More Compact E-Cash with Efficient Coin Tracing

Victor K. Wei

Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong
kwwei@ie.cuhk.edu.hk

May 26, 2005

Abstract. In 1982, Chaum [21] pioneered the anonymous e-cash which finds many applications in e-commerce. In 1993, Brands [8–10] and Ferguson [30, 31] published on single-term offline anonymous e-cash which were the first practical e-cash. Their constructions used blind signatures and were inefficient to implement multi-spendable e-cash. In 1995, Camenisch, Hohenberger, and Lysyanskaya [12] gave the first compact 2^ℓ-spendable e-cash, using zero-knowledge-proof techniques. They left an open problem of the simultaneous attainment of $O(1)$-unit wallet size and efficient coin tracing. The latter property is needed to revoke *bad* coins from over-spenders. In this paper, we solve [12]'s open problem, and thus enable the first practical compact e-cash. We use a new technique whose security reduces to a new intractability assumption: the *Decisional Harmonically-Tipped Diffie-Hellman (DHTDH) Assumption*.

1 Introduction

In 1982, Chaum[21] pioneered the anonymous (untraceable) e-cash, which finds many applications in e-commerce and m-commerce. In 1993, Brands[8–10] and Ferguson [31, 30] published on single-term offline anonymous e-cash which has at least the following properties:

1. *unforgeability* of the e-cash;
2. *offline verifiability* and "after-the-fact" *tracing* of over-spenders;
3. *irrevocable anonymity*: An honest user who does not over-spend cannot have his anonymity revoked by any authority;
4. *single-term*, i.e. efficiency of computation (resp. storage, bandwidth, tracing) is $O(1)$ unit, where each unit is λ_s bits and λ_s is the security parameter.

The *offline-ness* correspond to the following application design scenario: The system is by design, or stressed by traffic, to an offline state such that e-coins are verified without reference to past coins in any online database or concurrent coins. Then a necessary deterrent against over-spending is to compute the identity of the over-spender after the system eventually returns online with access to all past and concurrent coins. The *nonce* technology often plays an ultimate role in securing offline anonymous e-cash.

Most existing e-cash implementations used blind signatures [21] and were difficult to achieve 2^ℓ-spendable e-cash with good efficiencies [12]. The *compact 2^ℓ-spendable e-cash* possesses at least the following properties:

1. security and efficiency properties of the single-term offline anonymous e-cash listed above.
2. *compactness*, i.e. it can be computed (resp. stored, transmitted, traced) in $O(1)$ units, independent of the number of allowed usages 2^ℓ.
3. reductionist security proofs;
4. anonymity (i.e. zero-knowledge) of both the spender's identity and the usage (spending) count;
5. efficient *coin tracing*.

In *coin tracing*, Bank seeks to revoke unspent e-coins from an over-spender by maintaining an online coin revocation list. Alternative revocation technologies such as the dynamic revocation in [14] is not suitable because it requires all honest users to alter their e-coins in order to revoke each over-spent e-coin. There are simply too many e-coins. In this paper, we focus on a solution where all coins from an over-spender are traced and added to a *coin revocation list*. An efficient coin tracing algorithm to update this list is essential.

An example slow over-spender detection system is to compare the spending at hand against every past spending. The resulting complexity is proportional to the number of all past spendings, which could be an extremely large number. An example fast over-spender detection is to implement a coin serial number which is a one-to-one mapping of the tuple (coin, $\mathsf{count}_{\mathsf{coin}}$), i.e. the coin and its usage counter. To detect over-spender, merely query a hash table containing all past serials. A match triggers the tracing of the over-spender. The average time of a hash table search query is constant, independent of , or very weakly dependent on, the number of past spendings. The hash table can be filled to a high constant percentage.

Recently, Camenisch, Hohenberger, and Lysyanskaya [12] presented the literature's first compact e-cash scheme which satisfies all but one of the above properties. Their System One achieved all but efficient coin tracing. Their System Two achieved all but $O(1)$ compactness – its storage (resp. computation, bandwidth) is $O(\ell)$ units. Their System One combined CL02 [15]'s signature, Dodis-Yampolskiy [29]'s verifiable random function (VRF), and an innovative system of *serial number* and *security tag*. Their System Two also used Ateniese, Fu, Green, and Hohenberger [2]'s re-encryption, Camenisch and Damgård [11]'s generally applicable verifiable encryption.

In this paper, we improve upon [12]'s result by achieving all security and efficiency properties listed above, including the simultaneous achievement of efficient (in fact $O(1)$-time) coin tracing and compact (in fact $O(1)$-unit) wallet size. In details, our **contributions** are

1. We construct the literature's first practical compact e-cash scheme satisfying all properties listed above, including the simultaneous achievement of efficient (in fact $O(1)$-time) coin tracing and compact (in fact $O(1)$-unit) wallet size. We combined [12]'s techniques, with a new technique to trace the spender's secret key, in our construction. The security is reduced to a new and strong intractability assumption, the *Decisional Harmonically-Tipped Diffie-Hellman (DHTDH) Assumption*.
2. Using the same new technique, we construct an even more efficient solution but with a slightly weaker anonymity, *bout-wise anonymity* where an e-coin spending is only zero-knowledge about its spender, but the number of usages of a 2^ℓ-spendable coin is revealed. Bout-wise anonymity actually has its own applications where the above compact e-cash is not suitable. For example, bout-wise e-cash is more suitable in multi-round spendings where users are limited to a certain amount of purchases per round (bout) and surpluses from one round cannot be used to increase the spending ceiling of another round.
3. We also adapt [12]'s result and [44]'s result from a Strong RSA setting to a pairings setting.
4. We provide the most thorough security model of compact e-cash to date.

Organization: Section 2 reviews intractability assumptions. Section 3 presents the security model. Section 4 contains the constructions and reductionist security proofs. Section 5 contains discussions, applications, and conclusions.

Related literatures.

There are mainly two approaches to implement offline anonymous e-cash: by blind signatures [21, 8–10, 31, 30, 46] or by zero-knowledge proofs [37, 13, 14, 45, 39]. In either approach, there are mainly three stages: Withdraw, Spend, and Deposit. The two approaches differ in which stage the spender's identity is hidden. In the former, user obtains a blind signature from Bank which serves as an e-coin. The identity hiding starts in the first stage, Withdraw. In the latter, user obtains a certificate (i.e. certified public key) from Bank in Withdraw which serves as an e-coin. Then in Spend, user presents a zero-knowledge proof-of-knowledge of a certificate/coin and of its corresponding user secrete key. The hiding of the spender identity is in the second stage, Spend, by virtue of the zero-knowledge.

Literatures on e-cash from blind signatures included at least the following. Brands[8–10] presented single-term offline anonymous e-cash. Ferguson[31, 30] also published on single-term offline anonymous e-cash. Chaum[23] studied e-checks. Chaum, Fiat, and Naor [24] studied untraceable e-cash. Chaum [22]'s blind signature was a pivotal technology for implementing most previous e-cash schemes. Chaum and Pedersen [25] presented transferred e-cash which grew in size. Chaum and Pedersen [26] presented wallet with observers, Chan, et al. [20] constructed a kind of divisible e-cash. Franklin and Yung [32] studies security notions of e-cash. Okamoto and Ohta [43] studies zero-knowledge for e-cash. Tsiounis [46]'s PhD dissertation on e-cash provides a wealth of information on e-cash. Okamoto's divisible e-cash [42].

The core mechanism in the latter approach, zero-knowledge proof-of-knowledge of a certificate, is also the core mechanism of group signatures, e.g. [27, 3, 4, 1, 15, 6, 35, 36]. The fact that an e-cash scheme can be

viewed as a certain kind of group signatures, and a certain kind of e-cash schemes can be constructed from almost any group signature has been observed in the literature at least since [37, 13, 14, 28, 45, 39] and [38, 40, 17, 16]. In the above e-cash from group signature, the identity of the spender is required to be escrowed to an Open Authority (OA). Then it relies on *linkability* to detect a double spending and then it relies on OA to trace its identity. Unfortunately, OA also has the unnecessary power to trace spenders who have not over-spent. These e-cash schemes do not have irrevocable anonymity for the honest spenders, and therefore do not match the security properties of the 1993 e-cash of [8–10, 31, 30]. Jarecki and Shmatikov [33] proposed a good escrow scheme. Kiayias, et al. [34]'s traceable group signature allows the OA who is also the Bank to publish a coin revocation list.

Recently, Teranish, et al. [44] constructed a group authentication scheme which can be turned into a group signature and then into an e-cash scheme with the following properties: There is no OA, and an honest spender has irrevocable anonymity. Over-spenders can be traced without any trapdoor. In fact, [44]'s result implies a bout-wise compact e-cash even though the authors did not observe this explicitly.

The compact e-cash of [12] first achieved 2^ℓ-spendable e-cash that is not bout-wise anonymous. It used techniques similar to [44] to achieve irrevocable anonymity for honest signers and efficient tracing of double spenders. However, [12] surpassed previous results by using [29]'s Verifiable Random Function (VRF) to achieve zero-knowledge of both the spender identity and the coin usage counter value.

Intuitions of our result. Our main result improves [12] by adding a harmonic relation between two user secret keys. In [12], the e-coin spending is the following signature proof-of-knowledge:

$$SPK\{(A, e, x_1, x_2, x_3, J') : A^e h_1^{x_1} h_2^{x_2} h_3^{x_3} = h_0 \in QR_N \ \wedge \ S = \mathbf{g}_2^{1/(J'+x_2')} \in G_S \\ \wedge \ T = h_1^{x_1} h_3^{R/(J'+x_3)} \in QR_N \ \wedge \ x_2 = x_2' \ \wedge \ 1 \le J' \le 2^\ell \ \wedge \ |e - 2^{\ell_1}| \le 2^{\ell_2}\}(M) \quad (1)$$

The first relation is the well-known proof-of-knowledge of a certificate (A, e, x_1, x_2, x_3) using CL02 [15]'s group signature, (x_1, x_2, x_3) and $(h_1^{x_1}, h_2^{x_2}, h_3^{x_3})$ form the user sk-pk pair, N is the product of two safe primes of similar length, and e is a prime in a suitable range[1, 15]. The *coin serial number* S and the *coin security tag* T are [29]'s VRF (Verifiable Random Functions) which reveals zero-knowledge about the certificate or the coin usage count J'. The group G_S is a known-order group satisfying hte q-DHHI Assumption [29]. When the 2^ℓ-spendable coin/certificate is over-spent, a value J', $1 \le J' \le 2^\ell$ is used twice, resulting in identical serials S and two corresponding tags $T = h_1^{x_1} h_3^{R/(J'+x_3)}$ $\tilde{T} = h_1^{x_1} h_3^{\tilde{R}/(J'+x_3)}$. Then $T^{\tilde{R}} \tilde{T}^{-R} = h_1^{x_1(\tilde{R}-R)}$ which can be used to trace the user public key $h_1^{x_1}$ and then trace the user. But it cannot easily compute the user secret key, resulting in inefficient coin tracing. Further efforts in [12] achieved coin tracing in their System Two by using the general VE (Verifiable Encryption) of [11], but at an at least ℓ-fold increase in complexity.

