A Privacy-Preserving Ticketing System

Kristof Verslype[1], Bart De Decker[1], Vincent Naessens[2],
Girma Nigusse[1], Jorn Lapon[2] and Pieter Verhaeghe[1]

[1] Katholieke Universiteit Leuven, Department of Computer Science,
Celestijnenlaan 200A, 3001 Heverlee, Belgium
firstname.lastname@cs.kuleuven.be
[2] Katholieke Hogeschool Sint-Lieven, Department of Industrial Engineering
Gebroeders Desmetstraat 1, 9000 Gent, Belgium
firstname.lastname@kahosl.be

Abstract. Electronic identity (eID) cards are deployed in an increasing number of countries. These cards often provide digital authentication and digital signature capabilities, but have at the same time serious privacy shortcomings. We can expect that ordering and issuing tickets for events (e.g. soccer matches) will be increasingly done using eID cards, hence, severely threatening the user's privacy. This paper proposes two alternative ticketing systems that are using the eID card in a bootstrap procedure, but still are providing a high degree of privacy to the user.

Keywords: Privacy, Anonymity, Security, Ticketing, Electronic Identity cards

1 Introduction

Tickets are used for an innumerable number of events: soccer matches, music festivals, exhibitions, etc. These tickets are ever more bought electronically. An increasing number of countries issue electronic identity cards to their citizens. Examples are Belgium, Estonia and Austria. These eID cards usually allow the holder to authenticate and to digitally sign documents, but often, they are very privacy unfriendly. For example, authentication using the Belgian eID card will usually lead to the divulgement of important personal data such as your national registration number (NRN). Despite these privacy dangers, the use of the eID card is promoted by the governments. We can thus expect that in the near future, electronic ticketing systems will arise based on the eID card. A trivial solution is easy to devise. However, this solution is not acceptable because it further endangers the card holder's privacy as profiles can easily be compiled, linked to each other and to the identity of the card holder. An advantage of the use of eID cards is that it is straightforward to impose restrictions on the maximum number of tickets that can be bought by one user, hence, thwarting sales on black markets. Sometimes, special offers are available for buyers under or over a certain age or living in the region where the event is organized. Here too, eID cards can help in securely conveying (proving) that these conditions are satisfied for the buyer. However, the use of these cards will usually disclose more information than is required.

For big events with thousands of attendants, the police would be helped if tickets were not anonymous, but could be linked to the identity of the attendants, or at least to the identity of the buyers of these tickets. Especially, when rows or riots occur, it would make it easier to identify and prosecute the instigators. However, the use of tickets attributable to individuals poses severe privacy risks and brings us closer to a "Big Brother" state.

This paper proposes two solutions where the eID card is needed to obtain an anonymized permit, allowing a user to obtain tickets in a privacy friendly way. The role of the eID card is thus reduced to a bootstrapping role. A first solution is based on pseudonym certificates, i.e. X.509 certificates containing a user's nym instead of a real identity. A second solution is based on the more enhanced anonymous credential systems, which allow to anonymously disclose only a subset of the personal attributes (or properties thereof) embedded in the credential. Both solutions are validated and compared with the trivial solution and with each other.

The main requirements are given in section 2. Section 3 introduces the required technologies. Section 4 explains notations and specifies the assumptions. Sections 5, 6 and 7 discuss the trivial protocol and two privacy friendly alternatives and are followed by a comparison in section 8. Sections 9 and 10 examine the related work, draw the conclusions and describe future work.

2 Requirements

The requirements are now summed up. F4 and F5 are optional.

Functional/Security Requirements

- **F1** Every event may have a policy that limits the number of tickets obtainable by one buyer. The policy may discriminate between different groups of buyers.
- **F2** Event organizers may choose to offer a subscription for a series of events.
- **F3** Every event can have a pricing policy that differentiates between different groups of buyers (e.g. youngsters or elderly people).
- **(F4)** When abuse is detected or when serious incidents happen during the event, it should be possible to identify the buyer of the ticket(s) involved, but only with a court order.
- **(F5)** Individuals who have been imposed a banning order for a particular event type, should not be able to buy tickets for this kind of events.

Privacy Requirements

- **P1** Buyers of tickets should not directly be identifiable.
- **P2** Except when subscriptions are used, it should not be possible to compile buyer's profiles.
- **P3** It should not be possible to identify an individual on a blacklist.

3 Technologies

The main technologies required in this paper are pseudonym certificates, anonymous credentials, commitment schemes and provable one-way functions.

3.1 Pseudonym Certificates

Pseudonym certificates [1] are traditional certificates where the identity information is replaced by a pseudonym. The certificate states that the identity of the user referred to by that pseudonym and the properties certified in the certificate have been verified by the issuer. Different shows of the same certificate are linkable, which can undermine anonymity.

The relevant functions for both classical and pseudonymous certificates are:

- $U \leftrightarrows I$: $Cert \leftarrow \mathsf{issueCertificate}(attributes)$. I issues a certificate $Cert$ to U. I knows the certificate *attributes*, but not the private key corresponding to $Cert$. Pseudonyms, ids, expiry date, etc. are also considered attributes.
- $U \rightarrow V$: $\mathsf{authenticate}(Cert)$. U proves possession of $Cert$ to verifier V. As a result, V gets to know all the attribute values embedded in $Cert$.

Enhanced Pseudonymous Certificates. We further extend the privacy without requiring considerable computational capabilities by replacing each certificate attribute *att* that contains personal properties (date of birth, social security number, etc.) by $H(att, \textsc{random})$. Showing such an enhanced pseudonym certificate thus only reveals personal data if the owner of the certificate also discloses the corresponding $(att, \textsc{random})$ tuple to the verifier. Evidently, the linkability issue persists.

