Optimally Anonymous and Transferable Conditional E-cash*

Jiangxiao Zhang, Hua Guo, Zhoujun Li and Chang Xu

State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China
orange_0092008@163.com,hguo@buaa.edu.cn,lizj@buaa.edu.cn

Abstract. Transferable conditional electronic-cash (e-cash) allows a payer to spend an e-cash based on the outcome not known in advance. It also allows a payee to spend the e-cash to others, or deposit the e-cash to a bank based on the future outcome. Among security properties, the anonymity of the payer has been widely studied. However, the payer is linkable in the existing conditional e-cash schemes. This paper presents the first optimally anonymous and transferable conditional electronic-cash (e-cash) system based on two recent cryptographic primitives, i.e., the Groth-Sahai(GS) proof system and the commuting signatures, to obtain the user's unlinkability and optimal anonymity. A publisher is introduced to publish the conditions, and is firstly formalized. By dividing the deposit protocol into two parts, the anonymity of the user is obtained in the deposit protocol. Compared with the existing conditional e-cash schemes, this scheme has the constant size for the computation and communication. Finally, we give the security proof in the standard model.

Key words: Conditional E-cash; Transferability; Anonymity; Groth-Sahai Proofs; Commuting Signatures.

1 Introduction

Electronic cash (E-cash) is the digital equivalent of regular money. E-cash was introduced by Chaum [14] in 1982. It generally consists of three parts, i.e., the bank $\mathcal{B}$, the user $\mathcal{U}$ and the merchant $\mathcal{M}$. The user firstly establishes an account and withdraws coins from the bank $\mathcal{B}$. Then the user $\mathcal{U}$ spends the cash to the merchant $\mathcal{M}$. At last, the merchant $\mathcal{M}$ deposits the cash to the bank $\mathcal{B}$. There are some interesting varieties, such as divisible e-cash [14, 26, 19, 3, 4, 5, 6, 7, 31, 8, 9, 10, 11], transferable e-cash [26, 21, 20, 23, 27, 18] and conditional e-cash [25, 17], et. al. Among them, the transferable e-cash allows the recipient of a coin in a transaction to transfer it in a later payment transaction to a third person without contacting a bank; the conditional e-cash allows a payer to spend a e-cash based on the future outcome.

Conditional e-cash was firstly introduced by Shi *et al*. [25]. It allows a participant to spend an e-cash to others based on the future outcome. After the outcome publishes, only one user (a payer/a payee) deposits the conditional e-cash to the bank, if and only if the outcome is favorable to the user. If the outcome is not favorable to the payer, the payer loses and the payee cashes the e-cash from $\mathcal{B}$; otherwise, the payer cashes back the e-cash. Therefore, only one user (the winner) deposits the e-cash to $\mathcal{B}$. There are many applications of conditional e-cash, i.e., securities trading [25], prediction markets [25] and online betting [25].

The existing transferable conditional e-cash [17] consists of the users (payers and payees) $\mathcal{U}_1, \mathcal{U}_2, \cdots, \mathcal{U}_n$, the bank $\mathcal{B}$ and the publisher $\mathcal{P}$. $\mathcal{B}$ is responsible for issuing the conditional e-cash. $\mathcal{P}$ is responsible for publishing two conditional commitments of two outcomes. However, the payer recognizes the coin which he has observed previously and the payee $\mathcal{U}_n$'identity is not unlinkable. More precisely, $\mathcal{U}_n$ is not able to modify the zero-knowledge proof. Therefore, the payer can recognize the coin which he has

* This paper is the extended version of the paper [1] in SECURECOMM 2012.

observed previously; the bank knows the identity of the last payee. These do not satisfy the anonymity property. The problem is left as an open problem.

To solve these problems, we propose the new transferable conditional e-cash using the Groth-Sahai proof system and the commuting signature. We also introduce an judge $\mathcal{J}$ which is responsible for registering new users and recovering the identity of double-spender. The whole processing is divided into three parts. (1) The publisher publishes two conditions about two outcomes. In this paper, the commitments represent the conditions. The user $\mathcal{U}_1$ registers one of the two outcomes at the publisher. (2) $\mathcal{U}_1$ withdraws a coin *co* from the bank $\mathcal{B}$, and then spends the coin to the user $\mathcal{U}_2$. The payee $\mathcal{U}_2$ spends the coin to the third user $\mathcal{U}_3$, or deposits the coin to $\mathcal{B}$. (3) When the publisher publishes the outcome, only the winner wins the coin, and then the winner cashes from the bank $\mathcal{B}$. After $\mathcal{B}$ checks the correctness of the coin, he decides to credits the $\mathcal{M}$'s account or announce the judge to recover the identity of the double-spender.

The transferable conditional e-cash has similarities with traditional transferable e-cash. The most difference is that a condition is introduced in the transferable conditional e-cash. More precisely, in the transferable conditional e-cash, the payee can not deposit the e-cash unless the outcome of the condition is published and it is favorable to the payee, while in the traditional transferable e-cash the user spends an e-cash without any condition. Additionally, the payer can cash back the e-cash in the case of an unfavorable outcome to the payee, but this scenario is not applicable in the traditional transferable e-cash. Therefore, new tools are needed to construct the transferable conditional e-cash.

1.1 Related Results

Much research has been performed in the e-cash. Okamoto and Ohata proposed the first ideal untraceable electronic cash [26] using the cut-and-choose methodology and introduces some basic properties, i.e., untraceability, transferability and divisibility. The cut-and-choose methodology causes low efficiency of Okamoto and Ohata's scheme. Pailles constructed a new protocol for e-cash [2] which develops the anonymity and the divisibility of the e-cash. Unfortunately, the bank has to perform a huge amount of computations. As for divisibility, Eng and Okamoto proposed a single-term divisible e-cash [3] which is not a practical divisible e-cash. Then Okamoto presented the first practical divisible e-cash [4] which was subsequently improved by Chan *et al.* [5]. However, the schemes mentioned above are linkable, since anyone can decide whether several spends come from the same coin. In 2000, Nakanishi and Sugiyama provided an unlinkable divisible electronic cash [6] by introducing a trusted third party.

A trusted third party recovers the identity of double-spender. However, if the trusted third party is compromised, the anonymity of the user is impossible. To solve the problem, Camenisch, Hohenberger and Lysyanskaya described a compact e-cash [7] which allows the user to withdraw a wallet efficiently containing 2^L coins. Meanwhile, the scheme removes a trusted third party. Unfortunately, in the withdrawal protocol the user chooses the number of the coins which have to be only spent one by one in the spending protocol. Thus, the scheme is very inconvenient to the user, and the efficiency is very low in the spending protocol. The first anonymous divisible e-cash scheme [8] was proposed by Canard and Gouget. When a user spends a small number of coins, he must prove that the spending protocol is constructed correctly using non-interactive zero-knowledge proof of knowledge. In the spending protocol, the proof is constructed from the root node of the binary tree to the node spent, which is well-known very costly. Au *et al.* constructed a divisible e-cash [9] using bounded accumulators. The efficiency of the computation and the storage is improved in the spending protocol. Unfortunately, it does not fulfill unforgeability. In order to obtain unforgeability, Canard and Gouget proposed a divisible e-cash scheme [10]. The number of the accumulator is proportional to the level number of the binary tree in the withdrawal protocol.

Transferability is the other important property. Okamoto and Ohta proposed two transferable e-cash systems [30, 26] which only provide weak anonymity. Chaum and Pedersen [21] analyzed the size of the transferred e-cash. They claimed that it is impossible to transfer a coin without increasing its size. Later, Canard *et al.* proposed an anonymous transferable e-cash system [27], and analyzed the anonymity [20] in transferable e-cash.

The conditional e-cash is another interesting branch in e-cash. Shi *et al.* [25] firstly introduced the definition of the conditional e-payments. The payer anonymously spends the e-cash to the payee. The payee then transfers the e-cash to the next user or deposits the e-cash to the bank. The disadvantage of this scheme is that the bank has to be on-line, and that it depends on the expensive cut-and-choose techniques. Blanton [17] improved the efficiency of the conditional e-payments, then he instantiated it using zero-knowledge proof, CL signature and verifiable encryption. However, the payer decides whether he has already owned the coin which he has received. The identity of the last payee is known by the bank. Moreover, Blanton left an open problem which the last payee is not unlinkable in the spending and deposit protocol.

Groth and Sahai constructed the first efficient non-interactive proof system [24] which considers a large class of statements over bilinear group. It is witness indistinguishable i.e., any adversary cannot distinguish which witness is used by the user. The proof can be randomized to update the NIZK proof. Based on Groth-Sahai proof system, Fuchsbauer presented commuting signatures and verifiable encryption [22]. It preserves its public verifiability, and allows anyone to encrypt a message and the corresponding signature. The signer who is given a commitment to a message creates a verifiably encrypted signature on the committed message.

The security of schemes mentioned above are proved in the random oracle model. It is known that some schemes [13, 12] proven secure in the random oracle model, are not secure in the standard model, or can not be instantiated. Belenkiy, Chase, Kohlweiss and Lysyanskaya proposed a compact e-cash system [31] with non-interactive spending in the standard model. This scheme is based on P-signature, simulatable verifiable random functions and Groth-Sahai proofs systems. Fuchsbauer *et al.* constructed the first practical transferable constant-size fair e-cash [23] 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. Blazy *et al.* gave a similar construction [18] with stronger anonymity. More recently, Izabachene and Libert built the first divisible e-cash [11] in the standard model. This scheme uses a new method to split the wallet and authenticate node. Unfortunately, the scheme has very low efficient.

1.2 Optimal Anonymity Properties

Optimal anonymity properties are firstly divided into five levels: Weak Anonymity (WA), Strong Anonymity (SA), Full Anonymity (FA) and two types of restricted Perfect Anonymity (PA_1 and PA_2) by Canard and Gouget [20]. Then Blazy and Canard *et al.* [18] modified the terminologies of Full Anonymity (FA) and Perfect Anonymity (PA_1 and PA_2), and defined the observe-then-receive full anonymity (FA), spend-then-observe full anonymity (PA_1) and spend-then-receive full anonymity (PA_2) as OtR-FA, StO-FA and StR-FA respectively. The specific definitions are given as follows.

- Weak Anonymity (WA): An adversary is not able to link a spending to a withdrawal in a transaction. However, the adversary may know if two spends are done by the same user or not.
- Strong Anonymity (SA): An adversary can not decide if two transactions are done by the same user or not. However, the adversary may recognize a coin that he has already observed during previous spends.
- Observe-then-Receive Full Anonymity (OtR-FA): An adversary can not link a coin he has received to a previously observed coin between honest users. However, the adversary may be able to recognize a coin he has already owned.
- Spend-then-Observe Full Anonymity (StO-FA): An adversary can not link a coin he has already owned to a observed coin between honest users.
- Spend-then-Receive Full Anonymity (StR-FA): An adversary can not link two coins he has received.

In this paper, we use the definition of anonymity introduced by Blazy and Canard *et al.* [18]. If the scheme fulfils the OtR-FA property, it fulfils SA property. If the scheme fulfils the SA property, it fulfils WA property. Thus, the relation is OtR-FA $\Rightarrow$ SA $\Rightarrow$ WA [20]. However, OtR-FA, StO-FA and StR-FA are three separate properties [20] which are not comparable. Therefore, if the anonymous transferable conditional e-cash scheme satisfies OtR-FA, StO-FA and StR-FA, it achieves optimal anonymity.

1.3 Our Contribution

In this paper we propose an optimally anonymous and transferable conditional e-cash based on Groth-Sahai proofs [24] and the commuting signatures [22] in the standard model. Our contributions are listed as follows:

- A publisher is introduced to publish the outcomes. We firstly give the formal definition for the publisher.
- We solve an open problem introduced by Blanton [17], which is that the identity of payee is unlinkable in the spending and deposit protocol.
- The first optimally anonymous and transferable conditional e-cash is presented in the standard model. Optimal anonymity properties, i.e., OtR-FA, StO-FA and StR-FA, are achieved.
- We compare the efficiency the new protocol and the existing protocols [25, 17], and find that our protocol is the most efficient one.

1.4 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 for this scheme. The main protocol is proposed in Section 5. The security analysis is given in Section 6. And Section 7 concludes the paper.

2 Preliminaries

This section introduces some preliminaries which will be used in this paper.

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 $\hat{e} : G_1 \times G_2 \rightarrow G_3$ 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 $a, b \in \mathbb{Z}_p$ we have $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$.
- We can efficiently compute group operations, compute the bilinear map and decide membership.

2.2 $Diffie - Hellman\ pair$

A pair $(x, y) \in G_1 \times G_2$ is defined as a $Diffie - Hellman\ pair$ [22], if there exists $a \in \mathbb{Z}_p$ such that $x = g^a, y = h^a$, where g, h generate G_1 and G_2 respectively. We denote the set of $\mathcal{DH}$ pairs by $\mathcal{DH}_a = \{(g^a, h^a) | a \in \mathbb{Z}_p\}$.

2.3 Mathematical Assumptions

The security of this scheme is based on the following existing mathematical assumptions, i.e., the Symmetric External Diffie-Hellman (SXDH) [24] and the asymmetric double hidden strong Diffie-Hellman assumption (q-ADH-SDH) [15].

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, 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. For random a, b, $g_1, g_1^a, g_1^b \in G_1$ and $g_2, g_2^a, g_2^b \in G_2$ are given, it is hard to distinguish g_1^{ab} and g_2^{ab} from a random element from G_1 and G_2 respectively.*

Definition 2. ***(q-ADH-SDH)***. *Let $g, f, k \in G_1, h \in G_2$ and $x, c_i, v_i \in Z_p$ be random. Given $(g, f, k, g^x; h, y = h^x)$ and*

$$(a_i = (k \cdot g^{v_i})^{\frac{1}{x+c_i}}, b_i = f^{c_i}, d_i = h^{c_i}, u_i = g^{v_i}, w_i = h^{v_i})$$

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

$$\hat{e}(a, y \cdot d) = \hat{e}(k \cdot u, h), \hat{e}(b, h) = \hat{e}(f, d), \hat{e}(u, h) = \hat{e}(g, w).$$

2.4 Useful Tools

Groth-Sahai Proofs. Groth and Sahai [24] constructed the first NIZK proof systems. They prove a large class of statements in the context of groups with bilinear map in the standard model. In order to prove the statement, the prover firstly commits to group elements. Then the prover produces the proofs and sends the commitments and the proofs to the verifier. And last the verifier verifies the correctness of the proof.

In this paper, SXDH-based commitments are used to commit to group elements. The simple description of SXDH-based commitments is given in the following.

$SXDH-based\ commitments$.