We note that [44] used $S = u^x$ and $T = g^{xR} v^x$. Double spenders are detected by duplicated S and traced by $T^{-1}\tilde{T} = (g^x)^{-R+\tilde{R}}$. The user sk-pk pair is (x, g^x). Note it is difficult to implement 2^ℓ-spendable e-cash which keeps both the spender identity and the coin usage count zero-knowledge. E-cash solutions adapted from the original group authentication in [44] is 2^ℓ-size for 2^ℓ-spendable, or a straightforward modification (see our Appendix) achieves $O(1)$-unit size but loses zero-knowledge of the coin usage count.

Our new result modifies [12] to the following:

$$SPK\{(A, e, x_1, x_2, J') : A^e h_1^{x_1} h_2^{x_2} = h_0 \in QR_N \ \wedge \ S = \mathbf{g}_2^{1/(J'+x_2')} \in G_S \\ \wedge \ x_2 = x_2' \ \wedge \ 1 \le J' \le 2^\ell \ \wedge \ |e - 2^{\ell_1}| \le 2^{\ell_2} \ \wedge \ (J' + x_2)^{-1} = c x_1^{-1} + z\}(M) \quad (2)$$

where c is the challenge of the SPK proof system. The last relation is the crucial new technique. Over spending is detected by duplicated S and traced by solving x_1 and x_2 in the linear system:

$$(J' + x_2)^{-1} = c x_1^{-1} + z \quad (3)$$
$$(J' + x_2)^{-1} = \tilde{c} x_1^{-1} + \tilde{z} \quad (4)$$

In essence, over-spending results in a kind of *forking simulation* of the spender to extract two of its secret keys x_1 and x_2. Note the user secret key is traced directed, while [12, 44] traced the user public key. The consequence is the simultaneous achievement of efficient coin tracing (which is an implication of efficient

tracing of user secret key) and compactness. The down side is that a new and quite strong intractability assumption is needed, the *Decisional Harmonically-Tipped Diffie-Hellman (DHTDH) Assumption*. Details below.

2 Preliminaries: Intractability assumptions

For preliminaries on bilinear maps, zero-knowledge proofs, pseudorandom functions, CL02 signatures [15] that we will need, readers are referred to [12]'s Section 3, Preliminaries. Below, we review old and new intractability assumptions that we will need.

Convention: Unless otherwise noted, the *XYZ Assumption* is that no PPT algorithm can solve a random instance of the XYZ Problem with non-negligible probability over random guessing. Below, G_a, G_b, G_c are arbitrary groups of known or unknown orders..

The *co-Decisional Diffie-Hellaman Problem in* $G_a \times G_b$, denoted *co-DDH*(G_a,G_b), is, given random $g_1, g_1^{x_1} \in G_a$ and $g_2 \in G_b$, to distinguish $g_2^{x_2}$ from random. When $G_a = G_b$, it becomes the *Decisional Diffie-Hellman Problem in* G_a, denoted *DDH*(G_a). When there is a pairing $\hat{\mathbf{e}} : G_a \times G_a \rightarrow G_b$, it becomes the DDHV Problem, V=Variant [41].

The *q-DHI*(G_a,G_b) *(Diffie-Hellman Inverted)* (resp. *q-DDHI*(G_a,G_b) *(Decisional Diffie-Hellman Inverted) Problem*) is: Given $g, g^{\gamma^i} \in G_a$, $0 \leq i \leq q$, $h \in G_b$, to compute $h^{1/\gamma}$ (resp. to distinguish $h^{1/\gamma}$ from random). When there is a pairing $\hat{\mathbf{e}} : G_a \times G_a \rightarrow G_b$, then it becomes the q-DHIV (resp. q-DDHIV) Problem.

The *Decisional Linear* (resp. *Inverted, Harmonic*) *Diffie-Hellaman Problem in* $G_a \times G_b \times G_c$, denote *DLDH*($G_a$, G_b, G_c) (resp. *DIDH*(G_a,G_b,G_c), *DHDH*(G_a,G_b,G_c)) is, given random $g_1, g_1^{x_1} \in G_a$, $g_2, g_2^{x_2} \in G_b$, $h \in G_c$, to distinguish $h^{x_1+x_2}$ (resp. $h^{x_1+x_2^{-1}}$, $h^{x_1^{-1}+x_2^{-1}}$) from random. When $G_a = G_b = G_c$, it becomes the DLDH (resp DIDH, DHDH) Problem in G_a[6], denoted DLDH(G_a) (resp DIDH(G_a), DHDH(G_a)) [6]. When there is a pairing $\hat{\mathbf{e}} : G_a \times G_b \rightarrow G_c$, then it becomes the DLDHV Problem, with V=Variant [41].

The *Decisional Linearly-Tipped* (resp. *Inversely-Tipped, Harmonically-Tipped*) *Diffie-Hellaman Problem*, denoted *DLTDH*(G_a, G_b) (resp. *DITDH*(G_a, G_b), *DHTDH*(G_a, G_b)) is given random g_1, $g_1^{x_1} \in G_a$, $g_2 \in G_b$, order(G_a) = order(G_b), and c, z satisfying $x_2 = cx_1 + z$ (resp. $x_2 = cx_1^{-1} + z$, $x_2^{-1} = cx_1^{-1} + z$), to distinguish $g_2^{x_2}$ from random.

The *Strong RSA Assumption* There exists no PPT algorithm which, on input a random λ-bit safe product N and a random $z \in QR(N)$, returns $u \in \mathbb{Z}_N^*$ and $e \in \mathbb{N}$ such that $e > 1$ and $u^e = z (\mathrm{mod} N)$, with non-negligible probability and in time polynomial in λ. Let $e : G_1 \times G_2 \rightarrow G_3$ be a bilinear mapping.

The *q-SDH Assumptions* The *q-Strong Diffie-Hellman Problem (q-SDH)* is the problem of computing a pair $(g_1^{1/(\gamma+x)}, x)$ given $g_1 \in G_1$, and g_2, g_2^{γ}, $g_2^{\gamma^2}$, $\cdots$, $g_2^{\gamma^q} \in G_2$.

Several of the above DDH-assumptions are believable even in GDH groups (where the DDH Assumption fails and the CDH Assumption hold) relative to contemporary technology, include DLDH, DLDHV, DHTDH, DHHI, DHHIV. Note the DLTDH(G, G) Assumption (resp. the DITDH(G_1,G_2) Assumption) implies the DDH(G,G) Assumption (resp. the co-DDH(G_1,G_2) Assumption), and therefore it fails in GDH groups.

3 Security Model

We follow mainly the security model of [4, 12].

3.1 Syntax and correctness

Syntax: An *offline* 2^ℓ*-spendable e-cash* is a tuple (Init, BKeygen, UKeygen, Withdraw, Spend = (Sign, Vf), Link, Trace, RevoCoin) where

1. Init: Upon input a security parameter λ_s, output system parameters param which includes at least a pairing $\hat{\mathbf{e}} : g_1 \times G_2 \rightarrow G_3$, the number of usages per e-coin 2^ℓ, hashing functions, discrete logarithm bases, group orders, lengths and ranges, λ_s, the archive of all past spendings Archive, the online coin-spending revocation list CoinRevoList, the database of all legitimate customers with their public keys and coins withdrawal transactions CustomerDB. Below, we assume param is included in each Protocol's input by default unless specified explicitly otherwise.

2. $\mathsf{BKeygen}(B)$: outputs bank key pair $(\mathsf{bsk}_B, \mathsf{bpk}_B)$ and insert bpk_B to param.
3. $\mathsf{UKeygen}(U)$: outputs user key pair $(\mathsf{sk}_U, \mathsf{pk}_U)$.
4. $\mathsf{Withdraw}$ is a pair of interactive protocols $(\mathsf{U}(\mathsf{bpk}, \mathsf{sk}), \mathsf{B}(\mathsf{pk}, \mathsf{bsk}))$ where, in the end, User obtains fresh e-coins coin; Bank updates User's account balance and $\mathsf{CustomerDB}$.
5. $\mathsf{Spend} = (\mathsf{Sign}, \mathsf{Vf})$ is a pair of interactive protocols $(\mathsf{Sign}(\mathsf{sk}_U, \mathsf{coin}_{U,i}, \mathsf{count}_{U,i}), \mathsf{Vf}(\sigma))$. At the conclusion of interactions, either (i) Merchant outputs 0 and aborts; or (ii) Merchant outputs 1 and receives e-coin spendings σ and User obtains purchases and increments $\mathsf{count}_{U,i}$ by one, where $\sigma = \mathsf{Sign}(\mathsf{sk}_U, \mathsf{coin}_{U,i}, \mathsf{count}_{U,i})$.
6. $\mathsf{Deposit}$ is a pair of interactive protocols $(\mathsf{M}(\mathsf{msk}, \sigma, \mathsf{bpk}), \mathsf{B}(\mathsf{mpk}, \mathsf{bsk}))$: Merchant deposits σ. Bank inserts the spending σ to $\mathsf{Archive}$, and updates Merchant's account balance.
7. $\mathsf{Link}(\sigma, \mathsf{Archive})$: Verify $\mathsf{Vf}(\sigma) = 1$, the output 0 for no over-spending non-detection; or 1 and $\sigma' \in \mathsf{Archive}$ for overspending detected between σ and σ'.
8. $\mathsf{Trace}(\sigma, \sigma', \mathsf{CustomerDB})$: Verify $\mathsf{Link}(\sigma, \mathsf{Archive}) = (1, \sigma')$, then output a user public key $\mathsf{pk} \in \mathsf{CustomerDB}$.
9. $\mathsf{RevoCoin}(\sigma, \sigma', \mathsf{CoinRevoList})$: Verify $\mathsf{Link}(\sigma, \mathsf{Archive}) = (1, \sigma')$, then output 2^ℓ coin serial numbers and insert them to $\mathsf{CoinRevoList}$.
10. $\mathsf{OnlineVf}(\sigma, \mathsf{CoinRevoList})$: Output $\mathsf{Vf}(\sigma, \mathsf{Archive})$. In addition, when $\mathsf{Vf}(\sigma, \mathsf{Archive}) = 1$, output $\mathsf{revoked}$ if $\sigma \in \mathsf{CoinRevoList}$ or $\mathsf{not\text{-}revoked}$ otherwise.

