Mass Transit Ticketing with NFC Mobile Phones

Jan-Erik Ekberg and Sandeep Tamrakar

Nokia Research Center, Helsinki
{Jan-Erik.Ekberg,Sandeep.Tamrakar}@nokia.com

Abstract. Mass transport ticketing with mobile phones is already deployed in many metropolitan areas, but current solutions and protocols are not secure, and they are limited to one-time or fixed-time ticketing in non-gated transport systems. The emergence of NFC-enabled phones with trusted execution environments makes it possible to not only integrate mobile phone ticketing with existing and future transport authority ticket readers, but also to construct secure protocols for non-gated travel eliminating many associated possibilities for ticketing fraud. This paper presents an architecture and implementation for such a system.

1 Introduction

In mobile handsets, the Near-Field Communication (NFC) radio standards [13] is by many seen as *the* user-friendly enabler for the implementation of payment, ticketing, access control tokens and other services typically existing in the user's wallet and key ring. Information gathered from a survey [26] predicts that 400 million mobile subscribers will use mobile ticketing on their devices by 2013. Such an uptake is not surprising considering the benefits of having a local "ticketing user interface" on the device. The interface can not only provide handy information like ticket expiry time, account balance and real-time traffic information but can also be used to purchase new tickets. Also, many people use their mobile device while waiting for and during transport for communication, browsing or reading. Thus the mobile phone is anyway present and ready to be used when a ticketing activity is required.

All over the world, transport ticketing has for years been implemented with wireless technologies, increasingly using the ISO / IEC 14443 [12] contact-less card standard. Common references to such technologies include the Felica®[1] system in Asia, the Oyster card[2] e.g. in London, and protocols like MiFare[3] developed by NXP semiconductors.

This paper presents a trial implementation of a ticketing architecture for mobile phones that implements a new take on how to bind the ticketed identity to the place and time when the journey begins and ends. With this feature we enable deployment of complex fare and discount calculations, e.g. distance-based

[1] Felicity Card - www.sony.net/Products/felica
[2] https://oyster.tfl.gov.uk/oyster/entry.do
[3] http://mifare.net

L. Chen, M. Yung, and L. Zhu (Eds.): INTRUST 2011, LNCS 7222, pp. 48–65, 2012.

ticket pricing. Earlier, this has been practical only in gated transport systems, but not in systems with non-gated parts. Central to our approach is the use of the trusted execution environment in the phone not only for securing digital signature generation but also for local protocol state enforcement. To meet the very tight timing constraints for contact-less ticketing especially in gated transport, we streamline all NFC interaction and revisit the ticketing credential, where we leverage the message recovery property of RSA signatures not only to achieve size-efficiency but also as a privacy-implementing primitive.

We begin in Section 2 by listing related work. Section 3 outlines processes in ticketing systems and Section 4 lists functional and security requirements. Our system architecture is introduced in Section 5. Section 6 outlines our ticketing credentials. Sections 7 and 8 present the ticketing protocols in gated and non-gated systems. We discuss implementation in Section 9, and requirements analysis in Section 10.

2 Related Work and Background Technologies

NFC ticketing [11] — a project by RFID Lab of the University of Rome "Sapienza" provides a public transport NFC ticketing prototype for usability research. Their system is implemented as a Java application, and provisions data over SMS.

A thesis work by Kooman [17] presents a cryptographic model for using NFC enabled mobile phones for public transport payment. Strong privacy is achieved by selective blinded attribute verification between the phone and a validating device. These protocols are not size-efficient, and no implementation was presented in the thesis.

Since 2001, public transport ticketing using Short Message Service (SMS) has been deployed in various cities around the world e.g. Helsinki, Prague and Rome. Current systems are open to ticket copying attacks [20].

2.1 Trusted Hardware

For electronic ticketing, the default hardware element is the smart card, typically adhering to the ISO / IEC 7816 [16] interface primitives and to ISO / IEC 14443 [12] for the wireless interface. Smart card security confirms to the GlobalPlatform Card Specification standard [24], which defines key management and provisioning. Applications for smart cards are developed with the the JavaCard programming environment.

During the last decade, trusted execution environments (TEE)s have emerged based on the general-purpose secure hardware. These have been incorporated into mobile phones, and are widely deployed. Designs like Mobile Trusted Modules (MTM) [7], M-Shield [25] and ARM TrustZone [3] are available.

The user device TEE chosen for this work is the On-board Credentials (ObC) architecture [18], deployed in all Nokia Symbian ∧3 phones to date. ObC uses ARM TrustZone on a processor manufactured by Texas Instruments. ObC relies on the underlying hardware to isolate credentials from the operating system.

Additionally, it provides a provisioning system and a byte-code interpreter with an extensive cryptographic Application Programming Interface (API) for the implementation of credential algorithms (ObC programs).

2.2 NFC

NFC is a wireless Radio Frequency Identification (RFID) technology standardized in ISO / IEC 18092 [13] and ISO / IEC 21481 [14]. An industry consortium, the NFC forum [4], provides compliance-testing and additional standards for NFC use. NFC encompasses three different types (A, B and Felica) of radio communication, with theoretical speeds ranging from 106 to 424 kbps. Many NFC readers readily available in Europe are limited to 106 kbps.

The lower layers of NFC include no communication security primitives. It is also well known that NFC technology is susceptible to e.g. both eavesdropping and man-in-the middle attacks [6], despite the fact that NFC is a short-range radio technology. To date, some 20 phone models from different manufacturers support NFC[5]. Only a handful of models embed a secure element or TEE that can be used for securing a ticketing transaction.

