Privacy-Protecting Coupon System Revisited

Lan Nguyen

CSIRO ICT Centre, Australia
LanD.Nguyen@csiro.au

Abstract. At FC'05, Chen et al. introduced an elegant privacy protecting coupon (PPC) system, CESSS05 [13], in which users can purchase multi-coupons and redeem them unlinkably while being prevented from overspending or sharing the coupons. However, the costs for issuing and redeeming coupons are linear to the redeeming limit. Security of the system is not proved and only some arguments on system properties are provided. Coupons last indefinitely and can not be terminated. In this paper, we propose the first PPC system with constant costs for communication and computation. Coupons are revokable and the system is provably secure.
Keywords: coupon, transaction, anonymity and privacy.

1 Introduction

Coupons are a useful means by which businesses may attract the attention of new customers, or to increase the attractiveness of their business to existing customers. A coupon can be redeemed for value in a transaction, but only with the businesses associated with the coupon issuer, and so provides an incentive for the user to buy products from those businesses rather than from other potential suppliers [22]. A coupon may be able to be presented for more than one instance of service: e.g. a cinema may sell a booklet of prepaid vouchers at a discount that can be redeemed in the future for movie tickets. Prescriptions for medicines can also be seen as a form of coupon, where the value to users is not in price, but in access to some otherwise restricted goods, and a prescription may be valid for the supply of several courses of the medication. Such coupons are called multi-coupons. Because multi-coupons (like other coupons) can be only redeemed at outlets associated with the coupon issuer, they represent a form of loyalty scheme by giving a price incentive to use the issuer's preferred businesses over other businesses. Coupons are particularly interesting in Internet commerce, because some of the costs normally associated with changing one's business from one supplier to another, for example ease of access to their retail outlets, or familiarity with their staff, can be much lower on the Internet than in traditional retail markets.

If coupons can be associated with their users, they represent a further advantage for the business, since they allow purchases to be linked and the user's buying habits to be collected by the vendor, even for transactions otherwise paid in cash. This allows the business to target their marketing, including their marketing through coupons, more closely to the individual. However, it is not attractive to the coupon user, since they may wish to maintain the privacy of the nature of their purchasing from the business, as they can by using cash payments. If unlinkability between purchases and untraceability of users of the coupon can be assured, those who might otherwise do cash business with normal retail outlets might be attracted to do Internet business using coupons. A coupon issuer who wishes to attract privacy-sensitive users may wish to forgo some of the other marketing advantages offered by linkable or traceable coupons, and provide users with a coupon system that assures them that their purchasing history cannot be captured by their use of coupons, but maintains the other desirable properties of a coupon system. PPC systems can be used in such situations.

In a PPC system, a multi-coupon presented for redemption will only reveal that it is still valid for redemption and no further information can be deduced. Redemptions cannot be linked with each other, and no information is revealed that can link the coupon with the issuing process, so even if a coupon is purchased by credit card, its use cannot be linked back to the owner of the card.

Chen et al. argue that on-selling a coupon, where the whole of the coupon's remaining value is purchased by another user, or splitting a coupon, where several users each agree to only use some fraction of the coupon, can both be discouraged if it is not possible to determine from an examination of the multi-coupon how many of the component coupons are still redeemable [13]. In the case of on-selling, the purchaser must then trust the seller as to the number of valid coupons remaining on the multi-coupon. For splitting, each of the users who split a multi-coupon must trust all the other users not to use more than their share of the multi-coupon. A PPC system should at least provide this property, which is termed all-or-nothing-sharing.

The CESSS05 system does not provide revocability, another important property. It is common practice for coupons to have a limited validity in time, and they commonly carry an expiry date. This provides an incentive for the coupon user to possibly make a purchase earlier than they would otherwise, or consume more than they would otherwise (to, say, use all the coupons in a multi-coupon to avoid loss if they have some investment in the multi-coupons).

Our construction.
Taking a different approach from CESSS05, we propose a PPC system providing these properties with improved efficiency, security and functionality. This is the first multi-coupon system whose costs, both communication and computation, do not depend on the bound on the value of the multi-coupon. Our system is also the first to combine protection of the user's privacy with revocability of the multi-coupon. We present a security model for PPC systems and prove security of our system in this model. Our methodology for PPC can also be used to build a dynamic k-times anonymous authentication system with constant costs [17].

Several coupon systems have been proposed, but none of them provides the properties offered by our system, especially in term of efficiency. Some systems, such as [12], [5] and [20], protect user privacy but allow coupon splitting. Some systems [15, 24], reveal the coupon's remaining value or allow transaction linkability. And coupon redemption can not be limited in some other systems [6, 21]. The k-times anonymous authentication systems proposed in [23, 19] can be modified to be a PPC system of k-redeemable coupons, but its redeem protocol requires a proof of knowledge of one-out-of-k discrete logs, so its communication and computation costs for redeeming are linear to k.

The following section details some preliminary cryptographic primitives on which our system depends. Section 3 presents our PPC system and compares it in more detail with the CESSS05 system. Sections 4 presents the PPC security model and security proofs of our system are shown in section 5.

2 Preliminaries

This section reviews the bilinear mapping concept, related complexity assumptions, signature schemes with efficient protocols and the CESSS05 PPC system. The following notation, introduced in [10] for proofs of knowledge, will be used in this paper. For example,

$$PK\{(\alpha, \beta, \gamma) : y = g^{\alpha}h^{\beta} \wedge z = a^{\alpha}b^{\gamma}\}$$

denotes “a zero-knowledge Proof of Knowledge of integers α, β and γ satisfying $y = g^{\alpha}h^{\beta}$ and $z = a^{\alpha}b^{\gamma}$”. By $x \in_R S$ we mean x is randomly chosen from a set S. PPT denotes probabilistic polynomial time.

2.1 Bilinear Groups

Suppose $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are multiplicative cyclic groups of the same prime order p, and there is an isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$. Let g_1 and g_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, such that $\psi(g_2) = g_1$. A bilinear map is a function $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ satisfying the following properties:

1. Bilinear: $e(u^a, v^b) = e(u, v)^{ab}$ for all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}$.
2. Non-degeneracy: $e(g_1, g_2) \neq 1$.
3. Computability: There exists an efficient algorithm for computing e.

2.2 Complexity Assumptions

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be cyclic groups of prime order p, and let g_1 and g_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. The Strong Diffie-Hellman (SDH) [2] and Decisional Bilinear Diffie-Hellman Inversion (DBDHI) [3] assumptions are briefly reviewed as follows.