We assume the three online databases $\mathsf{CustomerDB}$, $\mathsf{Archive}$, $\mathsf{CoinRevoList}$ are honestly maintained. Each usage of a 2^ℓ-spendable *e-coin* is called a *spending*. typically denoted σ in this paper.

Correctness An e-cash scheme has the following correctnesses

1. *Withdraw Correctness*: Honest Bank and User with legitimate key pairs, following $\mathsf{Withdraw}$ Protocol, will output correct results.
2. *Spend Correctness*: Honest User, with legitimately obtained coins and not over-spending, and Merchant with legitimate key pairs, following Protocol Spend, will output correct results, i.e. coin acceptance.
3. *Deposit Correctness*: Honest Merchant, with legitimately obtained (untainted) coins from some user who have obtained the coins legitimately before and not over-spending, and Bank with legitimate key pairs, following Protocol $\mathsf{Deposit}$, will output correct results.
4. *Linking Correctness*: An honest user with honestly obtained coins but over-spending them, will be detected by Protocol Link.
5. *Tracing Correctness*: An honest user with honestly obtained coins but over-spending them, will be traced by Protocol Trace.
6. *Coin-Tracing Correctness*: An honest user with honestly obtained coins but over-spending them, the coin will be be coin-traced by Protocol $\mathsf{RevoCoin}$; and the spending of the coin will cause $\mathsf{OnlineVf}$ to output $\mathsf{revoked}$ after $\mathsf{RevoCoin}$'s insertion to $\mathsf{CoinRevoList}$.

In summary, an e-cash scheme is *correct* if it has all the above correctnesses.

3.2 Attacker tools: Oracles

1. Corrupt Bank Oracle $\mathcal{CBO}(B)$: obtain Banks secrete key. We assume the Bank remains honest in carrying out Protocol $\mathsf{Withdraw}$. The secret key is stolen and the Adversary can observe the Bank, but it cannot control or alter Bank's operations.
2. Corrupt User Oracle: $\mathcal{CUO}(U)$: obtain user's secrete key.
3. Bout Decisional Oracle $\mathcal{BDO}(\mathsf{coin}, J)$: returns 1 for yes that coin is in bout (coin usage count) J; or 0 for no.
4. Spend Oracle $\mathcal{SO}(U, \mathsf{coin})$: Upon inputs user U, one of its coins coin which has not been completely spent, output a valid coin spending σ.
5. Bout-Wise Spend Oracle $\mathcal{SOBW}(U, \mathsf{coin}, J)$: Upon inputs user U, one of its coins coin, which has not been depleted, and the coin's usage count J, $0 \leq J < 2^{\ll}$, output a valid coin spending σ.

Unless otherwise stated explicitly, all queries to $\mathcal{SO}$ must have (U, coin) that are contained in the list of valid user and coins; there cannot be queries to $\mathcal{SO}$ that duplicates a previous query input (U, coin, J); the number of $\mathcal{SO}$ queries with the same (U, coin) cannot exceed 2^ℓ, and lower in some specific security experiments.

Also, we do not allow the attacker to query $\mathcal{SO}$ and $\mathcal{CUO}$ with the same user U ever. This effectively rules out *adaptive attackers* [19] for the simplicity of the present presentation. The topic of adaptive attackers and the topic of universal composability[18] for the current protocols are left for future research.

3.3 Security notions and attack games

3.3.1 Irrevocable anonymity and bout-wise irrevocable anonymity

Experiment IA (Irrevocable Anonymity)

1. (*Initialization Phase*): Simulator $\mathcal{S}$ initializes one bank, a number of users, and a number of coins for these users.
2. (*Probe-1 Phase*): Adversary $\mathcal{A}$ queries $\mathcal{CBO}$ $\mathcal{CUO}$, $\mathcal{BDO}$, $\mathcal{SO}$, $\mathcal{SOBW}$, $\mathcal{H}$ in arbitrary interleaf.
3. (*Gauntlet Phase*): $\mathcal{A}$ sends two tuples $(U_i, \mathsf{pk}_{U_i}, \mathsf{coin}_i, \mathsf{count}_i)$, where $(U_i, \mathsf{pk}_{U,i}, \mathsf{coin}_i) \in \mathsf{CustomerDB}$ and $\mathcal{BDO}(\mathsf{coin}_1, J_1) = \mathcal{BDO}(\mathsf{coin}_1, J_2) = \mathcal{BDO}(\mathsf{coin}_2, J_1) = \mathcal{BDO}(\mathsf{coin}_2, J_2) = 0$ at this time, $i = 1, 2$, to $\mathcal{S}$. $\mathcal{S}$ confirms entries in $\mathsf{CustomerDB}$ and $\mathcal{BDO}$ query results, flips a fair coin $b \in \{0, 1\}$, computes a spending σ_b for $(U_b, \mathsf{pk}_{U_b}, \mathsf{coin}_b, \mathsf{count}_b)$ and sends σ_b to $\mathcal{A}$.
4. (*Probe-2 Phase*): Adversary $\mathcal{A}$ queries $\mathcal{CBO}$ $\mathcal{CUO}$, $\mathcal{BDO}$, $\mathcal{SO}$, $\mathcal{SOBW}$, $\mathcal{H}$ in arbitrary interleaf.
5. (*End Game*): $\mathcal{A}$ makes its estimate $\hat{b}$ on b.

Subsequently, U_b (resp. coin_b, J_b) is called the *gauntlet user* (resp. *gauntlet coin*, *gauntlet bout*).

In order to win, these prerequisites must be satisfied by $\mathcal{A}$:

1. The user U_1 (resp. U_2) has not been queried to $\mathcal{CUO}$. Oracle $\mathcal{SOBW}$ has never been queried.
2. The number of $\mathcal{SO}$ queries with coin_1 (resp. coin_2) does not exceed $2^\ell - 2$.
3. The pairs (coin_1, J_1), (coin_1, J_2), (coin_2, J_1), and $\mathcal{BDO}(\mathsf{coin}_2, J_2)$ have not been queried to $\mathcal{BDO}$ in the Probe-2 Phase.

After satisfying the prerequisites, $\mathcal{A}$ *wins* Experiment IA-U if $\hat{b} = b$, $U_1 \neq U_2$ and $J_1 = J_2$. It *wins* Experiment IA-C if $\hat{b} = b$, $U_1 = U_2$, $C_1 \neq C_2$, $J_1 = J_2$. It *wins* Experiment IA-J if $\hat{b} = b$, $U_1 = U_2$, $C_1 = C_2$, $J_1 \neq J_2$. $\mathcal{A}$'s *advantage* in Experiment IA-U (resp. IA-C, IA-J) is his probability, minus 1/2, of winning that Experiment.

Definition 1. *An offline 2^ℓ-spendable e-cash is* irrevocably anonymous *if no PPT algorithm has a non-negligible advantage in any of Experiments IA-U, IA-C, IA-J.*

Remark: Experiment IA allows the corruption of Bank. Were there an OA (Open Authority) or other related authorities in the Syntax, their corruption would be also allowed for experimenting irrevocable anonymity.

Experiment Boutwise-IA (BIA) is the same as Experiment IA except $\mathcal{A}$ is allowed to query $\mathcal{SOBW}$ but not $\mathcal{SO}$, and query $\mathcal{SOBW}$ at most once for each triple (U, coin, J). $\mathcal{A}$'s *advantage* is his probability of winning the modified Experiment BIA-U or BIA-C. There is no requirement to win Experiment BIA-J.

Definition 2. *An offline 2^ℓ-spendable e-cash is* bout-wise irrevocably anonymous *if no PPT algorithm has a non-negligible advantage in either Experiment BIA-U or BIA-C.*

3.3.2 Full Traceability

Experiment FT (Full Traceability)

1. (*Initialization Phase*): Simulator $\mathcal{S}$ initializes one bank, a number of users, and a number of coins for these users.
2. (*Probe Phase*): Adversary $\mathcal{A}$ makes q_B (resp. q_U, q_B, q_S, q_H) queries to $\mathcal{CBO}$ (resp. $\mathcal{CUO}$, $\mathcal{BDO}$, $\mathcal{SO}$, $\mathcal{H}$) in arbitrary interleaf.
3. (*Delivery Phase*): $\mathcal{A}$ delivers coin-spendings σ_i, $1 \leq i \leq q_C k + 1$, none of which is an output of $\mathcal{SO}$.

Let q_C be the total number of e-coins issued to the q_U users corrupted by $\mathcal{A}$. $\mathcal{A}$ *wins* Experiment FT if the following all hold:

1. $\mathsf{Vf}(\sigma_i) = 1$ and $\mathsf{Link}(\sigma_i, \mathsf{Archive}) = 0$, for all i and j, $1 \leq i < j \leq q_C k + 1$;
2. $q_B = 0$, i.e. Banks is not corrupted;

$\mathcal{A}$'s *advantage* is his probability of winning Experiment FT.

