User Privacy in Transport Systems Based on RFID E-Tickets

Ahmad-Reza Sadeghi[1], Ivan Visconti[2], and Christian Wachsmann[1]

[1] Ruhr-University Bochum
Horst-Görtz Institute for IT-Security (HGI), Germany
{ahmad.sadeghi,christian.wachsmann}@trust.rub.de

[2] Dipartimento di Informatica ed Applicazioni
University of Salerno, Italy
visconti@dia.unisa.it

Abstract. Recently, operators of public transportation in many countries started to roll out electronic tickets (e-tickets). E-tickets offer several advantages to transit enterprises as well as to their customers, e.g., they aggravate forgeries by cryptographic means whereas customers benefit from fast and convenient verification of tickets or replacement of lost ones.
Existing (proprietary) e-ticket systems deployed in practice are mainly based on RFID technologies where RFID tags prove authorization by releasing spatio-temporal data that discloses customer-related data, in particular their location. Moreover, available literature on privacy-preserving RFID-based protocols lack practicability for real world scenarios.
In this paper, we discuss appropriate security and privacy requirements for e-tickets and point out the shortcomings of existing proposals. We then propose solutions for practical privacy-preserving e-tickets based on known cryptographic techniques and RFID technology.

Key words: Location Privacy, E-Tickets, RFID

1 Introduction

Electronic tickets (e-tickets) gain increasing popularity among operators of public transit networks. However, besides offering many advantages, e-tickets also introduce several risks, in particular concerning privacy of their users.

Benefits of e-tickets. Transit enterprises benefit from e-tickets in various ways: First, e-tickets help to decrease maintenance costs. Second, the number of fare dodgers is expected to decrease if tickets can be verified efficiently. Moreover, cryptographic means help to aggravate the problem of ticket forgery.

From the user perspective, e-tickets allow for faster and more convenient verification. Moreover, an e-ticket system can automatically select the lowest fare, which saves the customer's time and money. Finally, revocation of e-tickets enables transit enterprises to replace lost tickets, which is not possible for conventional paper-based ticket systems.

Threats. Besides their advantages, e-tickets also introduce several risks, in particular regarding the privacy of users. Since authentication of transit tickets typically involves spatio-temporal data, users are at risk to loose their privacy if this information is leaked to unauthorized parties. This means that e-tickets should ensure that no information on users (*confidentiality*) or their movements (*location privacy*) should be revealed to entities that are not trusted by the users. There are existing implementations of e-tickets that allow the creation of movement profiles and, in some cases, even disclose personal information of users (cf. Section 3). Moreover, since e-tickets contain digital data, they may be easily copied (*cloning*). Additionally, the corresponding protocols to issue or verify e-tickets may be subject to different attacks (e.g., man-in-the middle or replay).

Current situation. Currently, there is a vast amount of existing proprietary solutions for e-tickets. Since the corresponding specifications are usually not publicly accessible, there is no publicly known solution in practice that explicitly considers the privacy of users. We stress that user privacy preservation has not been claimed among the features of such systems.

The preferred technology to implement electronic transit tickets is Radio Frequency IDentification (RFID), which enables fully automated wireless identification of objects. A typical RFID system consists of *transponders* and *transceivers*. The main component of a RFID system is the transponder, which consists of an integrated circuit that is connected to an antenna. Typically, transponders are integrated into plastic cards or stickers that can be attached to the object to be identified and thus are often called *tags*. Since transceivers are mainly used to read data from tags, they are called *readers*. RFID tags can be used to realize e-tickets that are issued and verified by readers. Thus, in the rest of this paper "e-ticket" refers to tickets based on RFID.

Related work. There is a large body of literature on different approaches to realize privacy-preserving mechanisms for RFID (e.g., [17,16,2,31,14,20,22,9,18,25]). However, as pointed out in Section 3, most of these solutions are not applicable to e-tickets since each of them lacks some important security and functional requirements, as usability, security and privacy.

In [15], the authors motivate research for privacy in the context of e-tickets and provide a rough description of how anonymous credential [6] and e-cash [5] systems may be used to implement an anonymous payment system for public transit. However, they assume that devices realizing tickets can perform computationally demanding protocols (i.e., use public-key cryptography and intensive interaction), which is not a reasonable assumption for currently available cheap RF tokens. Since RFID tags are devices with very limited capabilities, one has to provide an acceptable level of privacy still preserving usability.

Summing up, an e-ticket system is an authentication scheme that involves spatio-temporal information and the design and secure implementation of a privacy-preserving and usable system based on RFID, is currently an interesting open problem.

Our contribution. In this paper we study the levels of privacy that could be achieved in an e-ticket powered system. We point out the weaknesses of known solutions and explore how known cryptographic tools can be applied to realize anonymization of e-tickets with currently available RFID technology, while having the goal to obtain a usable system that ensures no information disclosure on the user or his location to entities that are not trusted by the user.

Structure of the paper. In Section 2, we demonstrate the problems related to e-tickets by introducing the setting of electronic transit tickets, and define appropriate security requirements. In Section 3, we analyze several proposals from literature on how to realize anonymity for RF-tokens with limited capabilities and discuss their applicability to e-tickets. Section 4 describes how recent cryptographic tools can be applied in order to achieve the desired requirements. Finally, we conclude with Section 5 by describing some open problems and motivating further research.

2 Scenario of Electronic Transit Tickets

To introduce the problems related to e-tickets for public transportation, we first give a short overview of the general application scenario and point out potential weaknesses.

2.1 General Application Scenario

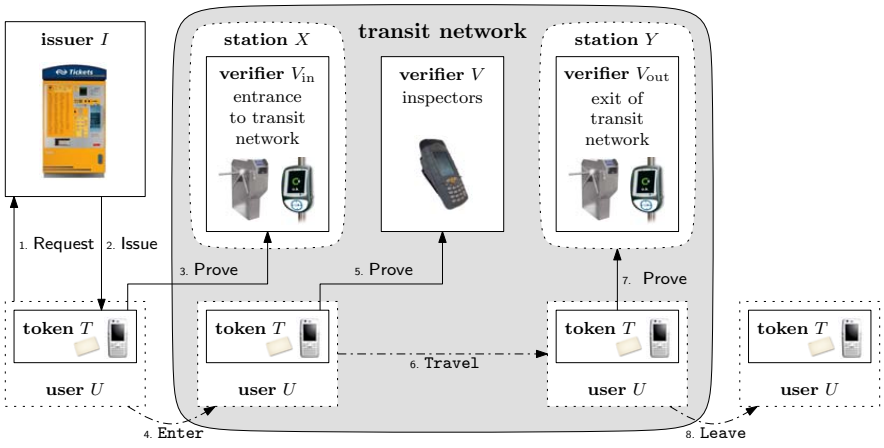

Fig. 1. General scenario for e-tickets.

