Private eCash in Practice (Short Paper)

Amira Barki[1,2], Solenn Brunet[1,3], Nicolas Desmoulins[1], Sébastien Gambs[4], Saïd Gharout[1], and Jacques Traoré[1]

[1] Orange Labs, Caen, France
[2] Sorbonne universités, Université de technologie de Compiègne (UTC), CNRS, UMR 7253 Heudiasyc, Compiègne, France
[3] Université de Rennes 1, Rennes, France
[4] Université du Québec à Montréal (UQAM), Montréal, Canada

Abstract. Most electronic payment systems for applications, such as eTicketing and eToll, involve a single entity acting as both merchant and bank. In this paper, we propose an efficient privacy-preserving post-payment eCash system suitable for this particular use case that we refer to, afterwards, as private eCash. To this end, we introduce a new partially blind signature scheme based on a recent Algebraic MAC scheme due to Chase et al. Unlike previous constructions, it allows multiple presentations of the same signature in an unlinkable way. Using it, our system is the first versatile private eCash system where users must only hold a sole reusable token (*i.e.* a reusable coin spendable to a unique merchant). It also enables identity and token revocations as well as flexible payments. Indeed, our payment tokens are updated in a partially blinded way to collect refunds without invading user's privacy. By implementing it on a Global Platform compliant SIM card, we show its efficiency and suitability for real-world use cases, even for delay-sensitive applications and on constrained devices as a transaction can be performed in only 205 ms.

Keywords: eCash, post-payment, refunds, partially blind signature, anonymity, eToll, eTicketing, EVC.

1 Introduction

Electronic Cash (eCash), introduced by Chaum [6], is the digital analogue of hard currency in which users withdraw electronic coins from a bank and spend them to merchants. Later, each merchant deposits the collected coins to the bank.

Using eCash, user's anonymity is protected both from the bank and merchants. To ensure this, it should be impossible to link the following pair of events: either a withdrawal and a spending or two spendings. However, owing to its digital nature, eCash has to be protected from duplication. Thus, eCash protocols must enable the detection of double-spending and the identification of defrauders.

To be as attractive as possible, the user's side of an eCash system is sometimes implemented on a mobile device or even a smart card. Therefore, protocols have to comply with both the limited resources of such environments as well as the stringent delay constraint arising from transactions requirements.

Public transport [2], electronic Toll (eToll) [7] and Electric Vehicle Charging (EVC) [3] are the main emerging uses cases of private eCash (*i.e.* involving a single merchant managed by the same entity acting as the bank) that significantly invade user's privacy. Indeed, transactions records may disclose user's location at a given time and reveal personal information such as work, home or habits.

Related work. Recently, several proposals have addressed this issue. However, finding a good tradeoff between necessary security properties and performance has not always been completely successful. In the sequel, we only focus on schemes related to eCash although other approaches exist in the literature [12, 4, 10].

Public transport users' privacy was tackled in [13], [11] and [2]. In the first two proposals, users pay for their trips through the use of payment tokens that are worth the highest possible fare. As fares have different values, a refund process is set up to guarantee the accurate charging of users. However, to ensure user's anonymity with respect to the transport company, the scheme of Rupp et al. [13] entails heavy verifications that constrained devices cannot handle [2]. The proposal of Milutinovic et al. [11] is also computationally expensive and less efficient than [13] as refunds are separately collected on distinct refund tokens. Arfaoui et al. [2] protocol meets the stringent delay requirement and is efficient even when implemented in constrained environment. It is also the only one allowing anonymity revocation under exceptional circumstances. Nevertheless, this scheme does not enable flexible prices. As regards to EVC user's privacy, it was addressed by Au et al. [3]. However, their scheme requires costly zero-knowledge proofs of knowledge and is not suitable for time-sensitive applications.

Finally, Day et al. [7] proposed two privacy-preserving payment systems for eToll. However, the first one is only partially private as it relies on spot checks that record some of the user's spatio-temporal information to detect and identify defrauders. Furthermore, through an exhaustive search on tokens, it is possible to trace all user's trips. In contrast, the second proposal provides full anonymity and enables double spending detection. Unfortunately, to be efficient, users have to hold a large number of tokens where each one can only be used at a specific time. Besides, similarly to eCoupons, both proposals do not allow flexible prices, which we believe to be an important issue.