Definition 3. *An offline 2^ℓ-spendable e-cash is* fully traceable *provided no PPT algorithm has a non-negligible advantage in Experiment FT(2^ℓ).*

3.3.3 Strong non-frameability

Experiment SNF

1. (*Initialization Phase*): Simulator $\mathcal{S}$ initializes one bank, a number of users, and a number of coins for these users.
2. (*Probe Phase*): Adversary $\mathcal{A}$ makes q_B (resp. q_U, q_B, q_S, q_H) queries to $\mathcal{CBO}$ (resp. $\mathcal{CUO}$, $\mathcal{BDO}$, $\mathcal{SO}$, $\mathcal{H}$) in arbitrary interleaf.
3. (*Delivery Phase*): $\mathcal{A}$ delivers a user identity U which has never been queried to $\mathcal{CUO}$, one of U's coin coin in $\mathsf{CustomerDB}$ which has never been queried to $\mathcal{SO}$, and a spending σ which is not an output of $\mathcal{SO}$.

Let $\mathsf{Archive}'$ be the result of inserting any 2^ℓ honest spending of the coin coin by user U to $\mathsf{Archive}$ (at the end of the game). $\mathcal{A}$ wins Experiment SNF-U if $\mathsf{Link}(\sigma, \mathsf{Archive}') = (1, \sigma')$ for some $\sigma' \in \mathsf{Archive}'$ and $\mathsf{Trace}(\sigma, \sigma', \mathsf{CustomerDB}) = \mathsf{pk}_U$. $\mathcal{A}$'s advantage in Experiment SNF-U is his probability of winning Experiment SNF-U. $\mathcal{A}$ wins Experiment SNF-C if $\mathsf{Link}(\sigma, \mathsf{Archive}') = (1, \sigma')$ for some $\sigma' \in \mathsf{Archive}'$ and $\mathsf{RevoCoin}(\sigma, \sigma', \mathsf{CoinRevoList})$ outputs $\mathsf{Archive}' \setminus \mathsf{Archive}$. $\mathcal{A}$'s advantage in Experiment SNF-C is his probability of winning Experiment SNF-C.

Definition 4. *An offline 2^ℓ-spendable e-cash is* strongly user non-frameable *(resp.* strongly coin non-frameable*) provided no PPT algorithm has a non-negligible advantage in Experiment SNF-U (resp. SNF-C). It is* strongly non-frameable *provided it is both strongly user non-frameable and strongly coin non-frameable.*

Remark: There exists a scenario where a user can be indicted by Trace, but the user can vindicate himself in a public *trial* overseen by a judge [4, 12]. In this scenario, the user remains "non-frameable". We have constructed pedagogical e-cash which instantiates this scenario. However, indictment and vindication remains a hassle and a vulnerability to DoS (Denial-of-Service) attacks. In this paper, we restrict ourselves to the higher goal of strong non-frameability where Trace indict the truly guilty with overwhelming probability.

3.3.4 Security.

In summary

Definition 5. *An offline 2^ℓ-spendable e-cash is* secure *provided it is* correct*,* irrevocably anonymous*,* fully traceable*, and* strongly non-frameable*. It is* bout-wise secure *it it is* correct*,* bout-wise irrevocably anonymous*,* fully traceable*, and* strongly non-frameable*.*

3.4 Efficiency goals

The storage of each 2^ℓ-spendable coin should be $O(\lambda_s + \ell)$ bits, where $\ell < \lambda_s$. The complexity of Link (resp. Trace) should be $O(1)$. Protocol $\mathsf{CoinTrace}$ should output 2^ℓ coin traces in time $O(2^\ell)$, or $O(1)$ per spending traced.

3.5 Comparing security notions with [12]

[12]'s *Balance* means coalition-resistant unforgeability which is included in our full traceability. Their *identification of double-spenders*, *tracing of double-spenders*, and *exculpability* are included in our linking correctness, tracing correctness and strong non-frameability.

4 New Constructions of Offline 2^ℓ-Spendable E-Cash

We construct offline 2^ℓ-spendable e-cash schemes with efficient size, bandwidth, computation, and coin tracing, and reduct its security to intractability assumptions. Section 4.1 contains a generic construction. Section 4.2 contains an instantiation in pairings. Section 4.3 contains another instantiation in a Strong RSA setting. Section 4.4 contains a different construction: a bout-wise secure offline 2^ℓ-spendable e-cash.

4.1 Generic Constructions

Following the intuitions at the end of Section 1, we construct our offline 2^ℓ-spendable e-cash with the prescribed efficiency and we reduct their security to intractability assumptions.

Protocol CEC-HT: Protocols Init, BKeygen, UKeygen: Let $\hat{\mathbf{e}} : G_1 \times G_2 \to G_3$ be a pairing with order(G_1)=order(G_2)=order(G_3)=q_1. The Bank's sk-pk pair is ($\gamma \in Z_{q_1}$, $(u, u^\gamma) \in G_2 \times G_2$). The user sk-pk pair is $((x_1, x_{2,0}), (h_1^{x_1}, h_2^{x_{2,0}}) \in G_1 \times G_1)$. Assume all system discrete logarithm bases are fairly generated: e.g. $g_i = \mathcal{H}('g', i) \in G_1$, $h_i = \mathcal{H}('h', i) \in G_1$, $u_i = \mathcal{H}('u', i) \in G_2$, $\mathbf{g}_i = \mathcal{H}('\mathbf{g}', i) \in G_3$. Sampling a random element of G_3 directly may cost time, but we can afford such in generating a system parameter. Assume ℓ is sufficiently large but sufficiently smaller than λ_s. We have in mind $\ell = \Theta(\lambda_s)$.

Withdraw **Protocol:** User present its public key and proof knowledge of its corresponding secret key. Bank verifies, and then randomly generates $x_{2,1}$ and a certificate (A, e) certifying the user public keys $(h_1^{x_1}, h_2^{x_2})$, where $x_2 = x_{2,0} + x_{2,1}$.

Spend = (Sign, Vf): **Protocol** Sign(sk, coin, count) accepts inputs user secret key $\mathsf{sk} = (x_1, x_2)$, an e-coin (a.k.a. certificate) $\mathsf{coin} = (A, e)$ corresponding to it, the number of times the 2^ℓ-spendable coin has been spent before, $\mathsf{count} = J$, $0 \le J < 2^\ell$. For each coin coin, Protocol Spend generates a prime p_{coin} of length close to ℓ to be used in all spendings of coin. Let $J' = p_{\mathsf{coin}}(J+1) \bmod 2^\ell$. Then Sign sends the following non-interactive proof-of-knowledge to Vf:

$$\begin{aligned}\sigma = SPK\{(A, e, x_1, x_2, J) : (A, e) \text{ is cert for user public key } (h_1^{x_1}, h_2^{x_2}) \\ \wedge\ 1 \le J \le 2^\ell\ \wedge\ S = \mathbf{g}_S^{1/(J+x_2)}\ \wedge\ x_1^{-1} = c(J+x_2)^{-1} + z\}(\mathsf{param}, nonce)\end{aligned} \tag{5}$$

where c is the *challenge* of the above proof system.

Protocol Vf verified the proof-of-knowledge. Protocol Link links two signatures with the same serial S, which is bound to happen when a coin is spent more than 2^ℓ times. When that happens, Protocol Trace obtains the following two equations from the two spendings corresponding to the repeated J, $x_1^{-1} = c(J+x_2)^{-1} + z$ and $x_1^{-1} = \tilde{c}(J+x_2)^{-1} + \tilde{z}$, and solve for x_1 to obtain a user public key $h_1^{x_1}$. Protocol CoinTrace then computes J and x_2 and traces all 2^ℓ spendings of the over-spent coin.

Below, we instantiate in a pairings setting and in a Strong RSA setting respectively.

4.2 Instantiation in pairings: Protocol CEC-HT-SDH

Our instantiation of offline 2^ℓ-spendable e-cash CEC-HT in pairings has the following details. The user public key $(h_1^{x_1}, h_2^{x_2}) \in G_1 \times G_1$. Bank's public key u^γ is in G_2. The certificate (A, e) satisfies $A^{e+\gamma} h_1^{x_1} h_2^{x_2} = h_0 \in G_1$. Protocol Sign is the following signature proof-of-knowledge

$$\begin{aligned}\sigma = SPK\{(A, e, x_1, x_2, J) : A^{e+\gamma} h_1^{x_1} h_2^{x_2} = h_0 \in G_1 \\ \wedge\ 1 \le J \le 2^\ell\ \wedge\ S = \mathbf{g}_S^{1/(J'+x_2)} \in G_S\ \wedge\ x_1^{-1} = c(J+x_2)^{-1} + z \in Z_{q_1}\}(\mathsf{param}, nonce)\end{aligned} \tag{6}$$

where c is the *challenge* of the above proof system. Further detailed instantiation of the proof system is as follows: Protocol Sign randomly generates s_1, s_2, $s_{5,2}$, $s_{6,2}$, s_9, and computes the commitments

$$\begin{aligned}t_1 = A g_1^{s_1}\ \wedge\ t_2 = t_1^e h_1^{x_1} h_2^{x_2} g_2^{s_2}\ \wedge\ t_3 = \hat{\mathbf{e}}(h_0^{-1} t_2, u)\hat{\mathbf{e}}(t_1, u^\gamma)\mathbf{g}_3^{s_1+s_2} = \hat{\mathbf{e}}(g_1, u)^{s_3}\hat{\mathbf{e}}(g_1, u^\gamma)^{s_1}\mathbf{g}_3^{s_1+s_2} \\ \wedge\ t_4 = \mathbf{g}_S^{1/(J'+x_2)}\ \wedge\ t_5 = g_{5,1}^{1/(J'+x_2)} g_{5,2}^{s_{5,2}} \in G_1\ \wedge\ t_6 = t_5^{J'+x_2} g_{6,1}^{J'} g_{6,2}^{s_{6,2}} \\ \wedge\ t_7 = t_6 g_{5,1}^{-1} = g_{6,1}^{J'} g_{6,2}^{s_{6,2}} g_{5,2}^{s_7},\quad \wedge\ t_8 = d_5 = g_{5,1}^{1/x_1} g_{5,2}^{r_{5,2}}\ \wedge\ t_9 = t_8^{x_1} g_9^{s_9} \\ \wedge\ t_{10} = t_9 g_{5,1}^{-1} = g_{5,2}^{s_{10}} g_9^{s_9}\ \wedge\ 1 \le J' \le 2^\ell\ \wedge\ x^{-1} = c(J'+x_2)^{-1} + z\end{aligned} \tag{7}$$

where

$$s_3 = s_1 e,\ \ s_4 = (J' + x_2)^{-1},\ \ s_{5,1} = (J' + x_2)^{-1},\ \ s_{6,0} = x_2,\ \ s_{6,1} = J',$$
$$s_7 = s_{5,2}(J' + x_2),\ \ s_{8,1} = r_4 = x_1^{-1},\ \ s_{8,2} = r_{5,2},\ \ s_{10} = r_{5,2}x_1 = s_{8,2}x_1. \tag{8}$$

