet scenario. As shown in Fig. 1, this framework is tailored to applications in a dynamic and insecure communication environment. First of all, it is obvious that vehicles can communicate with the ISP or SPG only via RSUs, sending and receiving the proofs of online payment, i.e., Message Encryption and Signature (Alg. 1). In our study, these proofs are mainly represented by the encrypted messages and their signatures. Second, RSU can provide primary online signature verification on encrypted messages. If the verification succeed, the RSU will forward the message to ISP or SPG; otherwise, it will submit the verified proofs and the tracking request to CA, i.e., RSU-Aided Online Message Verification (Alg. 2). Noted that, RSU cannot decrypt the encrypted messages due to its semi-honest nature, whereas the receivers, *e.g.*, ISP or SPG, can decrypt the encrypted messages and obtain the proofs needed for further payment services, i.e., Receiver Message Decryption (Alg. 3). On the other hand, if CA receives the tracking request from RSU, CA will conduct the profound signature verification on both messages from vehicular nodes and the verified proofs from RSUs. In such scenario, there are two possible results which need to be further considered, including RSU compromised or message verification failed, i.e., CA Traceability (Alg. 4). Finally, in order to improve the auditing efficiency, a CA-aided batch auditing algorithm is proposed to achieve the batch verifications, i.e., CA-Aided Batch Message Auditing (Alg. 5). In our study, to simplify the design and analysis, SPG is treated as a part of the CA,

Please cite this article in press as: J. Song, et al., Secure authentication in motion: A novel online payment framework for drive-thru Internet, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.06.011

which has intrinsic feature to achieve the securing messages transmission, i.e., confidentiality, integrity, and authenticity. In addition, it deserves noting that a typical process of gateway-enabled transaction often contains a lot of other details. Since they are less related to our proposed secure payment framework, we do not have discussed them in this work.

3.2. Security goals in drive-thru internet

The general security properties that the proposed framework can provide will be introduced as follows:

- **Authentication**: Any user needs to verify the validity of his identity in order to securely access the authorized services.
- **Authorization**: A user can do what he want to do only when his authentication request is approved.
- **Integrity**: One user should be able to detect any unintended message changes or data corruption caused by diverse factors.
- **Confidentiality**: The secret data or information is only visible to legitimate users.
- **Non-repudiation**: It ensures that no user can deny its past behaviors, i.e., sending a message, or refuse the validity of a message.
- **Anonymity**: Any message should not be linked to the real identity of a sender, so as to provide further privacy-preserving property.
- **Traceability**: The authority, i.e., CA, should be able to verify the behaviors of a user by means of recorded identification.

Other properties close related to the secure payment include certificates revocation and key updating. In case of an online payment scenario, owing to less requirements involved with the message authentication process, we do not further discuss them in this work. However, we believe that these properties can also be achieved in the proposed framework [13].

4. A novel pairing-free certificateless encryption scheme

In this section, we propose a novel certificateless encryption scheme (NCL-PKC), inspired by the *Lite-CA-Based* PKC [13] and the CL-Based PKC with pairing [22,24]. We first describe the definitions and the security model for the proposed encryption scheme. In addition, we present proofs on consistency, confidentiality, and security. Finally, we design a basic NCL-PKC encryption scheme based on the discrete-log problem of finite-field cryptography, or rather an *ElGamal* public key encryption scheme.

4.1. Definitions and framework design

Definition 1. A NCL-PKC encryption scheme is an eight-array tuple $\Pi = (\mathcal{G}_{\mathcal{CA}}, \mathcal{G}_{\mathcal{U}}, \mathcal{E}_{\mathcal{S}}, \mathcal{E}_{\mathcal{P}}, \mathcal{S}_{\mathcal{S}}, \mathcal{S}_{\mathcal{P}}, \mathcal{E}, \mathcal{D})$ defined as follows:

(1) `CA-Setup`, $\mathcal{G}_{\mathcal{CA}}$, is a probabilistic polynomial time (*ppt*) algorithm that takes the system's security parameter 1^k as input, and outputs the master public/private keys pair (pk_{CA}, sk_{CA}). This algorithm is run by a *CA*.

(2) `User-Setup`, $\mathcal{G}_{\mathcal{U}}$, is also a *ppt* algorithm that user takes 1^k as input and outputs the public/private keys pair (pk_U, sk_U). This algorithm is run by a user.

(3) `Extract-Partial-Private-Key`, $\mathcal{E}_{\mathcal{S}}$, is also a *ppt* algorithm that takes 1^k, sk_{CA}, sk_U, and the user's identity $ID_U \in \{0, 1\}^*$ as input, and outputs partial private key s^2k_U. This algorithm is run by the *CA* for each user.

(4) `Extract-Partial-Public-Key`, $\mathcal{E}_{\mathcal{P}}$, is also a *ppt* algorithm that takes 1^k, sk_{CA}, pk_U and ID_U as input, and outputs partial public key p^2k_U. This algorithm is run by a *CA* for each user.

(5) `Set-Private-Key`, $\mathcal{S}_{\mathcal{S}}$, is a deterministic algorithm that takes 1^k, pk_{CA}, sk_U, and y_{ID_U} as input, and outputs (sk_U, s^2k_U) as the user's final private key only if s^2k_U is a valid partial private key. This algorithm is run by the user.

(6) `Set-Public-Key`, $\mathcal{S}_{\mathcal{P}}$, is a deterministic algorithm that takes 1^k, pk_{CA}, pk_U, and y_{ID_U} as input, and outputs (pk_U, p^2k_U) as the user's final public key only if p^2k_U is a valid partial public key. This algorithm is run by the user.

(7) `Encrypt`, $\mathcal{E}$, is also a *ppt* algorithm that takes a plaintext $M \in \mathcal{M}$, pk_U, p^2k_U, and pk_{CA} as input and outputs a ciphertext $C \in \mathcal{C}$ or $\perp$, which means that pk_U or p^2k_U is invalid. This algorithm is run by anyone who wants to send a ciphertext to the user, i.e., key holder.

(8) `Decrypt`, $\mathcal{D}$, is a deterministic algorithm that takes a ciphertext $C \in \mathcal{C}$, sk_U and s^2k_U as input and outputs the corresponding plaintext $M \in \mathcal{M}$ or $\perp$ which means that C is not a valid ciphertext. This algorithm is run by the user.

Different from the *Lite-CA-Based* PKC scheme [13], the definition of our NCL-PKC scheme contains a nine-array tuple and or four secrets for each user, instead of a six-array tuple or three secrets. We introduce three new algorithms, i.e., $\mathcal{E}_{\mathcal{S}}$ and $\mathcal{S}_{\mathcal{S}}$, to achieve two secrets: sk_U, and s^2k_U. With the help of such two secrets, this proposed NCL-PKC scheme achieves an integration of the *Lite-CA-Based* PKC scheme [13] and the CL-Based PKC scheme [22]. We call this new types of encryption scheme *A Pairing-free Certificateless Encryption Scheme*, that is, so-called *NCL-PKC* scheme. It is expected not only to take advantage of cryptographic properties from two original schemes but to derive a new cryptographic primitives. In this proposed scheme, we mainly introduce some new security properties that the typical CL-Based PKC scheme does not have, i.e., [22]. For instance, this proposed scheme has not any explicit certificates and centralized certificate management center. However, it supports the explicit certificates verification mechanism, free from key escrow problem, and the higher trust level, *e.g.*, *Level III-malicious KGC attack resilience* [13]. That is, the authority, *e.g.*, the malicious KGC, who knows nothing about the users' private key, and the frauds of the authority can be detected.

4.2. Proposed NCL-PKC encryption scheme

In this section, we provide the design detailed towards a NCL-PKC encryption scheme without pairing. This new encryption scheme consists of five algorithms, including `CA-Setup`, `User-Setup`, `Extract-Partial-Key`, `Set-Public-Key`, and `Set-Private-Key`. It is noted that, to avoid repeatedly descriptions, we omit two algorithms, i.e., `Encryption` and `Decryption`, where in the following Section 5 we will give a detailed introduction.

(1) `CA-Setup`: In this scheme, this algorithm is run by a TTP, usually refers to *CA* or *KGC*. First, This algorithm takes security parameter k as input and returns a large secure prime p, a prime divisor q, where $q \mid p - 1$. Second, *CA* chooses a cyclic group $\mathbb{G}$ of prime order q, where g be a generator of $\mathbb{G}$. There is a secure hash function $H_1 : \{0, 1\}^* \times \mathbb{G} \to \mathbb{Z}_q^*$ denoted here. *CA* picks a random number $x_{CA} \leftarrow \mathbb{Z}_q^*$ and compute $y_{CA} \leftarrow g^{x_{CA}} \bmod p$. The pair ($x_{CA}$, y_{CA}) will be used as *CA*'s a master private key and a master public key, respectively. Finally, the system parameters $\{p, q, g, y_{CA}, H_1\}$ are published.

(2) `User-Registration`: When a user (or vehicle node) A wants to join the system, A must register to *CA* first and obtain its public key. This algorithm consists of the following four subalgorithms:

Please cite this article in press as: J. Song, et al., Secure authentication in motion: A novel online payment framework for drive-thru Internet, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.06.011

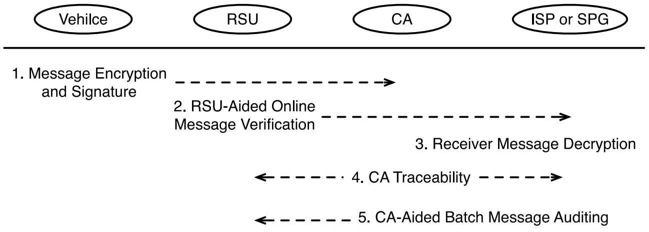
Vehilce
RSU
CA
ISP or SPG
1. Message Encryption and Signature
2. RSU-Aided Online Message Verification
3. Receiver Message Decryption
4. CA Traceability
5. CA-Aided Batch Message Auditing

Fig. 2. Protocol flow.