3.2 Anonymous Credentials

Anonymous credential systems ([6], [5], [2]) allow for anonymous yet accountable transactions between users and organizations and allow for *selective disclosure* by showing properties of credential attributes (e.g. $age > 18$) while hiding all the other credential attribute information. In the Idemix system [5], different usages of the same credential are *unlinkable* (except when unique attribute values are revealed). Credentials can have features such as an expiry date, the allowed number of times it can be shown and the possibility to be revoked. A mix network ([11], [10]) is required to provide for anonymity at the network layer.

The (simplified) anonymous credential protocols relevant in this paper are:

- $U \leftrightarrows O$: $(Nym, Sig) \leftarrow \mathsf{generateSignedNym}(Cert)$. One can establish multiple non-transferable pseudonyms (i.e. nyms) with the same organization. Here, the user signs the established Nym giving O a provable link between the nym and the identity certified in $Cert$.

- $U \leftarrow I$: $\mathit{Cred} \leftarrow \mathsf{issueCredential}(\mathit{Nym}, \mathit{attributes})$. A credential is issued by I on a pseudonym Nym. The credential is known only to the user and cannot be shared. Also, a number of attributes, not necessarily known by I, is embedded into the credential.
- $U \leftrightarrows V$: $\mathit{transcript} \leftarrow \mathsf{authenticate}(\mathit{Cred}, \mathit{properties}, [\mathit{DeanCond}], [\mathit{Msg}])$. A user U authenticates to verifier V by proving possession of a valid credential *Cred*. U can selectively reveal credential attributes or properties thereof. The resulting transcript for V may be deanonymizable upon fulfillment of condition *DeanCond* (cfr. the $\mathsf{deanonymize}()$). U may decide to sign a message *Msg* with his credential by a provable link between the transcript and the message. Different transcripts for the same credential are unlinkable (unless the value of a unique attribute is proved).
- $U \rightarrow V$: $\mathsf{prove}(\mathit{properties})$. Simplified notation of the above function. *Properties* will refer to the credential used in the proof.
- D: $(\mathit{Nym}, \mathit{DeanProof}) \leftarrow \mathsf{deanonymize}(\mathit{transcript}, \mathit{condition})$. If a credential show is deanonymizable, the pseudonym Nym on which the credential was issued can be revealed by a trusted deanonymizer D. $DeanProof$ proves the link between the transcript and the nym. D is only allowed to perform the deanonymization when *condition* fulfills *DeanCond* (included in the transcript).

3.3 Commitments

A commitment [14, 7] hides one (or more) values. Later the committer can open the commitment, or prove properties of the committed value(s). The following (simplified) commitment methods are relevant:

- $(\mathit{Com}, \mathit{OpenInfo}) \leftarrow \mathsf{commit}(\mathit{attribute})$. A new commitment containing a single attribute is generated as well as the opening info required to prove properties about the commitment (or to open it).
- $U \rightarrow V$: $\mathsf{prove}(\mathit{Com}, \mathit{properties}, \mathit{OpenInfo})$. Prove properties of commitments.

3.4 Provable One-Way Functions

We define a provable one-way function $out \leftarrow f(in)$ as a one-way function whereof the one knowing in can prove that he knows a in such that $out = f(in)$ in a zero-knowledge proof. Multiple arguments are possible as well.

As an example, according to the DL assumption, $out \leftarrow g^{in} \bmod p$ is such a function for p prime and g a generator of a multiplicative group with order q with $q|p-1$ and p and q sufficiently large.

4 Assumptions and Notation

The general assumptions and notation w.r.t. the protocols are now summed up.

4.1 Assumptions

- For every protocol, a server always first authenticates to U using a classical X.509 certificate. Also, an integrity and confidentiality preserving connection is established during a protocol. Anonymity at the network layer is added when necessary.
- A ticketing server can service multiple events. However, for each event, there is only one ticketing server.
- Tickets do only contain a ticket identifier (e.g. event name, date and seat number) and are unforgeable.

4.2 Notation

- Each protocol requires the following roles: user U (client), ticket server T (issues tickets to users), event organizer E and the court of justice J.
- $U \leftrightarrows B \leftrightarrows T$: $(\mathrm{PayProof}_{\mathrm{U}}, \mathrm{PayProof}_{\mathrm{T}}) \leftarrow \mathsf{pay}(\mathit{price}, \mathit{Msg})$. U pays an amount of money, via an intermediary bank B, to T. A message can be linked to the payment. The bank can deliver proofs of the payment to both parties. The payment protocols can preserve U's privacy.
- $U \leftrightarrows T$: $(\mathit{desc}[], \mathit{price}, [\mathit{Proof}]) \leftarrow \mathsf{negotiate}(\mathit{Cert} \vee \mathit{Cred}, \mathit{Nym} \vee \mathit{Id}, \mathit{event}, \mathit{eventPolicy}, \#\mathit{tickets}, \mathit{specification})$ allows U and T to agree on the exact seat numbers as well as on the total price. Therefore, U gives an identifier (*Nym* or *Id*), shows (properties of) credential/certificate attributes. The event policy can state e.g. that people younger than 18 get reductions. Evidently, the number and (general) specification of the tickets are given as well. The restrictions on the blacklists can further constrain the possibilities of the user. U can give T a proof of the agreement (signed by *Cert* or *Cred*).
- O: $\mathit{Nym} \leftarrow \mathsf{retrieveOrGenerateNym}(\mathit{Id} \vee \mathrm{Nym}^*)$ returns a newly generated nym if the user refered to by *Id* or Nym^* does not yet have a nym with O. However, if that user already has been given a nym in the past, it is simply retrieved from O's local storage system.
- T: $\mathrm{Restrictions} \leftarrow \mathsf{retrieveRestrictions}(\mathit{Blacklist}, \mathit{Nym} \vee \mathit{Id})$. T looks up in a blacklist the restrictions of a person referred to by *Nym* or *Id*.
- G: $\mathrm{Restriction}[] \leftarrow \mathsf{getRestrictionBooleans}(\mathit{Id})$ implicitly uses all blacklists, and returns for each event type whether or not the user is blacklisted or not.
- Other, self explaining methods are: add(), lookup(), store (), update () and generateTickets().