3 On Transport Ticketing

In recent years, new approaches [22,4] to electronic transport ticketing has emerged, stimulated by the ongoing mass deployment of EMV (EuroPay, MasterCard and Visa) contact-less credit cards [9] in many countries. From the perspective of the public transport authority, it has been identified that the cost of fare collection, i.e. operating the ticketing system and collecting the money from users is significant [19]. Thus, one option is to outsource this function to e.g. credit card companies, telecom operators, or any other stake-holder that is prepared to take on such a responsibility. The outsourcing option does not however apply to *fare calculation*, i.e. determining the price of a given trip, since it is intimately tied to the transport function itself.

In currently deployed non-gated ticketing systems, it is often hard to accurately determine journey length or duration, since it is impractical to collect data about the journey endpoint. When the end event cannot be collected, a typical solution is to define so called transport zones and to assume that the user buys an appropriate ticket for his travel under the threat of randomly applied ticket inspection. These mechanisms are coarse-grained, and often difficult to resolve for users traveling along unfamiliar routes. Another common shortcoming is that the incompatibilities between the ticketing principles (zones, validity time, gated vs. non-gated) and the deployed ticketing technologies of local transport within a single metropolitan area (e.g. between underground, buses and local trains) often makes combination ticketing difficult.

[4] `www.nfc-forum.org`
[5] `http://en.wikipedia.org/wiki/Near_field_communication`

Our work further develops the approach taken with e.g. credit-card ticketing. In this context, a transport user fundamentally is represented by a ticketing identity, or in credit card terminology by his Primary Account Number (PAN) [15]. The identity is presented to the transport authority at system entry (and exit), where the PAN is bound to the time and location of those respective events. A proper ticketing system also allows the ticket to be inspected during the journey of the user.

We construct a system where the PAN can be securely bound to the place and time of the ticketing "tap", where the mobile device touches a gate or bus stop touch point. If the resulting transaction evidence can be moved reliably to a back-end computing system, then the fare calculation for a specific journey, undertaken by a user / PAN is trivially achievable. We will need security-enabled protocols, since the transport user has a clear incentive to cheat. We also need to arrange for the transfer of transaction evidence to back-end systems in a manner that is cost-effective not only in metro stations with tens of thousands of users passing each day, but also at bus stops with, say only 10 people boarding a bus every workday morning.

4 Requirements

Many important requirements for a transport system are functional. In gated mass transport, rapid people throughput is a paramount consideration, and the Smart Card Alliance sets the unofficial maximum transaction time to 300 ms per gate entry [1]. This time constraint in practice eliminates the possibility of online verification supported by a back-end. Since a tap-and-hold transaction, say 700ms, is significantly more difficult for the user, speedy tap transaction times are also advantageous in a non-gated system. Additionally, in contemporary NFC ticket readers, hardware acceleration support is often available only for RSA, ruling out the use of more size-efficient cryptographic primitives like elliptic-curve cryptography.

In non-gated systems, the tapped "terminals" are located e.g at bus stops, train stations and the like. A user is required to tap a terminal prior to vehicle entry and after he or she finishes the journey. For cost saving reasons, such terminals shall not require electricity or back-end connectivity. Deploying tamper-resistant processor chips with NFC antennas (contact-less smartcards) for this purpose is one cost-efficient solution. Protocols and processes shall also be designed to minimize operating and maintenance cost of such terminals.

The security requirements for public transport ticketing stem from countering fraud and maintaining user privacy. Mayes et. al. [21] provide a good topical introduction to ticketing and fraud control. In terms of revenue loss, the main system fraud are individuals that enter and exit the system without paying. Thus, it is not surprising that the biggest reduction in public transport fraud to date happened with the introduction of reliable gates and machine-readable tickets or physical tokens, since this eliminates traveling without a ticket or with a cheaper minimal distance ticket in the gated systems. A similar paradigm

shift has not yet occurred in the non-gated systems. We set our target to implement a ticketing system that enables distance-based fare calculation also for the non-gated travel, and the main security requirements for such a system can be formulated as:

R1. The identity of the traveler must be determined off-line and cryptographic evidence must be produced during the transaction to provide non-repudiation.
R2. Eavesdropping and replay attacks shall not provide an attacker the possibility to impersonate another user.
R3. It shall not be possible to produce starting point (or ending point) evidence without being present at that location at the given time. Users shall not be able to tap in only when they see a ticket inspector.
R4. It shall not be possible to withhold evidence and hinder it from eventually reaching the back-end fare calculation engine
R5. It shall not be feasible to confuse the ticketing system by replacing / adding fake bus and train stop passive tapping terminals. These are placed in locations that cannot be assumed to be well guarded.
R6. Ticketing credentials shall be protected against modification. Since the identity is traveler-specific, credential migration needs to be controlled.

Additionally user privacy shall be maintained. In a ticketing system using a radio channel (NFC), the main privacy threat is eavesdroppers performing *device tracking*, to determine the movements of particular users. We assume that the back-end systems adhere to standard privacy norms and legislation in terms of handling user data, and back-end data privacy issues are not considered further in this paper.

5 Architecture

In this section, we briefly present a generalized architecture for a PAN-based ticket scheme as outlined in Figure 1. The *transport authority* operates the vehicles and also provides an integrated network for its gated ticket readers. The gated NFC readers are assumed to be connected to a back-end system. Therefore, these readers can receive information like certificate revocation lists (CRL)s which they refer to during user verification. All the information exchanged during such verification is collected as transaction evidence and forwarded to a back-end processing unit, e.g. a fare calculation engine. This database can e.g. be maintained by the transport authority.

The transport authority is also responsible for distributing and maintaining the terminals (i.e. smart cards) for non-gated travel. We assume that these smart cards are physically and firmly attached to their location, and also that these smart cards are tamper-resistant.

