A Privacy-Preserving Mobile Payment System for Mass Transit

Jeonil Kang, *Member, IEEE*, and DaeHun Nyang, *Member, IEEE*

***Abstract*— In the near future, mobile payment systems based on smartphones are expected to be widely applied in various environments, including transit services. When passengers use mass transit, their private information, such as their identity and route, may be made available to some related organizations, such as transit agencies, financial institutions, mobile carriers, and providers of smart cards, among others, even when the passengers may not want their information to be revealed. To protect passenger privacy, this paper proposes a privacy-preserving transit payment system based on traceable signatures, identity-based signatures, and anonymous signatures. In addition to passenger privacy, the proposed system facilitates the proactive blocking of misbehaving passengers, free-transfer services (or transfer discount), and postpaid programs. We demonstrate that the performance of this system is good enough for immediate deployment based on various experiments.**

***Index Terms*— Mobile payment, mass transit system, user privacy, anonymity, traceable signature.**

I. Introduction

MOBILE payment is a payment service performed with a mobile device instead of with plastic cards. To submit a payment in a mobile payment system, a user is required to pass some authentication test such as finger print scanning. In 1997, mobile payment in Helsinki, Finland, was made available in Coca Cola vending machines through the use of mobile phones; in late 2002, SK Telecom and KTF (KT in 2012), major mobile carriers in Korea, launched post-payment programs designed for mobile phones (Moneta and K-Commerce, respectively). Since 2006, East Japan Railways has provided Mobile Suica, a payment service based on mobile phones. Deutsche Telekom (T-Mobile) has offered mobile payment services in Germany and Poland since 2011 and in the U.S., the Netherlands, and the Czech Republic since 2012. On May 26, 2011, Google Inc. introduced Google Wallet, which allowed users to store debit cards and credit cards in a smart phone. Later, as a successor to Google Wallet, Google Inc. released Android Pay on September 11, 2015. On September 9, 2014, Apple Inc. announced Apple Pay, which operated on iPhone 6 or later. Apple Pay has been offered in the U.S. since October 20, 2014 and in the U.K. since July 14, 2015. Samsung Electronics introduced their mobile payment system, called Samsung Pay, on March 2015 and launched the service in Korea on August 20, 2015.

Among many applications, transit services represent a major application area for mobile payment systems, as demonstrated by the above discussion. In addition, many organizations, such as transit agencies, financial institutions, mobile carriers, and providers of smart cards, are involved in mobile payment systems for mass transit. In existing transit service systems, passengers' private information, such as their identity and route, can be revealed to such organizations, and they may share this information to balance accounts. However, it is difficult to ensure that the mishandling of passengers' private information by these organizations does not occur. This information can provide firms with various benefits; however, passengers may want to keep such information private. In other words, passengers' identities and routes should remain anonymous while organizations balance their accounts, which is called *anonymous balancing*.

One simple approach to anonymous balancing is to use pseudonyms instead of passengers' real identities. This approach has already been adopted in the field of prepaid transportation cards. Here, firms differentiate between passengers based only on the serial numbers on transportation cards. However, although the passengers' identities are hidden, firms can still trace their routes because each passenger tends to use the same transportation card for a long period of time. Therefore, to prevent the tracing of passengers' routes, the pseudonyms should be changeable at each use. Untraceable RFID authentication schemes protect RFID tags from eavesdroppers by changing the identifiers in every authentication attempt [9], [21], [23], [27]. However, we suppose that these schemes are not applicable to payment systems in mass transit (hereafter "transit payment systems"). In these authentication schemes, an online connection to the database that maintains the real IDs of RFID tags is essential. It is obvious that this online connection is a major burden for transit payment systems. For example, in Korea, one of the most developed countries in terms of communications infrastructure, people were not be able to pay for transit fares using their transit cards for about one hour on November 7, 2011. This disruption occurred because of a failure in the payment authorization sever. In addition, given other potential security

Manuscript received May 22, 2015; revised November 17, 2015 and July 12, 2016; accepted November 21, 2016. Date of publication January 4, 2017; date of current version July 31, 2017. This work was supported by the National Research Council of Science and Technology Grant by the Korean Government (MSIP) under Grant CRC-15-05-ETRI. The Associate Editor for this paper was B. F. Ciuffo. *(Corresponding author: DaeHun Nyang.)*

The authors are with the Department of Computer Science and Engineering, Inha University, Incheon 22212, South Korea (e-mail: dreamx@isrl.kr; nyang@inha.ac.kr).

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/TITS.2016.2636919

problems, such as identifier synchronizing, identifier cloning, or authentication forging, RFID authentication schemes must be carefully adopted in the transit payment system.

To eliminate the need for online connections, we can build a transit payment system based on anonymous signatures such as group and traceable signatures providing signers' anonymity. In addition, anonymous signatures are free from security problems associated with RFID authentication schemes. However, to use anonymous signatures in the transit system, we should consider three nontrivial problems in addition to the computing power. *The first problem* is related to the passenger signature. Assume that a passenger signs only a service fee to hide his or her route from the relevant organizations. In this case, however, the transit agency does not know whether or not the passenger provided false information, because it cannot know from when and where the passenger started to use the transit service. In contrast, if the passenger signs routes, then the financial institution that knows his or her real name may be able to trace him or her. *The second problem* involves the blockage of misbehaving passengers. Blocking misbehaving passengers is an essential part of any transit payment system. Many anonymous signatures, such as group and traceable signatures, can technically support the revocation of misbehaving users. For this, all users need to reissue their signing keys (reissuance-based revocation) or the verifier needs to check whether a given signature is listed on the revocation list (CRL-based revocation) [1]. Neither reissuance nor CRL-based revocation methods are readily applicable to transit payment systems. The former places substantial burdens on all passengers; therefore, if the system periodically collects and revokes misbehaving passengers, these passengers can abuse the transit service until the next revocation. The latter delays the entrance and exit procedures of transit services because the verification of an anonymous signature requires a number of computations proportional to the number of misbehaving passengers. On the other hand, revocable group signatures can be used to block misbehaving passengers [16], [17]. Using revocable group signatures, a passenger can prove that he or she does not belong to the revocation list. However, to generate revocable group signatures, the passenger should know the up-to-date revocation list, whose size increases according to the number of passengers and revoked passengers. Considering that there are millions of passengers, revocable group signatures are not yet suitable. *The third problem* arises from the fact that not only passengers, but also transit agencies can misbehave. Signatures generated by passengers confirm their use of transit services. However, transit agencies can intentionally discard some signatures from unspecified passengers in order to receive monetary penalties and insist on their misbehavior. To clear themselves of false accusations, affected passengers should also have evidence of their use of transit services.

In this paper, we propose a privacy-preserving transit payment system based on anonymous signatures that can address these three problems and thus

- guarantee anonymity of passengers' identities and routes,
- provide means for proactively blocking misbehaving passengers, and
- be robust to active attacks by passengers or transit agencies.

Additionally, the proposed system supports 1) *free-transfer services* (or transfer discount) and 2) *postpaid programs*. For these properties, we construct a transit payment system based on traceable and identity-based signatures.

The rest of this paper is organized as follows. Section II provides a literature review. Section III describes the system model, including the network model, trust relationships, and the threat model, and considers the security requirements for the proposed system. Section IV introduces some cryptographic primitives essential for the proposed system and explains the essential functions that organize the system. Section V details the proposed system, including a full system description of smartphones, which have a rich computing power. Section VI discusses the performance of the proposed system, including a comprehensive security analysis based on the security of cryptographic primitives and a performance analysis based on the experimental results, and Section VII presents the study conclusions.

II. Literature Review

A. Privacy Issues in Mobile Payment Systems

1) Privacy-Preserving Mobile Payment Systems: Many electronic cash or mobile payment systems have been proposed in the last decade. Konidala *et al.* [15] found the limit of the existing anonymous electronic cash payment systems in terms of a huge burden of banks and proposed a pre-paid mobile payment system based on blind signature and HTTPS. In the scheme, a smartphone submits a payment to a merchant by issuing cheques and updating pre-paid cash certificates. When making a payment, a customer remains anonymous to the bank because the cash certificate is signed using a blind signature and to the merchant because the bank pays instead of the customer. Isaac and Zeadally [14] introduced a mobile payment model, where the client does not need to communicate directly to the merchant. In their scheme, the payment gateway submits payments to merchants instead of mobile clients. Yang and Lin [29] pointed out and resolved some drawbacks of Isaac and Zeadally's payment system, caused by inadequate use of symmetric key encryption system.

For passenger convenience, the quick response from the back-end system is essential in the transit payment system. However, in some mobile payment systems, the participation of a payment gateway or bank, which may introduce a serious delay, is unavoidable for security reasons.

2) Mobile Payment Systems in the Real World: In the real world, only a few mobile payment systems such as Apple Pay, Android Pay, and Samsung Pay support off-line payment because certain hardware equipment such as NFC (Near Field Communication) terminals is required for communication. The main security concern of those systems is how the systems store the authentication tokens or PAN (Primary Account Number) in order to prevent card fraud.

Apple Pay by Apple Inc. uses authentication tokens received from credit card companies instead of storing PAN or credit card numbers and card verification codes (CVCs).

When making a payment to a merchant, the smartphone generates a one-time payment token, referred to as a device account number (DAN), from the authentication token after finishing the user authentication via Touch ID. The one-time payment token is transmitted to a terminal via NFC. The authentication token is stored in the security element in AP, as is the fingerprint. Android Pay by Google Inc. has similar structures to Apple Pay in various key features of one-time payment token, two-factor authentication, and NFC transaction. In those mobile payment systems, the user privacy such as user purchase history is totally dependent on the security policies of smartphone manufacturers.

B. Privacy Issues in Vehicular/Transit Systems

1) Privacy-Preserving Protocols for Vehicular Communication: Vehicular communication is an emerging network composed of roadside units (RSUs) and onboard units (OBUs). OBUs are short-range communications devices positioned on vehicles, and RSUs are located on the roadside to collect road information or to offer OBUs various services such as traffic information, route optimization, and pricing/payments. The ad hoc network made up of RSUs and OBUs is referred to as a vehicular ad hoc network (VANET). Because IEEE 1609.2 covers vehicular communication associated with security services, security is an important issue in vehicular communication. Many researchers have studied privacy-preserving protocols for vehicular communication [11], [18], [28].