Setup. On input the public parameter $pp = (p, G_1, G_2, G_3, \widehat{e}, g, h)$, choose $\alpha_1, \alpha_2, t_1, t_2 \in \mathbb{Z}_p$. The output is the commitment key $ck = (pp, \mathbf{u}_1, \mathbf{u}_2, \mathbf{v}_1, \mathbf{v}_2)$, where $\mathbf{u}_1 = (u_{1,1}, u_{1,2}) = (g, g^{\alpha_1}), \mathbf{u}_2 = (u_{2,1}, u_{2,2}) = (g^{t_1}, g^{\alpha_1 t_1}), \mathbf{v}_1 = (v_{1,1}, v_{1,2}) = (h, h^{\alpha_1})$ and $\mathbf{v}_2 = (v_{2,1}, v_{2,2}) = (h^{t_1}, h^{\alpha_1 t_1})$.

Commit. Define the commitment to a group element $X \in G_1$ as

$$c_X = Com(ck, X, \mathbf{r} = (r_1, r_2)) = (u_{1,1}^{r_1} \cdot u_{2,1}^{r_2}, X \cdot u_{1,2}^{r_1} \cdot u_{2,2}^{r_2}),$$

where $r_1, r_2 \in \mathbb{Z}_p$. So the commitment to $Y \in G_2$ is

$$c_Y = Com(ck, Y, \mathbf{s} = (s_1, s_2)).$$

where $s_1, s_2 \in \mathbb{Z}_p$.

Randomization to commitment. Define the randomization to the commitment c_X as

$$RdCom(ck, c_X, \mathbf{r}') = c_X \odot Com(ck, 1, \mathbf{r}') = (c_{X,1} \cdot u_{1,1}^{r_1'} \cdot u_{2,1}^{r_2'}, c_{X,2} \cdot u_{1,2}^{r_1'} \cdot u_{2,2}^{r_2'}),$$

where $\mathbf{r}' = (r_1', r_2'), r_1', r_2' \in \mathbb{Z}_p$, $\odot$ denotes component-wise multiplication.

To prove relations satisfied by the associated plaintexts, SXDH-based Groth-Sahai proofs are used. The simple description of SXDH-based Groth-Sahai proofs is given in the following.

$SXDH-based\ Groth\text{-}Sahai\ Proof$.

Using SXDH-based commitments, Groth and Sahai construct NIZK proofs. It asserts that a set of committed values satisfies the pairing product equation. The pairing product equation is denoted as

$$E(X_1, \cdots, X_m; Y_1, \cdots, Y_n) : \prod_{i=1}^{n} \hat{e}(A_j, Y_j) \prod_{i=1}^{m} \hat{e}(X_i, B_i) \prod_{i=1}^{m} \prod_{j=1}^{n} \hat{e}(X_i, Y_j)^{\gamma_{i,j}} = y,$$

where $X_i, A_j \in G_1, Y_j, B_i \in G_2, \gamma_{i,j} \in \mathbb{Z}_p$ for $1 \leq i \leq m, 1 \leq j \leq n$, and $y \in G_3$. We use E as a shorthand for the above pairing product equation.

Following the definition [22], we prove that $X_i \in G_1, Y_i \in G_2$ satisfy the above equation E. The proof is defined as $Prove(ck, E, (X_i, \mathbf{r}_i)_{i=1}^m, (Y_j, \mathbf{s}_j)_{j=1}^n; Z)$, where $\mathbf{r}_i, \mathbf{s}_j \in \mathbb{Z}_p^2$, and $Z \in \mathbb{Z}_p^{2\times 2}$ is the internal randomness.

Randomization to proof. It is similar to the randomization to the commitment. We random the proof by replacing the internal randomness Z. Therefore, the randomization to proof is defined as

$$RdProve(ck, E, (X_i, \mathbf{r}_i + \mathbf{r}'_i)_{i=1}^m, (Y_j, \mathbf{s}_j + \mathbf{s}'_j)_{j=1}^n; Z'),$$

where $\mathbf{r}'_i$, $\mathbf{s}'_j \in \mathbb{Z}_p^2$, $\mathbf{r}'_i$ and $\mathbf{s}'_j$ are the randomness of the commitments $c_i = Com(ck, X_i, \mathbf{r}'_i)$ and $c_j = Com(ck, Y_j, \mathbf{s}'_j)$. The Groth-Sahai proof is witness indistinguishability. It guarantees the anonymity of the payers and payees during the withdrawal protocol, spending protocol and deposit protocol. The randomization of the commitments and corresponding proofs provides unlinkability of transferable conditional e-cash.

Commuting Signatures. Commuting signatures [22] combines a signature scheme, an encryption scheme and a proof system. A signer can encrypt both signature and message and prove validity. Using the commuting signature, the signer can create a verifiably encrypted signature on the encrypted message. Thus the signing and the encrypting commute. The commitment and the verification key can be modified by *RdCom*. Meanwhile, the corresponding proof can also be updated by *RdProve*. This paper instantiates the signature scheme, encryption scheme and proof system with Structure-Preserving signature (SP-signature) [15, 29], SXDH-based Groth-Sahai commitment and Groth-Sahai proof system [24] respectively. We review two results of [22] relevant to this paper in the following.

SigCom. Given a commitment c_M to a message M and a signing key sk, this algorithm allows a signer to make a commitment c_σ to a signature σ on M under sk, and a proof that the content of c_σ is a valid signature on the message committed in c_M. The message space is a Diffie-Hellman pair.

In the following, we simply give the description of the signature of a message, the commitments to a signature on a committed message and a proof of validity. A Diffie-Hellman pair (M, N) is committed.

Setup. The input is $p, G_1, G_2, G_3, \widehat{e}, g, h$. The output is some additional generators $f, k, t \in G_1$. The message space is a Diffie-Hellman pair $(M = g^m, N = h^m)$ for $m \in \mathbb{Z}_p$.

KeyGen. The input is $x \in \mathbb{Z}_p$. The output is a verification key pair $vk = (X = g^x, Y = h^x)$ for $sk = x$.

*Commit.*The public parameters are $(ck, p, G_1, G_2, G_3, \widehat{e}, g, h, f, k, t)$. ck is the user's commitment key. The user sends the following commitments and proofs to the signer, $c_m = (c_M, c_N, \pi_M, c_P, c_Q, \pi_P, U, \pi_U)$, where $U = t^{\iota_1} \cdot M$, $\iota_1 \in \mathbb{Z}_p$, $P = g^{\iota_1}$ and $Q = h^{\iota_1}$. P and Q are auxiliary values for proof of c_m. The pairing product equations are

$$\begin{aligned} E_{\mathcal{DH}}(M, N) &: \hat{e}(g^{-1}, N)\hat{e}(M, h) = 1 \\ E_U(M, Q) &: \hat{e}(t^{-1}, Q)\hat{e}(M, h^{-1}) = \hat{e}(U, h)^{-1}. \end{aligned}$$

The first equation proves that (M, N) is a Diffie-Hellman pair. The second proves that U is constructed correctly. We use $E_{\mathcal{DH}}$ and E_U as a shorthand for the above equations. Thus, the corresponding proofs of the commitments are

$$\begin{aligned} \pi_M &\leftarrow Prove(ck, E_{\mathcal{DH}}, (M, \boldsymbol{\mu}), (N, \boldsymbol{\nu})), \\ \pi_P &\leftarrow Prove(ck, E_{\mathcal{DH}}, (P, \boldsymbol{\rho}), (Q, \boldsymbol{\varrho})), \\ \pi_U &\leftarrow Prove(ck, E_U, (M, \boldsymbol{\mu}), (Q, \boldsymbol{\varrho})), \end{aligned}$$

where $\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\rho}, \boldsymbol{\varrho} \in \mathbb{Z}_p^2$.

The user sends the commitment c_m of the message (M, N) to the signer.

Sign. The signer verifies the proofs. If these are OK, the signer chooses $c, r \in \mathbb{Z}_p$ and computes the "pre-signature"

$$s_0 = \{A = (k \cdot t^r \cdot U)^{\frac{1}{x+c}}, B = f^c, D = h^c, R' = g^r, S' = h^r\}$$

The signer constructs the following commitments to the signature s_0.

$$\begin{aligned} c_{s_0} = (c_A = Com(ck, A, \boldsymbol{\alpha}), c_B = Com(ck, B, \boldsymbol{\beta}), c_D = (ck, D, \boldsymbol{\delta}), \\ c_R = c_P \odot Com(ck, R', 0) = Com(ck, R, \boldsymbol{\rho}), \\ c_S = c_Q \odot Com(ck, S', 0) = Com(ck, S, \boldsymbol{\varrho})). \end{aligned}$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\delta} \in \mathbb{Z}_p^2$.

The corresponding proofs are $\{\pi_A, \pi_B, \pi_R\}$.

In order to complete the proof, it needs the following verification equations

$$\begin{aligned} E_A(A, M; S, D) &: \hat{e}(t^{-1}, S)\hat{e}(A, Y)\hat{e}(M, h^{-1})\hat{e}(A, D) = \hat{e}(k, h) \\ E_B(B; D) &: \hat{e}(f^{-1}, D)\hat{e}(B, h) = 1 \\ E_R(R; S) &: \hat{e}(g^{-1}, S)\hat{e}(R, h) = 1 \end{aligned}$$

We use E_A, E_B and E_R as a shorthand for the above equations respectively.

The proofs π_A, π_B and π_R attest that the values committed satisfy the above three equations.

To make the proof of π_A, another pairing product equation

$$E_{A^\dagger}(A; D) : \hat{e}(A, Y)\hat{e}(A, D) = 1$$

is given. We use $E_{A^\dagger}$ as a shorthand for the above equation. So the proofs of $\{\pi_A, \pi_B, \pi_R\}$ are

$$\begin{aligned} \pi'_A &\leftarrow \pi_U \odot Prove(ck, E_{A^\dagger}, (A, \boldsymbol{\alpha}), (h^c, \boldsymbol{\delta})) \\ \pi_A &\leftarrow RdProof(ck, E_A, (c_A, 0), (c_D, 0), (c_M, 0), (c_S, \boldsymbol{\varrho}'), \pi'_A) \\ \pi_R &\leftarrow RdProof(ck, E_R, (c_R, \boldsymbol{\rho}'), (c_S, \boldsymbol{\varrho}'), \pi_P) \\ \pi_B &\leftarrow Prove(ck, E_{\mathcal{DH}}, (f^c, \boldsymbol{\beta}), (h^c, \boldsymbol{\delta})). \end{aligned}$$

Obtain − Signature The user knows ι_1 and computes $R = R' \cdot g^t$ and $S = S' \cdot h^t$. At last, the user obtains a commitment signature $c_{s_0} = (c_A, c_B, c_D, c_R, c_S, \pi_A, \pi_B, \pi_R)$.

AdC_κ. This allows anyone to commit to the verification key. It also adapts a proof asserting that a commitment contains a valid signature on a new verification key.

Given a commitment c_m to a message pair (M, N), a commitment c_σ to a signature σ and a proof of validity to a verification key vk, the signer adapts the verification key and makes corresponding proof. Firstly, the signer makes a new commitment c_{vk} to a verification key $vk = (X = g^x, Y = g^y)$, where $x, y \in \mathbb{Z}_p$. By verifying the proof of $c_{vk} = (Com(ck, X, \boldsymbol{\xi}), Com(ck, Y, \boldsymbol{\psi}), Prove(ck, E_{\mathcal{DH}}, (X, \boldsymbol{\xi}), (Y, \boldsymbol{\psi})))$, anyone knows that the structure of the new verification key is correct, where $\boldsymbol{\xi}, \boldsymbol{\psi} \in \mathbb{Z}_p^2$. Then, a pairing product equation

$$E_{\widehat{A}}(A, M; S, Y, D) : \hat{e}(t^{-1}, S)\hat{e}(M, h)\hat{e}(A, Y)\hat{e}(A, D) = \hat{e}(k, h)$$

is given. We use $E_{\widehat{A}}$ as a shorthand for the above equation. And last, π_A is modified to

$$\pi_{\widehat{A}} = RdProof(ck, E_{\widehat{A}}, (c_A, 0), (c_M, 0), (c_S, 0), (Com(ck, Y, 0), \boldsymbol{\psi}), (c_D, 0), \pi_A).$$

Thus, new proof is $\pi = (\pi_{\widehat{A}}, \pi_B, \pi_R)$.

3 The Model

In this section, we introduces a publisher to publish two outcomes. Then, we give the definitions of the algorithms and security properties.

The anonymous transferable conditional e-cash needs two outcomes for the spending and deposit. Thus, we introduce a publisher to publish two outcomes. Shi and Blanton's schemes gave only informal arguments about the publisher. In this paper, a new algorithm Publish() and a oracle $\mathcal{O}_{Publ()}$ are introduced for the publisher. New algorithms DComGen(),PComGen() and JComGen() are used for generating the commitments key. To obtain optimal anonymity, the bank $\mathcal{B}$ is divided into $\mathcal{W}$ for the withdrawal phase and $\mathcal{D}$ for the deposit phase.

3.1 New Algorithms

The transferable conditional e-cash system consists of the conditional protocol, withdrawal protocol, spending (transferring) protocol, deposit protocol and identify procedure. The procedures are given as follows.