An e-ticket system as shown in Fig. 1 is a *token-based authentication scheme* whereas tickets are represented as tokens (e.g., RFID tags). It consists of at least

one token issuing entity (*issuer*), a set of *users*, *tokens*, and *verifiers* who verify whether tokens are valid.

Typically, a user U must buy a token from token issuer I. Therefore, U selects his desired ticket and pays it. Issuer I then checks whether U is eligible to obtain a token (e.g., whether U paid for the ticket), and, if applicable, issues a token T and passes it to U. From now on, U is able to use token T to prove that he is authorized to use the transit network. This means that every user who is in possession of a token that has been issued by a genuine issuer is considered to be an *authorized user*.

Now assume that, as shown in Fig. 1, user U wants to travel from a place X to some location Y. Before U is allowed to enter the transit system at X, he must first prove to a verifier V_{in} at the entrance of the transit network that he is authorized to access it. If V_{in} can successfully verify the user's token, U is allowed to enter. Otherwise access will be denied. During his trip, U may encounter arbitrary inspections where he must prove that he is authorized to use the transit network. Thus, a verifier V may check the user's token T. If verification of T is successful, U is allowed to continue his trip. Otherwise, U must leave the transit network and may be punished for using it without authorization. After arriving at Y, the user's token T can be checked for a last time. Again, if T cannot be verified successfully, U may be punished.

Note that authentication is typically bound to some limitations. For instance, this may be some geographical or timely usage restrictions. Additionally, a token may be bound to the identity of its *owner* (i.e., the entity that bought the ticket).

2.2 Potential Attacks

Obviously, the main goal of a ticket system is to prevent ineligible users from using the transit system. Thus, the most prominent attack is to violate this goal. However, there are some other, subtle attacks which we are going to consider in the following.

Impersonation. The most obvious attack against e-ticket systems is motivated by unauthorized entities. The adversary must obtain or simulate a token that is accepted by an honest verifier. To achieve this, the adversary may perform various attacks including man-in-the-middle or replay attacks against the underlying authentication protocols, or he may attempt to create forged tokens, or to copy tokens of honest users.

Tracing. A more subtle attack aims at obtaining information on users and their movements within the transit network. For instance, the transit enterprise may be interested in information on the behavior of its customers. When using conventional authentication protocols, a token can be easily identified during verification. This enables verifiers to trace tokens within the transit network. Moreover, if a user uses an identifying payment method (e.g., a credit card) to buy a token, the issuer can link the token to the identity of its owner. Since the issuer and the verifiers are typically under the control of the same entity (e.g.,

the transit enterprise), this results in a complete loss of the user's privacy. However, in this case user information is managed by the transit enterprise that is a known entity. Thus it can be subject by law to commit on the honest use of the collected user data and can be monitored by means of inspections (similar observations hold for credit card companies).

The concrete threat instead, comes from unknown adversaries. Tokens typically are wireless devices and thus all their communication can be eavesdropped or manipulated by an adversary. Moreover the adversary may unnoticeably interact with tokens. As a consequence, the user's token may also be traced by entities different from the verifiers or the token issuer.

In summary, a primary goal is that the e-ticket system prevents disclosure of information on users or their movements to entities not trusted by the users.

Denial-of-service attacks. Another type of adversary may want to harm (e.g., to blackmail) the transit enterprise by preventing honest users from accessing the transit network. As already mentioned, tokens are wireless devices that can be attacked unnoticeably. This means that an adversary may try to exploit deficiencies of the protocols such that a ticket is no longer accepted by an honest verifier.

Depending on the underlying business model, protocols for e-tickets must be carefully crafted to prevent some or all of these attacks. In Section 2.3, we introduce different reasonable trust and adversary models and set up a complete list of requirements for e-ticket systems in Section 2.4.

2.3 Trust and Adversary Model

In an ideal setting, no entity must be trusted. However, in practice, the transit enterprise must at least trust issuer I to only create tokens for eligible users. Moreover, the transit enterprise must trust each verifier V to only accept tokens that have been issued by issuer I. These are reasonable assumptions since in practice, the token issuing entity and the verifiers are typically physically controlled by the transit enterprise.

Ideally, users should be anonymous to every entity, including issuer I and all verifiers V. However, due to technical restrains this is not always feasible in practice. Thus, a reasonable trust model for a practical solution is that users must at least trust issuer I and, dependent on the implementation, also all verifiers V. However, a trust model which only requires issuer I to be trusted is preferable.

To summarize, issuer I must trust all verifiers V. Moreover, all verifiers V must trust token issuer I. For users, there are three possible trust models:

TM 1: User U must trust token issuer I and all verifiers V.
TM 2: User U must only trust token issuer I.
TM 3: User U needs not to trust anyone.

TM 1 means that the e-ticket system must preserve privacy to all entities outside the system. This is the trust model primarily used for the solution presented

in Section 4. Considering TM 2, the e-ticket system must additionally protect the user's privacy to the verifiers. The solution presented in Section 4 can achieve this by assuming each verifier V to be connected to a remote server or to be equipped with a security module that is controlled by issuer I. However, these hardware assumptions may be difficult to achieve in practice. To realize TM 3, the e-ticket scheme must provide full anonymity. As discussed in Section 1, this seems to be possible only with high computational and communication resources, which is inappropriate for low-cost RFID devices.