Strong Diffie-Hellman assumption. The q-SDH problem is defined as “computing a pair $(x, g_1^{1/(\gamma+x)})$, given a tuple $(g_1, g_2, g_2^{\gamma}, g_2^{(\gamma^2)}, \ldots, g_2^{(\gamma^q)})$”. The q-SDH assumption states that no PPT algorithm can solve the q-SDH problem with non-negligible probability.

Decisional Bilinear Diffie-Hellman Inversion assumption. The DBDHI problem is defined as "distinguishing between $(g_1, g_2, g_2^{\gamma}, \ldots, g_2^{(\gamma^q)}, e(g_1, g_2)^{1/(\gamma)})$ and $(g_1, g_2, g_2^{\gamma}, \ldots, g_2^{(\gamma^q)}, \Lambda)$, where γ is randomly chosen from $\mathbb{Z}_p^*$ and Λ is randomly chosen from $\mathbb{G}_T^*$". The DBDHI assumption states that no PPT algorithm can solve the DBDHI problem with non-negligible probability.

2.3 BB Signature Scheme

This signature scheme [2], which is unforgeable under a weak chosen message attack, allows simple and efficient signature generation and verification. It is also efficient to prove knowledge of a BB signature without revealing anything about the signature and message.

Key Generation. Let $\mathbb{G}_1$, $\mathbb{G}_2$, p, g_1 and g_2 be bilinear mapping parameters as generated above. Generate random $\gamma \in_R \mathbb{Z}_p^*$ and compute $w \leftarrow g_2^{\gamma}$. The public key is (g_1, g_2, w) and the secret key is γ.

Signing. Given a message $r \in \mathbb{Z}_p \setminus \{-\gamma\}$, output the signature $a \leftarrow g_1^{1/(\gamma+r)}$.

Verification. Given a public key (g_1, g_2, w), a message $r \in \mathbb{Z}_p \setminus \{-\gamma\}$, and a signature $a \in \mathbb{G}_1$, verify that $e(a, wg_2^r) = e(g_1, g_2)$.

2.4 CL-SDH Signature Scheme

Camenisch and Lysyanskaya proposed a signature scheme (CL) [8] which possesses 3 valuable properties. It is possible to generate a single CL signature for multiple messages. A signer can produce a CL signature for many messages without learning anything about the messages. And there is an efficient zero-knowledge proof of knowledge of a CL signature and its messages.

A variant with these properties is the CL-SDH signature scheme, whose security relies on the SDH assumption. As we do not use this scheme to generate a single signature for multiple messages, the following just present signing and verifying algorithms for a single message.

Key Generation. Let $\mathbb{G}_1$, $\mathbb{G}_2$, p, g_1 and g_2 be bilinear mapping parameters. Generate random $\gamma \in_R \mathbb{Z}_p^*$ and $b, c \in_R \mathbb{G}_1$ and compute $w \leftarrow g_2^{\gamma}$. The public key is (b, c, g_1, g_2, w) and the secret key is γ.

Signing. On input $x \in \mathbb{Z}_p^*$, generate random $s \in_R \mathbb{Z}_p$ and $d \in_R \mathbb{Z}_p \setminus \{-\gamma\}$ and compute $v \leftarrow (g_1^x b^s c)^{1/(\gamma+d)}$. The signature is (v, d, s).

Verification. Given a public key (b, c, g_1, g_2, w), a message $x \in \mathbb{Z}_p^*$, and a signature (v, d, s), verify that $e(v, wg_2^d) = e(g_1^x b^s c, g_2)$.

Security of the CL-SDH signature scheme is stated in Theorem 1, which can be proved similarly to the proof for the CL signature scheme [8].

Theorem 1. *The CL-SDH signature scheme is unforgeable under chosen message attacks if the SDH assumption holds.*

2.5 BlindSign-SDH Protocol

In the CESSS05 system, the vendor runs a *BlindSign* protocol [8] to issue a CL signature for multiple messages to a user without learning anything about the messages. Similarly, the *BlindSign-SDH* protocol allows a user to obtain a CL-SDH signature without revealing anything about the corresponding message to the signer. Let $\mathbb{G}_1$, $\mathbb{G}_2$, p, g_1 and g_2 be bilinear mapping parameters. Suppose $((b, c, g_1, g_2, w), \gamma)$ is a pair of CL-SDH public key and secret key, the BlindSign-SDH protocol between a user $\mathcal{U}$ and a signer $\mathcal{S}$ is executed as follows.

Common input: the CL-SDH public key (b, c, g_1, g_2, w).
User's input: a message $x \in \mathbb{Z}_p^*$.
Signer's input: the CL-SDH secret key γ.

Protocol:

– $\mathcal{U}$ chooses $s' \in_R \mathbb{Z}_p$, sends $D = g_1^x b^{s'}$ to $\mathcal{S}$ with a proof $PK\{(\xi, \sigma) : D = g_1^{\xi} b^{\sigma}\}$.
– After checking that the proof is valid, $\mathcal{S}$ generates random $s'' \in_R \mathbb{Z}_p$ and $d \in_R \mathbb{Z}_p \setminus \{-\gamma\}$ and computes $v \leftarrow (Db^{s''}c)^{1/(\gamma+d)}$. $\mathcal{S}$ then sends (v, d, s'') to $\mathcal{U}$.
– $\mathcal{U}$ computes $s \leftarrow s' + s''$, checks if $e(v, wg_2^d) = e(g_1^x b^s c, g_2)$, and obtains the CL-SDH signature (v, d, s) for x.

Note that x and s are $\mathcal{U}$'s secrets and $\mathcal{S}$ does not learn anything about x from the protocol.

2.6 CESSS05 PPC System

The system consists of an *Initialisation* algorithm and 2 protocols, *Issue* and *Redeem*. There is a *vendor* and many *users*. The vendor can issue a m-redeemable coupon to a user such that the user can *unlinkably* redeem the coupon for exactly m times. The system also provides *all-or-nothing-sharing* property.

The *initialisation* algorithm generates a system public key, which is a CL public key, and a vendor secret key, which is the corresponding CL secret key. In the *issue* protocol, the user generates m random messages $X = (x_1, x_2, \ldots, x_m)$, performs the BlindSign protocol with the vendor to obtain a CL signature (v, e, s) for the m messages without revealing anything about the messages. The user's m-redeemable coupon is (X, v, e, s). In the *redeem* protocol, the user reveals a message x_i to the vendor without revealing i and the vendor checks that x_i has not been revealed previously and stores x_i. The user then runs a zero-knowledge proof of knowledge of m messages X and their CL signature (v, e, s) and also proves that $x_i \in X$ without revealing i. A component coupon of the multi-coupon is redeemed if the vendor accepts these proofs as valid. The component coupon can not be redeemed again as the vendor has stored x_i. The protocols and proofs are zero-knowledge and X is the user's secret, hence, the redeem execution is unlinkable. The user is required to know (X, v, e, s) to perform the redeem protocol, so the coupon (X, v, e, s) has all-or-nothing-sharing property.