- (a) `User-Setup`: Supposing that the user A's real identity is ID_A. A picks a random number $x_A \leftarrow \mathbb{Z}_q^*$ and computes $y_A \leftarrow g^{x_A} \bmod p$, where x_A is the A's secret value and y_A is A's anonymous identity. The triplet $\langle ID_A, y_A, H_1(ID_A \parallel x_A)\rangle$ is sent to *CA* for registration.
- (b) `Extract-Partial-Key`: After receiving the triplet from A, *CA* first verifies the validity of ID_A, *e.g., ID card or driving license*, and then picks a random number $\alpha_A \leftarrow \mathbb{Z}_q^*$, computes $\mu_A \leftarrow g^{\alpha_A} \bmod p$, $h_A \leftarrow H_1(y_A \parallel \mu_A)$, $\sigma_A \leftarrow \alpha_A + h_A\, x_{CA} \bmod q$, and a ciphertext $\nu_A \leftarrow \sigma_A \oplus H_1(y_A \parallel {y_A}^{\sigma_A}) \bmod q$. As a result, *CA* sets a partial public key $p^2k_A \leftarrow \mu_A$ and a partial private key $s^2k_A \leftarrow \sigma_A$ for user A; and sends the triplet $\langle \nu_A, p^2k_A, s^2k_A\rangle$ to A via the secret channel.
- (c) `Set-Public-Key`: After receiving p^2k_A from *CA*, A first decrypts ν_A and then extracts σ_A, that is, $h_A \leftarrow H_1(y_A \parallel \mu_A)$, $\sigma_A \leftarrow \nu_A \oplus H_1(y_A \parallel {\mu_A}^{x_A}\, {y_{CA}}^{x_A h_A})$. Then, A enables $p^2k_A \leftarrow \mu_A$ as its partial public key and sets $pk_A \leftarrow y_A$ as its public key, respectively. Finally, A publishes the pair $\langle pk_A, p^2k_A\rangle$ as its final public key.
- (d) `Set-Private-Key`: After receiving s^2k_A from *CA*, A validates it by checking whether ${y_A}^{\sigma_A} \equiv {\mu_A}^{x_A} {y_{CA}}^{x_A H_1(y_A \parallel \mu_A)}$ holds. If not, A sends a "Complaint Message" against *CA*; otherwise, A sets a private key $sk_A \leftarrow x_A$ as well as a partial private key $s^2k_A \leftarrow \sigma_A$. Finally, A stores $\langle sk_A, s^2k_A\rangle$ as its final private key.

5. Secure online payment framework design

This section details our proposed framework design towards a novel secure online payment framework, especially in a drive-thru Internet scenario. This framework is based on our proposed new NCL-PKC scheme, and is further tailored to securing payment applications in an in-motion vehicular communication environment. For instance, a large number of in-motion vehicles compete for the communication channel so as to achieve message encryption and secure authentication with ISP or SPG. Under such a scenario, any one-hop authentication follows the protocol flow shown in Fig. 2. Seven main algorithms are involved in the authentication, including Message Encryption and Signature (Alg. 1), RSU-Aided Online Message Verification (Alg. 2), Receiver Message Decryption (Alg. 3), CA Traceability (Alg. 4), CA-Aided Batch Message Auditing (Alg. 5).

5.1. Protocol setup

We first review part of the notations and theorems that are closely related to our proposed framework.

5.1.1. System initialization

The notations of our framework are listed in Table 3. Our design utilizes the NCL-PKC scheme as above mentioned. Other cryptographic notions are presented as follows.

Definition 2. $\mathbb{Z}_q \overset{def}{=} \{0, 1, \ldots, q-1\}$ is an additive group and $\mathbb{Z}_q^*$ is multiplicative group, where q is a prime integer; For the consistency of modular exponentiations, in the rest of paper, random variables are independently and uniformly chosen from $\mathbb{Z}_q$ or $\mathbb{Z}_q^*$, respectively.

In addition, CA chooses a cyclic group $\mathbb{G}$ of prime order q, where g be a generator of $\mathbb{G}$. Another secure hash function is chosen: $H_2 : \{0, 1\}^* \times \{0, 1\}^* \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$.

- (a) `Gap Diffie-Hellman (GDH) Signature` [25]: Let secret key $x \leftarrow {\mathbb{Z}_q}^*$, the public key $y = g^x$, given x and a message $M \in \{0.1\}^*$, compute $h = H(M)$, and the signature $\omega = h^x$, where $H : \{0, 1\}^{(1)} \rightarrow \mathbb{G}^*$; The verification need to compute $h = H(M)$, and verify that (g, y, h, ω) is a valid Diffie–Hellman tuple.
- (b) `Dual (exponential) Challenge-Response (DCR) signature` [26]: Let public keys $A = g^a$ and $X = g^x$, $B = g^b$ and $Y = g^y$. The DCR signature (DS) of A and B on message m_1, m_2 is a tuple of values X, Y, and $DS_{A,B}$, respectively. Here, the same signature can be exchanged to compute (and verify) as follows: $DS_{A,B}(m_1, m_2, X, Y) = g^{(x+da)(y+eb)} = (YB^e)^{x+da} = (XA^d)^{y+eb}$, where d and e denote $H(X, m_1)$ and $H(Y, m_2)$.
- (c) `Twin Diffie-Hellman (TDH) Trapdoor Theorems` [27]: Using the above notations, suppose $X_1 \in \mathbb{G}$, $r, s \in \mathbb{Z}_q^*$, and $X_2 := g^s/X_1^r$. $Y, \hat{Z_1}, \hat{Z_2}$ are random variables in $\mathbb{G}$ and defined as functions of X_1 and X_2. Then, (1) X_2 is uniformly distributed over $\mathbb{G}$; (2) X_1 and X_2 are independent; (3) if $X_1 = g^{x_1}$ and $X_2 = g^{x_2}$, the probability that the value of $\hat{Z}_1^r\hat{Z}_2 = \hat{Y}^s$ does not agree with the value of $\hat{Z}_1 = \hat{Y}^{x_1} \wedge \hat{Z}_2 = \hat{Y}^{x_2}$ is at most $1/q$ (if the latter holds, the former certainly holds).

5.1.2. User registration

As Section 4 discussed, each user has a true and secret identity ID_A. User A holds a final private key pair $\{x_A, \sigma_A\}$ as well as a final public key pair $\{y_A, \mu_A\}$. The CA, who issues the certificates, has a master private key x_{CA}, a master public key y_{CA} where $y_{CA} \leftarrow g^{x_{CA}} \bmod p$, and a random and secret value α_A which is randomly chosen from $\mathbb{Z}_q^*$. The certificate is a pair (y_A, μ_A) where $\mu_A \leftarrow g^{\alpha_A}$, $y_A \leftarrow g^{x_A}$. Each user holds a sole private key σ_A which can be constructed only by either *CA* or *A*, i.e., $h_A \leftarrow H_1(y_A \parallel \mu_A)$ and $\sigma_A \leftarrow \nu_A \oplus H_1(y_A \parallel {\mu_A}^{x_A}\, {y_{CA}}^{x_A h_A})$. Thus, for a specific user A, its certificate is not deterministic. In addition, CA uses the secret keys, i.e., x_{CA} and α_A, to construct a certificate, and only CA holds the cryptographic materials to validate it. On the other hand, owing to the binding of both the true identity ID_A where $\langle H_1(ID_A \parallel x_A)\rangle$ and an anonymous identity y_A, each certificate is created for that specific user.

5.2. Encrypted signature and verification

As mentioned above, any user A can utilize the message signature and verification protocols to achieve further anonymous authentication. Considering the higher security requirements, i.e., confidentiality and privacy-preserving, the user A should encrypt and sign all messages so as to appropriate for the secure online payment case. Accordingly other users need to validate these received messages in terms of signatures, usually, which have been bound with messages before sent out. Besides that, in this proposed framework, there are two types of messages, i.e., the confidential message and the ordinary message. The confidential messages mainly refer to some typical e-commerce transactional proofs, i.e., bills, receipts, and account, whereas the ordinary messages, by contrast, provide some information opened to the general public, i.e., traffic information. Due to the broadcast nature in VANETs, all broadcasted messages are subject to be traced or

Table 3
Notations.

Notation	Explanation
$\mathbb{G}, q, g$	$\mathbb{G}$ is a Gap-Diffie–Hellman group of order q and generator g
$\mathbb{Z}_q, \mathbb{Z}_q{}^*$	An additive group and a multiplicative group of order q
A, B	The user A and the user B
x_{CA}, y_{CA}	A master private key and a master public key for CA
ID_A, x_A	The user A's true identity ID_A and secret value x_A
y_A	The user A's anonymous identity and public key y_A
σ_A	The user A's partial private key
μ_A, ν_A	The user A's partial public key
Enc, Dec	Encryption and decryption with a symmetric cryptosystem
φ_A	The signature of message from the user A
C_{msg}, M_{msg}	The ciphertext message and its plaintext

be observed by attackers. Thus, with the purpose to enhance the security, we use two versions of encryption. Version 1 is for the confidential message and version 2 is only used for the ordinary message. In addition, we use a variant of *Schnorr signature* [28] and *Reduced MR(p)-ElGamal signature* [29] as building block to provide the message signing and verifying features. That is, the security of the Alg. 1 and Alg. 2 depends partly on the *Schnorr signature* and *Reduced MR(p)-ElGamal signature* assumptions.

