Tapping and Tripping with NFC

Sandeep Tamrakar[1] and Jan-Erik Ekberg[2]

[1] Aalto University School of Science, Finland
sandeep.tamrakar@aalto.fi
[2] Nokia Research Center, Radio Systems Laboratory, Finland
jan-erik.ekberg@nokia.com

Abstract. In public transport ticketing, the tap-in / tap-out user experience is an established metaphor since contactless NFC cards were introduced as travel cards some ten years ago. In our solution fixed smart cards at train station are tapped by NFC-enabled mobile phones of users. By leveraging the phones' communication capabilities, a possible embedded trusted execution environment (TEE) and the user interface, we have constructed a secure solution for so-called non-gated ticketing, where end user devices produce and report ticketing evidence under the threat of inspection. This is technically quite different from the traditional model where a certified, secure reader is tapped by a passive card. Learnings from a public ticketing trial conducted in the Port Washington branch of the LIRR train network in New York is presented along with an overview of the NFC protocols used in that trial. We also discuss extensions to the protocol with the goal to enable ticketing also for NFC phones without TEE support.

1 Introduction

All over the world, transport ticketing has for years been implemented using proximity technologies, increasingly using the ISO / IEC 14443 [12] contactless card standard. Protocols like Mifare[1] developed by NXP semiconductors allow contactless memory cards to be used as secure tokens or travel cards. A travel card used as a ticket can convey the identity of a user, the validity period of the ticket, the balance available in the ticket and other ticketing attributes such as valid region, discount group etc.

Mifare-powered ticketing systems have proven to be very usable in practice, despite some vulnerabilities as mentioned in [9,8,4]. However, Mifare-based schemes do require that the ticket readers which communicate with the travel cards have access to the secret keys of the card issuer, making the readers a security-critical part of the ticketing system setup. Thus, these systems do not easily scale to implementations where the cards are not communicating with the trusted, certified readers.

A more open approach that is applicable to ticketing is defined in the Open Payment initiative from the Smart Card Alliance [15]. In their architecture, each

[1] http://mifare.net

M. Huth et al. (Eds.): TRUST 2013, LNCS 7904, pp. 115–132, 2013.

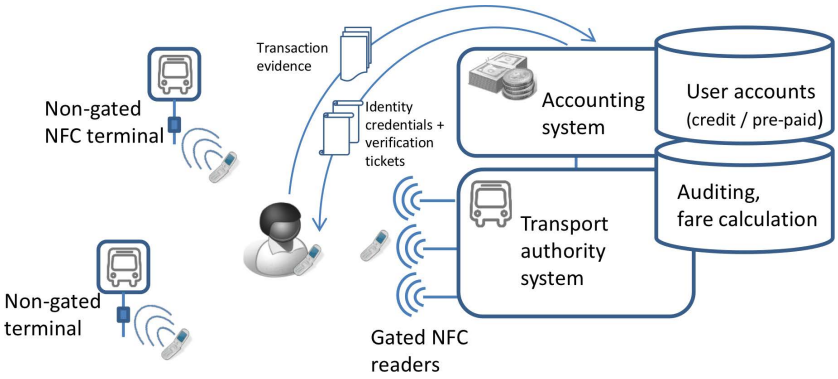

Fig. 1. System architecture

traveler is represented by a travel account in a server cloud. While traveling, only the identity of a user is verified using credentials stored e.g. in a travel card. In this approach, the ticketing and fare calculation operations can be totally separated from the evidence collected at the traveling endpoints. A very tempting variation of this system is to use a contactless credit card for identity verification, possibly allowing the user to travel with his card in many independent transport systems around the world.

An account-based ticketing system is technically easy to arrange in *gated transport*, where the transport end points are controlled via physical gates, e.g. as in the London Underground. We assume that these gates are equipped with a contactless reader that can verify the identity of a user and potentially consult a back-end cloud in real-time to perform all the necessary validation before allowing the user to travel. In our architecture, depicted in Figure 1, we extend this *identity-based ticketing* model[2] to *non-gated transport* such as S-Bahn in Berlin, where the travel tickets are not verified at the station gates. In non-gated transport, travelers are requested to perform certain transport-ticketing related functions on their own accord under the threat of sporadically occurring ticket inspection and system penalties imposed on dishonest travelers. We leverage the mobile phone as an NFC reader as well as its security and communication capabilities towards the back-end cloud in a model where the contactless smart cards are no longer used as end-user credentials, but as proof of location — i.e. identifying where (and partially when) the transport customer enters and exits the transport system.

This paper extends our previous work on non-gated transport [7] which was later publicly trialed with more than 100 participants in New York in 2012. Based on the ticketing data collected in the trial we can now confirm some of

[2] The public transport community uses the term *identity-based* for a ticketing system, where a travel account is assigned to each user and the identity of the user is verified at a transport station gate before allowing the user to travel. This does not necessarily imply the use of *identity-based encryption*.

our assumptions regarding system properties, transport user behavior as well as report on some of the data collected in participant interviews. This is the first contribution of this paper.

Additionally, our work on the protocols for the ticketing system has continued with a re-design for NFC-enabled phones where an embedded, programmable trusted execution environment (TEE) is unavailable or practically unaccessible. The design upgrades accommodate such *open devices* in the ticketing system without significantly changing the risk model of fare collection and auditing or inducing unnecessary liability for the travelers. This is a second contribution of this paper.

