MOTET: Mobile Transactions using Electronic Tickets

Daniele Quercia and Stephen Hailes
Department of Computer Science, University College London, London, WC1E 6BT, UK.
{D.Quercia, S.Hailes}@cs.ucl.ac.uk

Abstract

There has been considerable work within the field of digital cash protocols that aims to provide security guarantees - non-repudiation, authentication, overspending checking and off-line checking - whilst protecting anonymity. However, considerably less attention has been given to the question of electronic ticketing, and what exists has been rather abstract or limited. Although eTickets aim at providing the same security guarantees and privacy preservation properties as digital cash, they are significantly different. Digital cash derives much of its anonymity from the fact that the denominations of electronic coins and notes are sufficiently universal that it is not possible for the bank to know in advance how they might be spent. In an eTicketing system, however, this is not the case: at the point the ticket is purchased, the ticket vendor knows for what it will be used and, if a non-anonymous payment system is used, can associate this with the customer. We present a novel protocol that enables users to purchase and spend electronic tickets (eTickets) of a range of two different types: those that can only be used a certain number of times, and those that expire after a certain date.

1 Introduction

Notwithstanding the false start of the early digital cash deployments, payment mechanisms are a key enabling technology for mobile commerce. We choose to explore a heretofore rather neglected area of electronic payment mechanisms: that of electronic ticketing. Given the widespread adoption of a number of standard short-range networking technologies such as 802.11, Bluetooth, and Zigbee, we believe that to base universal eTicketing solutions on specific hardware platforms would be overly restrictive. Instead, agreement at the level of protocols for the purchase and exchange of eTickets is more scalable and more likely to open up a wide range of different business opportunities. Implicitly, then, there are two preconditions to achieving this: the definition of a standard form of a ticket, which is in progress [1, 2], and the definition of standard protocols, which this paper addresses.

Although digital cash and eTicketing aim at providing the same privacy and security properties, they are different in two main respects. First, current digital cash solutions cannot be directly integrated into an eTicketing framework. Customer identities are blindly embedded into eTickets so that overspenders can be identified, while honest customers' anonymity is preserved. This takes place when eTickets are bought. Therefore, anonymous electronic payment methods (e.g., digital cash) cannot be used for buying eTickets and, thus, an ad-hoc eTicket issue phase is required. Second, an entity issuing digital coins (e.g., a bank) does not know in advance how its coins might be spent, whereas a ticket agent knows for what its eTickets will be used. Therefore, spenders are more likely to be identified when using eTickets than when paying with digital cash.

The small body of work that has been published in the field of electronic ticketing fails to be fully satisfactory because they are very limited in vision. Thus they tend to assume a simple on-line verification model that does not realise the full potential of ticketing as defined by ticket standardisation activities, and instead address rather simple cases such as pay-tv systems and airline ticketing.

We propose a general-purpose electronic ticketing mechanism. Customers buy eTickets from ticket agents and spend them at service providers. The key features supported include: (i) preservation of customer anonymity, so long as they do not seek to overspend; (ii) non-repudiation for the customer and the service provider; (iii) off-line eTicket checking at any service provider without contacting the ticket agent; (iv) eTicket use at a range of the possible service providers; (v) support for a range of different ticketing models.

2 Related Work

Electronic payment systems have formed a subject of study in the academic community since work in the early 1980s by Chaum [8]. However, given their direct applicability to the real world, there has been more interest in the

commercial development of this academic research than is usually the case. Electronic payment systems have passed through two commercial generations: the initial enthusiasm by technologists for general-purpose electronic cash schemes ended largely in failure just before the millennium (the most notable being Chaum's DigiCash). However, even in advance of this failure, a second generation of schemes started to emerge, including PayPal, PayWeb, Millicent and many others. Although there continues to be market movement (e.g., the acquisition of CyberCash by Verisign, proposals to use Oystercard as a digital cash system, etc.) and failure (e.g., of beenz), many of the approaches would appear to be surviving, largely because there is clearer thinking about the marketplace in which they operate. The underlying general B2B marketplace remains slow in converting to electronic payment methods and, consequently, one might expect the more general B2C marketplace to remain suppressed for the foreseeable future.

At the same time as the second generation of e-cash schemes started to emerge, consideration was starting to be given to digital ticketing, with the publication by Fujimura and Nakajima [11] of requirements for a general-purpose eTicket, a further development of which was in the form of a generalized (XML-based) digital-ticket definition language [12]. More recently, this has led to an informational RFC that identifies requirements for a voucher trading system [10] to allow the interchange of digital coupons, gift certificates, and loyalty points. Although foundational, all of these approaches are high-level; they lack detailed consideration of how best to implement the protocols for ticket exchange.

Of the relatively small body of academic work published in this field, Song and Korba [17] propose an eTicket protocol for Pay-TV systems that provides both customer privacy protection and non-repudiation features. Initially, the customer purchases an eTicket from a provider. When she later wishes to subscribe to a set of TV channels, she presents an eTicket. It is not possible to determine to which channels the customer has subscribed; moreover, a trusted third party can provide a repudiation service in case of dispute. However, there are three problems with this approach. Firstly, dealing with disputes implies the involvement of a Trusted Third Party, which is not always desirable. Secondly, eTickets are not globally spendable, but can be spent only at particular predefined service providers. Finally, this protocol does not provide transferability: an eTicket is bound to the buyer's identity.

