A Privacy-Preserving E-Ticketing System for Public Transportation Supporting Fine-Granular Billing and Local Validation

Ivan Gudymenko[*]
Faculty of Computer Science, TU Dresden
01062 Dresden, Germany
ivan.gudymenko@mailbox.tu-dresden.de

ABSTRACT

Next generation systems for public transportation based on e-ticketing offer numerous advantages both for end users and providers of transportation service. At the same time, individuals using such systems tend to leave ubiquitous digital traces which raises serious concerns over privacy. This paper focuses on this issue and presents a framework for constructing privacy-preserving e-ticketing systems for public transportation. What differentiates our solution from other research contributions and real-world systems is that while being inherently privacy-preserving it *(1)* provides support for fine-granular billing (for registered customers) and *(2)* is based on loosely-coupled architecture (allows for local e-ticket validation). Our concept is additionally backed up by a practical evaluation of the most time-critical part of the system – handling of check-in/check-out events in the front-end.

Categories and Subject Descriptors

Domain-specific security and privacy architectures, Secure online transactions, Distributed systems security.

General Terms

Systems design

Keywords

Privacy, Public Transportation, E-ticketing, NFC.

1. INTRODUCTION

As ubiquitous computing has already transformed itself from a vision to reality, various parts of our daily lives are being rapidly affected by it. The area of public transportation with the new generation of intelligent transportation systems based on e-ticketing (ITSE) is one of the tangible examples of this process. In such systems, conventional paper tickets are replaced by their electronic counterparts which are referred to as e-tickets within this paper. In many large cities such as London, Singapore or São Paulo, ITSE have already been in operation for quite a while. However, the latest advances in mobile technology allowing to integrate RFID front-end (in form of NFC) into widely spread smart phones have opened new opportunities for ITSE at the same making such e-ticketing systems even more ubiquitous.

Along with the numerous benefits of future public transportation systems, serious privacy concerns arise. The utilization of e-tickets integrated into a customer's smart phone with NFC support or into a smart card is going to dramatically multiply the digital traces left by people using the system. This paves the way to various misuse scenarios if no mechanisms for privacy protection are explicitly considered. Till now, this issue has not been sufficiently addressed by the industry (see the related work discussion in Section 3). The academic solutions developed so far are either based on additional assumptions or are far too inefficient to be integrated into a real-life system.

The contribution of this paper, therefore, is a specifically designed privacy-preserving framework which explicitly addresses the issues of privacy protection in ubiquitous public transportation systems. On the one hand, it allows for implementation of flexible pricing schemes and fine-granular billing. A transport authority, therefore, would be able to fully leverage the potential of e-ticketing systems. At the same time, our solution addresses the issue of privacy protection from the outset through specific system design. Moreover, in contrast to several other solutions, the presented framework is based on a loosely-coupled architecture, that is terminals do not have to maintain permanent connection to the back-end in order to serve check-in/check-out requests. The paper is structured as follows. Firstly, the requirements for a target system are presented in Section 2. The discussion of the related work is performed in Section 3. An attacker model adopted for the target system is discussed in Section 4. Our privacy-preserving framework is presented in Section 5 and evaluated in Section 6. Section 7 concludes the paper.

[*] The author would like to express his gratitude for fruitful discussions and constant support to his fellow colleagues at chair for privacy and data security (TU Dresden). Furthermore, Pavel Plakhin and Manuel Weißbach have to be acknowledged for their input on the practical part. The author is financially supported by the Free State of Saxony and the ESF under grant #100097574

SIN'14 September 09 - 11 2014, Glasgow, Scotland, UK
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3033-6/14/09 ...$15.00.
http://dx.doi.org/10.1145/2659651.2659706.

2. SYSTEM REQUIREMENTS

Before analyzing the existing work and presenting our solution, the most important requirements for a target e-ticketing system (predominantly from a privacy view) are concisely outlined and discussed. Since the current paper focuses on privacy protection, the detailed discussion of functional and interdisciplinary requirements is left out of scope. Four major classes of requirements can be distinguished in our case: *(1)* Privacy, *(2)* Billing support, *(3)* Loose coupling, and *(4)* Efficiency.

(1) Privacy. The first requirement can be further classified into:

(a) *Privacy against terminals.* Terminals must not be able to identify and track valid e-tickets.

(b) *Privacy against the back-end.* The back-end is allowed to correlate travel records related to a single e-ticket while being prohibited from identifying valid e-tickets.

(c) *Privacy against external observers.* An external observer must be prevented from deriving any identifying information from interactions between e-tickets and terminals.

(2) Billing support. The support for fine-granular billing is highly essential in the dynamically evolving public transportation market, since it allows the companies to win more customers by creating attractive pricing schemes. Moreover, the users do not necessarily have to familiarize themselves with complex tariff schemes (e.g., depending on the number of zones to be crossed and/or time of day). The system can automatically calculate the best price (for instance, based on check-in/out data) and issue the bill accordingly. In [6], for instance, it was stated that a considerable number of customers choose personalized cards since they provide more services. Currently, several systems for public transportation have already reacted to this trend offering a regular billing approach [10, 26]. Certain privacy-preserving solutions with regular billing support consider specific tariff schemes tailored to the mechanisms for privacy protection employed in the system, see [13], for example. It is, however, important to *separate* the issues of system architecture (specifically privacy protection) from the development of tariff schemes. This enables flexibility of the latter (which is likely to be a subject to constant changes) as well as provides for the necessary updates of privacy-preserving mechanisms if required (e.g., relevant system patches).