- ParamSetup(1^λ). It is a probabilistic algorithm. 1^λ is the input. The output is the public parameters $params$. λ is the security parameter. In the following, we assume that $params$ contains λ and that it is a default input of all other algorithms.
- WKeyGen(), DKeyGen(), JKeyGen(), UKeyGen(), PKeyGen(). They are probabilistic algorithms executed respectively by $\mathcal{W}, \mathcal{D}, \mathcal{J}, \mathcal{U}$ or $\mathcal{P}$. The outputs are $(pk_\mathcal{W}, sk_\mathcal{W}), (pk_\mathcal{D}, sk_\mathcal{D}), (pk_\mathcal{J}, sk_\mathcal{J}), (pk_\mathcal{U}, sk_\mathcal{U})$ and $(pk_\mathcal{P}, sk_\mathcal{P})$.
- DComGen(), JComGen(), PComGen(). They are probabilistic algorithms executed by $\mathcal{D}, \mathcal{J}$ or $\mathcal{P}$ respectively. The outputs are the commitment keys $(ck_\mathcal{D}, ek_\mathcal{D}), (ck_\mathcal{J}, ek_\mathcal{J})$ and (ck_{pr}, ek_{pr}).
- Withdraw$(\mathcal{U}(sk_\mathcal{U}, pk_\mathcal{U}, pk_\mathcal{W}, pk_\mathcal{J}, ck_\mathcal{D}, ck_\mathcal{J}), \mathcal{W}(sk_\mathcal{W}, pk_\mathcal{W}, pk_\mathcal{U}, ck_\mathcal{D}, ck_\mathcal{J}))$. It is an interactive protocol, in which $\mathcal{U}$ withdraws a transferable conditional coin from $\mathcal{W}$. At the end, $\mathcal{U}$ outputs a coin co_1 or $\perp$. $\mathcal{W}$ checks the public key of the user, and deducts a coin from the user and outputs a view $\mathcal{V}$ or $\perp$.
- Publish$(\mathcal{D}/\mathcal{U}, \mathcal{P}(ck_{pr}, ek_{pr}, ck_{pe}, ek_{pe}))$. It is an unilateral protocol between $\mathcal{P}$ and $\mathcal{D}/\mathcal{U}$. $\mathcal{P}$ generates two commitments for two outcomes. The user registers a commitment for outcome he agrees with. Then $\mathcal{P}$ sends the two commitments to $\mathcal{D}$ and $\mathcal{U}$. After $\mathcal{P}$ publishes the outcome, $\mathcal{P}$ sends the extraction key to the winner and $\mathcal{D}$.
- Spend$(\mathcal{U}_1(co_1, sk_{\mathcal{U}_1}, pk_\mathcal{D}, pk_\mathcal{J}, ck_\mathcal{D}, ck_\mathcal{J}), \mathcal{U}_2(sk_{\mathcal{U}_2}, pk_\mathcal{D}, pk_\mathcal{J}, ck_\mathcal{D}, ck_\mathcal{J}))$. It is an interactive protocol in which $\mathcal{U}_1$ spends/transfers the coin co_1 to $\mathcal{U}_2$. At the end, $\mathcal{U}_2$ outputs a coin co_2 or $\perp$, and $\mathcal{U}_1$ outputs ok or $\perp$.
- Deposit$(\mathcal{U}(co_1, sk_\mathcal{U}, pk_\mathcal{D}, pk_\mathcal{J}, ck_\mathcal{D}, ck_\mathcal{J}), \mathcal{D}(sk_\mathcal{D}, pk_\mathcal{D}, pk_\mathcal{J}, ck_\mathcal{D}, ck_\mathcal{J}))$. It is divided into two parts: exchanging and cashing. We assume that the payer $\mathcal{U}$ is the winner. In exchanging, $\mathcal{U}$ needs to cash back the e-cash from $\mathcal{D}$. $\mathcal{U}$ firstly spends co_1 to $\mathcal{D}$. Then $\mathcal{D}$ verifies whether the coin is correct. If the coin is correct, $\mathcal{D}$ deposits the coin to the database and sends a value $mo = (p_m, j_{pk_\mathcal{U}}, c_{s_m}, \pi_{c_m})$ to $\mathcal{U}$. In cashing, $\mathcal{U}$ updates the commitment and proof of mo, and sends a new value mo' to $\mathcal{D}$. At the end, $\mathcal{D}$ outputs OK.
- Identify$(co, co', ek_\mathcal{J})$. If co and co' are correct, the algorithm is executed by the judge. The output is the public key $pk_\mathcal{U}$ of double-spender. Otherwise, it outputs OK.

3.2 Security Properties

This section gives 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 assume that the adversary arbitrarily and adaptively queries to oracles. The oracles are defined as follows.

- $\mathcal{O}_{Setup()}$. This oracle allows $\mathcal{A}$ to add a new user into the system, or to corrupt an honest user. When $\mathcal{A}$ interacts with the oracle, $\mathcal{A}$ obtains the key of the user or the bank. If an honest user is corrupted, the secret key is $\perp$.
- $\mathcal{O}_{With()}$. This oracle plays the role of the bank or the user in the withdrawal protocol. $\mathcal{A}$ can withdraw a conditional e-cash from the oracle acting the bank. He can also issue an e-cash to the oracle acting the user.
- $\mathcal{O}_{Spend()}$. This oracle enables $\mathcal{A}$ to act a payee, and then receive a conditional e-cash. $\mathcal{A}$ can also act a payer to spend a conditional e-cash.
- $\mathcal{O}_{Depo()}$. This oracle plays the role of the bank or the user in the deposit protocol. $\mathcal{A}$ can obtain a conditional e-cash from the oracle acting the user, or spend an e-cash to the oracle acting the bank.
- $\mathcal{O}_{ODepo()}$. This oracle permits the adversary to observe the transaction in the deposit protocol. He can not receive the output from the oracle.

- $\mathcal{O}_{Idt()}$. This oracle plays the role of the judge in the identity procedures. $\mathcal{A}$ can submit two e-cash to the oracle and obtain the identity of the double-spender.
- $\mathcal{O}_{Publ()}$. This oracle allows the publisher to extract the secret value. Then $\mathcal{A}$ can obtain the outcome by interacting with the oracle.

In this paper, the publisher and the judge are trust organizers. The judge can not recover the identity of an honest user except that the bank gives two double-spending e-cash. The publisher only publishes and announces the outcome correctly, and can not extract the outcome before publishing the outcome. Although the judge can trace coins and users, this is the requirement of the fair e-cash [28]. We require all the length of the conditional e-cash are same. Note that the publisher and the judge are honest, thus the adversary $\mathcal{A}$ can not obtain any information from the publisher and the judge. More precisely, in the conditional protocol, the publisher publishes two conditional commitments; in the Identify(), the judge recovers the identity of the double-spender. However, they can not supply any information for $\mathcal{A}$ to recover the identity of an honest user or generate a double-spending on the same outcome. The security properties are defined formally as follows.

Anonymity. This scheme achieves optimal anonymity and satisfies at the same time OtR-FA, StO-FA and StR-FA. The anonymity guarantees that no coalition of users, publisher and judge can distinguish which user executes the spending protocol. $pk = (\mathcal{RU}, \mathcal{SU})$ represents the public keys of users. The $\mathcal{RU}$ are the public keys of the users who have received a coin from $\mathcal{A}$. The $\mathcal{SU}$ are the public keys of the users who have only spent a coin to $\mathcal{A}$. The judge is honest, thus he can not recover the identity of an honest user.

Firstly, the security description of OtR-FA is given as follows.

- (*Initialization Phase.*) $\mathcal{A}$ runs $ParamSetup(1^\lambda)$ and obtains the public parameters $params$, 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 $pk_{\mathcal{B}}, pk_{\mathcal{J}}$ and $pk_{\mathcal{P}}$ to $\mathcal{C}$, and keeps $sk_{\mathcal{B}}, sk_{\mathcal{J}}$ and $sk_{\mathcal{P}}$ to herself. In order to simplify description, $(pk_{\mathcal{B}}, sk_{\mathcal{B}})$ includes $(pk_{\mathcal{W}}, sk_{\mathcal{W}})$ and $(pk_{\mathcal{D}}, sk_{\mathcal{D}})$.
- (*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 $\mathcal{O}_{Setup()}$. For each $\mathcal{O}_{With}()$ and $\mathcal{O}_{Spend}()$, $\mathcal{A}$ can act as bank or user in the withdrawal protocol or spending protocol. In deposit protocol, $\mathcal{A}$ interacts with $\mathcal{O}_{Depo()}$ and $\mathcal{O}_{ODepo()}$, and obtains any output of a deposit procedure. $\mathcal{A}$ can obtain the identity of the user from $\mathcal{O}_{Idt()}$, or obtain the extraction key from the oracle $\mathcal{O}_{Publ()}$. In OtR-FA property, the public keys obtained are from $\mathcal{SU}$.
- (*Challenge Phase.*) $\mathcal{C}$ randomly chooses two public keys $pk_{\mathcal{U}_0}$ and $pk_{\mathcal{U}_1}$, which come from $\mathcal{SU}$ in *Probing Phase*. Then $\mathcal{C}$ at random chooses one of them. $\mathcal{C}$ uses the public key $pk_{\mathcal{U}_i}$ $(i = 0/1)$ interacting with $\mathcal{A}$. The two public keys come from $\mathcal{SU}$. Thus, $\mathcal{A}$ only observes the public keys $pk_{\mathcal{U}_0}$ and $pk_{\mathcal{U}_1}$. $\mathcal{A}$ acting the bank or the user interacts with $\mathcal{C}$. $\mathcal{A}$ can not ask $\mathcal{C}$ to over-spend any coin, and can also not query on $\mathcal{O}_{Idt()}$ 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 StO-FA. Every phase is similar to OtR-FA except that the two public keys obtained are any keys in the *Probing Phase*, and $\mathcal{A}$ only observes the spending between two honest users in *Challenge Phase*. In *Challenge Phase*, $\mathcal{A}$ can not directly spend with $\mathcal{C}$. This prevents $\mathcal{A}$ from receiving co_0 or co_1. This is the requirement of the StO-FA property.

For the security description of StR-FA. The every phase is similar to OtR-FA except that the two public keys obtained are any keys in the *Probing Phase*, and $\mathcal{A}$ only interacts with $\mathcal{O}_{ODepo()}$ in the *Probing Phase*. The limited interaction makes sure that the adversary only observes the transaction in the deposit protocol. This is the requirement of StR-FA property.

For all non-uniform polynomial time adversary $\mathcal{A}$, 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 is an anonymous transferable conditional e-cash system.

If $Adv^{anon}_{TCE,\mathcal{A}}$ is negligible for any polynomial-time adversary $\mathcal{A}$, this scheme is anonymous.

Unforgeability. No coalition of users and merchants can deposit more coins than the coins that they withdrew. We give the following experiment and definition.

- (*Initialization Phase.*) $\mathcal{C}$ runs the $ParamSetup(1^\lambda)$ and obtains the public parameters $params$, the key pairs $(pk_{\mathcal{B}}, sk_{\mathcal{B}})$, $(pk_{\mathcal{J}}, sk_{\mathcal{J}})$ and $(pk_{\mathcal{P}}, sk_{\mathcal{P}})$. Then $\mathcal{C}$ sends $pk_{\mathcal{B}}, pk_{\mathcal{J}}$ and $pk_{\mathcal{P}}$ to $\mathcal{A}$ and keeps $sk_{\mathcal{B}}, sk_{\mathcal{J}}$ and $sk_{\mathcal{P}}$ to herself.
- (*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 $\mathcal{O}_{Setup()}$. The value of e-cash possessed by $\mathcal{A}$ is defined by v_{ua} and initialized with zero. For each $\mathcal{O}_{With}()$, $\mathcal{A}$ acts as user and withdraws a conditional e-cash of value v_{ui} in the withdrawal protocol. In the transferring protocol, $\mathcal{A}$ can perform queries to the oracle $\mathcal{O}_{Spend()}$. $\mathcal{A}$ acting as payer transfers an e-cash of value v_{uo} to the payee, or acting as payee receives an e-cash of value v^1_{ui}. $\mathcal{A}$ interacts with $\mathcal{O}_{Depo()}$ and deposits the e-cash of value v_{de} to $\mathcal{C}$ acting as the bank in the deposit protocol. $\mathcal{A}$ can obtain the identity of the user from $\mathcal{O}_{Idt()}$, or obtain the extraction key from the oracle $\mathcal{O}_{Publ()}$. At last, $\mathcal{A}$ obtains the e-cash of value $v_{ua} = v_{ui} + v^1_{ui} - v_{uo} - v_{de}$.
- (*End Game Phase.*) $\mathcal{A}$ wins the game if it can deposit $v_{ua} + 1$ to $\mathcal{C}$.

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

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

where TCE is an anonymous transferable conditional e-cash system.

If $Adv^{unfor}_{TCE,\mathcal{A}}$ is negligible for any polynomial-time adversary $\mathcal{A}$, this scheme is unforgeable.

Identification of Double-spender. It guarantees that coalition of users and merchants can not double-spend a coin with the same serial number. After the two conditionals commitments are publisher, the commitments can not be changed by $\mathcal{A}$. The publisher and judge are honest, thus they can not supply any information with $\mathcal{A}$ to generate a double-spending. We give the following experiment and definition.

- (*Initialization Phase.*) The *Initialization Phase* is similar to that in the unforgeability property.
- (*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 $\mathcal{O}_{Setup()}$. $\mathcal{A}$ engages in the withdrawal protocol, spending protocol and the deposit protocol as many times as he likes. If $\mathcal{A}$ deposits the same serial number coin twice, and the output of the algorithm Identify() is OK, he must break the unforgeability of the commuting signature and the soundness and witness indistinguishability of Groth-Sahai proofs.
- (*End Game Phase.*) $\mathcal{A}$ wins the game if it can deposit a coin twice, the output of the Deposit() is OK and the Identify() cannot output the public key.

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

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

where TCE is an anonymous transferable conditional e-cash system.

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

Exculpability. No coalition of the banks and users can accuse an honest user of double-spending a coin. The judge and the publisher are honest, thus they can not generate an honest user's e-cash and frame the user of double-spending. We give the following experiment and definition.

- (*Initialization Phase.*) The *Initialization Phase* is similar to that in the anonymity property.

– (*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 $\mathcal{O}_{Setup()}$. $\mathcal{C}$ runs the withdrawal protocol with $\mathcal{A}$ acting as the bank to obtain coins. Then $\mathcal{A}$ acts as the merchant and runs the spending protocol with $\mathcal{C}$ acting the user. $\mathcal{A}$ also acts the bank and accepts a deposit from $\mathcal{C}$. At last, if $\mathcal{A}$ can forge a conditional e-cash of the same serial number spent by $\mathcal{C}$, he frames an honest user of double-spending.
– (*End Game Phase.*) $\mathcal{A}$ wins the game if it can forge a e-cash of the same serial number and prove the spending is correct.

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

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

where TCE is an anonymous transferable conditional e-cash system.

If $Adv^{excu}_{TCE,\mathcal{A}}$ is negligible for any polynomial-time adversary $\mathcal{A}$, this scheme is exculpability.

4 General Description

In a transferable conditional e-cash, the payer anonymously transfers an e-cash until the outcome of the condition is published. The e-cash is valid to both the payer and the last payee, and only one of them can deposit the e-cash. When the outcome is published, if the outcome is favorable for the payer, the payer is the winner. Then the publisher sends the extraction key to the winner by an authenticated and secure channel. If a user 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, otherwise the e-cash is deposited to the bank.

The transferable conditional e-cash consists of the conditional protocol, the withdrawal protocol, spending (transferring) protocol, deposit protocol and the identify procedure. The message space of the instantiation signature of the commuting signature is the Diffie-Hellman pair. Thus, a coin is represented by a unique chain of $\mathcal{DH}$ pairs $s = \mathcal{DH}_n||\mathcal{DH}_m||\mathcal{DH}_{n_0}||\mathcal{DH}_{n_1}||\cdots||\mathcal{DH}_{n_i}$ for $n, m, n_0, \cdots, n_i \in \mathbb{Z}_p$. The n, m are chosen by the publisher, while $n_0, n_1, \cdots, n_i$ are randomly chosen by a consecutive owner of the coin. We provide a new algorithm to construct the transferable conditional e-cash system based on the outcome of the condition. The general description is given as follows.