Algorithm 1 Message Encryption and Signature

```
1:  procedure MESSAGE ENCRYPTION AND SIGNATURE($M_{msg}^A$; A; B)
2:      $k_A \leftarrow \mathbb{Z}_q^*$;  $r_A \leftarrow y_{CA}^{x_A} g^{-k_A}$
3:      if Version 1 then
4:          $C_{msg}^A \leftarrow M_{msg}^A \oplus H_1(y_B \parallel y_B{}^{x_A})$
5:      else if Version 2 then
6:          $C_{msg}^A \leftarrow M_{msg}^A \oplus H_1(y_B \parallel y_{CA})$
7:      end if
8:      $\rho_A \leftarrow H_2(y_A \parallel C_{msg}^A \parallel g^{k_A})$;  $\tau_A \leftarrow H_2(y_A \parallel C_{msg}^A \parallel r_A g^{k_A})$
9:      $s_A \leftarrow (k_A - \sigma_A \rho_A) \bmod q$;  $t_A \leftarrow (k_A - x_A \tau_A) \bmod q$
10:     $\varphi_A \leftarrow (y_A, s_A, t_A, \rho_A, \tau_A, TTL)$
11:     $RSU \leftarrow (\varphi_A, C_{msg}^A)$
12: end procedure
```

Algorithm 1 shows the process to encrypt and sign a message. In this paper, all messages from a user A will be encrypted and uploaded to a *RSU* over a shared broadcasting channel so as to enable the vehicle-to-infrastructure applications. The y_B is a public key of the receiver B, i.e., *ISP* or *SPG*. Thus B can use a private key x_B to decrypt C_{msg}^A. In this algorithm, for the sake of improving efficiency and simplifying procedures, an ordinary message is encrypted using a code obfuscation technique to avoid behavior analysis, whereas a confidential message conducts a standard encryption by introducing the private key x_A, i.e., line 3–6. With a view to seeking advanced security properties, two encryption operations can be developed into other two forms. For example, $H_1(y_B \parallel y_B{}^{\sigma_A})$ is for Version 1 and $H_1(y_B \parallel y_B{}^{x_A})$ is for Version 2. For the freshness concerns, all sessions have a limited and specified time to live (*TTL*). In addition, the r_A is a *Reduced MR(p)-ElGamal* signature which encapsulates a secret value k_A to ensure the freshness and confidentiality of signed message. In terms of the *Schnorr signature* assumption [28], ρ_A and τ_A are two hash values to support the verification of message integrity. s_A and t_A are two signature equations which encapsulate two pairs of secret values, i.e., (k_A, σ_A) and (k_A, x_A), respectively. Finally, φ_A is a signature on message that should be sent to the receivers, *e.g.*, *ISP*, *SPG* or *CA*, respectively.

RSU-aided online verification procedure is described in Alg. 2. To resist the reply attack or exhaustive attack, this algorithm first checks the validation of *TTL* and the number of access attempts *AccNum*. As for the latter case, the security of access control is provided by checking whether the *AccNum* has exceeded the allowed threshold value, i.e., *MaxAccNum*. According to the signature φ_A, RSU can validate two verification equations, i.e., $h_A{}^{(1)}$ and $g^{k_A{}^{(1)}}$, by which we can construct a commitment value $\rho_A{}^{(1)}$. Comparing with ρ_A and $\rho_A{}^{(1)}$, we can check if the signed message is successfully transmitted without any corruption. In particular, for Alg. 2, one main feature is to provide an initial judgment and to confirm the legitimacy of message signature. Only the verified messages can be forwarded to the target users, i.e., *ISP* or *SPG*; otherwise, it means the signature is invalid and thus RSU will submit a tracking request to CA. Details on tracking a suspicious user will be provided in the algorithm 4.

Algorithm 2 RSU-Aided Online Verification

```
procedure RSU-AIDED ONLINE VERIFICATION($C_{msg}^A$; A; $\varphi_A$)
    if $TTL$ has elapsed or $AccNum \geq MaxAccNum$ then
        return Reject
    end if
    $h_A{}^{(1)} \leftarrow H_1(y_A \parallel \mu_A^{(1)})$;  $g^{k_A{}^{(1)}} \leftarrow g^{s_A} \mu_A^{\rho_A} y_{CA}^{h_A{}^{(1)} \rho_A}$
    $\rho_A{}^{(1)} \leftarrow H_2(y_A \parallel C_{msg}^A \parallel g^{k_A{}^{(1)}})$
    $CA \leftarrow (\rho_A, \rho_A{}^{(1)}, g^{k_A{}^{(1)}})$
    if $\rho_A{}^{(1)} \neq \rho_A$ then
        $CA \leftarrow$ Tracking Request for the user A
    else
        $(ISP; SPG) \leftarrow C_{msg}^A$
    end if
end procedure
```

Furthermore, in Alg. 2, it is should be noted that RSU is semi-honest and provides three properties: (1) RSU can verify two hash values, $h_A{}^{(1)}$ and $\rho_A{}^{(1)}$, instead of CA, but it cannot check the correctness of the hash value τ_A; (2) Due to the limited number of failed access attempts, an attacker is difficult to pretend to be a legal user in a given time period *TTL* as long as RSUs have not been compromised; (3) It should be noted that RSU cannot decrypt any confidential message. That is, except the sender, this type of messages is visible only for the specified receivers or for CA in case of tracking.

Algorithm 3 Message Decryption

```
procedure RECEIVER MESSAGE DECRYPTION($C_{msg}^A$; B; A; $\varphi_A$)
    if Version 1 then
        $M_{msg}^A \leftarrow C_{msg}^A \oplus H_1(y_B \parallel y_A{}^{x_B})$
    else if Version 2 then
        $M_{msg}^A \leftarrow C_{msg}^A \oplus H_1(y_B \parallel y_{CA})$
    end if
return Accept
end procedure
```

The message decryption procedure is described in Alg. 3. This decryption procedure is straightforward. As mentioned, there are two encryption methods used by two types of messages respectively. As for the ordinary message, we use a simple and efficient scrambled data method to achieve the purpose of hiding plaintext message. To a certain extent, encrypting all messages

Please cite this article in press as: J. Song, et al., Secure authentication in motion: A novel online payment framework for drive-thru Internet, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.06.011

is to reduce the risk of being traced, modified or observed by attackers. In the case of confidential message, owing to the collision resistance property of hash functions, only the receiver can recover the plaintext message M_{msg}^{A} by using the private key x_{B}. In particular, the security in this decrypting operation depends on the *Gap Diffie–Hellman (GDH) Signature* [25] assumption.

5.3. *CA-aided traceability and batch auditing*

Considering the necessities of traceability and fast auditing, in this paper, we introduce a trusted third party CA to track abnormal activities and to further audit the verification results. As mentioned in Section 2, there exist two typical security threats in such a case. One is that, in order to gain access to private resources, the attacker may trick RSU by masquerading as a legitimate user. The another one is that, if an RSU is compromised, an attacker may act as a semi-trusted third party. That is, it is allowed to query or request additional information from the users and thus leads to serious security problems. Considering the above concerns, in this paper, we introduce two algorithms to ensure the security and the reliability of our designs, *e.g.*, Alg. 4 and Alg. 5.

Algorithm 4 CA Traceability

```
1: procedure CA TRACEABILITY($\mathsf{A}$; $\varphi_{\mathsf{A}}$; $\langle\rho_{\mathsf{A}}, {\rho_{\mathsf{A}}}^{(1)}\rangle$)
2:     ${\tau_{\mathsf{A}}}^{(2)} \leftarrow H_2(y_{\mathsf{A}} \parallel C_{msg}^{\mathsf{A}} \parallel y_{\mathsf{A}}^{x_{CA}})$
3:     if ${\tau_{\mathsf{A}}}^{(2)} \neq \tau_{\mathsf{A}}$ then
4:         $(\forall \mathsf{A};\ \forall RSU;\ \forall ISP;\ \forall SPG) \leftarrow$ broadcast "Risky User" $\mathsf{A}$
5:     else
6:         ${h_{\mathsf{A}}}^{(2)} \leftarrow H_1(y_{\mathsf{A}} \parallel \mu_{\mathsf{A}}^{(2)})$;   $g^{{k_{\mathsf{A}}}^{(2)}} \leftarrow g^{t_{\mathsf{A}}} y_{\mathsf{A}}^{\tau_{\mathsf{A}}^{(2)}}$
7:         ${\rho_{\mathsf{A}}}^{(2)} \leftarrow H_2(y_{\mathsf{A}} \parallel C_{msg}^{\mathsf{A}} \parallel g^{{k_{\mathsf{A}}}^{(2)}})$
8:     end if
9:     if ${\tau_{\mathsf{A}}}^{(2)} == \tau_{\mathsf{A}}$ and ${\rho_{\mathsf{A}}}^{(2)} \neq {\rho_{\mathsf{A}}}^{(1)}$ then
10:        $(\forall \mathsf{A};\ \forall RSU;\ \forall ISP;\ \forall SPG) \leftarrow$ broadcast "RSU Compromised"
11:    end if
12: end procedure
```

CA traceability is described in Alg. 4. This algorithm aims to detect abnormal entities, i.e., an illegitimate user or a compromised RSU. First, according to *Reduced MR(p)-ElGamal signature* [29], CA can construct the verification equation ${\tau_{\mathsf{A}}}^{(2)}$ via own holding a private key x_{CA} where $r_{\mathsf{A}} g^{{k_{\mathsf{A}}}^{(2)}} = y_{\mathsf{A}}^{x_{CA}}$. Then, CA checks if ${\tau_{\mathsf{A}}}^{(2)}$ is equal to τ_{A}. If it holds, it means that the signature from A is correct and thus RSU may exist as the possibility of being compromised; otherwise, it suggest that the user A has conducted a risky or malicious behavior. In order to verify this conclusion, CA constructs a commitment value ${\rho_{\mathsf{A}}}^{(2)}$ by calculating $h_{\mathsf{A}}^{(2)}$ and $g^{{k_{\mathsf{A}}}^{(2)}}$. It is noted that, due to the confidentiality of x_{A} and x_{CA}, only either CA or A can recover the ${\tau_{\mathsf{A}}}^{(2)}$ and τ_{A}, i.e., $r_{\mathsf{A}} g^{{k_{\mathsf{A}}}^{(2)}} = y_{\mathsf{A}}^{x_{CA}} \equiv y_{CA}^{x_{\mathsf{A}}}$. Finally, by a comparison of ${\rho_{\mathsf{A}}}^{(2)} \neq {\rho_{\mathsf{A}}}^{(1)}$ and ${\tau_{\mathsf{A}}}^{(2)} == \tau_{\mathsf{A}}$, it can check the results from RSU and the user A, respectively. If both of them hold, it means that RSU has been compromised; otherwise, it implies the user A is risky. Furthermore, CA will broadcast the alarm messages to other users and terminate all access attempts from user A.

