Anonymous and Transferable Electronic Ticketing Scheme

Arnau Vives-Guasch[1], M. Magdalena Payeras-Capellà[2],
Macià Mut-Puigserver[2], Jordi Castellà-Roca[1(✉)],
and Josep-Lluís Ferrer-Gomila[2]

[1] Departament d'Enginyeria Informàtica i Matemàtiques,
Universitat Rovira i Virgili, Av. Països Catalans 26, 43007 Tarragona, Spain
{arnau.vives,jordi.castella}@urv.cat

[2] Dpt. de Ciències Matemàtiques i Informàtica, Universitat de les Illes Balears,
Ctra. de Valldemossa, km 7,5., 07122 Palma de Mallorca, Spain
{mpayeras,macia.mut,jlferrer}@uib.es

Abstract. Electronic tickets demonstrate, without the use of paper, the possession of an authorization or access to a determined service. In this scenario, some security requirements must be accomplished. Moreover, some determined services should guarantee the anonymity of the users in the system. In addition to these requirements, the transferability of a ticket from one user to another (without involving a third party) is useful but also generates other issues to be solved in terms of security, as several attacks could be performed. In this article we present an electronic ticketing system with anonymity and transferability based on the use of group signatures, giving a solution to enable linkability between several group signatures, and also proving their ownership with the use of Zero-Knowledge Proofs (ZKPs).

Keywords: E-ticketing · E-commerce · Transferability · Privacy · Security

1 Introduction

Information technologies (IT) are being extended progressively in our society. Users can access online services regardless of place and time. For example, they can purchase a movie or theater ticket online. Nonetheless, in some cases, they have to print the ticket to access to the service. In other words, the process is not completely electronic because a printed ticket is required.

Thanks to the introduction of smartphones, all the processes can be performed electronically. These devices offer a good computation power, high storage capacity, and also different communication technologies, such as Near Field Communication (NFC). All these features can be available on a small device. That allows mobility and flexibility and makes the system perfectly suitable for management, e-ticketing and e-payment schemes [5]. Electronic tickets can be

J. Garcia-Alfaro et al. (Eds.): DPM 2013 and SETOP 2013, LNCS 8247, pp. 100–113, 2014.
DOI: 10.1007/978-3-642-54568-9_7, © Springer-Verlag Berlin Heidelberg 2014

defined as a representation of the owner's rights to act as a user of a determined service, preserving the same requirements as the ones offered in paper format. We would like to emphasize the following properties: anonymity and transferability.

In the same way as paper tickets, electronic tickets have different properties according to the services where they are used. These services can be classified by the anonymity offered. For instance, a plane e-ticket cannot be anonymous: the identity of the passenger is a fixed parameter that is part of the e-ticket. In the e-tickets with revocable anonymity, the beneficiary can use the ticket demonstrating its possession but without any need of identification. This modality helps to avoid fraud related to the reuse of e-tickets. E-tickets with non-revocable anonymity are not linked to a user at all. The user who owns this e-ticket is the one who can use this service. The verification phase of the e-ticket should check if it has previously been used. For that reason, we need some kind of centralized verification. This alternative would not be applied when we are talking about reusable e-tickets (e.g. monthly tickets), with which many uses can be given by the same user.

Regarding the transferability property, there are several e-ticket sales and distribution companies that allow the e-ticket transfer[1,2]. Nonetheless, the typical transfers of e-tickets are performed through a central service and are non-anonymous. Moreover, recent studies in related fields incorporate transferability as a desired requirement [2,11]. We would like to transfer an electronic ticket in the same way that we can transfer a paper ticket, i.e. anonymously and without the participation of a central service. In such system, we should note that we are giving the rights linked to that ticket to another user when we transfer a ticket. In some cases, it needs a change in the beneficiary role, because some service parameters are affected: the right to transfer, the service disposal and the beneficiary identity. According to the right to transfer, the tickets can be granted to another user with (resale) or without any counterpart (loan).

We can find new systems using cryptographic techniques that enable the online e-ticket issue and verification [3,6,8,10,12,13].

These actions are really important if the users purchase the electronic tickets before their use, and the e-ticket is not able to be used everywhere with an online connection to a central database. In [13] a recent implementation of electronic tickets over mobile devices with NFC technology has been performed. Another example is the InMoDo system (Mobile Phone as a Ticket)[3], which has been adopted by the Swedish national train company, among others.

In some concrete systems, the receiver of an electronic ticket uses smartcards to carry it. This is the case of Oyster card for public transport in London. This system was designed in order to make the scanners work independently when the central system connectivity was down.

[1] http://www.ticketmaster.com/transfer

[2] https://www.e-ticket.lu

[3] http://inmodo.com/

1.1 Contribution

A complete survey in this field can be found in [9]. The previous proposals already analysed use a central service that synchronizes the transfer of e-tickets between the users and does not allow revocable anonymity. Thus, the main goal of our contribution is to preserve the security properties of (a) revocable anonymity, where the identity of users could be only revealed in case of misbehaviour, (b) transferability, where the electronic tickets can be transferred as a resale or loan without the collaboration of a central service, and (c) short-term linkability, where the user can easily demonstrate that she is the same user at both moments of receiving the ticket and later transferring it. This can be achieved by using the same cryptographic technique, short-term linkability with group signatures. By fulfilling these security requirements, they could allow to deploy transferable electronic ticketing in real scenarios.

1.2 Document Organization

First, Sect. 2 details a brief background explaining the cryptographic techniques. In Sect. 3, we explain all the system proposal with its desired properties, the entities and the phases. The security analysis is performed in Sect. 4, and in Sect. 5, we finally state the conclusions and future work.