In Section 2, we explore related work in the general domain of identity verification with NFC-enabled phones. Section 3 outlines the security protocol used in the trial, which is a part of previously published work. Section 4 presents the trial results. The protocol extensions for open devices are presented in Section 5, along with a brief security analysis. Section 6 provides implementation details and measurements for the extended protocol on an Android phone. Section 7 gives the acknowledgments and finally conclusions in Section 8 end the paper.

2 Related Work

A prototype application developed by RFID lab of the University of Rome [10] implements a virtual transport ticket applet stored in a secure environment of an NFC-enabled mobile phone. A ticket can be purchased over SMS using a Java MIDlet application that interacts with the ticket applet running in the secure environment. Before traveling, the ticket information is transmitted over NFC to a ticket reader, which validates the ticket. A similar user-centric ticketing approach is proposed by Chaumette et. al. [3]. They present an architecture for event ticketing using NFC-enabled mobile phones that use SIMs as secure elements. Both of these systems work in online as well as partially offline modes, but require dedicated reading terminals.

A work by Derler et. al. [5] focuses on the anonymity of NFC ticketing in order to protect the privacy of a ticket holder. In their model, a ticket can be verified without divulging the identity of a user. This reduces the risk of an attacker profiling user's travel based on eavesdropped tap events. The Smart Card Alliance promotes an open payment system using an account-based architecture [15] for public transport systems, where the account information is verified by a ticket reader and forwarded to a back-end server. The back-end accumulates these records and later charges the account holder for transport system use.

The abovementioned systems use a secure element on a mobile phone to store the ticketing credentials. Dmitrienko et. al. [6] implement a software-based access token on mobile phones, where the software domains are isolated from each other using TrustDroid [2]. The above contributions neither discuss a complete ticketing system nor consider non-gated ticketing.

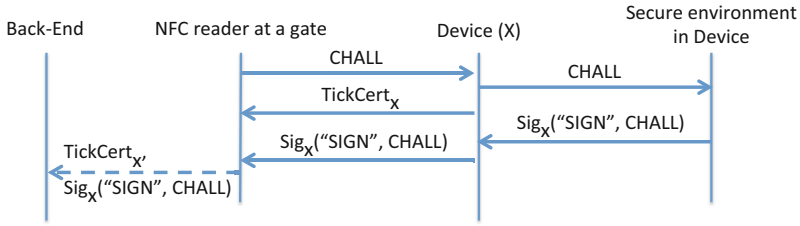

Fig. 2. An overview of gated ticketing protocol

3 Our First Ticketing Protocol

Our public-transport ticketing protocols that form the starting point for this work have been reported in [7]. In short, we built both gated and non-gated transport ticketing for mobile phones with a built-in TEE and NFC communication primitives. The protocol for gated transport, depicted in Figure 2 is a straight-forward challenge-response design. A user touches his device to the reader attached to a transport-station gate. The reader then initiates a session by sending a challenge to the user device. The challenge contains a nonce and the identity of the reader or the station. Upon receiving the challenge, the user device immediately sends back its ticketing certificate issued by the back-end server (or CA) and subsequently signs the challenge with its ticketing keys inside its TEE. The signature over the challenge is also returned to the reader. The reader is equipped with the public key of the back-end server (or CA), and it validates the certificate. From it the reader also extracts the public key information of the user device in order to validate the signature on the challenge. If all the verifications succeed, the station gate is opened.

For non-gated transport, our system is built to satisfy the following goals and user interaction patterns:

R1. The location, time, identity of a traveler and needed cryptographic evidence shall form a tuple that defines the trip end-points and the traveler in a reliable and non-repudiable manner,
R2. Trip end-points, e.g. touch points at bus stops, can be equipped with contactless smart cards, but not with gates or contactless devices that require continuous power supply or back-end connectivity.
R3. The mobile phone cannot be assumed to be connected to a back-end cloud infrastructure in real-time, i.e. the system must be designed to operate in a *partially offline* manner.
R4. The activity of a traveler with a touch point shall be modeled as 'tap', i.e. a traveler taps his phone to a touch point at a bus stop before begining a trip, and taps another touch point when he ends his trip at another bus stop.
R5. Travelers might be subjected to random ticket inspection, i.e. protocols must be designed to support this property.