(3) Loose coupling. In the real-world scenario, the check-in/check-out terminals do not always maintain a constant real-time connection to the back-end. Our discussions with the representatives of public transportation companies of Metrô São Paulo, Brazil and Dresdner Verskehrsbetriebe, Germany, have shown that tightly-coupled systems are highly impracticable and are not likely to gain acceptance in the real-world. Therefore, privacy-preserving solutions being developed for ITSE must be compatible with such kind of loosely-coupled architecture.

(4) Efficiency. Check-in/check-out events handling is the most time-critical in the system and directly affects the customer experience. In practice, the maximum tolerated value ranges from 0.2 sec (London Oyster Card) to 2 sec (Singapore EZ-Link) [18].

3. RELATED WORK

3.1 Real-World Systems

In case of a real-world ITSE, the technical part of the system is primarily focused on the issues of *(1)* direct functionality, *(2)* system security, and last but not least *(3)* resource effectiveness (having direct cost implications). Privacy protection as a whole is usually considered in the second place, if at all. Moreover, in order to provide for efficiency of security-relevant transactions (especially in the system front-end), customer privacy is often *traded off.* For example, during the authentication session between an e-ticket and a terminal, the former has to send its unique identifier in order to enable terminal-side key derivation [25]. The whole transaction is then usually stored at the terminal (often in plain text) and (subsequently) sent to the back-end for validation, control and maintenance purposes. Should it be possible to associate this identifier with the customer using an e-ticket, he or she immediately becomes ubiquitously traceable within the system (both in the back-end and in the front-end) and possibly even by adversarial parties exogenous to the current ITSE. Similarly, in the specification of eTicket Germany (known as Core Application) [27], each e-ticket can be uniquely identified within the system. During the conventional certificate-based authentication between a terminal and an e-ticket (during check-in/out), the latter has to provide its unique certificate by this making itself ubiquitously traceable not only by the back-end but also by each terminal.

From the legal prospective, a new EU framework was adopted in 2010 which is specifically targeted at intelligent transportation systems [8]. Although the importance of privacy preservation is explicitly underlined in it, no concrete recommendations beyond certain generic measures (e.g., data anonymization) were given.

3.2 Academic Solutions

The academic contributions directly considering privacy protection in the area of public transportation can be roughly divided into two categories: *(1)* tightly-coupled: the back-end is "always online" and can serve the requests originating from the front-end in real time; and *(2)* loosely-coupled: terminals can validate e-tickets without consulting the back-end in a real-time manner.

3.2.1 Tightly-coupled Systems

In [12], a privacy-preserving framework based on e-cash, anonymous credentials and proxy re-encryption was presented. Its privacy properties essentially inherit from the e-cash concept. Therefore, an honest customer adhering to the protocol remains completely untraceable. On the other hand, the proposed framework provides only a limited support for flexible pricing schemes which is one of the essential advantages of ITSE. Moreover, the efficiency of zero-knowledge proofs performed by an e-ticket during each check-in/out was not assessed. Furthermore, in case of a stored-value ticket, the price for a ride is calculated on the fly by the back-end which is likely to introduce additional delay during check-out. The authors of [21] presented a token-based solution based on partially homomorphic encryption due to Paillier [20]. This approach, however, solely considers one-time tickets which have the same price regardless of the distance travelled, city zone or time of day. Moreover, in [21] collision handling with respect to token generation was not addressed (in the real system, there would be tens of millions of tokens in operation during a single day).

3.2.2 *Loosely-coupled Systems*

The authors of [2] presented a refund oriented solution based on e-cash considering a single-trip tickets. In contrast to [12], terminals can validate e-tickets without consulting the back-end in real time. In order to perform a single trip, an e-cash-based token (corresponding to the maxim price of a single ride) must be spent in advance on check-in. The actual price together with the respective refund is determined by the terminal during check-out. The refund can be reimbursed later by a special refund machine connected to the back-end. This may introduce an additional burden for customers willing to use the system. Moreover, the supported pricing schemes are fairly simple, since the actual price must be determined by a check-out terminal in a timely fashion. In [22], the so-called trusted anonymizers were introduced which can be used in an add-on fashion by a customer and are decoupled from the direct functionality of a system. That is, the e-ticket can still be validated in a non-privacy-preserving way if the respective customer's anonymizer is for some reason unavailable. The solution is based on secure key storage with physically unclonable functions (PUFs), symmetric key based authentication, and rerandomizable encryption. Similarly to [2], it allows for local validation by terminals but does not allow for fine-granular billing.

Unfortunately, none of the reviewed solutions simultaneously satisfies both requirements, privacy and billing support (see Section 2). Moreover, as already mentioned in Section 2, the solutions falling into the first category (tightly-coupled systems) have little pertinence to real-world scenario. Therefore, the framework proposed in this paper is based on the decoupled architecture with terminals being able to locally validate e-tickets. The support for fine-granular billing is provided as well. Before presenting our solution, an adopted attacker model is discussed in the next section.

4. ATTACKER MODEL

The adopted attacker model can be presented as follows:

1. (Outsider) An observing attacker (outsider) must be prevented from deriving any identifiable information from interaction between terminals and e-tickets.
2. (Insider) Terminals must not be able to identify and track valid e-tickets.
3. (Insider) Back-end must be prevented from learning the identifiers of e-tickets.