2 Background

We use the short group signature (BBS) scheme [1] in order to verify that a user is a correct member of a certain group of users. Next, we introduce the main definitions related to the BBS signature, both the group signatures scheme and the Zero-Knowledge Proof (ZKP) of the group signatures.

2.1 Group Signatures Scheme

In this section we specify the procedures ($KeyGen_G$, $Sign_G$, $Verify_G$, $Open_G$, $SignLinkable_G$, $VerifyLinkable_G$) to be further used in the protocol with their parameters. $KeyGen_G, Sign_G, Verify_G$ and $Open_G$ are constructed from the same BBS scheme [1]. Both $SignLinkable_G$ and $VerifyLinkable_G$ have also been constructed in [7]. Consider bilinear groups G_1 and G_2 with respective generators g_1 and g_2.

Definition 1 ***The*** q***-Strong Diffie-Hellman problem*** *(q-SDH). Given two cyclic groups* G_1 *and* G_2 *of prime order* p*, two randomly chosen generators* $g_1 \in G_1$ *and* $g_2 \in G_2$ *of their respective groups, with an isomorphism* $\psi : G_2 \rightarrow G_1$ *where* $g_1 = \psi(g_2)$*, the q-SDH problem is a hard computational problem where the (q+2)-tuple* $(g_1, g_2, g_2^{\gamma}, g_2^{\gamma^2}, ..., g_2^{\gamma^q}) \in G_1 \times G_2^{q+1}$ *is the input and the pair* $(g_1^{\frac{1}{x+\gamma}}, x) \in G_1 \times \mathbb{Z}_p$ *is the output, for some* $x \in \mathbb{Z}_p^*$ *such that* $x + \gamma \neq 0$*.*

Definition 2 ***The Decision Linear Diffie-Hellman problem*** *(DLIN). Given a cyclic group G_1 of order p, and taking $u, v, h, u^a, v^b, h^c \in G_1$ as input, where $u, v, h \in G_1$ randomly chosen generators, and random $a, b, c \in \mathbb{Z}_p$, and output yes if $a + b = c$ and no otherwise.*

Suppose that the SDH assumption holds on (G_1, G_2), and that the DLIN assumption holds on G_1. The scheme uses a bilinear map $e : G_1 \times G_2 \to G_T$ and a hash function $H : \{0, 1\}^* \to \mathbb{Z}_p^*$. The public values are $g_1, u, v, h \in G_1$ and $g_2, w \in G_2$. Here $w = g_2^\gamma$ for some secret $\gamma \in \mathbb{Z}_p$. The functions are:

- $KeyGen_G(n)$. This algorithm takes a parameter n as input, which is the number of members of the group. The algorithm has the following steps:
  1. select a random value $h \overset{R}{\leftarrow} G_1 \backslash \{1_{G_1}\}$ and $gmsk = (\xi_1, \xi_2)$ where $\xi_1, \xi_2 \overset{R}{\leftarrow} \mathbb{Z}_p^*$, and set $u, v \in G_1$ such that $u^{\xi_1} = v^{\xi_2} = h$;
  2. select $\gamma \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and set $w = g_2^\gamma$; and
  3. generate for each user $\mathcal{U}_i$, $1 \leq i \leq n$, an SDH tuple (A_i, x_i) by performing: select $x_i \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and set $A_i \leftarrow g_1^{1/(\gamma + x_i)}$. The parameter γ is the private master key of the group key issuer.
- $Sign_G(gpk, gsk[i], M)$. Given a group public key $gpk = (g_1, g_2, h, u, v, w)$, a private user's key $gsk[i] = (A_i, x_i)$ and a message $M \in \{0, 1\}^*$, compute and output a signature of knowledge $\sigma = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$. Note that the tuple (T_1, T_2, T_3) is the linear encryption of A, that is: $(T_1, T_2, T_3) = (u^\alpha, v^\beta, Ah^{\alpha+\beta})$ for $\alpha, \beta \overset{R}{\leftarrow} \mathbb{Z}_p$. There are also some helper values $\delta_1 \leftarrow x\alpha$ and $\delta_2 \leftarrow x\beta$. The parameter c is the self-generated challenge (hash of the information in the commit information of the proof of knowledge). Finally, $(s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$ are the response values of the proof of knowledge.
  1. select $\alpha, \beta \overset{R}{\leftarrow} \mathbb{Z}_p$ and compute the linear encryption of A: $(T_1, T_2, T_3) \leftarrow (u^\alpha, v^\beta, Ah^{\alpha+\beta})$ together with the helper values $\delta_1 \leftarrow x\alpha$ and $\delta_2 \leftarrow x\beta$;
  2. select $r_\alpha, r_\beta, r_x, r_{\delta_1}, r_{\delta_2} \overset{R}{\leftarrow} \mathbb{Z}_p$ and compute the values $R_1 \leftarrow u^{r_\alpha}$, $R_2 \leftarrow v^{r_\beta}$, $R_3 \leftarrow e(T_3, g_2)^{r_x} \cdot e(h, w)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\delta_1} - r_{\delta_2}}$, $R_4 \leftarrow T_1^{r_x} \cdot u^{-r_{\delta_1}}$, $R_5 \leftarrow T_2^{r_x} \cdot v^{-r_{\delta_2}}$
  3. compute the challenge: $c \leftarrow H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$
  4. compute the values $s_j \leftarrow r_j + cj$ for $j \in \{\alpha, \beta, x, \delta_1, \delta_2\}$
  5. output $\sigma \leftarrow (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$.
- $Verify_G(gpk, M, \sigma)$. Given a group public key $gpk = (g_1, g_2, h, u, v, w)$, a message M and a group signature $\sigma = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$, verify that σ is a valid signature of the message.
  1. re-derive R_1, R_2, R_3, R_4, R_5: $\tilde{R_1} \leftarrow u^{s_\alpha}/T_1^c$, $\tilde{R_2} \leftarrow v^{s_\beta}/T_2^c$, $\tilde{R_3} \leftarrow e(T_3, g_2)^{s_x} \cdot e(h, w)^{-s_\alpha - s_\beta} \cdot e(h, g_2)^{-s_{\delta_1} - s_{\delta_2}} \cdot (e(T_3, w)/e(g_1, g_2))^c$, $\tilde{R_4} \leftarrow T_1^{s_x}/u^{s_{\delta_1}}$, $\tilde{R_5} \leftarrow T_2^{s_x}/v^{s_{\delta_2}}$
  2. checks that $c \overset{?}{=} H(M, T_1, T_2, T_3, \tilde{R_1}, \tilde{R_2}, \tilde{R_3}, \tilde{R_4}, \tilde{R_5})$.
- $Open_G(gpk, gmsk, M, \sigma)$. This algorithm is used in order to trace a signature to a concrete signer inside the group. It is only available for the group manager, as she is the holder of the $gmsk$ master key and knows all the pairs (A_i, x_i). Given a group public key $gpk = (g_1, g_2, h, u, v, w)$, the group

