Privacy-Preserving Public Transport Ticketing System

Milica Milutinovic, Koen Decroix, Vincent Naessens, Bart De Decker

Privacy-Preserving Public Transport Ticketing System

Milica Milutinovic[1], Koen Decroix[2], Vincent Naessens[2] and Bart De Decker[1]

[1] KU Leuven, Dept. of Computer Science, iMinds/DistriNet,
[2] KU Leuven, TC Ghent, Dept. of Computer Science, MSec, iMinds/DistriNet,
firstname.lastname@cs.kuleuven.be

Abstract. The public transport ticketing systems are undergoing significant changes in recent years. The tickets can now be issued and presented in digital form, significantly improving the user experience. The digital data is also used to improve the services' efficiency. Travelling patterns and route occupancy can be analysed to adjust the frequency and coverage of the service. However, data recorded by the providers extends the information that is needed for simple analysis. The travel passes that are issued usually contain unique identifiers, allowing to trace the movement of users, which can even be linked to their identities. In order to tackle these privacy issues, we propose a novel, privacy-preserving ticketing system, based on a scheme for issuing and redemption of unlinkable certified tokens. The design also allows offering advanced services, such as reduction plans or monthly passes, without introducing privacy concerns. Even though the travellers' actions cannot be linked, the service providers are given assurances against possible misuse, and are able to control the usage of the issued products. Additionally, experimental evaluation shows that the system performance is adequate for practical applications.

1 Introduction

The public transport systems have a four centuries long history and have undergone a considerable evolution. However, the corresponding ticketing systems have only recently started to experience significant changes. The early systems were based on paper tickets, usually bought on site. With the technological advances, the travellers are now able to verify the seats availability, reserve or purchase a ticket online and present it in a digital form on their smartphone or tablet. Digital handling of travellers' data also allows the transportation providers to better inspect the travelling patterns and accordingly adjust the provided services. Examples are the organisation of transportation during rush hours or to less frequented areas. This improves the transportation network's efficiency and lowers the cost for the customers.

However, the way these novel ticketing systems are deployed in practice is creating significant privacy concerns. Namely, the issued travelling passes usually contain unique identifiers, which are in most systems linked to the identities of their holders. This allows to make behaviour profiles, which is detrimental to

the travellers' privacy. Even when the user identity is not given, powerful data mining techniques can be utilised to determine who is the holder of a pass. The users have little or no control over their data in such systems. Moreover, with some implementations, not only can the provider see the behaviour or identity of the traveller, but an external party can mount a successful attack and obtain this data as well. A notable example of a successful attack on a public transport ticketing system's technology is the attack on the MIFARE Classic card [8].

Contributions. In order to tackle the aforementioned privacy concerns, we have devised a privacy-preserving ticketing system (PPTS). It is based on a scheme for unlinkable certified tokens and utilises cryptographic primitives, such as commitment schemes [16], partially blind signatures [1] and anonymous credentials [6] and is designed for smartphone technology. The resulting system allows the travellers to manage their personal data and purchased travel products, issued in the form of digital tokens. The products are linked to the purchasing traveller and cannot be transferred to another entity without the approval of the provider. The provider is able to control the products' spending and verify the validity of the traveller's registration, but cannot obtain the identity of the traveller or link together different interactions performed with the same traveller. Finally, efficiency improvements of the offered services are still possible, as the service provider can see the routes' occupancy and usage of specific products.

This paper is organised as follows: Section 2 describes the existing systems, their architecture and the attacker model. Section 3 details the proposed scheme and the underlying protocols with Section 4 evaluating protocols' security, privacy and efficiency and presenting possible design extensions. Finally, Section 5 discusses the related work and concluding remarks are given in Section 6.

2 Public Transport Ticketing Systems

The novel commercial ticketing systems are usually based on a *Personal Transport Pass*. It represents a pass linked to a particular traveller, which cannot be utilised by other individuals. In order to achieve this property, the pass records personal identifiable information in addition to the transport products it stores. Some of the personal data is printed on the cover of the pass and some is stored in its embedded microchip. The chip can only be accessed by the ticketing system equipment. It can store the unique chip number, ePurse balance (which is reduced with every trip the traveller takes), travel transactions history (such as validations of entering or leaving a vehicle) and personal transport products (such as a monthly discount plan). The cover usually contains the traveller's name, passport picture, date of birth and gender. This personal identifiable information is visible to everyone with visual access to the card. It is also used by the inspectors who verify whether the pass belongs to the traveller.

When a pass is issued, the user provides the recorded personal details. With online pass ordering, some of the personal data is not verified. However, as the pass is delivered to the given address, the provider is assured about the validity of this attribute. When the traveller utilises the issued pass, a random inspection

may request to check its validity. The picture and possibly other information can then be verified and the pass can be revoked in case of irregularities.

For privacy-aware users, *Anonymous Transport Passes* can be offered as well. In contrast to the personalised version, the anonymous pass can be shared with other travellers, as it contains no personal data. However, as a consequence, no personal transport products can be loaded on the pass and the range of provided services is limited. Even though such card is anonymous, it can still record the *travel transactions history*. This can lead to traveller profiling and raises serious privacy concerns as it allows to track movements and even link them to the identity of the traveller by using powerful data mining techniques.