The *accounting authority* is responsible for fare collection from the users. A transport authority can simultaneously be connected to several accounting authorities. All users have a relationship with one accounting authority, in the form of a prepaid or credit-based user account. Exactly how user invoices are

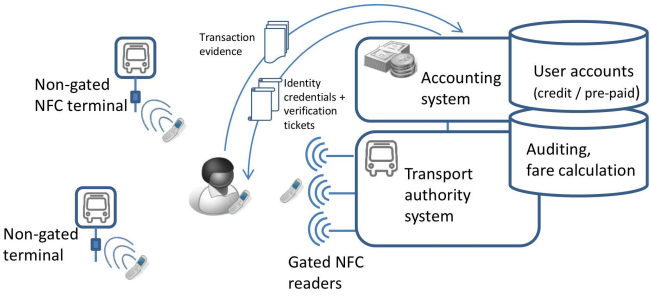

Fig. 1. Overall architecture

cleared by the accounting system is not relevant here, although it will affect e.g. the logic for black-listing users.

The accounting authority is also responsible for generating ticketing credentials and provisioning secrets to the TEE in user devices. In our system, we use the proprietary ObC provisioning system [18] for this activity, but e.g. GlobalPlatform compliant data and program provisioning [24] can also be used. The cryptographic validation of transport evidence and user backlisting are also likely to be the reponsibility of the accounting function.

Although the distinction between accounting authority and transport authority includes security-relevant interactions, e.g. related to user device blacklisting and auditing, these are not further explored in this paper. We use the term "back-end" as a collective term for all back-end operations.

6 Ticketing Credentials

It is typical not to use X.509 certificates in ticketing and payment systems, e.g. EMV [8] specifies its own size-optimized identity certificate. We go one step further in the minimization effort, and exploit the message recovery property in RSA signatures. In this way, the user identity and the public key of the user device can be extracted from the signature itself, thereby avoiding the need to transfer the device public key separately. Additionally we add a flavor of privacy protection to the construction.

The certificate outline is shown in Figure 2. A standard RSA signature consists of a private key exponentiation of an algorithm identifier and a 20 byte hash (SHA1) as payload, padded according to PKCS#1.5. In our ticketing credetials, we concatenate as much certificate payload to the hash as we can (inside the RSA exponentiation) to minimize the the overall certificate length. The shortest possible PKCS#1.5 padding is 11 bytes, so e.g. for a 1024 bit key we can save around 90 bytes with this approach. Like in the EMV standard, the certificate carries only a bare minimum of attributes. We include only the expiry time, the public key modulus of the client RSA key, and the user's identity, i.e. his primary account number (PAN) [15]. A truncated SHA-256 hash is put first so

that it always fits inside the RSA exponentiation. The public key exponent is not included and is fixed at 0x10001, which is compatible with both TCG and EMV standards.

To achieve address privacy, we add an additional twist. In case all certified data does not fit into the private key exponentiation of the certificate authority (CA) signature, we use the hash of the certificate data as a symmetric encryption key, and encrypt any overflowing bytes with AES in counter mode. With this packaging, the back-end can produce, for the user, a set of (short lived) certificates for a single RSA key such that their expiry times are e.g. set one second apart. Without knowledge of the CA public key, the certificates will reveal no plaintext data, and two certificates from the same set cannot be linked.

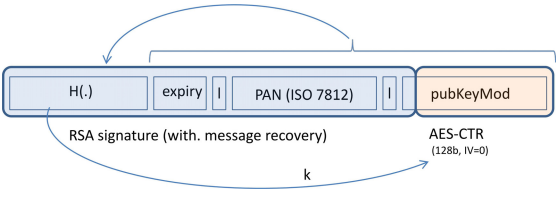

Fig. 2. Certificate with message recovery

This approach requires that the ticketing system treats the CA "public" key for user credentails as a secret. This is possible, since it will only be distributed to gated ticket terminals and ticket inspector devices. In both of these device categories, the CA key can with a high likelihood be remotely provisioned in a secure manner and locally be treated as a secret.

In our system, we use a CA key of 1408 bits, which is the current estimate for the minimum RSA key length accepted by EMV standards from 2013 onwards. The short-lived ticketing signature keys are 1024 bits to keep the certificate size small. With these parameters and assuming a PAN number length of 16 digits (typical credit card number), we end up with a certificate size of 176 bytes.

7 Gated Ticketing Protocol

The use of the ticketing credential is visible in the ticketing protocol at a gated system entry or exit. Figure 3 shows this very standard procedure for identity verification. As preconditions we assume that

1. The phone generates an RSA key pair, a transport key, inside its TEE. The phone sends the public component to the back-end, which generates a set, say 50, short-term transport "ticketing credentials" (with e.g. a one-week expiry time) for that public key component and the user's PAN.
2. The user's debit / credit rating is in order, and the device / user has not been blacklisted by the back-end since the credential was issued. The gated readers are updated with the latest CRLs.

3. For the transaction, the user's device always selects a ticketing credential that has not been presented to any ticket reader before.
4. The gated reader, being connected to the back-end, will eventually send any transaction evidence with timestamps and other context information to the backend for further processing and fare calculation.