On the other hand, some researchers have proposed traffic information systems using mobile phones and telecommunication infrastructures instead of using OBUs and RSUs. Herrera *et al.* [12] presented a traffic monitoring field experiment, Mobile Century, using 100 vehicles carrying GPS-equipped mobile phones. In their experiment, to manage privacy concerns, a sampling strategy based on virtual trip lines (VTLs) was used. Gisdakis *et al.* [10] introduced a secure and privacy-preserving traffic information system using AGPS-equipped smartphones and back-end infrastructure. To guarantee security and privacy, the smartphones obtain group credentials through TLS channels over TOR (The Onion Router) network and attach the group signatures on traffic reports.

2) Privacy-Preserving Electronic Toll/Traffic Pricing Systems: Privacy-preserving electronic toll pricing focuses on solving the user privacy problem in an electronic toll pricing (ETP) system that aims to reduce the delay on toll roads. To collect a toll, a tollbooth needs to know where a car has visited. Therefore, the goal of a privacy-preserving ETP system is to secure the travel records of drivers and to allow the toll collectors to correctly charge the tolls.

A privacy-preserving ETP system normally consists of three entities: an onboard unit (OBU), which is a GPS-equipped in-car device; a toll service provider (TSP); and a toll charger (TC). An OBU stores the trajectory data and submits the commitments of the trajectory data to the TSPs. The commitments are used to balance the tolls. To prevent cheating by the OBU, the TC should be able to observe a vehicle equipped with an OBU. Many researchers have studied privacy-preserving ETP systems [2], [5], [7], [20], [22], [25], [26]. To construct a privacy-preserving ETP system, various cryptographic primitives such as a zero-knowledge proof, a blind identity-based signature, and a group signature have been applied.

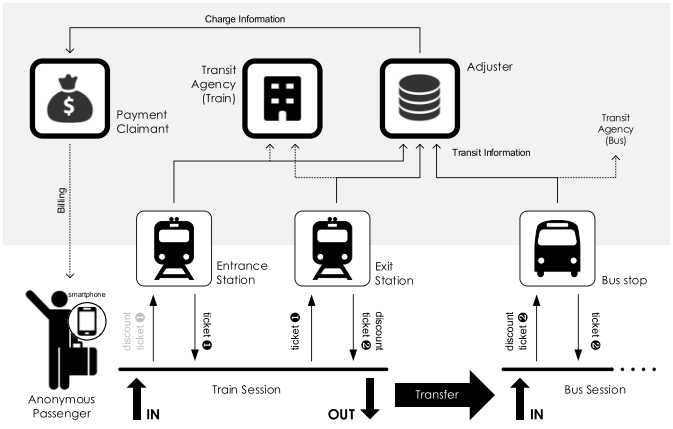


Fig. 1. Network model of the transit payment system. Passengers use transit services such as free-transfer services (or transfer discount) from several transit agencies. Transit information on how passengers receive transit services is delivered to an adjuster and a payment claimant; after a certain period of time, passengers pay for transit services through the payment claimant.

Privacy-preserving ETP systems have goals similar to those of the proposed system but have different network models and trust relationships. In ETP systems, the client (OBU) stores trajectory data and reports their commits to the service provider (the insurance provider, PA, or TSP). The use of commitments requires a strong assumption that the client never cheats or that a government organization should monitor the vehicle. To avoid this assumption, we can use a digital signature scheme to verify trajectory data. Here the client signs the data with challenges from the service provider, and signatures are instantly verified by the service provider. Unfortunately, this method is not suitable for vehicular networks because of their high speed. In the proposed system, passengers move slower than vehicles, which allows for the use of digital signatures to verify entrance and exit data. As a result, the government organization is not directly involved in the system. In addition, the proposed system facilitates stronger privacy than privacy-preserving ETP systems, which focus on hiding the vehicle's trajectory from the service provider.

III. System Model and Security Requirements

A. Network Model and Trust Relations

As shown in Fig. 1, we assume that four entities participate in the proposed system:

- ***Passengers*** can receive transit services from several transit agencies by showing the membership certificates received from the payment claimant. The system allows for free-transfer services, which allow passengers to freely change from transit services of one transit agency to those of another before arriving at their destination. At some point in the future, passengers will have to pay fees for transit services through a payment claimant.

To communicate with transit agencies, passengers need to hold wireless computing devices such as smartphones.

- ***Transit agencies*** provide passengers with transit services (e.g., bus or metro) after verifying their membership certificates. Afterward, these agencies collect passenger signatures as evidence of their use of transit services and send this information to an adjuster. The transit agencies can charge for their transit services by sending transit information (i.e., passenger signatures) to an adjuster. Each transit agency operates a terminal (hardware equipment) for each bus stop or subway station in order to communicate with passengers and to open/close gates. We assume that the transit agencies do not collude to access passengers' personal information or profit from passengers.
- ***Adjusters*** are the independent companies that collect passengers' transit records from transit agencies and help transit agencies receive fees for their services by cooperating with payment claimants. In the real world, providers of mass transit cards such as T-money (Korea Smart Card Co., Ltd) and TAP can play a role as the adjuster. Transit card firms currently provide passengers with means of payment for transit services from different transit agencies. In addition, they provide managed information to transit agencies as revocation managers, and they determine whether passengers are revoked and notify transit agencies with the result.
- ***Payment claimants*** charge passengers fees for transit services as financial institutions based on charge information sent by an adjuster. Payment claimants send all fees from passengers, except for brokerage fees, to the adjuster. In the real world, debit/credit card companies can play a role as a payment claimant. Today, many credit card companies are deeply involved in transit systems by integrating transit cards into their credit cards.

B. Threat Model

A back-end network established between terminals, transit agencies, adjusters, and payment claimants can be protected using existing security techniques. Therefore, we assume no attackers from outside the back-end network. On the other hand, a front-end network temporally established between passengers and terminals is open. An attacker in the front-end network may be able to listen to all messages between a passenger and a terminal and try to inject some messages by impersonating the passenger or terminal. We assume that an active attacker is a misbehaving passenger or terminal.

We also assume that no attacker uses physical characteristics of passengers such as radio fingerprinting, communication protocols, faces, or clothes. Such characteristics must be considered in the real world but are beyond the scope of this paper.

1) Passenger Privacy: An attacker may try to compromise various aspects of passenger privacy. The level of passenger privacy that can be achieved varies across entities because each entity obtains different information. To fully compromise passenger privacy, the attacker should be able to obtain both identity and route information.

- ***Identity anonymity***: An anonymous passenger should be able to keep his or her identity hidden.

Note that we cannot achieve identity anonymity for payment claimants because they know the real names of passengers.

Unlike identity information, the route information consists of several entrance and exit terminals of several transit agencies. Even a transit agency cannot obtain all information on routes of a certain passenger unless he or she used only the corresponding transit agency. Because we consider passenger anonymity, we need to classify passenger route privacy into session and transfer untraceability. To fully trace a passenger, an attacker should break both session untraceability and transfer untraceability.

- ***Session untraceability***: An anonymous passenger should be able to keep his or her session information (from the entrance terminal to the exit terminal) hidden.
- ***Transfer untraceability***: An anonymous passenger should be able to keep his or her transit information (from a transit agency to another transit agency) hidden.

2) Payment Security: Among the entities in the proposed system, only passengers and transit agencies can perform active attacks. A passenger's misbehavior leads directly to the transit agency's financial loss, whereas transit agency misbehavior causes passenger loss. To satisfy payment security, an anonymous passenger is forced to honestly pay for transit services (*passenger honesty*), and a transit agency is forced to honestly ask passengers to pay for its services (*transit agency honesty*).

C. Security Properties

Based on the threat model, we consider three potential attackers: eavesdroppers, passengers, and transit agencies. An eavesdropper can listen to every message, which can be a problem for passenger privacy. If a passenger is an active attacker, then he or she may be concerned only about finding ways to pay less. If a transit agency is an attacker, then it may be concerned about passenger privacy and try to find ways to charge more. Payment claimants and adjusters may be concerned about passenger privacy in the real world. Therefore, there is a clear need for minimizing passenger transit information given to payment claimants and adjusters. Table I shows the security requirements supported by the proposed system.

IV. Preliminaries

A. Bilinear Pairing and Linear Encryption

1) Bilinear Pairing: Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be cyclic additive groups of order q generated by P_1 and P_2, respectively. Let $\mathbb{G}_T$ be a cyclic multiplicative group of order q. Suppose the isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$. Then $P_1 = \psi(P_2)$. A bilinear pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ with the following properties:

- Bilinearity: For all $P, Q \in \mathbb{G}_1 \times \mathbb{G}_2$, and $a, b \in \mathbb{Z}_q^*$, $e(aP, bQ) = e(P, Q)^{ab}$.
- Non-degeneracy: $e(P_1, P_2)$ is a generator of $\mathbb{G}_T$, i.e. $e(P_1, P_2) \neq 1_{\mathbb{G}_T}$.
- Computable: There exists an efficient algorithm for computing $e(P, Q)$ for all $P, Q \in \mathbb{G}_1 \times \mathbb{G}_2$.

TABLE I
SECURITY PROPERTIES FOR PASSENGER ANONYMITY AND PAYMENT SECURITY IN THE PROPOSED SYSTEM

		Potential Attackers			Other Entities	
		Eavesdropper	Passenger	Transit Agency	Adjuster	Payment Claimant
Passenger Anonymity	Identity anonymity	✓		✓	✓	
	Session untraceability	✓				△
	Transfer untraceability	✓		△		△
Payment Security	Passenger honesty		✓			
	Transit agency honesty			✓		

△: achieved because of lack of information

2) Linear Encryption: Boneh *et al.* [3] propose a linear encryption scheme for encrypting a group member's identity in the group signature. The ElGamal encryption scheme [8], a traditional public key encryption scheme, may be broken by an attack on unlinkability if a bilinear pairing for which the isomorphism $\psi' : \mathbb{G}_1 \rightarrow \mathbb{G}_2$ exists is used. The linear encryption scheme has the following three functions:

- LE.Setup: For a security parameter, this outputs a public key $(U, V, H) \in \mathbb{G}_1$ and a private key $(\xi_1, \xi_2) \in \mathbb{Z}_q^*$ such that $\xi_1 U = \xi_2 V = H$.
- LE.Enc: This function encrypts the given plaintext $M \in \mathbb{G}_1$ by using the public key (U, V, H) and random numbers $\alpha, \beta \in \mathbb{Z}_q^*$ and outputs the ciphertext $C = (C_1, C_2, C_3) = (\alpha U, \beta V, M + (\alpha + \beta)H)$.
- LE.Dec: This decrypts the given ciphertext $C = (C_1, C_2, C_3)$ by using the private key (ξ_1, ξ_2) and outputs the plaintext $M = C_3 - (\xi_1 C_1 + \xi_2 C_2)$.