Bao *et al* [5] investigated eTicketing systems providing a distributed verification system for mobile devices. However, the description is again rather high-level and is limited to a relatively straightforward form of distributed verification. Nevertheless, it is worth noting that there are commercial activities in this space: for example, Vodafone in New Zealand are utilising SMS-based technology from mTicket to allow the purchase of tickets for a range of events online [4] . A range of suggestions for transport systems exist: Premasathian and Thainimit [13] propose a conceptual framework for highway toll collection that uses a similar model to the congestion charging mechanism in London. Likewise, trials using mobile 'phones as carriers of bus tickets are being undertaken by Nokia, Philips, and RMV, the public transport authority for Frankfurt [3]. Other studies on eTicket validation exist, e.g., [15] and [14], and there is an attempt at standardisation by Mobile Electronic Transactions Ltd. [1, 2] but there is little that even attempts to explore the generality of the vision of eTicketing presented by Fujimura [11] at a technical level.

The remainder of this paper will be focus on elucidating the technical details of a general-purpose electronic ticketing mechanism. In section 3 we lay the structural foundations for our mechanism: the system architecture and an attacker model are discussed. Section 4 discusses the protocols of our ticketing mechanism. In section 5 we assess the compliance of our approach against general eTicketing requirements. To complete the discussion, we describe future work in Section 6. Section 7 concludes the paper.

3 System architecture

3.1 Introduction

In general, eTicketing systems are a generalisation of digital cash, and share many properties in common with it. In this section, we give a brief overview of the key architectural elements that are considered within our system.

Tickets, so far as we are concerned, come in three forms: (i) they can be credit constrained (e.g., a phonecard); (ii) they can be usage constrained - restricted to a particular number of uses (e.g., a cinema ticket or the equivalent of a book of tickets); or (iii) they can be time-constrained - restricted by the expiry date on the ticket (e.g., a bus pass). In the real world, where they are represented by some identifiable physical token, tickets may further be classified as fully anonymous(e.g., a train ticket bought with cash) or those that are sold to a particular named individual (e.g., an airline ticket). However, in the electronic world, the representation of a ticket as a purely digital token means that it is simple to copy tickets; consequently, as for digital cash, we must address the problems of ensuring uniqueness and the problem of double-spending without compromising the anonymity of the individual unless and until they attempt to misbehave. In the latter case, any anonymity should be *revokable*.

The actors in our system fall into three classes: Tickets are sold by a *ticket agent* to a *customer*. The customer then spends the tickets with a *service provider* (sometimes

known as a verifier or ticket inspector), who verifies their validity before granting service. The ticket agent and the service provider can be the same entity, but they need not be, hence the logical separation. Regardless, the service provider must share a relationship (possibly indirectly) with the ticket agent issuing the tickets, and there will need to be an exchange of money between these entities in order to ensure that the agent has tickets to sell.

Four additional properties of tickets and ticketing protocols are important: (i) in any ticket use, *non-repudiation* is important to the customer and the service provider - neither should be able to claim that the use did not occur, without the other party being able plausibly to deny that claim; (ii) customers should be able to spend their eTickets at any appropriate service provider (i.e., eTickets should be *globally-spendable*); (iii) eTicket protocols should support *off-line checking*: eTickets can be spent and checked at any service provider without contacting the ticket agent. Finally, customers should, in some circumstances, be able to give up sell their eTickets, making them *transferable*.

For the purposes of this paper, we assume that communication takes place over an unsecured channel and that compromise of private keys is dealt with by mechanisms outside our consideration. Also, we are not concerned with the tamper resistance of devices.

3.2 Attacker model

In order to develop a protocol for eTicket exchange, it is necessary to understand what the different actors may have to gain by executing different types of attack on the system. Below, we give a classification of possible attacks categorised by attacking entity.

1. **Fraudulent Customer**

   Overspending : A customer breaches the spending constraints in the ticket: they attempt to exceed the credit limit, use it more than the number of allowable times, or for greater than the allowed length of time.

   Theft : A customer illegally obtains eTickets and attempts to spend them.

2. **Fraudulent Service Provider**

   Over-deposit : A service provider redeems a single spending eTicket at the ticket agent several times.

   Customer indentification : A service provider identifies a customer or links her spending behaviour across multiple tickets using information from spent eTickets, thus breaching customer privacy.

   Framing : A service provider gives an indication to the ticket agent that an eTicket has been overspent. Subsequently, an honest customer is incriminated and his/her identity is revealed.

3. **Fraudulent Ticket Agent**

   Customer indentification : The ticket agent identifies a customer or links their spending bevahiour across multiple tickets using information from deposited eTickets.

   Customer false accusation : The ticket agent unjustly accuses a customer of overspending eTickets or using stolen eTickets.

   Service Provider false accusation :The ticket agent unjustly accuses a service provider of overdepositing.

4. **Fraudulent Ticket Agent and Fraudulent Service Provider**

   Collusion :The ticket agent and a service provider collude and frame or identify the customer.

4 Protocol actions

4.1 Terminology and Notation

The basic terminology and notation used in this paper are defined as follows.

Symbol	Description
TA	Ticket Agent
C	Customer
SP_x	Service Provider x
$KR_i = (d_i, p_i, q_i)$	entity i's private key
$KU_i = (e_i, n_i)$	entity i's public key
$Timestamp_i$	Time stamp generated by entity i
ID_i	Entity i's ID
H	one-way hash function
f	one-way hash function
g	one-way hash function
$Signature_i$	signature using KR_i

4.2 Phases

Our eTicketing scheme is divided into four main phases:

1. eTicket wholesale phase, in which the ticket agent purchases tickets from the service provider. This phase is optional, and depends on the cost recovery model for the service provider. An alternative would be to allow authorised ticket agents to sell as many tickets as possible, and for the service provider to recover costs from

the agent when those tickets are used, or to pay the ticket agent a bonus per use. The difference, in effect, is whether the service provider is a subcontractor of the ticket agent, and is paid by use, or whether the service provider subcontracts the sale of tickets to the ticket agent, who purchases tickets wholesale in this stage. Additionally, the ticket agent may receive a bonus per use.