Contributions. In this paper, we propose an efficient post-payment private eCash system designed for scenarios in which the same entity acts as both merchant and bank like public transport, eToll and EVC. Indeed, we leverage this feature to strike a balance between efficiency and privacy. Our system relies on a new partially blind signature scheme built based on the recent Chase et al. Algebraic MAC scheme [5]. Unlike ordinary eCash systems, users must only hold a single token that can be reused a specified number of times without allowing anyone to trace users or link their transactions. Through a refund process[5], our eCash system supports both post-payments (*i.e.* users are charged after the use of the

[5] As shown in [13], an aggregate refund amount should not enable to deduce the different toll fares and hence, the details of the individual trips the user has taken.

service) and flexible prices while removing the need to withdraw several tokens of different values. Our proposal is proven secure in the random oracle model (ROM) and its implementation on a SIM card shows that it complies with the limited computational power of constrained devices and stringent time constraints.

The paper is organized as follows. Section 2 introduces our main notation and building blocks and details our partially blind signature scheme. Based on this first contribution, Section 3 explains our eCash system through the eToll use case. As an electronic payment system dedicated to toll roads, it illustrates the challenging requirements we have to face both in terms of privacy and performance. Finally, implementation results in Section 4 show that our system is truly efficient.

2 Preliminaries

2.1 Notation

To state that x is chosen uniformly at random from the set X, we use one of the two following notations $x \xleftarrow{R} X$ or $x \in_R X$. In addition, $\vec{x}$ and $\{g_i\}_{i=1}^{l}$ respectively denote the vector $(x_0, x_1, \ldots, x_n)$ and the set $\{g_1, g_2, \ldots, g_l\}$.

Zero-Knowledge Proofs of Knowledge (ZKPK) allow a prover $\mathcal{P}$ to convince a verifier $\mathcal{V}$ that he knows some secrets verifying a given statement without revealing anything else about them. They are denoted by the usual notation in which Greek letters correspond to $\mathcal{P}$'s knowledge: $\pi := \text{PoK}\{\alpha, \beta : \textit{ statements about } \alpha, \beta\}$.

2.2 Building Blocks

Algebraic MAC in Prime-Order Group. Chase et al. introduce in [5] two MAC schemes constructed using a cyclic group of prime order. An interesting feature of their schemes is that the issuer and the verifier are actually the same entity and consequently share a set of keys. To build our private eCash scheme, we focus on their $\mathsf{MAC}_{\mathsf{GGM}}$ construction, that can be seen as a digital signature scheme, proven unforgeable under chosen message and verification attack (UF-CMVA) in the generic group model. In the following, we briefly review their construction by explaining how to sign n distinct messages $(m_1, \ldots, m_n)$:

1. $\texttt{Setup}(1^k)$ creates the system public parameters denoted $pp := (\mathbb{G}, q, g, h)$ where $\mathbb{G}$ is a cyclic group of prime order q, a k-bit prime, and g, h are two random generators such that $\log_g h$ is unknown.
2. $\texttt{KeyGen}(pp)$ generates a secret key $sk := \vec{x} \in_R \mathbb{F}_q^{n+1}$ and a value $\tilde{x}_0 \in_R \mathbb{F}_q$ to build a commitment $C_{x_0} := g^{x_0} h^{\tilde{x}_0}$ to the secret value x_0. Denoted by $iparams$, $(C_{x_0}, X_1 := h^{x_1}, \ldots, X_n := h^{x_n})$ corresponds to the issuer's public parameters.
3. $\texttt{MAC}(sk, \vec{m})$ produces an authenticated token (u, u') on $\vec{m} := (m_1, \ldots, m_n)$ where $u \in_R \mathbb{G} \backslash \{1\}$ and $u' := u^{x_0 + x_1 m_1 + \cdots + x_n m_n}$.
4. $\texttt{Verify}(sk, \vec{m}, (u, u'))$ checks the validity of the token with respect to the message $\vec{m}$. The token is accepted only if $u \neq 1$ and $u' = u^{x_0 + x_1 m_1 + \cdots + x_n m_n}$.