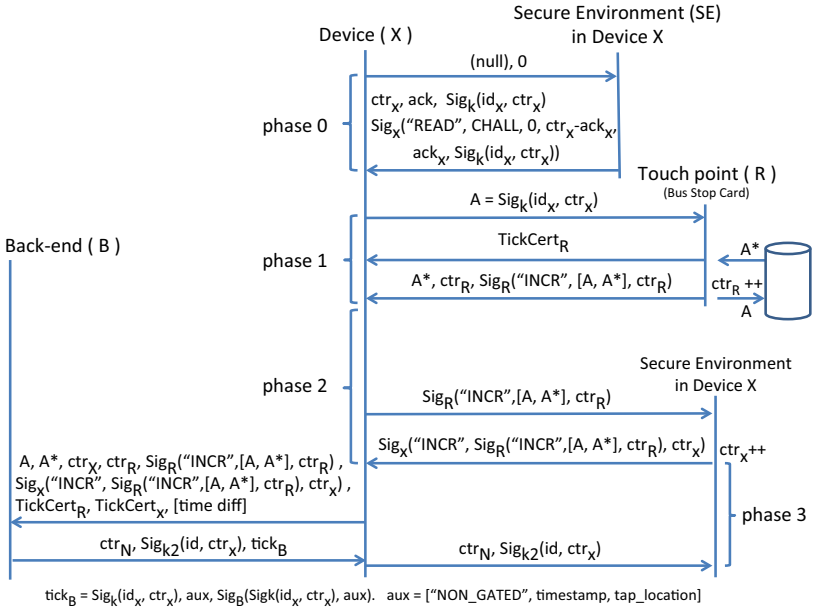

Fig. 3. An overview of non-gated ticketing protocol

Our solution is based on signed challenges produced by the TEEs, where each signature also includes a TEE-specific counter. A *sign and increment* command monotonically increases the counter. Future TEE counter update and signature operations are limited by an authenticated release of the counter window signed by the back-end cloud. This limits the amount of taps that a single user device can perform before being forced to report evidence to the back-end cloud in order to continue tapping. The same TEE logic is used both in the TEE of user devices and in the contactless smart card at touch points. However, the counter window in the contactless cards is not used — touch-point counter values are used to determine the order of tapping that the touch point is involved in, not to enforce reporting.

A complete non-gated ticketing protocol executes in three phases as depicted in the Figure 3. A user device X may enter phase 0 any time after the completion of the last ticketing event. This phase prepares the user device for the next ticketing event. In this phase, the user device X reads the TEE counter state and retrieves a counter commitment $Sig_k(id_X, ctr_X)$, for the latest counter value. The TEE also returns other data which is used as part of ticket inspection.

When the user taps his device X to a touch point R as shown in the Figure 8, the user device enters phase 1. In this phase, the device sends its latest counter commitment $Sig_k(id_x, ctr_x)$ obtained from phase 0 as a challenge to the touch point. In return, the touch point sends back its device certificate $TickCert_R$

as well as a signature Sig_R that binds its own counter ctr_R to the challenge. Additionally, the touch point R also returns some auxiliary data $A*$ that needs to be reported to the back-end server along with the terminal signature for verification. The challenges sent to the touch points are the commitments of the user device. These commitments can be later used by the back-end cloud to statistically infer the device that sent the challenge.

The signature $Sig_k(id_x, ctr_x)$ cannot be resolved without the knowledge of the key k and the value of ctr_x remains fresh each time it is retrieved. Therefore, we can assume that the identity of device X remains unknown to the entities other than the protocol participants. The user device X can determine the identity of a touch point from its certificate and validate the signature in order to protect against the interaction with rogue touch points.

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

The user device enters phase 3 immediately after completing phase 2. In phase 3, the device collects all the data generated or gathered from phases 0 - 2 and sends them to the back-end using a server-authenticated TLS channel. Additionally, the device also sends its estimation of the time that has passed between phase 1 and the first message of phase 3. In order to validate a transaction, the back-end identifies both the device X and the touch point R involved in the transaction. After validation, the back-end returns a release commitment for the counter of the device X. Without the release commitment from the back-end, the TEE of the device will eventually exhaust its counter window and refuse to sign any further taps. This mechanism forces the user to report tap evidence in phase 3. Additionally, the back-end will also return to the user device necessary information for ticket inspection.

Phase 3 requires network connectivity to the back-end cloud. In some cases this phase can be significantly delayed e.g. due to poor network coverage. To improve on this delay problem, we also store challenges (i.e. commitments) received from the user devices in the touch-point cards. Each stored challenge is probabilistically selected and cryptographically bound to at least two later invocations of phase 1 interaction (with some other user device). Similarly, every user device that taps a touch point is forced to relay two stored challenges from previous taps back to the back-end cloud in addition to the response for its own challenge. This provides a back channel of tap records, which can be used for security auditing and even fare calculation while waiting for the device that originated the tap to report its evidence.

A discussion on other system features, like enrolment, auditing, ticket verification as well as a protocol security analysis was presented in our previous work [7].

4 MTA/LIRR Mobile-Ticketing Trial

The mobile-ticketing trial was carried out in four phases from December 2011 to June 2012. The duration of the first three trials was about a week, where 20 employees from the Metropolitan Transport Authority (MTA) New York participated. The primary objective of these three trials was to test the system, improve it and add new features based on the feedback received from the participants. The final trial was carried out over a period of four weeks along the Port Washington Branch of Long Island Railway Road (LIRR). A total of 110 registered customers of MTA / LIRR with an annual subscription completed the trial.

The primary objective of the trial was to understand the user acceptance of mobile ticketing in non-gated transport and to learn the pattern of their ticket use. Particularly, we were interested to know the tapping behavior of participants; tap-in begins a trip and tap-out completes the trip. These tap events were reported to our back-end server. We wanted to know exactly how many participants complete their trips by tapping out. If the participants forgot to tap out within 3 hours, from the start of their trip, the application was designed to send a trip expiry message to the back-end server and notify the participants accordingly. We also wanted to learn about the connectivity to the back-end server from the mobile phones used in the trial, e.g. how soon a mobile phone reported a tap.

