Anonymous Transferable Conditional E-cash

Jiangxiao Zhang[1,2], Zhoujun Li[1,2], and Hua Guo[1]

[1] State Key Laboratory of Software Development Environment,
Beihang University, Beijing 100083, China
[2] Beijing Key Laboratory of Network Technology,
Beihang University, Beijing, China
orange_0092008@163.com, zhoujun.li@263.net, hguo@buaa.edu.cn

Abstract. We present the first anonymous transferable conditional e-cash system based on two recent cryptographic primitives, i.e., the Groth-Sahai(GS) proofs system and the commuting signatures, thus the unlinkability and anonymity of the user is obtained. We solve an open problem by dividing the deposit into two parts, so that the user is unlinkable in the transferrable protocol and the deposit protocol. Comparing the existing conditional e-cash, the size of the computation and communication of our scheme is constant.

Keywords: conditional e-cash, transferability, anonymity, Groth-Sahai proofs, commuting signatures.

1 Introduction

Conditional e-cash is introduced firstly by L. Shi [11], which allows a participant to spend cash bank-issued electronic coin based on the outcome in the future. If the outcome is not favorable to the payer, he loses and the payee can cash the electronic coin from the bank; otherwise, the payer cashes back e-cash. There are many applications of conditional e-cash. For example, outsourcing computations make sure that the worker has completed the computation and can retrieve the payment simultaneously. For another example, the prediction markets obtain the outcome based the prediction in the future and the betting systems, where many people can bet in an anonymous betting systems and obtain the electronic coin depending on the result in the future.

Conditional transferable e-cash consists of the payers (the payees) $\mathcal{U}_1, \mathcal{U}_2, \cdots, \mathcal{U}_n$, the bank $\mathcal{B}$, the judge $\mathcal{J}$ and the publisher $\mathcal{P}$. Firstly, the publisher publics two commitments about two event outcomes. The user $\mathcal{U}_1$ registers one of the two outcomes at the publisher. Secondly, he withdraws a coin *co* from the bank $\mathcal{B}$, and then spends the coin to the user $\mathcal{U}_2$. The payee $\mathcal{U}_2$ can spend the coin to the third user $\mathcal{U}_3$, or deposit the coin to the bank $\mathcal{B}$. When the publisher publishes the outcome, only one user can win the coin, and then the user cashes from the bank $\mathcal{B}$. The bank $\mathcal{B}$ checks the coin, and decides to exchange the coin for credit to the account of the user or announce the judge to recover the identity of the double spenders. Meanwhile, traditional transferable e-cash also

A.D. Keromytis and R. Di Pietro (Eds.): SecureComm 2012, LNICST 106, pp. 45–60, 2013.

allows the user to directly spend the e-cash to other ones without depositing the e-cash into the bank.

There are some differences between transferable conditional e-cash system and traditional transferable e-cash system [7,6,13,9,5,12]. Firstly, in the transferable conditional e-cash, the user anonymously spend the conditional e-cash, while in traditional e-cash, the merchant and the user must supply the identity in order to receive money and serves respectively. Secondly, in the transferable conditional e-cash, the payee can not spend the e-cash until the outcome of the condition is published and only if the outcome is favorable to the payee, while the user can spend an e-cash without any conditional in the traditional transferable e-cash. And last, the payer can cash the e-cash in case of an unfavorable to the payee outcome of the condition, but these can not also be done in the transferable e-cash. Thus, new tools need to be developed to construct the transferable conditional e-cash.

1.1 Related Results

E-cash is the digital equivalent of regular money. It has many properties, such as divisibility, transferability, conditionality, et. al. Transferability is one of the most important properties among these basic properties. Okamoto and Ohta [16,12] proposed two transferred e-cash systems, however their systems can only provide weak anonymity. Chaum and Pedersen [7] analyzed the size of the e-cash in the transferred e-cash system, and they claimed that it is impossible to transfer a coin without increasing its size. Next, Canard *et al.* [13] proposed an anonymous transferable e-cash system, and analyzed the anonymity [6] in transferred e-cash. To solve the problem about the size of the e-cash system, Fuchsbauer *et al.* [9] constructed the first practical transferred constant-size fair e-cash in the standard model. However, each user has to keep in memory the data associated to all past transactions to prove her innocence in case of a fraud. Moreover, the anonymity of all subsequent owners of a double-spent coin must be revoked.

The conditional e-cash is another application in e-cash. L. Shi [11] firstly introduced the definition of the conditional e-payments, where the payer can anonymously spend the e-cash to the payee or transfer the e-cash to another. However, the conditional e-payments is on-line and depends on the expensive cut-and-choose techniques.

M. Blanton [3] improved the efficiency of the conditional e-payments and formalized it by uisng zero-knowledge proof, CL signature and verifiable encryption. However the payer can recognize a coin he has already observed previously and also decide whether he has already owned the coin he is receiving. Moreover the deposit is not anonymous.

O. Blazy *et al.* [4] proposed an anonymously transferable e-cash, which is a traditional transferable e-cash. It achieves the optimal anonymity in the transferable e-cash, namely observe-then-receive full anonymity ($OtR - Fa$), spend-then-observe full anonymity ($StO - FA$) and spend-then-receive full anonymity ($StR - FA$).

J. Groth and A. Sahai [10] constructed the first efficient non-interactive proof systems in the standard model. It considers a large class of statements over bilinear groups. The witness indistinguishable guarantees that any adversary cannot distinguish the user uses which witness. Their randomizability allows us to improve the NIZK proofs.

G. Fuchsbauer [8] presented a system of commuting signatures and verifiable encryption. It allows one to encrypt a message and corresponding signature while preserving its public verifiability. Given a commitment, a signer can create a verifiably encrypted signature on the committed message.

1.2 Our Construction

We propose an anonymous transferable conditional e-cash based on Groth-Sahai proofs [10] and the commuting signatures [8]. In our paper, we firstly improve the commitments and the corresponding proofs by the commuting signatures, so the transferrable spending is unlinkable and the user is anonymous. Meanwhile, we divide the deposit protocol into two parts to obtain the anonymity of the user in the deposit protocol. Secondly, we introduce a publisher, who can commit two secret value for two outcomes, so the conditional e-cash is obtained. Finally, we compare the efficiency of our construction to that of [11] and [3]. our contribution is listed as follows:

- We solve an open problem, which the identity of payee is unlinkable in the conditional transfer and deposit protocol. Meanwhile, the payers cannot be linked to payees or to ongoing or past transactions.
- We present the first anonymous transferable condition e-cash.
- We compare the efficiency of our construction to that of [11] and [3], and show that the computation and communication is constant in our scheme.