Select $s_{8,1} = r_4 = x_1^{-1}$ as per the new technique. Additional commitments are (We omit range check [7] on J' for simplicity)

$$\begin{gathered} d_1 = Z_A g_1^{r_1} \ \wedge\ d_2 = t_1^{r_e} h_1^{r_{x,1}} h_2^{r_{x,2}} h_3^{r_{x,3}} g_2^{r_2} \ \wedge\ d_3 = \hat{\mathbf{e}}(g_1, u)^{r_3} \hat{\mathbf{e}}(g_1, u^\gamma)^{r_1} \mathbf{g}_3^{r_1 + r_2} \\ \wedge\ d_4 = \mathbf{g}_S^{r_4} \ \wedge\ d_5 = g_{5,1}^{r_{5,1}} g_{5,2}^{r_{5,2}} \ \wedge\ d_6 = t_5^{r_{5,1}} g_{6,1}^{r_{6,1}} g_{6,2}^{r_{6,2}} \ \wedge\ d_7 = g_{6,1}^{r_{6,1}} g_{6,2}^{r_{6,2}} g_{5,2}^{r_7}, \\ \wedge\ d_8 = g_{5,1}^{r_{5,1}} g_{5,2}^{r_{5,2}} \ \wedge\ d_9 = t_8^{r_{x,1}} g_9^{r_9} \ \wedge\ d_{10} = g_{5,2}^{r_{10}} g_9^{r_9} \end{gathered} \tag{9}$$

The challenge is

$$c = \mathcal{H}(\mathsf{param}, nonce,\ \text{all } t_i\text{'s but } t_3, t_7, t_{10},\ \text{all } d_i\text{'s}). \tag{10}$$

where param=($\hat{\mathbf{e}}$,$\mathcal{H}$, *nonce* algorithm, all discrete logarithm bases g_i's and h_i's and $\mathbf{g}_i$'s, u,γ^u). The responses are $Z_A = R_A A^{-c}$, $z_i = r_i - cs_i$ for all index i. The signature/spending is

$$\sigma = \mathcal{H}(\mathsf{param}, nonce,\ \text{all } t_i\text{'s but } t_3, t_7, t_{10}, c, Z_A,\ \text{all } z_i\text{'s}). \tag{11}$$

Protocol $\mathsf{Vf}(\sigma)$: Upon receiving a spending σ, parses σ, computes

$$t_3 = \hat{\mathbf{e}}(h_0^{-1} t_2, u) \hat{\mathbf{e}}(t_1, u^\gamma) \mathbf{g}_3^{s_1 + s_2},\ \ t_7 = t_6 g_{5,1}^{-1},\ \ t_{10} = t_9 g_{5,1}^{-1}, \tag{12}$$
$$d_1 = Z_A g_1^{z_1} t_1^c,\ \ d_2 = t_1^{z_e} h_1^{z_{x,1}} h_2^{z_{x,2}} h_3^{z_{x,3}} g_2^{z_2} t_2^c,\ \ d_3 = \hat{\mathbf{e}}(g_1, u)^{z_3} \hat{\mathbf{e}}(g_1, u^\gamma)^{z_1} \mathbf{g}_3^{z_1 + z_2} t_3^c, \tag{13}$$
$$d_4 = \mathbf{g}_S^{z_4} t_r^c,\ \ d_5 = g_{5,1}^{z_{5,1}} g_{5,2}^{z_{5,2}} t_5^c,\ \ d_6 = t_5^{z_{5,1}} g_{6,1}^{z_{6,1}} g_{6,2}^{z_{6,2}} t_6^c,\ \ d_7 = g_{6,1}^{z_{6,1}} g_{6,2}^{z_{6,2}} g_{5,2}^{z_7} t_7^c, \tag{14}$$
$$d_8 = g_{5,1}^{z_{5,1}} g_{5,2}^{z_{5,2}} t_8^c,\ \ d_9 = t_8^{z_{x,1}} g_9^{z_9} t_9^c,\ \ d_{10} = g_{5,2}^{z_{10}} g_9^{z_9} t_{10}^c$$

Verify that the received challenge c equals to that computed from Equation (10). If OK, output 1; else output 0.

Protocol $\mathsf{Link}(\sigma, \mathsf{Archive})$: If $\mathsf{Vf}(\sigma) = 0$, output $NULL$ and abort. Else parse σ to obtain its serial S. Make a search query to $\mathsf{Archive}$ for S. If found a match σ', output $(1, \sigma')1$, else output 0. Note that $\mathsf{Archive}$ is implemented as a hash table data structure, so the search query cost a small constant number of data accesses.

Protocol $\mathsf{Trace}(\sigma, \tilde{\sigma}, \mathsf{CustomerDB})$: Verify $\mathsf{Link}(\sigma, \mathsf{Archive}) = (1, \tilde{\sigma})$. Parse σ (resp. $\tilde{sigma}$) to obtain c and z_4 (resp. $\tilde{c}$ and $\tilde{z}_4$). Compute $\hat{s} = -(\tilde{c} - c)^{-1}(\tilde{z}_4 - z_4)$ and $x_1 = z_4 + c\hat{s}$. If $t_4 \neq \mathbf{g}_S^{1/\hat{s}}$ then output $NULL$ and abort. Else output user public key $h_1^{x_1}$.

Protocol $\mathsf{RevoCoin}(\sigma, \tilde{\sigma}, \mathsf{CoinRevoList})$: Verify $\mathsf{Link}(\sigma, \mathsf{Archive}) = (1, \tilde{\sigma})$ and $\mathsf{Trace}(\sigma, \tilde{\sigma}, \mathsf{CustomerDB}) \neq NULL$. Computes as in Trace, then append these steps: For each J', $-2^\ell \leq J' \leq 2^\ell$, test $\mathbf{g}_S^{x_2} \stackrel{?}{=} \mathbf{g}_S^{\hat{s} - J'}$ until success at J^*. If success at J^* then insert the serial numbers $\mathbf{g}_S^{\hat{s} - J^* - J'}$, $1 \leq J'J \leq 2^\ell$, to $\mathsf{CoinRevoList}$.

Reductionist security proof

Theorem 1. *Let $\hat{\mathbf{e}} : G_1 \times G_1 \to G_3$ be a pairing, order(G_1) = order(G_3) = q_1, and q_U (resp. q_H) be the number of Corrupt User (resp. Random) Oracle queries by the Adversary. Assume the random oracle (RO) model and assume the Discrete Logarithm Problem is hard. Protocol CEC-HT-SDH is an offline 2^ℓ-spendable e-cash with the following security reductions:*

1. *It is* correct*;*
2. *It is* irrevocably anonymous *provided the following assumptions all hold: the DLDH(G_1,G_3) Assumption, the q_U-DDHI(G_1,G_3) Assumption, the DHTDH(G_1,G_1) Assumption.*
3. *It is* fully traceable *provided the q_U-SDH Assumption holds and $2^\ell q_U^2 / q_1$ is negligible. More specifically, if there exists an algorithm $\mathcal{A}$ which completes in time T and has advantage ϵ in Experiment FT, then the q_U-SDH Problem can be solve in time T' and probability ϵ' satisfying $T' \leq 2T$ and $\epsilon' \geq (\epsilon - 2^{\ell+1} q_U^2 q_1^{-1})^2 q_H^{-1}$.*

4. It is strongly non-frameable *provided the q_U-DHI(G_1,G_3) Assumption holds.*

In summary, Protocol CEC-HT-SDH is a secure offline 2^ℓ-spendable e-cash *provided $2^\ell q_U^2/q_1$ is negligible, and the following assumptions all hold in the RO model: the DLDH(G_1,G_3) Assumption, the q_U-DDHI(G_1,G_3) Assumption, the DHTDH(G_1,G_1) Assumption, and the q_U-SDH Assumption.*

Proof Sketch in Appendix A.

4.3 Instantiation in Strong RSA: Protocol CEC-HT-SRSA

Let $N = pq$, $p = 2p' + 1$, $q = 2q' + 1$, where p, q, p', q' are primes of similar lengths. Let G_S be a group with known order order$(G_S) > N$. Protocols Withdraw is the same as in Section 4.1. The user sk-ps pair is $((x_1, x_2), (h_1^{x_1}, h_2^{x_2}) \in QR_N^2)$. The certificate/e-coin is (A, e) satisfying $A^e h_1^{x_1} h_2^{x_2} = h_0$, e is a prime within a suitable interval, $|e - 2^{\ell_1}| < 2^{\ell_2}$, with ℓ_1 and ℓ_2 selected according to [1]. Our instantiation of the CEC-DLTDH in the strong RSA setting is:

$$SPK\{(A, e, x_1, x_2, J) : A^e h_1^{x_1} h_2^{x_2} = h_0 \in QR_N \\ \wedge\ x_1^{-1} = c(J' + x_2)^{-1} + z\ \wedge\ 1 \le J' \le 2^\ell\ \wedge\ S = \mathbf{g}_S^{1/(J'+x_2)} \in G_S\} \tag{15}$$

where c is the *challenge* of the above proof system. A further-detailed instantiation is below: The commitments are

$$t_1 = A g_1^{s_1}\ \wedge\ t_2 = t_1^e h_1^{x_1} h_2^{x_2} g_2^{s_2}\ \wedge\ t_3 = h_0^{-1} t_2 g_3^{s_1+s_2} = g_1^{s_1} g_3^{s_1+s_2} \tag{16}$$

$$\wedge\ t_4 = \mathbf{g}_S^{1/(J'+x_2)} \in G_S\ \wedge\ t_5 = g_{5,1}^{1/(J'+x_2)} g_{5,2}^{s_{5,2}}\ \wedge\ t_6 = t_5^{J'+x_2} g_{6,1}^{J'} g_{6,2}^{s_{6,2}} \tag{17}$$

$$\wedge\ t_7 = t_6 g_{5,1}^{-1} = g_{6,1}^{J'} g_{6,2}^{s_{6,2}} g_{5,2}^{s_7},\quad \wedge\ t_8 = d_5 = g_{5,1}^{1/x_1} g_{5,2}^{r_{5,2}}\ \wedge\ t_9 = t_8^{x_1} g_9^{s_9} \tag{18}$$

$$\wedge\ t_{10} = t_9 g_{5,1}^{-1} = g_{5,2}^{s_{10}} g_9^{s_9}\ \wedge\ 1 \le J' \le 2^\ell\ \wedge\ x^{-1} = c(J' + x_2)^{-1} + z \tag{19}$$

where

$$s_4 = (J' + x_2)^{-1},\quad s_{5,1} = (J' + x_2)^{-1},\quad s_{6,0} = x_2,\quad s_{6,1} = J', \tag{20}$$

$$s_7 = s_{5,2}(J' + x_2),\quad s_{8,1} = r_4 = x_1^{-1},\quad s_{8,2} = r_{5,2},\quad s_{10} = r_{5,2} x_1 = s_{8,2} x_1. \tag{21}$$

Select $s_{8,1} = r_4 = x_1^{-1}$ as per the new technique. The rest is similar to Protocol CEC-HT-SDH and is omitted. The security analysis is in the following Theorem.