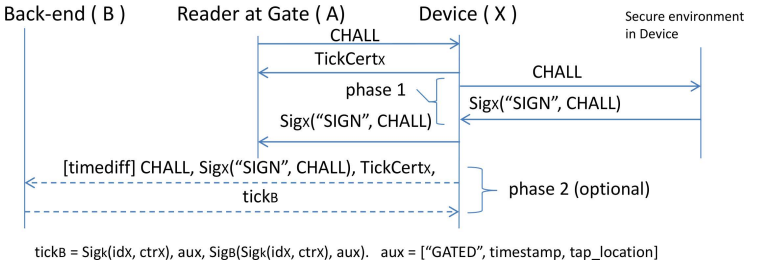

Fig. 3. Gated tap protocol

The operation proceeds as follows. The user touches the gate. Within the NFC transaction, the reader A sends a challenge to the device X. The challenge contains at least a nonce and a reader / station Id. The client immediately responds with the selected ticketing credential, and in parallel or subsequently signs the challenge with the current transport key. Once computed the signature is then returned to the reader. Since the ticket reader knows the CA public key, it can validate the credential. With the recovered device public key, in can further verify the signature on the challenge. On successful verification the gate is opened.

With an optional handshake between the user device and the back-end servers the user device can receive a ticket inspection token signed by the back-end, in return for the uploaded transaction evidence.

8 Non-gated Ticketing

The non-gated PAN-based ticketing variant uses the same basic building blocks as gated ticketing, i.e. signatures and certificates. Minimizing NFC transaction time is still of importance, thus we will not add the cost of mutual authentication over NFC. The off-line communication overhead for such authentication is at least two times the public key modulus size (i.e. optimized certificate + signature) which adds an extra 300 ms, see Section 9.

Compared to the gated protocol variant, the main new property that we add for non-gated operation is a device-specific counter that can be attested by a signature. Such a primitive is trivially implementable with ObC or an embedded secure element with JavaCard™. Even with TPMv1.2, which is not programmable, such a primitive can be constructed by attesting to a repeatedly extended PCR [23]. We believe this section will show the usefulness of this simple TEE construction.

8.1 TEE Operation

Figure 4 summarizes the TEE operation. The *sign challenge* is the basic signature primitive, used in gated ticketing. This operation signs an incoming challenge with the transport key x. All signatures generated with x contain a four-byte prefix that identifies the signature context, i.e. the invoked TEE command.

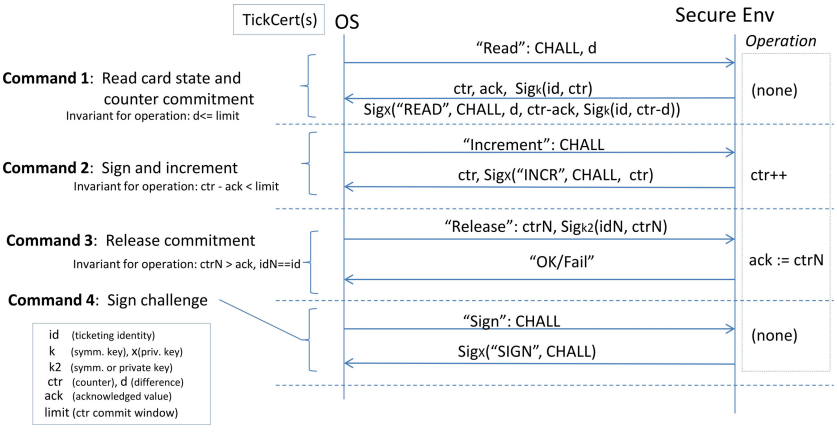

Fig. 4. TEE operations

The *sign and increment* is the augmented signature primitive, that signs a challenge, but also includes a counter value in the signature. The command primitive updates the monotonically increasing counter at every use. The TEE program state includes a counter window, i.e. the amount of *sign and increment* signatures are limited by the current size of that window. Only an external release, implemented by the *release commitment* command and secured by signature key $k2$, can open up that window and allow more counter-bound signatures to be made. This will be one incentive for user devices to report ticket evidence in non-gated operation. We use a symmetric, AES-based MAC for the signature with $k2$, since ObC-enabled devices on the market do not support public-key RSA operations inside the TEE.

The *read card state and counter commitment* is used for ticket inspection and for retrieving a counter value commitment, $Sig_k(id, ctr)$. This attribute uses a symmetric signature primitive with a key k that is shared between the device and the back-end and possibly ticket inspectors. Key k can e.g. be derived from a master secret with a key diversification algorithm: $k = KDIV(master, id)$. Using the d input parameter, a commitment for the current, or any past counter value can be requested. The difference between the current counter value and the counter indicated by the commitment value is visible in an additional asymmetric signature used with ticket inspection. That signature also binds the number of counter values used, but not yet released by a back-end server. For the device's

benefit the command also returns the current counter value and the last value acknowledged by the back-end server.

8.2 Non-gated Protocol

The non-gated ticketing operation is presented in Figure 5. The user device X taps a *terminal* R (labeled "Bus Stop Card" for clarity), and then performs an internal operation eventually followed by a reporting activity towards the accounting authority. We will look at these different stages in order:

In *phase 0*, which can be done inside the device at any time preparing for the next ticketing event, the device X invokes the *read card state and counter commitment* command with $d = 0$, i.e. it retrieves the commitment $Sig_k(id_X, ctr_X)$ for its latest counter value. The rest of the data returned by the TEE command is not used at this point.