2.1 System architecture

The ticketing service is offered through collaboration of multiple stakeholders. They usually comprise the *ticketing system operator* (TSO), who issues travellers' passes and handles the related interactions, and the *public transport organisers* (PTO), who organize the actual public transport. One TSO usually collaborates with multiple PTO entities. It manages the personal information obtained in the registration procedures and the identifiers recorded in the passes. The PTO is able to record the trips taken and the disclosed data, such as the unique pass identifier [20, 5]. The existing systems collect this travel data in order to optimize the provided services. However, recording unique identifiers, such as smart card serial numbers, is a major privacy concern. It allows for creating profiles and possibly linking them with registration information, including traveller's personal identifiable information. Some solutions for mitigating privacy concerns [17] rely on corporate level policies that separate travel transactions from user data and restrict access to only one of the databases. However, the privacy depends on the discipline inside the organization and may be prone to internal or external attacks which could link identifiable data to travel patterns.

2.2 Attacker Model

A public transport ticketing system needs to be resilient to the following attacks:

- Attacks on user privacy mounted by TSO and PTO entities. They can try to link the user activities, such as purchasing and utilising the travel products, and possibly even link them to the user identities.
- Attacks on user privacy by external entities. External parties may try reading the travel passes or intercepting the ongoing interaction with the provider.
- Inappropriate charging of users for the provided services.
- Unauthorised usage of the transport services. These attacks are carried out by travellers who try to make use of the service without obtaining the required authorisations. This can refer to unregistered entities using the transport services or registered travellers who utilise spent or expired tickets, tickets of other travellers or simply have an insufficient ePurse balance or do not possess the right travel product.

2.3 Cryptographic building blocks

This section provides an overview of the cryptographic building blocks used in the proposed protocols.

Commitment schemes allow an entity to commit to a set of values while keeping them secret. They can be compared to sealed, non-transparent envelopes. When a commitment is disclosed, the user cannot change the values she committed to, without this being detectable by the verifier. The commitment hides the chosen values, while allowing to prove certain properties about them. For committing to a value, we assume usage of the Pedersen commitment scheme [16]. Thus, for a group G of prime order q and generators g_1 and $g_2 \in G$, the user commits to a message m, by choosing a random value r and computing the commitment:

$C = g_1^m \times g_2^r$.

Partially blind signature schemes allow to sign a data structure, parts of which are not disclosed to the signer. They are an extension of the blind signatures concept, where the contents of the signed message are hidden from the signer [7]. In the partially blind signature scheme (PBS), the signer (S) and the receiver (R) agree on some *public* information, that is also included in the signature [1]:

$pbsig \leftarrow \{PBsign(SK_S; hidden_R; public)\}$.

Zero-knowledge proofs of knowledge (ZKPK) allow one party to prove that she knows certain values or secrets [3]. Namely, a prover can convince a verifier that a certain statement is true, without revealing any additional information. In the proposed protocols, we utilise a non-interactive signature proof of knowledge, which additionally allows the prover to sign a message when creating the proof of knowledge [6]. For a public value y and a private value x, such that $y = g^x$, and for a message m, we denote the signed proof of knowledge of x with:

$SPK\{(x) : y = g^x\}(m)$.

Anonymous credentials allow for authentication of users in a privacy-protecting manner [6, 4]. This credential technology offers selective disclosure of attributes, i.e. disclosing only a part of the recorded attributes, while hiding the others. Additionally, it is possible to only prove properties of the embedded attributes, without disclosing the actual values. For instance, it is possible to prove that the holder of the credential is older than 18, without revealing the birthdate embedded in the credential. Possession of a valid credential and properties of the recorded attributes are proven with ZKPK. They additionally allow to prove equality of a value hidden in a commitment to a value contained in the credential. In the remainder of the text, we will assume the usage of Idemix credentials [6], as they allow unlinkable use of the same credential.

3 Privacy-Preserving Ticketing System

Similarly to existing systems, the proposed scheme consists of interactions between a traveller, a ticketing system operator (TSO) and a public transport or-

ganiser (PTO). Before participating in the PPTS system, every traveller needs to install a smartphone application, which interfaces with the ticketing system. All travellers are also issued with credentials which serve as personal passes. Before utilising the public transport services, a traveller makes a purchase of the desired products, such as single tickets or monthly passes. Even though the products are linked to the traveller's credential in order to prevent unauthorised sharing, no identifying data is disclosed at the time of purchase. In order to use the transport services, a traveller's application contacts the TSO to be issued with a single-use ticket for the desired ride. For this, the application spends or proves possession of a previously purchased product and demonstrates that it is linked to the credential the traveller owns. The acquired temporary ticket is validated by the PTO's validation machine on the vehicle. At the end of the journey, the traveller's phone interacts with the validation machine once more to be issued with change in case the spent product is not fully used. The obtained change proof can then be exchanged with TSO for a long-term token. Although there are multiple interactions with TSO/PTO, they are mostly transparent to the traveller, as she only initiates the ticket issuance and travel start/end, while all the other operations are automatically performed by the application.

3.1 Traveller Credential Issuance

After installing the PPTS application, a traveller interacts with the online ticketing application, to be issued with an anonymous credential which serves as a personalised pass and is denoted as the PPTS credential. Credential attributes include the traveller's personal information (such as the name and date of birth), the validity information and a secret number - the traveller's *ticketing system secret* (tss), which is different from the credential's master secret and is not disclosed to the TSO[3]. The personal information is provided by the traveller, as in the online registrations in the existing systems. In case additional assurances are needed, the traveller's eID card can be used for proving this data. In systems where the smart cards are delivered to the traveller's home address, there is additional confirmation of the provided address information. This can be offered with the PPTS, by sending a code via post, which is used upon reception to complete the registration and credential issuance. On the whole, this approach improves the efficiency, while providing the same guarantees as the currently issued passes in (Section 2). Idemix credential technology [6] allows for all subsequent interactions with the TSO to remain unlinkable to the credential issuance. Additionally, for a better privacy-protection we assume that the network layer meta data does not allow linking activities of the same user.