The payer $\mathcal{U}_1$ firstly withdraws an e-cash co_1 from the bank, and decides to spend the e-cash to the payee $\mathcal{U}_2$ based on a condition. In order to protect the identities of $\mathcal{U}_1$, he randomizes co_1 as co'_1. Then $\mathcal{U}_1$ spends co'_1 to $\mathcal{U}_2$, and $\mathcal{U}_2$ continues to spend the e-cash to $\mathcal{U}_3$. Because $\mathcal{U}_2$ generates the new e-cash co_2 which includes the co'_1. However, the verification key of $\mathcal{U}_1$ is public to verify the correctness of $\mathcal{U}_1$'s commitment signature. Thus, we modify $\mathcal{U}_1$'s verification key and corresponding proof using commuting signature technology to protect the verification key of $\mathcal{U}_1$. At last, $\mathcal{U}_i$ do not want to spend the e-cash to others, he deposits the e-cash to the bank. If he directly deposits the e-cash to the bank, the identity of $\mathcal{U}_i$ would be recovered by the information in the bank. Therefore, the deposit protocol is divided into two parts, i.e., exchanging and cashing to obtain the anonymity of the last user.

In this 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 $\mathcal{U}_i$. This goal is achieved by introducing a publisher who 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 user by an authenticated and secure channel. The publisher is very important, since he publishes the conditions of the event. The judge is introduced to recover the identity of double-spender. The user registers the public key with the judge.

5 Transferable Conditional E-cash

Transferable conditional e-cash allows the user to spend a conditional e-cash to others based on the outcome in the future. In the following, we give the details of this scheme.

5.1 Setup

We choose bilinear groups (G_1, G_2, G_3) of order $p > 2^\lambda$. The elements $g \in G_1$ and $h \in G_2$ generate G_1 and G_2 respectively. It also generates a common reference string $(g, h, \mathbf{u}_1, \mathbf{u}_2, \mathbf{v}_1, \mathbf{v}_2)$ for the perfectly soundness reference. Therefore, the public parameter is $params = \{(G_1, G_2, G_3), (g, h, \mathbf{u}_1, \mathbf{u}_2, \mathbf{v}_1, \mathbf{v}_2)\}$.

The bank $\mathcal{W}$, the detector $\mathcal{D}$, the judge $\mathcal{J}$ and the publisher $\mathcal{P}$ respectively generate key pairs $(pk_\mathcal{W}, sk_\mathcal{W})$, $(pk_\mathcal{D}, sk_\mathcal{D})$, $(pk_\mathcal{J}, sk_\mathcal{J})$ and $(pk_\mathcal{P}, sk_\mathcal{P})$ for commuting signature. $\mathcal{W}$, $\mathcal{J}$ and $\mathcal{P}$ respectively choose a random group element $t, t_\mathcal{J}, t_\mathcal{P}$ for the public parameter of commuting singature.The double-spending detector $\mathcal{D}$ generates a pair of commitment/extraction key $(ck_\mathcal{D}, ek_\mathcal{D})$ which is used to commit the serial number of the e-cash. The publisher gives four pairs of commitment/extraction keys $(ck_{pr}, ek_{pr}), (ck_{pe}, ek_{pe}), (ck'_{pr}, ek'_{pr})$ and (ck'_{pe}, ek'_{pe}). (ck_{pr}, ek_{pr}) is used to commit the coin for the payer. The second key pair (ck_{pe}, ek_{pe}) is used to commit the coin for the payee. (ck'_{pr}, ek'_{pr}) and (ck'_{pe}, ek'_{pe}) are used for the proof. The publisher also generates the third commitment/extraction key $(ck_\mathcal{P}, ek_\mathcal{P})$ which is used to commit the serial number of the e-cash. The judge gives two pairs of commitment/extraction keys $(ck_\mathcal{J}, ek_\mathcal{J})$ and (ck_{sp}, ek_{sp}). The former is used for identification of double-spender. The latter is used for the proof of this scheme. The bank maintains a database DB which is used to save the spent e-cash. DB' is introduced to save the exchanging cash mo in the deposit protocol. These databases are initialized to be empty.

Each user $\mathcal{U}_i$ generates key pairs $pk_{\mathcal{U}_i} = ((g, h)^{sk_{\mathcal{U}_i}}, sk_{\mathcal{U}_i})$ for commuting signature, where $sk_{\mathcal{U}_i} \in \mathbb{Z}_p$. Each $\mathcal{U}_i$ also registers their public keys $pk_{\mathcal{U}_i} = (g^{sk_{\mathcal{U}_i}}, h^{sk_{\mathcal{U}_i}}) = (M, N)$ to the judge. $\mathcal{U}_1$'s register is given as follows.

The user firstly generates the following commitments and corresponding correctness proofs to her public key. To simplify, the commitment $j_{pk_{\mathcal{U}_1}}$ contains the commitments and correctness proofs for the committed value.

$$j_{pk_{\mathcal{U}_1}} = (c_M^{ck_\mathcal{J}}, c_N^{ck_\mathcal{J}}, c_P^{ck_\mathcal{J}}, c_Q^{ck_\mathcal{J}}, U = t_\mathcal{J}^{\iota_3} \cdot g^{sk_{\mathcal{U}_1}}, \pi_{j_{pk_{\mathcal{U}_1}}} = (\pi_M^{ck_\mathcal{J}}, \pi_P^{ck_\mathcal{J}}, \pi_U^{ck_\mathcal{J}}))$$

Then the user sends her public key, the commitments and proofs to the judge. The judge detects whether the public key, the commitments and proofs are correct. If these are correct, the judge generates a membership certificate $c_{s_{\mathcal{U}_1}}^{sk_\mathcal{J}}$ by algorithm $SigCom$, and sends the membership certificate $c_{s_{\mathcal{U}_1}}^{sk_\mathcal{J}} = (c_{A_{\mathcal{U}_1}}, c_{B_{\mathcal{U}_1}}, c_{D_{\mathcal{U}_1}}, c_{R_{\mathcal{U}_1}}, c_{S_{\mathcal{U}_1}}, \pi_{s_{\mathcal{U}_1}}^{sk_\mathcal{J}} = (\pi_{A_{\mathcal{U}_1}}, \pi_{B_{\mathcal{U}_1}}, \pi_{R_{\mathcal{U}_1}}))$ to the user, otherwise the judge aborts the protocol.

5.2 The Conditional Protocol

The conditional protocol allows the publisher to publish two commitments of two outcomes as described in Figure 1. The details of the conditional protocol are given as follows.

A commitment to message M is defined as p_M using $ck_\mathcal{P}$. The publisher chooses $n, m \in \mathbb{Z}_p$ for two outcomes. The publisher also generates two $Diffie-Hellman\ pairs$ ($\mathcal{DH}$ pairs) $\mathcal{DH}_n$ and $\mathcal{DH}_m$. Two commitments p_n and p_m are defined that $\mathcal{P}$ commits n and m using her commitment keys ck_{pr} and ck_{pe}. Two commitments $c_{g^n}^{ck_{pr}}$ and $c_{g^m}^{ck_{pe}}$ in p_n and p_m are defined that $\mathcal{P}$ commits g^n and g^m using her commitment keys ck_{pr} and ck_{pe} respectively. To make the security proof, two commitments[1] $\widetilde{p}_n$ and $\widetilde{p}_m$ are defined that $\mathcal{P}$ commits n and m using her commitment keys ck'_{pr} and ck'_{pe}. In the following, we give the protocol in detail.

1. To obtain two outcomes, $\mathcal{U}_1$ sends her public key $pk_{\mathcal{U}_1}$ and corresponding commitment $c_{pk_{\mathcal{U}_1}}$ to $\mathcal{P}$.

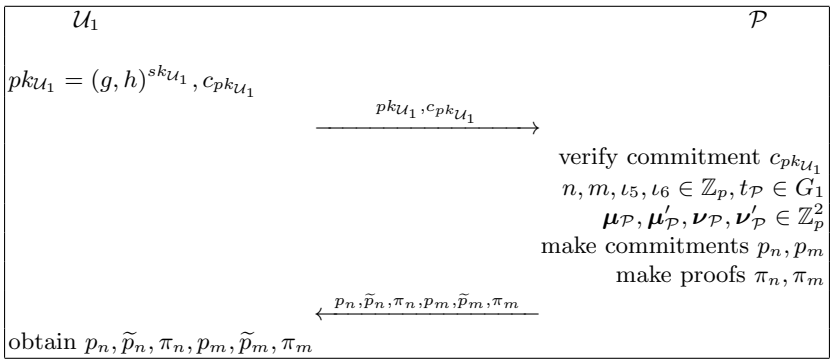

Fig. 1. $\mathcal{U}_1$ obtains two commitments of outcomes from $\mathcal{P}$.

2. $\mathcal{P}$ verifies $c_{pk_{\mathcal{U}_1}}$. If it is correct, $\mathcal{P}$ picks at random nonces $n, m, \iota_5, \iota_6 \in \mathbb{Z}_p, t_{\mathcal{P}} \in G_1, \boldsymbol{\mu}_{\mathcal{P}}, \boldsymbol{\mu}'_{\mathcal{P}} \in \mathbb{Z}_p^2$ and generates $\mathcal{DH}$ pairs $\mathcal{DH}_n = (g^n, h^n)$ and $\mathcal{DH}_m = (g^m, h^m)$. Then $\mathcal{P}$ generates two commitments and correctness proofs

$$p_n = (c_{g^n}^{ck_{pr}}, c_{h^n}^{ck_{pr}}, c_P^{ck_{pr}}, c_Q^{ck_{pr}}, U_n = t_{\mathcal{P}}^{\iota_5} \cdot g^n, \pi_{p_n} = (\pi_{g^n}^{ck_{pr}}, \pi_P^{ck_{pr}}, \pi_{U_n}^{ck_{pr}}))$$

and

$$p_m = (c_{g^m}^{ck_{pe}}, c_{h^m}^{ck_{pe}}, c_P^{ck_{pe}}, c_Q^{ck_{pe}}, U_m = t_{\mathcal{P}}^{\iota_6} \cdot g^m, \pi_{p_m} = (\pi_{g^m}^{ck_{pe}}, \pi_P^{ck_{pe}}, \pi_{U_m}^{ck_{pe}}))$$

using commitment keys ck_{pr} and ck_{pe} respectively. $\mathcal{P}$ also gives two commitments $\widetilde{p}_n$ and $\widetilde{p}_m$ which are introduced to prove the scheme completely. Meanwhile, two proofs π_n and π_m are given to prove that two committed values in p_n and p_n are equal. The proof π_n is given as follows.

$$\pi_n \leftarrow Prove(ck_{\mathcal{D}}; E_{eq}, (n, \boldsymbol{\mu}_{\mathcal{P}}), (n, \boldsymbol{\mu}'_{\mathcal{P}})),$$

where

$$E_{eq}(x,y) : \hat{e}(x, h^{-1})\hat{e}(y,h) = 1$$

for $x, y \in G_1$. This equation asserts that the variables x and y are same value.
The proof π_m is similar to the proof π_n.
$\mathcal{P}$ sends $\{p_n, \widetilde{p}_n, \pi_n, p_m, \widetilde{p}_m, \pi_m\}$ to $\mathcal{U}_1$ and the detector $\mathcal{D}$ by an authenticated and secure channel.

5.3 The Withdrawal Protocol

The withdrawal protocol allows $\mathcal{U}_1$ to withdraw a coin co from $\mathcal{W}$[2] as described in Figure 2. Two commitments j_M and d_M are defined that $\mathcal{J}$ and $\mathcal{D}$ commit a message M using $ck_{\mathcal{J}}$ and $ck_{\mathcal{D}}$. A commitment $\widetilde{j}_n$ is defined that $\mathcal{J}$ commits n using ck_{sp}. $\widetilde{j}_n$ is introduced to be an auxiliary value for our proof. The definitions of $c_{g^{n_1}}^{ck_{\mathcal{D}}}, c_{g^{n_1}}^{ck_{\mathcal{J}}}, c_M^{ck_{\mathcal{J}}}, c_N^{ck_{\mathcal{J}}}, c_P^{ck_{\mathcal{J}}}$ and $c_Q^{ck_{\mathcal{J}}}$ are similar to $c_{g^m}^{ck_{pe}}$. The commitment signatures $c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}}$ and $c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{W}}}$ are defined that $\mathcal{J}$ and $\mathcal{W}$ generate commitment signatures $c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}}$ and $c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{W}}}$ to the commitment $j_{pk_{\mathcal{U}_1}}$. Note that the definition of $c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}}$ is different from $c_{g^m}^{ck_{pe}}$. In $c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}}$, the symbol $s_{\mathcal{U}_1}$ is defined for a signature. Thus, $c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}}$ asserts that $\mathcal{U}_1$ generates a commitment signature $c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}}$. In the following, we give the protocol in detail.

[1] Groth-Sahai proofs are witness indistinguishable and a conditional e-cash is represented by randomizable extractable commitments, so the e-cash can not reveal any information about serial number, public keys and so on. Thus, $\widetilde{p}_n$ and $\widetilde{p}_m$ are introduced as auxiliary values for our proof.

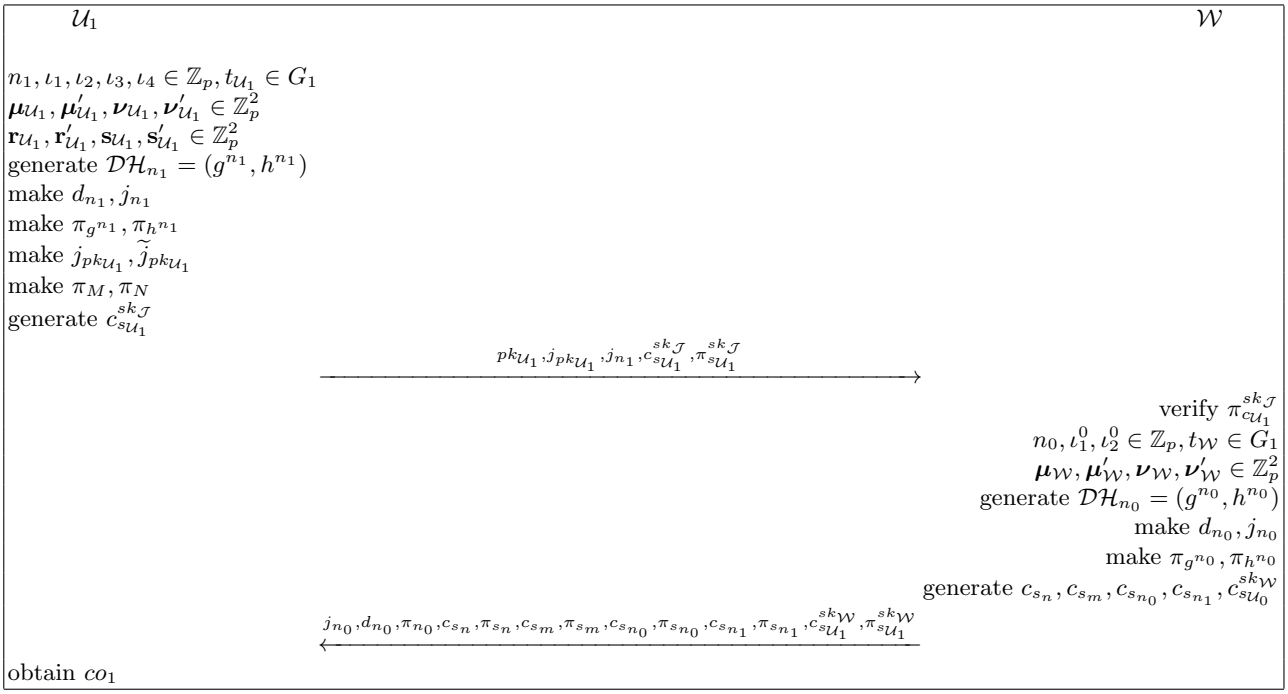

Fig. 2. $\mathcal{U}_1$ withdraws a conditional e-cash from $\mathcal{W}$.