5 Trivial eID-based Solution

Without alternatives, this protocol will most likely be implemented in Belgium as the government is really promoting the use of the eID card in commercial applications. However, this protocol has serious privacy drawbacks.

U uses his eID card to authenticate to T, revealing a lot of personal data to T. A government agency G maintains a blacklist containing identifiable user ids. This blacklist is checked by T before issuing tickets.

The user authenticates to T using his eID card. T first checks whether the user is blacklisted. Based on the user's id and personal attributes, the user can be given the possibility to buy a number of tickets as a result of the negotiation phase. After the payment and ticket issuance, T finally stores ticket selling info.

Identification in case of abuse is straight forward since T knows the *link* between the seat (or ticket) and the user's id.

The functional/security requirements are trivially fulfilled. However for the privacy requirements, this protocol fails completely. T knows the user's id and all other attributes contained in the eID certificate (P1). User profiling is trivial for T as well as sharing and linking of profiles (P2). The users' NRNs are on the blacklist (P3). In addition, many countries simply forbid blacklists on which users are identifiable due to privacy legislation. Deployment will often thus result in omitting the F5 requirement.

6 Solution based on Enhanced Pseudonym Certificates

6.1 Introduction

This approach improves the privacy of the user by introducing pseudonymous permits. First, each user is issued a unique pseudonymous root certificate by the government. This allows the user to obtain pseudonymous permit certificates from different permit servers. One permit server could for instance be responsible for one event type (e.g. soccer matches). With such a pseudonymous permit a user can buy tickets for events that happen in a small (permit specific) time period[3]. The user will thus most likely need multiple permits. The blacklists no longer contain user identifiers, but pseudonyms.

6.2 Roles

Besides the already defined U, T and E, a government agency G is needed to issue root certificates and a permit server PS issues permit certificates.

6.3 Assumptions

- All certificates contain a unique serial number, a pseudonym or id, a public key and an expiry date.
- There can be many pseudonym servers (PS) and many ticket servers (T).
- For every event, the ticket server (T) accepts permits issued by a limited set of pseudonym servers. However, the user sets of different pseudonym servers do not overlap (necessary for requirement F1).
- Only one entity G can issue valid pseudonymous root certificates.
- Nyms that are no longer valid, are forgotten by the permit server.

[3] The fixed time period is introduced to minimize linkabilities.

High Level Description and data structures. The user receives a pseudonymous root certificate ($\mathrm{Cert}^{\mathrm{R}}$), which contains a rootnym ($\mathrm{Nym}^{\mathrm{R}}$) and possibly other attributes (such as year of birth, citizenship, place of residency, ...). $\mathrm{Cert}^{\mathrm{R}}$ is used to authenticate to the permit server PS.
The user can apply to the PS for a pseudonym ($\mathrm{Nym}^{\mathrm{P}}$) that is valid during a predefined time period. $\mathrm{Nym}^{\mathrm{P}}$ will be certified in a (pseudonymous) permit certificate ($\mathrm{Cert}^{\mathrm{P}}$). Each certificate also contains a public key used to verify authentications with $\mathrm{Cert}^{\mathrm{P}}$, and possibly (properties of) other attributes that were copied from the root certificate ($\mathrm{Cert}^{\mathrm{R}}$). Using permit certificates with non-overlapping time-slots, each user can have at most one valid $\mathrm{Cert}^{\mathrm{P}}$ to order tickets for a particular event. The PS can refuse permits to users who have been sentenced to a banning order for events supported by the PS.

6.4 Protocols

Getting a root certificate. A governmental instance G assigns to each citizen one root pseudonym $\mathrm{Nym}^{\mathrm{R}}$. The first time U requests a root certificate $\mathrm{Cert}^{\mathrm{R}}$, a new $\mathrm{Nym}^{\mathrm{R}}$ is generated and included in $\mathrm{Cert}^{\mathrm{R}}$. In case the user was already assigned a $\mathrm{Nym}^{\mathrm{R}}$ in the past, that pseudonym is retrieved from G's local storage instead. G finally stores the user's NRN and $\mathrm{Cert}^{\mathrm{R}}$s (which include $\mathrm{Nym}^{\mathrm{R}}$).

Getting a permit certificate. U authenticates with a valid root certificate $\mathrm{Cert}^{\mathrm{R}}$ to the PS. PS will issue a number of permit certificates $\mathrm{Cert}^{\mathrm{P}}$s which have to be used before a (user specified) date (validThru). For instance, the user can request permit certificates that allow him to buy soccer tickets for the upcoming year. PS generates a set of nyms ($\mathrm{Nym}^{\mathrm{R}}$) or retrieves them (if they were already assigned in the past): one nym per time period[4]. Each nym $\mathrm{Nym}^{\mathrm{P}}$ is also certified in a permit certificate $\mathrm{Cert}^{\mathrm{P}}$ which also contains a validity period (for $\mathrm{Nym}^{\mathrm{P}}$), possibly a set of attributes, and an encryption of the user's root pseudonym $\mathrm{Nym}^{\mathrm{R}}$. The validity periods of $\mathrm{Nym}^{\mathrm{P}}$s are non-overlapping. Hence, users cannot buy tickets for the same event using different nyms. Also, when a user requests a new permit for the same period (e.g. because the previous one was lost or the private key was stolen), PS will always use the same nym ($\mathrm{Nym}^{\mathrm{P}}$). Each $\mathrm{Cert}^{\mathrm{P}}$ contains a probabilistic encryption of $\mathrm{Nym}^{\mathrm{R}}$ with the public key of J. This allows law enforcement to eventually identify the user involved in case of abusive behavior (see further). PS finally updates the list of $\mathrm{Cert}^{\mathrm{P}}$s that are issued to $\mathrm{Nym}^{\mathrm{R}}$. PS can derive the time intervals for which a $\mathrm{Nym}^{\mathrm{R}}$ has obtained a valid $\mathrm{Cert}^{\mathrm{P}}$ from that list.