Theorem 2. *Assume the random oracle (RO) model. Let N be the product of two safe primes. Let q_U (resp. q_H) be the number of Corrupt User (resp. Random) Oracle queries. Protocol CEC-DLTDH-SRSA is an offline 2^ℓ-spendable e-cash such that:*

1. *It is* correct*;*
2. *It is* irrevocably anonymous *provided the q_U-DDHI(QR_N,G_S) Assumption, the DLDH(QR_N,QR_N) Assumption, and the DHTDH(QR_N,QR_N) Assumption all hold.*
3. *It is* fully traceable *provided the Strong RSA Assumption holds and $2^\ell q_U^2 (N/4)^{-1}$ is negligible. More specifically, if there exists an algorithm $\mathcal{A}$ which completes in time T and has advantage ϵ in Experiment FT. then the Strong RSA Assumption can be solve in time T' and probability ϵ' satisfying $T' \le 2T$ and $\epsilon' \ge (\epsilon - 2^{\ell+1} q_U^2 (N/4)^{-1})^2 q_H^{-1}$.*
4. *It is* strongly non-frameable *provided the q_U-DHI(QR_N,G_S) Assumption holds.*

In summary, Protocol CEC-HT-SRSA is a secure offline 2^ℓ-spendable e-cash *provided the q_U-DDHI(QR_N,G_S) Assumption, the DLDH(QR_N,QR_N) Assumption, the DHTDH(QR_N,QR_N) Assumption, and the Strong RSA Assumption all hold in the RO model.*

Proof is in Appendix B of the full version.

4.4 New constructions of bout-wise secure offline 2^ℓ-spendable e-cash

We construct new compact e-cash satisfying all properties listed at the beginning of this paper, except it has only bout-wise irrevocable anonymity. The tradeoff is higher efficiency. We present a generic construction with two instantiations in pairings setting and in a strong RSA setting, respectively.

4.4.1 E-cash Protocol CEC-BW. Using the same parameters as in Section 4.1, our generic bout-wise scheme has the same Init, Withdraw, and the following Spend protocol:

$$\begin{aligned}\sigma = SPK\{&(A, e, x_1, x_2) : (A, e) \text{ is cert for user public key } (h_1^{x_1}, h_2^{x_2}) \\ &\wedge\ S = \mathbf{g}_J^{x_1}\ \wedge\ x_2^{-1} = c(x_1)^{-1} + z\}(\mathsf{param}, nonce)\end{aligned} \tag{22}$$

where c is the *challenge* of the above proof system. Protocol Vf verifies the above proof system. Protocol $\mathsf{Link}(\sigma, \mathsf{Archive})$ search queries the hash table Archive for a match to the serial S, denoted σ'. Protocols $\mathsf{Trace}(\sigma, \sigma', \mathsf{CustomerDB})$ and $\mathsf{CoinTrace}(\sigma, \sigma', \mathsf{CoinRevoList})$ are similar to those in Section 4.1 and omitted.

The biggest difference is now the usage count (i.e. bout) J is transmitted, i.e. included in σ. The e-cash scheme has only bout-wise security, in exchange for better efficiency by a constant factor.

4.4.2 E-cash protocol CEC-BW-SDH Using parameters from Section 4.2, we instantiate e-cash Protocol CEC-BW in a pairings setting. The core Spend Protocol is shown below. Other protocols are straightforward and omitted.

$$\begin{aligned}SPK\{&(A, e, x_1, x_2) : A^{e+\gamma} h_1^{x_1} h_2^{x_2} = h_0 \bmod N \\ &\wedge\ S = h_{1,J}^{x_1} \in G_3\ \wedge x_2^{-1} = cx_1^{-1} + z\}\end{aligned} \tag{23}$$

where c is the challenge of the proof system. Its security analysis is in the following Theorem.

Theorem 3. *Let $\hat{\mathbf{e}} : G_1 \times G_1 \to G_3$ be a pairing, order(G_1) = order$(G_3) = q_1$, and q_U (resp. q_H) be the number of Corrupt User (resp. Random) Oracle queries by the Adversary. Assume the random oracle (RO) model and assume the Discrete Logarithm Problem is hard. Protocol CEC-BW-SDH is an offline 2^ℓ-spendable e-cash with the following security reductions:*

1. *It is* correct*;*
2. *It is* bout-wise irrevocably anonymous *provided the following assumptions all hold: the DLDH(G_1,G_3) Assumption, the DHTDH(G_1,G_1) Assumption.*
3. *It is* fully traceable *provided the q_U-SDH Assumption holds and $2^\ell q_U^2/q_1$ is negligible. More specifically, if there exists an algorithm $\mathcal{A}$ which completes in time T and has advantage ϵ in Experiment FT, then the q_U-SDH Problem can be solve in time T' and probability ϵ' satisfying $T' \le 2T$ and $\epsilon' \ge (\epsilon - 2^{\ell+1} q_U^2 q_1^{-1})^2 q_H^{-1}$.*
4. *It is* strongly non-frameable *provided the co-CDH(G_1,G_3) Assumption holds.*

In summary, Protocol CEC-BW-SDH is a bout-wise secure offline 2^ℓ-spendable e-cash *provided $2^\ell q_U^2/q_1$ is negligible, and the following assumptions all hold in the RO model: the DLDH(G_1,G_3) Assumption, the DHTDH(G_1,G_1) Assumption, and the q_U-SDH Assumption.*

4.4.3 E-cash protocol CEC-BW-SRSA Using parameters from Section 4.3, we instantiate e-cash Protocol CEC-BW in a strong RSA setting: The core Spend Protocol is shown below. Other protocols are straightforward and omitted.

$$\begin{aligned}SPK\{&(A, e, x_1, x_2) : A^e h_1^{x_1} h_2^{x_2} = h_0 \in QR_N \\ &\wedge\ x_1^{-1} = c(J' + x_2)^{-1} + z\ \wedge\ S = \mathbf{g}_J^{1/(J'+x_2)} \in G_S\}\end{aligned} \tag{24}$$

where c is the *challenge* of the above proof system. Its security analysis is in the following Theorem.

Theorem 4. *Assume the random oracle (RO) model. Let N be the product of two safe primes. Let q_U (resp. q_H) be the number of Corrupt User (resp. Random) Oracle queries. Protocol CEC-BW-SRSA is an offline 2^ℓ-spendable e-cash such that:*

1. *It is* correct*;*
2. *It is* bout-wise irrevocably anonymous *provided the DLDH(QR_N, QR_N) Assumption, and the DHTDH(QR_N, QR_N) Assumption all hold.*
3. *It is* fully traceable *provided the Strong RSA Assumption holds and $2^\ell q_U^2 (N/4)^{-1}$ is negligible. More specifically, if there exists an algorithm $\mathcal{A}$ which completes in time T and has advantage ϵ in Experiment FT. then the Strong RSA Assumption can be solve in time T' and probability ϵ' satisfying $T' \leq 2T$ and $\epsilon' \geq (\epsilon - 2^{\ell+1} q_U^2 (N/4)^{-1})^2 q_H^{-1}$.*
4. *It is* strongly non-frameable *provided the co-CDH(QR_N, G_S) Assumption holds.*

In summary, Protocol CEC-BW-SRSA is a bout-wise secure offline 2^ℓ-spendable e-cash *provided the DLDH(QR_N, QR_N) Assumption, the DHTDH(QR_N, QR_N) Assumption, and the Strong RSA Assumption all hold in the RO model.*

5 Discussions, Applications, and Conclusions

Instantiating [12]'s System One to a pairings setting

Using parameters and Init, Withdraw protocols from 4.2, we instantiate [12]'s System One to a pairings setting. The original paper only instantiated it to a strong RSA framework. The core Spend protocol is presented below. Other protocols are straightforward and omitted.

$$\text{CEC-CHL-SDH-Spend}: \quad SPK\{(A, e, x_1, x_2, J) : 1 \leq J \leq 2^\ell$$
$$A^{e+\gamma} h_1^{x_1} h_2^{x_2} h_3^{x_3} = h_0 \in G_1 \ \wedge \ S = \mathbf{g}_S^{1/(J'+x_2)} \ \wedge \ T = \hat{\mathbf{e}}(g_1^{x_1}, g_1) \mathbf{g}_T^{R/(J'+x_3)} \ \wedge \ 1 \leq J \leq 2^\ell\} \qquad (25)$$

Theorem 5. *Let $\hat{\mathbf{e}} : G_1 \times G_1 \rightarrow G_3$ be a pairing, order(G_1) = order(G_3) = q_1, and q_U (resp. q_H) be the number of Corrupt User (resp. Random) Oracle queries by the Adversary. Assume the random oracle (RO) model and assume the Discrete Logarithm Problem is hard. Protocol CEC-CHL-SDH is an offline 2^ℓ-spendable e-cash with the following security reductions:*

1. *It is* correct*;*
2. *It is* irrevocably anonymous *provided the following assumptions all hold: the DLDH(G_1,G_3) Assumption, the q_U-DDHI(G_1,G_3) Assumption.*
3. *It is* fully traceable *provided the q_U-SDH Assumption holds and $2^\ell q_U^2 / q_1$ is negligible. More specifically, if there exists an algorithm $\mathcal{A}$ which completes in time T and has advantage ϵ in Experiment FT, then the q_U-SDH Problem can be solve in time T' and probability ϵ' satisfying $T' \leq 2T$ and $\epsilon' \geq (\epsilon - 2^{\ell+1} q_U^2 q_1^{-1})^2 q_H^{-1}$.*
4. *It is* strongly non-frameable *provided the q_U-DHI(G_1,G_3) Assumption holds.*

In summary, Protocol CEC-CHL-SDH is a secure offline 2^ℓ-spendable e-cash *provided $2^\ell q_U^2 / q_1$ is negligible, and the following assumptions all hold in the RO model: the DLDH(G_1,G_3) Assumption, the q_U-DDHI(G_1,G_3) Assumption, and the q_U-SDH Assumption.*

Note e-cash scheme CEC-CHL-SDH does not have an efficient CoinTrace protocol, just like [12]'s System One it instantiates.