1. $\mathcal{U}_1$ picks at random nonces $n_1 \in \mathbb{Z}_p, \iota_1, \iota_2, \iota_3, \iota_4 \in \mathbb{Z}_p, \boldsymbol{\mu}_{\mathcal{U}_1}, \boldsymbol{\mu}'_{\mathcal{U}_1}, \boldsymbol{\nu}_{\mathcal{U}_1}, \boldsymbol{\nu}'_{\mathcal{U}_1}, \mathbf{r}_{\mathcal{U}_1}, \mathbf{r}'_{\mathcal{U}_1}, \mathbf{s}_{\mathcal{U}_1}, \mathbf{s}'_{\mathcal{U}_1} \in \mathbb{Z}_p^2, t_{\mathcal{U}_1} \in G_1$ and generates a $\mathcal{DH}$ pair $\mathcal{DH}_{n_1} = (g^{n_1}, h^{n_1})$. Then $\mathcal{U}_1$ makes commitments and proofs

$$d_{n_1} = (c^{ck_{\mathcal{D}}}_{g^{n_1}}, c^{ck_{\mathcal{D}}}_{h^{n_1}}, c^{ck_{\mathcal{D}}}_{P}, c^{ck_{\mathcal{D}}}_{Q}, U_{d_{n_1}} = t^{\iota_1}_{\mathcal{U}_1} \cdot g^{n_1}, \pi_{d_{n_1}} = (\pi^{ck_{\mathcal{D}}}_{g^{n_1}}, \pi^{ck_{\mathcal{D}}}_{P}, \pi^{ck_{\mathcal{D}}}_{d_{n_1}}))$$

using the commitment key $ck_{\mathcal{D}}$, where

$$\begin{aligned}
c^{ck_{\mathcal{D}}}_{g^{n_1}} &= Com(ck_{\mathcal{D}}, g^{n_1}, \boldsymbol{\mu}_{\mathcal{U}_1}),\\
c^{ck_{\mathcal{D}}}_{h^{n_1}} &= Com(ck_{\mathcal{D}}, h^{n_1}, \boldsymbol{\nu}_{\mathcal{U}_1}),\\
c^{ck_{\mathcal{D}}}_{P} &= Com(ck_{\mathcal{D}}, P, \boldsymbol{\psi}_{\mathcal{U}_1}),\\
c^{ck_{\mathcal{D}}}_{Q} &= Com(ck_{\mathcal{D}}, Q, \boldsymbol{\chi}_{\mathcal{U}_1}),\\
\pi^{ck_{\mathcal{D}}}_{g^{n_1}} &\leftarrow Prove(ck_{\mathcal{D}}, E_{\mathcal{DH}}, (g^{n_1}, \boldsymbol{\mu}_{\mathcal{U}_1}), (h^{n_1}, \boldsymbol{\nu}_{\mathcal{U}_1})),\\
\pi^{ck_{\mathcal{D}}}_{P} &\leftarrow Prove(ck_{\mathcal{D}}, E_{\mathcal{DH}}, (P, \boldsymbol{\psi}_{\mathcal{U}_1}), (Q, \boldsymbol{\chi}_{\mathcal{U}_1})),\\
\pi^{ck_{\mathcal{D}}}_{U} &\leftarrow Prove(ck_{\mathcal{D}}, E_{U}, (g^{n_1}, \boldsymbol{\mu}_{\mathcal{U}_1}), (Q, \boldsymbol{\chi}_{\mathcal{U}_1})).
\end{aligned}$$

$c^{ck_{\mathcal{D}}}_{g^{n_1}}$ and $c^{ck_{\mathcal{D}}}_{h^{n_1}}$ are the commitments of g^{n_1} and h^{n_1}. $\pi^{ck_{\mathcal{D}}}_{g^{n_1}}$ is the proof that the two committed values in $c^{ck_{\mathcal{D}}}_{g^{n_1}}$ and $c^{ck_{\mathcal{D}}}_{h^{n_1}}$ are equal. $c^{ck_{\mathcal{D}}}_{P}$ and $c^{ck_{\mathcal{D}}}_{Q}$ are the commitments of P and Q, where $P = g^{\iota_1}$, $Q = h^{\iota_1}$. $\pi^{ck_{\mathcal{D}}}_{P}$ is the proof that the two committed values in $c^{ck_{\mathcal{D}}}_{P}$ and $c^{ck_{\mathcal{D}}}_{Q}$ are equal. $\pi^{ck_{\mathcal{D}}}_{U}$ proves that U is constructed correctly. To prove that two committed values in c_M and c_N are equal, the equation $E_{\mathcal{DH}}(M, N) : \hat{e}(g^{-1}, N)\hat{e}(M, h) = 1$ is verified in Groth-Sahai proofs. The equation $E_U(M, Q) : \hat{e}(t^{-1}, Q)\hat{e}(M, h^{-1}) = \hat{e}(U, h)^{-1}$ is used to verify that U is constructed correctly.

[2] To achieve the security property of StR-FA, $\mathcal{B}$ is divided to $\mathcal{W}$ and $\mathcal{D}$. Thus, $\mathcal{U}$ spending an e-cash to $\mathcal{W}$. This prevents the adversary $\mathcal{A}$ from acting as $\mathcal{D}$ [20].

Meanwhile, $\mathcal{U}_1$ makes commitments and proofs

$$j_{n_1} = (c_{g^{n_1}}^{ck_{\mathcal{J}}}, c_{h^{n_1}}^{ck_{\mathcal{J}}}, c_P^{ck_{\mathcal{J}}}, c_Q^{ck_{\mathcal{J}}}, U_{j_{n_1}} = t_{\mathcal{U}_1}^{\iota_2} \cdot g^{n_1}, \pi_{j_{n_1}} = (\pi_{g^{n_1}}^{ck_{\mathcal{J}}}, \pi_P^{ck_{\mathcal{J}}}, \pi_{U_{j_{n_1}}}^{ck_{\mathcal{J}}}))$$

using the commitment key $ck_{\mathcal{J}}$. The proof $\pi_{j_{n_1}}$ is similar to the proof $\pi_{d_{n_1}}$. When a double-spending happens, $\mathcal{J}$ extracts a $\mathcal{DH}$ pair $\mathcal{DH}_{n_1}$ to verify the double-spending. $\mathcal{U}_1$ also gives the following proof π_{n_1} that the two committed values in p_n and p_n are equal [23].

$$\pi_{g^{n_1}} \leftarrow Prove(ck_{\mathcal{D}}, ck_{\mathcal{J}}; E_{eq}, (g^{n_1}, \boldsymbol{\mu}_{\mathcal{U}_1}), (g^{n_1}, \boldsymbol{\mu}'_{\mathcal{U}_1})),$$
$$\pi_{h^{n_1}} \leftarrow Prove(ck_{\mathcal{D}}, ck_{\mathcal{J}}; E_{eq}, (h^{n_1}, \boldsymbol{\nu}_{\mathcal{U}_1}), (h^{n_1}, \boldsymbol{\nu}'_{\mathcal{U}_1})),$$

Moreover, $\mathcal{U}_1$ makes commitments and proofs

$$j_{pk_{\mathcal{U}_1}} = (c_M^{ck_{\mathcal{J}}}, c_N^{ck_{\mathcal{J}}}, c_P^{ck_{\mathcal{J}}}, c_Q^{ck_{\mathcal{J}}}, U_{j_{pk_{\mathcal{U}_1}}} = t_{\mathcal{U}_1}^{\iota_3} \cdot g^{sk_{\mathcal{U}_1}}, \pi_{j_{pk_{\mathcal{U}_1}}} = (\pi_M^{ck_{\mathcal{J}}}, \pi_P^{ck_{\mathcal{J}}}, \pi_{U_{j_{pk_{\mathcal{U}_1}}}}^{ck_{\mathcal{J}}}))$$

and

$$\widetilde{j}_{pk_{\mathcal{U}_1}} = (\widetilde{c}_M^{ck_{sp}}, \widetilde{c}_N^{ck_{sp}}, \widetilde{c}_P^{ck_{sp}}, \widetilde{c}_Q^{ck_{sp}}, U_{\widetilde{j}_{pk_{\mathcal{U}_1}}} = t_{\mathcal{U}_1}^{\iota_4} \cdot g^{sk_{U_1}}, \pi_{\widetilde{j}_{pk_{\mathcal{U}_1}}} = (\widetilde{\pi}_M^{ck_{sp}}, \widetilde{\pi}_P^{ck_{sp}}, \widetilde{\pi}_{U_{\widetilde{j}_{pk_{\mathcal{U}_1}}}}^{ck_{sp}}))$$

to the public key $pk_{\mathcal{U}_1} = (g^{sk_{\mathcal{U}_1}}, h^{sk_{\mathcal{U}_1}})$. The first commitment $j_{pk_{\mathcal{U}_1}}$ is used to recover the identity of the user when a double-spending happens. The second $\widetilde{j}_{pk_{\mathcal{U}_1}}$ is used for the security proof. The proofs $\pi_{j_{pk_{\mathcal{U}_1}}}$ and $\pi_{\widetilde{j}_{pk_{\mathcal{U}_1}}}$ are similar to the proof $\pi_{d_{n_1}}$. We need to prove that the committed value in $j_{pk_{\mathcal{U}_1}}$ and $\widetilde{j}_{pk_{\mathcal{U}_1}}$ are equal. $pk_{\mathcal{U}_1} = (M, N)$ and $pk_{\mathcal{U}_1} = (M', N')$ are defined for the committed values in $j_{pk_{\mathcal{U}_1}}$ and $\widetilde{j}_{pk_{\mathcal{U}_1}}$ respectively. A proof $\pi_{pk_{\mathcal{U}_1}}$ is given.

$$\pi_M \leftarrow Prove(ck_{\mathcal{J}}, ck_{sp}; E_{eq}, (M, \mathbf{r}_{\mathcal{U}_1}), (M', \mathbf{r}'_{\mathcal{U}_1})),$$
$$\pi_N \leftarrow Prove(ck_{\mathcal{J}}, ck_{sp}; E_{eq}, (N, \mathbf{s}_{\mathcal{U}_1}), (N', \mathbf{s}'_{\mathcal{U}_1})).$$

$\mathcal{U}_1$ also gives a membership certificate (commitment signature) $c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}} = (c_{A_{\mathcal{U}_1}}, c_{B_{\mathcal{U}_1}}, c_{D_{\mathcal{U}_1}}, c_{R_{\mathcal{U}_1}}, c_{S_{\mathcal{U}_1}}, \pi_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}} = (\pi_{A_{\mathcal{U}_1}}, \pi_{B_{\mathcal{U}_1}}, \pi_{R_{\mathcal{U}_1}}))$. The member certificate asserts that $c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}}$ contains a valid signature on the value committed in $j_{pk_{\mathcal{U}_1}}$. The corresponding proof $\pi_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}} = (\pi_{A_{\mathcal{U}_1}}, \pi_{B_{\mathcal{U}_1}}, \pi_{R_{\mathcal{U}_1}})$ [22] is given.

$$\pi_{A_{\mathcal{U}_1}} \leftarrow RdProof(ck_{\mathcal{J}}, E_A, (c_{A_{\mathcal{U}_1}}, 0), (c_{D_{\mathcal{U}_1}}, 0), (c_{M_{\mathcal{U}_1}}, 0), (c_{S_{\mathcal{U}_1}}, \boldsymbol{\varrho}'), \pi'_A),$$
$$\pi_{R_{\mathcal{U}_1}} \leftarrow RdProof(ck_{\mathcal{J}}, E_R, (c_{R_{\mathcal{U}_1}}, \boldsymbol{\rho}'), (c_{S_{\mathcal{U}_1}}, \boldsymbol{\varrho}'), \pi_P),$$
$$\pi_{B_{\mathcal{U}_1}} \leftarrow Prove(ck_{\mathcal{J}}, E_{\mathcal{DH}}, (f^c, \boldsymbol{\beta}), (h^c, \boldsymbol{\delta})),$$

where

$$E_A(A, M; S, D) : \hat{e}(t^{-1}, S)\hat{e}(A, Y)\hat{e}(M, h^{-1})\hat{e}(A, D) = \hat{e}(k, h),$$
$$E_R(R; S) : \hat{e}(g^{-1}, S)\hat{e}(R, h) = 1,$$
$$E_{\mathcal{DH}}(M, N) : \hat{e}(g^{-1}, N)\hat{e}(M, h) = 1.$$

The proofs π'_A and π_P are

$$\pi'_A \leftarrow \pi_U \odot Prove(ck, E_{A^\dagger}, (A, \boldsymbol{\alpha}), (h^c, \boldsymbol{\delta})),$$
$$\pi_P \leftarrow Prove(ck, E_{\mathcal{DH}}, (P, \boldsymbol{\rho}), (Q, \boldsymbol{\varrho})),$$

where

$$E_{A^\dagger}(A; D) : \hat{e}(A, Y)\hat{e}(A, D) = 1,$$
$$E_{\mathcal{DH}}(M, N) : \hat{e}(g^{-1}, N)\hat{e}(M, h) = 1.$$

The user $\mathcal{U}_1$ sends the following values to $\mathcal{W}$: $\{pk_{\mathcal{U}_1}, j_{pk_{\mathcal{U}_1}}, j_{n_1}, c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}}, \pi_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}}\}$.