2. eTicket issue phase, in which a customer contacts the ticket agent, requests a ticket and pays for it.

3. eTicket spending phase, in which a customer takes a ticket and spends it with a service provider.

4. eTicket cashing phase, in which the transfer of funds between the ticket agent and the service provider takes place according to the cost recovery model chosen. See the eTicket wholesale phase for details.

The eTicket wholesale phase is outside the scope of this paper and will form future work.

In the protocols described below, we assume that tickets are to be issued under conditions of revokable anonymity. To extend the protocol to deal with the other cases of full anonymity and full identification is straightforward.

4.3 eTicket issue phase

The purpose of this phase is to arrive at a point where a customer has a ticket in their possession, the ticket agent has been paid the correct amount for this, the parameters within the ticket are those agreed by the customer and the agent, and it is not possible to link the ticket purchased to the customer. Note that this is slightly different from digital cash systems in which unlinkability is a feature of the universality of money; in this case, tickets identify *ex ante* the ways in which they will be spent. Thus to achieve our aims we employ two subphases: the initial *bootstrap subphase* takes place under conditions in which the ticket agent knows with whom they are interacting and exists simply to negotiate parameters for the second *anonymous eTicket issuing subphase* in which the ticket in question is purchased. We will address them in turn.

4.3.1 Bootstrap subphase

In this subphase, the customer will pay for the ticket they wish to purchase. However, the identity of the customer with whom we are interacting may well be visible through, for example, the payment method they elect to use e.g., if they use a credit card. Consequently, the ticket cannot be issued in this phase, and we must instead rely on the use of blind signature protocols for the ticket agent to commit to a set of parameters that will later be used anonymously in the actual generation of the ticket. There are three parameters that require such treatment:

- In later interactions, both during the anonymous eTicket issue subphase and the eTicket spending phase, it will prove to be necessary to authenticate the customer as the legitimate owner of information used to build the ticket or of the ticket itself. In order to do this, a zero knowledge proof based on the use of a public/private keypair is employed; however, in order to ensure unlinkability, it is necessary to ensure that the customer's identity cannot be associated with the keypair to be used. Consequently, we employ temporary public keys that are authenticated by the ticket agent using a blind signature protocol.

- If the customer elects to pay for the ticket using a non-anonymous method of payment, then it is clearly necessary for the payment to occur here, and for an authenticated (but anonymous) receipt to be created that can be presented to the eTicket issue subphase.

- Finally, to ensure revokable anonymity, the customer identity, suitably protected, must be incorporated into the ticket itself. Since it is important that the ticket agent be confident that the correct identity is actually being embedded at the point of issue, but since it cannot, for reasons of unlinkability, be allowed to see the identity at that point in time, we must also negotiate a suitably blinded set of identity parameters here in such a way that they can be presented during the ticket issue subphase.

The bootstrap subphase involves the customer and the ticket agent and consists of the following steps:

1(a) Following the selection of the appropriate product to be bought, the customer prepares and securely sends: (i) a blinded element embedding both the eTicket cost and the temporary public key; (ii) a number of pieces of customer personal information, will be split and kept secret with a pair of blinding factors.

The customer creates a temporary public key (e_T, n_T) and a temporary private key (d_T, p_T, q_T). The public key is then blinded. We elect, for reasons of simplicity and comparability, to illustrate our technique using the blind signature technique shown in [8], in which the customer selects a random number r that blinds the authentication of the temporary public key (e_T, n_T): $(\textit{blinded\_TPK}) = r^{e_{TA}} H(e_T||n_T) \pmod{n_{TA}}$. However, in practice, alternative blinding techniques based on blind Schnorr signatures (or their variants) may well be more efficient.

To facilitate the revocation of anonymity in cases of attempted overspending, it is necessary to incorporate customer identity information into the ticket. However, as in

digital cash protocols, this information must be blinded in such a way that the identity information is not revealed under normal usage, but carries a very high probability of being revealed under conditions of misuse. There are essentially two ways to do this - the cut-and-choose protocol [9], which is simple if inefficient, and the restrictive blind signatures developed by Brands [7] and Scoenmakers [16], which are considerably more efficient. Purely for reasons of simplicity we elect to describe our protocol using the former.

As described in [9], we require that the customer takes their identity string and splits it into two parts in such a way that revealing one part reveals no Shannon information about the customer's identity, with the additional property that it is possible to verify when a part has been correctly revealed. A customer provides k independent such pairs to the ticket agent, who selects $\frac{k}{2}$ of them, and requires Alice to reveal both halves in order to verify that she is not seeking to mislead with respect to her identity. In our case, we require for each pair the customer supplied in the original digital cash scheme, she now supplies an *identity vector* consisting of n independent pairs, where n is the number of permissible uses of a usage-constrained ticket; moreover, where a pair was originally revealed in the cut-and-choose protocol, the corresponding identity vector must now be revealed in its entirety. Thus, in total, the customer will supply $n \cdot k$ independent identity pairs, of which $n \cdot \frac{k}{2}$ will be revealed. It is possible for the customer to supply a greater number of identity vectors than she expects to use, in order to obfuscate the nature of the ticket she intends to buy; it would even be possible to conduct this part of the protocol in advance, maintaining a cache of signed identity vectors for use when needed.