4.1 Tags in Trial

The final trial was carried out at 14 stations along the Port Washington Branch of LIRR. During initial testing, we found that the NFC operating distance for SmartMX tags with our protocol, using the chosen trial phones, was less than 2 cm whereas the corresponding distance for Mifare Classic tags was 4 cm. The more exact alignment of the antennas needed with the SmartMX tags compared to the Mifare tags was perceived to reduce the user experience. Therefore, Mifare tags were chosen as the primary tags during the trial and the SmartMX tags were left in only for protocol testing and reference.

A total of 105 Mifare classic tags were placed at the stations along with 10 SmartMX tags which were placed at only 5 stations. Each Mifare Classic tag stored a Station ID, corresponding to the station at which the tag was placed, as an NDEF message. During tapping the Station ID along with the Unique Identity (UID) [3] of the tag was collected as a tap record. In case of SmartMX tags, the challenge sent by the phone, the response returned by the tag and other context related information were collected as the tap record. The tap record was then cryptographically signed inside the TEE of the phone before being sent it to the back-end server.

[3] UID: a fixed 4 or 7 bytes identity that is assigned to each NFC tag at the time of manufacture.

4.2 Ticketing Application

A ticketing application designed for the trial was integrated as an extension to the Nokia Public transport application [4] and was installed in NFC-enabled Nokia 603 phones used in the trial. The phones were then provided to the participants. An interface to register participants to the back-end server was also included in the application. During registration, necessary ticketing credentials were enrolled into the TEE of the mobile phone. In Nokia 603 phones, we used the On-board Credential (ObC) [13] system to execute the credential algorithms in isolation from the operating system. We enrolled the TEE algorithms and secrets to ObC from the ticketing application using ObC APIs. After successful enrolment, the participants were ready to start their trips by tapping tags installed at the stations. Additionally, a ticket inspection application was provided to the MTA ticket inspectors. The inspection application was capable of interacting with the participants' ticketing applications over peer-to-peer mode of NFC in order to validate the tap-in event of a trip. The protocol used to validate tickets during ticket inspection has been reported in our previous work [7]. The participants were randomly inspected during the trial.

At the request of the MTA, an additional feature termed as *Checkout* was added, which allowed the participants to select any station used in the trial as a manual checkout station. In the ticketing application, a button that allowed manual checkout appeared one hour after the beginning of a trip and the feature was disabled along with the trip-expired event or if the participant selected a checkout station.

Each tap was immediately reported to the back-end server. However, if the network was unavailable at the time of the tap then the ticketing application periodically checked the network status and reported the taps to the server as soon as possible. Using the tap records, we were able to determine the start and the end points of a trip and associate appropriate ticket fare to the trip. However, no actual cost was associated to any trips travelled by trial participants.

4.3 Trial Results

Over the period of four weeks, 3166 complete trips were recorded with an average of around 29 trips per participant. Based on the number of trips made by the participants, we found that 80% of the participants used their ticketing application to make a single trip on a daily basis. 16% of the participants were using the ticketing application actively, i.e. for at least 20 trips. The remaining 4% of the participants were using the ticketing application less frequently, i.e. less than 10 complete trips, during the trial period. We also found that almost 90% of the trips were completed by a proper tapping out at a station. Around 2.4% of the trips were completed using the manual checkout feature and the remaining 7.4% of the trips were automatically expired.

Figure 4, shows that around 67% of the taps were reported immediately and more than 80% of the tap records were reported on the same day. The tap

[4] `http://store.ovi.com/content/237984`

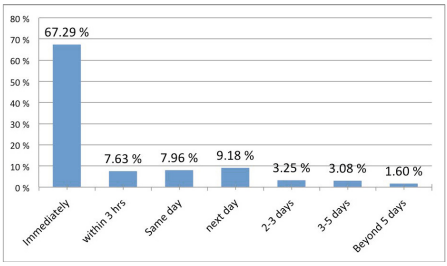

Fig. 4. Tap reporting time

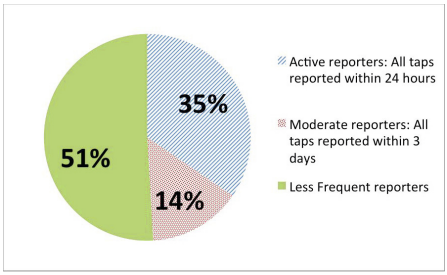

Fig. 5. Categorization of the participants based on their tap reporting

reporting allowed the back-end cloud to monitor the transport system with high accuracy. For example, a near real-time system usage statistic as depicted in Figure 6 could be constructed for the non-gated transport.

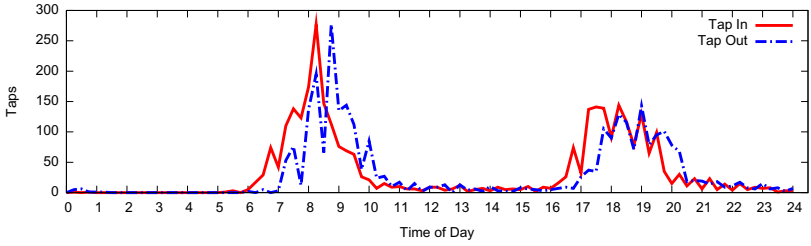

Fig. 6. 24h traveling statistic

We categorized the participants according to their tap reporting behavior as depicted in Figure 5. The median tap reporting time among the travelers in the active category was 109 seconds. As explained in Section 3, we designed the touch-point smart cards / SmartMX tags in such a way that they add old challenges from previous travelers to the currently tapping phone to be returned to the back-end cloud together with the current tap evidence. The impact of this system can be significantly increased if we add the tap reporting activity of travelers to the transport certificate and use this information within a SmartMX tag to assign tap evidences related to the least active travelers to be piggybacked by the most active travelers tapping the tag. Unfortunately, we also learned that trial participants on average had to tap SmartMX tags at least twice to achieve the successful tap event. This may be attributed to the mobile phone antenna not being ideally suited to power up the SmartMX cryptographic operations. In real deployments a 50% failure rate is of course not acceptable, and either the power transfer from phone needs to be improved or the energy consumption of SmartMX cards must be optimized.