master private key $gmsk = (\xi_1, \xi_2)$, a message M and a signature $\sigma = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$, it proceeds as follows. First, recover the user's A by performing $A \leftarrow T_3/(T_1^{\xi_1} \cdot T_2^{\xi_2})$. If the elements $\{A_i\}$ of the user's private keys are given to the group manager, then she can look up the user index corresponding to the identity A recovered from the signature.

– $SignLinkable_G(gpk, gsk[i], M', \sigma, \alpha, \beta)$. Given a group public key gpk, a private user's key $gsk[i]$, a new message M', a previous signature σ, and the values α, β used for that signature, compute and output a signature σ'. In order to use this procedure correctly, it is defined as follows:
  - First use: standard $Sign_G(gpk, gsk[i], M)$ obtaining a group signature σ and using (α, β).
  - Further uses: $SignLinkable_G(gpk, gsk[i], M', \sigma, \alpha, \beta)$:
    1. use the same pair (α, β) producing the same linear encryption of A as in the first time: $(T_1,\ T_2,\ T_3) = (u^\alpha,\ v^\beta,\ Ah^{\alpha+\beta})$; and
    2. given a message M', sign the message and output a signature $\sigma' \leftarrow (T_1, T_2, T_3, c', s'_\alpha, s'_\beta, s'_x, s'_{\delta_1}, s'_{\delta_2})$ where
    $c' \leftarrow H(M', T_1, T_2, T_3, R'_1, R'_2, R'_3, R'_4, R'_5) \in \mathbb{Z}_p$.

  It becomes trivial to verify that several signatures are produced by the same user, as the information (T_1, T_2, T_3) is public in the same signature. In addition, the random values $(r_\alpha, r_\beta, r_x, r_{\delta_1}, r_{\delta_2})$ must be different from the previous times, that is: $(r_\alpha{}' \neq r_\alpha, r_\beta{}' \neq r_\beta, r_x{}' \neq r_x, r_{\delta_1}{}' \neq r_{\delta_1}, r_{\delta_2}{}' \neq r_{\delta_2})$ in order not to reveal information.

– $VerifyLinkable_G(\sigma, \sigma')$. This algorithm takes two signatures σ and σ' as input and outputs $true$ or $false$ depending on whether the signatures have been produced by the same signer's pseudonym: $(T_1 \overset{?}{=} T_1{}', T_2 \overset{?}{=} T_2{}', T_3 \overset{?}{=} T_3{}')$.

2.2 ZKP of the Group Signatures Scheme

In our proposal, both standard and linkable group signatures are used, as they enable to verify the internal message information as well as to verify that determined signatures related to the same event or e-ticket belong to the same *anonymous* user. Despite these advantages, the signatures are generated by the same user, and the verifications can be performed *offline*, that is, the verifier does not take a role during the signature generation, so this verification needs to be performed. Then we detail the procedures $ZKP_GCommit$, $ZKP_GResponse$ and $ZKP_GVerify$:

– $ZKP_GCommit(M^*)$. This procedure is performed by the user that wants to demonstrate (prover) to another user (verifier) that she is the right holder of the ticket. This part is the commitment, the first procedure. Given a public group key $gpk = (g_1, g_2, h, u, v, w)$, a group private key for the user $gsk[i] = (A_i, x_i)$ and a signed message $M^* = (M, \sigma)$ where $\sigma = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$, it generates the commitment $m' = (T_1, T_2, T_3, R'_1, R'_2, R'_3, R'_4, R'_5)$ as output.

1. we have to demonstrate the ownership of the values $(\alpha, \beta, x, \delta_1, \delta_2)$ that have been generated by the signature of M^*, keeping then the resulting values with the linear encryption of A: $(T_1, T_2, T_3) = (u^\alpha, v^\beta, Ah^{\alpha+\beta})$;
2. the values $r_\alpha{}', r_\beta{}', r_x{}', r_{\delta_1}{}', r_{\delta_2}{}' \overset{R}{\leftarrow} \mathbb{Z}_p$ are selected and then the following values are generated:
   (a) $R'_1 \leftarrow u^{r_\alpha{}'}$; $R'_2 \leftarrow v^{r_\beta{}'}$;
   (b) $R'_3 \leftarrow e(T_3, g_2)^{r_x{}'} \cdot e(h, w)^{-r_\alpha{}' - r_\beta{}'} \cdot e(h, g_2)^{-r_{\delta_1}{}' - r_{\delta_2}{}'}$;
   (c) $R'_4 \leftarrow T_1^{r_x{}'} \cdot u^{-r_{\delta_1}{}'}$; $R'_5 \leftarrow T_2^{r_x{}'} \cdot v^{-r_{\delta_2}{}'}$.
3. the output $m' = (T_1, T_2, T_3, R'_1, R'_2, R'_3, R'_4, R'_5)$ is generated.