3 A New Privacy-Protecting Coupon System

3.1 Overview

We first outline general construction of our scheme and compare it with the CESSS05 system. As in the CESSS05 system, participants include a vendor and many users, and there is an Initialisation algorithm, an Issue protocol and a Redeem protocol, but there is one more algorithm, Terminate, which allows the vendor to terminate coupons.

The Initialisation algorithm generates a system public key and a vendor secret key. The vendor secret key consists of a CL-SDH secret key and a BB secret key. The system public key includes the corresponding CL-SDH and BB public keys, m random messages $a_1, a_2, \ldots, a_m$ and their BB signatures $o_1, o_2, \ldots, o_m$. Some other signature schemes, such as CL-SDH or those proposed in [2, 9], can be used in place of the BB scheme, but the BB scheme is used for the sake of efficiency and simplicity.

In the Issue protocol, the user generates a random x and runs the BlindSign-SDH with the vendor to obtain a CL-SDH signature (v, d, s) for x and the coupon is (x, v, d, s). So the issuing costs do not depend on the coupon bound m. In contrast, in the issue protocol of the CESSS05 system, the user and the vendor perform the BlindSign protocol to obtain a CL signature for m random messages, so the numbers of exponentiations and transmitted bytes are linear to m.

In the Redeem protocol, the user chooses a message a_i with a BB signature o_i and reveals $C = f_{a_i}(x)$ to the vendor without revealing a_i, where f is a one-way function. The vendor checks that C has not been revealed previously and stores C. The user then shows a zero-knowledge proof of knowledge of a BB message-signature pair (a, o) and a CL-SDH message-signature pair $(x, (v, d, s))$ such that $C = f_a(x)$. The coupon is redeemed if the vendor successfully verifies the proof. So the redeeming costs do not depend on m, whereas in previous PPC schemes these costs depend on m. One important issue is that, in order to protect privacy of the coupon system, the one-way function f must be "unlinkable". That means it is hard to distinguish between $(f_{a_i}(x), f_{a_j}(x))$ and $(f_{a_i}(x), f_{a_j}(x'))$. Fortunately, the DBDHI assumption [3] can be employed to construct such a one-way function that can be efficiently used for our scheme. The function is the same as a recent verifiable random function [14].

We use the "accumulating" approach [7, 4] for coupon revocation. To terminate a coupon, the vendor recomputes the system public key and adds some information to a public revocation list (RL) such that unrevoked coupons can be efficiently updated to be valid under the new system public key but it is hard to update the terminated coupon. This approach can also be used to provide coupon revocation for the CESSS05 system.

The CESSS05 system has not been presented in details, so we can not provide a thorough efficiency comparison with this system. Even if m is small, the numbers of transmitted bytes and exponentiations in our scheme are quite smaller than those in the CESSS05 scheme. Our scheme requires a number of pairings, but most of them can be pre-computed before the protocols: $e(g_1', g_2')$, $e(g_1, g_2)$, $e(b, g_2)$, $e(b, w)$, $e(b, g_2')$, $e(b, w')$ and $e(c, g_2)$. The user $\mathcal{U}$ can compute $e(v, wg_2^d)$, $e(g_1^x b^s c, g_2)$, $e(T_1, g_2)$ and $e(T_2, g_2')$ without computing any pairing online if $e(v, w)$, $e(v, g_2)$, $e(g_1, g_2)$, $e(b, g_2)$, $e(c, g_2)$, $e(o_i, g_2')$ and $e(b, g_2')$ are pre-computed. So only four pairings are needed to be computed online by the verifier in the PKSign Proof: $e(T_1, g_2)$, $e(T_1, w)$, $e(T_2, g_2')$ and $e(T_2, w')$.

3.2 Description

The system consists of an *Initialisation* algorithm, a *Terminate* algorithm, an *Issue* protocol and a *Redeem* protocol and involves a *vendor* $\mathcal{V}$ and a number of *users*.

Initialisation.
Let $\mathbb{G}_1$, $\mathbb{G}_2$, p, g_1 and g_2 be bilinear mapping parameters. Generate random $b, c, g_1' \in_R \mathbb{G}_1$, $g_2' \in_R \mathbb{G}_2$ and $\gamma, \gamma' \in_R \mathbb{Z}_p^*$ and compute $w \leftarrow g_2^{\gamma}$ and $w' \leftarrow {g_2'}^{\gamma'}$. Generate different $a_1, a_2, \ldots, a_m \in_R \mathbb{Z}_p \setminus \{-\gamma\}$ and their BB signatures $o_1, o_2, \ldots, o_m \in \mathbb{G}_1$ where $o_i = {g_1'}^{1/(\gamma' + a_i)}$, $(i = 1 \ldots m)$. The system public key is $PK = (A, R, b, c, g_1, g_2, w, g_1', g_2', w')$ and the vendor secret key is $SK = (\gamma, \gamma')$, where $A = (a_1, a_2, \ldots, a_m)$ and $R = (o_1, o_2, \ldots, o_m)$. There is a public Revocation List RL, which is empty initially.

Issue.
The vendor $\mathcal{V}$ and a user $\mathcal{U}$ perform this protocol. $\mathcal{U}$ chooses a random $x \in_R \mathbb{Z}_p^*$ and runs the *BlindSign-SDH* protocol with $\mathcal{V}$ to obtain a blind CL-SDH signature (v, d, s) on x (the CL-SDH public key is (b, c, g_1, g_2, w) and the CL-SDH secret key is γ). The user's multi-coupon is $M = (x, v, d, s)$, where v and d are known by $\mathcal{V}$ and x and s are $\mathcal{U}$'s secrets.

Terminate.
Suppose the current public key is $PK = (A, R, b, c, g_1, g_2, w, g_1', g_2', w')$, $\mathcal{V}$ terminates a coupon $(\cdot, \cdot, d_1, \cdot)$ by computing $\bar{g}_1 \leftarrow g_1^{1/(\gamma + d_1)}$, $\bar{g}_2 \leftarrow g_2^{1/(\gamma + d_1)}$, $\bar{b} \leftarrow b^{1/(\gamma + d_1)}$, $\bar{c} \leftarrow c^{1/(\gamma + d_1)}$ and $\bar{w} \leftarrow (\bar{g}_2)^{\gamma}$, setting the new public key $PK = (A, R, \bar{b}, \bar{c}, \bar{g}_1, \bar{g}_2, \bar{w}, g_1', g_2', w')$, and adding $(d_1, \bar{g}_2, \bar{b}, \bar{c})$ to RL.