5 Ticketing System Upgrade

The results in Section 4 show that the assumption of not having continuous back-end connectivity is reasonable. Even in the LIRR train system that operates completely over ground, only 35% of the devices were well-connected to the Internet while traveling. Around 15% of the evidence was reported by the travelers more than 24 hours after the travel occurred. Another insight is that the travelers seem to accept and remember to tap out after traveling — in the non-gated trial 90% of the trips were properly tapped out even though there was no stated penalty imposed for not tapping out. For non-roaming customers the back-end connectivity cost is likely not an issue. Already in 2010, a published report from mobiThinking [14] indicates that in the U.S. the penetration of flat-rate data plans (29%) was higher than the smartphone penetration (27%), so it is safe to assume that virtually all NFC smartphone are on fixed data plans and reporting the tap evidence back to the server has no marginal monetary cost.

On the other hand, very few NFC-enabled phones today include a programmable secure or trusted environment. Since we must assume that the users cannot be mandated to upgrade their existing NFC-enabled phones in real ticketing deployment, a protocol variant that decreases the dependency on user device security is needed.

The learning from the trial forms the basis of a ticketing system upgrade that enables the use of NFC-enabled mobile phones that do not have a programmable TEE. For this re-design, we revisit the system assumptions of the original non-gated system in the following manner:

1. The user device / mobile phone is not trustworthy. A virus or the traveler himself potentially has access to all the code and secrets in the phone, and may report on these secrets over the Internet.
2. We increase the expectation for the capability of the phone to connect to a back-end cloud. We will design the revised protocol around a time period of t minutes. A traveler must connect his device to the back-end cloud and receive "real-time" tokens at most t minutes before traveling.

Our main incentive for upgrading the ticketing system for open devices is to alleviate the risk of attacks potentially directed against the travelers with open devices. Since our system is Id-based, the main threat is the misuse of identities, i.e. a liability concern for the travelers.

We assume that the main protocols and functions presented in Section 3 still apply to open devices. These devices will still perform the same steps of enrolment, certificate renewal, signing touch-point smart-card responses and receiving authenticated release commitments for the device-specific counter. Compared to a device with a TEE, the trustworthiness of the open device is assumed to be weaker. The only operation that is partially directed against the traveler not reporting taps to the back-end is the requirement for counter release commitments. For open devices, we augment this functionality with a requirement to fetch the challenge for the touch-point smart cards in near-real time from the

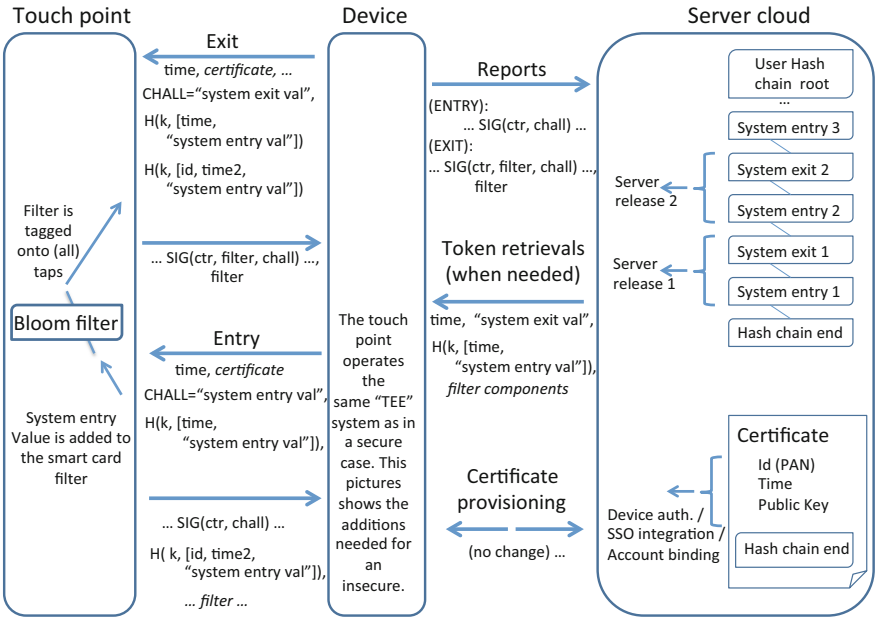

Fig. 7. Ticketing - insecure terminals

back-end cloud. In this manner, we still force the traveler's phone to periodically interact with the back-end cloud in the non-gated transport. This new interaction can also be protected by validating user credentials, e.g. a PIN, to further complicate the system infiltration required to mount any successful attack.