As shown in [3] and [9], the LE.Dec function in the linear encryption scheme can be used as the GS.Open function in group signature schemes. Therefore, a private key in the linear encryption scheme can also be used as an open key in group signature schemes.

B. Extension of Linear Encryption

In this section, we present one important construction of linear encryption with an additional capability, called label-extraction. Hwang *et al.* [13] propose a group signature scheme with controllable linkability and a method for extracting unique labels from the linear combination encryption (LCE). Their construction of LCE explicitly embeds a unique link label as well as a real identity into a cipher-text. Instead of using LCE, we extend the linear encryption scheme to extract unique labels as follows:

- LE.Setup$'$: For a security parameter, this outputs a public key $(U, V, H) \in \mathbb{G}_1$, a private key $(\xi_1, \xi_2) \in \mathbb{Z}_q^*$, and a link key (ℓ, L) such that $\xi_1 U = \xi_2 V = H$, $\ell = \xi_1/\xi_2$, and $L = -\xi_2 P_2$.
- LE.GetLabel: This extracts a label B from the given ciphertext $C = (C_1, C_2, C_3)$ such that $B = e(C_3, P_2) \cdot e(\ell C_1 + C_2, L)$.

For the given ciphertext $C = (\alpha U, \beta V, M + (\alpha + \beta)H)$, B is equal to $e(M, P_2)$. LE.GetLabel outputs the same label even from different ciphertexts if the encrypted message is the same.

This extension gives a shorter cipher-text than LCE [13], which consists of four elements in $\mathbb{G}_1$, and also it allows any traceable signature scheme using the linear encryption to be used in the proposed system. Note that the way to construct the GetLabel function varies depending on the encryption scheme used in the traceable signature.

C. Identity and Label

To ensure that an adjuster efficiently performs revocation verification, a payment claimant as a group master may provide the open key to the adjuster. If so, however, the passengers can be traced by the adjuster until they change their membership certificates. In addition, the use of an open key for the revocation verification is not acceptable in the conventional traceable signature scheme. To mitigate the problem, the payment claimant in the proposed system provides link keys rather than the open key. By periodically changing open key and re-issuing link keys, the payment claimant can prevent the adjuster from tracing passengers for a long period. If the payment claimant renews the open key, link keys are naturally changed as described in LE.Setup$'$. Consequently, labels are also changed, while identities are not; thus, passengers cannot be traced. The adjuster can only distinguish revoked passengers according to the labels. In short, the payment claimant permanently distinguishes passengers based on the passenger identity, and the adjusters temporally distinguish passengers based on the passenger label.

D. Notations and Essential Functions

Table II summarizes the notions. We use the following functions:

- CurrentTime : $void \rightarrow \{0, 1\}^{32}$
  This function returns the current time in the unix timestamp.
- H : $\{0, 1\}^* \rightarrow \{0, 1\}^{\ell}$
  This cryptographic hash function hashes a given bit string of arbitrary length and gives a bit string of constant length.
- Cert.Verify$_{pk}$: CERT $\rightarrow$ true/false
  This function determines whether a given certificate CERT is valid. It returns false if it is and true otherwise.

TABLE II
NOTIONS

Notions	Definitions
MTT	Maximum free-transfer duration
$isTransfer$	Flag to allow for free-transfer, $isTransfer \in \{\mathsf{true}, \mathsf{false}\}$
ID	Passenger identity
EID	Encrypted passenger identity (by the linear encryption)
LB	Label of the encrypted identity (by the extended linear encryption)
CERT	Public key certificate, $(pk, id) \in$ CERT
TTIK	Transfer ticket
TIK	Ticket
RCP	Receipt
pk / sk	Public and secret keys of the public key encryption
tpk / tik / tok / tlk / tsk	Public, issuing, open, link, and signing keys of the traceable signature
ipk / imk / isk	Public, master, and signing keys of the identity-based signature
apk / amk / ask	Public, master, and signing keys of the anonymous signature
σ^{ts} / σ^{as} / σ^{is}	Traceable, anonymous, and identify-based signatures
π^{cla} / π^{cla2}	Claim proof of the traceable signature scheme

Function 1 VerifyAndCheck Run by the Adjuster

Input: $(tpk, tlk, pk_{\text{adj}}, sk_{\text{adj}}), (id_{\text{on}} : t_{\text{on}}, \text{TTIK}, N_1, \sigma_1^{\mathsf{ts}})$
Output: RVK/DNY/OK, $isTransfer$
if $\mathsf{TS.Verify}_{tpk}(\mathsf{H}(t_{\text{on}} \| \mathsf{H}(\text{TTIK} \| N_1) \| id_{\text{on}}), \sigma_1^{\mathsf{ts}}) = \mathsf{true}$ **then**
  $\text{LB}_1 \leftarrow \mathsf{TS.GetLabel}_{tlk}(\sigma_1^{\mathsf{ts}})$
  if $\mathsf{SearchRVKDB}(\text{LB}_1) = \mathsf{true}$ **then**
    return RVK
  else
    $(\text{EID}, t_{\text{off}}, \sigma_4^{\mathsf{as}}) \leftarrow \mathsf{PK.Dec}_{sk_{\text{adj}}}(\text{TTIK})$;
    $isTransfer \leftarrow \mathsf{false}$
    if $\mathsf{AS.Verify}_{apk}(\text{EID} \| t_{\text{off}}, \sigma_4^{\mathsf{as}}) = \mathsf{true}$ **then**
      $\text{LB}_2 \leftarrow \mathsf{TS.GetLabelEncId}_{tlk}(\text{EID})$
      if $\text{LB}_1 = \text{LB}_2$ **and** $(t_{\text{off}} - t_{\text{on}}) < MTT$ **then**
        $isTransfer \leftarrow \mathsf{true}$
    return OK, $isTransfer$
else
  return DNY

CERT is issued by a trusted third party based on public-key infrastructure (PKI).

- $\mathsf{GetEncId} : \sigma^{\mathsf{ts}} \rightarrow \text{EID}$
  This function gives the encrypted identity EID from a traceable signature σ.
- $\mathsf{TS.GetLabelEncId}_{tlk} : \text{EID} \rightarrow \text{LB}$
  This function extracts a label LB from the encrypted identity EID of a given signature using a link key tlk. Inherently, $\mathsf{TS.GetLabel}_{tlk} = \mathsf{TS.GetLabelEncId}_{tlk}(\mathsf{GetEncId}(\sigma^{\mathsf{ts}}))$.
- $\mathsf{SearchRVKDB} : \text{LB} \rightarrow \mathsf{true}/\mathsf{false}$
  This function checks whether a given label LB is in the database containing labels of revoked passengers. It returns true if LB is in the database and false otherwise This function is used only in $\mathsf{VerifyAndCheck}$. See Function 1.
- $\mathsf{CalcFee} : (id_1, id_2, t_1, t_2) \rightarrow \Delta$
  This function calculates a fee for a passenger using identities of entrance and exit terminals and times.

V. PROPOSED PRIVACY-PRESERVING TRANSIT PAYMENT SYSTEM

A. Overview

To use the proposed system, a passenger follows entrance and exit procedures modeled on transit payment systems in the real world. Figure 1 shows abstracted entrance and exit procedures.

In the proposed system, transfer ticket TTIK and ticket TIK play core roles in preventing the misbehavior of passengers and transit agencies. The passenger basically generates a traceable signature in each procedure, which indicates his or her acknowledgment of the correctness of the procedure. Based on traceable signatures, entrance and exit terminals generate given identity-based and anonymous signatures for the passenger.

TIK is evidence that the passenger passed a certain entrance terminal. In the entrance procedure, a passenger generates a traceable signature by signing the entrance information. If the traceable signature is valid, the entrance terminal generates an identity-based signature by signing the traceable signature (with entrance information). The passenger stores the entrance information, the traceable signature, and the identity-based signature as ticket TIK. TIK is provided to the exit terminal in the next exit procedure. By doing so, TIK prevents the passenger from providing the exit terminal with false entrance information.

On the other hand, TTIK prevents exit terminals discarding the exit record of passengers and supports the free-transit service for passengers. By discarding an exit record, transit agencies may insist that a passenger does not process the exit procedure. In the exit procedure, a passenger submits TIK obtained from the previous entrance terminal to the exit terminal. With TIK, the exit terminal calculates the service fee and requests the passenger to sign the service free using a traceable signature. The passenger provides the claim proof in order to prove that both traceable signatures given in entrance and exit procedures are generated by a single signing key. The exit terminal confirms the claim proof and generates an anonymous signature by signing discount information. The passenger encrypts the anonymous signature with discount information using the adjuster's public key and stores the cipher-text as transfer ticket TTIK. TTIK is given to the next entrance terminal and is handed over to the adjuster. If the previous exit terminal discards the exit record, then the adjuster can detect it.