3.2 ePurse Balance Recharge

In order to recharge the ePurse balance, the traveller makes a request and pays the desired amount to the TSO. In return, the TSO issues a number of signed

[3] The user sends a commitment to a random number to the TSO, which applies a random offset to it before including it in the credential, resulting in the *tss* attribute.

tokens to the traveller. The tokens represent a partially blind signature on a public part, i.e. the information on the token value, and a private part, which is not disclosed to the TSO. The public information in a token represents its denomination and possibly constraints that apply (such as a validity period). The hidden part of the signed data in a token is a commitment to the secret value in the traveller's PPTS credential (tss). It serves as a link to the traveller's credential and prevents transferring tokens to other travellers.

Protocol 1: ePurse recharge.

$\mathsf{ePurseRecharge}(amount)$

(1)	$T_{app} \rightarrow TSO : \mathsf{request}(amount)$
(2)	$T_{app} \leftarrow TSO : c \in_{\mathcal{R}} \mathbb{Z}_q$
(3)	$T_{app} \rightarrow TSO : \pi \leftarrow SPK\{(MS) : Cred.validity\}(c)$
(4)	$TSO : \mathsf{if}\ (!\mathsf{verify}(\pi))\ \mathsf{abort}$
(5)	$T_{app} \leftrightarrow TSO : \mathsf{pay}_{\mathsf{bank}}(invoice : \{reference, amount, account_{TSO}\})$
(6)	$T_{app} \leftarrow TSO : \{info : D_g\}_{i \in \{1 \ldots n\}} \leftarrow \mathsf{generate}(amount)$
(7)	$T_{app} : \{r\}_{i \in \{1 \ldots n\}} \leftarrow \mathsf{generate}()$
(8)	$T_{app} : \{C\}_{i \in \{1 \ldots n\}} \leftarrow \mathsf{commit}(\{r\}_{i \in \{1 \ldots n\}}, Cred.tss)$
(9)	$T_{app} \leftrightarrow TSO : \{pbsig\}_{i \in \{1 \ldots n\}} \leftarrow \mathsf{PBSign}(SK_{TSO}; \{C\}_{i \in \{1 \ldots n\}}; \{info\}_{i \in \{1 \ldots n\}})$
(10)	$T_{app} : \mathsf{store}(\{eToken : \{C, r, info, pbsig\}\}_{i \in \{1 \ldots n\}})$

The detailed interaction for recharging the ePurse is illustrated with Protocol 1. The traveller initially requests a recharge via the installed mobile application, T_{app} *(step 1)*. The request contains only the requested amount to be added to the ePurse. In order to be granted the recharge, T_{app} also proves that the traveller holds a valid PPTS credential, *Cred*, by creating a zero-knowledge proof of knowledge, π, to a fresh challenge c received from the TSO *(2)*. Proof creation requires knowledge of the credential master secret *MS*, which is known only to the credential holder. In case of successful proof verification, the TSO replies with an adequate invoice. The traveller then performs the online payment *(5)* through a third-party payment service provider. After a successful payment, the TSO is notified and generates the public details for the tokens to be issued *(6)*. These details contain the information on the denomination of every token (D_g), which add up to the requested recharge amount. The D_g value is the guaranteed amount for one journey, i.e. the amount that a traveller needs to hold in order to be allowed to start a journey. It usually corresponds to the maximal charge for a ride. This way, the service provider is assured that the full trip fee will be paid, because the traveller spends one token with the guaranteed amount when entering the vehicle and is reissued with the change if a cheaper ride was taken. The T_{app} then creates a fresh commitment to the tss secret from the traveller's credential for each of the n tokens to be issued *(7–8)*. This way, the tokens issued to the same tss will not be linkable in the spending phase (see Section 3.4). In an interaction with the T_{app}, the TSO signs the public details and the *hidden* commitment with the partially blind signature scheme *(9)*. For sharing preven-

tion, the blinded commitments are signed only after verifying that they belong to the credential holder (more details on the properties of such unlinkable tokens can be found in [15]). In the final step, the traveller's application stores the signatures, the commitments and opening information with the tokens' details. These *eToken* structures represent the recharge of the ePurse.

3.3 Purchase of Travel Products

The proposed system also allows for the PTO providers to offer multiple-use products, such as monthly discounts, which is a service already present in current ticketing systems (e.g. the Dutch U-OV bus or NS railway services). The interaction is detailed in Protocol 2.

Protocol 2: Purchase of multiple-use products

purchaseTransportProduct($monthlyDiscount$)

(1) $T_{app} \rightarrow PTO$: request($monthlyDiscount$)
(2) $T_{app} \leftarrow PTO$: $c \in_{\mathcal{R}} \mathbb{Z}_q$
(3) $T_{app} \rightarrow PTO$: $\pi \leftarrow SPK\{(MS, dob) : Cred.validity \wedge 12 \leq \mathsf{age}(Cred.dob) \leq 18\}(c)$
(4) PTO : if (!verify(π)) abort
(5) $T_{app} \leftrightarrow PTO$: $\mathsf{pay}_{\mathsf{bank}}(invoice : \{reference, amount, account_{TSO}\})$
(6) $T_{app} \leftarrow PTO$: $info_{tp} : \{reduction, PTO, validity\} \leftarrow$ generate()
(7) T_{app} : $\{r_{tp}\}_{i \in \{1...n\}} \leftarrow$ generate()
(8) T_{app} : $\{C_{tp}\}_{i \in \{1...n\}} \leftarrow$ commit($\{r_{tp}\}_{i \in \{1...n\}}, Cred.tss$)
(9) $T_{app} \leftrightarrow PTO$: $\{pbsig_{tp}\}_{i \in \{1...n\}} \leftarrow$ PBSign($SK_{PTO}; \{C_{tp}\}_{i \in \{1...n\}}; info_{tp}$)
(10) T_{app} : store($\{eToken : \{C_{tp}, r_{tp}, info_{tp}, pbsig_{tp}\}\}_{i \in \{1...n\}}$)