- $ZKP_G Response(m', c')$. This procedure is the second part of the ZKP, where the user responds to the challenge of the verifier given a first commitment. Given a commitment m' where $m' = (T_1, T_2, T_3, R'_1, R'_2, R'_3, R'_4, R'_5)$ and a challenge c' given by the verifier, the prover generates the response $s' = (s_\alpha{}', s_\beta{}', s_x{}', s_{\delta_1}{}', s_{\delta_2}{}')$ where their values are given by: $s'_j \leftarrow r'_j + c'j$ for $j \in \{\alpha, \beta, x, \delta_1, \delta_2\}$
- $ZKP_G Verify(m', c', s')$. This part is performed by the verifier to check that the commitment, challenge and response of the ZKP match. Given a commitment m' where $m' = (T_1,\ T_2,\ T_3,\ R'_1,\ R'_2,\ R'_3,\ R'_4,\ R'_5)$, a challenge c' given by the verifier, and the response $s' = (s_\alpha{}',\ s_\beta{}',\ s_x{}',\ s_{\delta_1}{}',\ s_{\delta_2}{}')$ provided by the prover, the verifier checks
  1. $u^{s_\alpha{}'} \stackrel{?}{=} T_1^{c'} \cdot R'_1$; $v^{s_\beta{}'} \stackrel{?}{=} T_2^{c'} \cdot R'_2$;
  2. $e(T_3, g_2)^{s_x{}'} \cdot e(h, w)^{-s_\alpha{}' - s_\beta{}'} \cdot e(h, g_2)^{-s_{\delta_1}{}' - s_{\delta_2}{}'} \stackrel{?}{=} (e(g_1, g_2)/e(T_3, w))^{c'} \cdot R'_3$;
  3. $T_1^{s_x{}'} \cdot u^{-s_{\delta_1}{}'} \stackrel{?}{=} R'_4$; $T_2^{s_x{}'} \cdot v^{-s_{\delta_2}{}'} \stackrel{?}{=} R'_5$.

3 Description of the System

In this section, we describe our system proposal. First of all, the security requirements to achieve are briefly introduced in Sect. 3.1. Then, the details of the protocol are presented in Sect. 3.2.

3.1 Requirements

We can classify e-ticket requirements into two categories [9]. On the one hand, we have security requirements, and on the other hand we have functional requirements for e-tickets.

The desired main security requirements for our proposal are:

- Authenticity: the generated ticket is demonstrably genuine.
- Non-repudiation: the issuer of the ticket cannot deny its generation.
- Integrity: the ticket cannot be modified after its issue.
- Revocable anonymity: the ticket is anonymous, but if the user performs an unauthorized use, then it can be identified.
- Short-term linkability: different movements of a same user, although anonymous, cannot be traced between them, in order to avoid generation of profiles. However, a user could demonstrate being the same user in determined movements of a same journey/action, or to demonstrate ownership of an action.

- Non-overspending: the ticket cannot be used more times than the established in the issue.
- Transferability: the ticket can be transferred to other users, without losing any of the requirements previously stated.

The functional requirements considered are the following:

- Validity time: the ticket can be valid until an established time.
- Online/Offline: ticket verification can require (or not) a connection to the Internet or a centralized system for checking.

3.2 Details of the Protocol

There are three main entities in the system, User ($\mathcal{U}$), Issuer ($\mathcal{I}$) and Service provider ($\mathcal{P}$), and also a Group Manager $\mathcal{M}_{\mathcal{G}}$ that only interacts in case of conflict. There are six phases: *Ticket Issue*, between $\mathcal{I}$ and $\mathcal{U}$; *Ticket Transfer (first time)*, between two users $\mathcal{U}_1$ and $\mathcal{U}_2$; *Ticket Transfer (following 'k' times)*, also between two users $\mathcal{U}_1$ and $\mathcal{U}_2$; *Ticket Verification (standard)*, between $\mathcal{U}$ and $\mathcal{P}$; *Ticket Verification (transferred)*, also between $\mathcal{U}$ and $\mathcal{P}$; and finally the *Revocation of anonymity* phase, which is only used in case of conflict, and which can be called by the Group Manager $\mathcal{M}_{\mathcal{G}}$.

Ticket Issue. In this protocol, $\mathcal{U}$ receives a valid ticket from $\mathcal{I}$ in order to be later used, or transferred. It works as follows:

1. $\mathcal{I}$ generates and sends a random value $n_\alpha \overset{R}{\leftarrow} \mathbb{Z}_p$;
2. $\mathcal{U}$:
   (a) selects the service Sv;
   (b) generates $\mathsf{V}^* = (\mathsf{V}, \widehat{\mathsf{V}})$, where $\mathsf{V} = (\mathsf{Sv}, n_\alpha, \mathsf{flag\_issue})$ and $\widehat{\mathsf{V}} = Sign_G(gpk, gsk[i], \mathsf{V})$ is the group signature;
   (c) sends V^* to $\mathcal{I}$;
3. $\mathcal{I}$:
   (a) verifies the group signature: $Verify_G(gpk, \mathsf{V}, \widehat{\mathsf{V}})$;
   (b) generates the ticket information: $\mathsf{T} = (\mathsf{Sn}, \mathsf{V}, \mathsf{Ti}, \mathsf{Tv}, \mathsf{Tc})$ where the ticket includes V received from $\mathcal{U}$, Sn as the ticket serial number, Ti as the date of issue, Tv as the validity time, and Tc as the terms and conditions;
   (c) signs the ticket: $\mathsf{T}^* = Sign_{\mathcal{I}}(\mathsf{T})$ where this signature could be a standard RSA-like signature; and
   (d) sends the ticket T^* to $\mathcal{U}$;
4. $\mathcal{U}$ verifies the signature of T^*.