With the anonymous signature, an identity-based signature is given to the passenger in the exit procedure. The exit terminal generates the identity-based signature by signing the exit information as the entrance terminal does on entrance

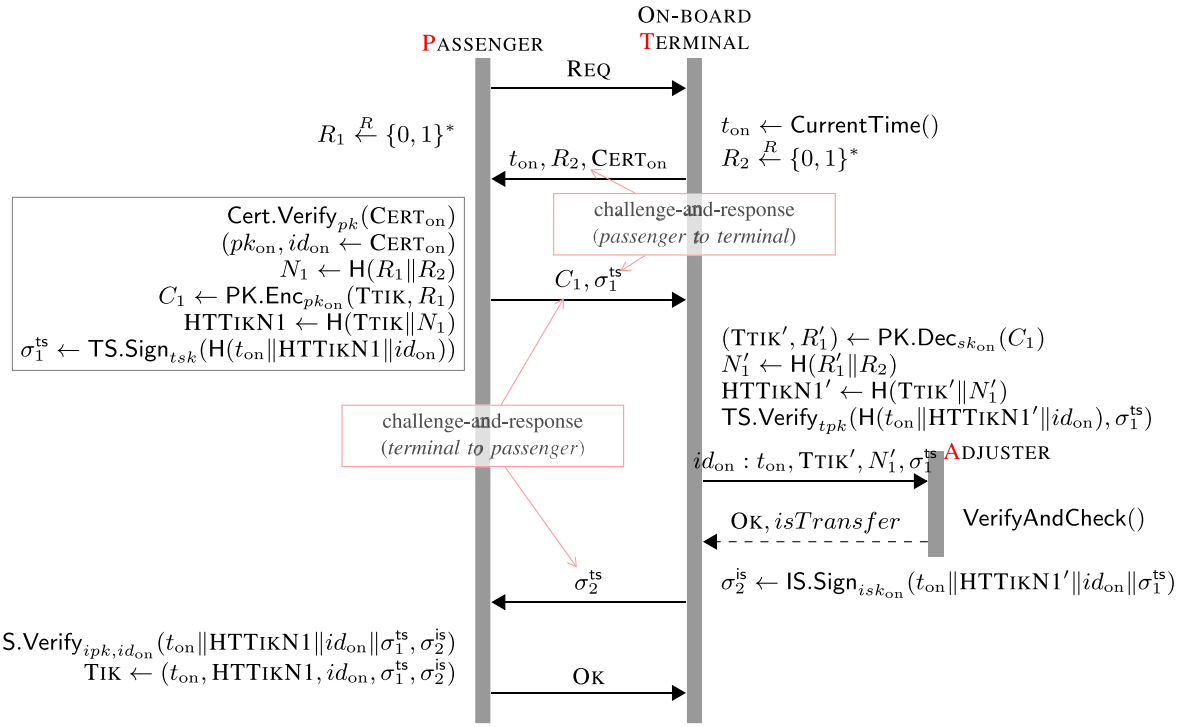


Fig. 2. Communication messages in the entrance procedure. At the end of the entrance procedure, the passenger obtains TIK, and the entrance terminal obtains σ_1^{ts}. The transfer ticket TTIK is forwarded to the adjuster, and the passenger sends TIK to the next exit terminal.

information. After the successful exit procedure, the passenger stores entrance and exit information with all traceable and identity-based signatures as the receipt RCP. RCP can be used in the abnormal situation for proving passenger honesty.

In the standard traceable signature scheme, the user revocation is a large burden. Instead of using reissuing/CRL-based revocation mechanisms of traceable signature schemes, the proposed system reduces the revocation costs using passenger labels. When the adjuster receives a traceable signature in the entrance procedure, the adjuster that has the link key extracts a label from the signature and compares it to the revoked passenger labels. However, the on-line connection for revocation verification is not mandatory; the temporary on-line failure or network delay during revocation verification does not seriously affect entrance procedure.

Periodically (e.g., once a day or once a week), the adjuster performs adjusting and charging tasks based on the entrance and exit records. During this adjustment, the adjuster recalculates fees according to the transportation policy. If the adjuster detects some misbehavior of a transit agency, then it charges the passenger the minimum fee. In contrast, if it detects some misbehavior of a passenger, it charges the maximum fee. If there is no evidence of any misbehavior, then the adjuster discounts the fee if the passenger uses the transit service again within a certain period of time or area. After the adjustment, the adjuster forwards records and requires billing to the payment claimant.

After finishing the adjustment, the payment claimant renews the adjuster's link key in order to prevent the adjuster from tracing passengers.

B. Entrance and Exit Procedures for Mobile Devices

1) Setup: The certificate authority, an outside entity of the proposed system, announces its public key pk and issues public key certificates for terminals. The payment claimant generates public keys (tpk, ipk, apk), master keys (tik, tok, imk, amk), and a link key tlk. It announces the group public key tpk and assigns the link key tlk to the adjuster.

The payment claimant issues a user secret key tsk to a passenger and assigns signing keys (isk, ask) to each terminal. The adjuster holds a link key tlk and its own public key pk_{adj} and private key sk_{adj}. Each terminal has its own identity x and a pair composed of a public key pk_x and a private key sk_x. It holds the public key certificate CERT_x, a signing key isk_x for the identity-based signature scheme, and ask_x for an anonymous signature scheme. Each passenger holds a secret key tsk.

2) Entrance Procedure: Figure 2 illustrates the entrance procedure. Note that id_{on} and t_{on} used in σ_1^{ts} and σ_2^{is} imply that the anonymous passenger PSR and the on-board terminal $\mathrm{TML}_{\mathrm{on}}$ implicitly agree upon the current time and $\mathrm{TML}_{\mathrm{on}}$ location.

(e1) Passenger submits transfer ticket. In the entrance procedure, PSR sends the transfer ticket TTIK to $\mathrm{TML}_{\mathrm{on}}$ and receives the ticket TIK. TTIK can be obtained after the previous exit procedure. If PSR does not hold TTIK, then he or she sets TTIK as null. However, if TTIK is revealed to eavesdroppers, they might trace the last terminal. Therefore, TTIK should be encrypted with $\mathrm{TML}_{\mathrm{on}}$'s public key pk_{on} before PSR sends the message. PSR can obtain the public key pk_{on} from $\mathrm{TML}_{\mathrm{on}}$'s public key certificate $\mathrm{CERT}_{\mathrm{on}}$.

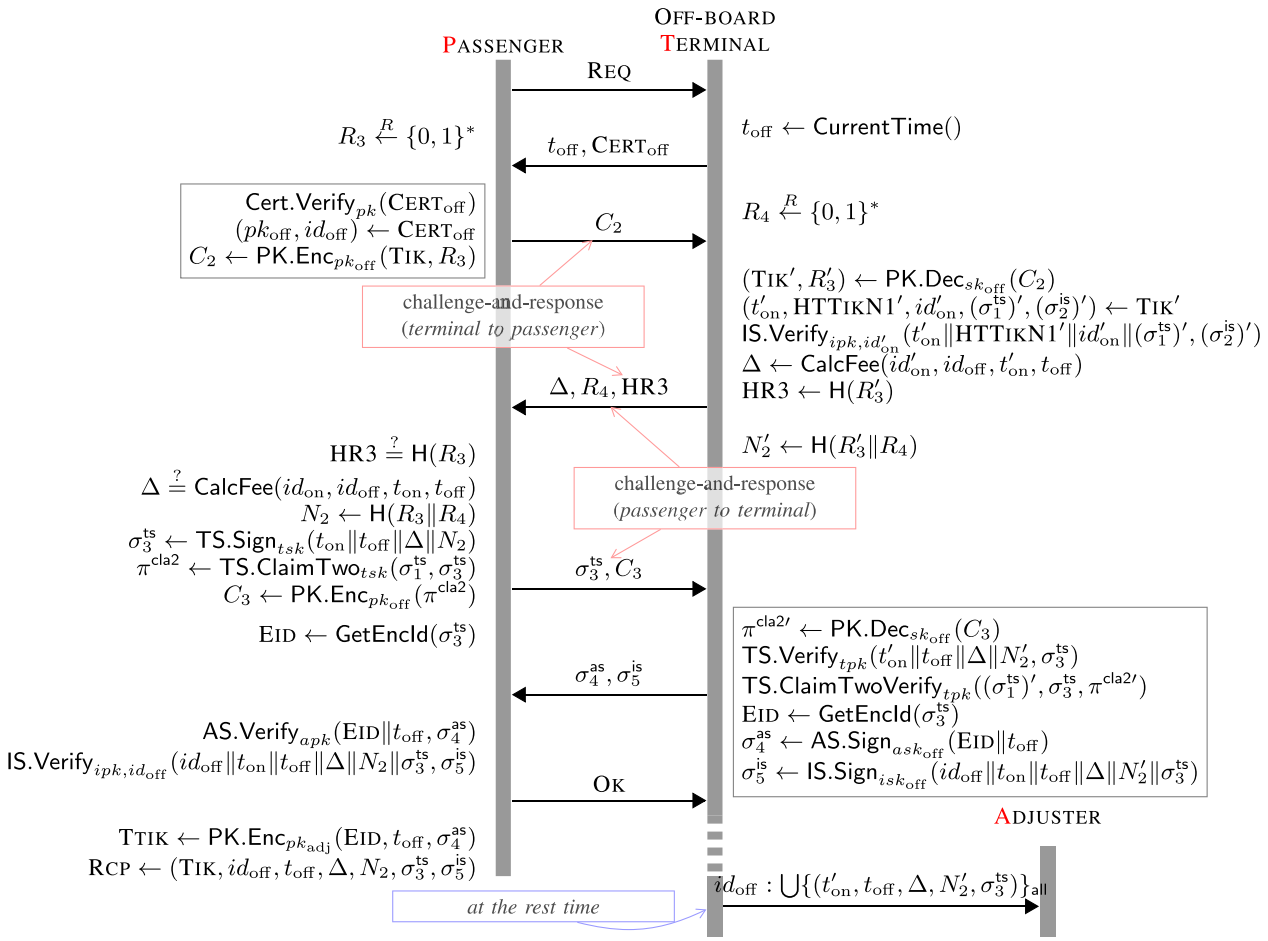


Fig. 3. Communication messages in the exit procedure. After the exit procedure ends, the passenger obtains TIK, and the exit terminal obtains σ_4^{as}. The passenger sends TIK to the next entrance terminal to receive a free transfer.

(e2) Passenger and terminal authenticate each other. To verify whether TML_{on} knows the secret key sk_{on}, PSR encrypts a random number R_1 with TIK to make the ciphertext C_1. If TML_{on} knows sk_{on}, it can successfully decrypt C_1 and obtain TIK$'$ and R_1'. From R_1', TML_{on} consequently computes $N_1' = \text{H}(R_1' \| R_2)$ and $\text{HTTIKN1}' = \text{H}(\text{TIK}' \| N_1')$. Then, TML_{on} signs for HTTIKN1$'$ to generate the identity-based signature σ_2^{is} and returns it to PSR. Note that HTTIKN1$'$ is unpredictable before TML_{on} receives C_1. If σ_2^{is} is valid, PSR can be convinced that TML_{on} knows both sk_{on} and isk_{on}.

On the other hand, PSR should prove to TML_{on} that he or she is a valid passenger. To do this, PSR signs for HTTIKN1 to generate the traceable signature σ_1^{ts} and sends it to TML_{on}. Note that HTTIKN1 is unpredictable before PSR receives R_2. If σ_1^{ts} is valid, TML_{on} can be convinced that PSR knows tsk and can be a valid passenger.