1.3 Organization of the Paper

The rest of this paper is organized as follows. In Section 2, we describe the preliminaries on the various cryptographic tools and assumptions. Security model of the conditional e-cash is presented in Section 3. In Section 4, we give the general description. The main protocol is presented in Section 5. The security analysis is given in Section 6. In Section 7, we conclude the paper.

2 Preliminaries

In this section, we introduce the background knowledge that will be used for our scheme.

2.1 Bilinear Map

A pairing is a bilinear mapping from two group elements to a group element. Let $\hat{e}$ be a bilinear map such that $\hat{e} : G_1 \times G_2 \to G_3$ and the following holds.

- G_1, G_2 and G_3 are cyclic multiplicative groups of prime order p.
- Each element of G_1, G_2 has unique binary representation.
- The elements g, h generate G_1 and G_2 respectively.
- $\hat{e} : G_1 \times G_2$ is a non-degenerate bilinear map so $\hat{e}(g, h)$ generates G_3 and for all $a, b \in \mathbb{Z}_n$ we have $\hat{e}(ag, bh) = \hat{e}(g, h)^{ab}$.
- We can efficiently compute group operations, compute the bilinear map and decide membership.

2.2 Mathematical Assumptions

The security of our construction is based on the following existing mathematical assumptions, namely, the Symmetric External Diffie-Hellman(SXDH) [10] and the asymmetric double hidden strong Diffie-Hellman assumption(q-ADH-SDH) [1].

Definition 1. ***Symmetric External Diffie-Hellman****. Let G_1, G_2 be cyclic groups of prime order, g_1 and g_2 generate G_1 and G_2 respectively, and let $\hat{e} : G_1 \times G_2 \to G_3$ be a bilinear map. The Symmetric External Diffie-Hellman (SXDH) Assumption states that the DDH problem is hard in both G_1 and G_2 of a bilinear group pair (G_1, G_2), namely, we give $g_1, g_1^a, g_1^b \in G_1$ and $g_2, g_2^a, g_2^b \in G_2$, for random a, b, it is hard to distinguish g_1^{ab} and g_2^{ab} from G_1 and G_2 respectively. It implies that there is no efficiently computable isomorphism from G_2 to G_1 or vice versa.*

Definition 2. ***q-ADH-SDH****. Let $g, f, k \in G_1, h \in g_2$ and $x, c_i, v_i \in Z_n$ be random. Given $(g, f, k, g^x; h, y = h^x)$ and $(a_i = (k \cdot g^{v_i})^{\frac{1}{x+c_i}}, b_i = f^{c_i}, d_i = h^{c_i}, u_i = g_i^v, w_i = h^{v_i})$*

for $1 \leq i \leq q-1$, it is hard to output a new tuple $(a = (k \cdot g^v)^{\frac{1}{x+c}}, b = f^c, d = h^c, u = g^v, w = h^v)$ with $(c, v) \neq (c_i, v_i)$ for all i. i.e. one that satisfies $\hat{e}(a, y \cdot d) = \hat{e}(k \cdot u, h), \hat{e}(b, h) = \hat{e}(f, d)$ and $\hat{e}(u, h) = \hat{e}(g, w)$.

2.3 Useful Tools

Groth-Sahai Proof. Groth and Sahai [10] constructed an NIZK proof system that lets us prove statements in the context of groups with bilinear maps in the standard model. In order to proof the statement, the prover firstly commits to the group elements or $\mathbb{Z}_p$ elements. Then the prover does the proof of the group elements or $\mathbb{Z}_p$ elements and sends the commitments, the proofs and corresponding parameters to the verifier. The last the verifier verifies the correct of the proof. In this paper, We use SXDH-based GS commitments and proofs to commit to elements and prove relations satisfied by the associated plaintexts. The witness indistinguishability guarantees the anonymity of the payers and the payees during the withdraw, conditional transfer and deposit.

Randomization. Belenkiy et al. [2] proposed the randomizable proofs of commitments and the NIZK proofs. For example, let $u_{1,1} = g_1, u_{1,2} = g_1^{\mu}, u_{2,1} =$

$g_1^v, u_{2,2} = (g_1^\mu)^v, r_1, r_2 \in \mathbb{Z}_p$, we obtain a commitment for $X \in G_1$, namely $c(X) = ((u_{1,1})_1^r \cdot (u_{2,1})_2^r, X \cdot (u_{1,2})_1^r \cdot (u_{2,2})_2^r) = (c_1, c_2)$. In order to randomize the commitment, we choose two random values $r'_1, r'_2 \in \mathbb{Z}_p$ and compute the randomization as $c(X)' = (c_1 \cdot (u_{1,1})^{r'_1} \cdot (u_{2,1})^{r'_2}, c_2 \cdot (u_{1,2})^{r'_1} \cdot (u_{2,2})^{r'_2})$. Meanwhile we adapt its proof (π, θ) for commitments $(c_i)_i$ to another (π, θ) for $(c'_i)_i$. The property guarantees the anonymity of the payers and the payees. The bank can not link any withdrawal protocol, spending protocol and deposit protocol. Meanwhile, the randomizable proof is publicly verifiable.

Commuting Signatures. Commuting signatures and verifiable encryption [8] combines a signature scheme with GS proofs. This allows one to commit a commitment to a message, a verification or a corresponding signature and proves that the committed values are correct, as he does these to that message. We only briefly review two results of [8] relevant to our paper in the following.

SigCom allows a signer who is given a commitment c to a message, to make a commitment to c_σ to a signature on that message and a proof that c_σ contains a valid signature on the value committed in c.

AdC_κ allows anyone to commit to a key and adapt a proof, and outputs a commitment and a proof asserting that a commitment contains a valid signature on a committed message.

Following the definition of [8], we instantiate the specific signature with the Structure-Preserving signature (SP-signature) [1,15].

3 The Model

In this section, we firstly describe the algorithms for anonymous transferable conditional e-cash. The main differences between our algorithms and [4] is that we introduce a publisher and a new algorithm Publish() to give the bank the two commitments of the outcomes. We also extend the model given in [4] to include the publisher.

3.1 Algorithms

The conditional transferable e-cash system consists of withdraw protocol, spending(transferring) protocol, deposit protocol and identify procedure. We give the procedures as follows, where λ is a security parameter.