The traveller initially requests a multiple-use product via his mobile application *(step 1)*. The request only records the kind of product that is requested. The traveller also proves ownership of a valid PPTS credential, *Cred*. It may additionally be required to prove certain properties, such as the age group, using the date of birth (*dob*) attribute of the credential *(step 3)*. The proof is provided in the form of a signed zero-knowledge proof (SPK) and it only convinces the provider that the attributes in the traveller's credential satisfy the given properties, and hide other information recorded in the credential. If the proof verifies, the PTO can generate an appropriate invoice for the traveller. The traveller makes the payment via a third-party payment service provider *(5)*. Similarly to the ePurse recharge, the requested product is issued in the form of partially blind signatures on the public *info* and fresh commitments to the *tss* secret (*6-9*). The *info* is the same in all tokens and records the product specification, such as validity, type and issuing PTO[4]. These tokens are spent with the TSO before starting the journeys (Protocol 3). Until the limitation on these tokens (such

[4] The *info* is assumed to have a limited set of possible values, as unique values would allow to link different interactions with the same traveller.

as validity date) is met, the application interacts with the PTO before the last token is used; the last token can be spent with the TSO to obtain a new bundle of signature tokens on fresh commitments.

The system also allows for purchases of single-use products, such as discounted tickets for a specific journey, vouchers for carying a bike on the train, or tickets for a pet. Similarly to multi-use products, the single-use products are issued as *eToken* structures with *info* representing the product description.

3.4 Validation of the Trip Start

For using the PTO service and charging the ePurse, the T_{app} interacts with the TSO to spend an *eToken* and obtain a single-use ticket. This ticket is sent to the PTO's validation machine over an NFC channel upon entering the bus, where it is verified. The interaction is detailed in Protocol 3.

Protocol 3: Validation of the trip start using the ePurse.

validateStartTrip($ePurse$)

		Before boarding the vehicle:
(1)	T_{app}	: $\{C, r, info : D_g, pbsig\} \leftarrow \mathsf{load}(eToken)$
(2)	T_{app}	: $\{C_{tp}, r_{tp}, info_{tp} : \{reduction, PTO, validity\}, pbsig_{tp}\}$ $\leftarrow \mathsf{load}(eToken_{tp})$
(3)	$T_{app} \rightarrow TSO$	: $\mathsf{request}(startTrip)$
(4)	$T_{app} \leftarrow TSO$	: $c_{start} \in_{\mathcal{R}} \mathbb{Z}_q$
(5)	T_{app}	: $\pi \leftarrow SPK\{(MS, r, r_{tp}, tss) :$ $Cred.validity \ \wedge C = g^{tss} \times h^r \ \wedge$ $C.tss = Cred.tss \ \wedge C_{tp} = g^{tss} \times h^{r_{tp}} \ \wedge$ $C_{tp}.tss = Cred.tss\}(c_{start})$
(6)	$T_{app} \rightarrow TSO$	: $loc_{start}, C, info, pbsig, C_{tp}, info_{tp}, pbsig_{tp}, \pi$
(7)	TSO	: if $(\mathsf{seen}(C, info) \vee \mathsf{seen}(C_{tp}, info_{tp}) \vee \mathsf{!verify}(PK_{TSO}; pbsig, C, info) \vee$ $\mathsf{!verify}(PK_{PTO}; pbsig_{tp}, C_{tp}, info_{tp}) \vee \mathsf{!verify}(\pi, c_{start}))$ abort
(8)	TSO	: $t_{val} \leftarrow \mathsf{generate}()$
(9)	TSO	: $sig_{val} \leftarrow \mathsf{sign}(SK_{TSO}; \{c_{start}, t_{val}, loc_{start}, info_{tp}\})$
(10)	TSO	: $\mathsf{store}(c_{start}, C, info, C_{tp}, info_{tp}, t_{val})$
(11)	$T_{app} \leftarrow TSO$	: sig_{val}, t_{val}
(12)	T_{app}	: $\mathsf{store}(sig_{val}, c_{start}, t_{val}, loc_{start}, info_{tp}), \mathsf{delete}(eToken, eToken_{tp})$
		On entering the vehicle:
(13)	$T_{app} \rightarrow PTO_{VM}$	: $sig_{val}, c_{start}, t_{val}, loc_{start}, info_{tp}$
(14)	PTO_{VM}	: if $(\mathsf{!verify}(PK_{TSO}; sig_{val}, c_{start}, t_{val}, loc_{start}, info_{tp}) \vee$ $\mathsf{!verify}(loc_{start}) \ \vee \mathsf{!verify}(t_{val}) \ \vee \mathsf{!verify}(c_{start} \leftrightarrow t_{val}))$ abort
(15)	PTO_{VM}	: $\{t_{start}, id_{bus}\} \leftarrow \mathsf{generate}()$
(16)	PTO_{VM}	: $sig_{start} \leftarrow \mathsf{sign}(SK_{PTO}; \{c_{start}, t_{start}, loc_{start}, id_{bus}, info_{tp}\})$
(17)	$T_{app} \leftarrow PTO_{VM}$	: $val_{start} : \{sig_{start}, c_{start}, t_{start}, loc_{start}, id_{bus}, info_{tp}\}$
(18)	T_{app}	: $\mathsf{store}(val_{start})$