(e3) Adjuster checks revoked passengers. TML_{on} sends TIK$'$ and σ_1^{ts} to the adjuster ADJ in order to check whether or not PSR is revoked. ADJ runs VerifyAndCheck and returns DNY, RVK, or OK ("the signature is not correct," "the passenger was revoked," and "no problem," respectively) to TML_{on} according to the result. Note that this revocation check is the *best effort*. If the check result from the ADJ does not arrive in time, then TML_{on} may process the procedure without waiting for the result.

In the traceable signature scheme that we assume in this paper, CRL-based revocation is not able to be used. Reissue-based revocation is not suitable for a mass transit system because there are a huge number of passengers. However, by adopting the modified linear encryption, our transit payment system solves this problem. ADJ extracts the label Lb_1 from σ_1^{ts} using the link key tlk. Then, using Lb_1, ADJ can search the revocation database in logarithm time at most.

(e4) Terminal issues a ticket to passenger. When TML_{on} generates σ_2^{is}, it signs for not only HTTIKN1$'$, but also σ_1^{ts} in order to prevent PSR from submitting another passenger's traceable signature to the exit terminal TML_{off}. The ticket TIK consists mainly of HTTIKN1, σ_1^{ts}, and σ_2^{is}, so that σ_1^{ts} can be delivered to TML_{off}.

3) Exit Procedure: Figure 3 illustrates the exit procedure in which PSR submits the ticket TIK to terminal TML_{off} and obtains a new transfer ticket.

(x1) Passenger submits ticket. TIK can be obtained after the previous entrance procedure. PSR should hold TIK in a normal case. If not, PSR sets TIK as null, but he or she may pay a penalty. Similar to the entrance procedure, TIK should be encrypted to frustrate eavesdroppers who want to trace PSR. TIK is encrypted using TML_{off}'s public key pk_{off} and is sent to TML_{off}.

(x2) Passenger and terminal authenticate each other. PSR selects R_3 at random, encrypts it with TIK to generate the cipher-text C_2, and sends C_2 to $\mathrm{TML}_{\mathrm{off}}$. If $\mathrm{TML}_{\mathrm{off}}$ knows the secret key sk_{off}, it can decrypt C_2 and return $\mathrm{HR3} = \mathsf{H}(R_3')$. If HR3 is equal to $\mathsf{H}(R_3)$, PSR is convinced of $\mathrm{TML}_{\mathrm{off}}$.

PSR signs for the random number $N_2 = \mathsf{H}(R_3 \| R_4)$, which is unpredictable before he or she receives R_4, to generate σ_3^{ts}. If σ_3^{ts} is valid, $\mathrm{TML}_{\mathrm{off}}$ can be convinced that PSR holds tsk. However, with this, $\mathrm{TML}_{\mathrm{off}}$ is not sure that PSR has really generated $(\sigma_1^{\mathsf{ts}})'$. In ordinary traceable signature schemes, $\mathsf{TS.Claim}$ proves that the signer knows a secret key used for generating a traceable signature. Therefore, $\mathrm{TML}_{\mathrm{off}}$ may ask PSR to individually run $\mathsf{TS.Claim}(\sigma_1^{\mathsf{ts}})$ and $\mathsf{TS.Claim}(\sigma_3^{\mathsf{ts}})$. If we assume that PSR can hold only one signing key tsk, this proves that PSR has employed the same key for generating σ_1^{ts} and σ_3^{ts}. Instead of running $\mathsf{TS.Claim}$ twice, in this paper, we assume $\mathsf{TS.ClaimTwo}$ that proves the signer of both σ_1^{ts} and σ_3^{ts} at the same time. We can easily construct $\mathsf{TS.ClaimTwo}$ by extending $\mathsf{TS.Claim}$ as in Appendix C.

If σ_2^{is} and σ_3^{ts} are valid and the claim proof $\pi^{\mathsf{cla2}} \leftarrow \mathsf{TS.ClaimTwo}_{tsk}(\sigma_1^{\mathsf{ts}}, \sigma_3^{\mathsf{ts}})$ is correct, $\mathrm{TML}_{\mathrm{off}}$ is convinced that PSR is a valid passenger. Note that our payment system omits the verification of $(\sigma_1^{\mathsf{ts}})'$ under the assumption that $\mathrm{TML}_{\mathrm{off}}$ trusts $\mathrm{TML}_{\mathrm{on}}$. Without the assumption, $\mathrm{TML}_{\mathrm{off}}$ should verify $(\sigma_1^{\mathsf{ts}})'$.

(x3) Passenger sends encrypted claim proof to terminal. The claim proof π should be hidden from eavesdroppers who gather $\{(\sigma_1^{\mathsf{ts}})_i\}$ by observing many entrance procedures. If π is disclosed to eavesdroppers, they may figure out the terminal PSR has passed by checking $\mathsf{TS.ClaimTwoVerify}((\sigma_1^{\mathsf{ts}})_i, \sigma_3^{\mathsf{ts}}, \pi^{\mathsf{cla2}}) \stackrel{?}{=} \mathsf{true}$ for all gathered $(\sigma_1^{\mathsf{ts}})_i$. Therefore, π^{cla2} should be encrypted by pk_{off}.

(x4) Terminal and passenger generate transfer ticket. The transfer ticket is submitted to the next entrance terminal in order to receive free transfer service. The transfer ticket does not contain any information about $\mathrm{TML}_{\mathrm{off}}$ but does possess verifiable information about PSR. $\mathrm{TML}_{\mathrm{off}}$ signs for PSR's encrypted identity $\mathrm{EID} = \mathsf{GetEncId}(\sigma_3^{\mathsf{ts}})$ with current time t_{off} to generate an anonymous signature σ_4^{as} and sends σ_4^{as} to PSR. If σ_4^{as} is valid, PSR encrypts EID, t_{off}, and σ_4^{as} using ADJ's public key pk_{adj} and stores the cipher-text TTIK as the transfer ticket.

(x5) Terminal sends transit information to adjuster. At the rest time, $\mathrm{TML}_{\mathrm{off}}$ sends all gathered transit information to ADJ. Each transit information is comprised of t'_{on}, t_{off}, Δ, N_2', and σ_3^{ts}. ADJ extracts a label LB_i by running $\mathsf{TS.GetLabel}_{tlk}((\sigma_3^{\mathsf{ts}})_i)$ for each $(\sigma_3^{\mathsf{ts}})_i$ and categorizes all the transit information into different labels.

C. Adjusting and Charging

Using the link key tlk, the adjuster has the ability to distinguish between all traceable signatures of a certain passenger. Once a passenger who is not revoked uses the proposed system, the adjuster receives $\mathsf{Ent} = \{t_{\mathrm{on}}, \mathrm{TTIK}, N_1, id_{\mathrm{on}}, \sigma_1^{\mathsf{ts}}\}$ from the entrance terminal, whose identity is id_{on}, and receives $\mathsf{Ext} = \{\hat{t}_{\mathrm{on}}, t_{\mathrm{off}}, \Delta, N_2, \sigma_3^{\mathsf{ts}}\}$ from the exit transit agency, which is the transit agency to which the exit terminal belongs. If the passenger uses the proposed system several times, then the adjuster receives messages in the following order:

$$\cdots, [\mathsf{Ent}_{i-1}, \mathsf{Ext}_{i-1}], [\mathsf{Ent}_i, \mathsf{Ext}_i], [\mathsf{Ent}_{i+1}, \mathsf{Ext}_{i+1}], \cdots$$

where i denotes the sequence in which the passenger uses the system.

Note that it should be $\cdots < (t_{\mathrm{on}})_{i-1} < (t_{\mathrm{off}})_{i-1} < (t_{\mathrm{on}})_i < (t_{\mathrm{off}})_i < (t_{\mathrm{on}})_{i+1} < (t_{\mathrm{off}})_{i+1} < \cdots$. Here Δ is calculated without considering the discount for free transfer services. If there is a discount for these services, then the adjuster recalculates Δ.

TTIK_{i+1}, which is encrypted by the adjuster's public key, can be decrypted to $\{\mathrm{EID}, t_{\mathrm{off}}, \sigma_4^{\mathsf{as}}\}$. Normally, we have $\mathrm{TTIK}_{i+1}[t_{\mathrm{off}}] = (t_{\mathrm{off}})_i$ and $\mathrm{TTIK}_{i+1}[\mathrm{EID}] = \mathsf{GetEncId}((\sigma_3^{\mathsf{ts}})_i)$. In other words, these messages are well connected in normal situations. Otherwise, the connection can be broken. That is, the passenger may not process the exit procedure, or the transit agency may not send transit information to the adjuster. We consider the following four cases to recalculate charges (note that $\varnothing$ means that the correspondent message is not available because it is missing):

Case 1: $\cdots, \mathsf{Ext}_i \neq \varnothing], [\mathsf{Ent}_{i+1} \neq \varnothing, \cdots$

If $(t_{\mathrm{on}})_i = (\hat{t}_{\mathrm{on}})_{i+1}$
The passenger reuses an old signature $(\sigma_1^{\mathsf{ts}})_i$. As the penalty for the passenger's misbehavior, the adjuster imposes the maximum charge instead of Δ_{i+1} on the passenger.

Else if $\mathrm{TTIK}_{i+1}[t_{\mathrm{off}}] = (t_{\mathrm{off}})_i \wedge (t_{\mathrm{on}})_i \neq (\hat{t}_{\mathrm{on}})_{i+1} \wedge \mathrm{TTIK}_{i+1}[\mathrm{EID}] = \mathsf{GetEncId}((\sigma_3^{\mathsf{ts}})_i)$
The adjuster can recalculate Δ_{i+1} by considering the discount for free transfer services.