- ParamSetup(1^λ). It is a probabilistic algorithm that outputs the public parameters *params*.
- BKeyGen(),JKeyGen(),UKeyGen(),PKeyGen(). It is a probabilistic algorithm executed respectively by $\mathcal{B}, \mathcal{J}$ or $\mathcal{U}$, that output a key pairs $(pk_\mathcal{B}, sk_\mathcal{B})$, $(pk_\mathcal{J}, sk_\mathcal{J})$, $(pk_\mathcal{U}, sk_\mathcal{U})$ and $(pk_\mathcal{P}, sk_\mathcal{P})$.
- Withdraw$(\mathcal{U}(sk_\mathcal{U}, pk_\mathcal{U}, pk_\mathcal{B}, pk_\mathcal{J}, C_\mathcal{B}, C_\mathcal{J}), \mathcal{B}(sk_\mathcal{B}, pk_\mathcal{B}, C_{pr}, C_{pe}, pk_\mathcal{U}))$. It is an interactive protocol where $\mathcal{U}$ withdraws one conditional transferable coin *co* from $\mathcal{B}$. At the end, $\mathcal{U}$ outputs a coin or $\perp$, and $\mathcal{B}$ checks the public key of the user, deducts a coin from the user and obtains a view $\mathcal{V}$ or $\perp$.

- Publish($\mathcal{B}(sk_{\mathcal{B}}, pk_{\mathcal{B}}, C_{\mathcal{B}}, C_{\mathcal{J}}).\mathcal{P}(C_{pr}, E_{pr}, C_{pe}, E_{pe})$. It is an interactive protocol between the $\mathcal{B}$ and $\mathcal{P}$. At the end, $\mathcal{B}$ obtains two commitments representing the two outcomes.
- Spend($\mathcal{U}_1(co, sk_{\mathcal{U}_1}, pk_{\mathcal{B}}, pk_{\mathcal{J}}, C_{\mathcal{B}}, C_{\mathcal{J}}), \mathcal{U}_2(sk_{\mathcal{U}_2}, pk_{\mathcal{B}}, pk_{\mathcal{J}})$). It is an interactive protocol in which $\mathcal{U}_1$ spends/transfers the coin co to $\mathcal{U}_2$. At the end, $\mathcal{U}_2$ outputs a coin co' or $\perp$, and $\mathcal{U}_1$ outputs ok or $\perp$.
- Deposit($\mathcal{U}(co, sk_{\mathcal{U}}, pk_{\mathcal{B}}, pk_{\mathcal{J}}), \mathcal{B}(pk_{\mathcal{B}}, sk_{\mathcal{B}}, pk_{\mathcal{U}})$). It is an interactive protocol where $\mathcal{U}$ deposits a coin co to the bank. The bank outputs a commitment and corresponding proof or $\perp$. $\mathcal{U}$ deposits corresponding coin to the bank using new commitment and corresponding proof.
- Identify($pk_{\mathcal{U}}, co, co', sk_{\mathcal{J}}$). It is a deterministic algorithm executed by the judge which outputs a public key $pk_{\mathcal{U}}$ of double spender.

3.2 Security Properties

In this section, we give the security definitions for the transferable conditional e-cash system. Every security property is given by a game between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$. Firstly, we describe the ability of the adversary as arbitrary and adaptive queries to oracles. The oracles are defined as follows.

- $\mathcal{O}_{Setup()}$. This oracle allows the adversary $\mathcal{A}$ to add a new user into the system, or to corrupt a honest user. When the adversary interacts with the oracle, $\mathcal{A}$ can obtain the keys of the user or the bank. If a honest user is corrupted, the secret key is $\perp$.
- $\mathcal{O}_{With()}$. This oracle can act the bank or the user in the withdrawal protocol. The adversary $\mathcal{A}$ can withdraw a conditional e-cash from the oracle acting the bank. He can also obtain some e-cash from the oracle acting the user.
- $\mathcal{O}_{Spend()}$. This oracle can allows the adversary $\mathcal{A}$ to act a payee to receive a conditional e-cash, or act a payer to spend a conditional e-cash.
- $\mathcal{O}_{Depo()}$. This oracle can act the bank or the user in the deposit protocol. The adversary $\mathcal{A}$ can obtain a conditional e-cash from the oracle acting the user, or spend some e-cash to the oracle acting the bank.
- $\mathcal{O}_{Idt()}$. This oracle can act the judge in the identity procedures. The adversary $\mathcal{A}$ can submit two e-cash to the oracle and obtain the identity of the user spending the two e-cash.
- $\mathcal{O}_{Publ()}$. This oracle can act the publisher to extract the secret value. Then the adversary $\mathcal{A}$ can obtain the secret by interacting with the oracle.

In our paper, we think the publisher, the bank and the judger are trust organizes. The judge can not remove the identity of an honest user except that the bank gives the two spending from double spenders. The publisher only publics and announces the events correctly, and can also not extract a outcome before publishing the outcome. Meanwhile, This is the request of the fair e-cash [14]. We require all the length of the conditional e-cash is the same.In the following, we will define the security properties formally.

Anonymity. We also used the definition about anonymity in [4], but our scheme only achieve the $OtR - FA(FA)$ and $StO - FA(PA_1)$. It guarantees that no

coalition of users, publisher and judger can able to distinguish if the the spending protocol is executed by users or by a simulator.

Firstly, we give the security description of FA.

- (Initialization Phase.) $\mathcal{A}$ runs the $ParamSetup(1^\lambda)$ and obtains the public parameters $params$ and the key pairs $(pk_{\mathcal{B}}, sk_{\mathcal{B}})$, $(pk_{\mathcal{J}}, sk_{\mathcal{J}})$ and $(pk_{\mathcal{P}}, sk_{\mathcal{P}})$. Then $\mathcal{A}$ gives $pk_{\mathcal{B}}$ to $\mathcal{C}$ and keeps the $sk_{\mathcal{B}}$ to itself.
- (Probing Phase.) $\mathcal{A}$ can perform a polynomially bounded number of queries to the oracles in an adaptive manner. $\mathcal{A}$ can add and corrupt any user by the $\mathcal{O}_{Setup()}$. For each $\mathcal{O}_{With}()$ and $\mathcal{O}_{Spend}()$, $\mathcal{A}$ can act as bank or user in the withdrawal protocol or spending protocol. The adversary $\mathcal{A}$ can obtain the identity of the user from the $\mathcal{O}_{Idt()}$, or extract the secret value from the oracle $\mathcal{O}_{Publ()}$.
- (Challenge Phase.) $\mathcal{C}$ chooses two public keys $pk_{\mathcal{U}_0}$ and $pk_{\mathcal{U}_1}$, and presents them to $\mathcal{A}$. The two public keys must be the conditional e-cash received by the adversary $\mathcal{A}$. Then $\mathcal{A}$ acting the bank or the user interacts with the $\mathcal{C}$. $\mathcal{A}$ can specify which public $\mathcal{C}$ uses, with the restriction that he can not ask $\mathcal{C}$ to over-spend any coin, and can also not require the oracle $\mathcal{O}_{Idt()}$ and $\mathcal{O}_{Publ()}$. And last, $\mathcal{A}$ obtains a coin $co_{\mathcal{M}}$.
- (End Game Phase.) $\mathcal{A}$ decides which public key $\mathcal{C}$ uses.