2. $\mathcal{W}$ verifies the NIZK proof $\pi_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}}$ and the public $pk_{\mathcal{U}_1}$. If they are correct, $\mathcal{W}$ chooses random nonces $n_0, \iota_1^0, \iota_2^0 \in \mathbb{Z}_p$ for the coin and generates a $\mathcal{DH}$ pair $\mathcal{DH}_{n_0} = (g^{n_0}, h^{n_0})$. Note that ι_1^0 and ι_2^0 are random values chosen by $\mathcal{W}$. $\mathcal{W}$ generates two commitments

$$d_{n_0} = (c_{g^{n_0}}^{ck_{\mathcal{D}}}, c_{h^{n_0}}^{ck_{\mathcal{D}}}, c_{P^0}^{ck_{\mathcal{D}}}, c_{Q^0}^{ck_{\mathcal{D}}}, U = t^{\iota_1^0} \cdot g^{n_0}, \pi_{d_{n_0}})$$

and

$$j_{n_0} = (c_{g^{n_0}}^{ck_{\mathcal{J}}}, c_{h^{n_0}}^{ck_{\mathcal{J}}}, c_{P^0}^{ck_{\mathcal{J}}}, c_{Q^0}^{ck_{\mathcal{J}}}, U = t^{\iota_2^0} \cdot g^{n_0}, \pi_{j_{n_0}})$$

$\mathcal{W}$ also gives a proof

$$\pi_{g^{n_0}} \leftarrow Prove(ck_{\mathcal{D}}, ck_{\mathcal{J}}; E_{eq}, (g^{n_0}, \boldsymbol{\mu}_{\mathcal{W}}), (g^{n_0}, \boldsymbol{\mu}'_{\mathcal{W}}))$$

This asserts that the two committed values in j_{n_0} and d_{n_0} are equal. Then $\mathcal{W}$ respectively produces committed signatures

$$\begin{aligned}
c_{s_n} &= (c_{A_n}, c_{B_n}, c_{D_n}, c_{R_n}, c_{S_n}, \pi_{s_n}) \\
c_{s_m} &= (c_{A_m}, c_{B_m}, c_{D_m}, c_{R_m}, c_{S_m}, \pi_{s_m}) \\
c_{s_{n_0}} &= (c_{A_{n_0}}, c_{B_{n_0}}, c_{D_{n_0}}, c_{R_{n_0}}, c_{S_{n_0}}, \pi_{s_{n_0}}) \\
c_{s_{n_1}} &= (c_{A_{n_1}}, c_{B_{n_1}}, c_{D_{n_1}}, c_{R_{n_1}}, c_{S_{n_1}}, \pi_{s_{n_1}}) \\
c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{W}}} &= (c'_{A_{\mathcal{U}_1}}, c'_{B_{\mathcal{U}_1}}, c'_{D_{\mathcal{U}_1}}, c'_{R_{\mathcal{U}_1}}, c'_{S_{\mathcal{U}_1}}, \pi_{s_{\mathcal{U}_1}}^{sk_{\mathcal{W}}})
\end{aligned}$$

on 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 $j_{pk_{\mathcal{U}_1}}$. The corresponding proofs are $\pi_{s_n}, \pi_{s_m}, \pi_{s_{n_0}}, \pi_{s_{n_1}}$ and $\pi_{s_{\mathcal{U}_1}}^{sk_{\mathcal{W}}}$. They are similar to the proof $\pi_{c_{\mathcal{U}_1}}^{sk}$.

$\mathcal{W}$ sends the following values to $\mathcal{U}_1$: $\{j_{n_0}, d_{n_0}, \pi_{n_0}, c_{s_n}, \pi_{s_n}, c_{s_m}, \pi_{s_m}, c_{s_{n_0}}, \pi_{s_{n_0}}, c_{s_{n_1}}, \pi_{s_{n_1}}, c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{W}}}, \pi_{s_{\mathcal{U}_1}}^{sk_{\mathcal{W}}}\}$.

Finally, $\mathcal{U}_1$ forms the coin $co_1 = (p_n, \widetilde{p}_n, \pi_n, p_m, \widetilde{p}_m, \pi_m, j_{n_0}, d_{n_0}, \pi_{n_0}, j_{n_1}, d_{n_1}, \pi_{n_1}, j_{pk_{\mathcal{U}_1}}, \widetilde{j}_{pk_{\mathcal{U}_1}}, \pi_{pk_{\mathcal{U}_1}}, c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}}, \pi_{c_{\mathcal{U}_1}}^{sk_{\mathcal{J}}}, c_{s_n}, \pi_{s_n}, c_{s_m}, \pi_{s_m}, c_{s_{n_0}}, \pi_{s_{n_0}}, c_{s_{n_1}}, \pi_{s_{n_1}}, c_{s_{\mathcal{U}_1}}^{sk_{\mathcal{W}}}, \pi_{s_{\mathcal{U}_1}}^{sk_{\mathcal{W}}})$.

5.4 The Spending (Transferring) Protocol

This protocol makes a payer $\mathcal{U}_1$ to transfer a coin to the payee $\mathcal{U}_2$ as described in Figure 3. In order to obtain the anonymity of the user, $\mathcal{U}_1$ needs to randomize the coin co_i by $RdCom$ and $RdProve$. $\mathcal{W}'s$ verification key is public while $\mathcal{U}_1's$ verification key must be hidden. Thus, after $\mathcal{U}_1$ generates commitment signatures $c_{s_{n_1}}, c_{s_{n_2}}$ and $c_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}}$ and corresponding proof, $\mathcal{U}_1$ updates the proof using the algorithm Adc_κ to protect $\mathcal{U}_1's$ verification key. This hides the identity of $\mathcal{U}_1$.

1. $\mathcal{U}_2$ picks at random nonces $n_2, \iota_1^2, \iota_2^2, \iota_3^2, \iota_4^2 \in \mathbb{Z}_p, \boldsymbol{\mu}_{\mathcal{U}_2}, \boldsymbol{\mu}'_{\mathcal{U}_2}, \boldsymbol{\nu}_{\mathcal{U}_2}, \boldsymbol{\nu}'_{\mathcal{U}_2}, \mathbf{r}_{\mathcal{U}_2}, \mathbf{r}'_{\mathcal{U}_2}, \mathbf{s}_{\mathcal{U}_2}, \mathbf{s}'_{\mathcal{U}_2} \in \mathbb{Z}_p^2, t_{\mathcal{U}_2} \in G_1$ and generates a $\mathcal{DH}$ pair $\mathcal{DH}_{n_2} = (g^{n_2}, h^{n_2})$. Note that the symbols $\iota_1^2, \iota_2^2, \iota_3^2$ and ι_4^2 are defined that $\mathcal{U}_2$ randomly chooses nonces $\iota_1^2, \iota_2^2, \iota_3^2$ and ι_4^2. $\mathcal{U}_2$ makes commitments d_{n_2} and j_{n_2}. The proof π_{n_2} asserts that two committed values in d_{n_2} and j_{n_2} are equal. Moreover, $\mathcal{U}_2$ makes commitments $j_{pk_{\mathcal{U}_2}}$ and $\widetilde{j}_{pk_{\mathcal{U}_2}}$. Meanwhile, the proof $\pi_{pk_{\mathcal{U}_2}}$ proves that two committed values are equal. The proof $\pi_{pk_{\mathcal{U}_2}}$ is similar to the proof $\pi_{pk_{\mathcal{U}_1}}$. $\mathcal{U}_2$ also gives a membership certificate $c_{s_{\mathcal{U}_2}}^{sk_{\mathcal{J}}} = (c_{A_{\mathcal{U}_2}}, c_{B_{\mathcal{U}_2}}, c_{D_{\mathcal{U}_2}}, c_{R_{\mathcal{U}_2}}, c_{S_{\mathcal{U}_2}}, \pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{J}}} = (\pi_{A_{\mathcal{U}_2}}, \pi_{B_{\mathcal{U}_2}}, \pi_{R_{\mathcal{U}_2}}))$. The proof $\pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{J}}} = (\pi_{A_{\mathcal{U}_2}}, \pi_{B_{\mathcal{U}_2}}, \pi_{R_{\mathcal{U}_2}})$ is similar to $\pi_{s_{\mathcal{U}_1}}^{sk_{\mathcal{J}}}$ in Section 5.3. It is given to assert that the value in $c_{s_{\mathcal{U}_2}}^{sk_{\mathcal{J}}}$ is a valid signature on the value in $j_{pk_{\mathcal{U}_2}}$.

   The user $\mathcal{U}_2$ sends the following values to $\mathcal{U}_1$: $\{j_{pk_{\mathcal{U}_2}}, j_{n_2}, c_{s_{\mathcal{U}_2}}, \pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{J}}}\}$.
2. $\mathcal{U}_1$ firstly checks the proof $\pi_{c_{\mathcal{U}_2}}^{sk_{\mathcal{J}}}$. If the verification is correct, $\mathcal{U}_1$ randomizes co_1 as $co_1^1 = (p_n^1, \widetilde{p}_n^1, \pi_n^1, p_m^1, \widetilde{p}_m^1, \pi_m^1, j_{n_0}^1, d_{n_0}^1, \pi_{n_0}^1, j_{n_1}^1, d_{n_1}^1, \pi_{n_1}^1, j_{pk_{\mathcal{U}_1}}^1, \widetilde{j}_{pk_{\mathcal{U}_1}}^1, \pi_{pk_{\mathcal{U}_1}}^1, j_{c_{\mathcal{U}_1}}^1, \pi_{c_{\mathcal{U}_1}}^1, c_{s_n}^1, \pi_{s_n}^1, c_{s_m}^1, \pi_{s_m}^1, c_{s_{n_0}}^1,$

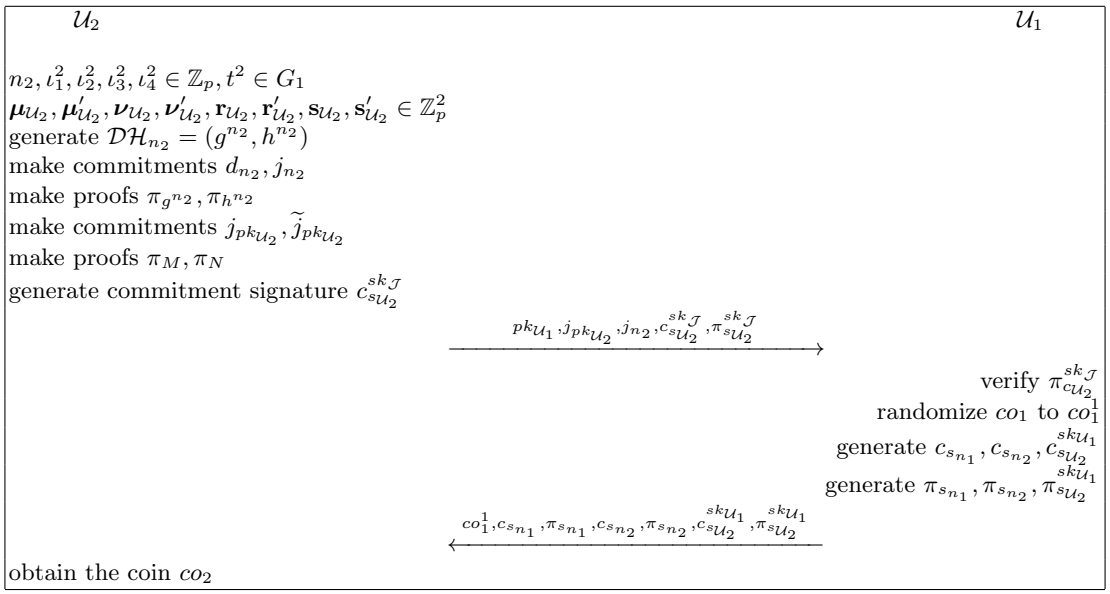

Fig. 3. $\mathcal{U}_1$ spends a conditional e-cash to $\mathcal{U}_2$.

$\pi^1_{s_{n_0}}, c^1_{s_{n_1}}, \pi^1_{s_{n_1}}, c^1_{s_{u_1}}, \pi^1_{s_{u_1}}$) using *RdCom* and *RdProve* [16, 23]. Then $\mathcal{U}_1$ respectively computes committed signatures $c_{s_{n_1}}, c_{s_{n_2}}$ and $c_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}}$ on the values which are committed in j_{n_1}, j_{n_2} and $j_{pk_{\mathcal{U}_2}}$ using *SigCom*. $\mathcal{U}_1$ also outputs the proofs $\pi'_{s_{n_1}}, \pi'_{s_{n_2}}$ and $\{\pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}}\}'$. To hide the verification key of $\mathcal{U}_1$, $\mathcal{U}_1$ converts $\pi'_{s_{n_1}}, \pi'_{s_{n_2}}$ and $\{\pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}}\}'$ to $\pi_{s_{n_1}}, \pi_{s_{n_2}}$ and $\pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}}$ by running AdC_κ. The transformation of $\pi'_{s_{n_1}}$ and $\pi'_{s_{n_2}}$ are similar to $\{\pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}}\}'$. In the following, $\mathcal{U}_1$ converts $\{\pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}}\}' = (\pi^{s_2}_{A'}, \pi^{s_2}_B, \pi^{s_2}_R)$ to $\pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}} = (\pi^{s_2}_A, \pi^{s_2}_B, \pi^{s_2}_R)$, where $pk_{\mathcal{U}_2} = (g^{sk_{\mathcal{U}_2}}, h^{sk_{\mathcal{U}_2}}) = (M, N)$. Note that since the last two proofs in $\{\pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}}\}'$ is the same as those in $\pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}}$, $\mathcal{U}_1$ only changes $\pi^{s_2}_{A'}$ to $\pi^{s_2}_A$.

$$\pi^{s_2}_A = RdProof(ck_{\mathcal{U}_1}, E_{\widehat{A}}, (c^{s_2}_A, 0), (c^{s_2}_M, 0), (c^{s_2}_S, 0), ck'_{\mathcal{U}_1} = (Com(ck_{\mathcal{U}_1}, Y^{s_2}, 0), \boldsymbol{\psi}), (c^{s_2}_D, 0), \pi^{s_2}_{A'}),$$

where

$$E_{\widehat{A}}(A, M; S, Y, D) : \hat{e}(t^{-1}, S)\hat{e}(M, h)\hat{e}(A, Y)\hat{e}(A, D) = \hat{e}(k, h).$$

$\mathcal{U}_1$ sends the following values to $\mathcal{U}_2$: $\{co_1^1, c_{s_{n_1}}, \pi_{s_{n_1}}, c_{s_{n_2}}, \pi_{s_{n_2}}, c_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}}, \pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}}\}$.

Finally, the user $\mathcal{U}_2$ generates the coin $co_2 = (co_1^1, j_{n_2}, d_{n_2}, \pi_{n_2}, j_{pk_{\mathcal{U}_2}}, \widetilde{j}_{pk_{\mathcal{U}_2}}, \pi_{pk_{\mathcal{U}_2}}, c_{s_{\mathcal{U}_2}}^{sk_{\mathcal{J}}}, \pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{J}}}, c_{s_{n_1}}, \pi_{s_{n_1}}, c_{s_{n_2}}, \pi_{s_{n_2}}, c_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}}, \pi_{s_{\mathcal{U}_2}}^{sk_{\mathcal{U}_1}})$.

5.5 The Deposit Protocol

After the outcome is published, we assume that the outcome is favorable to the payee, thus the payee proves that he owns the commitment of the condition and the correct proof of the commitment to the publisher by an authenticated and secure channel. If the proof is correct, the publisher sends the extraction key to the winner and the bank. Therefore, the payee obtains the extraction key ek_{pe} and extracts the secret value $\mathcal{DH}_m$. To achieve the anonymity of the user, the deposit protocol is divided into two parts, i.e., Exchanging and Cashing.

Exchanging. By Exchanging, $\mathcal{U}_{i+1}$ exchanges the conditional e-cash for a new value *mo*. The description of Exchanging is as follow.