Furthermore, we add some new attributes to the transport certificates issued by the back-end infrastructure to open devices. We augment the touch-point smart card logic with new auditing features that increase the probability of catching identity theft in non-gated transport and we also add a feature to make tail-gating attacks[5] more difficult.

Figure 7 shows the overall additions done to the system. The new data structures are as follows:

1. A reverse hash-chain attribute is added to the transport certificate, signed by the server trust root and bound to an account of a traveler. The reverse hash-chain is split into run lengths of m elements ($m = 2$ in Figure 7). The actual elements (tokens) of the hash chain are retrieved m at a time by the mobile phone of the traveler before traveling. The token retrieval is possibly subject to user authorization for improved end-user protection.
2. A monotonically increasing time value is added to the system, and maintained by the back-end cloud. The time value is updated e.g. once a second,

[5] A tailgating attack is where a customer intentionally throws a valid ticket back over the gate to let a friend defeat the physical access control of the gate.

and is consistent across a single transport system. This time value will be signed by the server distributing the hash-chain elements and be cryptographically bound to the last token from the set of m tokens in the hash chain, i.e. the one that is to be spent first, on system entry.

3. All touch-point smart cards are augmented with a time-dependent Bloom filter[1] which is maintained individually by every single smart card. This is in addition to the statistically returned challenges of earlier travelers. Like the statistically selected earlier challenges, the Bloom filter is also returned to the back-end cloud through the tapping client. Cryptographic binding of the filter to the response forces the end-user device that taps the touch-point card to return the filter along with the challenge-response to complete a valid transaction report. A time awareness within each card is built based on the entry tap-time commitments by the back-end, i.e. the reference time may be lagging for touch-point cards that are rarely used.

The extensions for the ticketing system operate according to the message flows outlined in Figure 7. The touch-point smart cards now include distinct operations for entry vs. exit — intermediate taps, if supported, can be modeled according to the exit template.

The entry operation with the touch-point smart card includes the validation of the transport certificate, and that the entry token maps to the hash-chain root. The entry operation will also validate that the time bound to the entry token is e.g. at most $t = 900$ (15 minutes) earlier than the last time seen by the smart card. If all validations succeed, the smart card will return a response to the end-user device that includes not only the signed challenges but also a verification ticket bound to the entry token value. This ticket can later be validated by all other smart cards in the system. Furthermore, the entry token value will be added to a Bloom filter in the card that is periodically emptied, i.e. it contains only entry taps accumulated during a t-minute period. The Bloom filter is a very efficient data structure for this kind of aggregation, since filters for many smart cards can be trivially combined in the server, and a search for possible double-spending of entry taps among all smart cards can be performed efficiently.

During exit, a touch-point smart card does not accept a tap operation without a matching system entry commitment returned by some other smart card in the transport system. An exit token must also be in the same hash chain as the entry token. These mechanisms alleviate identity theft, since an NFC eavesdropper may get the entry tap and the smart card signature, but not the exit token. Whenever tokens are retrieved with NFC eavesdropping or network-based attacks, the extra use of the token will trigger double-spending auditing mechanisms.

The traveler's incentive for reporting back evidence in the revised system is different from the protocols that use TEEs. In the latter case, the phone will force the traveler to report back on the threat of becoming dysfunctional, and all signatures are signed with keys that reside in the TEE. In the former case,

the blocking mechanism relies on the conditional reception of the tokens from the server and the assumption that reporting of travel conducted based on those tokens must be performed before the retrieval of the next set of tokens. Token retrieval with missing submitted evidence should by default be considered to represent the maximum fare of any trip that can be made on the system. In this way, the user is always incentivized to report the evidence correctly and promptly. Timely evidence feedback also benefits the user by improving the auditing mechanisms for catching double-spending.

Based on the Bloom-filter contents, and the knowledge of tokens active at a given time (the only condition by which they are accepted at touch-point smart cards), every card returns, on every tap, a statistical representation of the recent entry taps at the touch-point card that is being tapped. This information is channeled by the mobile phones to the back-end cloud. With the assumption that at least 50-70% of phones report back (their own taps) almost immediately, it is easy enough in the back-end cloud to aggregate the Bloom filters and pinpoint double-spending occurring in the transport system - since all tokens are generated in the back-end cloud, full information of their contents and validity (in terms of t) is known to the back-end.

5.1 Brief Security Analysis of the Added Features

A variant of our baseline non-gated ticketing protocol has been formally analyzed in the Ph.D dissertation by Enyang Huang [11]. For this work, we assume that the phones, in addition to the augmentation, operate the default signing scheme already deployed. Thus a replay attack entails both stealing the longer term signature key from a phone, capturing the token over the air (and replaying it) or alternatively mounting a harvesting attack using a real-time virus in the attacked phone. We can identify the following threat categories and corresponding ways the described solution mitigates these issues:

1. An attacker has learned the long-term secrets of a victim. If the attacker copies the entry code off the air, he can likely in a non-gated environment produce a tap and a smart card response that will withstand at least cursory ticket inspection. However, the system will catch double-spending by aggregation and inspection of the touch-point card Bloom filters. In a gated system, entry duplicates are likely caught immediately and even access may be denied for either the attacker or the correct traveler. With token copies retrieved by eavesdropping the NFC interface, the attacker cannot exit a gated system if he does not follow the victim like a shadow.
2. Any copying of the short-lived tokens is valid only for entry during the stated system allowance period t. In a gated transport this is an absolute measure, but old copies will also be caught at ticket verification in a non-gated system and by touch-point cards in case the use of the cards has advanced its notion of time past the time constraint of the token copy.