For the security description of PA_1. The every phase is similar to the FA except that the two public keys is any keys in the Challenge Phase.

For all non-uniform polynomial time $\mathcal{A}$, the advantage breaking the anonymity is defined by

$$Adv^{anon}_{TCE,\mathcal{A}} = Pr[Exp^{anon-1}_{TCE,\mathcal{A}}(\lambda) = 1] - Pr[Exp^{anon-0}_{TCE,\mathcal{A}}(\lambda) = 1]$$

where TCE be a anonymous transferable conditional e-cash system. The $Exp^{anon}_{TCE,\mathcal{A}}(\lambda)$ is the same as that in [4] except that we give the $\mathcal{A}$ an ability to access the private key sk_{pe} and sk_{pr}.

If the $Adv^{anon}_{TCE,\mathcal{A}}$ is negligible for any polynomial-time adversary $\mathcal{A}$, we will say that our scheme is anonymous.

Unforgeability. No coalition of users and merchants can deposit more coins than they have withdrawn from the bank.

- (Initialization Phase.) $\mathcal{C}$ runs the $ParamSetup(1^\lambda)$ and obtains the public parameters $params$ and the key pairs $(pk_{\mathcal{B}}, sk_{\mathcal{B}})$, $(pk_{\mathcal{J}}, sk_{\mathcal{J}})$ and $(pk_{\mathcal{P}}, sk_{\mathcal{P}})$. Then $\mathcal{C}$ gives $pk_{\mathcal{B}}$ to $\mathcal{A}$ and keeps the $sk_{\mathcal{B}}$ to itself.
- (Probing Phase.) $\mathcal{A}$ can perform a polynomially bounded number of queries to the oracles in an adaptive manner. $\mathcal{A}$ can add and corrupt any user by the $\mathcal{O}_{Setup()}$. We define the e-cash received by the $\mathcal{A}$ is co_a and initialize it with zero. For each $\mathcal{O}_{With}()$, $\mathcal{A}$ acting as user and withdraw a conditional e-cash co_0 in the withdrawal protocol. In the transferring protocol, the $\mathcal{A}$ acting as payer transfers an e-cash co_1 to the payee, or acting as payee receives an e-cash co_2. The $\mathcal{A}$ deposit an e-cash co_de to the $\mathcal{C}$ acting as the bank in the deposit protocol. And last, the counter of the $\mathcal{A}$ is $co_a = co_0 + co_2$.

– (End Game Phase.) $\mathcal{A}$ wins the game if it can deposit $co_a + 1$ to $\mathcal{C}$, namely $co_{de} > co_a$.

For all non-uniform polynomial time $\mathcal{A}$, the advantage breaking the unforgeability is defined by

$$Adv^{unfor}_{TCE,\mathcal{A}} = Pr[Exp^{unfor}_{TCE,\mathcal{A}}(\lambda) = 1]$$

where TCE be a anonymous transferable conditional e-cash system. The $Exp^{unfor}_{TCE,\mathcal{A}}(\lambda)$ is the same as that in [4] except that we give the $\mathcal{A}$ an ability to access the private key sk_{pe} and sk_{pr}.

If the $Adv^{unfor}_{TCE,\mathcal{A}}$ is negligible for any polynomial-time adversary $\mathcal{A}$, we will say that our scheme is unforgeable.

Double-Spending. It guarantees that coalition of users and merchants can not be able to double-spend a coin with the same serial number.

This is similar to the unforgeability in the Initialization Phase.

– (Probing Phase.) $\mathcal{A}$ can perform a polynomially bounded number of queries to the oracles in an adaptive manner. $\mathcal{A}$ can add and corrupt any user by the $\mathcal{O}_{Setup()}$. In order to identify the identity of an honest user, the adversary $\mathcal{A}$ extracts the message committed in commitments and corresponding proof, and then forges a new coin. Meanwhile, the adversary can deposit the new coin to the bank, and the output of the algorithm Identify() can not output the public key. If the adversary can give a new coin as above, he must break the unforgeability of the commuting signature and the soundness and witness indistinguishability of GS proofs.
– (End Game Phase.) $\mathcal{A}$ wins the game if it can deposit a coin, the output of the Deposit() is $\perp$ and the Identify() cannot output the public key.

For all non-uniform polynomial time $\mathcal{A}$, the advantage breaking the double-spending is defined by

$$Adv^{unfor}_{TCE,\mathcal{A}} = Pr[Exp^{ide}_{TCE,\mathcal{A}}(\lambda) = 1]$$

where TCE be a anonymous transferable conditional e-cash system. The $Exp^{ide}_{TCE,\mathcal{A}}(\lambda)$ is the same as that in [4] except that we give the $\mathcal{A}$ an ability to access the private key sk_{pe} and sk_{pr}.

If the $Adv^{ide}_{TCE,\mathcal{A}}$ is negligible for any polynomial-time adversary $\mathcal{A}$, we will say that our scheme can identify the double-spending.

Exculpability. No coalition of the banks and users can accuse an honest users of have double-spending a coin.

This is shown similarly to the FA in the Initialization Phase and the Probing Phase.

– (Probing Phase.) $\mathcal{A}$ can perform a polynomially bounded number of queries to the oracles in an adaptive manner. $\mathcal{A}$ can add and corrupt any user by the $\mathcal{O}_{Setup()}$. We know the commute of the user and the bank is anonymous. If the adversary $\mathcal{A}$ wants to forge another coin and frames an honest user, he must break unforgeability of the commuting signature.

– (End Game Phase.) $\mathcal{A}$ wins the game if it can forge a corresponding e-cash and prove the spending is correct.

For all non-uniform polynomial time $\mathcal{A}$, the advantage breaking the exculpability is defined by

$$Adv^{unfor}_{TCE,\mathcal{A}} = Pr[Exp^{excu}_{TCE,\mathcal{A}}(\lambda) = 1]$$

where TCE be a anonymous transferable conditional e-cash system. The $Exp^{excu}_{TCE,\mathcal{A}}(\lambda)$ is the same as that in [4] except that we give the $\mathcal{A}$ an ability to access the private key sk_{pe} and sk_{pr}.

If the $Adv^{excu}_{TCE,\mathcal{A}}$ is negligible for any polynomial-time adversary $\mathcal{A}$, we will say that our scheme can frame an honest user making a double-spending.