Redeem.
This is a protocol between the vendor $\mathcal{V}$ and a user $\mathcal{U}$. It consists of 2 stages, Updating and Proof of Signatures.

Updating:
In this stage, $\mathcal{U}$ updates the system public key and his unrevoked coupon, as the vendor may have terminated some coupons and changed the system public key.

$\mathcal{U}$ obtains RL and suppose $(d_1, g_1^*, b_1^*, c_1^*), (d_2, g_2^*, b_2^*, c_2^*), \ldots, (d_k, g_k^*, b_k^*, c_k^*)$ are the new items on RL (in that order) since the last time $\mathcal{U}$ updated the public key and his coupon. $\mathcal{U}$ can update the current public key and his coupon by repeating the following process for each of these items.

For terminated item $(d_1, g_1^*, b_1^*, c_1^*)$, $\mathcal{U}$ (or anyone) can simply compute a new public key from the old public key $PK = (A, R, b, c, g_1, g_2, w, g_1', g_2', w')$ as

$\hat{g}_1 \leftarrow \psi(g_1^*)$, $\hat{g}_2 \leftarrow g_1^*$, $\hat{b} \leftarrow b_1^*$, $\hat{c} \leftarrow c_1^*$ and $\hat{w} \leftarrow g_2(g_1^*)^{-d_1}$. Then $\hat{g}_1 = g_1^{1/(\gamma+d_1)}$ and $\hat{w} = (g_1^*)^{\gamma+d_1}(g_1^*)^{-d_1} = \hat{g}_2^{\gamma}$. The new public key is $PK = (A, R, \hat{b}, \hat{c}, \hat{g}_1, \hat{g}_2, \hat{w}, g_1', g_2', w')$.

$\mathcal{U}$ then updates his unrevoked coupon (x, v, d, s) by computing $v^* \leftarrow \psi(g_1^*)^x(b_1^*)^s c_1^*$ and $\hat{v} \leftarrow (v^*/v)^{1/(d-d_1)}$. Then $v^* = v^{(\gamma+d)/(\gamma+d_1)}$ and $\hat{v}^{\gamma+d} = (v^{(\gamma+d)/(\gamma+d_1)}/v)^{(\gamma+d)/(d-d_1)} = v^{(\gamma+d)/(\gamma+d_1)} = \psi(g_1^*)^x(b_1^*)^s c_1^*$. So the updated coupon $(x, \hat{v}, d, s)$ is valid.

By orderly repeating the process for each of the items $(d_1, g_1^*, b_1^*, c_1^*)$, $(d_2, g_2^*, b_2^*, c_2^*)$, ..., $(d_k, g_k^*, b_k^*, c_k^*)$, the user can update the current public key and his unrevoked coupon.

Proof of Signatures:
Suppose the updated system public key is $PK = (A, R, b, c, g_1, g_2, w, g_1', g_2', w')$ and $\mathcal{U}$'s updated coupon is (x, v, d, s).

$\mathcal{U}$ sends $h = f_{a_i}(x)$ to the vendor $\mathcal{V}$ where $f_{a_i}(x) = e(g_1', g_2')^{1/(x+a_i)}$. $\mathcal{V}$ checks if h has been revealed before (from its storage S of these values). If it has, $\mathcal{V}$ stops the protocol and output *reject*. Otherwise, $\mathcal{V}$ stores h in S. Then $\mathcal{U}$ shows $\mathcal{V}$ the following proof

$$\begin{aligned} PKSign = PK\{(\alpha, \tau, \beta, \delta, \epsilon, \varepsilon) : e(\delta, wg_2^{\epsilon}) = e(g_1^{\beta} b^{\varepsilon} c, g_2) \\ \wedge e(\tau, w' g_2'^{\alpha}) = e(g_1', g_2') \wedge h^{\alpha+\beta} = e(g_1', g_2')\} \end{aligned}$$

PKSign proves that $\mathcal{U}$ knows a CL-SDH message-signature pair $(x, (v, d, s))$ and a BB message-signature pair (a, o) such that $h = e(g_1', g_2')^{1/(x+a)}$. *PKSign* is presented in details in the next subsection.

3.3 PKSign Proof

The following PKSign proof between $\mathcal{U}$ and $\mathcal{V}$ is an honest-verifier zeroknowledge proof under the Discrete Log assumption in $\mathbb{G}_1$ (its proof is standard and so omitted).

- *Common input*: $PK = (A, R, b, c, g_1, g_2, w, g_1', g_2', w')$; $h = e(g_1', g_2')^{1/(x+a_i)}$.
- *$\mathcal{U}$'s input*: a_i, o_i, x, v, d, s.
- *Proof*: $PK\{(\alpha, \tau, \beta, \delta, \epsilon, \varepsilon)$:
  $e(\delta, wg_2^{\epsilon}) = e(g_1^{\beta} b^{\varepsilon} c, g_2) \wedge e(\tau, w' g_2'^{\alpha}) = e(g_1', g_2') \wedge h^{\alpha+\beta} = e(g_1', g_2')\}$.

Protocol:
$\mathcal{U}$ generates random $t_1, t_2, u_1, u_2 \in_R \mathbb{Z}_p$ and computes
$y_1 \leftarrow t_1 d$, $y_2 \leftarrow t_2 a_i$,
$T_1 \leftarrow v b^{t_1}$, $U_1 \leftarrow b^{t_1} c^{u_1}$, $T_2 \leftarrow o_i b^{t_2}$, $U_2 \leftarrow b^{t_2} c^{u_2}$.

$\mathcal{U}$ and $\mathcal{V}$ then perform a proof of knowledge of $(t_1, t_2, u_1, u_2, y_1, y_2, a_i, x, d, s)$ satisfying
$b^{t_1} c^{u_1} = U_1$, $b^{t_2} c^{u_2} = U_2$,
$e(g_1, g_2)^x e(b, g_2)^{s+y_1} e(T_1, g_2)^{-d} e(b, w)^{t_1} = e(T_1, w) e(c, g_2)^{-1}$,
$e(T_2, g_2')^{-a_i} e(b, w')^{t_2} e(b, g_2')^{y_2} = e(T_2, w') e(g_1', g_2')^{-1}$,

$h^{x+a_i} = e(g_1', g_2')$,
$U_1^d b^{-y_1} c^{-u_1 d} = 1$, $U_2^{a_i} b^{-y_2} c^{-u_2 a_i} = 1$.