Efficiency discussions In comparison to [12]'s System One, our e-cash scheme CEC-HT-SRSA (resp. CEC-BW-SRSA) has a similar complexity, CEC-HT-SDH (resp. CEC-BW-SRSA, CEC-CHL-SDH) has a higher complexity in Vf's online computation of two pairings. Schemes CEC-HT-SRSA and CEC-HT-SDH solve [12]'s open problem of simultaneous achievement of efficient coin tracing and compact wallet size. Schemes CEC-BW-SDH and CEC-BW-SRSA also do so, but with a slightly weaker anonymity: the bout-wise irrevocable anonymity.

Conclusion: We solve SDH solve [12]'s open problem in compact e-cash of simultaneous achievement of efficient coin tracing and compact wallet size. Our main tool is a new technique in zero-knolwedge proofs which extracts user secret keys when over-spending occurs. The complexity of our scheme is similar to the more efficient System One of [12]. The security of our compact e-cash schemes is reduced to a new intractability assumption.

References

1. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO 2000*, pages 255–270. Springer-Verlag, 2000.
2. G. Ateniese, K. Fu, M Green, and S. Hohenberger. Improved proxy-re-encryption schemes with applications to secure distributed storage. In *12th Annual NDSS*, pages 29–43, 2005. Full version available at http://eprint.iacr.org/2005/028.
3. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: formal definitions, simplified requirements and a construction based on general assumptions. In *EUROCRYPT'03*, volume 2656 of *LNCS*. Springer-Verlag, 2003.
4. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA 2005*, 2005. To appear.
5. D. Boneh and X. Boyen. Short signatures without random oracles. In *Eurocrypt 2004*, volume 3027 of *LNCS*, pages 56–73. Springer–Verglag, 2004.
6. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPYTO 2004*, volume 3152 of *LNCS*. Springer–Verglag, 2004.
7. F. Boudot. Efficient proofs that a committed number lies in an interval. In *Eurocrypt'00*, pages 431–444, 2000.
8. S. Brands. An efficient off-line electronic cash system based on the representation problem. Technical Report CS-R9323, CWI, CWI, April 1993.
9. S. Brands. Untraceable off-line cash in wallet with observers. In *CRYPTO'93*, pages 302–318. Springer-Verlag, 1993.
10. S. Brands. Untraceable off-line cash in wallets with observers. manuscript, CWI, 1993.
11. J. Camenisch and I. Damgård. Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In *Proc. ASIACRYPT 2000*, pages 331–345. Springer-Verlag, 2000. LNCS No. 1976.
12. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 302–321. Springer-Verlag, 2005. Also http://eprint.iacr.org/2005/060.
13. J. Camenisch and A. Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer-Verlag, 2001.
14. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *CRYPTO 2002*, volume 2442 of *LNCS*, pages 61–76. Springer-Verlag, 2002.
15. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *SCN 2002*, volume 2576 of *LNCS*, pages 268–289. Springer-Verlag, 2003.
16. M. Gaud Canard and J. Traore. On the anonymity of fair offline e-cash systems. In *FC 2003*, volume 2742 of *LNCS*, pages 34–50. Springer-Verlag, 2003.
17. S. Canard and J. Traore. On fair e-cash systems based on group signature schemes. In *ACISP 2003*, volume 2727 of *LNCS*, pages 237–248. Springer-Verlag, 2003.
18. R. Canetti. Universal composable security: a new paradigm for cryptographic protocols. In *FOCS 2001*, pages 136–145. IEEE Computer Society, 2001.
19. R. Canetti, R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Adaptive security for threshold cryptosystems. In *CRYPTO 1999*, volume 1666 of *LNCS*, pages 98–115. Springer-Verlag, 1999.
20. A. Chan, Y. Frankel, and Y. Tsiounis. Easy come - easy go divisible cash. In *EUROCRYPT '98*, volume 1403 of *LNCS*, pages 561–575. Springer-Verlag, 1998.
21. D. Chaum. Blind signatures for untraceable payments. In *Crypto'82*, pages 199–203. Plenum Press, 1982.
22. D. Chaum. Blind signature systems. In *CRYPTO '83*, LNCS. Springer-Verlag, 1983.
23. D. Chaum. Online cash checks. In *EUROCRYPT '89*, volume 434 of *LNCS*, pages 288–293. Springer-Verlag, 1989.
24. D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *Proceedings on Advances in cryptology*, pages 319–327. Springer-Verlag, 1990.
25. D. Chaum and T. P. Pedersen. Transferred cash grows in size. In *EUROCRYPT '92*, volume 658 of *LNCS*, pages 390–407. Springer-Verlag, 1992.
26. D. Chaum and T. P. Pedersen. Wallet databases with observers. In *EUROCRYPT '92*, volume 658 of *LNCS*, pages 89–105. Springer-Verlag, 1992.
27. D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT'91*, volume 547 of *LNCS*, pages 257–265. Springer-Verlag, 1991.
28. A. die Solages and J. Traore. An efficient fair off-line electronic cash system with extensions to checks and wallets with observers. In *FC '98*, volume 1465 of *LNCS*, pages 275–295. Springer-Verlag, 1998.

29. Y. Dodis and A. Yampolskiy. A verifiable random function with short proofs and keys. In *PKC 2005*, pages 416–431. Springer-Verlag, 2005. Lecture Notes in Computer Science No. 3386.
30. N. Ferguson. Extensions of single-term coins. In *CRYPTO'93*, pages 292–301. Springer-Verlag, 1993.
31. N. Ferguson. Single term off-line coins. In *Eurocrypt'93*, pages 318–328. Springer-Verlag, 1993.
32. M. Franklin and M. Yung. Towards provably secure efficient electronic cash. In *ICALP '93*, volume 700 of *LNCS*, pages 265–276. Springer-Verlag, 1993.
33. S. Jarecki and V. Shmatikov. Handcuffing Big Brother: an abuse-resilient transaction escrow scheme. In *Eurocrypt 2004*, volume 3027 of *LNCS*, pages 590–608. Springer-Verlag, 2004.
34. A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *Eurocrypt 2004*, LNCS, pages 571–589. Springer-Verlag, 2004.
35. A. Kiayias and M. Yung. Extracting group signatures from traitor tracing schemes. In *Eurocrypt 2003*, LNCS, pages 630–648. Springer-Verlag, 2003.
36. A. Kiayias and M. Yung. Group signatures: provable security, efficient constructions, and anonymity from trapdoor-holders. Cryptology ePrint Archive, Report 2004/076, 2004. http://eprint.iacr.org/.
37. A. Lysyanskaya and Z. Ramzan. Group blind digital signatures: A scalable solution to electronic cash. In *FC'98*, pages 184–197. Springer-Verlag, 1998.
38. G. Maitland and C. Boyd. Fair electronic cash based on a group signature scheme. In *ICICS'01*, volume 2229 of *LNCS*. Springer-Verlag, 2001.
39. G. Maitland, J. Reid, E. Foo, C. Boyd, and E. Dawson. Linkability in practical electronic cash design. In *ISW 2000*, volume 1975 of *LNCS*, pages 149–163. Springer-Verlag, 2000.
40. G. Medvinsky and C. Neuman. Netcash: a design for practical electronic currency on the internet. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 102–106. ACM Press, 1993.
41. L. Nguyen and R. Safavi-Naini. Efficient and provably secure trapdoor-free group signature schems from bilinear pairings. In *ASIACRYPT 2004*, pages 372–386, 2004. Lecture Notes in Computer Science No. 3329.
42. T. Okamoto. An efficient divisible electronic cash scheme. In *CRYPTO'95*, pages 438–451. Springer-Verlag, 1995.
43. T. Okamoto and K. Ohta. Disposable zero-knowledge authentications and their applications to untraceable electronic cash. In *CRYPTO '89*, volume 435 of *LNCS*, pages 481–496. Springer-Verlag, 1989.
44. I. Teranishi, J. Furukawa, and K. Sako. k-times anonymous authentication. In *Asiacrypt 2004*, volume 3329 of *LNCS*, pages 308–322. Springer-Verlag, 2004.
45. J. Traoré. Group signatures and their relevance to privacy-protecting off-line electronic cash systems. In *ACISP'99*, pages 228–243. Springer-Verlag, 1999.
46. Y. S. Tsiounis. *Efficient Electronic Cash: New Notions and Techniques*. PhD dissertation, Northeastern University, 1997.
47. Victor K. Wei. Tracing-by-linking group signatures. Cryptology ePrint Archive, Report 2004/370, 2004. http://eprint.iacr.org/.

A Proof Sketch of Theorem 1

Intuition of the proof: Our scheme can be viewed as modified from [6]'s group signatures by deleting Open Authority implementation and then adding more relations. Therefore, our full traceability and non-frameability tend to be enhanced from those of [6], and our irrevocable anonymity faced more pressure than the anonymity of [6]. Note the probability that $J' + x_2 = \bar{J}' + \bar{x}_2$ must be figured in, where x_2 and $\bar{x}_2$ belong to two users corrupted by Adversary. This scenario attacks the *validity* of Protocol Link where $\mathsf{Link}(\sigma, \mathsf{Archive}) = (1, \sigma')$ but $\mathsf{Trace}(\sigma, \sigma', \mathsf{CustomerDB})$ outputs $NULL$. This attack probability is upper bounded by $2^{\ell+1} q_U^2 / q_1$.