Buying tickets for an event. The user first authenticates to the ticket server T using the permit certificate $\mathrm{Cert}^{\mathrm{P}}$ that is valid for that specific event and specifies the number of tickets he wants to order. T then obtains the restrictions

[4] The length of the non-overlapping time periods is chosen by the PS in such a way that the number of events that fall in each period is limited.

associated with $\mathrm{Nym}^{\mathrm{P}}$ on the blacklist. The user and the ticket server agree on the price of the tickets and the seats, based on the user's nym, allowing to limit the number of tickets for that user. The limitations and price can depend on certain attributes that are embedded in the permit certificate (such as the user's age) and on the restrictions on the blacklist. Finally, the ticket server updates the number of tickets that are sold to $\mathrm{Nym}^{\mathrm{P}}$ for that event.

Updating anonymous blacklists. To fulfill requirement F4, anonymous blacklists are used. Four entities are involved in updating blacklists (see table 2).
A law enforcement entity J forwards the court orders (NRN, Restrictions) to G. G substitutes the NRNs with the corresponding $\mathrm{Nym}^{\mathrm{R}}$s and forwards the list to the permit server PS. PS can then add $\mathrm{Nym}^{\mathrm{R}}$ to a blacklist for certain event types (i.e. PS will no longer issue $\mathrm{Cert}^{\mathrm{P}}$s to $\mathrm{Nym}^{\mathrm{R}}$ for the event types that are specified in the blacklist).
Finally, PS retrieves the valid $\mathrm{Nym}^{\mathrm{P}}$s for each $\mathrm{Nym}^{\mathrm{R}}$ with a banning order, substitutes every $\mathrm{Nym}^{\mathrm{R}}$-record in the blacklist with a number of $\mathrm{Nym}^{\mathrm{P}}$-records and forwards the new list to the ticket server T. T no longer issues tickets to pseudonyms in the blacklist. Note that the ticket service can even revoke tickets that were already issued to pseudonyms in the blacklist.

Identifying buyer of a ticket. To reveal the identity of a participant with a specified seat number, the ticket service T looks up the $\mathrm{Nym}^{\mathrm{P}}$ of the user that ordered the ticket. The corresponding permit certificate $\mathrm{Cert}^{\mathrm{P}}$ is kept by the ticket server and is passed to J. The latter can link $\mathrm{Cert}^{\mathrm{P}}$ to $\mathrm{Nym}^{\mathrm{R}}$ (as $\mathrm{Nym}^{\mathrm{R}}$ is encrypted with the public key of J in $\mathrm{Cert}^{\mathrm{P}}$). G can reveal the user behind $\mathrm{Nym}^{\mathrm{R}}$ (as G knows the mapping between NRN and $\mathrm{Nym}^{\mathrm{R}}$).

Increasing privacy with enhanced pseudonym certificates. For each personal attribute *att* in $\mathrm{Cert}^{\mathrm{R}}$, G can include the hash value $H(att, rand_G)$ in $\mathrm{Cert}^{\mathrm{R}}$ instead, where $rand_G$ is a random value. Each $rand_G$ is sent to U as part of the issuance of $\mathrm{Cert}^{\mathrm{R}}$. After authentication by U to a PS with $\mathrm{Cert}^{\mathrm{R}}$, the user sends to PS the $rand_G$ value of those attributes of $\mathrm{Cert}^{\mathrm{R}}$ that need to be included in the $\mathrm{Cert}^{\mathrm{P}}$ certificate. In a similar way, PS includes hash values using freshly generated $rand_{PS}$ values in the new $\mathrm{Cert}^{\mathrm{P}}$. This allows U to selectively disclose personal attributes to T. More complex constructions are possible as well, but are not discussed in this paper.

6.5 Evaluation

F1 This requirement is easily fulfilled as each user has only one $\mathrm{Nym}^{\mathrm{P}}$to buy tickets for a particular event.

F2 If $\mathrm{Nym}^{\mathrm{P}}$ can be used to order tickets for multiple events (e.g. multiple soccer games during a World Cup contest), T can even restrict the total number of tickets that can be bought for the whole contest (i.e. a set of events).

F3 A user can get a lower price for some tickets based on the attribute values of $\mathrm{Cert}^{\mathrm{P}}$. However, tickets can be passed on. Hence, T should be careful with price reductions.

(1.a) Getting a pseudonymous root certificate $\mathrm{Cert}^{\mathrm{R}}$		
(1) U → G	:	authenticate(eID)
(2) G	:	$\mathrm{Nym}^{\mathrm{R}}$← retrieveOrGenerateNym(eID.NRN)
(3) U ← G	:	$\mathrm{Cert}^{\mathrm{R}}$← issueCertificate({$\mathrm{Nym}^{\mathrm{R}}$, attributes . . . })
(4) G	:	store [eID.NRN, $\mathrm{Cert}^{\mathrm{R}}$]
(1.b) Getting a permit certificate $\mathrm{Cert}^{\mathrm{P}}$		
(1) U → PS	:	authenticate($\mathrm{Cert}^{\mathrm{R}}$)
(2) U → PS	:	validThru, *attributes to include*
(3) PS	:	∀ [from,till], from ≤ validThru:
(4) PS	:	$\mathrm{Nym}^{\mathrm{P}}$← retrieveOrGenerateNym($\mathrm{Cert}^{\mathrm{R}}$.$\mathrm{Nym}^{\mathrm{R}}$, [from,till])
(5) U ← PS	:	$\mathrm{Cert}^{\mathrm{P}}$← issueCertificate({$\mathrm{Nym}^{\mathrm{P}}$, [from,till], *attributes*, enc_{pk_J}(RANDOM $\parallel$ $\mathrm{Cert}^{\mathrm{R}}$.$\mathrm{Nym}^{\mathrm{R}}$)})
(6) PS	:	store [$\mathrm{Cert}^{\mathrm{R}}$.$\mathrm{Nym}^{\mathrm{R}}$. [from,till], $\mathrm{Cert}^{\mathrm{P}}$]
(1.c) Buying tickets		
(1) U → T	:	authenticate($\mathrm{Cert}^{\mathrm{P}}$)
(2) U → T	:	event, #tickets, specification
(3) T	:	Restrictions ← retrieveRestrictions($\mathrm{Cert}^{\mathrm{P}}$.$\mathrm{Nym}^{\mathrm{P}}$, EventType)
(4) U ⇆ T	:	(SeatNb[], price) ← negotiate($\mathrm{Cert}^{\mathrm{P}}$.$\mathrm{Nym}^{\mathrm{P}}$, event, #tickets, eventPolicy, $\mathrm{Cert}^{\mathrm{P}}$.attr, specification, [Restrictions])
(5) U, T	:	if (SeatNb[] = ⊘) abort
(6) U ⇆ B ⇆ T	:	pay(price, Hash(SeatNb[], . . .))
(7) U ← T	:	tickets[] ← generateTickets(SeatNb[])
(8) T	:	update [$\mathrm{Cert}^{\mathrm{P}}$, event, tickets[]]