The division into outsider/insider is made with respect to the attacker's involvement into system information flow. That is, the attacker of type 1 is an entity exogenous to the system. It is assumed that an observing attacker is polynomial-time bounded and not able to physically tamper with the device carrying an e-ticket. Terminals (attacker type 2) are widely distributed within the transport network and are, therefore, situated for the most part in an unsecured area and in certain cases may be subject to compromisation. Moreover, the wireless interface used for ticket validation can be misused by third parties representing an additional attack vector (e.g., a buffer overflow attack mounted via NFC). This is further backed up by the claim made in [24] that transport authorities are willing to store as minimum critical data at the terminal side as it is possible. Therefore, in this attacker model, terminals are prohibited from identifying and tracking e-tickets as well as distinguishing between them. Privacy protection against the back-end is represented by the attacker of type 3. In order to issue a bill for personalized e-tickets, the back-end is allowed to correlate different rides to a customer pseudonym but is at the same time prohibited from learning the underlying user identity.

5. OUR SOLUTION

5.1 Basic Intuition Behind the Approach

In general, each e-ticketing system for public transportation can be divided into two distinct parts: the front-end and the back-end. The former consists of e-tickets which are interacting with terminals through a user device (an NFC-enabled smart phone or a contactless smart card). The back-end is (usually asynchronously) connected to terminals and incorporates powerful interconnected computing centers which maintain system functionality. In this paper, a further external component is introduced to enhance privacy – a trusted third party (TTP), see Figure 1. It acts as a trusted mediator between a transport authority (which controls the system) and its end users. The main idea behind this concept is the following. In order to issue a bill, the back-end does not necessarily have to possess identifying information about a particular user of an e-ticket. It merely needs to be able to correlate different rides performed by a customer to a certain pseudonym which has been negotiated with a TTP in advance and based on this information to apply the deployed pricing schemes with the subsequent billing procedure. The resultant bill together with the corresponding pseudonym is periodically sent to TTP. The latter has no knowledge of travel history of a customer but is solely aware of the overall bill and the user behind the pseudonym. Individual payments are then forwarded to the transport authority (TA) by TTP in an aggregated form. Thus, TA trusts TTP that users are correctly billed (together with payment enforcement) while customers rely on TTP to protect their privacy and forward payments to TA. In order to enhance user experience, the rides history can be additionally stored at the user device (e.g. a smart phone) so that it can be locally viewed by a customer later.

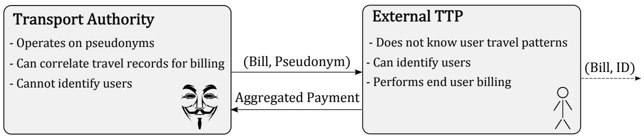
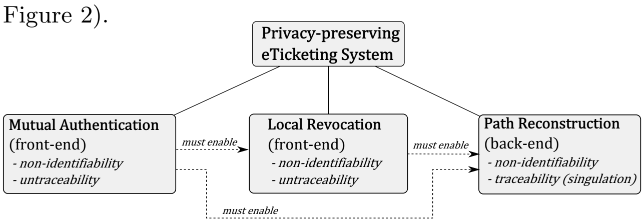


Figure 1: Solution overview.

In system front-end, terminals do not necessarily have to possess uniquely identifiable information about an e-ticket (e.g., an e-ticket ID) beyond the mere fact of its validity. Therefore, in the proposed framework, terminals perform validation without requiring such kind of information from the e-ticket side (unlike it is done in the majority of the real-world systems, see Section 3). Rather, on each check-in/out, an e-ticket solely proves that it belongs to a certain group (e.g., monthly or yearly tickets) and possesses corresponding credentials which are not expired. In order to prevent revoked users from entering the system, a terminal additionally checks the received message from an e-ticket (constructed in a specific way) against a regularly updated blacklist. In what follows (Section 5.2), the main building blocks of our solution are outlined. This is followed by the description of a general information flow in the system in Section 5.3. Having outlined the system as whole, each component is described in more detail in Sections 5.4, 5.5, 5.6.

5.2 Solution Building Blocks

Our privacy-preserving framework for e-ticketing systems essentially consists of three main building blocks depicted in Figure 2).



Privacy-preserving eTicketing System
Mutual Authentication (front-end)
- non-identifiability
- untraceability
must enable
Local Revocation (front-end)
- non-identifiability
- untraceability
must enable
Path Reconstruction (back-end)
- non-identifiability
- traceability (singulation)
must enable

Figure 2: Solution building blocks.

In Figure 2, privacy properties such as non-identifiability and untraceability refer to an e-ticket (either against a terminal in the front-end or against the back-end system). All of the three aforementioned building blocks must be implemented in a privacy-preserving way corresponding to the attacker model defined in Section 4 and being conform to the requirements discussed in Section 2. That is, on each check-in/out the mutual authentication should not leak any auxiliary information about the e-ticket which can be misused by terminals or exogenous entities for its *(a)* identification (the worst case), *(b)* tracking, or *(c)* linking different communication sessions with the terminal(s) together. At the same time, invalid e-tickets (i.e. the ones which have been blacklisted) must be prohibited from being granted the public transportation service. Therefore, terminals must be able to check if the e-ticket communicating with it has been blacklisted or not. An important condition here is similar to the one for mutual authentication above: valid e-tickets must remain anonymous (to the terminal) and untraceable. Lastly, in order to issue a bill, different rides must be correlated to each other in such a way that the underlying identity of each user remains unknown to the transport authority (referred to as path reconstruction in Figure 2).