CA-aided batch auditing procedure is described in Alg. 5, which emphasizes to verify the signatures efficiently and to support the batch auditing for those verified results. As above mentioned, we assume that n fast-moving vehicles may submit the authentication requests simultaneously to a nearby *RSU*, such as in a business district. For this case, if not a more effective authentication mechanism is provided, the *RSU* would have to drop these authentication requests to void the local network congestion. From the security protocol point of view, as an effective approach to address this concern is either to improve the verification efficiency or to mitigate the security workloads to others, i.e., the trusted third party. In this secure online payment framework, we implement a two-phase authentication schemes which integrate two above mentioned features. That is, the preceding algorithms provide the verifying and auditing features to the first phase, i.e., RSU-aided verification algorithm 2 and CA traceability algorithm 4, whereas the CA-aided batch auditing algorithm 5 accomplishes that to the second phase. In particular, the Alg. 5 has an important feature to support the batch verification and batch auditing, by which CA can reduce the computation workloads and improve authentication efficiency obviously.

Algorithm 5 CA-Aided Batch Auditing

```
1: procedure CA-AIDED BATCH AUDITING(RSU; $\varphi_{\mathbb{A}}$; $\langle\rho, \rho^{(1)}\rangle$; $g^{k^{(1)}}$)
2:     if Version 1 then
3:         $\mathbb{A}_{[1]}^{(1)} \leftarrow g^{{k_1}^{(1)}}$;  $\mathbb{C}_{[1]}^{(1)} \leftarrow s_1$;  $\mathbb{D}_{[1]} \leftarrow \alpha_1\rho_1$;  $\mathbb{E}_{[1]} \leftarrow x_{CA}h_1\rho_1$
4:         for $i$ from 2 to $n$ do
5:             $\mathbb{A}_{[i]}^{(1)} \leftarrow g^{{k_i}^{(1)}}\mathbb{A}_{[i-1]}$;  $\mathbb{C}_{[i]}^{(1)} \leftarrow s_i + \mathbb{C}_{[i-1]}^{(1)}$
6:             $\mathbb{D}_{[i]} \leftarrow \alpha_i\rho_i + \mathbb{D}_{[i-1]}$;  $\mathbb{E}_{[i]} \leftarrow x_{CA}h_i\rho_i + \mathbb{E}_{[i-1]}$
7:         end for
8:     else if Version 2 then
9:         ${\tau_1}^{(2)} \leftarrow H_2(y_1 \parallel C_{msg}^1 \parallel y_1^{x_{CA}})$;  $\mathbb{A}_{[1]}^{(2)} \leftarrow g^{t_1}{y_1}^{\tau_1^{(2)}}$;  $\mathbb{B}_{[1]} \leftarrow$
   $y_1^{\tau_1}$;  $\mathbb{C}_{[1]}^{(2)} \leftarrow t_1$
10:        for $i$ from 2 to $n$ do
11:            ${\tau_i}^{(2)} \leftarrow H_2(y_i \parallel C_{msg}^i \parallel y_i^{x_{CA}})$;  $\mathbb{A}_{[i]}^{(2)} \leftarrow g^{t_i}{y_i}^{\tau_i^{(2)}}\mathbb{A}_{[i-1]}^{(2)}$
12:            $\mathbb{B}_{[i]} \leftarrow y_i^{\tau_i}\mathbb{B}_{[i-1]}$;  $\mathbb{C}_{[i]}^{(2)} \leftarrow t_i + \mathbb{C}_{[i-1]}^{(2)}$
13:        end for
14:    end if
15:    if $\mathbb{A}_{[n]}^{(1)} \neq g^{\mathbb{C}_{[n]}^{(1)}+\mathbb{D}_{[n]}+\mathbb{E}_{[n]}}$ or $\mathbb{A}_{[n]}^{(2)} \neq \mathbb{B}_{[n]}g^{\mathbb{C}_{[n]}^{(2)}}$ or $\mathbb{A}_{[n]}^{(1)} \neq \mathbb{A}_{[n]}^{(2)}$ then
16:        return Reject
17:    else
18:        return Accept
19:    end if
20: end procedure
```

As shown in Alg. 5, the CA-aided batch auditing processes include two versions. Noted that Version 1 can be used as a batch version of Alg. 2, which focuses on the batch verifying signatures in an efficient way, but without introducing the auditing feature. By contrast, the Version 2 can not only support the batch checking but also achieve the batch auditing feature as well. Owing to a relative deep-level auditing, Version 2 is certainly slow in aspect of processing speed. Thus it is necessary to select one appropriate scheme in accordance with the actual requirements, i.e., Version 1 is for the busy time and Version 2 is for idle time. In this paper, one summation result of a user i denotes $\mathbb{A}_{[i]}^{(1)}$ where $1 \leq i \leq n$, and other similar expressions include $\mathbb{A}_{[i]}^{(2)}$, $\mathbb{B}_{[i]}$, $\mathbb{C}_{[i]}^{(1)}$, $\mathbb{C}_{[i]}^{(2)}$, $\mathbb{D}_{[i]}$, and $\mathbb{E}_{[i]}$ as well. During the CA-aided batch auditing process, CA first iteratively calculates $\mathbb{A}_{[i]}^{(1)}$, $\mathbb{C}_{[i]}^{(1)}$, $\mathbb{D}_{[i]}$, and $\mathbb{E}_{[i]}$ by validating a checked result from RSU, i.e., $g^{{k_{\mathsf{A}}}^{(1)}}$, and extracting other components from user i, i.e., $s_{\mathbb{A}}$, $\alpha_{\mathbb{A}}\rho_{\mathbb{A}}$, and $x_{CA}h_{\mathbb{A}}\rho_{\mathbb{A}}$. So far, it achieves the checking with signatures, namely, Version 1. Second, CA constructs a new hash value ${\tau_{\mathbb{A}}}^{(2)}$ via own holding master secret key x_{CA}. As mentioned above, ${\tau_{\mathbb{A}}}^{(2)}$ or $\tau_{\mathbb{A}}$ can be constructed by CA or by user only with their own private keys, respectively. In this case, CA can calculate a summation value $\mathbb{A}_{[i]}^{(2)}$ by using ${\tau_{\mathbb{A}}}^{(2)}$ and $t_{\mathbb{A}}$. In addition, other two summation results $\mathbb{B}_{[i]}$ and $\mathbb{C}_{[i]}$ can be calculated by CA according to the signature $\varphi_{\mathbb{A}}$, respectively, i.e., $\tau_{\mathbb{A}}$, y_{A}, and $t_{\mathbb{A}}$. Here, it accomplishes the batch auditing feature and is denoted Version 2. As mentioned in Alg. 1, note that these three summation results can be computed only by users and by CA. Finally, if we can prove the equivalence among these summation results, i.e., $\mathbb{A}_{[n]}^{(1)}$, $\mathbb{A}_{[n]}^{(1)}$, $g^{\mathbb{C}_{[n]}^{(1)}+\mathbb{D}_{[n]}+\mathbb{E}_{[n]}}$, and $\mathbb{B}_{[n]}g^{\mathbb{C}_{[n]}^{(2)}}$, it means that these signatures and verifications on messages are true and valid; otherwise, it implies that there exist suspect users or RSUs. To find out the cause, it should run the CA traceability algorithm 4 further.

Please cite this article in press as: J. Song, et al., Secure authentication in motion: A novel online payment framework for drive-thru Internet, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.06.011

Correctness. The proposed CA-aided batch auditing scheme is correct and consistent, such as: first,

$$\mathbb{A}_{[n]}^{(1)} = g^{{k_1}^{(1)}} \times g^{{k_2}^{(1)}} \times \cdots \times g^{{k_{n-1}}^{(1)}} \times g^{{k_n}^{(1)}} = \prod_{i=1}^{n} g^{{k_n}^{(1)}}$$

$$\mathbb{A}_{[n]}^{(2)} = g^{{k_1}^{(2)}} \times g^{{k_2}^{(2)}} \times \cdots \times g^{{k_{n-1}}^{(2)}} \times g^{{k_n}^{(2)}} = \prod_{i=1}^{n} g^{{k_n}^{(2)}}$$

$$\mathbb{B}_{[n]} = (y_1)^{\tau_1} \times (y_2)^{\tau_2} \times \cdots \times (y_{n-1})^{\tau_{n-1}} \times (y_n)^{\tau_n} = \prod_{i=1}^{n} (y_n)^{\tau_n}$$

$$\mathbb{C}_{[n]}^{(1)} = s_1 \times s_2 \times \cdots \times s_{n-1} \times s_n = \sum_{i=1}^{n} s_n$$

$$\mathbb{C}_{[n]}^{(2)} = t_1 \times t_2 \times \cdots \times t_{n-1} \times t_n = \sum_{i=1}^{n} t_n$$

$$\mathbb{D}_{[n]} = \alpha_1\rho_1 \times \alpha_2\rho_2 \times \cdots \times \alpha_{n-1}\rho_{n-1} \times \alpha_n\rho_n = \sum_{i=1}^{n} \alpha_n\rho_n$$

$$\begin{aligned}\mathbb{E}_{[n]} &= h_1\rho_1x_{CA} \times h_2\rho_2x_{CA} \times \cdots \times h_{n-1}\rho_{n-1}x_{CA} \times h_n\rho_nx_{CA} \\ &= \sum_{i=1}^{n} h_n\rho_nx_{CA}\end{aligned}$$

second,

$$\begin{aligned}\mathbb{A}_{[n]}^{(1)} &= \prod_{i=1}^{n} g^{{k_n}^{(1)}} = \prod_{i=1}^{n} g^{s_n} \times {\mu_n}^{\rho_n} \times {y_{CA}}^{h_n\rho_n} \\ &= \prod_{i=1}^{n} g^{s_n} \times \prod_{i=1}^{n} g^{\alpha_n\rho_n} \times \prod_{i=1}^{n} g^{h_n\rho_nx_{CA}} \\ &= g^{\sum_{i=1}^{n} s_n} \times g^{\sum_{i=1}^{n} \alpha_n\rho_n} \times g^{\sum_{i=1}^{n} h_n\rho_nx_{CA}} = g^{\mathbb{C}_{[n]}^{(1)}+\mathbb{D}_{[n]}+\mathbb{E}_{[n]}}\end{aligned} \tag{1}$$

third,

$$\begin{aligned}\mathbb{A}_{[n]}^{(2)} &= \prod_{i=1}^{n} g^{{k_n}^{(2)}} = \prod_{i=1}^{n} g^{t_n} \times {y_n}^{\tau_n} \\ &= g^{\sum_{i=1}^{n} t_n} \times \prod_{i=1}^{n} {y_n}^{\tau_n} = \mathbb{B}_{[n]}g^{\mathbb{C}_{[n]}^{(2)}}.\end{aligned} \tag{2}$$

If both Eqs. (1) and (2) hold, and while $\mathbb{A}_{[n]}^{(1)}$ is equivalent to $\mathbb{A}_{[n]}^{(2)}$, it indicates that the CA-aided batch auditing procedure can successfully finish the batch verification and the batch auditing.