Before taking the ride, the traveller performs certain precomputations and interacts with the TSO application server online. It spends an *eToken* represent-

ing the required guarantee amount and an applicable discount. This way, the efficiency of the protocol is improved and the time needed for the interaction between the traveller's application T_{app} and the validation machine PTO_{VM}, is reduced. The T_{app} initially loads the tokens and requests a temporary ticket from the TSO *(steps 1-3)*. Next, the *eToken* is spent with the TSO's application server *(steps 4-6)*, by showing the partially blind signatures, commitments, *info* descriptions, and providing a zero-knowledge proof π on a received challenge c_{start}. It proves ownership and validity of a PPTS credential and the fact that the commitments are created with the secret of the same credential. The application also sends the starting location of the trip, loc_{start}. The TSO checks if the same commitments were used before and verifies the signatures, validity information (recorded in the *info*) and the SPK *(7)*. It records the received commitment and *info*, to prevent the double spending of the token in future transactions *(10)*. The *info* field also contains the validity information, thus allowing the TSO to delete the expired commitments from the database of spent commitments. It also issues a temporary ticket, which is a signed timestamp t_{val}, starting bus stop loc_{start}, challenge c_{start} and reduction information, $info_{tp}$ *(9)*. When the temporary ticket is issued, the T_{app} stores it and can delete the *eTokens (12)*.

At the start of the ride, the traveller's application establishes a short-range anonymous channel[5] with the validation device PTO_{VM}, which corresponds to scanning a smart card in the existing systems. The application shows the signed ticket sig_{val} *(13)*. After verifying it, as well as the starting time and location *(14)*[6], the validation machine creates a signature on the current time and location, vehicle identifier, challenge c and reduction information *(16)*. The user stores the signed data as a ticket with a validated start *(18)*.

3.5 Validation of the Trip End Using the ePurse

In the existing systems, the travellers validate their cards when exiting the vehicle as well, in order to receive back the difference between the guaranteed amount and the price of their journey. Similarly, in PPTS the smartphone establishes an anonymous short-range communication with the validation machine to receive the change (Protocol 4). Initially, T_{app} sends the signature sig_{start} and corresponding information from the beginning of the journey *(2)*. The validation machine verifies the signature and the bus identifier *(3)* and calculates the applicable fare *(5)* before generating the trip-end ticket *(6)*. The trip details, including *fare* and a fresh nonce, c_{end}, are signed to create a single-use ticket. The ticket and corresponding details are sent to the traveller's application *(7)* and are used to have the change reissued from the TSO *(13-15)*. The change is received in the form of an *eToken*. The traveller is awarded with the change only if the trip was not refunded before, the signature of the PTO_{VM} verifies and the time duration of the trip is within the maximal boundaries, ensuring that the

[5] We employ the establishment of communication based on device-generated identifiers, which change with every new ride the traveller takes [9].

[6] The PTO allows for a sufficient ticket validity, i.e. difference between t_{val} and t_{start}.

traveller is not trying to combine two different trips as one *(10)*. For a limited time the TSO stores the *trip* data to prevent double refunds *(11)*.

Protocol 4: Validation of the trip end using the ePurse.

validateEndTrip($ePurse$)

(1)	T_{app}		: $\{sig_{start}, c_{start}, t_{start}, loc_{start}, id_{bus}, info_{tp}\} \leftarrow \mathsf{load}(val_{start})$
(2)	$T_{app} \rightarrow$	PTO_{VM}	: $sig_{start}, c_{start}, t_{start}, loc_{start}, id_{bus}, info_{tp}$
(3)		PTO_{VM}	: if (!verify($PK_{PTO}; sig_{start}, c_{start}, t_{start}, loc_{start}, id_{bus}, info_{tp}$) $\vee$!verify(id_{bus})) abort
(4)		PTO_{VM}	: $\{t_{end}, loc_{end}, c_{end}\} \leftarrow \mathsf{generate}()$
(5)		PTO_{VM}	: $fare = \Delta(loc_{end}, loc_{start}) \times tariff_{km} \times reduction$
(6)		PTO_{VM}	: $sig_{trip} \leftarrow \mathsf{sign}(SK_{PTO}; trip : \{c_{end}, fare, id_{bus}, t_{start}, t_{end}\})$
(7)	$T_{app} \leftarrow$	PTO_{VM}	: $sig_{trip}, trip$
(8)	T_{app}		: store($sig_{trip}, trip$), delete(val_{start})
			After exiting the vehicle:
(9)	$T_{app} \rightarrow$	TSO	: $sig_{trip}, trip$
(10)		TSO	: if (seen(c_{end}) $\vee$!verify($PK_{PTO}; sig_{trip}, trip$) $\vee$!verify($\Delta(t_{end}, t_{start}) \subset boundary$)) abort
(11)		TSO	: store($trip$)
(12)		TSO	: $diff = \Delta(D_g, fare)$
(13)	$T_{app} \leftarrow$	TSO	: $info : D_{diff} \leftarrow \mathsf{generate}(diff)$
(14)	T_{app}		: $C \leftarrow \mathsf{commit}(r, Cred.tss)$
(15)	$T_{app} \leftrightarrow$	TSO	: $pbsig \leftarrow \mathsf{PBSign}(SK_{TSO}; C; info)$
(16)	T_{app}		: store($eToken : \{C, r, info, pbsig\}$)

When the accumulated change tokens exceed the guaranteed amount, D_g, the T_{app} interacts with the TSO online to spend these smaller denominations and receive one token of value D_g. The spending is performed similarly as when a trip is started (Protocol 3, *steps 5-7*), without the reduction-related interactions, and a new *eToken* is issued with a fresh commitment to the secret *tss* (Protocol 4, *steps 13-15*). Using a new commitment makes the spending of this *eToken* unlinkable to its earning interaction, or the interactions when the change used for its creation was obtained.