Table 1. Protocols with pseudonym certificates

F4 Fulfilled (cfr. *"Identifying buyer of a ticket"* protocol).

F5 Three entities are needed to ban a user from event types for which a user already has a permit certificate, namely G, PS and T. Two entities are needed to ban a user from event types for which a user does not yet have a permit certificate, namely G and PS.

P1 As discussed in *"Identifying buyer of a ticket"*, four entities are needed to reveal the user's identity. Moreover, G (and maybe PS) are governmental instances. Hence, users can trust that players in the commercial sector (such as E and T) cannot identify users without help of governmental instances.

P2 Each $\mathrm{Nym}^{\mathrm{P}}$ only has a limited validity period. The number of tickets that is issued to the same $\mathrm{Nym}^{\mathrm{P}}$ is restricted. Hence, T and E can only compile limited profiles. PS can link all $\mathrm{Nym}^{\mathrm{P}}$s to the same $\mathrm{Nym}^{\mathrm{R}}$. However, multiple pseudonym servers PS can be used. If each PS can only issue permit certificates for specific types of events, the one PS cannot link multiple interests of the same $\mathrm{Nym}^{\mathrm{R}}$. Moreover, no PS obtains more personal attributes than needed. Only a subset of the attributes in $\mathrm{Cert}^{\mathrm{P}}$ are revealed to T by U when the latter wants to buy a ticket. Evidently, different T's affiliated with

(2.a) anonymizing the blacklists		
(1) J → G	:	[NRN, Restrictions, eventType]
(2) G	:	$\mathrm{Nym}^{\mathrm{R}} \leftarrow$ lookupNym(NRN)
(3) G → PS	:	[$\mathrm{Nym}^{\mathrm{R}}$, Restrictions, eventType]
(4) PS	:	$\mathrm{Nym}^{\mathrm{P}} \leftarrow$ lookupNym($\mathrm{Nym}^{\mathrm{R}}$, eventType)
(5) PS → T	:	[$\mathrm{Nym}^{\mathrm{P}}$, Restrictions, eventType]
(2.b) Identifying buyer of a ticket		
(1) J ← E	:	complaint, seatNb
(2) J → T	:	event, seatNb
(3) J ← T	:	[$\mathrm{Cert}^{\mathrm{P}}$, event, ticket] ← lookup(event, seatNb)
(4) J	:	(RANDOM ‖ $\mathrm{Nym}^{\mathrm{R}}$) ← dec_{prk_J}($\mathrm{Cert}^{\mathrm{P}}$.enc)
(5) J → G	:	$\mathrm{Nym}^{\mathrm{R}}$
(6) J ← G	:	NRN← lookup($\mathrm{Nym}^{\mathrm{R}}$)

Table 2. Protocols with pseudonym certificates (bis)

the same *PS* can collaborate in order to get hold of more personal attribute values.

P3 Only $\mathrm{Nym}^{\mathrm{R}}$'s and $\mathrm{Nym}^{\mathrm{P}}$'s are kept in blacklists.

7 A Ticketing System Based on Anonymous Credentials

7.1 Introduction

We further increase the user's privacy. The user needs a single permit - issued by a government agency - which allows the user to buy tickets for every event. In case of abuse, the transcript resulting from the permit show can be deanonymized. For each event type, there is a privacy-preserving blacklist, summing up the user's rights restrictions.

7.2 Roles

Besides *U*, *E*, *T*, and *J*, we define *G* as a government agency that issues permits and manages blacklists.

7.3 Assumptions

In the ticketing system based on anonymous credentials, we assume the following:

- The anonymous credential system provides the unlinkability property to permits. The user does not reveal identifiable permit attribute properties.
- All *E*s and all *T*s and *G* have a unique, publicly available provable one-way function; $\mathrm{f}^{\mathrm{E}}()$ for *E*, $\mathrm{f}^{\mathrm{T}}()$ for *T* and $\mathrm{f}^{\mathrm{G}}(.\ ,\ .)$ for *G*. Note that the latter requires two arguments. These functions could for instance be included in their X.509 certificate.

- The opening info generated by a commit method does not reveal any information about the content contained in the commitment. This is easily achieved using a symmetric key K:
  $Com^{new} \leftarrow (Com,\ enc_K(OpenInfo))$ and $OpenInfo^{new} \leftarrow K$ combined with integrity preserving measures (e.g. MACs).

7.4 High Level Description

The permit is an anonymous credential containing a set of personal attributes, a boolean value for each event type indicating whether or not the user is blacklisted, and two nyms. One nym ($\mathrm{Nym}^{\mathrm{R}}$) is known to G and used to blacklist persons. The other nym ($\mathrm{Nym}^{\mathrm{P}}$), is not known to G, but is used to generate an event specific nym, allowing T to keep track of the number of tickets sold to that person for that specific event.