Ticket Transfer (First Time). In this protocol, $\mathcal{U}_1$ transfers the original ticket to $\mathcal{U}_2$ by giving the permission to use it with a group signature which is linked to the commitment of the issued ticket V^*. It works as follows:

1. $\mathcal{U}_1$:
   (a) generates a commitment $m_{\beta_0} = ZKP_G Commit(\mathsf{T}^*)$; and
   (b) sends the commitment and the ticket $(m_{\beta_0}, \mathsf{T}^*)$;
2. $\mathcal{U}_2$:
   (a) verifies the information and signature of T^*: $Verify_{\mathcal{I}}(\mathsf{T}^*)$;
   (b) verifies the group signature: $Verify_G(gpk, \mathsf{T.V}, \mathsf{T.}\widehat{\mathsf{V}})$;
   (c) generates a random value $n_{\lambda_0} \xleftarrow{R} \mathbb{Z}_p$; and
   (d) generates the first *challenge* with the number of transferred times $k = 0$ and the agreed *price* for the transfer: $c_{\beta_0} = H(n_{\lambda_0}, k = 0, price, \mathsf{flag\_transfer})$ and sends it;
3. $\mathcal{U}_1$:
   (a) generates the response $s_{\beta_0} = ZKP_G Response(m_{\beta_0}, c_{\beta_0})$;
   (b) generates a random value $n_{\beta_0} \xleftarrow{R} \mathbb{Z}_p$; and
   (c) sends $(s_{\beta_0}, n_{\beta_0})$;
4. $\mathcal{U}_2$:
   (a) verifies the response: $ZKP_G Verify(m_{\beta_0}, c_{\beta_0}, s_{\beta_0})$; and
   (b) generates $\mathsf{W_0}^* = (\mathsf{W_0}, \widehat{\mathsf{W_0}})$ where $\mathsf{W_0} = (n_{\beta_0}, \mathsf{T}^*, \mathsf{flag\_transfer})$ and its group signature $\widehat{\mathsf{W_0}} = Sign_G(gpk, gsk[i], \mathsf{W_0})$ and sends it;
5. $\mathcal{U}_1$:
   (a) verifies the group signature: $Verify_G(gpk, \mathsf{W_0}, \widehat{\mathsf{W_0}})$; and
   (b) generates $\mathsf{X_0}^* = (\mathsf{X_0}, \widehat{\mathsf{X_0}})$ where $\mathsf{X_0} = \mathsf{W_0}^*$ and $\widehat{\mathsf{X_0}}$ is a group signature which is linkable only to V^*:
   $\widehat{\mathsf{X_0}} = SignLinkable_G(gpk, gsk[i], \mathsf{W_0}^*, \widehat{\mathsf{V}}, \alpha, \beta)$, and sends it;
6. $\mathcal{U}_2$:
   (a) verifies the group signature: $Verify_G(gpk, \mathsf{X_0}, \widehat{\mathsf{X_0}})$ and
   (b) verifies that the two signatures have been performed by the same user: $VerifyLinkable_G(\mathsf{T.}\widehat{\mathsf{V}}, \widehat{\mathsf{X_0}})$. $\mathsf{X_0}^*$ works as a transfer agreement of the ticket of the user $\mathcal{U}_1$ to the user $\mathcal{U}_2$.

Ticket Transfer (Following '*k*' Times). In this protocol, $\mathcal{U}_1$ transfers the (already transferred in the past) ticket $\mathsf{X_{k-1}}$ to $\mathcal{U}_2$ by giving the permission to use it with a group signature which is linked to the commitment of the ticket previously received.

1. $\mathcal{U}_1$:
   (a) generates a commitment $m_{\beta_k} = ZKP_G Commit(\mathsf{X_{k-1}})$; and
   (b) sends the commitment and ticket $(m_{\beta_k}, \mathsf{X_{k-1}})$;
2. $\mathcal{U}_2$:
   (a) verifies the information and signature of T^*: $Verify_{\mathcal{I}}(\mathsf{T}^*)$, the group signature: $Verify_G(gpk, \mathsf{V}, \widehat{\mathsf{V}})$, and the linkability of the two group signatures of the beginning of the first transfer:
   $VerifyLinkable_G(\widehat{\mathsf{X_0}}, \mathsf{T.}\widehat{\mathsf{V}})$;
   (b) for each transfer $\forall i \in [0, k)$, verifies the group signatures
   $Verify_G(gpk, \mathsf{X_i}, \widehat{\mathsf{X_i}})$ and $Verify_G(gpk, \mathsf{W_i}, \widehat{\mathsf{W_i}})$, and checks the linkability of $VerifyLinkable_G(\widehat{\mathsf{X_i}}, \widehat{\mathsf{W_{i-1}}})$ if $i > 0$;