3.6 Random Trip Inspection

With the PPTS system, it is also possible to perform random inspections of the travellers and their tickets. The protocol is performed between the inspection authority, i.e. the I_{MD} and the traveller's mobile application (Protocol 5). The channel that is established is anonymous and short-range communication is used to prevent interception [9]. The I_{MD} initially sends a challenge c_I and requests proof of the currently held ticket. Using the challenge, T_{app} creates a proof of possession of a valid PPTS credential and shows the picture attribute recorded in it. The provided proof is verified by the machine and the inspector can check the picture displayed on the machine's screen. In case the proof is not valid,

the traveller is identified by showing the PPTS credential's name and address attributes and with a document, such as a driver's licence. In addition, if the PPTS credential picture belongs to another person, the credential is also revoked.

Protocol 5: Random Trip Inspection

tripInspection($trip$)

(1) $T_{app} \leftarrow I_{MD}$: requestInspection(c_I)
(2) T_{app} : $\{sig_{start}, c_{start}, t_{start}, loc_{start}, id_{bus}, info_{tp}\} \leftarrow$ load(val_{start})
(3) T_{app} : $\pi \leftarrow SPK\{(MS) : Cred.validity \wedge Cred.picture\}(c_I)$
(4) $T_{app} \rightarrow I_{MD}$: $\pi, Cred.picture, c_{start}, t_{start}, loc_{start}, id_{bus}, sig_{start}$
(5) I_{MD} : if (!verify($PK_{PTO}; sig_{start}, c_{start}, t_{start}, loc_{start}, id_{bus}, info_{tp}$)$\vee$!verify(π, c_I)) {
(6) $T_{app} \leftrightarrow I_{MD}$: $\{name, address\} \leftarrow$ showCred($Cred$)
}
(7) I_{MD} : if (!verify($Cred.picture$)) {
(8) $T_{app} \leftrightarrow I_{MD}$: $\{name, address\} \leftarrow$ showCred($Cred$)
(9) $TSO_{AS} \leftarrow I_{MD}$: revoke($Cred$)
}

4 Evaluation

The proposed system allows for a privacy-preserving, yet full-fledged ticketing services. It aims to allow the travellers to manage their own data, without introducing security issues for the providers. Both the personal information and travelling history are held and managed by the traveller. All the travellers equipped with an NFC-enabled smartphone are able to benefit from this privacy-protecting scheme. The providers' equipment could at the same time support the interface with the contactless smart cards, thus allowing travellers to use the basic system flavour, however, without the privacy assurances of the PPTS scheme.

4.1 Security and privacy of the system

This section evaluates the security and privacy of the system against the defined attacker model (Section 2.2). In the following analysis we assume that the traveller device is communicating with the provider's equipment over an encrypted channel with server-side authentication.

Traveller privacy refers to preventing disclosure of the travellers' identities and travel patterns. It is ensured by means of underlying cryptographic technologies. Traveller's personal information is only disclosed to the TSO at registration time. It is certified in an anonymous credential, which allows proving existence of a valid registration, while hiding all other data it records. The credential properties ensure that different uses of the same credential remain unlinkable [6]. The

traveller also interact with the TSO to obtain the *eTokens*, which are partially blind signatures applied on the commitments to the secret *tss*. By signing a commitment to the credential secret, the token is linked to the traveller, while the commitment hides the actual value of the unique secret to prevent profiling. The link between the interactions for issuance of a token and its spending cannot be derived, due to properties of the partially blind signature scheme [15]. Similarly, the signed tickets issued by PTO_{VM} at the end of a journey have a fresh nonce included in them, preventing the TSO from linking the interactions for spending an *eToken* and obtaining the change. Even if the TSO and PTO entities collude by merging their databases, no new information would be learned. The individual rides of a traveller recorded by the PTO cannot be linked together or to any identifying information, as no unique identifiers are used. Finally, since the picture data disclosed during random inspections is a unique identifier, we assume these checks are infrequent and do not allow for profiling travellers.

The traveller data is also protected from *external attackers* trying to extract the data from the T_{app} or listen in on the communication with the server, with encrypted communication with authenticated entities. Moreover, no action, such as credential attributes disclosure, can be carried out without user knowledge.

The *unauthorised usage* of travel products refers to forging, double-spending, sharing or changing details of valid tokens. In order to create a new valid token, which is a partially blind signature, an attacker needs access to the secret signing key, which we assume is prevented. In the similar manner, the token details (*info*) cannot be altered, as they are part of the signed structure. For utilising a valid token, it is necessary to prove possession of a valid PPTS credential, linked to the token (*step 5*, Protocol 3). Only entities with the knowledge of the credential's master secret can create the required proofs. Even if registered users try to share *eTokens*, they are not able to provide proofs for products which are not linked to their credential (e.g. *steps 5-7* in Protocol 3). Although sharing credentials and their master secrets would allow to exchange travel products amongst users, we assume that existing mechanisms for credential sharing prevention are in place [6]. In order to prevent sharing the PTO-issued tickets, they include a nonce c_{start}. The validation machine at the end of a journey can verify that the ticket has been used only once, even without online contact with the backend. Finally, double-issuance of the change tokens is prevented with nonces added to the temporary tickets for trip end, c_{end}. The ticket verification carried out by TSO includes a check whether the nonce is already recorded in its database linked to a particular time period. As these tickets have temporary validity, the databases do not have unlimited growth. The scheme also prevents fare evasion, as travellers 'spend' the maximal ticket price when starting a journey and are not allowed to board the bus in case of insufficient balance (*step 14*, Protocol 3).