Blinded customer information consists of the ordered set of k identity vectors, each of which has n entries:

$$< \textit{blind\_C\_info} >= \{(\textit{blind\_C\_info})_{ij}\} = \\ = \{blind_{ij}^{e_{TA}} f(x_{ij}, y_{ij})\} \pmod{n_{TA}}$$

$\forall i \in [1,k]$, $\forall j \in [1,n]$ where $x_{ij} = g(\textit{id\_reveal}_{ij}$ XOR $(\textit{C\_info})$ XOR $j)$, $y_{ij} = g(\textit{id\_reveal}_{ij})$, f and g are publicly known one-way hash functions, and $blind_{ij}$ and $\textit{id\_reveal}_{ij}$ are randomly chosen.

Finally, the customer encrypts (with the ticket agent's public key) the temporary public key element, (*blinded_TPK*), and the blinded customer identity information, *blind_C_info*, together with payment details and the cost of the eTicket. The signature attached to the message indicates a commitment on the part of the customer to complete the transaction at this price. The ticket agent may provide this information to the credit card company to authorise payment, or they may be capable of charging an arbitrary amount to the credit card. If the latter, the customer can later repudiate the transaction and, without a signed commitment to pay by the customer, the ticket agent will be seen to be guilty of attempted fraud.

$$C \rightarrow TA : KU_{TA}\{< \textit{blind\_C\_info} >, (\textit{blinded\_TPK}), \\ (\textit{payment}), (KU_C), (\textit{eTicket\_cost}), \textit{Timestamp}_C, \\ \textit{Signature}_C\}$$

1(b) At this point, the ticket agent has received a request to purchase a ticket. In order to issue a ticket in the next subphase when the identity of the customer is no longer visible, it needs to check the consistency of the identity information, take payment, and authenticate the blinded temporary keys and the remaining blinded identity information.

The ticket agent (i) challenges the customer to reveal a subset of the blinding factors; (ii) signs and returns the blinded eTicket cost and the blinded temporary public key which are, thus, certified by the ticket agent.

The ticket agent decrypts and verifies the integrity of the received message. This includes assessing whether it is sufficiently timely to be worthy of consideration, which it does by examining whether the timestamp is within an acceptable distance, T_Δ of the current time at the ticket agent. Since we do not assume synchronised clocks, we need to hold information about requests with timestamps in the past T_Δ seconds, effectively using a sliding window to eliminate replay. If the time on the request postdates the current time at the ticket agent, then either we must hold the request until that time has passed, or we must reject the request and hold a list of such rejections that is pruned at the time when the sliding window would have passed the rejected request. Note, however, that this process consumes resources and, consequently, renders the ticket agent subject to a denial of service attack. This threat can be ameliorated by introducing an extra stage that requires the client to solve a puzzle before proceeding with the remainder of the process; however, the dimensioning of such puzzles is a difficult issue and either discriminates against resource poor clients or has little effect on DoS.

If the message is correct the ticket agent stores the customer's credit card information for later debit according to the eTicket cost. It then signs the (*blinded_TPK*) and *eTicket_cost* with its secret key d_{TA}: $(\textit{signed\_blinded\_TPK}) = (\textit{blinded\_TPK})^{d_{TA}} \pmod{n_{TA}}$ and $(\textit{signed\_eTicket\_cost}) = (\textit{eTicket\_cost})^{d_{TA}} \pmod{n_{TA}}$.

Next, the ticket agent creates a challenge for the customer. It chooses $\frac{k}{2}$ at random out of the received set of blinded customer personal identity vectors in order to verify that they are consistently the same. Note that there is no

need for this identity to be the same as the identity of the individual paying; for example, the tickets may be intended as a gift. However, there are undoubtedly circumstances in which one might care to draw negative inferences if the identities do not match, in the same way as it is sometimes reasonable to draw negative inferences if the intended delivery address is different to the billing address of a credit card in an online purchase. The ticket agent returns the selected indices $<$ *ID_verify_indexes* $>$ to the customer encrypted with the customer's public key.

$$\begin{aligned}&TA \rightarrow C : KU_C\{< ID\_verify\_indexes >,\\ &(signed\_blinded\_TPK), (signed\_eTicket\_cost),\\ &Timestamp_{TA}, Signature_{TA}\}\end{aligned}$$

1(c) At this point, the ticket agent has returned a message that contains a signed item to the effect that customer has agreed to purchase a particular eTicket at a given price and a blind signed temporary public key from the customer. It has also issues a set of challenges for identity vectors to which the customer must respond.

Firstly, the customer checks the signature on *signed_eTicket_cost*. Next, it removes the blinding factor from (*signed_blinded_TPK*), and checks the signature on that.

$$\begin{aligned}(signed\_TPK) &= \frac{((signed\_blinded\_TPK))}{r} \pmod{n_{TA}}\\ &= H^{d_{TA}}(e_T||n_T) \pmod{n_{TA}}\end{aligned}$$

Lastly, the customer encrypts with the ticket agent's public key the requested pairs of blinding factors $blind_{ij}$ and id_reveal_{ij} $\forall i \in <$ *ID_verify_indexes* $>$ and $\forall j \in [1, n]$ together with the customer personal information (*C_info*). It returns them to the ticket agent.

$$\begin{aligned}&C \rightarrow TA : KU_{TA}\{\{blind_{ij}, id\_reveal_{ij}\}_{\substack{i \in <ID\_verify\_indexes>\\ j \in [1,n]}}\\ &(C\_info), Timestamp_{C'}, Signature_{C'}\}\end{aligned}$$

1(d) After receiving the requested blinding factors, the ticket agent verifies that the customer embedded the correct personal information within the customer personal information vectors. If the customer did not cheat, the ticket agent debits the credit card according to the information stored it stored in step 4.3.1, then blindly signs and returns the remaining customer personal information vectors. This is, however, achieved in stages. Firstly: $\forall j \in [1, n]$:

$$\begin{aligned}(signed\_blinded\_id\_vector)_j &= \\ &= \prod_{i \notin <ID\_verify\_indexes>} (blind\_C\_info)_{ij}^{d_{TA}}\end{aligned}$$

Without loss of generality we assume that the blinded customer information array is reorder such that the verified indices run from $\frac{k}{2}+1$ to k, in other words, the first $\frac{k}{2}$ entries remain blinded.