3. The attacker travels using a complete copy of all ticketing data in the original traveler's phone [6]. This means that the attacker will report all travels to the back-end just like the original traveler would do. Based on the protocol and its secrets, there is no way of differentiating the attacker from the original traveler since we assume that the attacker has full access to the mobile phone of the original traveler. However, double-spending mechanisms will notice parallel usage quickly, and in gated transport one of the two phones may even be denied system access or exit. In any case, the fraud is quickly unearthed, and appropriate measures can be taken. For example, while fetching a fresh token in non-gated transport, the back-end cloud may require additional out-of-band user authentication or verification of one-time-PIN sent to the traveler via a separate channel such as SMS.
4. The attacker travels using the identity of a traveler, but does not report anything to the back-end if ticket verification is not encountered. In this case, the smart card filters will provide information to the auditing server about non-reported taps. Further, tap information becomes available as part of the back-channel from smart cards to the server through other tapping travelers. Using this mechanism, or by the attacker encountering ticket verification, the system will get information of an attacked identity.
5. A traveler may collaborate with an associate and make a copy of his entry tap to his associate. The associate then taps out at a nearby station while the traveler continues his trip. Later, the associate sends the exit-tap information back to the traveler, which the traveler submits to the back-end cloud after he completes his trip. In this way, a traveler pays for only a short distance trip fare while actually traveling a longer distance.
   The probability of spotting such an attack relies on the number of honest travelers who report their tap-evidence immediately. As each reported tap carries evidence of the two earlier taps made at the same touch point, the back-end cloud may receive the information about the dishonest traveler's exit tap made at short distance station before he actually completes his trip. During ticket inspection, such fraud can be identified provided that the inspection device consults the back-end during ticket verification. In case of a TEE implementation the current state of the device's counter commitment will always reveal the early tap out to an inspector.
6. A widespread software attack, where a vast number of phones are infected as a botnet and, for example, one trip from each victim is used by the attacker, will be impossible to protect against with the above assumptions. To alleviate this kind of attack some other reactive security mechanism, for example a virus checker, needs to be deployed.

The protocol additions for open devices put in place several separate mechanisms to protect both the traveler and the system against undue fraud and misplaced liability. Nevertheless, when deployed in a mass-market scenario, there is a clear

[6] All needed information is only available for copying at most t seconds before traveling because of the requirement to fetch fresh tokens before traveling, all needed information is only available after the tokens have been fetched.

threat that widespread attacks can cause significant disturbances in the perception of the ticketing system by travelers, since an attacker can easily cause denial-of-service and cases where many kinds of plausible undeniability may surface. Clearly, a system where the traveler's phone is equipped with a TEE is the more user-friendly choice.

6 Implementation and Measurements

We have used Google Nexus S phone running Android 4.1.1, Jelly Bean, to implement our non-gated protocol on the phones without TEE support. Although Google Nexus S has an embedded secure element (SE), an API to access the SE is not included in the publicly available SDK [6]. We have implemented the mobile-ticketing application on the phone using Android SDK with level 14 APIs. The touch-point smart cards implemented using SmartMX tags were upgraded with the protocol addition described in Section 5. Figure 8 shows an example of the touch-point with a SmartMX tag used in the trial. The application on the phone uses NFC reader / writer mode to communicate with the SmartMX tags using Android NFC IsoDep class APIs.

The application reported on in this work retrieves two elements of a hash chain from the server at a time. Therefore the phone can be used twice to interact with the touch-point tags before a new token is required. In other words, a phone can be used for a single journey beginning with a tap-in event and terminating with a tap-out event. Additionally, the token has been generated in such a way that it is valid for the tap-in event only during the first 15 minutes after the token was received. Figure 9 shows the mobile-ticketing application used in the trial running on Nokia 603 phone and the application with the revised protocol for open devices running on Nexus S phone.

Table 1 presents the measurements of average execution times for different SmartMX operations measured from the Nokia 603 with a TEE (as used in the trial) and the Nexus S phone without TEE support. The Nokia 603 and

Fig. 8. touch-point used in the trial

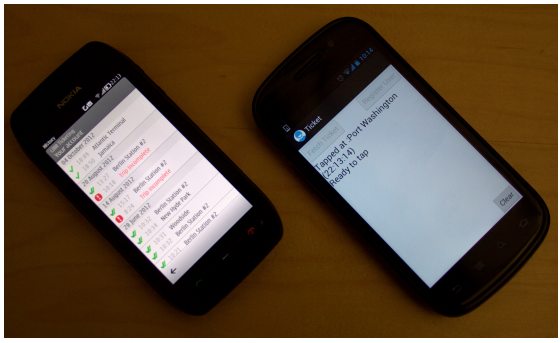

Fig. 9. Mobile-ticketing application running on Nokia 603 and Nexus S

Table 1. Time measurements of SmartMX operations in non-gated protocol

Platform	Initialize and read certificate	Challenge Response	Total
Nokia 603	45 ms	212 ms	305 ms
Nexus S	55 ms	718 ms	820 ms

the SmartMX tag run the baseline protocol intended for devices with a TEE. The Nexus S interacts with the SmartMX tag using the revised protocol for open devices. The measurements do not include the wake-up time taken by the operation systems to indicate the NFC touch event to the application. The new protocol additions increase the amount of SmartMX tag operations which is visible as the 500 milliseconds (ms) increase in the challenge response time. The table also shows that the NFC data exchange speed in Nokia 603 Symbian phone is slightly faster than Android Nexus S phone.