It is also assumed that all communication that takes place during the process of creating a ticket cannot be eavesdropped or manipulated by an adversary. This is reasonable in practice since a user U may either use out-of-band communication or a secure channel to communicate to issuer I. However, following the traditional adversarial models, an adversary can eavesdrop all communication of a token T. Moreover, an adversary may perform active attacks on the corresponding protocols, which means that he can interact with all parties on the protocol level. Additionally, an adversary can corrupt tokens and verifiers (though this can only happen for a limited number of tokens and verifiers). The adversary is not allowed to corrupt the token issuer.

2.4 Requirement Analysis

Authentication. As mentioned in Section 2.1, the most important security goal for transit enterprises is authentication. Thus no unauthorized user (i.e., who is not in possession of a valid token) should be able to convince an honest verifier that he is authorized to access the transit system.

Another major requirement for any token-based authentication scheme is the resilience to remote tampering with tokens, which would allow denial-of-service attacks.

We summarize the security goals concerning authentication as follows:

Authentication: Only valid tokens are accepted by honest verifiers.
Unforgeability: Emulation and copying of valid tokens should be infeasible.
Availability: Unauthorized altering of token data must be infeasible.

Privacy. Since e-tickets enable efficient detection and identification of a huge number of tickets, a detailed dossier about user profiles (e.g., personal data or movements) can be created. The problem aggravates if tickets can be associated with the identity of their corresponding users since this results in a complete loss of user privacy.

Thus, the security objectives concerning privacy are:

Confidentiality: Unauthorized access to user-related data should be infeasible.
Anonymity: Unauthorized identification of tokens should be infeasible.
Location Privacy: Unauthorized tracing of tokens should be infeasible.

A stronger notion of location privacy considers traceability of tokens in case the internal state (i.e., the secrets) of a token has been disclosed. To distinguish traceability in past or future protocol runs, [18] consider the notion of *forward* and *backward traceability*.

Backward traceability: Accessing the current state of a token should not allow to trace the token in previous protocol runs.
Forward traceability: Accessing the state of a token should not allow to trace the token in future protocol runs.

In addition to these security and privacy requirements it is important to consider functional requirements for a practical solution.

Functional requirements. The costs per e-ticket should be minimal. Therefore, in case each ticket is implemented as a physical token (e.g., as RFID tag), the computational and storage requirements to the token should be as low as possible.

Additionally, verification of tickets must be fast. For instance, it should be possible to verify an e-ticket while a user is walking by, or shortly holding his ticket near a verifying device (e.g., while entering a bus). Therefore, protocols for e-tickets must be designed carefully to minimize the amount of computation and communication that must be performed. Moreover, an e-ticket system must be able to handle a huge amount of tokens.

Therefore, the functional requirements to e-tickets are:

Efficiency: Verification of tokens must be fast.
Scalability: The system should be able to handle a large amount of tokens.

Depending on the underlying business case and the technological restraints a practical realization may not fulfill all of these requirements.

3 Analysis of Existing Solutions

Most e-ticket systems are proprietary solutions whose specifications are not publicly available. This section exemplary shows the most common approach of implementing authentication of e-tickets in practice by the Calypso e-ticket system [1,26], of which at least some information is public. Moreover, to the best of our knowledge, there is no solution for e-tickets in practice that explicitly considers privacy of users.

Calypso e-ticket standard. Calypso is an e-ticket standard based on RFID tokens that is widely used in Europe and North and South America [1]. The roles in the Calypso system correspond to the model presented in Fig. 2. However, Calypso does not consider privacy of users and thus does not fulfill any of the privacy requirements of Section 2.4 w.r.t. any of the trust models presented

in Section 2.3. In fact, all transactions involving a Calypso e-ticket provide no confidentiality at all [26]. Moreover, Calypso tokens store personal data of their owner ("holder information") that can be queried by every verifier. Thus the Calypso e-ticket system leaks user-related information and allows the creation of movement profiles by everyone who is in possession of a standard RFID reader. However, all messages of a Calypso token are authenticated by a symmetric-key-based authentication mechanism. Thus, Calypso seems to fulfill all of the authentication requirements of Section 2.4.

Calypso implements a common approach to authenticate a low-cost RFID token based on a simple challenge-response protocol. Each token has a symmetric authentication key K_T that can be computed as a function of the serial number S_T of the token and a global master secret. All verifiers are equipped with a tamper-resistant security module (secure application module, SAM) that knows and protects this master secret and can be used as a black-box to compute K_T from S_T. To authenticate a token, a verifier sends a random challenge N_V to the token,which then computes $H_T \leftarrow f(K_T, N_V)$ where f is some one-way function. Finally, the token returns (S_T, H_T) to the verifier who uses its SAM to drive K_T and then verifies H_T. If verification is successful, the token has been authenticated. Obviously, this approach cannot provide privacy since all transactions of a token can be linked by its serial number S_T that is transmitted in clear in every protocol run. All subsequent transactions to update or to read data from a Calypso token are authenticated this way but are not encrypted.

Other e-ticket systems. There are many other proprietary solutions for e-tickets in practice. Most of them are based on widely used RFID transponders. Prominent examples are FeliCa [11] and MiFare [24].

FeliCa [11] is provided by Sony and is a contactless smartcard that is used mainly in the Asia-Pacific area for different purposes including e-tickets for public transportation.

MiFare is a family of contactless smartcards produced by Philips/NXP Semiconductors. These transponders are widely used for different purposes including e-tickets for public transportation. There were several publications on attacks against MiFare Classic transponders [21,23], that use a proprietary encryption algorithm that has been completely broken [7]. However, other MiFare products are claimed not to be affected.

The attacks on MiFare Classic transponders demonstrate a major problem of proprietary security solutions: Manufacturers of low-cost hardware try to find a compromise between speed and security of their products. Thus, they often implement proprietary lightweight crypto algorithms whose specifications are not public, and thus are typically not sufficiently evaluated. As the attack against MiFare Classic shows, these algorithms can often be reverse-engineered, which allows cryptanalysis or efficient key search by running the algorithms on more powerful hardware. In case of MiFare Classic, both ways enabled to break the security goals of these tags at a point in time where they were already widely used in practice.