Conditional Transfer. The payer can cash back his coin when an unfavorable outcome happens. The payee can anonymously transfer the coin from the payer to the payee until the time that the outcome is published. After publishing the outcome, the publisher can extract the secret value, and send it to the corresponding user and the bank. Then the user deposit the conditional e-cash to the bank. Thus, the coin can be transfer to many users.

4 General Description

In a transferable e-cash with a condition, the payer anonymously transfers e-cash before the outcome of the condition is published. The e-cash is valid to the payer and the last payee, and only one of the two users can deposit the e-cash. When the outcome is published, the publisher sends the extraction key to the winner by an authenticated and secure channel. If a user submits the e-cash to the publisher and wants to deposit the e-cash to the bank, the bank detects whether the user has happened a double-spending. If so, the bank recovers the identity of the user by the identify procedure. The transferable conditional e-cash consists of the withdrawal protocol, spending (transferring) protocol, deposit protocol and the identify procedure. We provide a new algorithm to construct the transferable e-cash system based on the outcome of a condition. The general description is given as follows.

The payer firstly withdraws an e-cash from the bank, and decides to spend the e-cash to other user based on a condition. The transferring protocol is anonymous to protect the identity of the payer $\mathcal{U}_1$ and the payee $\mathcal{U}_2$. To achieve anonymity, two tricks are adopt to our e-cash system. Firstly, commuting signature technology is used in our system. Generally, commitments and the corresponding proofs are used in the interaction between $\mathcal{U}_1$ and $\mathcal{U}_2$ to achieve anonymity. However this is not enough, i.e., if $\mathcal{U}_2$ transfers the e-cash to other payee $\mathcal{U}_3$, the bank knows that the e-cash is from the same user $\mathcal{U}_1$. Thanks to commuting signature technology, we can modify the commitments and proofs to achieve anonymity. Secondly, we divide the deposit into two parts to obtain the anonymity of the last user in the deposit protocol. More precisely, since the identity of the user

is supplied in the deposit protocol, the bank can link the spending of the last user and the deposit of the user. To obtain the anonymity of the last user in the deposit protocol, we divide the deposit into two parts. The first part is exchanging in which we can exchange the spending to another spending mo provided by the bank. The other part is cashing, where the user updates the spending mo to mo' using the commuting signatures, and provides mo' and an account number to the bank. In our scheme, the most important problem is how to obtain the conditional e-cash, namely, two outcomes for the user $\mathcal{U}_1$ and another user $\mathcal{U}_i$. We achieve this goal by introducing a publisher, which gives two commitment/extraction keys. The two commitment/extraction keys commit two secret value for the two outcomes. When the outcome is favorable to a user, the corresponding secret value is sent to the corresponding user by an authenticated and secure channel. This publisher is very important, since he publishes the conditions of two events and the correct outcome to decide who can deposit the e-cash to the bank.

5 Conditional Transferable E-cash

Conditional transferable e-cash allows the user to spend a conditional e-cash to other one based on the outcome in the future. In the follow, we give the details of our scheme.

5.1 Setup

The bank $\mathcal{B}$, the judge $\mathcal{J}$, the publisher $\mathcal{P}$ and each user $\mathcal{U}$ generate key pairs $(pk_{\mathcal{B}}, sk_{\mathcal{B}})$, $(pk_{\mathcal{J}}, sk_{\mathcal{J}})$, $(pk_{\mathcal{P}}, sk_{\mathcal{P}})$ and $(pk_{\mathcal{U}}, sk_{\mathcal{U}})$ respectively. The bank also generates a pair of commitment/extraction key $(C_{\mathcal{B}}, E_{\mathcal{B}})$, which is used to committed the serial number of the e-cash. The users register their public keys to the judge as membership certificate: $cert_i = SPSign_{pk_{\mathcal{J}}}(pk_{\mathcal{U}_i})$, this is similar to the action that the users obtain an identity from the country and then open a bank account from the bank. The publisher gives two pairs of commitment/extraction keys, one is (C_{pr}, E_{pr}) which is used to commit the coin for the payer, and another is (C_{pe}, E_{pe}) which is used to commit the coin for the payee. If the outcome is favorable to the payer, the payer will obtain the secret key C_{pr}, otherwise the payee will obtain C_{pe}. The judge also generates two pairs of commitment/extraction keys $(C_{\mathcal{J}}, E_{\mathcal{J}})$ and (C_{sp}, E_{sp}), the first key will be used for identification of double spenders, and another key is used for the proof of our scheme. The bank maintains a database DB, which is used to save the e-cash spent. The database is initialized to empty.

5.2 The Withdrawal Protocol

The withdrawal protocol allows the user $\mathcal{U}_1$ to withdraw a coin co from the bank $\mathcal{B}$. We define the commitments of the publisher as p_n, p_m for the outcomes, which is favorable to the payer, the payee respectively, and the commitments

$\widetilde{p}_n, \widetilde{p}_{n'}$, which is used to supply some information to adversary for the security proof. The publisher publics the two commitments. And then the user registers the outcome at the publisher. The j_n and $\widetilde{j}_n$ are defined as commitments that commit a message n using $C_{\mathcal{J}}$ and C_{sp}. The b_n is defined as commitment that commits a message n using $C_{\mathcal{B}}$. The p_n and $\widetilde{p}_n$ are defined as commitments that commit a message n using C_{pr} and C_{pe}. In the following, We will give the protocol in detail.