At this point, it might seem reasonable to aggregate and encrypt the set of $(signed\_blinded\_id\_vector)_j$ with the customer's public key and to send them to the customer:

$$\begin{aligned}&TA \rightarrow C : KU_C\{(signed\_blinded\_id\_vector)_j,\\ &Timestamp_{TA'}, Signature_{TA'}\}\end{aligned}$$

However, unless an acknowledgement is forthcoming from the customer, they could argue that the ticket agent had debited their credit card but had never returned the blind-signed identity vectors. Since the next stage of the protocol is anonymous, the ticket agent would necessarily be unable to associate the spending of the ticket with this repudiation, so there would be nothing to prevent both from occurring. There are two approaches to solving this problem, depending on whether a trusted arbitrator is involved. In the first solution, the agent does indeed send this message either directly or through a mutually trusted arbitrator. If directly, it resends the blind-signed identity vectors the through the arbitrator in case of dispute. If the customer has already used the ticket, they gain no advantage from this. Alternatively, if no arbitrator is available, the parties could employ a simultaneous contract signing protocol [6]: the customer to obtain the signed identity vectors, and the ticket agent to obtain an acknowledgement of their correct receipt. This would take place in two phases:

$$\begin{aligned}&TA \rightarrow C : KU_C\{H((signed\_blinded\_id\_vector)_j),\\ &Timestamp_{TA'}, Signature_{TA'}\}\end{aligned}$$

This message means that the ticket agent commits to a hash of the blind-signed identity vectors, which it requires the customer to sign. Next, the ticket agent sends and commits to the actual values at the same time as the customer send and commits to the hash.

4.3.2 Anonymous eTicket issuing subphase

After the end of the initial subphase, the customer will possess signed keys, eTicket cost, and identity pairs in such a way that the ticket agent cannot link them to the customer's identity. As a result, if, from now on, the customer uses the anonymous temporary private key to authenticate themselves, neither the ticket agent nor any service provider can discover the customer's identity and

the customer can obtain an eTicket both securely and anonymously.

1(e) The customer removes the blinding factors from all $(\textit{signed\_blinded\_id\_vector})_j$, $\forall j \in [1, n]$ and forms them into an array with the j^{th} *signed_id_vector* forming the j^{th} row.

$$\begin{aligned} < \textit{signed\_id\_array} > &= \{(\textit{signed\_id\_vector})_j\} \\ &= \frac{(\textit{signed\_blinded\_id\_vector})_{ij}}{blind_{ij}} \end{aligned}$$

The i^{th} column of $<$ *signed_id_array* $>$ is used during the i^{th} use of the ticket. We denote this $<$ *signed_id_array* $>_l$.

The customer then requests the services he wishes to access, sending the authenticated eTicket cost token *signed_eTicket_cost*, for which they have previously paid, and the authenticated temporary public key, *signed_TPK*, and the authenticated identity vectors. Should a usage-constrained multi-use ticket be required, the customer generates a hash chain and commits to the anchor. A hash chain is formed from a series of related elements:$\textit{chain}_0, \ldots, \textit{chain}_n$, where

$$\begin{aligned} &\textit{chain}_n = H(\textit{random}); \\ &\textit{chain}_{l-1} = H(\textit{chain}_l) \quad \forall l \in [1, n]; \\ &\qquad\qquad\qquad\quad \textit{random}\text{: random number;} \end{aligned}$$

$\textit{chain}_0$ is known as the anchor of the chain and $\textit{chain}_n$ is known as its root. Any host that knows the root of the chain can recreate any element within it. If a principal knowing the root, and having committed to the anchor, has revealed $\textit{chain}_i$, then they can demonstrate authenticity for some action by revealing $\textit{chain}_{i+1}$. Given the nature of the one-way hash function, it is computationally infeasible to generate $\textit{chain}_{i+1}$ knowing only $\textit{chain}_i$ but trivial to check that $\textit{chain}_i$ came from $\textit{chain}_{i+1}$. Thus, someone with the ability to reveal $\textit{chain}_{i+1}$ is highly likely to be the same individual who revealed $\textit{chain}_i$, and so on transitively to $\textit{chain}_0$. We know that our principal committed to $\textit{chain}_0$, so it must still be our principal who just revealed $\textit{chain}_{i+1}$.

If the customer requires a usage-constrained multi-use ticket, they generate a hash chain of length matching that of the array of identity vectors $(\textit{signed\_id\_vector})_j$. They then sign the anchor: $(\textit{signed\_chain\_anchor}) = (\textit{chain}_0)^{d_T} \pmod{n_T}$.