As indicated in Figure 2, there are important interdependencies between the building blocks. Privacy-preserving authentication must enable local revocation at the terminal side as well as path reconstruction in the back-end. Similarly, local revocation (with the described privacy properties) must not prevent path reconstruction. The aforementioned interdependencies introduce a serious challenge for system design, since such properties as non-identifiability and untraceability on the one hand and user revocation together with path reconstruction on the other hand are inherently contradicting. In what follows, our solution addressing all of these aspects is presented in more detail.

5.3 Basic Information Flow

Initialization

Before being able to actively use the transportation system, each customer has to engage into an initialization phase with TTP. This can be carried out via a special issuing machine, at the office of transport authority, or via the internet. On registering the customer, TTP creates the respective pseudonym P_i^T and forwards it to TA. The latter further transforms the received pseudonym into its encrypted form P_i^A and operates on it (pseudonymization is explained in Section 5.4 in detail). The customer in turn gets the necessary credentials from TTP to start using the system, namely: a TA public key k_{ta}^+, a key pair corresponding to the subscription group (e.g. a monthly or yearly pass), (k_{gr}^+, k_{gr}^-), as well as the specifically created customer pseudonym P_i^T (together with its TA-form P_i^A). A subscription group key can be constructed in accordance with the concept of group signatures, see [17, 5], for example. Another more straightforward and sometimes more efficient approach is to use a conventional key pair (e.g. RSA) for each ticket group. The received key pair is signed by the TA and can be viewed as a kind of a digital certificate.

Using The System

On entering the public transportation system, a user performs check-in at the entrance terminal. This involves three stages: *(1)* secure session establishment, *(2)* mutual authentication, and *(3)* blacklist check (see Figure 3). The secure session can be established either using an algorithm defined by the e-ticketing application itself (e.g., through the application-defined Diffie-Hellman key agreement) or alternatively by leveraging the standard techniques defined in ISO 7816-4 [14] or in NFC-SEC-01 [7]. Note that depending on the way the secure session has been established during the first stage, additional binding of the exchanged keying material (e.g., DH ephemeral keys) to the corresponding certificates may be required to prevent man-in-the-middle attacks. It has to be mentioned, however, that due to the physical properties of communication between terminals and e-tickets, man-in-the-middle attack (in contrast to the relay attack) is extremely unlikely in practice [11]. The subsequent communication between an e-ticket and a terminal is, therefore, secured against an observing attacker. Afterwards, mutual authentication between an e-ticket and a terminal is performed as follows. The terminal has its unique public key k_T^+ signed by the back-end. The e-ticket uses its group key pair (k_{gr}^+, k_{gr}^-) which is as well signed by the back-end. Mutual authentication is then essentially performed according to the certificate-based challenge-response. Lastly, the terminal (locally) checks if the credentials of the current e-ticket have not been revoked by consulting the blacklist (see BL Check in Figure 3). This is performed in a privacy-preserving way (in contrast to the majority of conventional systems, see Section 3). That is, each e-ticket stays anonymous and untraceable against the terminal as long as it has not been included into a terminal-side black list (similarly to the notion of conditional anonymity defined in [3]). On successful check, the terminal creates the so-called travel record (TR) corresponding to the current check-in/out event. It usually contains a timestamp, location, and other pieces of information pertaining to the e-ticket (e.g., its session pseudonym, see Section 5.4). A set of travel records maintained by each terminal, *TR*, is regularly sent to the back-end system (BE) via the backbone network (see *Send TR* in Figure 3) where they are processed for billing purposes (Figure 3, *Billing*). Terminal-side blacklists are regularly updated as well. The frequency of such updates is mainly determined by the connection type between terminals and the back-end (e.g., nightly updates as considered in [1] or more frequent updates if the connection allows).

5.4 A Privacy-Preserving Path Reconstruction

As it was already mentioned in Section 5.2, our framework essentially consists of three building blocks (see Figure 2). In this section, a privacy-preserving path reconstruction required for fine-granular billing is going to be discussed. The following challenges should be considered in this case. On the one hand, the supported fare schemes which are applied during the billing phase need to be *flexible* and *extensible*. That is, they should not be hard-tailored to a specific fare collection approach let alone to the privacy-preserving mechanisms in use (as it was done, for example, in [13]). Moreover, it should be possible to combine the rides to issue discounts (consider the example of a ”short ride” ticket up to 4 stations). On the other hand, the process of fare calculation and subsequent billing must be carried out in a privacy-preserving way, that is without directly identifying the customer and leaking information about his/her travel habits. The aforementioned issues introduce a severe contradiction between fare scheme flexibility and user privacy (in terms of individual traceability). In case of a relatively simple fare scheme where the price for a ride between each two stations (that is, between two successive check-in/check-out events) can be represented in form of a fare matrix, it is already possible to implement billing with decent privacy properties, see [16], for example. However, using such matrix-based approach entails considerable disadvantages, namely *(1)* billing inefficiency (in case of [16], cubic complexity in the number of travel records processed), and *(2)* billing inflexibility, e.g., inability to combine (i.e., to link) several rides to issue a discount, etc. Therefore, in our framework an approach based on pseudonymisation is chosen. More specifically, a reasonable trade-off is considered, namely whereas check-in/out events are completely unlinkable and untraceable in the front-end, the back-end is able to correlate different rides to a single pseudonym and to subsequently apply the deployed fare scheme for billing. The back-end still does not gain any information about the underlying user identity, which is managed by the trusted third party in our framework. The employed pseudonymisation scheme functions as follows.