Otherwise
TTIK *is not valid. The passenger may be using the system for the first time or sending invalid* TTIK *information. The adjuster need not recalculate any charge.*

Case 2: $\cdots, \mathsf{Ext}_i = \varnothing], [\mathsf{Ent}_{i+1} \neq \varnothing, \cdots$

If $\mathrm{TTIK}_{i+1}[t_{\mathrm{off}}] > (t_{\mathrm{on}})_i \wedge \mathsf{TS.GetLabelEncId}_{tlk}(\mathrm{TTIK}_{i+1}[\mathrm{EID}]) = \mathsf{TS.GetLabel}_{tlk}((\sigma_3^{\mathsf{ts}})_{i+1}) = \cdots \wedge \mathsf{AS.Verify}_{apk}(\mathrm{TTIK}_{i+1}[\mathrm{EID}] \| \mathrm{TTIK}_{i+1}[t_{\mathrm{off}}], \sigma_4^{\mathsf{as}}) = \mathsf{true}$ *The transit agency does not send $(t_{\mathrm{on}}, t_{\mathrm{off}}, \Delta, N_2, \sigma_3^{\mathsf{ts}})_i$ intentionally. The adjuster imposes the minimum charge for Δ_i on the passenger and can recalculate Δ_{i+1} by considering the discount for free transfer services based on $(\mathrm{TTIK}_{i+1}[t_{\mathrm{off}}], (t_{\mathrm{on}})_{i+1})$.*

Otherwise
The passenger does not process the exit procedure as his or her decision or shares his or her key with other passengers. The adjuster imposes the maximum charge for Δ_i on the passenger.

Case 3: $\cdots, \mathsf{Ext}_i \neq \varnothing], [\mathsf{Ent}_{i+1} = \varnothing, \cdots$

If $\Delta_i > \mathsf{CalcFee}((id_{\text{on}})_i, (id_{\text{off}})_i, (t_{\text{on}})_i, (t_{\text{off}})_i)$

The passenger shares his or her key with other passengers. The adjuster imposes the maximum charge for Δ_i on the passenger.

Otherwise

The entrance terminal does not send Ent_{i+1} to avoid the discount for free transfer services. The adjuster imposes the minimum charge instead of Δ_{i+1} on the passenger.

Case 4: $\cdots, \mathsf{Ext}_i = \varnothing], [\mathsf{Ent}_{i+1} = \varnothing, \cdots$

This case can occur when the entrance terminal does not send Ent_{i+1} after the passenger does not process the exit procedure or the exit transit agency does not sent Ext_i. In this case, the passenger is not charged Δ_i because of the transit agency's misbehavior.

To charge for transit services, the adjuster sends the payment claimant $\{[\mathsf{H}(t_{\text{on}} \| \mathsf{H}(\text{TTIK} \| N_1) \| id_{\text{on}}), \sigma_1^{\mathsf{ts}}]_i, [t_{\text{off}}, \Delta, N_2, \sigma_3^{\mathsf{ts}}]_i\}_{i \in 1..}$ with recalculated $\{\hat{\Delta}_i\}_{i \in 1..}$. From traceable signatures $\{(\sigma_1^{\mathsf{ts}})_i, (\sigma_3^{\mathsf{ts}})_i\}_{i \in 1..}$, the claimant can extract the identity of anonymous passengers by running $\mathsf{TS.Open}_{gmk}((\sigma_1^{\mathsf{ts}})_i)$ or $\mathsf{TS.Open}_{gmk}((\sigma_3^{\mathsf{ts}})_i)$. According to Δ and the identity of the passenger, the claimant charges $\sum \Delta_i$ for transit services provided to the passenger with a real name.

Because the payment claimant cannot know exit transit agencies, it cannot directly distribute charges from the passenger to transit agencies. The payment claimant delivers charges to the adjuster. Different from the payment claimant, the adjuster knows exit transit agencies and thus can fairly distribute charges to entrance and exit transit agencies.

D. Renewing a Link Key

After finishing the adjustment, the payment claimant should renew the link key. Note that the public key tpk includes $(U, V, H) \in \mathbb{G}_1$, the open key tok includes $(\xi_1, \xi_2) \in \mathbb{Z}_q^*$, and $\xi_1 U = \xi_2 V = H$. The link key is comprised of $\ell (= \xi_1/\xi_2)$ and $L (= -\xi_2 P_2)$. The payment claimant reissues its open key as $(\xi_1', \xi_2') \in_R \mathbb{Z}_q^*$ and chooses a new generator P_2' of $\mathbb{G}_2$. It then computes $U' \leftarrow 1/\xi_1 H$, $V' \leftarrow 1/\xi_2 H$, $\ell' \leftarrow \xi_1'/\xi_2'$ and $L' \leftarrow -\xi_2' P_2'$. Finally, the payment claimant publishes (U', V', P_2') to all passengers and the adjuster and assigns $tlk' = (\ell', L')$ to the adjuster.

Let $(T_1, T_2, T_3)_{\mathsf{old}}$ be $(\alpha_1 U, \beta_1 V, A + (\alpha_1 + \beta_1)H)$ and $(T_1, T_2, T_3)_{\mathsf{new}}$ be $(\alpha_2 U', \beta_2 V', A + (\alpha_2 + \beta_2)H)$. The former and the later are the encrypted identities from $\sigma_{\mathsf{old}}^{\mathsf{ts}}$ and $\sigma_{\mathsf{new}}^{\mathsf{ts}}$ of the same passenger before and after renewing a link key, respectively. Then,

$$\mathsf{TS.GetLabel}_{tlk}(\sigma_{\mathsf{old}}^{\mathsf{ts}}) = e(A, P_2) \quad \text{and} \tag{1}$$

$$\mathsf{TS.GetLabel}_{tlk'}(\sigma_{\mathsf{new}}^{\mathsf{ts}}) = e(A, P_2'). \tag{2}$$

In addition,

$$\begin{aligned} &\mathsf{TS.GetLabel}_{tlk}(\sigma_{\mathsf{new}}^{\mathsf{ts}}) \\ &\quad = e(A + \underbrace{(\alpha_2 + \beta_2)H - (\alpha_2 \frac{\xi_1}{\xi_1'} + \beta_2 \frac{\xi_2}{\xi_2'})H}_{\neq O}, P_2) \\ &\quad \neq e(A, P_2) \end{aligned} \tag{3}$$

where O is the infinite zero. In other words, using tlk', the adjuster cannot extract the same label from new traceable signatures.

TABLE III

EXECUTION TIMES FOR THE SHORT TRACEABLE SIGNATURE SCHEME ON SMARTPHONES (MILLISECONDS)

	LG-LU3700	YP-GB1
TS.Sign (pre-computation)	530.43±32.30	252.85±2.95
TS.Sign (running time)	0.70±0.02	0.29±0.01
TS.ClaimTwo (pre-computation)	103.15±7.07	51.25±1.08
TS.ClaimTwo (running time)	0.28±0.02	0.10±0.00

(MNT curve, base field = 201 bits, embedding degree = 6)

VI. PERFORMANCE ANALYSIS

We examine the performance of the proposed system by considering smartphone, terminal, and adjuster servers. We use the short traceable signature scheme [6] for the traceable signature scheme, the RSA with OAEP for the IND-CCA2 secure encryption scheme, Adi Shamir's signature scheme [24] for the identity-based signature scheme, and the direct anonymous attestation (DAA) scheme [4] for the anonymous signature scheme.

In short, passengers can finish each entrance and exit procedure in about 0.3∼0.4 seconds according to the experiment.

A. Smartphones

For the experiments, we used an LG-LU3700 smartphone and a YP-GB1 multimedia player. LG-LU3700 has a 600MHz Qualcomm MSM 7227 (ARM1136EJ-S) CPU with 256 MB RAM and runs on Android OS 2.2. YP-GB1 has the 1GHz Samsung S5PC111 (ARM Cortex-A8) CPU with 512 MB RAM and runs on Android OS 2.2.1.

To implement the short traceable signature scheme, we used the pairing-based cryptography (PBC) library (ver. 0.5.11). For this, we ported the GNU multiple precision arithmetic (GMP) library (ver. 5.0.1) into Android OS though the Android native development kit (NDK). We implemented the identity-based signature scheme using the GMP library. For the PBC library, we used a parameter with an MNT curve (a 201-bit base field and a pairing map with six embedding degrees). For the RSA encryption, we used the 1,024-bit RSA encryption with OAEP-SHA1-MGF1-Padding mode, which is inherently supported by Android OS.

Table III shows the execution times for TS.Sign and TS.ClaimTwo. To generate a short traceable signature and a claim proof, LG-LU3700 spent 0.53 s and 0.19 s, respectively, and YP-GB1, 0.25 s and 0.05 s. Using the PBC library, we suppose that we can enable smartphones to complete entrance and exit procedures within a reasonable period of time even with no precomputation.

Figure 4 shows the execution times for the RSA encryption on smartphones. The upper ▼ means 75% percentile, and the lower ▲ means 25% percentile. Even with differences in computing power, there is little difference in encryption times for one block (86 bytes) between the two smartphones. LG-LU3700 spent an average of 4.7 ms on encrypting, and

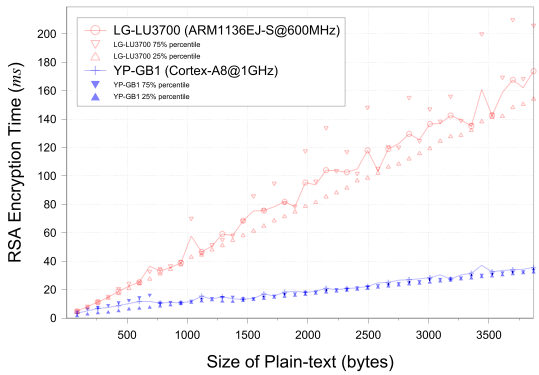


Fig. 4. Encryption times for RSA on smartphones (1024-bit, OAEP-SHA1-MGF1-Padding).

YP-GB1, 3.3 *ms*. We can effectively reduce these time by using a hybrid encryption technique such as RSA-OEAP with AES-GCM instead of a public encryption system if the size of given plain-text is larger than one block.[1] On the other hand, the verification time for a signature is the same as the encryption time for one plaintext block. To verify a public key certificate during entrance and exit procedures (i.e., to run Cert.Verify), the smartphones require 4.7 *ms* (LG-LU3700) and 3.3 *ms* (YP-GB1) regardless of the size of the certificate.