It proceeds as follows. $\mathcal{U}$ generates random
$r_{t_1}, r_{t_2}, r_{u_1}, r_{u_2}, r_{y_1}, r_{y_2}, r_{a_i}, r_x, r_d, r_s, r_{u_1 d}, r_{u_2 a_i} \in_R \mathbb{Z}_p$
and computes

$R_1 \leftarrow b^{r_{t_1}} c^{r_{u_1}}$, $R_2 \leftarrow b^{r_{t_2}} c^{r_{u_2}}$,
$R_3 \leftarrow e(g_1, g_2)^{r_x} e(b, g_2)^{r_s + r_{y_1}} e(T_1, g_2)^{-r_d} e(b, w)^{r_{t_1}}$,
$R_4 \leftarrow e(T_2, g_2')^{-r_{a_i}} e(b, w')^{r_{t_2}} e(b, g_2')^{r_{y_2}}$,
$R_5 \leftarrow h^{r_x + r_{a_i}}$,
$R_6 \leftarrow U_1^{r_d} b^{-r_{y_1}} c^{-r_{u_1 d}}$, $R_7 \leftarrow U_2^{r_{a_i}} b^{-r_{y_2}} c^{-r_{u_2 a_i}}$.

$\mathcal{U}$ sends $(T_1, T_2, U_1, U_2, R_1, \ldots, R_7)$ to $\mathcal{V}$. $\mathcal{V}$ returns a challenge $\mu \in_R \mathbb{Z}_p$. $\mathcal{U}$ then responses with the following values

$z_{t_1} \leftarrow r_{t_1} + \mu t_1$, $z_{t_2} \leftarrow r_{t_2} + \mu t_2$, $z_{u_1} \leftarrow r_{u_1} + \mu u_1$, $z_{u_2} \leftarrow r_{u_2} + \mu u_2$,
$z_{y_1} \leftarrow r_{y_1} + \mu y_1$, $z_{y_2} \leftarrow r_{y_2} + \mu y_2$,
$z_{a_i} \leftarrow r_{a_i} + \mu a_i$, $z_x \leftarrow r_x + \mu x$, $z_d \leftarrow r_d + \mu d$, $z_s \leftarrow r_s + \mu s$
$z_{u_1 d} \leftarrow r_{u_1 d} + \mu u_1 d$, $z_{u_2 a_i} \leftarrow r_{u_2 a_i} + \mu u_2 a_i$.

$\mathcal{V}$ finally verifies that

$b^{z_{t_1}} c^{z_{u_1}} = U_1^{\mu} R_1$, $b^{z_{t_2}} c^{z_{u_2}} = U_2^{\mu} R_2$,
$e(g_1, g_2)^{z_x} e(b, g_2)^{z_s + z_{y_1}} e(T_1, g_2)^{-z_d} e(b, w)^{z_{t_1}} = (e(T_1, w) e(c, g_2)^{-1})^{\mu} R_3$,
$e(T_2, g_2')^{-z_{a_i}} e(b, w')^{z_{t_2}} e(b, g_2')^{z_{y_2}} = (e(T_2, w') e(g_1', g_2')^{-1})^{\mu} R_4$,
$h^{z_x + z_{a_i}} = e(g_1', g_2')^{\mu} R_5$,
$U_1^{z_d} b^{-z_{y_1}} c^{-z_{u_1 d}} = R_6$, $U_2^{z_{a_i}} b^{-z_{y_2}} c^{-z_{u_2 a_i}} = R_7$.

$\mathcal{V}$ *accepts* if and only if all equations hold.

3.4 Remark

- By removing the Terminate algorithm and the Updating stage of the Redeem protocol, we have a PPC system without coupon revocation that has the same functionality as the CESSS05 system.
- Both of the CESSS05 system and our system can be modified to allow the vendor to assign different redeeming bounds to different coupons, where the bounds are not greater than some value m. For instance, the vendor $\mathcal{V}$ wants to issue n-redeemable coupon to a user $\mathcal{U}$ where $n \leq m$. In the Issue protocol of the CESSS05 system, $\mathcal{U}$ also reveals $x_{n+1}, \ldots, x_m$ to $\mathcal{V}$ with a zero-knowledge proof that $x_{n+1}, \ldots, x_m \in X$. In the Issue protocol of our system, $\mathcal{U}$ also reveals $e(g_1', g_2')^{1/(x+a_{n+1})}, \ldots, e(g_1', g_2')^{1/(x+a_m)}$ with a zero-knowledge proof of knowledge of x corresponding to (v, d, s).

4 Security Model

4.1 Syntax

A Privacy-Protecting Coupon System is a tuple of polynomial-time algorithms (*Init*, *Issue*^{U}, *Issue*^{V}, *Terminate*, *Redeem*^{U}, *Redeem*^{V}). Participants include a *vendor* and a number of *users*. There is a public *Revocation List* (RL) which consists of terminated coupons and is empty initially.

- *Init*: The *initialisation* algorithm Init on input 1^k returns a pair (PK, SK) where PK is the system public key and SK is the vendor secret key.
- (*Issue*^{U}, *Issue*^{V}): These interactive algorithms form the *issue* protocol between a user (Issue^U) and the vendor (Issue^V). The common input is PK and Issue^V also takes SK as the private input. Issue^U outputs a coupon M and Issue^V returns the coupon's public information d_M.
- *Terminate*: This *termination* algorithm takes as input PK, SK, RL and a coupon's public information d. It outputs a new pair $(\bar{PK}, \bar{SK})$ and appends (d, inf) to RL where inf is some public information about this termination. The new system public key is $\bar{PK}$ and the new vendor secret key is $\bar{SK}$. Coupon d can no longer be redeemed afterwards.
- (*Redeem*^{U}, *Redeem*^{V}): These interactive algorithms form the *redeem* protocol which allows a user (Issue^U) to redeem a coupon with the vendor (Issue^V). The common input includes the current public key PK and the revocation list RL. The vendor's private input is the current secret key SK and the user's private input is its coupon M. At the end, Issue^V returns *accept* or *reject*, which indicates if the coupon is redeemed or it is invalid, respectively. Issue^U outputs $\bar{M}$ which is a coupon updated from M.

Correctness: Informally, it requires that if a coupon is obtained by correctly performing the issue protocol, has not been redeemed more than its bound and has not been terminated, then an execution of the redeem protocol for the coupon shall end successfully with the vendor outputting accept, with overwhelming probability.

4.2 Oracles

Security requirements of PPC systems are formulated in experiments between an adversary and an honest party. In each experiment, the adversary plays a vendor and the party plays a user or vice versa. The adversary's capabilities are modelled by access to the following oracles.

$\mathcal{O}_{Is}(\cdot,\cdot)$: The adversary can request this *issue* oracle to execute the Issue protocol. The oracle takes 2 inputs and the first input can be either $'user'$ or $'vendor'$. If the adversary plays a user, it does not need to call $\mathcal{O}_{Is}('user', \cdot)$, and if the adversary plays a vendor, it does not need to call $\mathcal{O}_{Is}('vendor', \cdot)$. If the first input is $'user'$, the oracle plays the honest user, takes the second input as the vendor's message of the Issue protocol and outputs a message back to the vendor. If the first input is $'vendor'$, the oracle plays the honest vendor, takes the second input as the user's message of the Issue protocol and outputs a message back to the user.