Taken together, the customer prepares a list of services $<$ *services_rqst* $>$ he wishes to access and encrypts it together with the signed temporary public key and the signed eTicket cost and the signed anchor of the hash chain. He encrypts and sends the message to the ticket agent.

$$\begin{aligned} C \rightarrow \textit{TA} : \textit{KU}_{\textit{TA}}\{&< \textit{services\_rqst} >, (e_T, n_T), \\ &(\textit{signed\_eTicket\_cost}), (\textit{signed\_TPK}), (\textit{signed\_id\_array}), \\ &(\textit{signed\_chain\_anchor}), \textit{Timestamp}_{C''}, \textit{Signature}_T\} \end{aligned}$$

1(f) The ticket agent verifies whether its signatures on the eTicket cost, the temporary public key, and the identity vectors are correct and checks that *signed_eTicket_cost* is at least as much as the cost of the service requested. The ticket agent creates an eTicket that contains the previously authenticated fields along with information specific to the services to which the ticket relates; this might, for example, include an expiry date:

$$\begin{aligned} &< \textit{eTicket} >= \{(e_T, n_T), (\textit{signed\_eTicket\_cost}), \\ &(\textit{signed\_chain\_anchor}), < \textit{granted\_services\_info} >, \\ &(\textit{signed\_id\_array}), \textit{Timestamp}_{\textit{TA}''}, \textit{Signature}_{\textit{TA}''}\} \end{aligned}$$

The ticket agent encrypts the eTicket with the $temporary$ public key and sends them to the customer, which then has a valid ticket it can spend. Again, a simultaneous contract signing approach may be employed in obtaining a receipt for the ticket and so prevent later contestation over whether this message was or was not sent.

$$\textit{TA} \rightarrow C : \textit{KU}_T\{< \textit{eTicket} >\}$$

The ticket agent retains a copy of the eTicket.

4.4 eTicket spending phase

2(a) At this point, the customer possesses a signed eTicket. When he wishes to spend the eTicket, he encrypts it with the service provider's public key and sends it to the service provider. In addition, the customer also encrypts and sends: (i) the next non-spent spending chain element $\textit{chain}_l$ together with its position l in the chain; (ii) the identity of the ticket agent $ID_{\textit{TA}}$ that signed the information within the ticket.

$$\begin{aligned} C \rightarrow SP_x : \textit{KU}_{SP_x}\{&< \textit{eTicket} >, l, \textit{chain}_l, ID_{\textit{TA}}, \\ &\textit{Timestamp}_{C'''}, \textit{Signature}_{T''}\} \end{aligned}$$

2(b) The service provider needs to be sure that the ticket is legitimate, current and possesses sufficient credit to allow its use. Thus the service provider:(i) verifies the ticket agent's signature on $<$ *eTicket* $>$; (ii) verifies the correctness of $\textit{chain}_l$ using the committed $\textit{chain}_0$ in the eTicket; (iii) verifies whether the conditions of use of the $<$ *eTicket* $>$ have been met, e.g., whether it has been locally overspent.

To verify (*signed_id_array*), the service provider creates a random binary string $< S >$ of $\frac{k}{2}$ elements. This determines which of the two blinding factors, for each element in the l^{th} column of the array the customer should send. The service provider encrypts the challenge $< S >_l$ with the temporary public key obtained from the eTicket. It then sends it to the customer.

$$SP_x \rightarrow C : KU_T\{< S >_l, Timestamp_{SP_x}, Signature_{SP_x}\}$$

2(c) The customer sends the requested blinding factors associated with the l^{th} use of the ticket. For each element in $< S >$, the customer generates a pair of values, $<$ *response_S*$_{ilb}$ $>$ where $b = \{0, 1\}$.

$$\begin{cases} id\_reveal_{il} & \text{if } S_i = 0, b = 0 \\ g(id\_reveal_{il} \text{ XOR } (C\_info) \text{ XOR } l) & \text{if } S_i = 0, b = 1 \\ id\_reveal_{il} \text{ XOR } (C\_info) & \text{if } S_i = 1, b = 0 \\ g(id\_reveal_{il}) & \text{if } S_i = 1, b = 1 \end{cases}$$

where $i \in [1, \frac{k}{2}], b \in \{0, 1\}, l$: use number.

The customer produces $<$ *response_S* $>_l$ by aggregating these value pairs; it then encrypts it with the service provider's public key and sends it to the service provider.

$$C \rightarrow SP_x :$$
$$KU_{SP_x}\{< response\_S >_l, Timestamp_{C''}, Signature_{T'''}\}$$

2(d) At this point the service provider has a set of responses to its challenge and it needs to verify the validity of those responses. In order to accomplish this, the service provider computes two factors x'_i and y'_i $\forall i \in [1, \frac{k}{2}]$, as follows.

$$x'_{il} = \begin{cases} response\_S_{il1} & \text{if } S_i = 0 \\ g(response\_S_{il0} \text{ XOR } l) & \text{if } S_i = 1 \end{cases}$$
$$y'_{il} = \begin{cases} g(response\_S_{il0}) & \text{if } S_i = 0 \\ response\_S_{il1} & \text{if } S_i = 1 \end{cases}$$

If the responses are valid, then we have that:

$$x'_{il} = g(id\_reveal_{il} \text{ XOR } (C\_info) \text{ XOR } l),$$
$$y'_{il} = g(id\_reveal_{il}).$$

The service provider calculates an identity check value for the l^{th} use of the ticket from *signed_id_array*$_l$:

$$< identity\_check\_value >=$$
$$= \frac{\prod_{i=1}^{\frac{k}{2}} < signed\_id\_array >_{il}}{\prod_{i \in [1, \frac{k}{2}]} blind_{ij}}$$
$$= \prod_{i \in [1, \frac{k}{2}]} f^{d_{TA}}(x_{il}, y_{il})$$

Consequently, the service provider checks whether:

$$< identity\_check\_value >^{e_{TA}} \equiv \prod_{i \in [1, \frac{k}{2}]} f(x'_{il}, y'_{il})$$

If so, $<$ *signed_id_array* $>_l$ is valid. Note that the service provider has enough information to confirm that the customer's identity in $<$ *signed_id_array* $>_l$ is of the proper form, but does not have enough information to compromise the customer's anonymity.