To generate TTIK after completing the exit procedure, the smartphones encrypt more than 2,195 bytes ($|\text{EID}| = 26 \times 3, |t_{\text{off}}| = 14$, and $|\sigma_4^{\text{as}}| \approx 2,103$), taking about 103.6 *ms* (LG-LU3700) and 20.0 *ms* (YP-GB1). The size of TTIK is greater than or equal to 3,328 bytes ($= 128 \times \lceil 2195/86 \rceil$). To generate C_1 in the entrance procedure, the smartphones encrypt more than 3,338 bytes ($|\text{TTIK}| \geq 3,328$ and $|R_1| = 10$). In fact, they spend 135.1 *ms* (LG-LU3700) and 31.5 *ms* (YP-GB1). The size of TIK is 706 bytes ($|t_{\text{on}}| = 14, |\mathsf{H}(\text{TTIK} \| N_1)| = 16, |id_{\text{on}}| = 16, |\sigma_1^{\text{ts}}| \approx 404$, and $|\sigma_2^{\text{is}}| = 256$); therefore, the smartphones encrypt more than 716 bytes to generate C_2 in the exit procedure, spending 35.5 *ms* (LG-LU3700) and 11.0 *ms* (YP-GB1). Similarly, to encrypt π ($\approx$ 306 bytes), the smartphones spend 14.6 *ms* (LG-LU3700) and 7.4 *ms* (YP-GB1).

The verification of the identity-based signature σ_2^{is} requires two exponentiations and one multiplication under the RSA modulus. Both exponents are very small (approximately 17 bits and 160 bits); therefore, the verification process is very fast. LG-LU3700 spends 0.87 *ms*, and YP-GB1, 0.41 *ms*. The verification of the anonymous signature σ_4^{as} requires 16 exponentiations and 12 multiplications. To reduce the number of exponentiations, we implement Shamir's simultaneous squaring multi-exponentiation algorithm, which is also known as Shamir's trick, using the GMP library. Here LG-LU370 spends 259.0 *ms*, and YP-GB1, 115.5 *ms*. Table IV summarizes the results.

B. Adjuster

In the proposed system, VerifyAndCheck in Function 1 needs to reply as quickly as possible. We run VerifyAndCheck on a server composed of the dual 2.0 GHz Intel XEON E5-2620 CPU (24 cores) with 64 GB RAM running on Ubuntu 12.04.4 Server. For VerifyAndCheck, we made a socket-based server daemon using Perl (ver. 5.18.2) because of the simplicity of implementation. In the daemon process, we used the PBC library (ver. 0.5.14), the GMP library (ver. 5.1.3), and the OpenSSL library (ver. 1.0.1) to construct the short traceable signature scheme, the direct anonymous attestation scheme, and the RSA encryption, respectively. The RSA public key encryption scheme used OAEP padding defined PKCS#1 v2.0 with SHA-1 and MGF1. To handle those libraries in Perl, we used the Perl wrappers Crypt::PBC, Math::GMPz, and Crypt::OpenSSL::RSA. To run SearchRVKDB in VerifyAndCheck, we had the daemon process access a MySQL server (ver. 5.5.35) with a revocation list containing hashed labels of 20,000 revoked passengers.

To measure the server's response time, we sent 1,000 queries each for regular and revoked passengers over the gigabit Ethernet link. In addition, we sent 1,000 queries with invalid traceable signatures. Each query was encoded using BSON (specified in MongoDB), and its size was 4,409 bytes. Note that the sizes of RSA cipher-texts or signatures in the experiments are larger than those in an ideal case because they include the management information for the simplicity of implementation. Figure 5 shows the distribution of response times for each case, and Table V shows the execution times for the main functions in VerifyAndCheck.

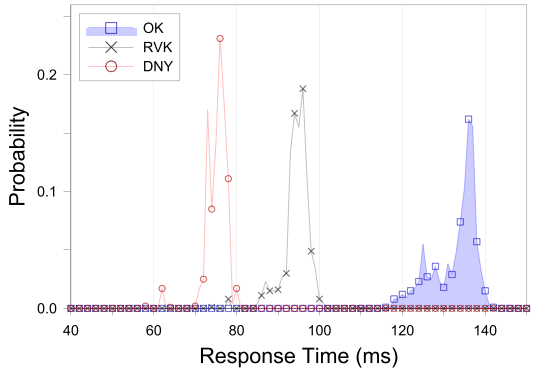
The adjuster replies to DNY in 75.69 ($\pm$2.77) *ms*, RVK in 94.88 ($\pm$3.18) *ms*, and OK in 133.17 ($\pm$5.43) *ms* on average ($\pm$ standard deviation). At least, it does to DNY in 93.93 *ms*, RVK in 101.36 *ms*, and OK in 147.98 *ms*. Therefore, entrance terminals can efficiently block revoked passengers when the terminals wait for replies from the adjuster for a certain period (102 *ms* in our experiment).

TABLE IV
ESTIMATED TIME CONSUMPTION FOR ENTRANCE AND EXIT PROCEDURES ON SMARTPHONES (MILLISECONDS)

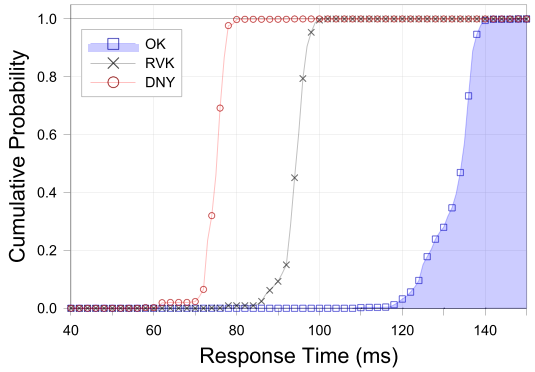
	LG-LU3700	YP-GB1
Entrance Procedure		
Cert.Verify (CERT_{on})	4.7	3.3
PK.Enc (C_1)	135.1	31.5
TS.Sign (σ_1^{ts})	0.7*	0.3*
IS.Verify (σ_2^{is})	0.9	0.4
total	**141.4**	**35.5**
Exit Procedure		
Cert.Verify (CERT_{off})	4.7	3.3
PK.Enc (C_2)	33.0	10.6
TS.Sign (σ_3^{ts})	0.7*	0.3*
TS.ClaimTwo (σ_1^{ts}, σ_3^{ts})	0.3*	0.1*
PK.Enc (C_3)	14.6	7.4
AS.Verify (σ_4^{as})	259.0	115.5
IS.Verify (σ_5^{is})	0.9	0.4
total	**313.2**	**137.6**
After Exit Procedure (rest time)		
PK.Enc (RSA, TTIK)	103.6	20.0

(*: with pre-computation, with ideal size of elements)

[1] In spite of the performance benefit of the hybrid encryption technique, we assume the proposed system uses a public encryption system because of the simplicity of security proof.



(a)



(b)

Fig. 5. Distribution of response times for the adjuster. (a) Response probabilities in certain time durations. (b) Cumulative response times.

TABLE V
EXECUTION TIMES FOR THE ADJUSTER FOR VerifyAndCheck (MILLISECONDS)

		mean time
DNY	Data receiving	1.05±2.96
	BSON decoding	0.36±0.08
	TS.Verify	73.12±1.79
RVK	TS.GetLabel	19.29±0.84
	SearchRVKDB*	0.37±0.07
OK	PK.Dec	9.79±0.73
	AS.Verify	9.72±0.86
	TS.GetLabelEncId	18.31±1.74

(*: without database connection and disconnection)

C. Entrance and Exit Terminals

Current entrance and exit gates such as turnstiles do not have rich computation power. To process the proposed system, however, the gates should provide sufficient computing power. The installation of a processing unit on each gate (i.e., change of all gates) may be undesirable because of economic burden.

We can solve this problem by installing a proxy computer in each terminal. Gates of today have wired communication modules to communicate with other entities such as a control center. By communicating with the proxy computer, the gates can mitigate the computation burden in the entrance and exit procedures.

TABLE VI
EXECUTION TIME FOR ENTRANCE AND EXIT PROCEDURES ON TERMINALS (MILLISECONDS)

		mean time
Entrance	PK.Dec (C_1)	19.81±1.01
	TS.Verify (σ_1^{ts})	60.33±0.66
	IS.Sign (σ_2^{is})	3.36±0.45
Exit	PK.Dec (C_2)	8.18±0.60
	IS.Verify (σ_2^{is})	2.83±0.50
	PK.Dec (C_3)	2.88±0.43
	TS.Verify (σ_3^{ts})	59.70±1.19
	TS.ClaimTwoVerify (σ_1^{ts}, σ_3^{ts})	28.10±1.12
	AS.Sign (σ_4^{as})	14.60±0.93
	IS.Sign (σ_5^{is})	2.01±0.15

TABLE VII
VIRTUAL MACHINES FOR SIMULATING GATES AND PROXY COMPUTER

for		Host Computer	Virtual Machine
Gates	CPU	Intel Core i5-4460 (@ 3.2 GHz)	4 vCPUs
	RAM	8 GB	2 GB
	OS	Windows 8 Enterprise	Ubuntu 15.10 Server
	hyper-visor	VMware workstation 12 player	
Proxy Computer	CPU	Intel XEON E5-2620 CPUs ×2 (@ 2.0 GHz)	8 vCPUs
	RAM	64 GB	16 GB
	OS	Ubuntu 14.04.3 Server	Ubuntu 15.10 Server
	hyper-visor	KVM 2.0.0	

Using the same server settings for the adjuster, we measured the execution time consumed by the proxy computer for entrance and exit procedures. We enforced sleep for 1 second before processing the terminal job, and each function is measured 1,000 times. Table VI shows the results.

Because most of the terminal jobs are signature verification, the proxy computer cannot use a pre-computation technique. In the experiment, the proxy computer consumed about 84 *ms* and 118 *ms* for entrance and exit terminals, respectively. Even in the same computing setting, PK.Dec() for the entrance terminal was slower than that for the adjuster, and TS.Verify() for entrance and exit terminals was faster than that of the adjuster. This is due to the task management by OS; when continuous computation is required, the system is likely to make the CPU's clock faster.

D. Rush Hour

Considering rush hour in the real world, we examined the performance of the proxy computer installed in a terminal. We prepared four virtual machines on four personal computers and another virtual machine on a workstation, as shown in Table VII; the former four virtual machines were for simulating 10 entrance gates and 10 exit gates, and the latter virtual machine was for the proxy computer. All host computers were connected to a gigabit Ethernet switch.

We made client programs and a server daemon using Perl (ver. 5.20.2) with the PBC libary (ver. 0.5.14), the GMP library (ver. 6.0.0a), and the OpenSSL (ver. 1.0.2). Five client

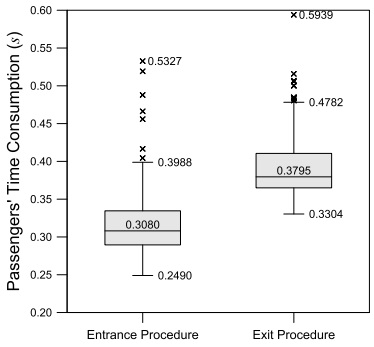


Fig. 6. Passenger time consumption on entrance and exit procedures.