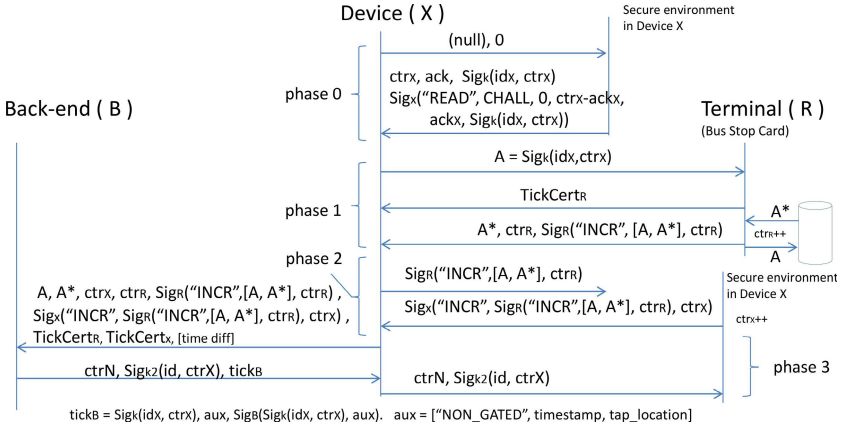

Fig. 5. Non-gated tap protocol

The tap on the terminal R, i.e. *phase 1*, is more or less exactly the gated protocol, run by device X as the challenger. As a result, the terminal R will return its device certificate $TickCert_R$ to X as well as a signature with counter binding for the terminal's counter ctr_R over the user device challenge which was its counter commitment $Sig_k(id_X, ctr_X)$. Note that the terminal R also adds to the signed response some auxiliary data $A*$, which we will discuss later. For now it suffices to note that $A*$ will have to be transported along with the terminal signature Sig_R towards the back-end in order for that signature to be verifiable.

The TEE logic in terminals R can be equivalent to the one in user devices X. However, for this protocol, the counter limit in terminals is not really used, and neither is the read operation, so in essence only the *sign and increment* operation is necessary on terminals B.

As $Sig_k(id_X, ctr_X)$ cannot be resolved by entities without knowledge of k, and since ctr_X is always a fresh value, we can deduce that the user device X maintains unlinkability when transmitting $Sig_k(id_X, ctr_X)$. On the contrary, terminal R has no privacy requirement, thus $TickCert_R$, which is of the format described in Section 6, can be signed by a CA key whose public component is truly public, e.g. known by device X. Based on R's ticketing certificate and signature, X can determine the identity and validity of R - an important protection against a specific denial-of-service attack (requirement R5). As we can adjust the lengths of A, $A*$ and ctr_R to be roughly of the same size as the challenge in the gated operation, we will see that the tap time is composed of the transmission cost (250ms) and the speed of the RSA exponentiation in a legacy smart card which unfortunately can be 400ms or higher [2,22].

In *phase 2*, device X re-invokes its own TEE, and issues the *sign and increment* command, with the *phase 1* protocol response received from R as the challenge. This operation binds the current identity and counter state of X to the identity and counter state of R in a non-repudiable way.

We assume that the mobile phones X have back-end connectivity, and soon after *phase 2* is completed, device X in *phase 3* takes all data available from phases 0-2 and sends these to a back-end over a server-authenticated TLS channel. In addition, the device will send its estimate of the time that did pass between phase 1, and the first message of *phase 3*. We do not assume a secure clock inside the TEE, thus this value is a best-effort service augmented by the absolute and verifiable time when the back-end server receives the *phase 3* message. The server can identify the parties X and R present in the transaction, and validates all data related to the transaction, including the fact that the counter values ctr_X in commitment $Sig_k(id_X, ctr_X)$ and $Sig_x(\ldots, ctr_X)$ match. Of special interest for further auditing by the back-end will be the respective identities, counters and estimated transaction time.

As a response, the back-end server will return the data for a *release commitment* in X, typically for the used counter value ctr_X. The back-end will also return some information to X for ticket inspection. With respect to the evidence collected, a back-end auditing function will execute a process for dealing with "lost" counter values, i.e. counter values of X that are never reported back to the system.

We see that if X consistently suppresses the *phase 3*, transaction will eventually cause the TEE to "lock up" due to window-size exhaustion. This is the main enforcement for reporting evidence. Nevertheless, we did not want to put the immediate availability of the back-end communication channel from X in the critical path for ticket inspection and neither did we want to only rely on local window enforcement to suppress last-minute tap reporting. We return to these issues in the next two subsections.

8.3 Non-gated Ticket Inspection

In a PAN-based system, ticket inspection takes a different role than it typically has today. What the inspector will determine is that the device has tapped

on a terminal R that is consistent (in time and place) with the ongoing user journey. By default, not much can be determined about the final destination of the transport user, if that information is not provided as an add-on commitment by the user's device X. Such information should not be necessary, though.

The default inspection protocol is outlined in Figure 6, and relies on the inspection data $tick_B$. In essence, $tick_B$ is a statement, signed by the back-end, that includes the counter commitment for X, i.e. $Sig_k(id_X, ctr_X)$, as well as auxiliary information containing e.g. the estimated time of the transaction and the tap location (identity of terminal R). The ticket validation device V will challenge the device X. The difference between the current counter value and the counter value at the time of the tap is represented as d. Typically this value is 1, if we validate the last tap. The device invokes the *read card state and counter commitment* TEE command for that d, and returns the signed response Sig_x to the validator along with $TickCert_X$ and the values d and $ctr - ack$, since they are required for signature validation. From this information the validator device can determine that $Sig_k(id_X, ctr_X)$ has been emitted from this specific device X, and it can hence trust the time and location information present in $tick_B$ and use those values to determine whether the user being present in this vehicle is consistent with the provided evidence.