Based on $\mathsf{MAC}_{\mathsf{GGM}}$, Chase et al. proposed a keyed-verification anonymous credentials scheme allowing the blind issuance of credentials. However, it requires ZKPK for each hidden attribute and does not provide *perfect unlinkability* as the credential attributes are sent to the issuer encrypted, using ElGamal encryption scheme, before being signed. Thus, it is not suitable for eCash systems.

Partially blind signatures. A variation of basic digital signatures, called *blind signature*, allows a receiver $\mathcal{R}$ to get a signature on a message without revealing any information about it to the signer $\mathcal{S}$. However, in use cases like eCash, $\mathcal{S}$ may want to add some information to the blind signature such as a date, a validity period or an amount. To address this issue, Abe et al. [1] proposed an extension known as *partially blind signature*. It allows $\mathcal{R}$ and $\mathcal{S}$ to agree on a common information `info` to be added in the blind signature of a message $\vec{m}$. A partially blind signature scheme should be (1) *one-more unforgeable* (*i.e.* it should be impossible to obtain $L+1$ signatures with at most L signing requests) and (2) *unlinkable* (*i.e.* it should be impossible to link two signatures or identify for whom the signature was issued).

Through the interactive protocol $\texttt{BlindIssue}(\mathcal{R}(\vec{m}), \mathcal{S}(sk))$ described below, we detail our partially blind signature scheme based on $\mathsf{MAC}_{\mathsf{GGM}}$ and which is executed between $\mathcal{R}$ holding $\vec{m}$ and $\mathcal{S}$ who acts as the issuer holding sk:

1. $\mathcal{R}$ sends the common value `info`, a commitment $C_{\vec{m}} := h^r X_1^{m_1} \dots X_n^{m_n}$ to the message $\vec{m}$ where $r \in_R \mathbb{Z}_q^*$ as well as the ZKPK π_1 defined as follows: $\pi_1 := PoK\{\alpha_1, \alpha_2, \dots, \alpha_n, \beta : C_{\vec{m}} = h^\beta X_1^{\alpha_1} \dots X_n^{\alpha_n}\}$.
2. If π_1 is valid, $\mathcal{S}$ computes $u'' := u^{x_0}(C_{\vec{m}}(X_n)^{\texttt{info}})^b$ *s.t.* $b \in_R \mathbb{Z}_q^*$ and $u := h^b$. Then, he provides $\mathcal{R}$ with the partially blind signature $((u, u''), \texttt{info})$ as well as a ZKPK π_2 proving that $u'' := u^{x_0 + x_1 m_1 + \dots + x_{n-1} m_{n-1} + x_n (m_n + \texttt{info})} h^{br}$ and $\pi_2 := PoK\{\alpha, \beta, \gamma : u'' = u^\alpha (C_{\vec{m}}(X_n)^{\texttt{info}})^\beta \wedge C_{x_0} = g^\alpha h^\gamma\}$.
3. Finally, $\mathcal{R}$ unblinds (u, u'') and obtains the signature $(u, u' := \frac{u''}{u^r})$.

To show the obtained signature in an anonymous way, the receiver has just to randomize it by computing $(u^l, (u')^l)$ where $l \in_R \mathbb{Z}_q^*$.

In our eCash system, `info` will correspond to the refund amount and will be used with m_n to aggregate refunds. Usually, m_n may be set to zero and `info` would be the validity period, thus enabling a convenient update of signatures.

Theorem 1. *Our partially blind signature scheme is* perfectly unlinkable *and* one-more unforgeable *under the assumption that* $\mathsf{MAC}_{\mathsf{GGM}}$ *is* UF-CMVA *secure.*[6]

3 Our private eCash system: the eToll use case

3.1 System framework

Stakeholders. Our private eToll system involves three main entities: a user $\mathcal{U}$, a toll company $\mathcal{TC}$ and a set of revocation authorities $\mathcal{RA}s$ that must collaborate to revoke user's anonymity or tokens.