1. $\mathcal{U}_1$ picks at random a nonce n_1 and makes commitments b_{n_1} using the commitment key of the bank, which represents the serial number of the coin, j_{n_1} to n_1 using the commitment key of the judge, which is used to verify the spending chain of users and a proof π_{n_1} that two committed values are equal [9]. Moreover, $\mathcal{U}_1$ makes commitments $j_{pk_{\mathcal{U}_1}}$, $\widetilde{j}_{pk_{\mathcal{U}_1}}$ to the public key of the user, which is used to recover the identity of users when a double-spending happens, and a proof $\pi_{pk_{\mathcal{U}_1}}$ that two committed values are equal. $\mathcal{U}_1$ also gives the commitment of the signature of the public key of user $j_{c_{\mathcal{U}_1}}$, which is made by the judge. At last, $\mathcal{U}_1$ gives the proof $\pi_{c_{\mathcal{U}_1}}$ [10,4] that the value in $j_{c_{\mathcal{U}_1}}$ is a valid signature on the value in $j_{pk_{\mathcal{U}_1}}$.
   The user $\mathcal{U}_1$ sends the following values to the bank: $\{pk_{\mathcal{U}_1}, j_{pk_{\mathcal{U}_1}}, j_{n_1}, j_{c_{\mathcal{U}_1}}, \pi_{c_{\mathcal{U}_1}}\}$.
2. In order to supply two outcomes, $\mathcal{B}$ picks at random n and m, and generates two commitments p_n and p_m using the commitment keys C_{pr} and C_{pe} respectively. $\mathcal{B}$ also gives the other two commitments $\widetilde{p}_n$ and $\widetilde{p}_m$ which is used to prove the scheme completely. Meanwhile two proofs π_n and $\pi_{n'}$ are given to prove two committed values are equal respectively. If the outcome is favorable to the payer, the publisher $\mathcal{P}$ will give the extraction key to the payer. Then the payer can open the commitment p_n and obtain the value n.
3. $\mathcal{B}$ verifies the NIZK proof $\pi_{c_{\mathcal{U}_1}}$ and the public $pk_{\mathcal{U}_1}$. If they are correct, the bank $\mathcal{B}$ chooses a random nonce n_0 for the coin and generates two commitments j_{n_0} and b_{n_0} using the commitment keys $C_{\mathcal{J}}$ and $C_{\mathcal{B}}$ respectively. The bank also gives a proof π_{n_0} that the two committed values are equal. Then the bank $\mathcal{B}$ produces a committed signature $c_{s_{c_1}}$ on the values n,m,n_0,n_1 and $pk_{\mathcal{U}_1}$ by running $SigCom$ on p_n,p_m,j_{n_0}, j_{n_1} and $j_{pk_{\mathcal{U}_1}}$, and also outputs the proof π_{s_1} [10,4]that the signature $c_{s_{c_1}}$ is valid to commitments p_n,p_m,j_{n_0}, j_{n_1} and $j_{pk_{\mathcal{U}_1}}$.

The bank $\mathcal{B}$ sends the following values to the user: $\{p_n, \widetilde{p}_n, \pi_n, p_m, \widetilde{p}_m, \pi_m, j_{n_0}, b_{n_0}, \pi_{n_0}, c_{s_{c_1}}, \pi_{s_1}\}$.

Finally, the user $\mathcal{U}_1$ forms the coin $co_1 = (p_n, \widetilde{p}_n, \pi_n, p_m, \widetilde{p}_m, \pi_m, j_{n_0}, b_{n_0}, \pi_{n_0}, j_{n_1}, b_{n_1}, \pi_{n_1}, j_{pk_{\mathcal{U}_1}}, \widetilde{j}_{pk_{\mathcal{U}_1}}, \pi_{pk_{\mathcal{U}_1}}, j_{c_{\mathcal{U}_1}}, \pi_{c_{\mathcal{U}_1}}, c_{s_{c_1}}, \pi_{s_1})$.

5.3 The Spending(Transferring) Protocol

This is a protocol which makes a payer $\mathcal{U}_1$ to transfer a coin to the payee $\mathcal{U}_2$. In order to obtain the anonymity of the user, $\mathcal{U}_1$ needs randomize the coin co_i before he spends the coin to the third user. $\mathcal{U}_1$ also converts the proof π_{s_i} to a new proof by running AdC_κ, which will hide the identity of the user.

1. $\mathcal{U}_2$ picks at random a nonce n_2 and makes a commitments b_{n_2} using the commitment key of the bank, which represents the coin, j_{n_2} to n_2 using the commitment key of the judge, which is used to verify the spending chain of users and a proof π_{n_2} that two committed values are equal. Moreover, $\mathcal{U}_2$ makes commitments $j_{pk_{\mathcal{U}_2}}$, $\widetilde{j}_{pk_{\mathcal{U}_2}}$ and a proof $\pi_{pk_{\mathcal{U}_2}}$ that two committed values are equal. At last, $\mathcal{U}_2$ gives the proof $\pi_{c_{\mathcal{U}_2}}$ that the value in $j_{c_{\mathcal{U}_2}}$ is a valid signature on the value in $j_{pk_{\mathcal{U}_2}}$.
   The user $\mathcal{U}_2$ sends the following values to $\mathcal{U}_1$: $\{j_{pk_{\mathcal{U}_2}}, j_{n_2}, j_{c_{\mathcal{U}_2}}, \pi_{c_{\mathcal{U}_2}}\}$.
2. $\mathcal{U}_1$ firstly checks the proof $\pi_{c_{\mathcal{U}_2}}$. If the verification is correct, $\mathcal{U}_1$ randomizes co_1 to $co_1^1 = (p_n^1, \widetilde{p}_n^1, \pi_n^1, p_m^1, \widetilde{p}_m^1, \pi_m^1, j_{n_0}^1, d_{n_0}^1, \pi_{n_0}^1, j_{n_1}^1, d_{n_1}^1, \pi_{n_1}^1, j_{pk_{\mathcal{U}_1}}^1, \widetilde{j}_{pk_{\mathcal{U}_1}}^1, \pi_{pk_{\mathcal{U}_1}}^1, j_{c_{\mathcal{U}_1}}^1, \pi_{c_{\mathcal{U}_1}}^1, c_{s_{c_1}}^1, \pi_{s_1}^1)$. Then $\mathcal{U}_1$ computes a committed signature $c_{s_{c_2}}$ on the values committed in $j_{n_1}^1, j_{n_2}$ and $j_{pk_{\mathcal{U}_2}}$ using $SigCom$, and also outputs the proof π'_{s_2} that the signature $c_{s_{c_2}}$ is valid to commitments $j_{n_1}^1, j_{n_2}$ and $j_{pk_{\mathcal{U}_2}}$. To hide the verification key of $\mathcal{U}_1$, $\mathcal{U}_1$ also converts π'_{s_2} to π_{s_2} by running AdC_κ.

The $\mathcal{U}_1$ sends the following values to $\mathcal{U}_2$: $\{co_1^1, C_{s_{c_2}}, \pi_{s_2}\}$.

Finally, the user $\mathcal{U}_2$ forms the coin $co_2 = (co_1^1, j_{n_2}, b_{n_2}, \pi_{n_2}, j_{pk_{\mathcal{U}_2}}, \widetilde{j}_{pk_{\mathcal{U}_2}}, \pi_{pk_{\mathcal{U}_2}}, j_{c_{\mathcal{U}_2}}, \pi_{c_{\mathcal{U}_2}}, c_{s_{c_2}}, \pi_{s_2})$.

5.4 The Deposit Protocol

When the outcome is published, the winner contacts the publisher and provides the corresponding proof to the publisher by an authenticated channel. If the proof is correct, the publisher sends the secret value to the winner. Without loss of generality, the outcome is favorable to the payee, so the payee will obtain the corresponding extraction key E_{pe}. To achieve the anonymous of the user during the deposit, we divide the deposit protocol into two sections, exchanging and cashing.