1. $\mathcal{U}_{i+1}$ randomizes co_i as $co_i^1 = (p_n^i, \widetilde{p}_n^i, \pi_n^i, p_m^i, \widetilde{p}_m^i, \mathcal{DH}_m, \pi_m^i, p_m^i, co_1^i, co_2^{i-1}, \cdots, co_{i-1}^2, j_{n_i}^1, d_{n_i}^1, \pi_{n_i}^1, j_{pk_{\mathcal{U}_i}}^1, \widetilde{j}_{pk_{\mathcal{U}_i}}^1, \pi_{pk_{\mathcal{U}_i}}^1, \{c_{s\mathcal{U}_i}^{sk_{\mathcal{J}}}\}^1, \{\pi_{c\mathcal{U}_i}^{sk_{\mathcal{J}}}\}^1, c_{s_{n_{i-1}}}^1, \pi_{s_{n_{i-1}}}^1, c_{s_{n_i}}^1, \pi_{s_{n_i}}^1, \{c_{s\mathcal{U}_i}^{sk_{\mathcal{U}_{i-1}}}\}^1, \{\pi_{s\mathcal{U}_i}^{sk_{\mathcal{U}_{i-1}}}\}^1)$, and sends co_i^1 to $\mathcal{D}$. $\mathcal{U}_{i+1}$ runs the spending protocol with $\mathcal{D}$.
2. In order to detect the double-spending, $\mathcal{D}$ firstly verifies the correctness of the secret value $\mathcal{DH}_m$ and commitment p_m^i, and checks whether $\mathcal{DH}_m$ is the value committed in p_m^i. If not, this protocol aborts, otherwise $\mathcal{D}$ opens the commitments $d_{n_0}^i, d_{n_1}^{i-1}, \cdots, d_{n_{i-1}}^2, d_{n_i}^1$ contained in the coin using the extraction key of $\mathcal{D}$. $\mathcal{D}$ obtains the serial number $s = \mathcal{DH}_m||\mathcal{DH}_{n_0}||\mathcal{DH}_{n_1}||\mathcal{DH}_{n_2}||\cdots||\mathcal{DH}_{n_i}$, and checks whether the coin is found in the database DB. If not, $\mathcal{D}$ sends the value $mo = (p_m^i, j_{pk_{\mathcal{U}_i}}^1, c_{s_m}, \pi_{c_m})$ to $\mathcal{U}_i$. c_{s_m} is the commitment signature to the commitments p_m^i and $j_{pk_{\mathcal{U}_i}}^1$, and π_{c_m} proves that the committed value in c_{s_m} is a valid signature to the values committed in p_m^i and $j_{pk_{\mathcal{U}_i}}^1$. mo is a correct deposit of the user. $\mathcal{D}$ also saves mo to the database DB'. Otherwise $\mathcal{D}$ runs the Identify Procedures.

Cashing. By Cashing, $\mathcal{U}_{i+1}$ cashes from $\mathcal{D}$. The description of Cashing is as follow.

1. To protect the identity of $\mathcal{U}_{i+1}$, $\mathcal{U}_{i+1}$ converts the value $mo = (p_m^i, j_{pk_{\mathcal{U}_i}}^1, c_{s_m}, \pi_{c_m})$ to $mo' = (\{p_m^i\}', \{j_{pk_{\mathcal{U}_i}}^1\}', c'_{s_m}, \pi'_{c_m}, c'_{s_m}, \pi_{s_m})$ by running $RdCom$, $RdProve$ and AdC_κ. The committed value in c'_{s_m} is a valid signature under the value committed in $\{j_{pk_{\mathcal{U}_i}}^1\}'$, and π_{s_m} is the corresponding proof. Then $\mathcal{U}_{i+1}$ directly contacts $\mathcal{D}$ through an authenticated and secure channel and supplies his account number and mo'.
2. $\mathcal{D}$ detects that mo' includes the correct signature, $\mathcal{U}_{i+1}$ exchanges this piece of currency for credit to his account. If the user double-supplies mo, $\mathcal{D}$ recovers the identity of double-spender by the Identify Procedures.

If the outcome is favorable to the payer $\mathcal{U}_1$, $\mathcal{U}_1$ cashes back the coin from the bank as the above procedure except that $\mathcal{U}_1$ spends the coin co_1^1 and $\mathcal{D}$ obtains the serial number $\mathcal{DH}_n||\mathcal{DH}_{n_0}||\mathcal{DH}_{n_1}||\cdots||\mathcal{DH}_{n_i}$.

5.6 The Identify Procedures

The Identify Procedures makes sure that when a double-spending is found, $\mathcal{D}$ sends co_i and co_{i+1}^1 to $\mathcal{J}$ to recover the identity of double-spender. The description of the Identify Procedures is as follow.

$\mathcal{D}$ finds another serial number which begins with n in DB, i.e., $s' = \mathcal{DH}_n||\mathcal{DH}'_{n_0}||\mathcal{DH}'_{n_1}||\mathcal{DH}'_{n_2}||\cdots||\mathcal{DH}'_{n_i}$, this asserts that U_{i+1} is a double-spender. $\mathcal{D}$ compares the two serial numbers s and s' and stops at the last t such that $\mathcal{DH}_{n_t} = \mathcal{DH}'_{n_t}$. Finally, $\mathcal{D}$ sends two coins to $\mathcal{J}$. In order to identify the defrauder, $\mathcal{J}$ extracts the identity committed in $j_{pk_{\mathcal{U}_{i+1}}}$ using extraction key $ek_{\mathcal{J}}$.

5.7 Efficiency

Schemes	Shi [25]	Blanton [17]	Ours
Efficiency	$\mathcal{O}(m_1 m_2 k)$	$\mathcal{O}(\lambda log m_3)$	$\mathcal{O}(\lambda')$
Security model	Random oracle model		Standard model

Table 1. Efficiency Comparison between related work and our scheme

We analyze the efficiency and security model by comparing the computation and communication in Table 1. In Shi *et al.*'s scheme [25], the computation and communication are $\mathcal{O}(m^2 k)$ to achieve the probability $1/m$ in cut-and-choose techniques with a security parameter m. When $\mathcal{U}_1$ withdraws the e-cash from $\mathcal{B}$, $\mathcal{B}$ needs to do $m_1 - 1$ verification for achieving the probability $1/m_1$. When $\mathcal{U}_1$ transfers the e-cash to $\mathcal{U}_2$, $\mathcal{U}_2$ needs $m_2 - 1$ verification. Therefore, Shi *et al.*'s scheme [25] requires $\mathcal{O}(m_1 m_2 k)$

computation and communication, where m_1 and m_2 are the security parameters of the cut and choose techniques, and k is a security parameter for RSA-based systems.

In Blanton's scheme, he constructed a conditional e-cash using verifiable encryption. The verifiable encryption [32] needs $\mathcal{O}(logk_1)$ computation and communication, where k_1 is the security parameter of the verifiable encryption. When an user spends the conditional e-cash to another, he needs to prove the correctness of the construction using the verifiable encryption. Blanton's scheme thus needs $\mathcal{O}(\lambda logm_3)$, where λ is a security parameter for groups with bilinear maps and m_3 is the security parameter of the verifiable encryption.

On the contrary, the commitments and corresponding proofs are used to represent a conditional e-cash in this scheme. The proofs are given using Groth-Sahai proofs. The Groth-Sahai proofs are not-interactive, thus the prover only needs to supply the commitments and corresponding proofs, and then the verifier verifies the correctness of the proofs. Therefore, the user only needs to transfer some commitments and proofs in the conditional protocol, the withdrawal protocol, spending protocol and the deposit protocol. Moreover, the communication number only needs one round. Therefore, computation and communication in this scheme are constant, i.e., $O(\lambda')$, where λ' is the security parameter of this scheme.

As for the security model, Shi *et al.* and Blanton's schemes are proven in the random oracle model. However, our new scheme is proven in the standard model.

6 Security Analysis

This section gives the rigid security proof. The scheme fulfills four security requirements, i.e., Anonymity, Unforgeability, Identification of Double-spender and Exculpability. An adversary is defined by $\mathcal{A}$. A challenger is defined by $\mathcal{C}$. A series of games is given in $\mathcal{A}$ and $\mathcal{C}$ to prove the security properties. In all games, we suppose that $\mathcal{U}_1$ firstly withdraws an e-cash co_1 from $\mathcal{B}$. Then $\mathcal{U}_1$ spends the e-cash co_1 to $\mathcal{U}_2$, and $\mathcal{U}_2$ obtains an e-cash co_2. And last, $\mathcal{U}_i$ deposits an e-cash co_i to $\mathcal{B}$.

Theorem 1. *The transferable conditional e-cash scheme provides anonymity under the following assumptions: SXDH assumption and soundness and witness indistinguishability of Groth-Sahai proofs.*

Proof. Our scheme achieves optimal anonymity: StO-FA, StR-FA and OtR-FA.

- *Spend-then-Observe* Full Anonymity (Sto-FA). To achieve Sto-FA property, the adversary $\mathcal{A}$ can not observe the coin in the *challenge phase* [18]. In this scheme, the coins transferred between the users are represented using SXDH-based commitments. If $\mathcal{A}$ wants to determine which public key is chosen by the challenger $\mathcal{C}$ in *Challenge phase*, he needs to know the public key committed in $j_{pk_{\mathcal{U}_i}}$. Thus, $\mathcal{A}$ can break the soundness and witness indistinguishability of Groth-Sahai proofs [20].
- *Spend-then-Receive* Full Anonymity (StR-FA). For StR-FA property, $\mathcal{A}$ is allowed to obtain any challenge coins co_0 and co_1 for the public keys $pk_{\mathcal{U}_0}$ and $pk_{\mathcal{U}_1}$ in *Probing phase* while he can not act as $\mathcal{D}$ in *Challenge phase*. Therefore, $\mathcal{A}$ only observes the deposit in the *Probing pahse*. Because $\mathcal{A}$ is given the challenge coin owned before, he must not know $\mathcal{D}'s$ key to detect double-spending.
  Groth-Sahai proofs are witness indistinguishable. If the commitment key is set up as perfectly hiding, the commitments and the random values are indistinguishable. Meanwhile, the corresponding proof can also be simulated by the trapdoor information. In our scheme, the coins are represented by randomizable extractable commitments. Thus, the coins can not reveal anything about the serial number, the public keys and certificates. In order to simulate the deposit oracle, identification oracle and conditional oracle, three commitments: d_i, $\widetilde{j}_i$ and $\widetilde{p}_i$ are introduced for auxiliary proof. We simulate this scheme by the following game.
  Game 0: This is the real scheme.
  Game 1: As Game 0, except that the committed values from the commitments d_i, $\widetilde{j}_i$ and $\widetilde{p}_i$ are extracted to detect, trace double-spending and obtain the condition respectively. Under soundness and witness indistinguishability of Groth-Sahai proofs, this change is negligible.

Game 2: As Game 1, except that the judge and publisher's keys for the commitments j and p are set up as perfectly hiding. Under SXDH assumption, this change is negligible.
Game 3: As Game 2, except that all proofs under the different keys are simulated. This can be done using the trapdoor information for the commitments.
Game 4: As Game 3, except that all commitments d_{n_i}, j_{n_i} and p_{n_i} are replaced with random values. Under SXDH assumption, the commitments and the random values are indistinguishable.
For StO-FA, $\mathcal{A}$ only observes the spending in the *Probing phase*. The SXDH-based commitments and Groth-Sahai proofs are witness indistinguishable. When a spending is challenged, $\mathcal{A}$ observes the spending while he can not distinguish the coin with another coin which is perfectly random. Therefore, if $\mathcal{A}$ win the games, he can break SXDH assumption, the soundness and the witness indistinguishable of Groth-Sahai proofs.

– *Observe-then-Receive* Full Anonymity (OtR-FA). The proof is similar to that in StO-FA, except that $\mathcal{A}$ can not receive the challenge coin while he can control the deposit oracle. $\mathcal{A}$ has never seen the challenge coin. Thus, a spending is challenged in $\mathcal{O}_{Spend()}$ query, we can replace all commitments by random values or simply leave the commitments. Under SXDH assumption, the commitments and random values are indistinguishable.

It is known $\mathcal{A}$ can win if he can solve the SXDH assumption, and break the soundness and the witness indistinguishable of Groth-Sahai proofs.

Theorem 2. *The transferable conditional e-cash scheme provides unforgeability under the following assumptions: SXDH assumption, unforgeability of the commuting signature scheme and soundness and witness indistinguishability of Groth-Sahai proofs.*

Proof. Let $\mathcal{A}$ be an adversary, and $\mathcal{C}$ be a challenger. The probability that $\mathcal{A}$ breaks the unforgeability is $Pr[Exp^{unfor}_{TCE,\mathcal{A}}(\lambda) = 1]$. We prove that the probability is negligible. $\mathcal{C}$ acting the bank generates both the public key and the secret key of the bank. co_{ua} is the value of the e-cash received by $\mathcal{A}$ in the whole protocol, and is initialized with zero. We gradually modify $\mathcal{A}$'s views in each of the following games.

Game 0: This is the real scheme. In this game, $\mathcal{C}$ runs $ParamSetup(1^\lambda)$ and obtains the public parameters $params$ and the key pair $(pk_{\mathcal{B}}, sk_{\mathcal{B}})$. Then $\mathcal{C}$ sends $pk_{\mathcal{B}}$ to $\mathcal{A}$ and keeps $sk_{\mathcal{B}}$ to itself.

Game 1: In $\mathcal{O}_{With()}$ query, $\mathcal{A}$ acting $\mathcal{U}_1$ randomly chooses n_1, and gives the corresponding commitments and proofs. $\mathcal{A}$ sends the public key and other commitments to $\mathcal{C}$ acting $\mathcal{B}$. Then $\mathcal{A}$ obtains two conditional commitments p_n and p_m. And last, $\mathcal{C}$ verifies the public key and proofs. If these are correct, $\mathcal{C}$ chooses n_0, and gives the corresponding commitments and proofs. Clearly, $\mathcal{A}'$views are the conditional commitments, the random nonce sent by the bank, and the committed signatures. Under SXDH assumption, the commitments and the random value are indistinguishable. Therefore, the Game is identical to her view in Game 0. v_{ui} is defined for the value of the e-cash withdrew by $\mathcal{A}$.

Game 2: $\mathcal{O}_{Spend()}$ is similar to $\mathcal{O}_{With()}$ except that $\mathcal{A}$ acting $\mathcal{U}_1$ randomizes the e-cash co_1 to co_1^1, and updates the committed signature to protect the identity of $\mathcal{A}$. v_{uo} is defined for the value of the e-cash spent by $\mathcal{A}$ in $\mathcal{O}_{Spend()}$ query. v_{ui}^1 is defined for the value of the e-cash obtained by $\mathcal{A}$ in $\mathcal{O}_{Spend()}$ query. In $\mathcal{O}_{Depo()}$, $\mathcal{A}$ acting $\mathcal{B}$ firstly spends an e-cash to $\mathcal{C}$ acting $\mathcal{B}$, which is similar to $\mathcal{O}_{Spend()}$. v_{de} is defined for the value of the e-cash deposited. Then $\mathcal{C}$ obtains the e-cash v_{de}, and checks whether the database contains the e-cash. If not, the Game aborts. Otherwise, $v_{ui} + v_{ui}^1 - v_{uo} > v_{de}$. This asserts that $\mathcal{A}$ has spent a coin which is not withdrew from the bank.