[6] Owing to the lack of space, we defer the proofs of Theorem 1 and Theorem 2 to an extended version.

Overview. To benefit from the eToll service, a user must first register to obtain a badge that will perform all computations on his behalf. At the beginning of each billing period, registered users receive a unique reusable token generated using our partially blind signature scheme detailed in Section 2.2. This token is worth the highest possible fare and can be reused at most $N_{\mathtt{max}}$ times. To be granted access while preserving his anonymity, the user shows a randomized version of his token. Concurrently, as toll fares are generally different, the user's token is updated in a blinded way, using our partially blind signature scheme, to add the refund amount associated to the transaction. At the end of the billing period, users are charged according to their token value. Such a post-payment approach prevent them from refilling a prepaid account with a large amount of money. However, if a user does not return his token, he will pay the maximal allowed amount corresponding to $N_{\mathtt{max}}$ trips with the highest fare.

Security and performance requirements. None of the entities can be fully trusted since all of them have some incentives to cheat. Only the user's badge is subject to the limited trust assumption that all the computations it performs are correct. Nevertheless, any attempt to cheat by tampering it must be detected. To this end, in addition to the usual *correctness* property, some security properties must be satisfied. Our private eCash system should provide (1) *unlinkability* (*i.e.* it should be impossible to link together two events such as two transactions or a transaction and a given token) which implies the regular *anonymity* property, (2) *revocability* (*i.e.* $\mathcal{RA}$s can always revoke user's anonymity and tokens), (3) *non-frameability* (*i.e.* nobody should be able to falsely accuse another user of performing a given transaction) and, (4) *unforgeability* (*i.e.* it should be impossible for users to cheat by paying less charges than what they have to). To be effective and suitable for most use cases, a transaction must be performed in at most 300 ms [2].

3.2 Description of the protocols

Our private eCash system consists of six phases. (1) The public parameters and required keys are initialized during *setup*. (2) The *Registration* phase enables a user to register to the system and to obtain his badge. (3) The *Token Issuance* phase provides legitimate users with a unique reusable token. (4) During *Access Control*, a user uses his token to be granted access at tollbooths. (5) The *Toll Computation* phase allows the computation of the user's bill based on his token value. Finally, (6) the *Revocation* phase enables user's anonymity and token revocations. Below, we explain these phases and detail the main protocols in Figure 1. Owing to space limitations, the ZKPKs are not detailed. Except otherwise specified, they are quite standard and many values in these proofs can be *precomputed*.

Setup. Let $pp = (\mathbb{G}, g, h, q, g_R, N_{\mathtt{max}}, \{g_i\}_{i=1}^{N_{\mathtt{max}}})$ denote the public parameters where $\mathbb{G}$ is a cyclic group of prime order q and $(g, h, g_R, \{g_i\}_{i=1}^{N_{\mathtt{max}}})$ a set of random generators. $N_{\mathtt{max}}$ indicates the allowed number of reuses of a token and could be set according to user's needs. Each user $\mathcal{U}$ is also provided with a pair of keys (sk_u, pk_u) that identifies him.

The toll company shares the secret key $\vec{x} := (x_0, x_1, x_2)$ with tollbooths that are denoted by $\mathcal{TC}$ as well. The associated public parameters are $C_{x_0} := g^{x_0} h^{\tilde{x}_0}$, a commitment to x_0 where $\tilde{x}_0 \in_R \mathbb{Z}_q^*$, and $X_1 := h^{x_1}$, $X_2 := h^{x_2}$. They are also provided with a pair of keys (sk_{tc}, pk_{tc}) used to sign transaction data.

The revocation authorities jointly generate two pairs of keys: (sk_{ra}, pk_{ra}) of the threshold ElGamal and (sk_{rp}, pk_{rp}) of the threshold Paillier cryptosystems. Paillier encryption scheme is used as an *extractable commitment* (see [9]) to satisfy the unforgeability requirement, even in a concurrent setting, where an adversary is allowed to interact with $\mathcal{TC}$ in an arbitrarily interleaving (concurrent) manner. Let g_E and g_P be two generators of $\mathbb{G}$. ElGamal keys are defined as $sk_{ra} := x_T \in \mathbb{Z}_q^*$ and $pk_{ra} := (g_E, X_T := g_E^{x_T})$. Paillier pair of keys consists of $sk_{rp} := (a, b)$ and $pk_{rp} := (g_P, n := ab)$ where a and b are two different random primes such that $|a| = |b|$ and $gcd(ab, (a-1)(b-1)) = 1$. The private keys are shared among $\mathcal{RA}$s [8] and at least t of them should cooperate to identify the user or revoke a token.