During the initialization phase, a static pseudonym P_i^T is created by TTP for each e-ticket ID. The mapping[1] between P_i^T and the respective e-ticket ID is kept secret at the TTP side. P_i^T is then sent to TA to be included into the TA's pseudonym set P^T. TA, therefore, is only operating on pseudonyms and stays unaware of the underlying e-ticket ID. In order to further separate the processes of end user billing (performed by TTP) and TA-internal processes including bill calculation, P_i^T is transformed into a TA-specific pseudonym: $P_i^A \xleftarrow{trans} P_i^T$ (the notations are summarized in Table 1). This transformation is performed in such a way, that a TTP even having gained access to several records containing TA-specific pseudonyms, would neither be able to *(1)* restore the underlying P_i^T *(2)* nor to distinguish between records pertaining to different e-tickets. Such properties are required to further enforce the ”separation of concerns” between TTP and TA, namely to make sure that TTP does not gain additional information (it does not require to operate) concerning the history of rides. The transformation can be carried out as follows: $P_i^A = E_{k_{ta}^+}(P_i^T,\, s_i)$, where $E_{k_{ta}^+}$ denotes encryption under the TA public key and s_i is a random value (salt). In order to be able to restore P_i^T, the encrypted salt value is stored together with P_i^A in the back-end: $P_i^T \mapsto \left(P_i^A,\, E_{k_{ta}^+}(s_i)\right)$.

Table 1: Pseudonymisation: notation used.

Notation	Meaning
P_i^T	a static pseudonym created by TTP;
P_i^A	a static pseudonym created by TTP from P_i^T;
SP_j	a session pseudonym (randomized P_i^A).

In order to prevent terminals from tracking e-tickets (covering attacker type *2*, see Section 4), a session pseudonym SP_j is created at the e-ticket side on each interaction with a terminal:

$$SP_j = E_{k_{ta}^+}\left(P_i^A \cdot r_j\right), \tag{1}$$

where r_j is nonce number generated by the e-ticket. Since a terminal is not in the possession of a TA's decryption key (k_{ta}^-), it is infeasible for it to tell if two session pseudonyms obtained from different check-in/out events pertain to the same e-ticket or not. Neither can the terminal gain any knowledge from interaction with an e-ticket about the static pseudonym (P_i^A) of the latter. Thus, for each particular e-ticket, travel records created by terminals on check-it/out contain different session pseudonyms SP_j (depicted at the bottom of Figure 3). In order to enable bill calculation in the back-end part of the system, the pseudonym singulation step is required to correlate different session pseudonyms $\{SP_j\}$ with the respective static one P_i^A using the private key of TA k_{ta}^-. Afterwards the billing process is carried out on static pseudonyms $\left\{P_i^A\right\}$ which are finally decrypted to the initial TTP pseudonyms $\left\{P_i^T\right\}$. The result of the billing step is a set of tuples $\left(bill, P_i^T\right)$ which is regularly (e.g., monthly) send to TTP for end user billing.

5.5 Privacy-preserving Local Revocation

Enabling privacy-preserving revocation of blacklisted e-tickets at the terminal side in such a way that valid e-tickets stay completely anonymous (and untraceable) against the terminal is quite a challenging task. One of the possible solutions would be to utilize the concept of cryptographic accumulators, see [4]. In this case, instead of checking if the current e-ticket has been blacklisted, it is proved (in zero-knowledge) that a certain commited value securely stored at the e-ticket side, has been included into the terminal-side whitelist, or an accumulator. Revocation then is essentially removing the specific value from the accumulator and recalculating it anew. The new accumulator must be redistributed to all terminals in the system, similarly to the blacklist. An important caveat is, however, that along with the terminal-side accumulators, the e-ticket credentials must be updated as well on each revocation (for the proof of membership to function properly). This requirement renders it highly impractical to apply the cryptographic accumulator concept to the public transportation scenario where e-tickets

[1] One of the ways to implement such mapping is to probabilistically encrypt the e-ticket ID (for semantic security) with the private key of TTP and to keep the latter secret.

are for the most time offline. Even in case the updated accumulator value were dynamically delivered to a user device during check-in/check-out, recalculating the credentials (which have to be performed on the fly) would introduce an additional computational overhead and hence add a costly time delay to the handling of time critical check-in/check-out events. Another solution to the problem of privacy-preserving revocation could be the concept of "anonymous blacklists". An example of this approach is an anonymous blacklisting scheme called "Nymble" [15]. In this system, the multi-party concept is used to protect user privacy, namely along with the service provider and users there are two additional trusted parties: the so-called pseudonym manager and the nymble manager. In order to obtain an anonymous credential (consisting of "nymbles") for using the service, a user has to *(1)* contact the pseudonym manager at first to request a pseudonym which is *(2)* subsequently presented to the nymble manager who issues the so-called "nymbles" to the user. Nymbles are essentially one-time credentials empowering the user to get the service. This approach is, however, hardly applicable to the target scenario of public transportation. Firstly, additional interaction must be carried out by each user device (an e-ticket) on a regular basis to obtain credentials (which are the basis for blacklist check) from the respective trusted parties. In [15], it is recommended that new credentials are obtained every 24 hours. This may introduce additional nuisance for the users who would have to carry out the update procedure on such a regular basis (especially if an additional piece of equipment like RFID/NFC reader is required for this). Secondly, using one-time credentials requires some kind of a global register to keep track of the already used credentials in order to prevent double spending, etc. This is prohibitive considering a highly decentralized system architecture employing offline terminal-side validation. Thirdly, in order to get a credential, an anonymous connection must be established from a user device to the nymble manager. The implementation of such anonymous channel is far from straightforward especially in case of a smart card or a smart phone as a user device. Lastly, the user device must possess the clock to chose the right credential which is associated with the current time epoch. In case of a passive smart cards as a user device, this could be especially difficult.