3.1 Protocols for Anonymous Authentication

In an ideal e-ticket system, verifiers should learn nothing from the verification except that a token is genuine and valid. It is possible to realize this by using privacy-preserving techniques like anonymous credential systems [15]. An anonymous credential system is a cryptographic tool that enables zero-knowledge proofs of knowledge of certified data [19]. However, using anonymous credentials implies high computational (public-key cryptography) and typically also high communication (many rounds of interaction) requirements to all devices involved. Apparently, this is a contradiction to the functional requirements described in Section 2.4. Thus, these techniques are not applicable unless the e-ticket system can fall back upon appropriate mobile computing devices that are already possessed by the users. However, using mobile computing devices, like mobile phones, has several disadvantages. For instance, in case a user's phone runs out of power (which probably happens very often) he will no longer be able to prove authorization. Moreover, these devices can also be compromised by Trojans, which brings up new challenges. Furthermore, many users do not yet own a NFC[3] compatible mobile phone that has sufficient computing power to run computationally demanding protocols like anonymous credential systems or e-cash as proposed in [15].

3.2 Privacy-Preserving Protocols for RFID

There is a large body of literature on different approaches to implement privacy-preserving mechanisms for low-cost RFID transponders. For instance, [16] gives a comprehensive overview of different approaches. The author classifies RFID transponders as basic tags and symmetric-key tags. *Basic tags* refers to tokens that have no computational and no cryptographic capabilities. *Symmetric-key tags* means tags that are capable of performing at least some symmetric cryptographic functions (e.g., random number generation, hashing, or encryption). Using the classification of [16], we discuss the applicability of different proposed solutions to e-tickets.

Basic tags. As basic tags cannot perform any cryptographic operations they disqualify for authentication purposes. Tags that only provide wireless readable memory can only forward the data stored in their memory and thus are subject to replay and cloning attacks. This means that all data stored on such a tag can be read and be used to create identical copies or to simulate the original tag to an honest reader. Another problem related to cloning is *swapping*. This means that an adversary can copy the data stored on tag A to another tag B and vice versa and thus change the identities of these tags. Therefore, basic tags cannot fulfill the requirement of *unforgeability*.

[3] Near Field Communication (NFC) [10] is a RFID standard for contactless smartcards that is also supported by some currently available mobile phones.

Moreover, many solutions to enhance privacy of basic tags require tags to provide *many-writable* memory (e.g., [17,13,2]). The basic idea of these schemes is to frequently update the information stored on the tags such that an adversary cannot link them. However, due to the lack of secure access control mechanisms it is impossible to prevent unauthorized writes to such tags. A simple denial-of-service attack is to write some garbage data to a tag. Thus, an honest verifier will no longer accept the tag until it is reinitialized with correct data. This violates the *availability* requirement.

Therefore, tags that provide no cryptographic functionality cannot be used in applications that require reliable authentication. Thus, it is inevitable to use tags that are capable of performing at least some cryptographic functions if authentication is of concern.

Symmetric-key tags. A general problem of implementing privacy-preserving authentication based on symmetric keys is how to inform the other party which key must be used. Apparently, a tag cannot disclose its identity before the reader has been authenticated since this would violate its *location privacy*. Therefore, the reader does not know which authentication key it should use, and thus cannot authenticate to the tag. The basic idea to circumvent this problem has been introduced by [31] as *Randomized Access Control*:

Let $f_K(m)$ be a keyed one-way function on message m using key K. To authenticate to a reader, a tag first computes $h_T \leftarrow f_{K_T}(R)$ where K_T is a tag-specific key and R is a random value chosen by the tag. On receipt of (h_T, R), the reader forwards this tuple to a trusted server that computes $h_i \leftarrow f_{K_i}(R)$ for all keys $K_i \in \mathcal{K}$ where $\mathcal{K}$ denotes the set of the keys of all authorized tags. The server accepts if it finds a $K_i \in \mathcal{K}$ such that $h_i = h_T$. Finally, the server sends its decision whether to accept or reject the tag to the reader. Since R is randomly chosen each time the tag is queried, it always emits a different tuple (h_T, R) which cannot be linked to the tuples sent in previous protocol runs. Moreover, the reader does not learn the identity (i.e., the key K_T) of the tag since it only receives the response from the server. An obvious drawback of this solution is that the computational costs for the server to verify a tag are linear in the number of authorized tags. Therefore, this basic approach does not fulfill the *efficiency* and *scalability* requirement. Another disadvantage of this solution is that readers must have an online connection to the server, which, depending on the use case, may not be practical. Moreover, the tag must trust the server to respect its privacy since the server can identify the tag when it found the right key. Furthermore, this solution provides no security against replay-attacks and thus violates the *unforgeability* requirement. There is many subsequent work (including [20,9,18,25]) that follows and optimizes this approach by introducing new setup assumptions or by lowering the security or privacy requirements.

Other approaches rely on updating the identity of a tag each time it has been authenticated [14,27]. These approaches allow authentication of a tag in constant time. However, they require the verifiers to have permanent access to a

trusted database that verifies tags for them and manages all updates of the tag identities. As discussed above, this may be inappropriate for e-ticket systems.

Section 4 provides a simple solution that allows anonymous authentication of tags with constant computational costs for the readers without the need for a permanent online connection.

4 Solution for Practical Privacy-Preserving E-Tickets

RFID tags that are capable of performing public-key operations disqualify for practicable implementations of e-tickets because of their relatively high price and low performance. Thus, RFID tokens that are limited to symmetric-key cryptography (i.e., random number generation and hashing) are the most practical choice for e-tickets. However, as discussed in Section 3.2 the use of symmetric-key cryptography seems to have the drawback that at least the token issuer must be trusted not to disclose personal information or movement profiles of users. Thus, our solution is based on the trust and adversary model for e-tickets that we discuss in the following.

Trust and Adversary Model. Following Section 2.3, for e-tickets based on RFID tokens that are limited to symmetric cryptography, either trust model TM 1 or trust model TM 2 must be chosen. This means that a user U must at least trust token issuer I. Whether user U must additionally trust all verifiers V depends on the corresponding setup assumptions. This means that, if verifiers are considered to be untrusted, all operations that disclose user-related information must be dropped from the verifiers. For instance, these computations may be carried out on a local tamper-resistant[4] security module as it is done by many implementations in practice (cf. Section 3). Another simple approach used by various anonymous symmetric-key-based authentication protocols, is to employ a remote trusted server (cf. Section 3.2).