Registration. To use the service, $\mathcal{U}$ must provide $\mathcal{TC}$ with his public key pk_u and a ZKPK proving the knowledge of the secret key sk_u. If the proof is valid, $\mathcal{U}$ receives a personal badge $\mathcal{B}_u$ including a SIM card. It allows $\mathcal{U}$ to anonymously use the service. Moreover, pk_u is saved in a dedicated database denoted by $\mathtt{DB}_{\mathtt{REG}}$.

Token Issuance. The *token issuance* phase occurs at the beginning of each billing period upon a signed request of a registered user. During this phase, $\mathcal{TC}$ provides $\mathcal{U}$ with a permission token $T := (u, u' := u^{x_0 + x_1 s_u + x_2 m})$. It is a partially blind signature on the unknown message $s_u = s + s'$ and the common information m corresponding to the refund amount, initially set to 0. In fact, s_u is a secret value only known by $\mathcal{U}$: it involves a secret $s \in_R \mathbb{Z}_q^*$ chosen by $\mathcal{U}$ and hidden from $\mathcal{TC}$ and $s' \in_R \mathbb{Z}_q^*$ chosen by $\mathcal{TC}$ and provided to $\mathcal{U}$. The token is worth the highest possible toll fare and can be reused $N_{\mathtt{max}}$ times. Two ZKPKs π_1 and π_2 ensure that exchanged values are well-formed (see Figure 1). Besides, (g^s, s', D) is saved in $\mathtt{DB}_{\mathtt{REG}}$ where D is a Paillier encryption of s necessary for token revocation.

Access Control. To be granted access at tollbooths, $\mathcal{U}$ provides a randomized version of his token T and an ElGamal encryption E of g^{s_u} along with a ZKPK π_3 proving that these values are well-formed. Upon receipt, $\mathcal{TC}$ checks the token both for the allowed number of uses and validity. Indeed, whenever reaching a tollbooth, $\mathcal{B}_u$ randomly chooses a g_i among the set F of unused ones. The selected g_i is then removed from F to prevent over-spending of a token. If checks succeed, $\mathcal{U}$ receives an updated T with a new m aggregating all the refunds collected so far. This new token is computed using our partially blind signature scheme with a common value equal to the current refund amount. Due to delay constraint, the associated ZKPK π_4 cannot be instantly verified. Thus, $\mathcal{U}$ is also provided with S, an RSA signature with a short public verification exponent, of all the received values. S can be quickly verified upon receipt while π_4 is rather checked