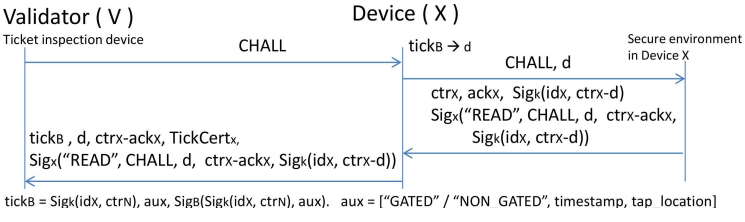

Fig. 6. Ticket inspection

Ticket inspection in the default scenario is mostly resistant against device tracking. The commitment will be visible, thus an eavesdropped message exchange during the tap can be matched with the exchange during inspection. Also the difference between the current counter value and the ticketed counter value (d), as well as the available counter window in X are visible, however the resolution of these values are too low to be useful for tracking. Ticket inspection can also be conducted before X has contacted the back-end. This is described in Appendix A.

8.4 Transaction Evidence Feedback

To identify devices X whose TEE has been broken (especially the counter window feature), and to provide a way to generally audit non-reporting devices, the stand-alone smart cards terminals R collect a log of past transactions, containing terminal R's counter value at the time of the transaction (4 bytes), the

challenge from a device X (20 bytes) and the number of times this record has been reported (1B). E.g. 100 KB of card storage can accommodate a ring buffer of 4000 such records. These logs are assumed to be available for an evidence collector, e.g. ticket validation devices, but the protocol for that information retrieval is not considered here. However, for each ticketing tap, the card will at random select several (at least 2) of these past transactions into a tuple set $A*$ and bind that data to the return message signature, forcing the tapping device to convey $A*$, along with its own evidence A, back to the back-end. If the length of $A*$ is two records, then in normal (non-attacked) operation, the back-end will typically receive each tap challenge three times even without explicit evidence collection from terminals R.

The challenge commitments $Sig_k(id_X, ctr_X)$ of all participating devices can be calculated a-priori by the back-end server, since it knows the key k and the last used counter value. If such future commitment values are put in a database, the back-end can resolve identities from the $A*$ commitments even if the device X that generated a tap does not immediately report it. The main benefit of this system is to catch devices that repeatedly use the same non-reported commitment as well as devices where the counter window is broken, thereby allowing them to forward the counter and related commitments without being forced to report any evidence to the back-end.

As there is no authentication of the user device X that touches R, and since terminal R has no notion of time, the event log of the card is erasable by repeatedly performing dummy tap operations. Still, for e.g. 4000 records and a transaction time of 300-600ms, a card history erasure takes a full 20-40 minutes to complete. So even though the card event log theoretically is erasable, we claim that the feedback system motivates its existence in terms of being a practical deterrent mechanism against the active fraudster.

9 Implementation and Measurements

This paper reports on ongoing work for building a ticketing system for trialing the presented concepts in a real environment. At the time of writing, the gated protocol implementation is fully functional. Our terminal (smart card) implementation is also ready, but for non-gated operation our phone client is still under development. Thus, non-gated measurements are done between a PC and the smart card, whereas all other values are measured between a Symbian phone and its respective counterparts.

We have implemented the ticketing application on a Nokia C7 phone, which includes necessary hardware and software support for both the TEE and NFC capabilities. The terminal is a NXP SmartMX card, produced in 2009. The TEE application for the terminal is written in JavaCard 2.2.1, and the one for the phone in ObC bytecode. The NFC channel is operating at 106 kbps. Our reference terminal/reader is a Linux PC, running Ubuntu Maverick with an NFC reader (ACR 122U) connected to it. The open-source project *libnfc* provides the Linux NFC stack, whereas a "gate application" controls the protocol flow.

In gated operation the NFC communication runs in peer-to-peer mode [10], whereas in non-gated the phone operates as a card reader. Back-end reference servers, e.g. for transaction evidence collection and certificate provisioning, are traditional LAMP setups, but those interfaces are not time-critical and therefore not measured.

Table 1. Time taken for each ticketing protocol execution

Protocol	Target discovery	Message 1	NFC Hand Over	Message 2	Process 1	Message 3	Process 2	Total
Gated	36 ms	24 ms	18 ms	94 ms	[∼60 ms]	74 ms	10 ms	256 ms
Non-gated	36 ms	14 ms	[18 ms]	∼ 50 ms	≥ 297 ms	∼ 102 ms	—	∼ 499 ms
Ticket Inspection	36 ms	14 ms	18 ms	290 ms	20 ms	—	—	378 ms

Table 1 presents the time taken by different events in each NFC transaction to complete. Messages 1, 2 and 3 represent PDUs that are exchanged in each protocol. For instance, in the gated protocol these represent $CHALL$, $TickCert_x$ and Sig_x, respectively. Similarly Process 1 and 2 represent the TEE cryptographic operations. For some protocols, the TEE can be consulted in parallel with transmission which eliminates those timings from the critical path in the protocol.

The main insight from the measurements is that for now, the TEE invocation cost (60-70 ms in ObC) in legacy devices is insignificant compared to the cost caused by very poor NFC throughput (only around 1 byte/ms).

10 Requirements Analysis

We are likely to fulfill the main functional arguments. For gated transactions we already know that the implementation meets the 300ms transaction time goal. The non-gated protocol shows that we can implement passive terminals with contact-less smart card technology, and we believe that the transaction time for tapping the terminals can be kept acceptable.

For security, the **isolation** of credential secrets and all operations on them, i.e. the RSA private key, its use as well as counter management is for user devices X guaranteed by the TEE and the ObC, as is outlined in [18], and for terminals by the smart cards which are integrated components with tamper-protection. The same reasoning holds for the confidentiality and integrity of credential data provisioning, say for the secret k. In ObC this is guaranteed by its own provisioning protocol [18], and for smart cards the use GlobalPlatform protocols [24] ascertain the same properties.