   (c) generates a random value $n_{\lambda_k} \xleftarrow{R} \mathbb{Z}_p$; and
   (d) generates the first *challenge*: $c_{\beta_k} = H(n_{\lambda_k}, k, price, \mathsf{flag\_transfer})$ (with the price agreed for the transfer) and sends it;
3. $\mathcal{U}_1$:
   (a) generates the response $s_{\beta_k} = ZKP_G Response(m_{\beta_k}, c_{\beta_k})$;
   (b) generates a random value $n_{\beta_k} \xleftarrow{R} \mathbb{Z}_p$; and
   (c) sends $(s_{\beta_0}, n_{\beta_0})$;
4. $\mathcal{U}_2$:
   (a) verifies the response: $ZKP_G Verify(m_{\beta_k}, c_{\beta_k}, s_{\beta_k})$; and
   (b) generates $\mathsf{W_k}^* = (\mathsf{W_k}, \widehat{\mathsf{W_k}})$ where $\mathsf{W_k} = (n_{\beta_k}, \mathsf{X_{k-1}}, \mathsf{flag\_transfer})$ and its group signature is $\widehat{\mathsf{W_k}} = Sign_G(gpk, gsk[i], \mathsf{W_k})$ and sends it;
5. $\mathcal{U}_1$:
   (a) verifies the group signature: $Verify_G(gpk, \mathsf{W_k}, \widehat{\mathsf{W_k}})$; and
   (b) generates $\mathsf{X_k}^* = (\mathsf{X_k}, \widehat{\mathsf{X_k}})$ where $\mathsf{X_k} = \mathsf{W_k}^*$ and $\widehat{\mathsf{X_k}}$ is a group signature which is linkable only to $\mathsf{W_{k-1}}^*$:
   $\widehat{\mathsf{X_k}} = SignLinkable_G(gpk, gsk[i], \mathsf{W_k}^*, \widehat{\mathsf{W_{k-1}}}, \alpha, \beta)$, and sends it;
6. $\mathcal{U}_2$:
   (a) verifies the group signature: $Verify_G(gpk, \mathsf{X_k}, \widehat{\mathsf{X_k}})$; and
   (b) verifies that the two signatures have been performed by the same user: $VerifyLinkable_G(\widehat{\mathsf{X_k}}, \widehat{\mathsf{W_{k-1}}})$. $\mathsf{X_k}^*$ works as a transfer agreement of the ticket of the user $\mathcal{U}_1$ to the user $\mathcal{U}_2$.

Ticket Verification (Standard). This protocol is used when no transfer has been performed since its issue. Here, $\mathcal{U}$ shows the ticket to $\mathcal{P}$ in order to be verified to receive the associated service. It works as follows:

1. $\mathcal{U}$ sends the ticket T^* to $\mathcal{P}$;
2. $\mathcal{P}$:
   (a) verifies the information and signature of T^*; and
   (b) generates a random value $n_\gamma \xleftarrow{R} \mathbb{Z}_p$ and sends it back;
3. $\mathcal{U}$ generates $\mathsf{Y}^* = (\mathsf{Y}, \widehat{\mathsf{Y}})$ where $\mathsf{Y} = (n_\gamma, \mathsf{T.Sn}, \mathsf{flag\_spend\_standard})$ with a group signature which is linkable only to V^*:
   $\widehat{\mathsf{Y}} = SignLinkable_G(gpk, gsk[i], \mathsf{Y}, \widehat{\mathsf{V}}, \alpha, \beta)$;
4. $\mathcal{P}$:
   (a) verifies the group signature: $Verify_G(gpk, \mathsf{Y}, \widehat{\mathsf{Y}})$ and that the two signatures are generated by the same user:
   $VerifyLinkable_G(\mathsf{T}.\widehat{\mathsf{V}}, \widehat{\mathsf{Y}})$; and
   (b) store $\mathsf{T.Sn}$ in $\mathcal{P}$'s centralized database.

Ticket Verification (Transferred). This protocol is used when some transfer has been performed since its issue. Here, $\mathcal{U}$ shows the ticket to $\mathcal{P}$ in order to be verified and receive the associated service. It works as follows:

1. $\mathcal{U}$ sends the transferred ticket $\mathsf{X_k}^*$ to $\mathcal{P}$;
2. $\mathcal{P}$:
   (a) verifies the information and signature of $\mathsf{X_k}^*$: $Verify_{\mathcal{I}}(\mathsf{T}^*)$ and $Verify_G(gpk, \mathsf{X_k}, \widehat{\mathsf{X_k}})$. The service provider $\mathcal{P}$ can detect if the ticket has been transferred or not depending on its content;
   (b) verifies that the two signatures which are included into the ticket have been generated by the same user: $VerifyLinkable_G(\widehat{\mathsf{X_0}}, \mathsf{T}.\widehat{\mathsf{V}})$.
   (c) for all the transfers, $\forall i \in [0, k]$: verify all the group signatures $Verify_G(gpk, \mathsf{X_i}, \widehat{\mathsf{X_i}})$ and $Verify_G(gpk, \mathsf{W_i}, \widehat{\mathsf{W_i}})$, and also the linkability of $\mathsf{W_{i-1}}^*$: $VerifyLinkable_G(\widehat{\mathsf{X_i}}, \widehat{\mathsf{W_{i-1}}})$ where needed; and
   (d) generates a random $n_\gamma \xleftarrow{R} \mathbb{Z}_p$ and sends it back;
3. $\mathcal{U}$ generates $\mathsf{Y}^* = (\mathsf{Y}, \widehat{\mathsf{Y}})$ where $\mathsf{Y} = (n_\gamma, \mathsf{T.Sn}, \mathsf{flag\_spend\_transferred})$ and its linkable group signature to $\mathsf{W_k}^*$ as follows:
   $\widehat{\mathsf{Y}} = SignLinkable_G(gpk, gsk[i], \mathsf{Y}, \widehat{\mathsf{W_k}}, \alpha, \beta)$;
4. $\mathcal{P}$:
   (a) verifies the group signature: $Verify_G(gpk, \mathsf{Y}, \widehat{\mathsf{Y}})$ and that the two signatures are generated by the same user:
   $VerifyLinkable_G(\widehat{\mathsf{W_k}}, \widehat{\mathsf{Y}})$ where the ticket receiver has then to demonstrate that is the same user both in the transfer and verification phases; and
   (b) store $\mathsf{T.Sn}$ in $\mathcal{P}$'s centralized database.