$\mathcal{O}_{Rd}(\cdot,\cdot,\cdot)$: This *redeem* oracle allows the adversary to run the Redeem protocol. It takes 3 input where the first input can be $'user'$, $'vendor'$ or $'correct'$, the second input is the public information of the coupon to be redeemed. If the adversary plays a user, it does not need to call $\mathcal{O}_{Rd}('user', \cdot, \cdot)$, and if the adversary plays a vendor, it does not need to call $\mathcal{O}_{Rd}('vendor', \cdot, \cdot)$. If the first input is

$'user'$, the oracle plays the honest user, takes the third input as the vendor's message of the Redeem protocol and outputs a message back to the vendor. If the first input is $'vendor'$, the oracle plays the honest vendor, takes the third input as the user's message of the Redeem protocol and outputs a message back to the user. If the first input is $'correct'$, the oracle just correctly executes the Redeem protocol on the coupon and outputs either accept or reject to indicate whether the vendor accepts the coupon or not.

$\mathcal{O}_{Op}(\cdot)$: On input a coupon's public information, this *open* oracle returns the coupon's content.

$\mathcal{O}_{Te}(\cdot)$: This *terminate* oracle takes as input a coupon's public information and terminates the coupon.

$\mathcal{O}_{Ch}(\cdot,\cdot)$ This *challenge* oracle takes as input 2 coupons with public information d_0 and d_1. It randomly chooses a bit j, correctly runs the Redeem protocol on the coupon d_j and returns the transcript.

$\mathcal{O}_{Co}(\cdot)$ This *count* oracle takes as input either $'total'$ or a coupon's public information. If it is $'total'$, the oracle returns the total number of successful redemptions. If it is a coupon's public information d, the oracle returns the number of times d has been successfully redeemed. In case d has been terminated, the oracle outputs the number of successful redemptions by using d before it was terminated.

4.3 Security Notions

A PPC system is secure if it satisfies 3 security requirements: Unlinkability, Unforgeability and Unsplittability. Each requirement is defined by an experiment between a PPT adversary and an honest party. Suppose m is each coupon's redeemable number.

Unlinkability.
Loosely stated, Unlinkability requires that, given 2 coupons and a redeeming transcript generated by using one of the coupons, the adversary can not decide which coupon has been used. In the Unlinkability experiment, the adversary $\mathcal{A}$ plays a vendor $\mathcal{V}$ and the party plays an honest user $\mathcal{U}$. The adversary can access oracles $\mathcal{O}_{Is}$, $\mathcal{O}_{Rd}$, $\mathcal{O}_{Op}$, $\mathcal{O}_{Ch}$, $\mathcal{O}_{Co}$. $\mathcal{O}_{Te}$ is not needed as $\mathcal{A}$ can terminate any coupon.

The adversary first runs the Init algorithm on input 1^k to obtain a key pair $(PK, SK) \leftarrow \mathsf{Init}(1^k)$. The adversary can then query $\mathcal{O}_{Is}$ to issue many coupons to $\mathcal{U}$, query $\mathcal{O}_{Rd}$ to redeem any coupon, query $\mathcal{O}_{Op}$ to open any coupon, query $\mathcal{O}_{Co}$ to obtain the number of successful redemptions totally or for any coupon, and terminate any coupon. At some point, the adversary query $\mathcal{O}_{Is}$ twice to issue $\mathcal{U}$ two coupons with public information d_0 and d_1. The adversary then continues the experiment as before, except that it can not query $\mathcal{O}_{Op}$ on d_0 and d_1 and can not terminate d_0 and d_1. After a while, the adversary query $\mathcal{O}_{Ch}(d_0, d_1)$ to obtain a challenge transcript, which has been generated by $\mathcal{O}_{Ch}$ using a random bit j. It then continues the experiment as before, except that it now can not query $\mathcal{O}_{Co}$ and $\mathcal{O}_{Op}$ on d_0 and d_1. The adversary finally output a bit j'. At this point, it is required that $\mathcal{O}_{Co}(d_0), \mathcal{O}_{Co}(d_1) < m$. Let

$$\mathbf{Adv}_{\mathcal{A}}^{unli}(k) = |\mathsf{Pr}(j' = 0 \mid j = 0) - \mathsf{Pr}(j' = 0 \mid j = 1)|$$

Definition 1. *A PPC system is said to be unlinkable if $\boldsymbol{Adv}_{\mathcal{A}}^{unli}(k)$ is negligible for any PPT adversary $\mathcal{A}$.*

Unforgeability.
Intuitively, Unforgeability requires that the adversary can not forge any successful redemption, i.e. it can not forge a new valid coupon or successfully redeem an overspent or terminated coupon. In the Unforgeability experiment, the adversary $\mathcal{A}$ plays a user $\mathcal{U}$ and the party plays an honest vendor $\mathcal{V}$. The adversary can access oracles $\mathcal{O}_{Is}$, $\mathcal{O}_{Rd}$, $\mathcal{O}_{Te}$, $\mathcal{O}_{Co}$. $\mathcal{O}_{Op}$ is not needed as $\mathcal{U}$ obtains all coupons issued from $\mathcal{V}$ ($\mathcal{A}$ actually represents all users).

The party first runs the Init algorithm on input 1^k to obtain a key pair $(PK, SK) \leftarrow \mathsf{Init}(1^k)$ and publish PK. The adversary can then query $\mathcal{O}_{Is}$ to obtain many coupons from $\mathcal{V}$, query $\mathcal{O}_{Rd}$ to redeem any coupon, query $\mathcal{O}_{Te}$ to terminate any coupon and query $\mathcal{O}_{Co}$ to obtain the number of successful redemptions totally or for any coupon. At the end, suppose $\mathcal{A}$ has obtained l unrevoked coupons from the vendor. If $\mathcal{O}_{Co}('total') > m \times l + \sum_{d \in RL} \mathcal{O}_{Co}(d)$, then the adversary is considered to be successful. Let $\mathbf{Adv}_{\mathcal{A}}^{unfo}(k)$ denote the probability that the adversary is successful.