The security requirements are met as follows: The identity of the traveler can at any time be proven based on a signature-based challenge-response protocol and a ticketing credential for the used key. This can be done off-line (R1). Impersonation is not possible due to the interactiveness of all protocols involving the

user device (R2). In the current implementation the ticketing identity is bound to the TEE and non-migratable. The ticketing credentials are signed by the CA, thus their integrity is assured (R6).

We acknowledge the potential cryptographic danger of using PKCS#1.5 padded RSA signatures with recovery for our certificates[5]. We still chose the approach, since we need to target both privacy and size requirements. With more bytes to spare in our protocols, our preferred solution is to use that additional transmission budget for mutual authentication, which can solve the privacy issue more reliably than by applying a safer (and bigger) signature primitive with message recovery.

The tap location evidence is bound in time by the counter of terminal R. For very remote locations with little traffic, an interactive relay attack to the terminal may be successful. Not reporting an executed tap is very likely to be caught by auditing based on the terminal feedback channel $A*$ through other devices. (R3). The TEE will only cooperate within a limited window if evidence is withheld, thereby suppressing the release commitments from the backend (R4). The terminals are authenticated by the user devices tapping them, so impersonation attacks on terminals is not feasible (R5).

For privacy, our user-device to back-end connections are run over server-authenticated TLS. For the NFC transactions, the client is given many "anonymized" certificates that are valid simultaneously. We also assume that the system is set up with two CAs, one for certifying users' devices, which is kept secret as outlined in Section 6, and a public CA key used for certifying terminal cards. With these preconditions, if the client devices vary the certificates they use, an eavesdropper cannot bind the NFC ticketing transactions to any given PAN or user — there is no plaintext data visible in the certificate nor in the challenge - response protocols. The Nokia C7 phone that we use also randomizes the NFC radio identity, eliminating the address tracking threat also on lower protocols levels. Also ticket inspection, in the presence of $tick_B$, will reveal no user data on the NFC radio.

11 Future Work and Conclusions

This paper provides a protocol framework for combining gated and non-gated ticketing into one coherent system. Some parts of the architecture are already implemented, and we are working on the rest of the system to get it ready for trialing. Especially the non-gated protocols leverage the TEE present in mobile phones to securely bind the ticket system state of the phone to the one present in the terminal. In combination with the TEE-enforced transaction window limit we can reliably use the phone as a reporting channel. To our knowledge, we are also the first to report on a RSA-based system that deploys full identity verification (with certificate and signature) for NFC ticketing within 300ms using keys of acceptable length. Our system additionally provides protection against device tracking, also a first in this context.

References

1. Smart Card Alliance. Transit and contactless financial payments: New opportunities for collaboration and convergence. A Smart Card Alliance Transportation Council White Paper (October 2006), http://www.smartcardalliance.org/resources/lib/Transit_Retail_Pmt_Report.pdf (accessed: August 2011)
2. Anderson, R., Bond, M., Choudary, O., Murdoch, S.J., Stajano, F.: Might Financial Cryptography Kill Financial Innovation? – The Curious Case of EMV. In: Danezis, G. (ed.) FC 2011. LNCS, vol. 7035, pp. 220–234. Springer, Heidelberg (2012)
3. ARM. Technical reference manual: Arm 1176jzf-s (trustzone-enabled processor), http://www.arm.com/pdfs/DDI0301D_arm1176jzfs_r0p2_trm.pdf
4. Brakewood, C.E.: Contactless prepaid and bankcards in transit fare collection systems. Master's thesis, Massachusetts Institute of Technology (2010), http://hdl.handle.net/1721.1/60796
5. Coron, J.-S., Naccache, D., Stern, J.: On the Security of RSA Padding. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 1–18. Springer, Heidelberg (1999)
6. de Koning Gans, G., Hoepman, J.-H., Garcia, F.: A Practical Attack on the MIFARE Classic. In: Grimaud, G., Standaert, F.-X. (eds.) CARDIS 2008. LNCS, vol. 5189, pp. 267–282. Springer, Heidelberg (2008), 10.1007/978-3-540-85893-5_20
7. Ekberg, J.-E., Kylanpaa, M.: Mobile trusted module. Technical Report NRC-TR-2007-015, Nokia Research Center (November 2007), http://research.nokia.com/files/NRCTR2007015.pdf
8. EMV. Integrated Circuit Card Specifications for Payment System. Version 4.2, EMVCo (2008)
9. EMV. Contactless Specifications for Payment System. Version 2.1, EMVCo (2011)
10. NFC Forum. Logical Link Control Protocol. NFCForum-TS-LLCP_1.0, Technical Specification (2009)
11. Ghiron, S.L., Sposato, S., Medaglia, C.M., Moroni, A.: Nfc ticketing: A prototype and usability test of an nfc-based virtual ticketing application. In: First International Workshop on Near Field Communication, NFC 2009, pp. 45–50 (February 2009)
12. ISO/IEC 14443. Identification cards – Contactless integrated circuit cards – Proximity cards. ISO, Geneva, Switzerland (2008)
13. ISO/IEC 18092:2004. Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1), 1st edn., ISO, Geneva, Switzerland (2004)
14. ISO/IEC 21481:2005. Information technology – Telecommunications and information exchange between systems – Near Field Communication Interface and Protocol -2 (NFCIP-2), 1st edn., Geneva (2005)
15. ISO/IEC 7812-1:2006. Identification Cards - Idnetification of issuers - Part 1: Numbering system, 3rd edn., ISO, Geneva (2006)
16. ISO/IEC 7816-4:2005. Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange, 2nd edn., ISO, Geneva, Switzerland (2005)
17. KooMan, F.: Using mobile phones for public transport payment. Master's thesis, Radboud University Nijmegen (2009)
18. Kostiainen, K., Ekberg, J.-E., Asokan, N., Rantala, A.: On-board credentials with open provisioning. In: ASIACCS 2009: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, pp. 104–115. ACM, New York (2009)