7 Conclusions

The research question how to achieve a workable ticketing solution for NFC phones can be seen as the task to minimize fraud opportunity balancing between networked-based auditing and security mechanisms which can be leveraged in contemporary end-user devices. Requirements stemming from the target transport system and end-user usability must be considered non-negotiable.

In this paper, we studied approaches for using NFC phones with limited platform security in a ticketing system. Likewise, we presented mechanisms by which the fare collection properties of the system can be upheld with insecure devices. We noticed the importance of categorizing users based on their tap reporting behavior and using this information to accelerate the reporting time of tap events to the back-end cloud, thereby improving the auditing accuracy. Speedy identification of misbehavior, such as double-spending attempts, minimizes the liability consequences for the traveler in an identity-based ticketing system.

We also reported on a big-scale trial in a real transport system with more than 100 participants. A user-study conducted at the end of the four-week trial in New York in the summer of 2012 gave very promising feedback and a motivation to continue exploring this field: 42% of the participants felt that a smart phone was the preferred user credential for transport ticketing, compared to only 24% in favor of a smart card or payment card. The perceived comfort level of the participants of this system was also surprisingly high considering that this was a first for most participants — 64% of the travelers were pleased with using the phone as a "travel token" compared to only 11% that were not satisfied with the solution. We believe that this level of user acceptance indicates that ticketing with NFC phones may grow to become a "killer use case" for mobile phone NFC use.

Acknowledgements. The referenced field trial was conducted as a collaboration between MTA in New York and the Nokia Location and Commerce business unit. Without the hard work of Peter Preuss, Justus Brown, Jerome Beaurepaire, Andreas Graf from Nokia L&C, and the technical expertise and contacts of Jukka Virtanen and Jarkko Sevanto the trial would never have seen the light of day.

References

1. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Commun. ACM 13(7), 422–426 (1970)
2. Bugiel, S., Davi, L., Dmitrienko, A., Heuser, S., Sadeghi, A.-R., Shastry, B.: Practical and lightweight domain isolation on android. In: Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, SPSM 2011, pp. 51–62. ACM, New York (2011)
3. Chaumette, S., Dubernet, D., Ouoba, J., Siira, E., Tuikka, T.: Architecture and comparison of two different user-centric NFC-enabled event ticketing approaches. In: Balandin, S., Koucheryavy, Y., Hu, H. (eds.) NEW2AN 2011 and ruSMART 2011. LNCS, vol. 6869, pp. 165–177. Springer, Heidelberg (2011)
4. de Koning Gans, G., Hoepman, J.-H., Garcia, F.D.: 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)
5. Derler, D., Potzmader, K., Winter, J., Dietrich, K.: Anonymous ticketing for NFC-enabled mobile phones. In: Chen, L., Yung, M., Zhu, L. (eds.) INTRUST 2011. LNCS, vol. 7222, pp. 66–83. Springer, Heidelberg (2012)
6. Dmitrienko, A., Sadeghi, A.-R., Tamrakar, S., Wachsmann, C.: SmartTokens: Delegable access control with NFC-enabled smartphones. In: Katzenbeisser, S., Weippl, E., Camp, L.J., Volkamer, M., Reiter, M., Zhang, X. (eds.) Trust 2012. LNCS, vol. 7344, pp. 219–238. Springer, Heidelberg (2012)
7. Ekberg, J.-E., Tamrakar, S.: Mass transit ticketing with NFC mobile phones. In: Chen, L., Yung, M., Zhu, L. (eds.) INTRUST 2011. LNCS, vol. 7222, pp. 48–65. Springer, Heidelberg (2012)
8. Garcia, F.D., de Koning Gans, G., Muijrers, R., van Rossum, P., Verdult, R., Schreur, R.W., Jacobs, B.: Dismantling MIFARE classic. In: Jajodia, S., Lopez, J. (eds.) ESORICS 2008. LNCS, vol. 5283, pp. 97–114. Springer, Heidelberg (2008)
9. Garcia, F., van Rossum, P., Verdult, R., Schreur, R.W.: Wirelessly pickpocketing a mifare classic card. In: IEEE Symposium on Security and Privacy, pp. 3–15 (2009)
10. 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)
11. Huang, E.: Automated Security Analysis of Payment Protocols. Ph. D. Thesis, Massachusetts Institute of Technology, Dept. of Civil and Environmental Engineering (2012)
12. ISO/IEC 14443: Identification cards – Contactless integrated circuit cards – Proximity cards. ISO, Geneva, Switzerland (2008)
13. 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)

14. mobiThinking: Global mobile statistics 2012 part b: Mobile web; mobile broadband penetration; 3g/4g subscribers and networks, http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/b (accessed: February 2013)
15. Smart Card Alliance: Transit and contactless open payments: An emerging approach for fare collection. A Smart Card Alliance Transportation Council White Paper (November 2011), http://www.smartcardalliance.org/resources/pdf/Open_Payments_WP_110811.pdf (accessed: February 2013)