User $\mathcal{U}$		**Toll Company $\mathcal{TC}$**
Public Input: pp, X_1, X_2, C_{x_0}, pk_{tc}, pk_{ra}, pk_{rp} and pk_u		
(1) Token Issuance Protocol		
Private Input: sk_u		**Private Input:** (x_0, x_1, x_2), $\texttt{DB}_{\texttt{REG}}$
Choose $r, r_1, s \xleftarrow{R} \mathbb{Z}_q^*$ **Compute** $C \leftarrow h^r X_1^s$, $W = g^s$, $D = g_P^s r_1^n$		
Build $\pi_1 = \text{PoK}[\alpha, \beta, \gamma : C = h^\alpha X_1^\beta$ $\wedge\ W = g^\beta \wedge\ D = g_P^\beta \gamma^n]$	$\xrightarrow[Sign(C,\pi_1,W,D)]{C,\pi_1,W,D}$	**Check** π_1 **and Choose** $s', b \xleftarrow{R} \mathbb{Z}_q^*$ **Compute** $u \leftarrow h^b$ $u'' \leftarrow u^{x_0}(CX_1^{s'})^b$
Check π_2 **Compute** $s_u \leftarrow s + s'$ and $u' \leftarrow \frac{u''}{u^r}$ $T \leftarrow (u, u'), m \leftarrow 0, F \leftarrow \{g_i\}_{i=1}^{N_{\max}}$	$\xleftarrow{s',(u,u''),\pi_2}$	**Build** $\pi_2 = \text{PoK}[\alpha, \beta, \gamma : u = h^\alpha$ $\wedge u'' = u^\beta (CX_1^{s'})^\alpha \wedge C_{x_0} = g^\beta h^\gamma]$ **Save** $(W,\ D,\ s')$ in $\texttt{DB}_{\texttt{REG}}$
(2) Access Control Protocol		
Public Input: m'		
Private Input: $T := (u, u')$, m, s_u, F		**Private Input:** (x_0, x_1, x_2), $\texttt{DB}_{\texttt{AC}}$
Choose l, r, z_1, z_2, t and $b \xleftarrow{R} \mathbb{Z}_q^*$, $g_i \xleftarrow{R} F$ **Compute** $w \leftarrow u^l; w' \leftarrow (u')^l$; $c' \leftarrow w' g^r$ $c_1 \leftarrow w^{s_u} h^{z_1}$; $c_2 \leftarrow w^m h^{z_2}$; $F \leftarrow F \backslash \{g_i\}$ $V \leftarrow g^{-r} X_1^{z_1} X_2^{z_2}$; $A \leftarrow h^t X_1^{s_u} X_2^m$ $E \leftarrow (e_1 = g_E^b, e_2 = g^{s_u} X_T^b)$; $T_i \leftarrow g_i^{s_u}$		
Build $\pi_3 = \text{PoK}[\alpha, \beta, \gamma, \delta, \sigma, \mu, \eta : T_i = g_i^\sigma$ $\wedge\ c_2 = w^\mu h^\gamma \wedge V = g^{-\alpha} X_1^\beta X_2^\gamma \wedge e_1 = g^\eta$ $c_1 = w^\sigma h^\beta \wedge A = h^\delta X_1^\sigma X_2^\mu \wedge e_2 = g^\sigma X_T^\eta]$	$\xrightarrow[T_i,A,g_i,\pi_3,E]{w,c',c_1,c_2,V}$	**Check if** $V \stackrel{?}{=} \frac{w^{x_0} c_1^{x_1} c_2^{x_2}}{c'}$ **Check** π_3 and $T_i \notin \texttt{DB}_{\texttt{AC}}$ **Choose** $d \xleftarrow{R} \mathbb{Z}_q^*$; **Compute** $y \leftarrow h^d$ **Compute** $y'' \leftarrow y^{x_0}(AX_2^{m'})^d$ $S = Sign_{RSA}(m', y, y'', \pi_4)$
Check π_4 and S **Compute** $y' \leftarrow \frac{y''}{y^t}$, $m \leftarrow m + m'$ and $T \leftarrow (y, y')$	$\xleftarrow{m',y,y'',\pi_4,S}$	**Build** $\pi_4 = \text{PoK}[\alpha, \beta, \gamma : y = h^\alpha$ $\wedge y'' = y^\beta (AX_2^{m'})^\alpha \wedge C_{x_0} = g^\beta h^\gamma]$ **Save** (E, T_i) in $\texttt{DB}_{\texttt{AC}}$
(3) Toll Computation Protocol		
Private Input: $T := (u, u')$, m, s_u		**Private Input:** (x_0, x_1, x_2), $\texttt{DB}_{\texttt{REG}}$
Choose $l, r, z_1 \xleftarrow{R} \mathbb{Z}_q^*$ **Compute** $R \leftarrow u^l$; $R' \leftarrow (u')^l$; $c' \leftarrow R' g^r$ $c_1 \leftarrow R^{s_u} h^{z_1}$; $V \leftarrow g^{-r} X_1^{z_1}$; $T_g \leftarrow g_R^{s_u}$		
Build $\pi_5 = \text{PoK}[\alpha, \beta, \gamma : T_g = g_R^\alpha$ $\wedge\ c_1 = R^\alpha h^\beta \wedge V = g^{-\gamma} X_1^\beta]$	$\xrightarrow[V,\ T_g,\pi_5]{m,R,c',c_1}$	**Check if** $V \stackrel{?}{=} \frac{R^{x_0+mx_2} c_1^{x_1}}{c'}$ **Check** π_5 and $T_g \notin \texttt{DB}_{\texttt{REG}}$ **Save** T_g in $\texttt{DB}_{\texttt{REG}}$