19. Lau, P.S.C.: Developing a contactless bankcard fare engine for transport for london. Master's thesis, Massachusetts Institute of Technology (2009), http://hdl.handle.net/1721.1/55337
20. Luptak, P.: Public transport sms ticket hacking. Presented in Hacking at Random (2009), https://har2009.org/program/events/89.en.html
21. Mayes, K.E., Markantonakis, K., Hancke, G.: Transport ticketing security and fraud controls. Information Security Technical Report 14(2), 87–95 (2009); Smart Card Applications and Security
22. Mehta, S.: Analysis of future ticketing scenarios for transport for london. Master's thesis, Massachusetts Institute of Technology (June 2006), http://hdl.handle.net/1721.1/34592
23. Parno, P., Lorch, J., Douceur, J., Mickens, J., McCune, J.: Memoir: Practical state continuity for protected modules. In: IEEE Symposium on Research in Security and Privacy (2011)
24. Global platform. Globalplatform card specification v2.2.1 (2011), http://www.globalplatform.org/specificationscard.asp
25. Srage, J., Azema, J.: M-Shield mobile security technology. TI White paper (2005), http://focus.ti.com/pdfs/wtbu/ti_mshield_whitepaper.pdf
26. Wilcox, H.: Mobile ticketing: Transport, sport, entertainment event 2008-2013. Technical report, Juniper Research (October 2008), http://www.juniperresearch.com/reports.php?id=155 (accessed: July 2011)

Appendix A: Ticket Inspection w.o. Back-End Confirmation

If the *phase 3* of the ticketing protocol has not yet been conducted by device X, ticket inspection can be done based on the complete tap transaction evidence and an additional device identification signature as shown in Figure 7. For non-gated transactions this amounts to 600-700 bytes, and will take around 1s to transfer over NFC. The validation device will not be able to reliably determine the tap time, so in this sense the protocol is weaker than the default validation. However, if the user consistently uses old tap evidence, the inspection transaction is enough to catch him. Even as the validation device may not have an exact time for the tap, it can have a list of terminal counters from e.g. a day back, since the back-end will get this information and can periodically distribute it to validation devices. Since the terminal R counter values will be visible in this validation protocol option, also the counters of legitimately tapping users traveling on a given route (e.g. in the same bus) will provide current counter ranges for the bus stop cards relevant for this vehicle. Furthermore, the ticket inspection devices can also be assumed to eventually report their validation evidence back to the back-end, which has more accurate terminal R counter vs time information. In combination, all of these mechanisms makes it very hard for an active fraudster to not be caught by ticket inspection, espcially as his TEE identity will be unconditionally mapped to the inspection.

A privacy drawback of the ticket inspection without back-end confirmation is that it exposes the exact counter values of device X to an eavesdropper. Those values may be enough to track a single user.

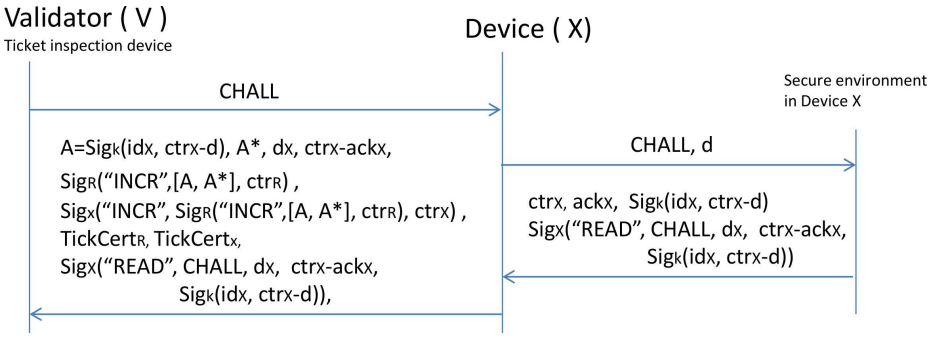

Fig. 7. Ticket inspection before phase 3

Appendix B: Back-End Data Auditing

In our ticketing system, the taps that are received either for gated or non-gated travel can be collected and indexed for a given user PAN. The taps form a series of waypoints in place and time for the fare calculation system to determine the applicable user charges. The system allowes charging based on fixed monthly, regionalized fees, hourly fees or any other schemes in use today. The main advantage of the approach is however that it does open up the possibility for much more flexible charging options where the user is charged with much finer granularity than today, e.g. based on travelled distance, time-of-day, volume discounts etc.

It is also relevant to perform some data mining on the received data, to identify fraud attempts and anomalies of various kinds. The input data is structurally simple - terminals R and gates represent locations, for gates the transaction time for each entry or exit of devices X is trivially logged, in non-gated operation, the monotonic counter value of the terminals R is a representation of time. The back-end receives time estimates for those values from devices X, and a hard bound for the time, based on the time of reporting. Thus, for each counter increment in R a statistically accurate mapping can be made for most terminals R, even in the absence of some reported evidence. Combined with the identification provided by the commitments returned in the $A*$ data elements, the back-end server should be able to patch together a fairly accurate picture of who tapped what and when, and evidence for all of these events should eventually flow back to the back-end. Based on the data, information about fraud attempts as well as malfunctioning or non-reporting devices can be mined from the data. With this information the appropriate user can be notified, and if necessary, extra invoicing, device black-listing or even legal penalties can be applied.

We believe that both the fare calculation and security auditing processes are good research problems in their own right, and constitute excellent further work when the system is deployed in a live test or field trial.