6. Security analysis

In this section, we show a detailed security analysis on three proposed schemes, including NCL-PKC encryption scheme, RSU-aided online verification scheme, and CA-aided traceability and batch auditing scheme, according to our previous security goals, i.e., Section 3.

6.1. Security of NCL-PKC encryption scheme

Considering the encryption scheme in Section 4, CA constructs a partial public key $p^2k_{\mathrm{A}} = \mu_{\mathrm{A}}$ for user A using an own holding secret number α_{A}, e.g., $\mu_{\mathrm{A}} = g^{\alpha_{\mathrm{A}}} \bmod p$, whereas the public key pk_{A} is denoted $pk_{\mathrm{A}} = y_{\mathrm{A}}$ where $y_{\mathrm{A}} = g^{x_{\mathrm{A}}}$. It should be noted that the x_{A} is a private key sk_{A} and is kept secret by A. In addition, a partial private key $s^2k_{\mathrm{A}} = \sigma_{\mathrm{A}}$ is provided by CA using the secret keys α and x_{CA} where $\sigma_{\mathrm{A}} = \alpha_{\mathrm{A}} + h_{\mathrm{A}}x_{CA} \bmod q$. Thus no adversary except *CA* can replace the A's final public key pair $\langle pk_{\mathrm{A}}, p^2k_{\mathrm{A}}\rangle$ without being detected. As mentioned in Table 2, it avoids the possible public key replacement attack and achieves the explicit authentication with the public key. Similarly, it is clear that no adversary including CA may successfully forge the A's final private key pair $\langle sk_{\mathrm{A}}, s^2k_{\mathrm{A}}\rangle$ with non-negligible probability. On the other hand, owing to the $H_1(ID_{\mathrm{A}} \parallel x_{\mathrm{A}})$ kept by CA, no efficient user should succeed in forging a pair of new private key without being given the master private key x_{CA}. In addition, this encryption scheme achieves the trust level 3 and no key escrow problem, as indicated in Table 2.

Moreover, with the purposes of consistency and security, we introduce the classic indistinguishability (*ind-atk*)-based security model for the NCL-PKC scheme, that is, under three types of attacks [13,30], including chosen plaintext attack (*cpa*), chosen ciphertext attack (*cca*1) and adaptive chosen ciphertext attack (*cca*2).

Definition 3. Let $\Pi = (\mathcal{G}_{\mathcal{CA}}, \mathcal{G}_{\mathcal{U}}, \mathcal{E}_{\mathcal{S}}, \mathcal{E}_{\mathcal{P}}, \mathcal{S}_{\mathcal{E}}, \mathcal{S}_{\mathcal{S}}, \mathcal{S}_{\mathcal{P}}, \mathcal{E}, \mathcal{D})$ be a NCL-PKC encryption scheme and $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2\}$ be a probabilistic polynomial-time (*ppt*) adversary. For attacks $atk \in \{cpa, cca1, cca2\}$ and $1^k \in \mathbb{N}$, let $\mathcal{O}_1$ and $\mathcal{O}_2$ be an oracle that outputs $b = b' \in \{0, 1\}$. We say that the scheme Π is secure against *atk* if the advantage of any *ppt* adversary $\mathcal{F}$ wins the following *ind-atk* game [30] is negligible:

Algorithm 6 The game in which $\mathcal{F}$ interacts with the challenger

```
$(pk_{CA}, sk_{CA}) \leftarrow \mathcal{G}_{\mathcal{CA}}(1^k)$
$(pk_U, sk_U) \leftarrow \mathcal{G}_{\mathcal{U}}(1^k)$
$(s^2k_U) \leftarrow \mathcal{E}_{\mathcal{S}}(1^k, sk_{CA}, sk_U, ID_U)$
$(p^2k_U) \leftarrow \mathcal{E}_{\mathcal{P}}(1^k, sk_{CA}, pk_U, ID_U)$
$(y_{IDU}) \leftarrow \mathcal{S}_{\mathcal{E}}(1^k, pk_{CA}, ID_U, x_{ID_U})$
$(m_0, m_1) \leftarrow {\mathcal{F}_1}^{\mathcal{O}_1}(p^2k_U, pk_U, pk_{CA}, sk_{CA})$
$b \leftarrow \{0, 1\}$
$\bar{c} \leftarrow \mathcal{E}(m_b, pk_U, p^2k_U, pk_{CA})$
$b' \leftarrow {\mathcal{F}_2}^{\mathcal{O}_2}(m_0, m_1, \bar{c})$
```

where algorithm 6 shows the game processes with simple symbolic description and

$$\begin{cases}\mathcal{O}_1 = \epsilon \quad \text{and} \quad \mathcal{O}_2 = \epsilon, & \text{if } atk = cpa, \\ \mathcal{O}_1 = D_{sk}(\cdot) \quad \text{and} \quad \mathcal{O}_2 = \epsilon, & \text{if } atk = cca1, \\ \mathcal{O}_1 = D_{sk}(\cdot) \quad \text{and} \quad \mathcal{O}_2 = D_{sk}(\cdot), & \text{if } atk = cca2.\end{cases}$$

Here, $\mathcal{F}_1$ outputs m_0 and m_1 with the same length and $\mathcal{F}_2$ is prohibited to query $\mathcal{O}_2(\bar{c})$. $\mathcal{F}$ is an *ind-atk* adversary, the security parameter is k, and there exists a negligible function $\texttt{negl}$. At the end of this game, the adversary outputs a guess b' for b. If $b' = b$, we say that the adversary wins. The adversary's advantage to win this game is defined as

$$Adv_{\mathcal{F},\Pi}^{int\text{-}atk}(1^k) = \left|P_r\left[b' = b\right] = 1\right| \le \frac{1}{2} + \texttt{negl}(1^k).$$

Definition 4. We say that the Π is secure against *ind-atk* if the advantage of any *ppt* adversary $\mathcal{F}$ win the following $Adv_{\mathcal{F},\Pi}^{int\text{-}atk}(1^k)$ game is negligible. The adversary's advantage in winning the game is defined as

$$Adv_{\mathcal{F},\Pi}^{int\text{-}atk}(1^k, t, q_o) = max\{Adv_{\mathcal{F},\Pi}^{int\text{-}atk}(1^k)\} \tag{3}$$

where the *max* function is taken over all adversaries that run for time t and make at most q_o queries to the decryption oracle.

6.2. Security of RSU-aided online verification

Security of message encryption and signature: As the mentioned in Section 5, the security of the message encryption and signature is mainly guaranteed by a variant of *Schnorr* (SCH) signature [28] and a *Reduced MR(p)-ElGamal* (RMR) signature [29]. In addition, r_A is a RMR signature which encapsulates a secret value k_A, so as to ensure the freshness and confidentiality of signature. Considering the weak collision resistance property, a one-way hash value $H_1(y_B \parallel y_B{}^{x_A})$ is used for the encryption of the confidential message M^A_{msg}. Here, the ciphertext message C^A_{msg} encapsulates a secret key x_A and further provides the confidentiality and authenticity. Similarly, another hash functions H_2 and a secret key k_A are used to encapsulate the secret values ρ_A and τ_A, respectively. In terms of the SCH assumption [28], it is obvious that ρ_A and τ_A can achieve the security properties, i.e., message integrity and authenticity. In particular, the s_A and t_A are two signature equations which are embedded into three secret values, k_A, σ_A, and x_A. Because of the hardness of RMR signature [29], the advantage of any *ppt* adversary being able to recover s_A and t_A without k_A, σ_A, and x_A known is negligible. Finally, the signature φ_A is typically guaranteed by the GDH signature [25] assumption. That is, without knowing the secret key, k_A, σ_A, and x_A, a malicious user is infeasible to forge a valid certificate.

Security of RSU-aided online verification: In general, the security of an RSU-aided online verification procedure typically focuses on three properties, *e.g.*, the integrity of message forwarding, the authenticity of sender and receiver, and the non-repudiation of transactional proofs [31]. During the RSU-aided online verification procedure, RSU is treated as a semi-honest entity which can check the validity of the one-way hash value $\rho_A{}^{(1)}$. To realize this purpose, RSU has to calculate two commitment values such as $h_A{}^{(1)}$ and $g^{k_A{}^{(1)}}$, notably whose security is based on the GDH signature [25] assumption. In addition, due to the security of SCH assumption [28], the hash value $\rho_A{}^{(1)}$ achieves three above mentioned security properties, including integrity, authenticity, and non-repudiation. Three cryptographic components, i.e., y_A, C^A_{msg}, and $g^{k_A{}^{(1)}}$, are encapsulated to ensure the above properties, respectively. Because of the hardness of the GDH assumption, the probability of finding a feasible $\mu_A^{(1)}$ or $\rho_A{}^{(1)}$ via forging is negligible. To calculate the commitment value $g^{k_A{}^{(1)}}$, an adversary has to forge a signature equation s_A, which is guaranteed by the hardness of RMR signature assumption [29]. On the other hand, as for any *ppt* adversary, it is infeasible to construct a validated verification equation $g^{s_A}\mu_A^{\rho_A}y_{CA}^{h_A{}^{(1)}\rho_A}$ based on the TDH trapdoor theorems [27].

6.3. Security of CA-aided traceability and batch auditing

Security of the CA traceability: In this paper, the CA is viewed as a trusted entity. Similarly with Alg. 1, CA can construct most of the cryptographic components, i.e., $h_A{}^{(2)}$, $g^{k_A{}^{(2)}}$, $\rho_A{}^{(2)}$, and $\tau_A{}^{(2)}$, etc., by means of the holding keys except for the private key x_A. From the security point of view, it provides the properties of privacy preserving and confidentiality so as to avoid revealing the user's true identity. In addition, the commitment value $\tau_A{}^{(2)}$ can only be rebuild by CA or by user himself, which also provides the assurance of the integrity and the non-repudiation of transmitted messages. CA can utilize own holding keys to reconstruct the other cryptographic components, i.e., $h_A^{(2)}$ and $g^{k_A{}^{(2)}}$. Furthermore, the master private key x_{CA} belongs to CA solely and no other party can forge it. Alternately, CA can calculate the commitment value $\tau_A{}^{(2)}$ according to the secret key x_{CA}. Both values all provide the properties such as integrity and traceability.