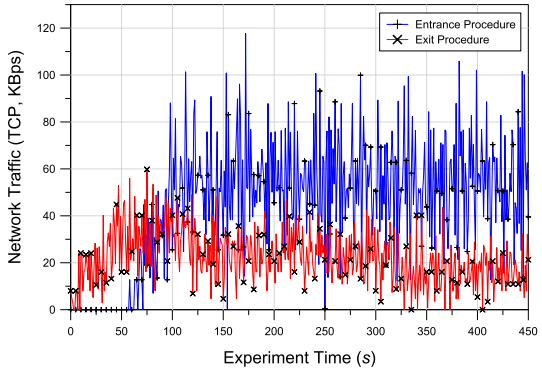


Fig. 7. Network traffic between gates and proxy computer.

programs ran in each virtual machine for gates, and one server daemon with 20 threads ran in the virtual machine for the proxy computer. The server daemon waited 102 *ms* after sending a query to the adjuster. For the real workload to the proxy computer, client programs computed appropriate messages and waited reasonable time durations based on the former experiment results for YP-GB1 before sending messages. Considering passengers' behaviors in the real world, client programs waited 2 seconds on average after finishing the previous procedure. Every message between client programs and a server daemon was encoded using BSON. We performed the experiment for 450 seconds.

Figure 6 shows passenger time spent in entrance and exit procedures. According to the experiment, passengers could finish their entrance procedure in 532.7 *ms* and their exit procedure in 593.9 *ms* at least. Normally, they could finish entrance and exit procedures in about 0.3∼0.4 seconds, which, we believe, is normally acceptable. Figure 7 shows the network traffic between gates and the proxy computer in seconds. Since we manually started client programs, the network traffic progressively increased until 100 seconds. During the experiment, the maximum network traffic consumed by our protocol reached only 142.340 KBps, which is quite small considering that today's wireline LAN speed supports 10Gbps. From 100 to 450 seconds, the proxy computer processed 1495 entrance passengers and 931 exit passengers.

VII. Conclusion

We proposed a privacy-preserving transit payment system with many interesting properties. In the proposed system, transit agencies cannot obtain passengers identities, and the payment claimants cannot obtain passenger routes. The proposed system is robust to the misbehavior of passengers and transit agencies. Our system supports free transfer services with postpaid programs by allowing for mobile payments using smartphones. Results of the experiments showed that the proposed system can be deployed immediately through existing mobile devices. The passengers can complete their entrance and exit procedures in about 0.3∼0.4 seconds, including revocation check, which took about 0.1 seconds.

Though the proposed system is designed for mass transit services such as metro or subway systems, it can be applied to off-line mobile payment systems supporting privacy by adequately merging the entrance and exit procedures into one procedure. We leave these works for future investigation.

References

[1] G. Ateniese, D. Song, and G. Tsudik, "Quasi-efficient revocation of group signatures," IACR Cryptol. ePrint Arch., Tech. Rep. 2001:101, 2001. [Online]. Available: http://eprint.iacr.org/2001/101

[2] J. Balasch, A. Rial, C. Troncoso, B. Preneel, I. Verbauwhede, and C. Geuens, "PrETP: Privacy-preserving electronic toll pricing," in *Proc. USENIX Secur. Symp.*, 2010, pp. 63–78.

[3] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 3152, M. K. Franklin, Ed. Berlin, Germany: Springer, 2004, pp. 41–55.

[4] E. Brickell, J. Camenisch, and L. Chen, "Direct anonymous attestation," in *Proc. 11th ACM Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2004, pp. 132–145.

[5] X. Chen, G. Lenzini, S. Mauw, and J. Pang, "A group signature based electronic toll pricing system," in *Proc. 7th Int. Conf. Availability, Rel. Secur. (ARES)*, Aug. 2012, pp. 85–93.

[6] S. G. Choi, K. Park, and M. Yung, "Short traceable signatures based on bilinear pairings," in *Advances in Information and Computer Security* (Lecture Notes in Computer Science), vol. 4266, H. Yoshiura, K. Sakurai, K. Rannenberg, Y. Murayama, and S. Kawamura, Eds. Berlin, Germany: Springer, 2006, pp. 88–103.

[7] W. de Jonge and B. Jacobs, "Privacy-friendly electronic traffic pricing via commits," in *Formal Aspects in Security and Trust* (Lecture Notes in Computer Science), P. Degano, J. Guttman, and F. Martinelli, Eds. Berlin, Germany: Springer, 2008, pp. 143–161.

[8] T. El Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Proc. CRYPTO*, 1984, pp. 10–18.

[9] L. Gao, M. Ma, Y. Shu, and C. Liu, "RFID security protocol trace attack and desynchronizing attack deep research," in *Proc. Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, vol. 2. Dec. 2011, pp. 918–922.

[10] S. Gisdakis, V. Manolopoulos, S. Tao, A. Rusu, and P. Papadimitratos, "Secure and privacy-preserving smartphone-based traffic information systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1428–1438, Jun. 2015.

[11] J. Guo, J. P. Baugh, and S. Wang, "A group signature based secure and privacy-preserving vehicular communication framework," in *Proc. Mobile Netw. Veh. Environ.*, May 2007, pp. 103–108.

[12] J. C. Herrera, D. B. Work, R. Herring, X. J. Ban, and A. M. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: The mobile century field experiment," *Transp. Res. C*, vol. 18, pp. 568–583, 2010.

[13] J. Y. Hwang, S. Lee, B.-H. Chung, H. S. Cho, and D. Nyang, "Group signatures with controllable linkability for dynamic membership," *Inf. Sci.*, vol. 222, pp. 761–778, Feb. 2013.

[14] J. T. Isaac and S. Zeadally, "An anonymous secure payment protocol in a payment gateway centric model," *Procedia Comput. Sci.*, vol. 10, pp. 758–765, 2012.

[15] D. Konidala *et al.*, "Resuscitating privacy-preserving mobile payment with customer in complete control," *Pers. Ubiquitous Comput.*, vol. 16, no. 6, pp. 643–654, Aug. 2012.

[16] B. Libert, T. Peters, and M. Yung, "Group signatures with almost-for-free revocation," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 7417, R. Safavi-Naini and R. Canetti, Eds. Berlin, Germany: Springer, 2012, pp. 571–589.

[17] B. Libert, T. Peters, and M. Yung, "Scalable group signatures with revocation," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 7237, D. Pointcheval and T. Johansson, Eds. Berlin, Germany: Springer, 2012, pp. 609–627.
[18] 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.
[19] T. Makita, Y. Manabe, and T. Okamoto, "Short group signatures with efficient flexible join," in *Proc. Symp. Cryptogr. Inf. Secur. (SCIS)*, 2006.
[20] S. Meiklejohn, K. Mowery, S. Checkoway, and H. Shacham, "The phantom tollbooth: Privacy-preserving electronic toll collection in the presence of driver collusion," in *Proc. USENIX Secur. Symp.*, 2011, p. 32.
[21] M. Ohkubo, K. Suzuki, and S. Kinoshita, "RFID privacy issues and technical challenges," *Commun. ACM*, vol. 48, no. 9, pp. 66–71, Sep. 2005.
[22] R. A. Popa, H. Balakrishnan, and A. J. Blumberg, "VPriv: Protecting privacy in location-based vehicular services," in *Proc. USENIX Secur. Symp.*, 2009, pp. 335–350.
[23] S. E. Sarma, S. A. Weis, and D. W. Engels, "RFID systems and security and privacy implications," in *Cryptographic Hardware and Embedded Systems—CHES* (Lecture Notes in Computer Science), vol. 2523, B. S. Kaliski, Ç. K. Koç, and C. Paar, Eds. Berlin, Germany: Springer, 2002, pp. 454–469.
[24] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*. New York, NY, USA: Springer-Verlag, 1985, pp. 47–53.
[25] C. Troncoso, G. Danezis, E. Kosta, J. Balasch, and B. Preneel, "PriPAYD: Privacy-friendly pay-as-you-drive insurance," *IEEE Trans. Depend. Sec. Comput.*, vol. 8, no. 5, pp. 742–755, Sep./Oct. 2011.
[26] C. Troncoso, G. Danezis, E. Kosta, and B. Preneel, "PriPAYD: Privacy friendly pay-as-you-drive insurance," in *Proc. ACM Workshop Privacy Electron. Soc. (WPES)*, 2007, pp. 99–107.
[27] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," in *Security in Pervasive Computing* (Lecture Notes in Computer Science), vol. 2802, D. Hutter, G. Müller, W. Stephan, and M. Ullmann, Eds. Berlin, Germany: Springer, 2003, pp. 201–212.
[28] Q. Wu, J. Domingo-Ferrer, and U. González-Nicolás, "Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications," *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 559–573, Feb. 2010.
[29] J.-H. Yang and P.-Y. Lin, "A mobile payment mechanism with anonymity for cloud computing," *J. Syst. Softw.*, vol. 116, pp. 69–74, Jun. 2016.

Jeonil Kang (M'14) received the B.S. degree in computer engineering, and the M.S. and Ph.D. degrees in information and telecommunication engineering from Inha University, South Korea, in 2003, 2006, and 2014, respectively. He is currently a Post-Doctoral Researcher with the Information Security Research Laboratory, Inha University. His research interests lie in cryptography, network security, authentication, privacy, and security issues of Internet of Things and biometrics.

DaeHun Nyang (M'03) received the B.Eng. degree in electronic engineering from Korea Advanced Institute of Science and Technology in 1994, and the M.S. and Ph.D. degrees in computer science from Yonsei University, South Korea, in 1996 and 2000, respectively. He was a Senior Member of the Engineering Staff with the Electronics and Telecommunications Research Institute, South Korea, from 2000 to 2003. Since 2003, he has been a Full Professor with the Computer Information Engineering Department, Inha University, South Korea, where he is currently the Founding Director of the Information Security Research Laboratory. His research interests include cryptography and network security, privacy, usable security, biometrics, and their applications to authentication and public key cryptography. He is a member of the Board of Directors and Editorial Board of the Korean Institute of Information Security and Cryptology.