Fig. 1: Our private eCash system: the eToll use case

during the idle time of the SIM card. Concurrently, $(E, T_i := g_i^{s_u})$ is saved in the database of transactions $\texttt{DB}_{\texttt{AC}}$. Note that, to provide *full unlinkability*, one may add randomness in T_i using a pseudo-random function as in [2].

Toll Computation. At the end of the billing period, $\mathcal{U}$ shows his randomized token T and a tag $T_g := g_R^{s_u}$ to be charged for all his trips. The tag ensures that $\mathcal{U}$ has not already asked for a refund during that period. Based on the refund amount m, $\mathcal{TC}$ computes the user's charges and saves T_g in $\texttt{DB}_{\texttt{REG}}$. If a user's token has been used less than $N_{\max}$ times, a specific process emulates their use with no associated charges to ensure that $\mathcal{U}$ will only pay for the trips he took.

	Card Precomputation (1)	Get Data from card (2)	$\mathcal{TC}$ computation (3)	Send Data to card (4)	Card Verification (5)	**Total On-line part**
Battery-On:	(1672-1688) 1678	(66-68) 67	(9-34) 22	(96-115) 102	(501-522) 511	(186-224) **205**
Battery-Off:		85		(175-184) 182		(298-322) **315**

Table 1: Timings ((min-max) average in ms) of the *Access Control* Protocol. The off-line computations (steps 1 and 5) are launched from the smartphone (battery-on). On-line computations concern steps 2, 3 and 4, and can be done battery-off.

Revocation. Two different revocations may be triggered in exceptional circumstances: the revocation of user's anonymity or tokens. In the former, the goal is to identify the user who performed a given access control (*e.g.* for national security reasons). To do so, $\mathcal{TC}$ sends to $\mathcal{RA}$s the ElGamal encryption E of g^{s_u}. At least t of them should collaborate to recover g^{s_u}. Using the information stored in $\mathtt{DB}_{\mathtt{REG}}$, $\mathcal{TC}$ identifies the corresponding user. In the latter, the aim is to revoke a token following, for example, the loss or theft of the badge. To this end, $\mathcal{RA}$s are provided with the Paillier encryption D of the secret s that they jointly decrypt. Thereby, $\mathcal{TC}$ can compute $s_u = s + s'$ and thus blacklists all the $\{T_i := g_i^{s_u}\}_{i=1}^{N_{\max}}$.

Theorem 2. *Our private eCash system is* unlinkable *under the Decisional Composite Residuosity (DCR) and the Decisional Diffie-Hellman (DDH) assumptions,* unforgeable *and* revocable *under the assumption that* $\mathsf{MAC}_{\mathsf{GGM}}$ *is UF-CMVA secure and* non-frameable *under the Discrete Logarithm assumption, in the ROM.*

4 Performance assessment

Table 1 gives timing results of the implementation of the *Access Control* protocol on a Javacard 2.2.2 SIM card, Global Platform 2.2 compliant, embedded in a Samsung galaxy S3 NFC smartphone. The only particularity of our card, compared to the javacard specifications, is some additional API provided by the card manufacturer enabling modular and elliptic curve operations. Although the used SIM card is more powerful than most cards, as it requires a cryptoprocessor to be able to handle asymmetric cryptography, it is worth emphasizing that such powerful SIM cards with cryptoprocessors are already widely deployed by some mobile phone carriers, such as Orange in France, to provide NFC-based services.