Security of the CA-aided batch auditing: The correctness of CA-aided batch auditing scheme has been explained in Section 5. There are some similar processes among Alg. 2, Alg. 5, and Alg. 4, so we focus on the main security properties of two versions. The Version 1 emphasizes on the batch verifying signatures, whereas the Version 2 can provide the batch checking and the batch auditing feature. Similar to Alg. 2, the security of Version 1 also relies on the hardness of the DCR signature [26] problem. Additionally, CA extracts two secret components $\alpha_{\mathbb{A}}\rho_{\mathbb{A}}$ and $x_{CA}h_{\mathbb{A}}\rho_{\mathbb{A}}$ to validate $\mathbb{A}^{(1)}_{[i]}$, $\mathbb{C}^{(1)}_{[i]}$, $\mathbb{D}_{[i]}$, and $\mathbb{E}_{[i]}$, respectively. In particular, CA constructs a new hash value $\tau_{\mathbb{A}}{}^{(2)}$ using the master secret key x_{CA}. As mentioned above, $\tau_{\mathbb{A}}{}^{(2)}$ can be calculated only by CA. In Version 2, CA can construct the summation values $\mathbb{A}^{(2)}_{[i]}$, $\mathbb{B}_{[i]}$, and $\mathbb{C}_{[i]}$ by using $\tau_{\mathbb{A}}{}^{(2)}$, $\tau_{\mathbb{A}}$, y_A, and $t_{\mathbb{A}}$, respectively. In terms of *DCR* assumption, the above summation results is secure as long as the adversary does not the secret primitives being used, i.e., $\alpha_{\mathbb{A}}$ and x_{CA}. In addition, the security of message decryption algorithm 3 is straightforward, whose security obviously relies on collision-resistant hash function property. That is, it implies that an adversary succeeds to decrypt the ciphertext C^A_{msg} with at most negligible probability.

Table 4
Time consumption on cryptographic operations.

Operations	Timings (ms)	Forms
aP	3.08	i.e., $a \in \mathbb{Z}_q^*$ and $P \in \mathbb{G}_1$
$e(P, Q)$	2.97	i.e., $P, Q \in \mathbb{G}_1, e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$
E_1	1.92	i.e., $a^b \pmod N$ and (a/N) for $a, b \in \mathbb{Z}_N^*$
E_2	0.77	i.e., g^r for $g \in GF(q)$ and $r \in \mathbb{Z}_N^*$
A	–	i.e., $a + b$ or $a \oplus b$ for $a, b \in \mathbb{Z}_N^*$
M	–	i.e., $a \cdot b \pmod N$ for $a, b \in \mathbb{Z}_N^*$
H	–	i.e., SHA-1 or MD5

7. Performance evaluation

In this section, we show a comprehensive evaluation with the proposed online payment framework. To concern the implementation details, we focus on the efficiency analysis, i.e., computation overhead and communication overhead, and simulation and numerical analysis specific to encryption/decryption cost and communication cost. The evaluation results show the feasibility of our framework in the drive-thru Internet environment.

7.1. Efficiency analysis

(1) *Computation overhead*: To achieve an 80-bit level of security, a previous study [32] shows the performance results, comparing three different pairing schemes with a standard 1024-bit RSA decryption scheme, which are achieved on the Pentium IV 3.0 GHz machine. According to the existing implementation results, i.e., [32,13], there are four main time consumption operations, as shown in Table 4: i.e., the elliptic operation P, the pairing operation $e(P, Q)$, the modular exponentiation operation E_1, and field exponentiation operation E_2. The experiment results in [32] show that the average time consumption is about 3.08 and 2.97 ms, with respect to a scalar multiplication in $\mathbb{G}_1$ and a $E(\mathbb{F}_p)$ *Tate* pairing, respectively. Three other operations include: modular addition A, modular multiplication M, and hash function H. In the future work, HPC methods [33,34] can be used to deal with computation overhead.

Furthermore, to evaluate computation overhead, we list the workload of each proposed algorithm, as shown in Table 5. In terms of the workloads for different operations, if without regard for the CA-aided batch auditing process, we can compute the total workloads for our framework, which is about 8.47 ms for version 1 and 7.7 ms for version 2, respectively. It should be noted that we neglect the time consumption of three operations, including A, M, and H, which takes very little time in terms of [32,13]. In addition, some iterative operations, i.e., nA, nM, and nH, will take a certain time if the n is large, i.e., 1000 times.

Table 5
Computation overhead.

Algorithms	Operations	Timings (ms)
Message *Enc*. and *Sig*. #1	$4E_2 + 4M + 3A + 3H$	3.08
Message *Enc*. and *Sig*. #2	$3E_2 + 4M + 3A + 3H$	2.31
RSU-Aided Online *Ver*.	$3E_2 + 3M + 2H$	2.31
Message *Dec*. #1	$1E_2 + 1A + 1H$	0.77
Message *Dec*. #2	$1A + 1H$	–
CA Traceability	$4E_2 + 1M + 3H$	3.08
CA-Aided Batch *Aud*. #1	$nE_2 + 3nM + 3nA$	–
CA-Aided Batch *Aud*. #2	$3nE_2 + nM + nA + nH$	–

#1: Version 1 #2: Version 2.
Enc.: Encryption *Dec*.: Decryption *Sig*: Signature.
Ver.: Verification *Aud*.: Auditing.

Table 6
Communication overhead.

Algorithms	Operations	Size (kb)
Message *Enc*. and *Sig*.	$6\|q\| + 1\|N\|$	4.096
RSU-Aided Online *Ver*.	$3\|q\| + 1\|N\|$	2.560
Message *Dec*.	–	–
CA Traceability	$1\|q\|$	0.512
CA-Aided Batch *Aud*.	$1\|q\|$	0.512

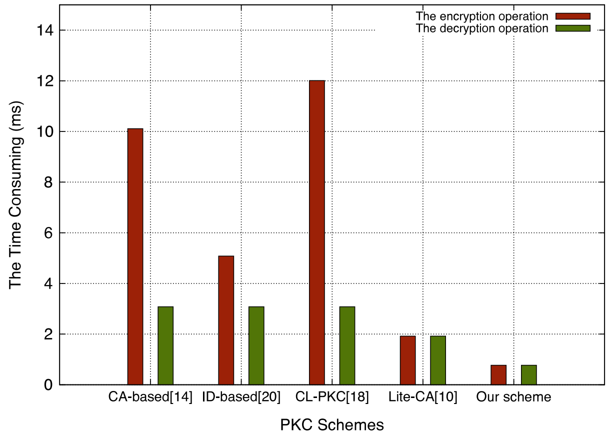

Fig. 3. Encryption and decryption cost comparison.

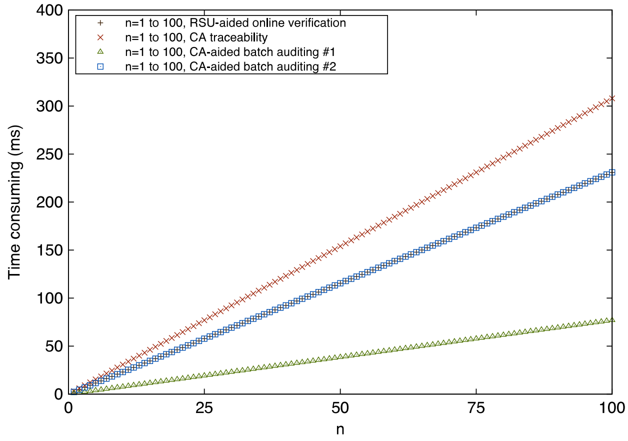

Fig. 4. Computational cost comparison.

(2) *Communication overhead*: As discussed above, to achieve an 80-bit level of security, a 1024-bit RSA signature to equal level need be chosen accordingly. Without loss of generality, we use a 512-bit long prime q and element length of 160-bit long based on the group $\mathbb{G}$, that is, $|N| = 1024$ bits, $|q| = 512$ bits, and $|\mathbb{G}| = 160$ bits. In our proposed framework, there exist three types of communication overheads, which come from the ciphertext message $|N|$, cryptographic tuples $|q|$, and the announcement $|q|$, respectively. If the length of an announcement is smaller than 512 bits, the total communication overheads will be less. In general, the message signature is one most used for each broadcasting. For instance, in Alg. 1, the length of signature φ is about $6|q| = 3.072$ kb. As for the ciphertext or plaintext message, i.e., C_{msg} or M_{msg}, they need to be broken up into blocks of size 1024 bits so as to matching the size of hash value, i.e., H_1. Thus the maximum size of these sent messages is approximately $6|q| + 1|N| = 4.096$ kb, and its average size is about 1.536 kb. In terms of the above settings, we list the communication overhead of each algorithm, as shown in Table 6.

7.2. Simulation and numerical analysis

For the detailed comparison, we evaluate the encryption and decryption cost with different schemes, including CA-based PKC [17] (specific to the *BasicCBE*), identity-based PKC (IBC) [23], certificateless (CL)-PKC [21], and *lite*-CA based PKC [13] and our proposed scheme. Table 7 shows the comparison of the encryption and decryption operations from different schemes [32,13]. Combining the results in Tables 5 and 7, we can see that the total time consumption of different schemes [17], [23], [21], [13], and ours are about 13.2, 8.1, 15.3, 3.8 and 1.54 ms for our Version 1, respectively. Here, Version 2 is neglected due to the very little time consumption. It also indicates that our proposed scheme is efficient in respect of encryption and decryption cost. Moreover, Fig. 3 shows that in our scheme the encryption cost and decryption cost are relatively better than the other four schemes, which implies the efficient performance results of our proposed framework.