The users who interact for using their travel products are also protected from *seamless overcharging*, as the user is presented with the amount that is going to be spent and receive proofs of journey details.

4.2 System efficiency

For improved efficiency, some of the operations are carried out offline. The mobile application can create commitments and store them to be used at the time of check-in or check-out. Additionally, when the ePurse recharge is performed, issued digital tokens have the value required to show when taking one standard ride. This means that when taking a ride, the user usually only spends one token. When a user collects enough change tokens to create a new token with standard fee, the application interacts with the TSO to spend the tokens with smaller amount and have the standard-value token issued.

The performance evaluation was done by measuring the execution time for the cryptographic primitives utilised in the protocols[7]. The measurements for the client-side (T_{app}) and PTO's validation machine (PTO_{VM}) were performed on a Samsung Galaxy S3 (GT-I9300) with a quad-core 1.4 GHz Cortex-A9 processor, 1GB of memory and Android 4.1.2. The TSO operations were done on a workstation with Intel® Core™ i7-3770 CPU, 16 GB of memory and Ubuntu 13.04.

The most time-critical operations of the scheme are interactions between the traveller's application and the validation machine at the start and end of a trip. The cryptographic operations they comprise, i.e. creation and verification of digital signatures, have proven to be very efficient. For 100 measurements using a 2048 bit RSA key pair, signing and signature verification took on average 40.93 $\pm$ 0.18 ms, with 95% confidence intervals.

In addition to these operations, the traveller's application communicates with the TSO before boarding the vehicle to receive a temporary validation confirmation. This interaction incorporates creation of a signed zero-knowledge proof on the client side and verification of this proof and two partially blind signatures on the server side. The average time of client execution is 197.22 $\pm$ 33.4 ms for 100 measurements and 95% confidence intervals. The server execution takes 60.15 $\pm$ 7.8 ms for verifying the SPK and 8.14 $\pm$ 2.9 ms for verifying a partially blind signature. When the traveller exits the vehicle, the interaction with the validation machine again incorporates digital signature creation and verification. In a subsequent interaction with the TSO, the traveller is issued with an *eToken*, which takes 104.62 $\pm$ 22.46 ms for client executions and 172.71 $\pm$ 43.74 ms on the server side. These values show adequate efficiency. In addition, they are performed after the user leaves the vehicle and are not time-critical. Overall, the results show that the protocol's efficiency with the mentioned efficiency improvements allows the approach to be deployed in a practical ticketing system.

4.3 Design Extensions

The described proposal focuses on local transport, such as bus services, but can easily be extended to other transport systems. Also, the issuance and utilisation

[7] For more information about the implementation utilised for the performance evaluation, please consult: `https://mobcom.org/deliverables/inshopnito_code`

of the transport products is designed in a way that allows for a flexible system. In some systems a maximal charge for a single day is set (e.g. 'capping' in London Oyster system). To achieve this property in PPTS, the users would be issued with a proof-of-spending *eToken* at the end of every ride. The issuance would be performed by the TSO, as in Protocol 4. After the maximal amount is reached, the application contacts the TSO to be issued with *eTokens* which correspond to single rides for that day. There is no limit on the number of these additional rides, as the traveller can exchange one *eToken* for a new batch (see Section 3.3).

It is also possible to allow the providers to learn how buses should be synchronised, by linking together multiple jumps that comprise one journey. A traveller would create a domain pseudonym using the PPTS credential (which does not reveal any additional information), show it on all jumps of one journey and would change it for every new journey. The credential properties provide assurances that different credentials cannot produce same domain pseudonyms [6].

5 Related Work

As novel technologies advance the ticketing systems, research initiatives also increase their focus on this field. However, multiple studies identify that privacy is a serious issue in the novel designs, as they allow to collect information about the users, such as locations and movements [10, 17]. The way some commercial systems are deployed also create grounds for concern [24]. For instance, the Washington D.C. Metro was functioning for years without a clearly defined privacy policy [22, 11]. Many solutions rely on cards with unique identifiers and utilise other personal information, even credit card data [21]. To tackle these privacy concerns, there is a limited number of research proposals. One of the initial proposals by Heydt-Benjamin et al. [11] uses an e-cash payment scheme, anonymous credentials and proxy re-encryption for concealing personal data, while ensuring correct payment. However, the system describes the functioning on a higher level of abstraction and system flexibility is limited compared to currently offered services. On the other hand, work of Jorns et al. [12] focuses on the problem of location services. As the network operators gradually open their interfaces for mobile applications that use travellers' location and presence information, privacy issues arise. The paper proposes a pseudonymous system for protection of user identity. However, there are still possibilities for user profiling, as the users are pseudonymous and every ticket contains a unique identifier, linking its purchase and usage. For ensuring unlinkability, a proposal by Reza et al. [19] relies on trusted anonymisers and employing physically unclonable functions (PUFs) technology on all the used tokens. However, we focus on the software solutions and limit hardware requirements. Similarly, some proposals protect user privacy using anonymous credentials or e-cash schemes. A system based on anonymous credentials is designed by Verslype et al. [23]. However, the system is not flexible enough for public transport systems, as it does not consider the option of pricing per travelled distance or reduction plans. Rupp et at. [18] propose a lightweight payment scheme for transit based on Brands e-cash

scheme and blind Boneh-Lynn-Shacham signatures. It is based on users purchasing bundles of credentials which represent single travel tickets. At the end of a ride, the traveller is refunded with change. While the scheme is protecting the privacy of the travellers, it does not allow for services such as discount plans or monthly passes, which we aim to support with our proposal. Finally, a practical proposal of Kerschbaum et al. [13] addresses the use case of the Singapore ticketing system where on every card top-up, the unique identifier is disclosed and the recorded travelling data is leaked from the card. They design a post-paid billing system based on partially homomorphic encryption that allows for data analysis, although requiring expensive computation on the server side. Unlike in this proposal, we only focus on the pre-paid schemes, while limiting server side computation. Additionally, similar systems for privacy-preserving travel billing exist in the domain of toll systems. The privacy-protecting solutions [14, 2], however, are not applicable to the public transport ticketing systems, as they have different hardware assumptions, namely the existence of trusted on-board units.