Exchanging. $\mathcal{U}_i$ spends the coin $co_\ell = (n^\ell, p_n^\ell, m, p_m^\ell, co_1^\ell, co_2^{\ell-1}, \cdots, co_{\ell-1}^2, j_{n_\ell}^1, b_{n_\ell}^1, \pi_{n_\ell}^1, j_{pk_{\mathcal{U}_\ell}}^1, \widetilde{j}_{pk_{\mathcal{U}_\ell}}^1, \pi_{pk_{\mathcal{U}_\ell}}^1, c_{j_{\mathcal{U}_\ell}}^1, \pi_{c_{\mathcal{U}_\ell}}^1, c_{s_{c_\ell}}^1, \pi_{s_\ell}^1)$ to the bank, that is, $\mathcal{U}_i$ runs the protocol with the bank playing the role of $\mathcal{U}_2$. In order to detect the double-spending, the bank firstly verifies the correctness of the secret value and commitment (m, p_m) and checks whether m equals the value committed in p_m. Then $\mathcal{B}$ opens the commitments $b_{n_0}^\ell, b_{n_1}^{\ell-1}, \cdots, b_{n_{\ell-1}}^2, b_{n_\ell}^1$ contained in the coin, using the extraction key of the bank. And last, the bank obtains the serial number $s = m||n_0||n_1||n_2||\cdots||n_\ell$, and checks whether the coin is found in the database DB. If it does not, the bank sends the value $mo = (m, c_m, c_{\sigma_m}, \pi_{c_m})$ to the user $\mathcal{U}_i$, which represents a correct deposit of the user. The bank also saves the serial number to the database DB. Otherwise, the bank running the following identify procedures.

Cashing. In order to cash the coin from the bank, the user $\mathcal{U}_i$ converts the value $mo = (m, c_m, c_{\sigma_m}, \pi_{c_m})$ to $mo' = (m, c'_m, c'_{\sigma_m}, \pi'_{c_m})$ using AdC_κ. Then

$\mathcal{U}_i$ directly contacts the bank $\mathcal{B}$ through an authenticated channel, supplies his account number and exchanges this piece of currency for credit to his account.

If the outcome is favorable to the first payer $\mathcal{U}_1$, the user $\mathcal{U}_1$ deposits the coin to the bank as the above procedure except that the $\mathcal{U}_1$ spends the coin co_1^1 and the bank $\mathcal{B}$ will obtain the serial number $s = n||n_0||n_1|| \cdots ||n_\ell$.

5.5 The Identify Procedures

If $\mathcal{B}$ can find another serial number beginning with n in his database, i.e. $s' = n||n'_0||n'_1||n'_2|| \cdots ||n'_\ell$. He compares the two serial numbers s and s' and stops at the last t such that $n_t = n'_t$. Finally, the bank sends the two coins to the judge. In order to identify the defrauder, the judge computes the identity committed in $j_{pk_{\mathcal{U}_\ell}}$ using his extraction key $E_{\mathcal{J}}$.

5.6 Efficiency

We analysis the efficiency by comparing the computation and communication. Shi et al. [11] requires $O(m_1 m_2 k)$ computation and communication, where m_1 and m_2 are cut-and-choose parameters and k is a security parameter for RSA-based systems. We know that in cut-and-choose techniques with a parameter m a dishonest user can cheat with probability $1/m$, therefore a protocol that has the overhead of $O(m^2 k)$ is too heavy for any user. Blanton needs $O(\lambda log m_3)$, where λ is a security parameter for groups with bilinear maps and m_3 is the probability of cheating.

On a contrary, we use the commitments and corresponding proofs for representing a conditional e-cash. The proofs are given by the NIZK proof using GS proofs. Thus, the payer only needs to send some commitments and proofs to the payee in the spending protocol, transferring protocol and the deposit protocol, and the communication number only needs one time. Therefore, computation and communication in our scheme are constant, namely $O(\lambda)$, where λ is the system parameter or a security parameter for groups with bilinear maps.

6 Security Analysis

We now give the security of our scheme. The scheme fulfills the security requirements given in Section 3.

Theorem 1. *Our conditional transferable e-cash scheme provides anonymity, unforgeability, identification of double-spender and exculpability under the following assumptions: SXDH assumption, unforgeability of the commuting signature scheme, soundness of NIZK proofs and witness indistinguishability of GS proofs.*

Proof. We briefly analyze the security properties as follows.

Anonymous. The anonymous of our scheme only achieves the FA and PA_1. Our scheme commits all messages sent between the users when transferring a

coin. If the adversary wants to determine which one is the user chosen by the Challenge in Challenge Phase, he needs to extract the public key committed in $j_{pk_{\mathcal{U}_i}}$. Thus, the adversary can break the soundness of NIZK proofs and witness indistinguishability of GS proofs [6].

In the following, we give the prove of the FA by a game between an adversary $\mathcal{A}$ and a challenge $\mathcal{C}$. For fair e-cash, the judge and the publisher is necessary, so they can not extract the secret key except that the bank give the proof that the user happens a double-spending.

(Initialization Phase.)Let $\mathcal{C}$ supplies a system parameter λ to the adversary $\mathcal{A}$ acting as the bank. The adversary $\mathcal{A}$ obtains the key pairs of the bank, the judge and the publisher. And then (A) sends the public keys and the commitment keys to the $\mathcal{C}$.

(Probing Phase.)In the withdrawal protocol, the $\mathcal{C}$ acting the payer withdraws conditional e-coins, then converts the coins to new coins by the algorithm $SigCom$. The $\mathcal{A}$ any payee accepts the e-coins from the $\mathcal{A}$ act the payer in the spending protocol. In the deposit protocol, the user deposits a coin to the $\mathcal{A}$ acting the bank. By the above interact, the $\mathcal{A}$ obtains some commitments and corresponding proofs. The length of the coins are the same.

(End Game Phase.)The $\mathcal{C}$ chooses two coins co_0 and co_1 from these coins in the Probing Phase, and then flips a fair coin to decide to use co_0 or co_1 for the deposit protocol. The GS proofs is soundness, so we know the commitments and the corresponding proofs are correct. If the adversary $\mathcal{A}$ can distinguish which coin the user deposits, co_i(i=0/1), he must distinguish which public key the user uses, $pk_{\mathcal{U}_i}$, namely $i = 0$ or $i = 1$. We know if the $\mathcal{A}$ can solve the SXDH problem, he can distinguish which commitment the user uses. But the probability of solving the SXDH problem is ignore. For the commuting signatures, the $\mathcal{A}$ can not forge any commuting signatures, so he can not give any help to distinguish $i = 0$ or $i = 1$.