Therefore, we resort to a custom and relatively simple yet privacy-preserving blacklisting scheme. It is based on (inherently) homomorphic properties of the underlying encryption scheme in use. More specifically, the following property is exploited:

$$E(x \cdot r) = E(x)^r, \tag{2}$$

where for clarity and conciseness x represents the TA-side pseudonym P_i^A (see Table 1), r is a nonce value as given in equation (1). Therefore, $E(x \cdot r)$ corresponds to the session pseudonym SP_j. The notations used in this section together with the respective associations are summarized in Table 2.

The terminal-side blacklist (BL) contains a set of blacklisted static pseudonyms $\{y : y \in BL\}$, which are checked against during the e-ticket verification procedure. After mutual authentication (see Section 5.6), an e-ticket presents its Session Pseudonym $E_{k_{ta}^+}(x \cdot r)$ to a terminal along with the encrypted nonce value $E_{k_{ta}^+}(r)$ used for masking. Session pseudonym and the encrypted nonce form the so-called *Session Pseudonym Tuple (SP tuple)*:

$$SPT \leftarrow \big(E_{k_{ta}^+}(x \cdot r), E_{k_{ta}^+}(r)\big).$$

Table 2: Blacklist check: notations used.

Notation	Meaning/Association
x	an e-ticket static pseudonym, P_i^A ;
y	a blacklisted x;
$BL : \{y\}$	a blacklist (a set of y);
r	a random nonce;
$E\,(x \cdot r)$	a session pseudonym, SP_j;
$\big(E(x \cdot r), E(r)\big)$	a session pseudonym tuple (SPT).

Having obtained this tuple, the terminal can use the homomorphic property (2) to perform blacklist check. For this, it creates an auxiliary temporary check set C and computes its elements as follows:

$$\forall y \in BL, E_{k_{ta}^+}(r) \in SPT \; : \; c \leftarrow E_{k_{ta}^+}(r)^y. \tag{3}$$

Then a terminal pairwise compares the computed c elements with the delivered Session Pseudonym:

$$c \overset{?}{=} E_{k_{ta}^+}(x \cdot r) \;\; \forall c \in C. \tag{4}$$

If a match is found, the e-ticket is in the Blacklist set BL and must be rejected.

Therefore, due to the randomized nature of session pseudonyms, terminals are prevented from tracking valid e-tickets. Should an e-ticket be on the blacklist, however, the algorithm would find a match (see equation (4)) and the user would be prohibited from entering the public transportation network.

In order to boost the performance of the black list checking procedure, an e-ticket can additionally deliver its k-anonymous identifier [23] (signed by the transport authority) to a terminal after mutual authentication substantially lowering the search time over the respectively partitioned black list.

Choosing an Appropriate Encryption

As an example of an encryption function possessing the homomorphic property (2), the scheme based on the intractability of the Discrete Logarithm Problem[2] (DLP) can be used. Thus, $\forall x \in \mathbb{G}_q : \mathbb{G}_q \subset \mathbb{Z}_p^*$ (with p, q being large primes, $q|p-1$) the encryption can be written as $E(x) = g^x \ (mod\, p)$. A session pseudonym (see equation (1)), therefore, can be expressed as:

$$SP_j \leftarrow g^{x \cdot r_j} \qquad (mod \;\; p), \tag{5}$$

where x is an e-ticket pseudonym, r_j is a session nonce generated to mask x; $r_j, x \in \mathbb{G}_q$.

The homomorphic property (2) can then be expressed as follows:

$$\begin{aligned} E(x \cdot r) &= g^{(x \cdot r)} \\ &= \big(g^x\big)^r \qquad (mod \;\; p) \\ &= E(x)^r. \end{aligned}$$

In order to enable efficient singulation (correlating different session pseudonyms SP_j to a single static one P_i^A, see Section 5.4), a trapdoor due to Okamoto-Uchiyama [19] can be used. That is, knowing the factorization of the prime p, the discrete logarithm can be efficiently computed. The choice

[2] DLP follows from the hardness to extract x out of g^x in $\mathbb{Z}_p^*$ (see, for example, [9]).

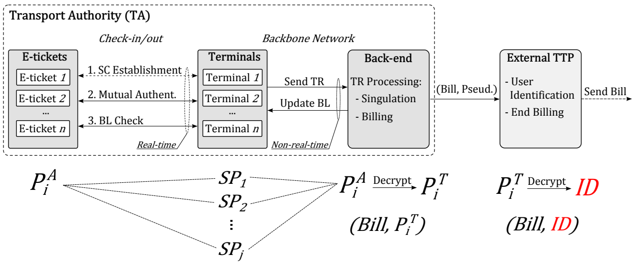


Figure 3: A Privacy-preserving Framework. *TR* – Travel Record; *BL* – Blacklist; *SC* – Secure Channel.

of the underlying encryption scheme is not limited to the one presented above. In principle, any other (deterministic[3]) encryption function with homomorphic properties (2) can be used.