Revocation of Anonymity. To spend an e-ticket, any user has to do a new signature at the 3rd step of the ticket verification phase. In case of controversy (such as a e-ticket overspending case), the group manager $\mathcal{M}_G$ could take part in the resolution of the controversy and revoke the anonymity of the signer that misbehaved by calling the $Open_G$ procedure.

4 Security and Transferability of the System

In this section we discuss the security properties of our protocol. The discussion is organized in four propositions that state the security features of the scheme. Then the respective claims discuss and provide evidence to support propositions' arguments. This discussion does not provide any demonstration of the security of the cryptographic primitives and does not pretend to be a formal analysis of the security of the protocol, but it substantiates the security properties of the protocol. The common security properties of authenticity, non-repudiation and integrity, which are based on the security of the signature scheme used by $\mathcal{I}$, are attested in the Proposition 1, then revocable anonymity in Proposition 2, non-overspending in Proposition 3 and, finally, Proposition 4 is devoted to the requirement of transferability.

Proposition 1. *The proposed e-ticketing system preserves authenticity, non-repudiation and integrity of the e-ticket.*

Claim 1. It is computationally infeasible to make a new fraudulent e-ticket.

Security Argument. A valid e-ticket has the form $\mathsf{T}^* = (\mathsf{T}, \mathsf{Sign}_{\mathcal{I}}(\mathsf{T}))$. Then, the first step that $\mathcal{P}$ does when an e-ticket is received is the verification of the signature. The Ticket Verification protocol will continue only if this verification is positive; otherwise, $\mathcal{P}$ refuses $\mathcal{U}$'s request. Thus, making a new fraudulent valid e-ticket would be equivalent to breaking the signature scheme, which would be computationally infeasible as we have supposed that $\mathcal{I}$ uses a secure signature scheme.

Claim 2. The issuer $\mathcal{I}$ can not deny the emission of a valid e-ticket.

Security Argument. A valid e-ticket has $\mathcal{I}$'s signature and the signature scheme used is secure. Consequently, the identity of the issuer is associated to the ticket i.e. the signature is a non-repudiation evidence of origin.

Claim 3. The content of the e-ticket cannot be modified.

Security Argument. Suppose that someone modifies the content of the ticket, then a new $\mathcal{I}$'s signature has to be generated over the modified content; otherwise, the e-ticket will not pass the verification i.e. it will not be valid. Again, if it is computationally infeasible to forge $\mathcal{I}$'s signature, it is infeasible to modify the content of the e-ticket.

Proposition 2. *The e-ticketing system described in Sect. 3 is anonymous. The offered service is revocable anonymous.*

Claim 4. The protocol to get an e-ticket is anonymous.

Security Argument. The user establishes a connection with the ticket issuer $\mathcal{I}$ in order to receive the e-ticket. This connection could be established through an anonymous channel like TOR [4], guaranteeing then the user's privacy. There are current contributions[4] that have implemented TOR for mobile devices with Android. Additionally, the user does not use any personal authentication method to get the e-ticket. $\mathcal{I}$ generates and sends the e-ticket to $\mathcal{U}$ if she accredits to be member of the group of users by producing a valid group signature over a challenge sent by $\mathcal{I}$. The $Verify_G(\mathsf{V})$ procedure performed by $\mathcal{I}$ cannot identify the user.

Claim 5. An e-ticket has revocable anonymity.

Security Argument. A valid e-ticket does not have any information related to the user's identity. The e-ticket is generated by $\mathcal{I}$ who does not know the user's identity as we have discussed in the previous claim. The only item inside the e-ticket that can identify the user is the group signature V and only the group manager $\mathcal{M}_{\mathcal{G}}$ can reveal this information by performing $Open_G(gpk, gmsk, M, \sigma)$, because it is the only entity that knows $gmsk$. Therefore, e-tickets can be spent anonymously but the anonymity can be revoked by $\mathcal{M}_{\mathcal{G}}$. $\mathcal{M}_{\mathcal{G}}$ plays the role of a trusted third party, thus it will only do that by law enforcement.

[4] http://sourceforge.net/apps/trac/silvertunnel/wiki/TorJavaOverview

Claim 6. In spite of the anonymity of the e-ticket, a fake user cannot spend an e-ticket impersonating another user.

Security Argument. In order to spend the e-ticket, the legitimate $\mathcal{U}$ has to prove the ownership of the e-ticket by means of a linkable signature Y with the element V, placed inside the e-ticket. An illegitimate $\mathcal{U}$ cannot perform properly the $SignLinkable_G$ operation: the fraud will be detected because the $VerifyLinkable_G()$ operation performed at the ticket verification protocol will warn $\mathcal{P}$ about this impersonation attack.

Proposition 3. *The protocol controls overspending.*

Claim 7. If the ticket is only validated by one $\mathcal{P}$, the verification can be offline and $\mathcal{P}$ can control any overspending attempt.

Security Argument. $\mathcal{P}$ maintains a database with the serial numbers of the e-tickets already validated (i.e used). $\mathcal{P}$ can check both the issuer's signature and whether the e-ticket has not been spent before by using the information stored in the database. So the provider does not need to contact any party during the validation of an e-ticket.

Claim 8. If the ticket is validated with several providers, all $\mathcal{P}$'s must then be connected and share a database of spent tickets.

Security Argument. The set of providers maintains a shared database with the serial numbers of the e-tickets that have been already validated. The contents of this database are used by the providers to decide if they accept and validate a new ticket. So the provider does not need a connection with the issuer during the verification of an e-ticket, but the set of providers must share a database instead, so that the overspending can then be detected and the identity of the overspender can be revealed by the group manager through the $Open_G(gpk, gmsk, M, \sigma)$ procedure.