Per event type, a blacklist is maintained by G. This blacklist contains user pseudonyms ($\mathrm{Nym}^{\mathrm{R}}$s). These nyms are converted to event specific nyms ($\mathrm{Nym}^{\mathrm{E}}$s) before the blacklist is sent to a specific T in order to avoid linkabilities.

7.5 Protocols

Getting an anonymous Permit Certificate. The actual issue of the permit (3.a.5) includes a subset of the user's personal attributes (*attributes*) contained in the user's eID. These can be selectively disclosed during a credential show protocol.

The permit contains for each event type a boolean Restrictions[EventType] stating whether or not the user is blacklisted. G can easily extract this information out of the blacklists it manages (cfr. below).

Each permit contains two user unique pseudonyms $\mathrm{Nym}^{\mathrm{R}}$ and $\mathrm{Nym}^{\mathrm{P}}$. $\mathrm{Nym}^{\mathrm{R}}$ is known to both U and G and is the nym under which the permit is issued by G. G possesses a provable link $\mathrm{Sig}^{\mathrm{R}}$ between the U's id and his $\mathrm{Nym}^{\mathrm{R}}$. This can be used in case of disputes.

The second pseudonym in the permit, $\mathrm{Nym}^{\mathrm{P}}$, is known to the user U only and is included in the permit as an attribute that is not known to G. This is done using a commitment, whereof U proves that he knows the corresponding *UserSecret* and $\mathrm{Nym}^{\mathrm{P}}$ (underlined in table 3) such that $\mathrm{Nym}^{\mathrm{P}} \leftarrow \mathrm{f}^{\mathrm{G}}(\mathrm{Nym}^{\mathrm{R}},$ *UserSecret*).

To obtain a new permit, after the previous one was lost, step 6 changes. After recalculating $\mathrm{Nym}^{\mathrm{P}} \leftarrow \mathrm{f}^{\mathrm{G}}(\mathrm{Nym}^{\mathrm{R}},$ *UserSecret*) and generating a new commitment $Com2 \leftarrow \mathsf{commit}(\mathrm{Nym}^{\mathrm{P}})$ (Step 4 and 5), U decrypts c, resulting in the opening info of the previous commitment. This allows U to prove that $Com.\mathrm{Nym}^{\mathrm{P}} = Com2.\mathrm{Nym}^{\mathrm{P}}$ (corresponds to step 6), convincing G that the same $\mathrm{Nym}^{\mathrm{P}}$ was used.

Buying a Ticket. For each ticket order, U sends $\mathrm{Nym}^{\mathrm{E}} \leftarrow \mathrm{f}^{\mathrm{E}}(\mathrm{Nym}^{\mathrm{P}})$ to T and proves possession of the corresponding $\mathrm{Nym}^{\mathrm{P}}$ (3.b.1,2). The use of one-way functions gives the user for each event a different, but event-unique nym. This gives T the possibility to limit the number of tickets per user while at the same time, this function avoids linking of T's customers to the customers of other Ts. Collusion with G does not help, because G does not even know Nym$^{\mathrm{P}}$.

When ordering a ticket, the user proves that he is not blacklisted by showing Restrictions[EventType]. If U is blacklisted, he sends Nym$^{\mathrm{T}}$$\leftarrow$ f$^{\mathrm{T}}$(Nym$^{\mathrm{R}}$) to T and proves that Nym$^{\mathrm{T}}$ is correctly formed with Cred$^{\mathrm{P}}$.Nym$^{\mathrm{R}}$. T now looks up the exact restrictions associated with Nym$^{\mathrm{T}}$ on the blacklist (3.b.3). This limits linking possibilities and possible collusion with G. The latter can only be done for blacklisted Us.

The negotiation phase (3.b.4) requires the user's permit as input, such that RequestProof can be generated. RequestProof is a proof for G that U did request the negotiated tickets at the negotiated price. This proof is also deanonymizable by J which provably reveals Nym$^{\mathrm{R}}$.

(3.a) Getting the first anonymous permit certificate Cred$^{\mathrm{P}}$

(1) U $\rightarrow$ G	:	authenticate(eID)
(2) G $\leftrightarrows$ U	:	(Nym$^{\mathrm{R}}$, Sig$^{\mathrm{R}}$) $\leftarrow$ generateSignedNym(eID.NRN)
(3) G	:	Restriction[] $\leftarrow$ getRestrictionBooleans(eID.NRN)
(4) **U** $\leftrightarrows$ G	:	Nym$^{\mathrm{P}}$$\leftarrow$ f$^{\mathrm{G}}$(Nym$^{\mathrm{R}}$, *UserSecret*)
(5) U $\rightarrow$ G	:	(*Com*, *OpenInfo*) $\leftarrow$ commit(Nym$^{\mathrm{P}}$)
(6) U $\rightarrow$ G	:	*Com*, prove($\underline{Com.\mathrm{Nym}^{\mathrm{P}}}$ = f$^{\mathrm{G}}$(Nym$^{\mathrm{R}}$, $\underline{UserSecret}$)), $c \leftarrow enc_{H(UserSecret)}(OpenInfo)$
(7) U $\leftrightarrows$ **G**	:	Cred$^{\mathrm{P}}$$\leftarrow$ issueCredential(Nym$^{\mathrm{R}}$, {*Com*.Nym$^{\mathrm{P}}$, Restriction[], *attributes*})
(8) G	:	store [eID.NRN, Nym$^{\mathrm{R}}$, Sig$^{\mathrm{R}}$, *Com*, *c*]