5.6 Privacy-preserving Mutual Authentication

Mutual authentication between an e-ticket and a terminal (during check-in/out events) is required for a number of reasons: *(1)* to ensure that an e-ticket can be queried only by a legitimate terminal (privacy), *(2)* a terminal should process the events triggered only by the legitimate e-tickets (correctness and integrity), and *(3)* man-in-the-middle attacks must be prevented (even though such kind of attack is considered to be inapplicable to NFC in practice, see [11]). For the current e-ticketing scenario, the fundamental challenge is essentially bootstrapping authentication process without a terminal being able to *(1)* distinguish between e-tickets, let alone *(2)* to identify them. Moreover, mutual authentication should not prohibit path recovery in the back-end, see Section 5.2. We suggest that such kind of mutual authentication is implemented using a specialized certificate-based approach. That is, a terminal provides its unique certificate, which is in turn signed by the transport authority (TA). An e-ticket can issue signatures as well using another certificate signed by TA. Unlike the terminal's signature, the one of an e-ticket solely proves that the latter belongs to a valid *ticket group* (e.g., a monthly or a yearly ticket, possibly with certain attributes like a student or an elderly) and does not reveal any identifiable information on each particular e-ticket. This can be done, for example, by using the concept of group signatures, see [17, 5]. The aforementioned challenge of bootstrapping the authentication process, therefore, can be solved by essentially checking the respective certificate chain. Unlike the conventional certificate-based authentication, the e-ticket can not be traced and uniquely identified due to its group signature.

6. EVALUATION

6.1 Concept Evaluation

The proposed privacy-preserving framework satisfies the requirements presented in Section 2 under the assumed attacker model (Section 4). Namely, observing outsiders and terminals can not identify and track (valid) e-tickets which corresponds to requirements *1.1*, *1.3* (privacy against terminals and external observers, see Section 4). Back-end can correlate different travel records pertaining to a certain static pseudonym but is not able to identify an e-ticket (requirement *1.3*). By design our system inherently allows for fine-granular billing (requirement *2*). To protect customer privacy, the billing procedure is distributed between the two non-colluding parties: the back-end (under control of TA) and the external TTP. Due to the loose-coupling between the back-end and terminals, the e-ticket validation can be performed locally at the terminal side thus satisfying requirement *3*. Furthermore, sensitive pieces of information pertaining to the valid e-tickets are not known to terminals which further enhances the privacy-friendliness of the system (especially in case one of the terminals gets compromised).

6.2 Practical Evaluation

In order to evaluate the most time-critical part of the system – check-in/check-out events handling – a respective front-end prototype of an e-ticketing system was created which consists of: *(1)* a user device (in form of an NFC-capable smart phone), *(2)* an NFC-reader, and *(3)* a terminal. More specifically, we implemented an e-ticketing app on Samsung Galaxy Nexus GT-I9250 smart phone. The terminal was represented by Raspberry Pi Model B (256MB RAM) connected via SPI (serial peripheral interface) to PN532 Breakout Board acting as the NFC front-end. The terminal was controlled by a commodity computer Dell Vostro 3700 with Intel Core i5 M 460 (2,53GHz), 8 GB RAM running Gentoo Linux, Kernel 3.14.4, where all terminal-side computations have been performed. In our implementation, Diffie-Hellman (DH) key exchange (2048 bit modulo length) was used to establish a secure channel between an e-ticket and a terminal (see Figure 3). The further communication was secured with the exchanged AES 256-bit key (AES in Cipher Block Chaining (CBC) mode). For mutual authentication (see Figure 3), RSA-based certificates (key length 2048 bits) were used with terminals possessing unique certificates and e-tickets sharing group certificates corresponding to a particular subscription group (e.g. students, etc.). The size of blacklists was varied from 100 to 10000 elements. The performance was assessed by measuring the total execution time required to serve a single check-in/check-out event (excluding the delay introduced by the underlying NFC channel), see Figure 4. The area corresponding to the acceptable customer experience in terms of the maximum execution time of 2 sec (see requirement 4 in Section 2) is respectively marked in Figure 4 (see point A). The performance can be further boosted by partitioning blacklists into groups with k elements each, where k is an optimally chosen value. That is, k corresponds to the additional group identifier (e.g., in k-anonymity fashion) and therefore must be relatively large. At the same time, it must still allow for an acceptable execution time (for the most part influenced by blacklist check). As Figure 4 suggests, k_{opt} could be equal to 1000 in our particular setting (see point B).

7. CONCLUSION AND FUTURE WORK

In this paper, privacy protection in the area of intelligent transportation systems based on e-ticketing (ITSE) was addressed. The shortcomings of the existing real-world systems and academic solutions have been shortly discussed. Then, our own solution was presented which allows for loosely-coupled architecture (for local e-ticket validation) and inherently supports fine-granular billing. The proposed framework was assessed against the four major requirements dis-

[3] In case of probabilistic encryption, the randomization factor must be additionally delivered to the terminal for blacklist check.

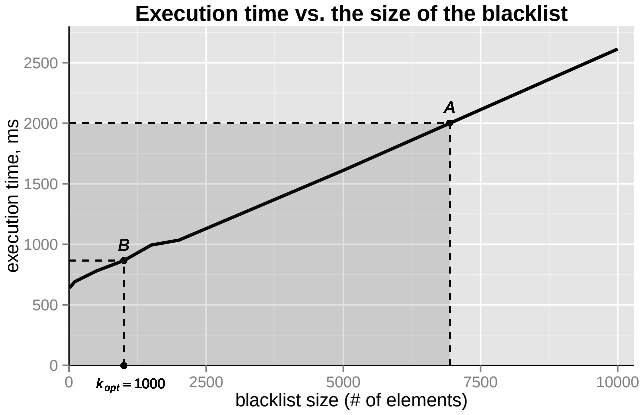


Figure 4: Execution time vs. blacklist size.

cussed at the beginning of the paper. The practical evaluation was presented as well.