On the other hand, we evaluate the total workloads of different algorithms with an increasing frequency from 1 to 100, including RSU-aided online verification algorithm 2, the CA traceability algorithm 4, and two versions of CA-aided batch auditing algorithm 5. Fig. 4 shows that three proposed algorithms in our framework are efficient. As shown in Fig. 4, two types of points overlap. They belong to the RSU-aided online verification algorithm 2 and the Version 2 in CA-aided batch auditing algorithm 5, respectively. As for algorithm 4 and algorithm 2, with the increasing frequency, the computation cost of cryptographic protocols increases rapidly, leading to a higher computation workloads. With respect to the same frequency, the efficiency of the CA-aided batch auditing algorithm 5 increases slowly, producing a less computation workloads. Specific to the frequency 100 or higher, it suggests that the workloads of the algorithm 5 take only about one quarter or two-thirds proportions in comparison with 4 and algorithm 2, respectively. Thus, with respect to communication overhead, our designs show a certain performance advantages and reflect the practical feasibility of the proposed framework.

Additionally, we conduct simulation evaluation and performance analyses with our proposed framework. To be more exact, we mainly investigate the network performance of the proposed protocol by introducing the previous security properties, especially for a typical drive-thru Internet along a highway segment. Our focus is on the contention nature of the communication between vehicular nodes and RSUs. With respect to our simulation, it is supposed that there are different numbers of vehicles to communicate competitively with an RSU simultaneously. To compare the total overhead with or without security measures introduced, we use a typical network simulator *NS*-2 to implement our simulation. In this paper, we mainly investigate the amount of the data uploaded by a vehicle in a one-hop scenario and show the possible extension to multiple hops. According to a representative analytical model [35], we analyze the intrinsic relationships among the vehicle density, the coverage of the AP, the network throughput

Table 7
Encryption and decryption cost.

Scheme	Encryption	Decryption	T.T.C (ms)
CA-based PKC [17] #1	$2e + 2P + 1M + 1A$	$1e + 1A$	13.24
ID-based PKC [23]	$1e + 1P + 1A$	$1e + 1A$	8.16
CL-PKC [21]	$3e + 1P + 1E_2 + 2A$	$1e + 2A$	15.39
lite-CA PKC [13]	$1E_1 + 2M + 3A$	$1E_1 + 0.5M + 4.5A$	3.84
Our Scheme #1	$1E_2 + 1A$	$1E_2 + 1A$	1.54
Our Scheme #2	$1A$	$1A$	–

T.T.C: Total Time Consumption.
#1: especially to *BasicCBE* scheme.

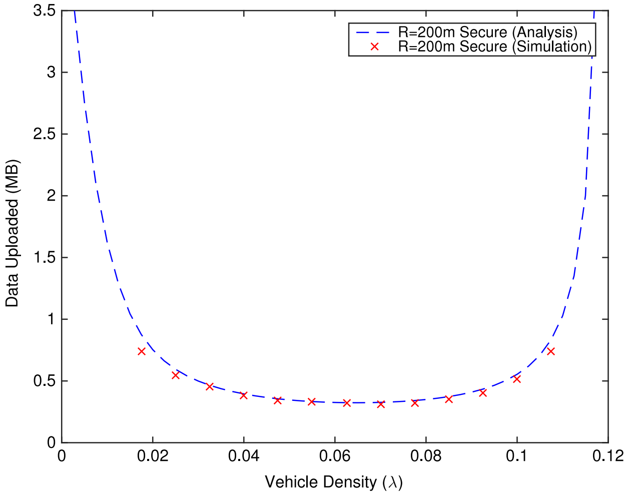

Fig. 5. Data uploaded per drive-thru with $R = 200$ m.

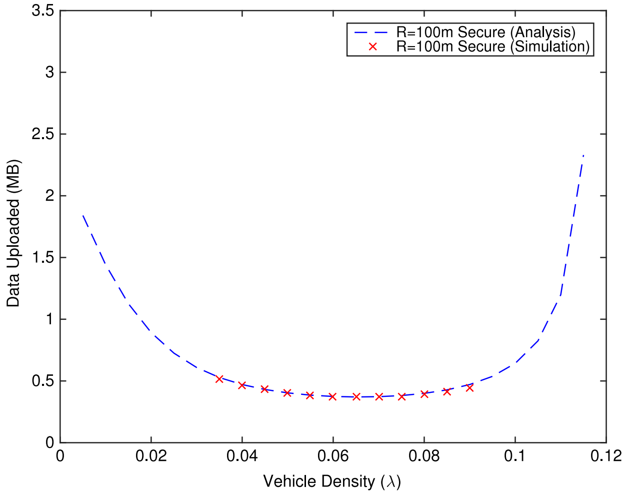

Fig. 6. Data uploaded per drive-thru with $R = 100$ m.

and the data uploaded amount from vehicles with the properties achieved.

The parameters of the MAC protocols used in our simulation are summarized as follows: two types of communication range are adopted respectively, *e.g.*, $R = 200$ m and $R = 100$ m at 11 Mbps, based on the setting for IEEE 802.11-based VANETs. The vehicles reach the RSU in terms of a Poisson arrival process with rate $\lambda \in [0.002, 0.12]$ for vehicles per meter, which reflects free-flow to jammed scenarios. Vehicles approach the RSUs at a speed of $v_f(1-\lambda/\lambda_{jam})$, where a common free flow speed $v_f = 56$ mi/hr ≈ 25.03 m/s and $\lambda_{jam} = 0.12$. The other MAC layer parameters are set and the analytical model is built as described in [35,36]. In terms of these existing solutions, we can obtain the frame service time, including the idle time slot, the transmission delay, the collision delay, and the transmission delay introduced by the maximum size of message. As above mentioned, this maximum size is about 4.096 kb, which is nearly one-half size of the data frame per drive-thru, i.e., 1 kb.

Considering this proposed framework, three algorithms, including message encryption and decryption, CA traceability, and CA-based batch auditing, take much less communication overheads, as shown in Table 6, so as hardly to embody the differences before and after introducing security properties. In addition, the RSU-aided online verification procedure has not introduced communication overhead since the all service requests should be handled locally by RSUs themselves. Hence, we focus on the simulation evaluation for the proposed message signature and verification procedure. Here, we mainly analyze the amount of data uploaded by each vehicle during the drive-thru interval, and further evaluate the performance influence by introducing the communication overhead in this procedure. Figs. 5 and 6 show the performance of uploading data by a vehicle respectively special with the range of 100 and 200 m. This explains that the analytical results closely match the simulation results as well. Both the simulation results demonstrate a non-monotonic relationship between the uploaded data amount and the vehicle density λ. It is determined by the nature of drive-thru contending communication.

Furthermore, as shown in both Figs. 5 and 6, when the transmission range R is fixed, i.e., $R = 200$ m or $R = 100$ m, it means that the less vehicles communicate with the RSU and the drive-thru time is short in such a case. Thus the more data can be uploaded by a specific vehicle. From another perspective, at very low vehicle densities, i.e., $\lambda \leq 0.01$, it implies higher vehicles speed and shows a high amount of uploading data. With the vehicular density increasing, when the communication contention among vehicles is intensified, thus it leads to a lower throughput, i.e., $\lambda \in [0.01, 0.08]$. As we have seen from Fig. 5 and Fig. 6, when the vehicular density further increases to the almost jam density λ_{jam}, the amount of uploaded data will approximate infinity, *e.g.*, $\lambda \geq 0.1$. This is theoretically reasonable because vehicles encounter congestion and have more time to upload. However, it is only an ideal condition which does not exist in reality. It can be explained that, when the contention is severe, it is difficult for each vehicle to keep a steady state to upload.

Comparing with both figures, we can see that the shape of curves is similar to each other. This is determined by the primitive traffic flow property, which mainly depends on the relationship between vehicle density, speed and flow-rate. On the other hand, we also compare the amount of data uploaded with or without the security properties achieved, in particular to two transmission ranges, i.e., $R = 100$ m or $R = 200$ m. Fig. 7 shows their simulation results. As mentioned, for two transmission ranges, with the increase of the vehicle density λ, the amount of uploading data by introducing two security measures is close. This indicates the feasibility of our proposed framework, especially for intermittently connected drive-thru Internet communication environment. In addition, when $\lambda = 0.06$, that means that vehicle moves at the speed of 12.5 m/s to pass by the RSU. With the same setting,

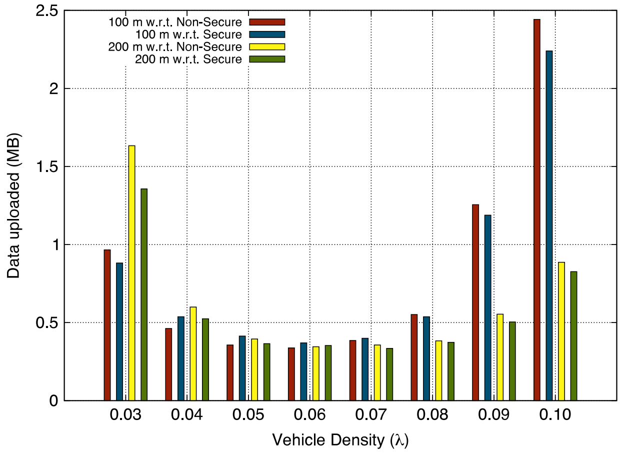

Fig. 7. Data uploaded comparison with $R = 100$ m and $R = 200$ m.

the data amount to be upload can reach 0.37 MB for $R = 100$ m and 0.33 MB for $R = 200$ m with security property, respectively. As a conclusion, we can see the overhead introduced by our security framework is limited. Furthermore, our simulation results also reflect the practical feasibility for the proposed framework, especially for the drive-thru Internet scenario.

8. Conclusions

Secure online payment services have been extensively investigated in networks with good stability and high reliability, i.e., wireless local area networks. Considering the security properties for drive-thru Internet applications, especially for the secure authentication services from the in-motion vehicles, existing solutions of online payment may introduce undesired overhead and fail to provide the security properties of authentication. In this paper, we propose an advanced secure online payment framework, providing various security and privacy properties, i.e., a new and secure CL-PKC encryption scheme, a more practical security model, and the optimized authentication efficiency. A comprehensive evaluation was conducted to show the security and feasibility of the proposed framework.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61303212, Grant 61332019, and Grant *U*1135004, by Hubei Provincial Natural Science Foundation of China under Grant 2014*CFB*192, and by the Fundamental Research Founds for National University, China University of Geosciences (Wuhan).