Model for Anonymous E-Ticket Systems. As discussed in Section 2.2, to provide privacy of users it is necessary to prevent tracing of tokens. This means that all entities that are not trusted by the user of a token should not be able to decide whether the user's token has been used in a protocol run (*unlinkability*).

Therefore, it is necessary to employ some mechanism that hides the identity of a token each time it is queried. This can either be some special hardware (e.g., as proposed by [2]) or a cryptographic primitive that inherently provides anonymity of users (e.g., anonymous credentials as proposed in [15]). In the following, we refer to this mechanism as *anonymizer*.

Analogous to Section 2.1, an anonymous e-ticket system consists of at least one token issuer, a set of users, tokens, verifiers, and anonymizers. The token issuer creates tokens for users. These tokens can be used by users to prove to verifiers that they are authorized to use the transit system. Additionally,

[4] Tamper-resistance means that the device will delete all its secrets when it detects any kind of physical tampering.

anonymizers ensure anonymity of tokens. We say that tokens are *anonymized*. Fig. 2 illustrates the model for anonymous e-ticket systems.

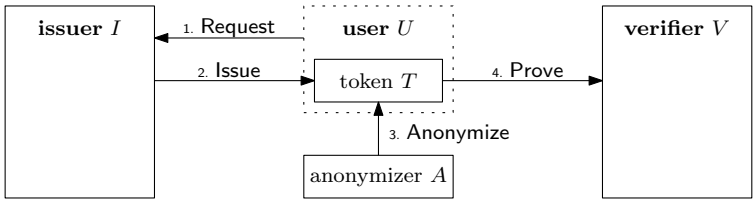

Fig. 2. Model for anonymous e-ticket systems.

Description of the Solution. In the following, we focus on solutions for privacy-preserving authentication based on RFID tokens that are at most capable of performing symmetric cryptography.

The players are as shown in Fig. 2. The anonymizer is either a dedicated hardware device or a software running on a mobile computing device (e.g., the mobile phone) of the user. Note that, a separate anonymizer device may suffer from the same problems as discussed in Section 3.1. However, in case a user's anonymizer runs out of power, the user will indeed loose some privacy until his anonymizer is operable again but he can still prove authorization using his RFID token. Moreover, since anonymizers can also be available in public places and their capabilities can be embedded in the verifiers (when this does not significantly affect the performance of the system), the user's privacy is not completely lost.

Since our solution relies on symmetric-key-based authentication, the token must store an authentication secret. To achieve the security requirement of *unforgeability*, it should be impossible to determine this secret by attacking the protocols involving the token, as well as by physically attacking the token. One solution to counterfeit physical attacks is to employ physical protection mechanisms that aggravate reading out the memory of the tag. However, this would increase the price of tag such that it would be improvident to use them. Another solution to prevent cloning can be implemented by means of a recent physical cryptographic primitive: Physically Unclonable Functions (PUFs) [28,29].

4.1 Building Blocks

Secure key storage with PUFs. A *Physically Unclonable Function* (PUF) is an inherently unclonable function embedded into a physical object [29]. The unclonability of the PUF comes from random and uncontrollable manufacturing processes during creation of the corresponding object. A PUF maps challenges to responses. A *challenge* is a stimulus that, when applied to the PUF makes it to return a *response* that is specific for the PUF w.r.t. to the stimulus. Since the

response of a PUF relies on physical properties of the corresponding physical object, which is subject to noise (e.g., temperature, pressure, etc.), the PUF will always return slightly different responses to the same stimulus.

A PUF can be embedded into a microchip, e.g., by exploiting statistical variations of delays of gates and wires within the chip. These deviations are unique for every sample from a set of chips that implement the same circuit. Therefore, in [28], the authors propose to use a PUF as secure key storage.

The adversary model for PUFs is that an attacker is assumed to know how the PUF is challenged and how responses are measured. Moreover, the attacker is allowed to know the exact challenges for deriving the secret stored in the PUF. The requirements to the chip that incorporates a PUF to securely store a secret are as follows [29]:

1. The PUF must be inseparably bound to the chip such that any attempt to separate them results in significant damage to the PUF and the chip.
2. Any measurements to the chip must not reveal detailed information on the structure of its PUF.
3. The PUF, the sensors for measuring responses, the processing unit, and the volatile memory of the chip must be opaque.
4. Even if details on the structure of the PUF are known, it must be infeasible to create a physical copy or to set up a mathematical model of the PUF that allows to predict challenge-response pairs with non-negligible probability (*unclonability*).
5. Tampering with the chip or the PUF must significantly change the challenge-response behavior of the PUF (*tamper-evidence*).
6. The chip must contain tamper-proof read-only memory that stores public data (e.g., algorithms) whose integrity is important.

The first requirement prevents an adversary from accessing the output of (and thus, the secret stored in) the PUF. The second prevents an adversary from collecting data that may help to create a clone or to set up a mathematical model of the PUF that can be used to obtain the secret. The third is to prevent any kind of attacks (e.g., side-channel attacks) that try to disclose the internal state of the PUF, the processing unit, or the volatile memory of the chip that may temporally contain parts of the secret. The fourth is to prevent cloning of the PUF. The fifth prevents invasive inspections, which means that any attempt to physically access or manipulate the chip or the PUF must destroy both of them. The last requirement prevents an attacker from injecting malicious code that may force the chip to disclose its secret.

Storing a secret in a PUF. To use a PUF as a secure key storage, a key K is generated and stored as follows: A trusted party (e.g., issuer I) first generates key $K \in_{\mathcal{R}} \{0,1\}^l$ using an appropriate security parameter l. Then, it chooses a random challenge z to challenge the PUF. On receipt of response r, the trusted party computes some helper data w such that key K can later be recovered by evaluating $\mathsf{PUF}(z, w)$ and stores (z, w, K) in a database. The tuple (z, w) is stored in the (unprotected) memory of the chip.