Definition 2. *A PPC system is said to be unforgeable if $\boldsymbol{Adv}_{\mathcal{A}}^{unfo}(k)$ is negligible for any PPT adversary $\mathcal{A}$.*

Unsplittability.
Unsplittability, i.e. all-or-nothing-sharing, intuitively means that if a user can spend a coupon once, then he can spend it for m times. We do not model the complete protection against splitting a coupon (Strong Unsplittability) as neither CESSS05 nor our scheme satisfies this requirement. In the Unsplittability experiment, the adversary $\mathcal{A}$ plays a user $\mathcal{U}$ and the party plays an honest vendor $\mathcal{V}$. The adversary can access oracles $\mathcal{O}_{Is}$, $\mathcal{O}_{Rd}$, $\mathcal{O}_{Te}$, $\mathcal{O}_{Co}$. As for Unforgeability, $\mathcal{O}_{Op}$ is not needed as $\mathcal{U}$ obtains all coupons issued from $\mathcal{V}$.

The party first runs the Init algorithm on input 1^k to obtain a key pair $(PK, SK) \leftarrow \mathsf{Init}(1^k)$ and publish PK. The adversary can then query $\mathcal{O}_{Is}$ to obtain many coupons from $\mathcal{V}$, query $\mathcal{O}_{Rd}$ to redeem any coupon, query $\mathcal{O}_{Te}$ to terminate any coupon and query $\mathcal{O}_{Co}$ to obtain the number of successful redemptions totally or for any coupon. At some point, $\mathcal{A}$ outputs a coupon with public information d and terminates. At that time, if $0 < \mathcal{O}_{Co}(d) < m$, d is not in RL and $\mathsf{reject} \leftarrow \mathcal{O}_{Rd}('correct', d, \emptyset)$, then the adversary is considered to be successful. Let $\mathbf{Adv}_{\mathcal{A}}^{unsp}(k)$ denote the probability that the adversary is successful.

Definition 3. *A PPC system is said to be unsplittable if $\boldsymbol{Adv}_{\mathcal{A}}^{unsp}(k)$ is negligible for any PPT adversary $\mathcal{A}$.*

4.4 Remarks

This model can be simplified for PPC systems without coupon termination by removing the Terminate algorithm, the revocation list RL and $\mathcal{O}_{Te}$.

The above requirements are strong enough to capture the informal requirements listed in [13]. *Minimum disclosure*, which means the vendor should not learn from the Redeem protocol how many times more the coupon can be redeemed, follows from the *unlinkability* requirement. *Unforgeability* implies *double spending detection* and *redemption limitation*, which means an m-redeemable coupon should not be accepted more than m times. And *unsplittability* means *all-or-nothing-sharing.*

5 Security of the Privacy-Protecting Coupon System

Correctness can be easily checked and proofs of the security requirements are quite routine using approaches in [2, 3, 18]. Due to space limitation, we only provide sketches of the security proofs.

Theorem 2. *The PPC scheme provides unlinkability if the DBDHI assumption holds.*

Proof Sketch. We show that if a PPT adversary $\mathcal{A}$ can break the unlinkability property of the PPC system, then we can construct a PPT adversary $\mathcal{B}$ that can break the DBDHI assumption. Suppose $\mathbb{G}_1$, $\mathbb{G}_2$, p, $\tilde{g}_1$ and $\tilde{g}_2$ are bilinear mapping parameters. Suppose a tuple $\theta = (\tilde{g}_1, \tilde{g}_2, \tilde{g}_2^{\vartheta}, \ldots, \tilde{g}_2^{(\vartheta^q)}, \Theta)$ is uniformly chosen from one of the sets $S_0 = \{(\tilde{g}_1, \tilde{g}_2, \tilde{g}_2^{\nu}, \ldots, \tilde{g}_2^{(\nu^q)}, e(\tilde{g}_1, \tilde{g}_2)^{1/(\nu)})|\nu \in_R \mathbb{Z}_p^*\}$ and $S_1 = \{(\tilde{g}_1, \tilde{g}_2, \tilde{g}_2^{\nu}, \ldots, \tilde{g}_2^{(\nu^q)}, \Lambda)|\nu \in_R \mathbb{Z}_p^*, \Lambda \in_R \mathbb{G}_T^*\}$. To decide whether θ is chosen from S_0 or S_1, $\mathcal{B}$ simulates the Unlinkability experiment with $\mathcal{A}$ where $\mathcal{B}$ plays the honest party and provides oracles.

$\mathcal{B}$ selects a random bit $j \leftarrow \{0, 1\}$ and let j' be the other bit. From θ, $\mathcal{B}$ can construct $H_j = \{e(g_1', g_2')^{1/(x_j + a_1)}, \ldots, e(g_1', g_2')^{1/(x_j + a_{m-1})}\}$ and $\bar{\Theta}$ for some g_1', g_2', x_j and $\{a_i\}_{i=1}^m$, where $m < q$ and x_j is $\mathcal{B}$'s only unknown value, such that $\bar{\Theta} = e(g_1', g_2')^{1/(x_j + a_m)}$ if and only if $\Theta = e(\tilde{g}_1, \tilde{g}_2)^{1/\vartheta}$. $\mathcal{B}$ then generates $x_{j'} \in_R \mathbb{Z}_p^*$ and computes $H_{j'} = \{e(g_1', g_2')^{1/(x_{j'} + a_1)}, \ldots, e(g_1', g_2')^{1/(x_{j'} + a_m)}\}$. $\mathcal{B}$ then generates random $b, c, g_1 \in_R \mathbb{G}_1$, $g_2 \in_R \mathbb{G}_2$ and $\gamma, \gamma' \in_R \mathbb{Z}_p^*$ and computes $w \leftarrow g_2^{\gamma}$ and $w' \leftarrow g_2'^{\gamma'}$. For $a_1, a_2, \ldots, a_m$, $\mathcal{B}$ computes their BB signatures $o_1, o_2, \ldots, o_m \in \mathbb{G}_1$ where $o_i = g_1'^{1/(\gamma' + a_i)}$, $(i = 1 \ldots m)$. The system public key is $PK = (A, R, b, c, g_1, g_2, w, g_1', g_2', w')$ and the vendor secret key is $SK = (\gamma, \gamma')$, where $A = (a_1, a_2, \ldots, a_m)$ and $R = (o_1, o_2, \ldots, o_m)$. As knowing γ, γ' and playing the honest user, $\mathcal{B}$ can easily simulate oracles $\mathcal{O}_{Is}$, $\mathcal{O}_{Op}$ and $\mathcal{O}_{Co}$ and a pair of challenge coupons (x_0, v_0, d_0, s_0) and (x_1, v_1, d_1, s_1) for $\mathcal{O}_{Ch}$.