As future work, the following issues are going to be addressed: *(1)* exploring more advanced cryptographic techniques preventing a malicious e-ticket (that is, an e-ticket not adhering to the protocol) from cheating during the blacklist check, *(2)* finding more efficient ways to perform a privacy-preserving, offline blacklist check, and *(3)* adding the evaluation on a smart card platform.

8. REFERENCES

[1] G. Avoine, *et al.* When Compromised Readers Meet RFID. In H. Y. Youm and M. Yung, editors, *Information Security Applications*, volume 5932 of *LNCS*, pages 36–50. Springer Berlin Heidelberg, 2009.

[2] F. Baldimtsi et al. Pay as you go. In *HotPETs 2012*.

[3] J. Camenisch *et al.* How to Win The Clonewars: Efficient Periodic n-times Anonymous Authentication. In *Proceedings of the 13th ACM*, CCS '06, pages 201–210, New York, NY, USA, 2006. ACM.

[4] J. Camenisch and A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *CRYPTO '02*, pages 61–76, London, UK, UK, 2002. Springer-Verlag.

[5] D. Chaum and E. Heyst. Group Signatures. In D. Davies, editor, *EUROCRYPT'91*, vol. 547 of *LNCS*, pages 257–265. Springer Berlin Heidelberg, 1991.

[6] A. de Panizza et al. RFID: Prospects For Europe. Item-level Tagging And Public Transportation. Report, eur 24416 en, EC, JRC, 2010.

[7] ECMA International. NFC-SEC. NFCIP-1 Security Services and Protocol. Cryptography Standard using ECDH and AES, 2008. White paper.

[8] European Parliament, Council of the EU. Directive 2010/40/EU of the European Parliament and of the Council of 7 July 2010 on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport. Official Journal of the EU, L 207, vol. 53, 6 August 2010.

[9] O. Goldreich. *Foundations of Cryptography: Vol. I - Basic Tools*. Cambridge University Press, 2004.

[10] G.-V. Hannover. HANNOVERmobil. `http://www.gvh.de/hannovermobil.html?&L=1`, 2013. Accessed online on 19.04.2013.

[11] E. Haselsteiner and K. Breitfuß. Security in NFC. Strengths and Weeknesses. In *RFIDSec'06*, Graz, '06.

[12] T. Heydt-Benjamin et al. Privacy for Public Transportation. In G. Danezis and P. Golle, editors, *PETs-2006*, vol. 4258 of *LNCS*, pages 1–19. Springer Berlin Heidelberg, 2006.

[13] J.-H. Hoepman and G. Huitema. Privacy Enhanced Fraud Resistant Road Pricing. In J. Berleur et al., editors, *HCC9/CIP-2010*, volume 328 of *IFIP AICT*, pages 202–213. Springer Berlin Heidelberg, 2010.

[14] ISO. ISO/IEC 7816-4:2005. Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange, 2005.

[15] P. Johnson *et al.* Nymble: Anonymous IP-Address Blocking. In N. Borisov and P. Golle, editors, *PETs*, vol. 4776 of *LNCS*, pages 113–133. Springer Berlin Heidelberg, 2007.

[16] F. Kerschbaum, H. W. Lim, and I. Gudymenko. Privacy-Preserving Billing for e-Ticketing Systems in Public Transportation. In *WPES'2013* , pages 143–154, New York, NY, USA, July 2013. ACM.

[17] A. Kiayias *et al.* Traceable Signatures. In C. Cachin and J. Camenisch, editors, *EUROCRYPT '04*, vol. 3027 of *LNCS*, pages 571–589. Springer Berlin Heidelberg, 2004.

[18] M. Mezghani. Study on electronic ticketing in public transport. Technical report, EMTA, May 2008.

[19] T. Okamoto and S. Uchiyama. A New Public-Key Cryptosystem as Secure as Factoring. In *EUROCRYPT '98, Espoo, Finland*, vol. 1403 of *LNCS*, pages 308–318. Springer, 1998.

[20] P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer Berlin Heidelberg, 1999.

[21] K. Peng and F. Bao. A Secure RFID Ticket System for Public Transport. In S. Foresti and S. Jajodia, editors, *Data and Applications Security and Privacy XXIV*, volume 6166 of *LNCS*, pages 350–357. Springer Berlin Heidelberg, 2010.

[22] A.-R. Sadeghi et al. User Privacy in Transport Systems Based on RFID E-Tickets. In C. Bettini et al., editors, *PiLBA-2008, Malaga, Spain*, 2008.

[23] L. Sweeney. k-anonymity: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, Oct. 2002.

[24] S. Tamkar *et al.* Identity verification schemes for public transport ticketing with NFC phones. In *STC'11*, Chicago, Illinois, USA, 2011. ACM.

[25] Transportation Research Board. TCRP Report 115: Smartcard Interoperability Issues for the Transit Industry. Technical report, National Academy of Sciences, 2006.

[26] P. Valley Metro. Platinum Pass Program. `http://www.valleymetro.org/employer_programs/platinum_pass`, 2013. Accessed online on 19.04.2013.

[27] VDV-KA KG. Spezifikation von Luftschnittstellen in einem VDV-Kernapplikations-konformen interoperablen Mobile Ticketing in Verbindung mit einer passiven Near Field Communication (NFC) Verkaufs- und Erfassungsinfrastruktur, April 2011.