Helper data w has two different purposes [29]: First it should help to remove the effects of noise on measurements of the responses of the PUF, and second, since the responses of a PUF are typically not uniformly distributed, w should guarantee that secret K is uniform.

Reconstructing a secret from a PUF. To reconstruct secret K, the chip reads (z, w) from its memory, challenges its PUF with (z, w) and obtains K.

Efficient implementation. According to [28], a PUF can be integrated into a chip with less than 1000 extra gates. Moreover, [8] presents an implementation of a PUF for RFIDs.

Symmetric-key-based authentication. In order to authenticate tokens, standard authentication mechanisms based on symmetric-key cryptography that are secure against impersonation under passive (imp-pa) and active (imp-aa) attacks [4, p. 10] can be used. Since low-cost RFID tags are not capable of running multiple sessions, concurrent attacks (imp-ca) must not be considered. To provide *confidentiality* and *location privacy*, the authentication scheme must not disclose user-related information (e.g., user data or movement profiles).

As described in Section 3.2, the major problem of realizing anonymous authentication based on shared secrets is how to inform the other party about which secret should be used without revealing the own identity. This problem can be solved by employing rerandomizable public-key encryption [13,2].

Rerandomizable encryption. A *rerandomizable encryption scheme* means an encryption scheme for which there is a probabilistic function $\mathsf{Rand}(\cdot)$ that maps ciphertexts c to ciphertexts $c' \neq c$ such that the corresponding plaintext stays the same. The rerandomizable encryption scheme must be semantically secure [12] and should provide key privacy [3]. Semantic security means that, given two different chosen plaintexts $m_0 \neq m_1$, and a ciphertext $c_b = \mathsf{Enc}_{pk}(m_b)$ for some fixed public-key pk and $b \in_{\mathcal{R}} \{0, 1\}$, it should be hard to decide whether c_b encrypts m_0 or m_1. Key privacy means that, given two different public-keys $pk_0 \neq pk_1$, and a ciphertext $c_b = \mathsf{Enc}_{pk_b}(m)$ for some fixed message m and $b \in_{\mathcal{R}} \{0, 1\}$, it should be hard to decide whether c_b has been created by using pk_0 or pk_1.

Use of rerandomizable encryption. Rerandomizable public-key encryption can be used to provide a symmetric authentication key to authorized communication partners (e.g., trusted verifiers) without disclosing the identity of the token to unauthorized entities. Moreover the computations performed by the token are still contained in the more efficient symmetric-key setting.

During creation of a token T, the token issuer encrypts the *token authentication key* K_T of token T with a public encryption key pk_V whose corresponding secret decryption key is known to all verifiers (or their security modules or the trusted server). The resulting ciphertext $c_T = \mathsf{Enc}_{pk_V}(K_T)$ is then stored in the

memory of the token. Whenever token T engages a protocol run with a verifier, it first sends its ciphertext c_T. In case the recipient knows the correct decryption key, it can decrypt K_T and use it in a subsequent authentication protocol.

Since an honest verifier must verify that a token has been created by a genuine issuer, a digital signature scheme is used to certify the token authentication key. However, this signature is static data and thus cannot be transmitted to the verifier as plaintext since this would enable tracing of the token and thus violate *location privacy*. Therefore, the signature must be included into the rerandomizable ciphertext.

However, c_T is a static ciphertext and must be frequently rerandomized in order to provide *location privacy*. Therefore, anonymizers must read c_T, rerandomize it to $c'_T \leftarrow \mathsf{Rand}(c_T)$, and replace c_T with c'_T [2]. Since all known rerandomizable encryption schemes require public-key operations (which in turn implies modular exponentiations) to rerandomize a ciphertext, a symmetric-key token cannot rerandomize its ciphertext on its own. Thus, unlinkability relies on the availability of anonymizers that are not controlled by the token. Basically, there are four possibilities to realize anonymizers:

1. *Integrated anonymizers:* The anonymizer may be integrated into the token. This would enable gapless location privacy while improving practicability. However, all known rerandomizable public-key encryption schemes require to compute public-key operations (e.g., exponentiations) in order to rerandomize a ciphertext. Thus, this approach is not applicable to symmetric-key tags.
2. *Public anonymizers:* Anonymizers may be public, which means that they can be constructed and run by everyone. However, public anonymizers as proposed by [2] enable adversaries to put up malicious anonymizers that can perform denial-of-service attacks. Therefore, to fulfill security requirement *availability*, it is necessary that anonymizers are trusted by the users. In return, a trusted anonymizer must authenticate to a token before it is allowed to anonymize it. In practice there may be a variety of public anonymizing service providers the user may choose from the one he trusts.

   Authentication of anonymizers can be realized in the same way as described above for verifiers. Each token may be initialized with an additional rerandomizable ciphertext c_A that encrypts a token-specific symmetric *anonymizer authentication key* K_A under a public-key pk_A whose secret key sk_A is known to all anonymizers trusted to anonymize the specific token. Thus, only trusted anonymizers can decrypt c_A to obtain K_A and use it to authenticate to the token to be anonymized.
3. *Anonymizers controlled by transit enterprise:* Anonymizers may be controlled by the (trusted) transit enterprise. For instance, anonymizers may be included into verifiers or mounted at the stations or in the vehicles of the transit enterprise.
4. *User-controlled anonymizers:* Each user may own an anonymizer that can only be used to rerandomize his own tags. Therefore the user must provide

the public key pk_A of his anonymizer A to the token issuer during the process of issuing an e-ticket.

To summarize, the user must trust the anonymizer to respect his privacy. However, this is a reasonable assumption since the anonymizer is either under his control or managed by a trusted entity (e.g., the transit enterprise).

4.2 Protocol Descriptions

The issue protocol. A user U requests token issuer I to create a token with his desired usage conditions ρ_T (e.g., ticket type, expiration date, geographical usage restrictions, etc.) and therefore provides public-key pk_A of his anonymizer A. Issuer I then creates the token authentication key K_T and anonymizer authentication key K_A for token T. After that, issuer I derives the corresponding helper data (z_T, w_T) and (z_A, w_A) for the PUF of token T as described in Section 4.1. Then, issuer I creates a certificate $\sigma_T = \mathsf{Sign}_{sk_I}(K_T, \rho_T)$ and two rerandomizable ciphertexts $c_T = \mathsf{Enc}_{pk_V}(K_T, \rho_T, \sigma_T)$ and $c_A = \mathsf{Enc}_{pk_A}(K_A)$. Finally, issuer I writes the tuple (w_T, w_A, c_T, c_A) to the (unprotected) memory of token T and physically passes token T to user U.