(3.b) Buying tickets

(1)	U $\rightarrow$ T	:	Nym$^{\mathrm{E}}$$\leftarrow$ f$^{\mathrm{E}}$(Cred$^{\mathrm{P}}$.Nym$^{\mathrm{P}}$), event
(2)	U $\rightarrow$ T	:	authenticate(Cred$^{\mathrm{P}}$, {Cred$^{\mathrm{P}}$.Nym$^{\mathrm{P}}$$\simeq$Nym$^{\mathrm{E}}$, Cred$^{\mathrm{P}}$.Restriction[EventType]})
(3)	T	:	if(Cred$^{\mathrm{P}}$.Restriction[EventType] == true) do
(3.a)	U $\rightarrow$ T	:	Nym$^{\mathrm{T}}$$\leftarrow$ f$^{\mathrm{T}}$(Cred$^{\mathrm{P}}$.Nym$^{\mathrm{R}}$)
(3.b)	U $\rightarrow$ T	:	prove(Nym$^{\mathrm{T}}$$\simeq$Cred$^{\mathrm{P}}$.Nym$^{\mathrm{R}}$)
(3.c)	T	:	Restrictions $\leftarrow$ retrieveRestrictions(Blacklist$_{\mathrm{T}}$, Nym$^{\mathrm{T}}$)
(3.d)	T	:	end if
(4)	U $\leftrightarrows$ T	:	(SeatNb[], price, RequestProof) $\leftarrow$ negotiate(Cred$^{\mathrm{P}}$, event, Nym$^{\mathrm{E}}$, #tickets, eventPolicy, [Restrictions])
(5)	U $\leftrightarrows$ B $\leftrightarrows$ T	:	(PayProof$_{\mathrm{U}}$, PayProof$_{\mathrm{T}}$) $\leftarrow$ pay(price, Hash(SeatNb[], . . .))
(6)	U $\leftarrow$ T	:	tickets[] $\leftarrow$ generateTickets(SeatNb[])
(7)	T	:	update [event, Nym$^{\mathrm{E}}$, RequestProof, tickets[]]

Table 3. Protocols with anonymous credentials

Blacklist Maintenance and Retrieval. A law enforcement entity J forwards the court orders (NRN, Restrictions) to G. G substitutes the NRNs with the corresponding Nym$^{\mathrm{R}}$s. Each Nym$^{\mathrm{R}}$is further converted to Nym$^{\mathrm{T}}$← f$^{\mathrm{T}}$(Nym$^{\mathrm{R}}$) before the blacklist is sent to a specific T to avoid linkabilities and profiling by T (4.b).

Misbehaviour and Deanonymization. Protocol 4.c illustrates how the collaboration of E, T and G is required in order to obtain a (provable) link between the ticket and the user's id. The proof is (RequestProof, deanProof, Sig$^{\mathrm{R}}$). If someone is put on a blacklist for EventType, his permit Cred$^{\mathrm{P}}$ is revoked. U can obtain a new Cred$^{\mathrm{P}}$, with the updated restrictions booleans Restriction[EventType], immediately.

		(4.a) Maintaining the blacklists
(1) J → G	:	Nym$^{\mathrm{R}}$, Restrictions, EventType
(2) G	:	Blacklists[EventType].add(Nym$^{\mathrm{R}}$, Restrictions)
(3) J → G	:	revokeCert(Nym$^{\mathrm{R}}$)
		(4.b) Obtaining a blacklist
(1) G	:	for each (Nym$^{\mathrm{R}}$, Restrictions) in Blacklists[EventType]: Blacklist$_{\mathrm{T}}$.add(f$^{\mathrm{T}}$(Nym$^{\mathrm{R}}$), Restrictions)
(2) T ← G	:	Blacklist$_{\mathrm{T}}$
		(4.c) Identifying buyer of a ticket
(1) J ← E	:	complaint, seatNb
(2) J → T	:	event, seatNb
(3) J ← T	:	RequestProof← lookup(event, seatNb)
(4) J	:	Nym$^{\mathrm{R}}$, deanProof ← deanonymize(RequestProof, *complaint*)
(5) J → G	:	(NRN, Sig$^{\mathrm{R}}$) ← lookup(Nym$^{\mathrm{R}}$)

Table 4. Protocols with anonymous credentials (bis)

7.6 Evaluation

We now evaluate by checking the requirements

Functional and Security Evaluation

F1 Nym$^{\mathrm{E}}$← f$^{\mathrm{E}}$(Nym$^{\mathrm{P}}$) enables T to link ticket orders of the same U for the same event.

F2 A subscription can be issued by T or a coordinating organization. It can be an anonymous credential that contains Nym$^{\mathrm{P}}$, Nym$^{\mathrm{R}}$, the Restriction[EventType] booleans and information about the subscription. It can be pseudonymously shown to a ticketing service in order to obtain tickets without a payment phase. Alternatively, a multiple-use ticket with an expiry date can be issued.

F3 The user can selectively disclose properties in the permit.

F4 is explained in section 7.5.

F5 is done using the anonymized blacklists. Revocation of tickets issued to persons that were blacklisted after the ticket order is possible if $\mathrm{Nym}^{\mathrm{R}}$ is systematically shown to T. However, the price is an increase in linkabilities.

Privacy Evaluation

P1 Deanonymization requires the collaboration of T, G and J as we argued in *Misbehaviour and Deanonymization.*

P2 We argued that a user has for each E a different $\mathrm{Nym}^{\mathrm{E}} \leftarrow \mathrm{f}^{\mathrm{E}}(\mathrm{Nym}^{\mathrm{P}})$. Different Es thus should know the user's $\mathrm{Nym}^{\mathrm{P}}$ – which remains hidden – to do linking. For blacklisted users, G can link $\mathrm{Nym}^{\mathrm{R}}$ and $\mathrm{Nym}^{\mathrm{T}}$. Collusion of T and G is then possible.

P3 G knows the links between nyms on a blacklist and the user's id. However, such convictions are publicly available. Collusion of T and G can reveal the identity associated with $\mathrm{Nym}^{\mathrm{T}}$.

8 Comparison and Feasibility

Table 5 compares the three approaches; the main functional/security requirements can be fulfilled while boosting privacy. To maintain user-friendliness, the interactions with e.g. PS can be done transparently to the user. The proposed solutions disallow a banned person to buy tickets for someone else (e.g. father for his children) and it is still possible that a person buys tickets and gives them to a banned person.