References

[1] G. Yan, D. Wen, S. Olariu, M.C. Weigle, Security challenges in vehicular cloud computing, IEEE Trans. Intell. Transp. Syst. 14 (1) (2013) 284–294. http://dx.doi.org/10.1109/TITS.2012.2211870.

[2] S. Gillani, F. Shahzad, A. Qayyum, R. Mehmood, A survey on security in vehicular ad hoc networks, in: Communication Technologies for Vehicles, 5th International Workshop, Nets4Cars/Nets4Trains 2013, Villeneuve d'Ascq, France, May 14–15, 2013. Proceedings, 2013, pp. 59–74.

[3] M. Wang, Q. Shen, R. Zhang, H. Liang, X. Shen, Vehicle-density-based adaptive MAC for high throughput in drive-thru networks, IEEE Internet Things J. 1 (6) (2014) 533–543. http://dx.doi.org/10.1109/JIOT.2014.2371897.

[4] L. Zhang, Q. Wu, A. Solanas, J. Domingo-Ferrer, A scalable robust authentication protocol for secure vehicular communications, IEEE Trans. Veh. Technol. 59 (4) (2010) 1606–1617. http://dx.doi.org/10.1109/TVT.2009.2038222.

[5] Y. Hao, Y. Cheng, C. Zhou, W. Song, A distributed key management framework with cooperative message authentication in vanets, IEEE J. Sel. Areas Commun. 29 (3) (2011) 616–629. http://dx.doi.org/10.1109/JSAC.2011.110311.

[6] L. Yeh, Y. Lin, A proxy-based authentication and billing scheme with incentive-aware multihop forwarding for vehicular networks, IEEE Trans. Intell. Transp. Syst. 15 (4) (2014) 1607–1621.

[7] M.D. Felice, A.J. Ghandour, H. Artail, L. Bononi, Enhancing the performance of safety applications in IEEE 802.11p/wave vehicular networks, in: 2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2012, San Francisco, CA, USA, June 25–28, 2012, 2012, pp. 1–9. http://dx.doi.org/10.1109/WoWMoM.2012.6263694.

[8] T.H. Luan, X. Ling, X. Shen, MAC in motion: Impact of mobility on the MAC of drive-thru Internet, IEEE Trans. Mob. Comput. 11 (2) (2012) 305–319. http://dx.doi.org/10.1109/TMC.2011.36.

[9] M. Khabbaz, W. Fawaz, C.M. Assi, A simple free-flow traffic model for vehicular intermittently connected networks, IEEE Trans. Intell. Transp. Syst. 13 (3) (2012) 1312–1326. http://dx.doi.org/10.1109/TITS.2012.2188519.

[10] S. Al-Sultan, M.M. Al-Doori, A.H. Al-Bayatti, H. Zedan, A comprehensive survey on vehicular ad hoc network, J. Netw. Comput. Appl. 37 (2014) 380–392. http://dx.doi.org/10.1016/j.jnca.2013.02.036.

[11] X. Lin, R. Lu, C. Zhang, H. Zhu, P.-H. Ho, X. Shen, Security in vehicular ad hoc networks, IEEE Commun. Mag. 46 (4) (2008) 88–95.

[12] A. Wasef, X.S. Shen, EMAP: expedite message authentication protocol for vehicular ad hoc networks, IEEE Trans. Mob. Comput. 12 (1) (2013) 78–89. http://dx.doi.org/10.1109/TMC.2011.246.

[13] X. Dong, L. Wei, H. Zhu, Z. Cao, L. Wang, An efficient privacy-preserving data-forwarding scheme for service-oriented vehicular ad hoc networks, IEEE Trans. Veh. Technol. 60 (2) (2011) 580–591. http://dx.doi.org/10.1109/TVT.2010.2095432.

[14] C. Zhang, X. Lin, R. Lu, P. Ho, RAISE: an efficient rsu-aided message authentication scheme in vehicular communication networks, in: Proceedings of IEEE International Conference on Communications, ICC 2008, Beijing, China, 19–23 May 2008, 2008, pp. 1451–1457.

[15] C. Lee, Y. Lai, Toward a secure batch verification with group testing for VANET, Wirel. Netw. 19 (6) (2013) 1441–1449.

[16] C. Chen, W. Tsai, Y. Chen, W. Tsaur, Using a stored-value card to provide an added-value service of payment protocol in VANET, ITC 42 (4) (2013) 369–379.

[17] C. Gentry, Certificate-based encryption and the certificate revocation problem, in: Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003, Proceedings, 2003, pp. 272–293.

[18] K.R. Choo, J. Nam, D. Won, A mechanical approach to derive identity - based protocols from Diffie-Hellman-based protocols, Inform. Sci. 281 (2014) 182–200.

[19] C. Boyd, K.-K.R. Choo, Security of two-party identity-based key agreement, in: Ed Dawson, Serge Vaudenay (Eds.), Progress in Cryptology – Mycrypt 2005: First International Conference on Cryptology in Malaysia, Kuala Lumpur, Malaysia, September 28–30, 2005. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-540-32066-1, 2005, pp. 229–243. http://dx.doi.org/10.1007/11554868_17.

[20] K.R. Choo, Secure Key Establishment, in: Advances in Information Security, vol. 41, Springer, 2009.

[21] S.S. Al-Riyami, K.G. Paterson, Certificateless public key cryptography, in: Advances in Cryptology - ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, November 30–December 4, 2003, Proceedings, 2003, pp. 452–473.

[22] G. Lippold, C. Boyd, J.M.G. Nieto, Strongly secure certificateless key agreement, in: Pairing-Based Cryptography - Pairing 2009, Third International Conference, Palo Alto, CA, USA, August 12–14, 2009, Proceedings, 2009, pp. 206–230.

[23] D. Boneh, M.K. Franklin, Identity-based encryption from the weil pairing, SIAM J. Comput. 32 (3) (2003) 586–615. http://dx.doi.org/10.1137/S0097539701398521.

[24] S. Jun, H. Chunjiao, Z. Lei, T. Shanyu, Z. Huanguo, Toward an rsu-unavailable lightweight certificateless key agreement scheme for vanets, Commun. China 11 (9) (2014) 93–103.

[25] D. Boneh, B. Lynn, H. Shacham, Short signatures from the weil pairing, in: Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9–13, 2001, Proceedings, 2001, pp. 514–532.

[26] A.P. Sarr, P. Elbaz-Vincent, J. Bajard, A secure and efficient authenticated Diffie-Hellman protocol, in: Public Key Infrastructures, Services and Applications - 6th European Workshop, EuroPKI 2009, Pisa, Italy, September 10–11, 2009, Revised Selected Papers, 2009, pp. 83–98.

[27] D. Cash, E. Kiltz, V. Shoup, The twin Diffie-Hellman problem and applications, J. Cryptology 22 (4) (2009) 470–504.

[28] C. Schnorr, Efficient signature generation by smart cards, J. Cryptology 4 (3) (1991) 161–174.

[29] K. Nyberg, R.A. Rueppel, Message recovery for signature schemes based on the discrete logarithm problem, Des. Codes Cryptogr. 7 (1–2) (1996) 61–81.

[30] J. Katz, Y. Lindell, Introduction to Modern Cryptography, Chapman and Hall/CRC Press, 2007.

[31] K.R. Choo, C. Boyd, Y. Hitchcock, The importance of proofs of security for key establishment protocols: formal analysis of Jan-Chen, Yang-Shen-Shieh, Kim-Huh-Hwang-Lee, Lin-Sun-Hwang, and Yeh-Sun protocols, Comput. Commun. 29 (15) (2006) 2788–2797.

[32] M. Scott, N. Costigan, W. Abdulwahab, Implementing cryptographic pairings on smartcards, in: Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10–13, 2006, Proceedings, 2006, pp. 134–147.

Please cite this article in press as: J. Song, et al., Secure authentication in motion: A novel online payment framework for drive-thru Internet, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.06.011

[33] D. Chen, L. Wang, A.Y. Zomaya, M. Dou, J. Chen, Z. Deng, S. Hariri, Parallel simulation of complex evacuation scenarios with adaptive agent models, IEEE Trans. Parallel Distrib. Syst. 26 (3) (2015) 847–857.

[34] D. Chen, X. Li, L. Wang, S.U. Khan, J. Wang, K. Zeng, C. Cai, Fast and scalable multi-way analysis of massive neural data, IEEE Trans. Comput. 64 (3) (2015) 707–719.

[35] Y. Zhuang, J. Pan, V. Viswanathan, L. Cai, On the uplink MAC performance of a drive-thru Internet, IEEE Trans. Veh. Technol. 61 (4) (2012) 1925–1935.

[36] J. Song, Y. Zhuang, J. Pan, L. Cai, Certificateless secure upload for drive-thru Internet, in: Proceedings of IEEE International Conference on Communications, ICC 2011, Kyoto, Japan, 5–9 June, 2011, 2011, pp. 1–6. http://dx.doi.org/10.1109/icc.2011.5962528.

Jun Song is currently an associate professor of computer science at the China University of Geosciences, Wuhan, China. He received his Ph.D. degree in computer science from Wuhan University, China. His area of specialization is cryptography application and information security, and his current research interests include security analysis of cryptography application in wireless networks, applied network security, and cryptography security for big data.

Fan Yang received the Bachelor and Master degree and the Ph.D. from Wuhan University, all in computer science. She is currently an Associate Professor of computer science with the China University of Geosciences. Her research area of specialization is information security, and her current research interests include applied security for cloud computing and big data.

Wang is the Dean and "ChuTian" Chair Professor at School of Computer Science, China Univ. of Geosciences (CUG), and a Professor at Inst. of Remote Sensing & Digital Earth, Chinese Academy of Sciences (CAS). Prof. Wang received B.E. & M.E. from Tsinghua Univ. and Doctor of Eng. from Univ. Karlsruhe (Magna Cum Laude), Germany. Prof. Wang is a Fellow of IET, Fellow of British Computer Society. Prof. Wang serves as an Associate Editor of IEEE Tran. Computers and IEEE Tran.on Cloud Computing. His main research interests include high performance computing, e-Science, and spatial data processing.