For a query to $\mathcal{O}_{Rd}$ from $\mathcal{A}$ on d_0 or d_1, $\mathcal{B}$ can choose h from either H_0 or H_1, respectively and simulates the $PKSign$ proof (as $PKSign$ is zero-knowledge). For the query to $\mathcal{O}_{Ch}$, $\mathcal{B}$ uses $\bar{\Theta}$ as h to simulate $PKSign$. Finally, if $\mathcal{A}$ returns j, $\mathcal{B}$ decides that θ is chosen from S_0. Otherwise, $\mathcal{B}$ decides that θ is chosen from

S_1. Therefore, if $\mathcal{A}$ can break the unlinkability property of the PPC system, then $\mathcal{B}$ can break the DBDHI assumption.

Theorem 3. *The PPC scheme provides unforgeability if the SDH assumption holds.*

Proof Sketch. We show that if a PPT adversary $\mathcal{A}$ can break the unforgeability property of the PPC system, then we can construct a PPT adversary $\mathcal{B}$ that can break the SDH assumption. As $\mathcal{A}$ breaks unforgeability, one of the 3 cases below happens. $\mathcal{B}$ will randomly play one of 3 corresponding games such that if $\mathcal{B}$ plays a game when the corresponding case happens, then $\mathcal{B}$ can break the SDH assumption.

As PKSign is zero-knowledge, if it is accepted (when a coupon is successfully redeemed), $\mathcal{A}$ knows (a, o, x, v, d, s) satisfying the equations in PKSign. Following the Unforgeability experiment, if $\mathcal{A}$ is successful ($\mathcal{O}_{Co}('total') > m \times l + \sum_{d \in RL} \mathcal{O}_{Co}(d)$), there are 3 cases.

- $\mathcal{A}$ can generate a new BB message-signature pair $(a_{m+1}, o_{m+1}) \notin \{(a_1, o_1), (a_2, o_2), \ldots, (a_m, o_m)\}$. If $\mathcal{B}$ plays a game similar to the game in the proof of BB's unforgeability under a weak chosen message attack [2], then $\mathcal{B}$ can break the SDH assumption.
- $\mathcal{A}$ can redeem a revoked coupon. If $\mathcal{B}$ plays a game constructed using the approach of proofs in [18], then $\mathcal{B}$ can break the SDH assumption.
- $\mathcal{A}$ can forge a new CL-SDH message-signature pair $x, (v, d, s)$. If $\mathcal{B}$ plays a game similar to the game in the proof of CL-SDH's unforgeability under a chosen message attack, then $\mathcal{B}$ can break the SDH assumption.

Theorem 4. *The PPC scheme provides unsplittability.*

Proof Sketch. Following the Unsplittability experiment, suppose the adversary outputs a coupon (x, v, d, s) at the end, such that it has been successfully redeemed at least once but less than m times and it has not been terminated. That means there are elements of $\{(a_1, o_1), (a_2, o_2), \ldots, (a_m, o_m)\}$ which have not been used by the coupon. So if $\mathcal{O}_{Rd}$ is queried with input $'correct'$ on the coupon, one of the unused elements can be used and $\mathcal{O}_{Rd}$ outputs accept. Therefore, the adversary is not successful and it indicates unsplittability.

Acknowledgements: Thanks go to Peter Lamb, who contributed to this paper but declined to be a co-author.

References

1. M. Bellare, H. Shi, and C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. CT-RSA 2005, Springer-Verlag, LNCS 3376, pp. 136-153.
2. D. Boneh, and X. Boyen. Short Signatures Without Random Oracles. Eurocrypt 2004, Springer-Verlag, LNCS 3027, pp. 56-73.
3. D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. Eurocrypt 2004, Springer-Verlag, LNCS 3027, pp. 223-238.

4. D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. Crypto 2004, Springer-Verlag, LNCS 3152, pp. 41-55.
5. S. Brands. An Efficient Off-line Electronic Cash System Based On The Representation Problem. Technical Report CS-R9323, Centrum voor Wiskunde en Informatica, 1993.
6. S. Brands. Rethinking Public Key Infrastructure and Digital Certificates . Building in Privacy. PhD thesis, Eindhoven Institute of Technology, The Netherlands, 1999.
7. J. Camenisch and A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. Crypto 2002, Springer-Verlag, LNCS 2442, pp. 61-76.
8. J. Camenisch and A. Lysyanskaya. A Signature Scheme with Efficient Protocols. SCN 2002, Springer-Verlag, LNCS 2576.
9. J. Camenisch and A. Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. CRYPTO'04, Springer-Verlag, LNCS 3152, 2004.
10. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. Crypto 1997, Springer-Verlag, LNCS 1296, pp. 410-424.
11. D. Chaum. Blind signature system. Crypto 1983, Plenum Press, pp. 153-153.
12. D. Chaum. Privacy protected payments: Unconditional payer and/or payee untraceability. Smart Card 2000, Proceedings. North Holland, 1989.
13. L. Chen, M. Enzmann, A. Sadeghi, M. Schneider, and M. Steiner. A Privacy-Protecting Coupon System. Financial Cryptography 2005, Springer-Verlag, LNCS 3570, pp. 93-109.
14. Y. Dodis and A. Yampolskiy. A Verifiable Random Function with Short Proofs and Keys. Public Key Cryptography 2005, Springer-Verlag, LNCS 3386, pp. 416-431.
15. N. Ferguson. Extensions of single term coins. Adv. in Cryptology - CRYPTO '93, LNCS 773. Springer, 1993.
16. A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. Crypto 1986, Springer-Verlag, LNCS 263, pp. 186-194, 1986.
17. L. Nguyen. Efficient Dynamic k-Times Anonymous Authentication. VietCrypt 2006, LNCS, Springer-Verlag, 18 pages, 2006.
18. L. Nguyen. Accumulators from Bilinear Pairings and Applications. RSA Conference 2005, Cryptographers' Track (CT-RSA), Springer-Verlag, LNCS 3376, pp. 275-292, 2005.
19. L. Nguyen and R. Safavi-Naini. Dynamic k-Times Anonymous Authentication. Applied Cryptography and Network Security (ACNS) 2005, Springer-Verlag, LNCS 3531, 2005.
20. T. Okamoto and K. Ohta. Disposable zero-knowledge authentications and their applications to untraceable electronic cash. CRYPTO '89, LNCS 435. Springer, 1990.
21. P. Persiano, I. Visconti. An efficient and usable multi-show non-transferable anonymous credential system. Financial Cryptography, LNCS 3110. Springer, Feb. 2004.
22. G. Shaffer and Z. Zhang. Competitive coupon targeting. Marketing Science, 14(4), 1995.
23. I. Teranisi, J. Furukawa, and K. Sako. k-Times Anonymous Authentication. Asiacrypt 2004, Springer-Verlag, LNCS 3329, pp. 308-322.
24. E. Verheul. Self-blindable credential certificates from the weil pairing. Adv. in Cryptology - ASIACRYPT '01, LNCS 2248. Springer-Verlag, 2001.