Finally, the service provider(i) determines which services have been granted to the customer; (ii) stores $<$ *eTicket* $>$, l, $< responses_S >_l$ and *chain*$_l$ in case of overspending.

It can then grant the service. If the service is in the form of an electronic transaction, then the service provider should obtain a receipt from the customer as before; without this the customer could deny having received the service. If the service is a real-world service, then alternative mechanisms could be used by the service provider in proving that the customer actually used the service (e.g., CCTV footage). However, the situation here is no different to one in which a customer claims that a service provider has accepted and taken a physical ticket but refused to honour it.

4.5 eTicket cashing phase

3(a) There are two possible mechanisms by which payment may occur: (i) the service provider wishes to take a pre-agreed cut from the revenue received by the ticket agent for each use of a ticket. It therefore needs to establish this usage in such a way that the ticket agent can verify it; and (ii) the ticket agent is paid a bonus that is determined by ticket usage. Unfortunately, it is always possible for the service provider to under-report usage. However, the use of hidden inspectors who purchase and use tickets, but report this use additionally to the ticket agent, can show when underreporting is occurring.

In either case the service provider sends the eTicket to the ticket agent, together with the last received chain element and corresponding blinding factors in the form of $< responses_S >$'s.

$$SP_x \rightarrow TA : KU_{TA}\{< eTicket >, chain_{last},$$
$$< response\_S >, Timestamp_{SP'_x}, Signature_{SP'_x}\}$$

The ticket agent checks for overspending by examining the ticket identity, the set of revealed blinding factors, and the last chain element. If these are the same as on a previous occasion, the ticket agent will, in the case that it is paying the service provider per use, object. If the last chain element is the same, but $<$ *response_S* $>$ differs, then the customer has attempted to double-use a ticket and their identity is forfeit as in the following situation.

Assume that the customer attempts to use the l^{th} identity column at two different service providers A and B. Consequently, he must then answer both challenge S_A and challenge S_B sent by A and B, respectively. It is very likely that vectors S_A and S_B differ in at least one value for a given vector position. Let be $S_A = (\ldots, 1, \ldots)$ and $S_B = (\ldots, 0, \ldots)$. As such, the corresponding responses to A and B will have included $\{\textit{response\_S}_{il0\,A}\} = (\ldots, \textit{id\_reveal}_{il}\ \text{XOR}\ (\textit{C\_info}), \ldots)$ and $\{\textit{response\_S}_{il1\,B}\} = (\ldots, \textit{id\_reveal}_{il}, \ldots)$. Upon receiving the responses from A and B, the ticket agent combines them to find out the customer's personal information (*C_info*) as follows:

$$(\textit{C\_info}) = (\textit{id\_reveal}_{il}\ \text{XOR}\ (\textit{C\_info}))\ \text{XOR}\ \textit{id\_reveal}_{il}$$

5 Analysis of the requirements

Customer anonymity Our approach preserves the privacy of the customer's identity from both the ticket agent and service providers. Clearly, there must be a phase in which the customer pays the price of an eTicket, during which they will not be anonymous if their payment method is not anonymous. However, this phase is unlinked from subsequent eTicket generation by virtue of the use of blind signatures. The only remaining linkages are (i) the price of the ticket and (ii) timing. The latter can be addressed to some extent by introducing delays into the system, but this affects its usability. The former could be more problematic (especially when coupled with the latter), particularly if the negotiation of the price of a ticket takes place when the ticket agent knows the identity of the customer: for example, if the ticket agent issues tickets with values that differ slightly from customer to customer, then they are able to narrow the number of possible candidates for the identity of the customer, particularly if timing is taken into account. If, however, ticket selection precedes negotiation with the ticket agent (the client obtains a price from the newspaper, for example) or the customer is willing to overpay in order to obscure the true value of the ticket they are purchasing, then this mechanism may be less successful. In general, identity is less likely to be compromised in a busy ticket agent with a small range of products than in one that is largely idle, but weak linkage is unavoidable.

Following purchase and issue, neither the ticket agent nor service providers can, in general, trace the customer identity from an eTicket. The eTicket is anonymous and both issue and spending phases are carried out without exposing customer's identity: messages are encrypted with anonymous temporary keys.

Revocable anonymity Our scheme allows customer the maintenance of properties of anonymity described above, as long as the customer is honest. If he is not, there are mechanisms for revealing the identity of eventual over-spenders. A service provider accepts the eTicket from the customer knowing that if, and only if, he cheats, the ticket agent can reveal the identity embedded into the eTicket and punish the customer to make up the loss.

Non-repudiation Our work includes a non-repudiation feature that prevents any party involved from denying previous commitments or actions. Customers can neither deny their eTicket requests nor their spending because they sign eTicket requests with their secret keys and spending messages with their anonymous private keys. On the other hand, the ticket agent cannot overcharge the customer because an eTicket value is embedded in the eTicket.

Offline checking The service provider can check eTicket validity by using the ticket agent's public key, which can be done offline without contacting the ticket agent. In cases where tickets may be expended at several outlets, overspending is possible if all such outlets are offline at the point at which the ticket is spent, but anonymity will later be forfeit if these spending records are ever merged. This is an inevitable consequence of offline checking and prevention of such overspending relies either on having outlets online, or on restricting the use of tickets to particular outlets. There is no universal way of resolving this problem and solution adopted must be based on a calculation of business risk.