The implementation uses a 256-bit prime elliptic curve. To have the fastest possible verification on card, $\mathcal{TC}$ uses an RSA signature scheme with a short public verification exponent. Since our private eCash system is not only intended for eToll but also for public transport, communications between the SIM card in the smartphone and the PC (Intel Xeon CPU 3.70GHz) acting as $\mathcal{TC}$ was done in NFC using a standard PC/SC reader (an Omnikey 5321). "Battery-Off" denotes a powered-off mobile phone either by the user or because its battery is flat. In this case, as stated by NFC standards, NFC-access to the SIM card is still possible, but with degraded performances. On average, the on-line part of the *Access Control* protocol is very fast even with a powered-off phone. In fact, data exchange is the most time-consuming task.

5 Conclusion

In this paper, our contribution is twofold. First, we proposed a new partially blind signature scheme that relies on Chase et al. Algebraic MAC scheme. Then, based on it, we designed a private eCash system that only requires users to hold a unique reusable token while preserving their privacy. Through a refund process, it also enables flexible prices as well as post-payments. Finally, implementation results show its efficiency even when implemented on a SIM card.

References

1. Abe, M., Fujisaki, E.: How to date blind signatures. In: Kim, K., Matsumoto, T. (eds.) Advances in Cryptology ASIACRYPT '96, LNCS, vol. 1163, pp. 244–251. Springer Heidelberg (1996)
2. Arfaoui, G., Lalande, J., Traoré, J., Desmoulins, N., Berthomé, P., Gharout, S.: A practical set-membership proof for privacy-preserving NFC mobile ticketing. Proceedings on Privacy Enhancing Technologies abs/1505.03048 (2015)
3. Au, M.H., Liu, J., Fang, J., Jiang, Z., Susilo, W., Zhou, J.: A new payment system for enhancing location privacy of electric vehicles. IEEE Transactions on Vehicular Technology 63(1), 3–18 (Jan 2014)
4. Balasch, J., Rial, A., Troncoso, C., Preneel, B., Verbauwhede, I., Geuens, C.: PrETP: Privacy-preserving electronic toll pricing. In: Proceedings of the 19th USENIX Conference on Security. pp. 5–5. USENIX Security'10 (2010)
5. Chase, M., Meiklejohn, S., Zaverucha, G.: Algebraic MACs and keyed-verification anonymous credentials. In: Proceedings of the 2014 ACM SIGSAC CCS. pp. 1205–1216. CCS '14, ACM, New York, NY, USA (2014)
6. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R., Sherman, A. (eds.) Advances in Cryptology, pp. 199–203. Springer US (1983)
7. Day, J., Huang, Y., Knapp, E., Goldberg, I.: SPEcTRe: spot-checked private ecash tolling at roadside. In: WPES. pp. 61–68. ACM (2011)
8. Fouque, P.A., Poupard, G., Stern, J.: Sharing decryption in the context of voting or lotteries. In: Frankel, Y. (ed.) Financial Cryptography, LNCS, vol. 1962, pp. 90–104. Springer Berlin Heidelberg (2001)
9. Hufschmitt, E., Traoré, J.: Fair blind signatures revisited. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing-Based Cryptography Pairing 2007, LNCS, vol. 4575, pp. 268–292. Springer Berlin Heidelberg (2007)
10. Meiklejohn, S., Mowery, K., Checkoway, S., Shacham, H.: The phantom tollbooth: Privacy-preserving electronic toll collection in the presence of driver collusion. In: Proceedings of the 20th USENIX Conference on Security. pp. 32–32. SEC'11 (2011)
11. Milutinovic, M., Decroix, K., Naessens, V., De Decker, B.: Privacy-preserving public transport ticketing system. In: Samarati, P. (ed.) Data and Applications Security and Privacy XXIX, LNCS, vol. 9149, pp. 135–150. Springer International Publishing (2015)
12. Popa, R.A., Balakrishnan, H., Blumberg, A.J.: VPriv: Protecting privacy in location-based vehicular services. In: Proceedings of the 18th Conference on USENIX Security Symposium. pp. 335–350. SSYM'09 (2009)
13. Rupp, A., Hinterwlder, G., Baldimtsi, F., Paar, C.: P4R: Privacy-preserving pre-payments with refunds for transportation systems. In: Sadeghi, A.R. (ed.) FC 2013, LNCS, vol. 7859, pp. 205–212. Springer Heidelberg (2013)