The anonymize protocol. In order to anonymize a token T, anonymizer A broadcasts an anonymization request. On receipt of this request, token T uses its random number generator to create a random challenge N_T, reads both ciphertexts c_T and c_A from its memory, and sends the tuple (N_T, c_T, c_A) to anonymizer A that then uses the rerandomization function of the rerandomizable encryption scheme to rerandomize both ciphertexts (c_T, c_A) to (c'_T, c'_A). After that, anonymizer A uses its secret decryption key sk_A to decrypt the anonymizer authentication key K_A from ciphertext c_A, uses K_A to authenticate message (K_A, c'_T, c'_A, N_T), which A then sends to token T. On receipt of this message, token T recovers its anonymizer authentication key K_A by reading helper data w_A from its memory and challenging its PUF as described in Section 4.1. If token T can successfully verify the authenticity of tuple (K_A, c'_T, c'_A, N_T) w.r.t. to key K_A, token T updates both ciphertexts (c_T, c_A) stored in its memory to the ciphertexts (c'_T, c'_A) received from anonymizer A. If the authenticity of the response of anonymizer A cannot be verified, token T aborts.

The prove protocol. To verify the authenticity of token T, a verifier V first broadcasts a verification request. On receipt of this request, token T reads ciphertext c_T from its memory and sends it to the verifier V, who then uses its secret decryption key sk_V to decrypt (K_T, ρ_T, σ_T) from c_T. Then, verifier V uses public verification key pk_I of token issuer I to verify σ_T. If verification of σ_T fails or the usage conditions ρ_T associated with token T are violated, verifier V rejects. Otherwise, it continues by using K_T to engage an symmetric-key based authentication protocol with token T. Token T can recover its token authentication key K_T by reading helper data w_T from its memory and challenging its

PUF as described in Section 4.1. Verifier V accepts token T as authentic token if token T successfully completes the authentication protocol w.r.t key K_T. If authentication fails, verifier V rejects token T.

4.3 Analysis of the Framework

This section informally analysis which of the requirements of Section 2.4 are fulfilled by the solution presented in Section 4. We will provide formal proofs in an extended version of the security and privacy model of [30] in a follow-up paper.

Authentication. The solution presented in Section 4 fulfills all of the authentication requirements of Section 2.4:

Authentication: Honest verifiers will only accept tokens whose token authentication key has been certified by a genuine token issuer.

Unforgeability: The properties of the PUF, the underlying authentication protocol, and the semantic security of the rerandomizable encryption scheme ensure that a valid token cannot be cloned or simulated by an adversary since its secrets cannot be extracted. Moreover, the security of the digital signature scheme guarantees that an adversary cannot create valid tokens on his own since he cannot forge signatures.

Availability: The token only updates its internal memory with data that has been authenticated by an authorized (i.e., trusted) anonymizer. Thus, an adversary cannot tamper with the data stored on the token.

Privacy. The solution presented in Section 4 fulfills the following privacy requirements of Section 2.4 w.r.t. to trust model TM 2 (or trust model TM 3 if verifiers are equipped with security modules or are connected to a remote trusted server) as described in Section 4:

Confidentiality: Since the rerandomizable encryption scheme is required to be semantically secure, no information on the secrets of the token is revealed by the corresponding ciphertexts. Moreover, the underlying authentication scheme is required not to disclose any user-related information. Thus, an adversary cannot obtain any information on the token or the user.

Location Privacy: Tokens can be traced between two randomizations. However, if an adversary misses only one rerandomization, he cannot trace a token any more because of the semantic security of the rerandomizable encryption scheme and the properties of the authentication scheme.

Our framework currently does not provide backwards and forward traceability, and we leave this as an interesting open problem.

Functional requirements. The solution presented in Section 4 fulfills all of the functional of Section 2.4:

Efficiency: Verification of a token requires to run a symmetric-key-based authentication protocol between the token and the verifier. Moreover, a single public-key decryption and a single signature verification must be performed by the verifier. This computational effort is comparable to existing schemes currently used in practice (e.g., [1,26]) since the additional operations that must be performed by the verifier can be neglected due to the computing power of currently available RFID readers.

Scalability: The solution does not depend on the number of tokens.

5 Conclusion, Open Problems, and Future Work

Summary of contribution. We analyzed the viability of current proposals for privacy-preserving e-tickets and examined the applicability of privacy-enhancing RFID-based protocols. We showed that existing approaches are not suited for the application scenario of e-tickets and presented a solution based on existing cryptographic tools and current RFID technology.

Open research problems. As discussed in Section 3.2, all currently known privacy-preserving authentication schemes for tokens that are limited to symmetric cryptography seem to require the token issuer to be trusted. Therefore, it would be interesting to find a scheme based on symmetric cryptography but similar to the one that provides similar properties as anonymous credential systems.

Currently, our approach does not provide forward and backward security. Forward and backward-secure anonymous symmetric-key based authentication schemes require frequent update of the secrets of the tokens [18]. However, since secrets are protected by PUFs it is not trivial to update them for both, the token and the verifier, in a way that ensures forward and backwards traceability.

Acknowledgments. This work has been supported in party by the European Commission through the Network of Excellence ECRYPT II.

The work of the third authors has been also supported in part by the European Commission through the FP7 Information Communication Technologies programme, under Contract FET-215270 FRONTS (Foundations of Adaptive Networked Societies of Tiny Artefacts).

References

1. Calypso Networks Association. Web site of Calypso Networks Association. `http://www.calypsonet-asso.org/`, May 2007.
2. Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros. Untraceable RFID tags via insubvertible encryption. In *Proceedings of the 12th ACM Conference on Computer and Communications Security, November 7–11, 2005, Alexandria, VA, USA*, pages 92–101. ACM Press, 2005.

3. Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Gold Coast, Australia, December 9–13, 2001, Proceedings*, volume 2248 of *LNCS*, pages 566–582. Springer Verlag, 2001.
4. Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Security proofs for identity-based identification and signature schemes. Cryptology ePrint Archive: Report 2004/252, September 2004. Available at `http://eprint.iacr.org/2004/252`.
5. Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In *24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22–26, 2005, Proceedings*, volume 3494 of *Lecture Notes on Computer Science (LNCS)*, pages 302–321. Springer Verlag, 2005.
6. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15–19, 2004, Proceedings*, volume 3152 of *LNCS*, pages 56–72. Springer Verlag, 2004.
7. Nicolas T. Courtois, Karsten Nohl, and Sean O'Neil. Algebraic attacks on the Crypto-1 stream cipher in MiFare classic and oyster cards. Cryptology ePrint Archive, Report 2008/166, 2008. Available at `http://eprint.iacr.org/2008/166/`.
8. Srinivas Devadas, Edward Suh, Sid Paral, Richard Sowell, Tom Ziola, and Vivek Khandelwal. Design and implementation of PUF-based unclonable RFID ICs for anti-counterfeiting and security applications. In *IEEE International Conference on RFID 2008, Las Vegas, NV, USA, 16–17 April, 2008*, pages 58–64. IEEE Computer Society, 2008.
9. Tassos Dimitriou. A lightweight RFID protocol to protect against traceability and cloning attacks. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm), September, 05–09, 2005*, pages 59–66. IEEE Computer Society, 2005.
10. Near Field Communication Forum. Web site of Near Field Communication (NFC) Forum. `http://www.nfc-forum.org/`, April 2008.
11. Sony Global. Web site of Sony FeliCa. `http://www.sony.net/Products/felica/`, June 2008.
12. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
13. Philippe Golle, Markus Jakobsson, Ari Juels, and Paul Syverson. Universal re-encryption for mixnets. In *The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23–27, 2004, Proceedings*, volume 2964 of *LNCS*, pages 163–178. Springer Verlag, 2004.
14. Dirk Henrici and Paul Müuller. Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, March, 14–17, 2004*, pages 149–153. IEEE Computer Society, 2004.
15. Thomas S. Heydt-Benjamin, Hee-Jin Chae, Benessa Defend, and Kevin Fu. Privacy for public transportation. In *6th International Workshop, PET 2006, Cambridge, UK, June 28–30, 2006, Revised Selected Papers*, volume 4258 of *Lecture Notes on Computer Science (LNCS)*, pages 1–19. Springer Verlag, 2006.
16. Ari Juels. RFID security and privacy: A research survey. *Journal of Selected Areas in Communication (J-SAC)*, 24(2):381–395, February 2006.

17. Ari Juels and Ravikanth Pappu. Squealing euros: Privacy protection in RFID-enabled banknotes. In *7th International Conference, FC 2003, Guadeloupe, French West Indies, January 2003, Revised Papers*, volume 2742 of *LNCS*, pages 103–121. Springer Verlag, 2003.
18. Chae Hoon Lim and Taekyoung Kwon. Strong and robust RFID authentication enabling perfect ownership transfer. In *8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4–7, 2006, Proceedings*, volume 4307 of *LNCS*, pages 1–20. Springer Verlag, 2006.
19. Anna Lysyanskaya. *Signature Schemes and Applications to Cryptographic Protocol Design*. PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, Massachusetts, USA, September 2002.
20. David Molnar and David Wagner. Privacy and security in library RFID: Issues, practices, and architectures. In *Proceedings of the 11th ACM Conference on Computer and Communications Security, Washington, DC, USA, October 25–29, 2004*, pages 210–219. ACM Press, 2004.
21. Karsten Nohl and Henryk Plötz. MiFare — little security despite obscurity. `http://events.ccc.de/congress/2007/Fahrplan/events/2378.en.html`, 2007.
22. Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Efficient hash-chain based RFID privacy protection scheme. International Conference on Ubiquitous Computing (UbiComp), Workshop Privacy: Current Status and Future Directions, Nottingham, UK, September, 2004, September 2004.
23. Ronny Wichers Schreur, Peter van Rossum, Flavio Garcia, Wouter Teepe, Jaap-Henk Hoepman, Bart Jacobs, Gerhard de Koning Gans, Roel Verdult, Ruben Muijrers, and Ravindra Kali andVinesh Kali. Security flaw in MiFare Classic. `http://www.sos.cs.ru.nl/applications/rfid/pressrelease.en.html`, March 2008.
24. NXP Semiconductors. Web site of MIFARE. `http://mifare.net/`, May 2007.
25. Boyeon Song and Chris J. Mitchell. RFID authentication protocol for low-cost tags. In *Proceedings of the First ACM Conference on Wireless Network Security, Alexandria, Virginia, USA, March 31 – April 2, 2008.*, pages 140–147. ACM Press, 2008.
26. Spirtech. CALYPSO functional specification: Card application, version 1.3. `http://calypso.spirtech.net/`, October 2005.
27. Gene Tsudik. YA-TRAP: Yet Another Trivial RFID Authentication Protocol. In *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops, March 13–17, 2006*, volume 2802 of *LNCS*, pages 640–643. IEEE Computer Society, 2006.
28. Pim Tuyls and Lejla Batina. RFID-tags for anti-counterfeiting. In *The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13–17, 2005, Proceedings*, volume 3860 of *Lecture Notes on Computer Science (LNCS)*, pages 115–131. Springer Verlag, 2006.
29. Pim Tuyls, Boris Škoriç, and Tom Kevenaar, editors. *Security with Noisy Data — On Private Biometrics, Secure Key Storage, and Anti-Counterfeiting*. Springer-Verlag, 2007.
30. Serge Vaudenay. On privacy models for RFID. In *13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2–6, 2007, Proceedings*, volume 4833 of *LNCS*, pages 68–87. Springer Verlag, 2007.
31. Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and privacy aspects of low-cost radio frequency identification systems.

In *First International Conference on Security in Pervasive Computing, Boppard, Germany, March 12–14, 2003, Revised Papers*, volume 2802 of *LNCS*, pages 50–59. Springer Verlag, 2003.