Global spending Customers can spend their eTickets at any suitable service provider as the ticket agent's public key, used to verify the integrity of eTickets, is publicly available.

Ticketing models We have presented a mechanism that permits the creation of usage-constrained tickets. In addition to this, it is possible to encode other ticket-specific constraints such as expiry date and class of service into the ticket, supporting a wide range of different ticket models.

6 Future Work

The work contained in this paper represents a start to the implementation of eTickets; the presentation is intended

to be illustrative rather than prescriptive and, as a consequence, there is considerable future work to be done before this can be made into a commercially deployable system. So, for example, the Chaum cut-and-choose and blind signature approach used above was selected for ease of explanation rather than efficiency. We are currently developing this approach to utilise the more efficient Brands' scheme, which is based on Schnorr signatures.

For reasons of space, we have elided the eTicket wholesale Phase, in which the ticket agent buys tickets from the service provider. Likewise, we have elided issues of transferability. The property of non-transferability implicitly assumes that the ticket can be linked to the identity of the purchaser and, in an anonymous system, this cannot be. It would be possible to achieve this, forfeiting anonymity, by encoding identifying features (for example fingerprints) into the ticket and requiring the checking of these at the point of use. Transferability, other than transfer to a proxy, cannot simply be achieved by communicating both the temporary keys and the eTicket. If it were, misuse of the transferred ticket by the recipient would result in the revelation of the identity of the original purchaser, added to which there is nothing to prevent ticket use by the original customer.

7 Conclusion

We presented the first proof-of-concept eTicketing scheme that approaches the degree of flexibility that has been discussed at an architectural level for some years and have argued that it possesses a range of appropriate properties: (i) preservation of customer anonymity, so long as they do not seek to overspend; (ii) non-repudiation for the customer and the service provider; (iii) off-line eTicket checking at any service provider without contacting the ticket agent; (iv) eTicket use at a range of the possible service providers; (v) support for a range of different ticketing models.

We believe that there is a pressing need for more published research in this field if the full benefits of mobile commerce are to be realised.

References

[1] MeT Ticketing Requirements. Proposed Specification. Version 1.0, September 2002.

[2] MeT Ticketing Specification. Proposed Specification. Version 1.0, November 2002.

[3] Nokia, Philips and German Public Transport Network Operator RMV trial NFC for ticketing. Press release, November 2004.

[4] mTicket Mobile Ticketing. http://www.mticket.net/, June 2005.

[5] F. Bao, L. Anantharaman, and R. Deng. Design of portable mobile devices based e-payment system and e-ticketing system with digital signature. In *Proceedings of* 1^{st} *International Conferences on Info-tech and Info-net*, volume 6, pages 7–12. IEEE, 2001.

[6] M. Ben-Or, O. Goldreich, S. Micali, and R. L. Rivest. A fair protocol for signing contracts. In *Proceedings of the* 12^{th} *Colloquium on Automata, Languages and Programming*, volume 36, pages 43–52. Springer-Verlag, 1985.

[7] S. Brands. Untraceable off-line cash in wallet with observers. In *Proceedings of the* 13^{th} *Annual International Cryptology Conference on Advances in Cryptology*, volume 773 of *Lecture Notes in Computer Science*, pages 302–318, Santa Barbara, California, United States, August 1994. Springer-Verlag.

[8] D. Chaum. Blind signatures for untraceable payments. In *Proceedings of the* 2^{nd} *Annual International Cryptology Conference on Advances in Cryptology*, volume 55, pages 199–203, Santa Barbara, California, USA, August 1983. Plenum Press.

[9] D. Chaum, A. Fiat, and M. Naor. Untraceable Electronic Cash. In *Proceedings of the* 8^{th} *Annual International Cryptology Conference on Advances in Cryptology*, volume 403, pages 319–327, Santa Barbara, California, August 1989. Springer-Verlag.

[10] K. Fujimura and D. Eastlake. Requirements and Design for Voucher Trading System (VTS). RFC3056, March 2003.

[11] K. Fujimura and Y. Nakajima. General-purpose Digital Ticket Framework. In *Proceedings of the* 3^{rd} *USENIX Workshop on Electronic Commerce*, pages 177–186, Boston, Massachusetts, USA, August 1998.

[12] K. Fujimura, Y. Nakajima, and J. Sekine. XML Ticket: Generalized Digital Ticket Definition Language. In *Proceedings of the the* 1^{st} *W3C Signed XML Workshop*, Taormina, Italy, April 1999.

[13] P. N. and S. Thainimit. Structure free highway toll collection using non-repudiated tickets. In *Proceedings of the International Symposium on Information and Communication Technologies*, 2003.

[14] F. Pedone. Optimistic Validation of Electronic Tickets. In *Proceedings of the* 20^{th} *Symposium on Reliable Distributed Systems*, pages 110–119, New Orleans, LA, USA, October 2001. IEEE Computer Society.

[15] T. S. K. Reddy, H. Mohanty, R. K. Ghosh, and S. K. Madria. Two Distributed Algorithms for E-ticket Validation protocols for Mobile Clients. In *Proceedings of the* 1^{st} *IEEE Conference on Electronic Commerce*, pages 223–230, Newport Beach, California, June 2003. IEEE Computer Society.

[16] B. Schoenmakers. An efficient electronic payment system withstanding parallel attacks. Technical Report. Centre for Mathematics and Computer Science. Amsterdam, The Netherlands, 1995.

[17] R. Song and L. Korba. Pay-TV System with Strong Privacy and Non-Repudiation Protection. *IEEE Transactions on Consumer Electronics*, 49(2):408– 413, 2003.