Proposition 4. *Users can transfer their e-ticket to other users making use of the proposed scheme. The transferability operation among users preserves the security properties no matter how many transfers of the e-ticket have been made.*

Claim 9. A transferred e-ticket can guarantee authenticity, non-repudiation and integrity properties as a non-transferred e-ticket.

Security Argument. During a transfer, the format of the e-ticket is not substantially altered. Only a new group signature of the new owner is added to the e-ticket so that the properties of the signature keep authenticity, non-repudiation and integrity of the transferred e-ticket since the discussion of the Proposition 1 is already valid.

Claim 10. A transferred e-ticket preserves the anonymity of its owner.

Security Argument. During a transfer, the owner of the e-ticket proves its ownership with a ZKP operation in order not to disclose her identity. The receiver of the transferred e-ticket includes a new group signature in it, so her identity is similarly protected in the same way, as we see in the step 4 of the ticket verification protocol (transferred version).

Claim 11. A transferred ticket cannot be overspent.

Security Argument. The overspending detection procedure described in Proposition 3 is also valid for transferred e-tickets as well as the anonymity revocation which can be made using $Open_G(gpk, gmsk, M, \sigma)$ because this property relies on the verification made by $\mathcal{P}$ on the serial number of the e-ticket stored in the database. The transfer of any e-ticket does not change its serial number.

5 Conclusions

We have presented a proposal for an electronic ticketing system which guarantees the anonymity for their users and also allows the transferability of the tickets between them through payment or loan.

The proposed scheme is anonymous, as in the ticket issue protocol, a group signature scheme has been used, which allows the issuer to verify that the user belongs to a valid group of users, yet cannot identify which one she is. If the user tries to commit fraud, the group manager can revoke her anonymity.

Moreover, the protocol introduces the requirement of ticket transferability between two users. This property aims to increase the system flexibility since users can share their tickets with friends or they can give them to other users. To do that, we use a linkable group signature scheme. With this technique, group signatures from the users involved in the transfer operation are used in order to generate a ticket transfer agreement, which could be further used as an evidence proof in case of any conflict between the parties. As future work, the main goal will be to develop and evaluate the performance of the protocol in a mobile platform, in order to check its feasibility.

Disclaimer and Acknowledgements. This work was partially supported by the Spanish Ministry of Science and Innovation [eAEGIS TSI2007-65406-C03-01, ARES--CONSOLIDER INGENIO 2010 CSD2007-00004, Audit Transparency Voting Process IPT-430000-2010-31, CO-PRIVACY TIN2011-27076-C03-01, ICWT TIN2012-32757, BallotNext IPT-2012-0603-430000]; and the Government of Catalonia [2009 SGR1135]. The authors are solely responsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organization.

References

1. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
2. Canard, S., Gouget, A.: Anonymity in transferable e-cash. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 207–223. Springer, Heidelberg (2008)

3. Chen, Y.-Y., Chen, C.-L., Jan, J.-K.: A mobile ticket system based on personal trusted device. Wirel. Pers. Commun. Int. J. **40**(4), 569–578 (2007)
4. Dingledine, R., Mathewson, N., Syverson, P.: Tor: the second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium (2004)
5. Ghiron, S., Sposato, S., Medaglia, C., Moroni, A.: NFC ticketing: a prototype and usability test of an NFC-based virtual ticketing application. In: Workshop on Near Field Communication 2009, NFC '09, pp. 45–50. IEEE, February 2009
6. Heydt-Benjamin, T.S., Chae, H.-J., Defend, B., Fu, K.: Privacy for public transportation. In: Danezis, Ge, Golle, P. (eds.) PET 2006. LNCS, vol. 4258, pp. 1–19. Springer, Heidelberg (2006)
7. Isern-Deyà, A.P., Vives-Guasch, A., Mut-Puigserver, M., Payeras-Capellà, M., Castellà-Roca, J.: A secure automatic fare collection system for time-based or distance-based services with revocable anonymity for users. Comput. J. **56**, 1198–1215 (2012)
8. Jorns, O., Jung, O., Quirchmayr, G.: A privacy enhancing service architecture for ticket-based mobile applications. In: Availability, Reliability and Security, Vienna, Austria, pp. 374–383, ARES 2007 - The International Dependability Conference, vol. 24, April 2007
9. Mut-Puigserver, M., Payeras-Capellà, M.M., Ferrer-Gomila, J.-L., Vives-Guasch, A., Castellà-Roca, J.: A survey of electronic ticketing applied to transport. Comput. Secur. **31**(8), 925–939 (2012)
10. Quercia, D., Hailes, S.: Motet: mobile transactions using electronic tickets. In: Proceedings of the Security and Privacy for Emerging Areas in Communications Networks, vol. 24, pp. 374–383, Greece, Sept. 2005
11. Sunitha, N., Amberker, B., Koulgi, P.: Transferable e-cheques: an application of forward-secure serial multi-signatures. In: Ao, S.-I., Rieger, B., Chen, S.-S. (eds.) Advances in Computational Algorithms and Data Analysis. Lecture Notes in Electrical Engineering, vol. 14, pp. 147–157. Springer, Netherlands (2009)
12. Vives-Guasch, A., Castellà-Roca, J., Payeras-Capella, M., Mut, M.: An electronic and secure automatic fare collection system with revocable anonymity for users. In: Advances in Mobile Computing & Multimedia (MoMM) (2010)
13. Vives-Guasch, A., Payeras-Capellà, M.M., Mut-Puigserver, M., Castellà-Roca, J., Ferrer-Gomila, J.L.: A secure e-ticketing scheme for mobile devices with near field communication (NFC) that includes exculpability and reusability. IEICE **E95–D**(1), 78–93 (2012)