If the above event occurs, the unforgeability of commuting signature is broken. $\mathcal{A}$ can break SXDH assumption, unforgeability of the commuting signature scheme and soundness and witness indistinguishability of Groth-Sahai proofs. It is known that the probability is negligible.

Theorem 3. *The transferable conditional e-cash scheme provides identification of double-spender under the following assumptions: SXDH assumption, unforgeability of the commuting signature scheme and soundness and witness indistinguishability of Groth-Sahai proofs.*

Proof. We assume that $\mathcal{A}$ is an adversary who double-spends coins while the identification algorithm does not output $\mathcal{A}'s$ public key. In the following, a series of games is given to prove that the probability of double-spending is negligible.

Game 0: This is the real scheme. In this game, $\mathcal{C}$ supplies the judge's public key. $\mathcal{C}$ runs the $ParamSetup(1^\lambda)$ and obtains the public parameters $params$ and the key pair $(pk_\mathcal{J}, sk_\mathcal{J})$. Then $\mathcal{C}$ sends $pk_\mathcal{J}$ to $\mathcal{A}$ and keeps $sk_\mathcal{J}$ to itself.

Game 1: In $\mathcal{O}_{With()}$, $\mathcal{A}$ acting $\mathcal{U}_1$ firstly chooses n_1, and gives the corresponding commitments and proofs. $\mathcal{A}$ sends the public key and other commitments to $\mathcal{C}$ acting $\mathcal{B}$. Then $\mathcal{A}$ obtains two conditional commitments p_n and p_m. And last, $\mathcal{C}$ verifies the public key and proofs. If these are correct, $\mathcal{C}$ chooses n_0, and gives the corresponding commitments and proofs. Clearly, $\mathcal{A}'$views are the conditional commitments, the random nonce sent by the bank, and the committed signatures. As in the proof of unforgeability, this Game is identical to her view in Game 0.

Game 2: A failure event E_1 is introduced. Let $\mathcal{C}$ aborts if the extracted key - which $\mathcal{C}$ obtains by running the knowledge extractor of the commitments - is such that $pk_{\mathcal{U}_1} \neq (g,h)^{sk_{\mathcal{U}_1}}$. Therefore, Game 2 and Game 1 are identical until E_1 occurs. If this occurs, $\mathcal{A}$ is able to break the soundness of Groth-Sahai proofs. It is known that this is negligible. Game 2 is thus indistinguishable from Game 1.

Game 3: $\mathcal{O}_{Spend()}$ and $\mathcal{O}_{Depo()}$ queries are similar to that in Game 2 of unforgeability. In this Game, $\mathcal{C}$ acting $\mathcal{B}$ aborts if $\mathcal{A}$ acting $\mathcal{U}_i$ deposits an e-cash co which is not withdrew from $\mathcal{C}$. Under the soundness of Groth-Sahai proofs, each valid coin must contain a valid certificate for the public key corresponding to each transfer. Thus, $\mathcal{A}$ has forged the certificate of co. Meanwhile, $\mathcal{A}$ acting $\mathcal{U}_i$ has also forged a commitment signature for the certificate. Therefore, by the unforgeability of commuting signature and the soundness of Groth-Sahai proofs, Game 3 and Game 2 are indistinguishable. These imply that $\mathcal{A}$ cannot produce a new coin which is not signed by $\mathcal{C}$.

If the above event occurs, the unforgeability of commuting signature is broken. $\mathcal{A}$ can break SXDH assumption, unforgeability of the commuting signature scheme and soundness and witness indistinguishability of Groth-Sahai proofs. It is known that the probability is negligible.

Theorem 4. *The transferable conditional e-cash scheme provides exculpability under the following assumptions: SXDH assumption, unforgeability of the commuting signature scheme and soundness and witness indistinguishability of Groth-Sahai proofs.*

Proof. The exculpability is that $\mathcal{A}$ can accuse an honest user of happening a double-spending. The proof is similar to that of the unforgeability, except that we request the commuting signature is issued by an honest user rather than the bank. In the following, a series of games is given to prove that the probability of exculpability is negligible.

Game 0: This is the real scheme. In this game, $\mathcal{C}$ supplies a security parameter to $\mathcal{A}$. $\mathcal{A}$ runs $ParamSetup(1^\lambda)$ and obtains the public parameters $params$ and the key pair $(pk_\mathcal{B}, sk_\mathcal{B})$. Then $\mathcal{A}$ sends $pk_\mathcal{B}$ to $\mathcal{C}$ and keeps $sk_\mathcal{B}$ to itself.

Game 1: In $\mathcal{O}_{With()}$ query, $\mathcal{C}$ acting $\mathcal{U}_1$ chooses n_1, and gives the corresponding commitments and proofs. $\mathcal{C}$ also sends the public key and other commitments to $\mathcal{A}$ acting $\mathcal{B}$. Then $\mathcal{C}$ obtains two conditional commitments p_n and p_m. And last, $\mathcal{A}$ verifies the public key and proofs. If these are correct, $\mathcal{A}$ chooses n_0, and gives the corresponding commitments and proofs. Clearly, $\mathcal{A}$ generates both the public key and the secret key, so he know the e-cash withdrew by $\mathcal{C}$. From this game, $\mathcal{A}$ can not obtain more knowledge than Game 0.

Game 2: In $\mathcal{O}_{Spend()}$ query, $\mathcal{A}$ acting $\mathcal{U}_2$ firstly chooses n_2, and sends the corresponding commitments and proofs to $\mathcal{C}$. $\mathcal{C}$ acting $\mathcal{U}_1$ checks the proofs. If the proofs are correct, $\mathcal{C}$ generates the commuting signature $c_{s_{u_2}}$ for the the commitment of $\mathcal{U}_2$'s public key using her private key. Then $\mathcal{C}$ sends the commuting signature and corresponding proofs to $\mathcal{A}$. As for $\mathcal{O}_{Depo()}$ query, $\mathcal{C}$ acting $\mathcal{U}_i$ spends the e-cash to $\mathcal{A}$ acting $\mathcal{B}$. This spending is similar to that in $\mathcal{O}_{Spend()}$ query. After $\mathcal{A}$ checks the e-cash, $\mathcal{A}$ exchanges the e-cash with a value mo. And last, $\mathcal{C}$ cashes from $\mathcal{A}$.

In this Game, we introduce a event E_1 and let $\mathcal{A}$ gives two coins co and co' with serial numbers $\mathcal{DH}_m = \mathcal{DH}'_m$, $\mathcal{DH}_{n_0} = \mathcal{DH}'_{n_0}, \cdots \mathcal{DH}_{n_i} = \mathcal{DH}'_{n_i}$ and $\mathcal{DH}_{n_{i+1}} \neq \mathcal{DH}'_{n_{i+1}}$. $c_{s_{u_i}}$ contains the user's public key pk_{u_i}. By the soundness of the Groth-Sahai proofs, they contain two commuting signatures on pk_{u_i} and pk'_{u_i}. These imply that $\mathcal{A}$ generates a commuting signature $c'_{s_{u_i}}$ for the private key of $\mathcal{C}$. It is known that the commuting signature is unforgeability. Thus, Game 2 and Game 1 is indistinguishable.

If the above event occurs, the unforgeability of commuting signature is broken. $\mathcal{A}$ can break SXDH assumption, unforgeability of the commuting signature scheme and soundness and witness indistinguishability of Groth-Sahai proofs. It is known that the probability is negligible.

7 Conclusion

In this paper, we presented the first optimally anonymous and transferable conditional e-cash in the standard model. To obtain the conditional e-cash, a publisher is introduced and firstly formalized. One important in this scheme is that the spending and deposit protocol are anonymous. Meanwhile, we used a verifiable commitment signature and updated verification key by commuting signature to obtain the optimal anonymity. To solve the linkability issue of the last payee, we modified commitments and corresponding proof using the randomization of Groth-Sahai proofs. Compared with the existing conditional e-cash schemes, our scheme has the constant size for the computation and communication. At last, we prove the security properties in the standard model.

Acknowledgments

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. Zhang J. X., Li Z. J., Guo H. Anonymous Transferable Conditional E-cash *SECURECOMM 2012*, Padua, Italy, 3-5 September.
2. Pailles, J. C. New protocols for electronic money. *ASIACRYPT 1992, LNCS 718*, Queensland, Australia, 13-16 December, pp. 263-274.
3. Eng, T. and Okamoto, T. Single-Term Divisible Electronic Coins. *EUROCRYPT 1994, LNCS 950*, Perugia, Italy, 9-12 May, pp. 306-319.
4. Okamoto, T. An efficient divisible electronic cash scheme. *CRYPTO 1995, LNCS 963*, California, USA, 27-31 August, pp. 438-451.
5. Chan, A. H., Frankel, Y. and Tsiounis, Y. Easy come-easy go divisible cash. *EUROCRYPT 1998, LNCS 1403*, Espoo, Finland, 31 May-4 June, pp. 561- 575.
6. Nakanishi, T. and Sugiyama, Y. Unlinkable divisible electronic cash. *ISW 2000*, NSW, Australia, 20-21 December, pp. 121-134.
7. Camenisch, J., Hohenberger, S., and Lysyanskaya, A. Compact e-cash. *EUROCRYPT 2005, LNCS 3494*, Aarhus, Denmark, 22-26 May, pp. 302-321.
8. Canard, S., and Gouget A. Divisible e-cash systems can be truly anonymous. *EUROCRYPT 2007, LNCS 4515*, Barcelona, Spain, 20-24 May, pp. 482-497.
9. Au, M. A., Susilo, W., and Mu, Y. Practical anonymous divisible e-cash from bounded accumulators. *FC 2008 LNCS 5143*, Cozumel, Mexico, 28-31 January, pp. 287-301.
10. Canard, S., and Gouget A. Multiple denominations in E-cash with compact transaction data. *FC 2010 LNCS 6052*, Tenerife, Canary Islands, 25-28 January, pp. 82-97.
11. Izabachene, M. and Libert, B. Divisible e-cash in the standard model. *Pairing 2012, LNCS 7708*, Cologne, Germany, 16-18 May 2012, pp. 314-332.
12. Bellare, M., Boldyreva, A. and Palacio, A. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. *EUROCRYPT 2004, LNCS 3027*, Interlaken, Switzerland, 2-6 May, pp. 171-188.
13. Canetti, R., Goldreich, O. and Halevi, S. The random oracle methodology, revisited. *ACM STOC*, 51, 557-594.

14. Chaum, D. Blind signatures for untraceable payments. *CRYPTO 1982*, Santa Barbara, California, USA, 23-25 August, pp. 199-203.
15. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., and Ohkubo, M. Structure-preserving signatures and commitments to group elements. *CRYPTO 2010, LNCS 6223*, California, USA, 15-19 August, pp. 209- 236.
16. Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., and Shacham, H. Randomizable proofs and delegatable anomymous credentials. *CRYPTO 2009, LNCS 5677*, Dakar, Senegal, 5-7 July, pp. 108-125.
17. Blanton, M. Improved conditional e-payments. *ACNS 2008, LNCS 5037*, NY, USA, 3-6 June, pp. 188-206.
18. Blazy, O., Canard, S., Fuchsbauer, G., Gouget, A., Sibert, H., and Traore, J. Achieving optimal anonymity in transferalbe e-cash with a judge. *AFRICACRYPT 2011, LNCS 6737*, 5-7 July, Dakar, Senegal, pp. 206-223.
19. Camenisch, J., Lysyanskaya, A., and Meyerovich, M. Endorsed e-cash. In IEEE Security and Privacy 2007. pp. 101-115.
20. Canard, S. and Gouget A. Anonymity in transferable e-cash. *ACNS 2008, LNCS 5037*, NY, USA, 3-6 June, pp. 207-223.
21. Chaum, D. and Pedersen, T. Transferred cash grows in size. *EUROCRYPT 1992, LNCS 658*, Balatonfred, Hungary, 24-28 May, pp. 390-407.
22. Fuchsbauer, G. Commuting signatures and verifiable encryption. *CRYPTO 2011, LNCS 6632*, California, USA, 14-18 August, pp. 224-245.
23. Fuchsbauer, G., Pointcheval, D., and Vergnaud, D. Transferable constant-size fair e-cash. *CANS 2009, LNCS 5888*, Kanazawa, Japan, 12-14 December, pp. 226-247.
24. Groth, J. and Sahai, A. Efficient non-interactive proof systems for bilinear groups. *EUROCRYPT 2008, LNCS 4965*, Istanbul, Turkey, 13-17 April, pp. 415-432.
25. Shi, L., Carbunar B. and Sion, R. Conditional e-cash. in financial cryptography and data security. *FC 2007, LNCS 4886*, Scarborough, Trinidad and Tobago, 12-16 February, pp. 15-28.
26. Okamoto, T. and Ohta, K. Universal electronic cash. *CRYPTO 1991, LNCS 576*, California, USA, 11- 15 August, pp. 324-337.
27. Canard, S., Gouget, A. and traore, J. Improve- ment of efficiency in (unconditional) anonymous transferable ecash. *FC 2008, LNCS 5143*, Cozumel, Mexico, 28-31 January, pp. 202-214.
28. Canard, S., Delerablee, Cecile, Gouget, A., Hufschmitt, E., Laguillaumie, F., Sibert, H., traore, J. and Vergnaud, D. Fair E-Cash: Be Compact, Spend Faster. *ISC 2009, LNCS 5735*, Pisa, Italy, 7-9 September, pp. 294-309.
29. Abe, M., Groth, J., Haralambiev, K. and Ohkubo, M. Optimal structure-preserving signatures in asymmetric bilinear groups. *CRYPTO 2011, LNCS 6841*, California, USA, 14-18 August, pp. 649-666.
30. Okamoto, T. and Ohta, K. Diposable zero- knowledge authentications and their applications to untraceable electronic cash. *CRYPTO 1989, LNCS 435*, California, USA, 20-24 August, pp. 481-496.
31. Belenkiy, M., Chase, M., Kohlweiss, M. and Lysyan- skaya, A. Compact E-Cash and Simulatable VRFs Revisited. *Pairing 2009, LNCS 5671*, Califor- nia, USA, 12-14 August, pp. 114-131.
32. Camenisch, J. and Damgard, I. Verifiable encryption, group encryption, and their applications to group signatures and signature sharing schemes. *ASIACRYPT 2000, LNCS 1976*, Kyoto, Japan, 3-7 December, pp. 331-345.