A.1 Proof sketch of full traceability

Intuition: Compared to the attacker to the group signature of Boneh, et al. [6], our attacker $\mathcal{A}$ must do more. [6]'s full traceability attacker must solve a q_U-SDH Problem instance. Our attacker $\mathcal{A}$ must do at least that much. We sketch the two most crucial parts of the proof only: How to simulating the oracles and how to set up and extract the witness.

In order to win Experiment FT, the Adversary $\mathcal{A}$ must be able to forge all three parts:

1. The proof-of-knowledge of certificate (A, e) which certifies the user public key $(h_1^{x_1}, h_2^{x_2})$ and proof-of-knowledge of its corresponding user secret key (x_1, x_2); These involve t_1, t_2, t_3.
2. t_4.
3. The remaining t_i's.

Below, we sketch a proof which shows $\mathcal{A}$'s forgery of Item 1 above reduces to the Theorem assumption on full traceability, even if it is assisted by oracles to forge the other two Items. Note Simulator $\mathcal{S}$ enjoys the same Assisting Oracle. The Assisting Oracles can, upon input J and $h_2^{x_2}$ (resp. $h_1^{x_1}$), output $\mathbf{g}_S^{1/(J'+x_2)}$ and $g_{5,1}^{1/(J'+x_2)}$ (resp. $g_{5,1}^{1/x_1}$). The Assisting Oracles can also compute some limited number of CDH queries, to compute $t_5^{x_2}$, $t_8^{x_1}$.

Simulating the attack oracles. We sketch how to simulate the toughest attack oracle to simulate, $\mathcal{SO}$. The simulation of other oracles is relatively easy. $\mathcal{SO}$ queries not involving users who are ever corrupted by $\mathcal{CUO}$ are simulated by the method of special HVZK (honest-verifier zero-knowledge) simulation, with a twist to accommodate the relation $x_1^{-1} = c(J' + x_2)^{-1} + z$. $\mathcal{SO}$ queries involving any of the q_U corrupted users are simulated by the typical method such as in [6] to simulate for the q_U-SDH Assumption. Details below.

Denote U's public key as $(h_1^{x_1}, h_2^{x_2})$, and $\mathsf{coin} = (A, e)$. Simulator computes the oracle query $\mathcal{SO}(U, \mathsf{coin})$, where U was never queried to $\mathcal{CUO}$, as follows:

1. Randomly generate the challenge c. In simulating special HVZK proofs, c is generated first, and backpatched last by the random oracle.
2. Randomly generate s_1, s_2, $s_{5,2}$, $s_{6,2}$, s_9, J, $1 \le J \le 2^\ell$, J has not been simulated for this coin before. Compute the other s_i's by Equation (8).
3. Compute all commitments t_i's in Equation (7), using Assisting Oracle if necessary. Except $t_8 = d_5$ will be computed later.
4. Randomly generate all responses z_i's.
5. Compute all d_i's according to Equation (9). Except d_8 because the needed t_8 is not ready.
6. Now set $t_8 = d_5$. Compute d_8 according to Equation (9). Backpatch c to Equation (10).

To simulate $\mathcal{SO}(U, \mathsf{coin})$ queries where U has been queried to $\mathcal{CUO}$, $\mathcal{S}$ computes as follows:

1. $\mathcal{S}$ received a q_U-SDH Problem instance to solve: $g^{\gamma^i} \in G_1$, $0 \le i \le q_U$, compute $(g^{1/(\gamma+\bar{e})}, \bar{e})$.
2. Randomly generate e_i, $x_1^{(i)}$, $x_2^{(i)}$, $1 \le i \le q_U$. Randomly generate $\alpha_{\prime h', i}$ and backpatch the random oracle to $h_i = \mathcal{H}('h', i) = h_0^{\alpha_{\prime h', i}}$. Note $\alpha_{\prime h', 0} = 1$.
3. Let $f(\gamma) = \prod_{1 \le i \le q_U} (\gamma + e_i)$. Backpatch to $h_0 = g^{f(\gamma)}$. For $1 \le i \le q_U$, let $A_i = g^{f(\gamma)(\gamma+e_i)^{-1}(1-\alpha_{h,1}x_1-\alpha_{h,2}x_2)^{-1}}$; note $A_i^{e_i+\gamma} h_1^{x_1^{(i)}} h_2^{x_2^{(2)}} = h_0$.
4. $\mathcal{S}$ outputs certificate/coin (A_i, e_i) and user secret key (x_1, x_2) upon query $\mathcal{CUO}(U_i)$.

5. $\mathcal{S}$ simulates the query $\mathcal{SO}(U_i, \mathsf{coin})$ for corrupted user U_i by knowing user secret keys.

Witness Extraction: Assume $\mathcal{A}$ wins Experiment FT. $\mathcal{S}$ sets up as above in order to solve a given q_U-SDH Porblem instance as follows: Rewind $\mathcal{A}$ to extract secrets $\hat{A}$, $\bar{e}$, $\hat{x}_1$, $\hat{x}_2$, $hats_1$, $\hat{s}_2$, and $\hat{s}_3$ satisfying

$$t_1 = \hat{A} g_1^{\hat{s}_1}, \tag{26}$$
$$t_2 = t_1^{\bar{e}} h_1^{\hat{x}_1} h_2^{\hat{x}_2} g_2^{\hat{s}_2}, \tag{27}$$
$$t_3 = \hat{\mathbf{e}}(h_0^{-1} t_2, u) \hat{\mathbf{e}}(t_1, y^{\gamma}) \mathbf{g}_3^{\hat{s}_1 + \hat{s}_2} \tag{28}$$
$$= \hat{\mathbf{e}}(g_1, u)^{\hat{s}_3} \hat{\mathbf{e}}(g1, u^{\gamma})^{\hat{s}_1} \mathbf{g}_3^{\hat{s}_1 + \hat{s}_2} \tag{29}$$

and then we have

$$\hat{A}^{\bar{e}+\gamma} h_1^{\hat{x}_1} h_2^{\hat{x}_2} h_0^{-1} = g_1^{\delta'} \tag{30}$$

where $\delta' = \hat{s}_3 - \hat{s}_1 \bar{e}$. Backpatch the random oracle to $g_1 = h_0^{\alpha_{g,1}}$. Let $\tilde{A} = \hat{A}^{\delta}$, $\tilde{x}_1 = \hat{x}_1 \delta$, $\tilde{x}_2 = \hat{x}_2 \delta$, where $\delta = (\delta' \alpha_{g,1})^{-1}$, then we have

$$\tilde{A} = h_0^{(1 - \alpha_{h,1} \tilde{x}_1 - \alpha_{h,2} \tilde{x}_2)/(\gamma + \bar{e})} \tag{31}$$
$$= g^{f(\gamma)(1 - \alpha_{h,1} \tilde{x}_1 - \alpha_{h,2} \tilde{x}_2)/(\gamma + \bar{e})} \tag{32}$$
$$= g^{\sum \bar{f}_i \gamma^i} g^{\bar{f}_{-1}/(\bar{e}+\gamma)} \tag{33}$$

Then compute $g^{1/(\bar{e}+\gamma)}$ which, together with $\bar{e}$, solved the q_U-SDH Problem instance. □

A.2 Proof sketch of strong non-frameability

We prove for Trace only. The other case, CoinTrace is similar and omitted. Assume Adversary $\mathcal{A}$ has a non-negligible advantage in Experiment SNF. Among other things, $\mathcal{A}$ can compute $t_4 = \mathbf{g}_S^{1/(J'+x_2)}$ from given $h_2^{x_2}$, after corrupting q_U other users and corrupting Bank.

Given a q_U-DHI Problem instance: g^{γ^i}, $0 \le i \le q_U$, $\mathcal{S}$ computes as in the above proof of full traceability as follows: Randomly generate e_i, $x_1^{(i)}$, $x_2^{(i)}$, $1 \le i \le q_U$ and use these to answer the q_U queries to the Corrupt User Oracle $\mathcal{CUO}$. Eventually $\mathcal{A}$ delivers $t_4 = \mathbf{g}_S^{1/(J'+x_2)}$ from an uncorrupted user public key $h_2^{x_2}$. which solves the q_U-DHI Problem. □

A.3 Proof sketch of irrevocable anonymity

Intuitions on Irrevocable Anonymity: $\mathcal{A}$ wins Experiment IA if it distinguishes via any of the t-type commitments. Distinguishing via t_1, t_2, t_3 is stopped by the DLDH(G_1,G_3) Assumption. The proof is very similar to [6] which pioneered the DLDH(G_1, G_1) Assumption, called LDA (Linear Decisional Assumption) in their paper. Our first three t-type commitments are essentially the same as their t-type commitments, except two things:

1. We omitted to prove the relation $s_3 = es_1$. Therefore we omitted some t-type commitments from [6]. The consequence should be enhanced anonymity and pressured soundness than [6]. Indeed we had to use more proof techniques in our full traceability (concurrent soundness) in the previous subsections.
2. We modified [6] so the reduction is to DLDH(G_1,G_3) instead of to DLDH(G_1,G_1). The former assumption is generally believed to be at least at good as the latter assumption.

Distinguishing via S is stopped by the q_U-DDHI(G_1,G_3) Assumption. The proof is essentially identical to that in [12], which used exactly the same design of serial S. In [12], the user public key $h_2^{x_2}$, and the serial $S = \mathbf{g}_S^{1/(J'+x_2)}$ are in the same group QR_N where N is the product of two safe primes. The anonymity of their serial reduces to the q_U-DDHI(QR_N,QR_N) Assumption. In our case, the user public key $h_2^{x_2}$ is in G_1 and the serial S is in G_3. Our anonymity, in this part, reduces to the q_U-DDHI(G_1, G_3) Assumption.

To distinguish via the special relation $x_1^{-1} = c(J' + x_1)^{-1} + z$: With the secretes s_i's in the remaining t-type commitments t_5 through t_{10} generated anew each spending, the only real threat to anonymity is to use the special relation and distinguish user public keys $(h_1^{x_1^{(b)}}, h_2^{x_2^{(b)}})$, $b = 1, 2$. But this is stopped by the DHTDH Assumption. □