6 Conclusion

In this work we have presented a privacy-preserving ticketing system, which ensures unlinkability of travellers' interactions and prevents creating identifiable or even pseudonymous profiles. The proposal is flexible and can offer the services delivered in the currently deployed ticketing solutions. Although the privacy of the user is protected and the products' issuance and utilisation cannot be linked together, the service providers are still receiving the needed security assurances and can impose restrictions on utilisation and sharing of the purchased products. Our system prevents traceability of user actions even if the traditional and linkable payment systems are used. The trust assumptions are also reduced, as the users do not need to rely on the service providers to handle their data in a way that corresponds to their privacy preferences and trust that the security of the stored information cannot be defeated. They can be assured that their personal data is not revealed and only need to trust the application developer, while the implementation itself can easily be audited to assure the users that the performed actions correspond to the expected behaviour. At the same time, the minimisation of information is done so that it is still possible to perform the statistical analyses for improving the efficiency of the provisioned services.

The deployment of the proposal in real systems is eased by limited requirements, as it relies on the NFC-enabled smarphone technology on the traveller's side. It also demonstrates a usable efficiency. Similarly to the existing solutions, the validation machines do not require real-time communication with the backend to verify the validity of presented products. Finally, the performance evaluation illustrates a usable and efficient system.

References

1. M. Abe and T. Okamoto. Provably secure partially blind signatures. In *Advances in Cryptology–CRYPTO 2000*, pages 271–286. Springer, 2000.

2. J. Balasch, A. Rial, C. Troncoso, B. Preneel, I. Verbauwhede, and C. Geuens. Pretp: Privacy-preserving electronic toll pricing. In *USENIX Symposium*, 2010.
3. M. Bellare and O. Goldreich. On defining proofs of knowledge. In *Advances in Cryptology – CRYPTO'92*. 1993.
4. S. A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, Cambridge, MA, USA, 2000.
5. Calypso. Calypso functional specification, card application. `http://www.calypsostandard.net/index.php/documents/specifications/public-documents/78-010608-functional-card-application`, 2014.
6. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. *EUROCRYPT*, 2001.
7. D. Chaum. Blind signatures for untraceable payments. In *CRYPTO*, 1982.
8. G. de Koning Gans, J.-H. Hoepman, and F. D. Garcia. A practical attack on the mifare classic. In *Smart Card Research and Advanced Applications*. 2008.
9. H. Eun, H. Lee, and H. Oh. Conditional privacy preserving security protocol for nfc applications. *Consumer Electronics, IEEE Transactions on*, 59(1):153–160, 2013.
10. T. Foss. Safe and secure intelligent transport systems (its). In *Transport Research Arena 5th Conference: Transport Solutions from Research to Deployment*, 2014.
11. T. S. Heydt-Benjamin, H.-J. Chae, B. Defend, and K. Fu. Privacy for public transportation. In *Privacy Enhancing Technologies*, volume 4258 of *LNCS*, 2006.
12. O. Jorns, O. Jung, and G. Quirchmayr. A privacy enhancing service architecture for ticket-based mobile applications. In *Availability, Reliability and Security, ARES 2007. The Second International Conference on*, pages 139–146. IEEE, 2007.
13. F. Kerschbaum, H. W. Lim, and I. Gudymenko. Privacy-preserving billing for e-ticketing systems in public transportation. In *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, WPES '13, 2013.
14. S. Meiklejohn, K. Mowery, S. Checkoway, and H. Shacham. The phantom tollbooth: Privacy-preserving electronic toll collection in the presence of driver collusion. In *USENIX Symposium*, 2011.
15. M. Milutinovic, I. Dacosta, A. Put, and B. De Decker. An efficient and unlinkable incentives scheme. CW Reports CW659, Dept. Computer Science, KU Leuven, 2014.
16. T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in CryptologyCRYPTO91*, pages 129–140. Springer, 1992.
17. M.-P. Pelletier, M. Trpanier, and C. Morency. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4):557 – 568, 2011.
18. A. Rupp, G. Hinterwlder, F. Baldimtsi, and C. Paar. P4r: Privacy-preserving pre-payments with refunds for transportation systems. In A.-R. Sadeghi, editor, *Financial Cryptography and Data Security*, volume 7859. 2013.
19. A. Sadeghi, I. Visconti, and C. Wachsmann. User privacy in transport systems based on RFID e-tickets. In *Proceedings of the 1st International Workshop on Privacy in Location-Based Applications, Malaga, Spain, October 9, 2008*, 2008.
20. N. Semiconductors. Mifare standard 4kbyte card ic functional specification, 2012.
21. The Smart Card Alliance. Hong Kong Octopus Card, 2006. January issue.
22. The Smart Card Alliance. Smart Card Talk Standards, 2006. January issue.
23. K. Verslype, B. De Decker, V. Naessens, G. Nigusse, J. Lapon, and P. Verhaeghe. A privacy-preserving ticketing system. In *Data and Applications Security*, pages 97–112. Springer, 2008.
24. N. Winters. Personal privacy and popular ubiquitous technology. *Proceedings of Ubiconf*, 2004.