So we know the $\mathcal{A}$ can win if he can forge a commuting signature, solve the $SXDH$ problem and break the soundness of the NIZK proofs.

Unforgeability. Let $\mathcal{A}$ be an adversary. We outline the success probability of $\mathcal{A}$ is negligible by interacting with a challenger $\mathcal{C}$. The $\mathcal{C}$ gives the public key of the bank. The $\mathcal{A}$ generates the remaining parameters.

For each KeyGen(), $\mathcal{A}$ can create a new user or corrupt an honest user. So $\mathcal{A}$ obtains some key pairs. In withdrawal protocol, $\mathcal{A}$ can act as a user and withdraw some coins from the challenger, we defined the coins as co_{ui}. For each spending protocol, $\mathcal{A}$ can act as a user and spend(transfer) some coins to challenger, we defined the coins as co_{uo}. he also gets some coins from challenger, we defined the coins as co^1_{ui}. In deposit protocol, $\mathcal{A}$ deposits some coins to challenger, and other users deposit some coins which come from $\mathcal{A}$, to challenger. We defined all the deposit as co^1_{uo}. So the $\mathcal{A}$ obtains the value of the coins is $co = co_{ui} + co^1_{ui} - co_{uo} - co^1_{ui}$. If $co > 0$, the $\mathcal{A}$ has spent a coin which is not withdrew from the bank.

We know the other users and the bank are honest, and the NIZK proof π_{s_i} is soundness. If the $\mathcal{A}$ can spend more coins which are not withdrew from the

bank, he must give a forgery of a new triples $(n, n_0, pk_{\mathcal{U}_1})$ from the bank. Thus, $\mathcal{A}$ outputs a new signature on a message as a forgery, namely $\mathcal{A}$ breaks the unforgeability of commuting signature.

Double-Spending. $\mathcal{C}$ gives the public key of the judge for identifying the identity of the double-spender. $\mathcal{A}$ sets up the remaining parameters.

When $\mathcal{A}$ can spend a coin twice without revealing the identity of the $\mathcal{A}$, the output of the Identify() is not the public key of any user. Because each valid coin contains a valid certificate for the public key, by the soundness of NIZK proofs, we know the $\mathcal{A}$ must forge a new valid certificate for the public key registered in judge. Therefore, $\mathcal{A}$ breaks the soundness of NIZK proofs and the unforgeability of commuting signature.

Exculpability. The exculpability is that the $\mathcal{A}$ acting the bank can accuse an honest user of happening a double-spending. We request that the signature is issued by an honest user rather than the bank.

If a user is identified as a double spending, the $\mathcal{A}$ needs to supply two correspond spending to the user. The two spending containing two serial numbers s and s' in which $n_i = n'_i$. By the soundness of NIZK proofs, the two coins contain correct signatures. In order to accuse the honest user of happening the double-spending, the $\mathcal{A}$ forges a signature on a coin. Therefore, $\mathcal{A}$ breaks the unforgeability of commuting signature.

7 Conclusion

In this paper, we presented the first anonymous transferable conditional e-cash. One of the most features in our protocol is that the spending and deposit protocol is anonymous. In this protocol, we can modify commitments and corresponding proof using the commuting signature and the GS proofs. How to design a efficient conditional transferable e-cash will be a new think.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (grant number 60973105, 90718017, 61170189), the Research Fund for the Doctoral Program of Higher Education (grant number 20111102130003) and the Fund of the State Key Laboratory of Software Development Environment (grant number SKLSDE-2011ZX-03, SKLSDE-2012ZX-11).

References

1. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-Preserving Signatures and Commitments to Group Elements. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 209–236. Springer, Heidelberg (2010)
2. Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., Shacham, H.: Randomizable Proofs and Delegatable Anonymous Credentials. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 108–125. Springer, Heidelberg (2009)
3. Blanton, M.: Improved Conditional E-Payments. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 188–206. Springer, Heidelberg (2008)

4. Blazy, O., Canard, S., Fuchsbauer, G., Gouget, A., Sibert, H., Traoré, J.: Achieving Optimal Anonymity in Transferable E-Cash with a Judge. In: Nitaj, A., Pointcheval, D. (eds.) AFRICACRYPT 2011. LNCS, vol. 6737, pp. 206–223. Springer, Heidelberg (2011)
5. Camenisch, J., Lysyanskaya, A., Meyerovich, M.: Endorsed e-cash. In: IEEE Security and Privacy 2007. Springer (2007)
6. Canard, S., Gouget, A.: Anonymity in Transferable E-cash. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 207–223. Springer, Heidelberg (2008)
7. Chaum, D., Pedersen, T.: Transferred Cash Grows in Size. In: Rueppel, R.A. (ed.) EUROCRYPT 1992. LNCS, vol. 658, pp. 390–407. Springer, Heidelberg (1993)
8. Fuchsbauer, G.: Commuting Signatures and Verifiable Encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 224–245. Springer, Heidelberg (2011)
9. Fuchsbauer, G., Pointcheval, D., Vergnaud, D.: Transferable Constant-Size Fair E-Cash. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 226–247. Springer, Heidelberg (2009)
10. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
11. Shi, L., Carbunar, B., Sion, R.: Conditional E-Cash. In: Dietrich, S., Dhamija, R. (eds.) FC 2007 and USEC 2007. LNCS, vol. 4886, pp. 15–28. Springer, Heidelberg (2007)
12. Okamoto, T., Ohta, K.: Universal Electronic Cash. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 324–337. Springer, Heidelberg (1992)
13. Canard, S., Gouget, A., Traoré, J.: Improvement of Efficiency in (Unconditional) Anonymous Transferable E-Cash. In: Tsudik, G. (ed.) FC 2008. LNCS, vol. 5143, pp. 202–214. Springer, Heidelberg (2008)
14. Canard, S., Delerablée, C., Gouget, A., Hufschmitt, E., Laguillaumie, F., Sibert, H., Traoré, J., Vergnaud, D.: Fair E-Cash: Be Compact, Spend Faster. In: Samarati, P., Yung, M., Martinelli, F., Ardagna, C.A. (eds.) ISC 2009. LNCS, vol. 5735, pp. 294–309. Springer, Heidelberg (2009)
15. Abe, M., Groth, J., Haralambiev, K., Ohkubo, M.: Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 649–666. Springer, Heidelberg (2011)
16. Okamoto, T., Ohta, K.: Disposable Zero-Knowledge Authentications and Their Applications to Untraceable Electronic Cash. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 481–496. Springer, Heidelberg (1990)