Estimates of the feasibility on the server side where done on an Intel 1.83GHz CPU. In the case of pseudonym certificates, steps 2.a.3 and 2.b.5, i.e. key generation, will be dominant if RSA is used; on average 377ms for 1024 bits and 4110 ms for 2048 bits. For the anonymous credential based protocols, issueCred and showCred/prove are dominant (steps 4.a.6, 4.a.7, 4.b.2 and optionally 4.b.3.b). showCred will require less than 400ms and less than 1,500ms for 1024 and 2048 bits respectively, while issuing lasts less than 600 ms and 2000 ms. Happily, obtaining (one or more) permit certificates will usually be spread in time.

9 Related Work

Ticketing framework [8], hybrid electronic ticketing [15] and ticket for mobile user and communication [3] [13] are valuable contributions for building future ticketing systems. However, except for [15], all fall short in properly addressing user privacy. In comparison, we propose two solutions that preserve the user's privacy and avoid arbitrary blacklisting.

Heydt-Benjamin et al.[15] propose a hybrid electronic ticketing system which uses passive RFID transponders and higher powered computing devices such as

	Trivial	**Pseudonym certs.**	**Anon. creds.**
F1 - # Tickets	✓	✓	✓
F2 - Subscription	✓	✓	✓
F3 - Pricing	✓	✓	✓
F4 - Deanon.	✓	✓- J interacts with E, T, PS, G.	✓- J interacts with E, T, G.
F5 - Ban	—	✓+ ticket revocability	✓(2)
P1 - User anon.	T knows user id	If no collusion of E, T, PS, G. T knows permit atts.	✓
P2 - User profiles	T can link everything.	Linkability during limited, fixed period.	✓(1)
P3 - Anon. blacklists	—	If no collusion PS, G.	only G can identify. U.

(1): If the user is blacklisted, G can collude with one or more Ts.

(2): Ticket revocability is possible at the cost of increased linkabilites.

Table 5. Comparison of the three approaches

smart phones or PDAs. Their hybrid ticketing system framework takes the advantage of e-cash, anonymous credentials and proxy re-encryption[9] to alleviate the concern of privacy in public transportation ticketing systems.

In general, anonymous credential protocols as described in [5], [4] commonly use a Trusted Third Party (TTP) to selectively deanonymize (or link) misbehaving users. However, Patrick et al. [12] strongly argued that deanonymizing a user with the help of TTP is a too heavy measure against a misbehaving user in a privacy-preserving system. Some applications might not necessarily need deanonymization to discourage misbehaving users, they can simply blacklist user pseudonyms, to block a user without actually revealing that user's identity. Thus, the authors propose a scheme where user misbehaviour is judged *subjectively* and blacklisted by each individual service provider (SP) without the need for TTP. Although subjective blacklisting reduces the size of a blacklist in comparison with the usual centralized blacklisting approach, it can empower a SP to arbitrarily discriminate (or freely blacklist) among its ticket users. In comparison, our protocols do not allow SPs to blacklist a user or to maintain its own blacklist. As discussed previously, in our protocols the blacklist is centrally managed by a *trusted* government instance and forwarded to the SPs. Moreover, arbitrary user blacklisting is forbidden without a judicial verdict.

10 Conclusions and Future Work

Two privacy preserving ticketing systems were proposed; one based on pseudonym certificates and one on anonymous credentials. We showed that it is possible to offer the user a high degree of privacy, while the other requirements remain fullfilled. Still the privacy unfriendly eID card is used as bootstrap.

A prototype implementation will be made, using an applet for registration and ticket ordering. Entering the event can be done using a bar code reader. The influence of mix networks on the overall performance must be examined.

Acknowledgements. This research is a contribution to the European PRIME project and is partially funded by the Interuniversity Attraction Poles Programme Belgian State, Belgian Science Policy, the Research Fund K.U.Leuven and the IWT-SBO project (ADAPID) "Advanced Applications for Electronic Identity Cards in Flanders".

References

1. N. Asokan, Els Van Herreweghen, and Michael Steiner. Towards a framework for handling disputes in payment systems. Technical Report RZ 2996, 1998.
2. S. Brands. A technical overview of digital credentials, 1999.
3. L. Buttyn and J. P. Hubaux. Accountable anonymous access to services in mobile communication systems. In *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems*, 1999.
4. J. Camenisch and E.V. Herreweghen. Design and implementation of the idemix anonymous credential system. In *ACM Computer and Communication Security.* 2002.
5. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pages 93–118, London, UK, 2001. Springer-Verlag.
6. David Chaum. Security without identification: transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985.
7. I. Damgard, T. Pedersen, and B. Pfitzmann. Statistical secrecy and multi-bit commitments, 1996.
8. K. Fujimura and Y. Nakajima. General-purpose digital ticket framework. In *Proceedings of the 3rd USENIX Workshop on Electronic Commerce*, pages 177–186, 1998.
9. M. Green G. Ateniese, K. Fu and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In *In: Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS)*, 2005.
10. Matt Hooks and Jadrian Miles. Onion routing and online anonymity. *CS182S*, 2006.
11. D. M. Goldschlag P. F. Syverson and M. G. Reed. Anonymous connections and onion routing. In *SP '97: Proceedings of the 1997 IEEE Symposium on Security and Privacy*, page 44, Washington, DC, USA, 1997. IEEE Computer Society.
12. A. Kapadia P. P. Tsang, M. H. Au and S. W. Smith. Blacklistable anonymous credentials: blocking misbehaving users without TTPs. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 72–81. ACM, 2007.
13. B. Patel and J. Crowcroft. Ticket based service access for the mobile user. In *In Proceedings of Mobicom'97*, 1997.
14. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pages 129–140, London, UK, 1992. Springer-Verlag.
15. B. Defend T. S. Heydt-Benjamin, H. Chae and K. Fu. Privacy for public transportation. In *Proceedings of the Sixth Workshop on Privacy Enhancing Technologies (PET 2006).* Springer, 2006.