Fair Off-Line e-Cash Made Easy

Yair Frankel* Yiannis Tsiounis** Moti Yung***

Abstract. Anonymous off-line electronic cash (e-cash) systems provide transactions that retain the anonymity of the payer, similar to physical cash exchanges, without requiring the issuing bank to be on-line at payment. Fair off-line e-cash extend this capability to allow a qualified third party (a "trustee") to revoke this anonymity under a warrant or other specified "suspicious" activity. Extensions for achieving fair off-line e-cash based on off-line e-cash require modularity to be applicable in general settings. Simplicity (for ease of understanding and implementation) and efficiency (for cost effectiveness) are of high importance, otherwise these generic extensions will be hard and costly to apply. Of course, security must also be guaranteed and understood, yet, to date, there have been no efficient systems that offer provable security.

A system which is (1) provably secure based on well understood assumptions, (2) efficient and (3) conceptually easy, is typically "elegant." In this work we make a step towards elegant fair off-line e-cash system by proposing a system which is provably anonymous (i.e., secure for legitimate users) while its design is simple and its efficiency is similar to the most efficient systems to date. Security for the bank and shops is unchanged from the security of *non*-traceable e-cash. We also present ways to adapt the functionality of "fairness" into existing e-cash systems in a modular way, thus easing advancement and maintaining version compatibility; these extensions are also provably anonymous.

Keywords: Electronic cash, anonymity revocation, decision Diffie-Hellman.

1 Introduction

Simplicity is the crux of system design; when it comes to secure systems it is even more important for two reasons: first, it limits the possibility of errors during design and implementation and eases the proof of security; second, it potentially allows the algorithms to run on reduced computational resources.

In this work we simplify the method of achieving fair off-line e-cash based on any (single-term) off-line e-cash system (we demonstrate functionality under [Bra93b]). We do so without affecting the security of the basic system, while we prove the security (i.e., anonymity of legitimate users) of the "fairness" extension using a better understood assumption, that of the decision Diffie-Hellman. Our goal is to move a step closer to "elegant" fair e-cash, i.e., minimize number of added requirements, security assumptions and overhead while extending the e-cash systems into fair ones. We utilize the recent result [TY98,NR97] showing equivalence of the semantic security (namely, security in the sense of indistinguishability) of ElGamal encryption and the decision Diffie-Hellman assumption.

The model: Fair off-line electronic cash (FOLC), independently introduced by [FTY96] and [CMS96], extends anonymous off-line electronic cash and involves a bank ($\mathcal{B}$), a collection of users (a single user is called $\mathcal{U}$), a collection of receivers/shops (a single receiver is denoted by $\mathcal{R}$), and a collection of Trustees (judges/escrow agents) which act like one party[1] (and

* CertCo, NY, NY. e-mail: frankely@certco.com
** GTE Laboratories, Inc., Waltham, MA. e-mail: ytsiounis@gte.com
*** CertCo, NY, NY. e-mail: moti@certco.com, moti@cs.columbia.edu

[1] It is outside the scope of this paper to show how the power of the Trustees can be equally distributed. $\mathcal{T}$ should be envisioned as being a single trusted entity.

K. Ohta and D. Pei (Eds.): ASIACRYPT'98, LNCS 1514, pp. 257–270, 1998.

are denoted as $\mathcal{T}$). FOLC includes five basic protocols, three of which are the same as in off-line electronic cash: a *withdrawal protocol* with which $\mathcal{U}$ withdraws electronic coins from $\mathcal{B}$ while his account is debited, a *payment protocol* with which $\mathcal{U}$ pays the coin to $\mathcal{R}$, and a *deposit protocol* with which $\mathcal{R}$ deposits the coin to $\mathcal{B}$ and has his account credited.

The two additional protocols are: *owner tracing* in which $\mathcal{B}$ gives to $\mathcal{T}$ the view of a deposit protocol and $\mathcal{T}$ returns a string that contains the identifying information of the coin's owner (which $\mathcal{B}$ can use to identify the owner via its account databases); and *coin tracing* in which $\mathcal{T}$, given the view of a withdrawal protocol from $\mathcal{B}$, returns some information that originated from this withdrawal. $\mathcal{B}$ can use the returned value to find the coin(s) by accessing its views of the deposit protocols. Hence, owner tracing allows tracing of suspicious payments, while coin tracing allows the authorities to find the destination of suspicious withdrawals. We do not consider the strong bank robbery attacks [JY96].

Previous work: [FTY96] introduced the notion of "indirect discourse proofs" and used it to implement FOLC; however, payments had to be interactive, while security required novel assumptions. Here we implement non-interactive indirect discourse proofs, while our complete solution is more secure and as efficient as owner tracing alone on that system. [FTY96] also proved that anonymity in FOLC cannot be unconditional. [CMS96] introduced efficient owner and coin tracing protocols; coin tracing in particular was much faster than [FTY96]. However, security was not analyzed, while owner tracing was performed against the database of withdrawn coins, i.e., $\mathcal{T}$ returns to $\mathcal{B}$ a value appearing in a withdrawal transcript instead of the user's identity; this reduces the computational requirements at withdrawal and payment (i.e., "enforcement" of owner tracing capability), but requires more time for owner tracing. Here we perform owner tracing against the account database (i.e., we escrow the "users' identities") while retaining the efficiency of [CMS96]. [DFTY97] simplified the protocols of [FTY96] using faster coin tracing techniques, on par with [CMS96]; security however still required novel assumptions while payments were again interactive. [dST98] recently presented efficient protocols (for account-based owner tracing) with non-interactive payments, but their anonymity depends on more complex assumptions (these are not specified in a strict sense, but our evaluation shows that the main assumption is a variant of the decision Diffie-Hellman assumption, similar to the "matching Diffie-Hellman" introduced in [FTY96]). Our efficiency is on par with this system but we can concretely prove anonymity under the decision Diffie-Hellman assumption.

Security assumptions: Security of the basic off-line e-cash scheme is based on the blind signature protocol that we use as an underlying block; in the case we demonstrate here, this is the same as in [Bra93b] but other protocols can be used. All such protocols are based on the random oracle model and although the unforgeability of the resulting signatures is provable [PS96a] their restrictive properties are still unproven. We prove the anonymity of our system based on the decision Diffie-Hellman assumption.

Structure of the paper: In section 2 we present the ElGamal encryption scheme and the decision Diffie-Hellman assumption, as well as some known impossibility results for FOLC. In section 3 we present the building blocks for our protocols, namely a blind signature scheme with some restrictive properties, proofs of equality of logarithms and non-interactive indirect discourse proofs. In section 4 we show how FOLC can be added in a modular way in existing systems, while in section 5 we show how to achieve FOLC efficiently and securely if we have more freedom in the system design phase. We discuss the security in section 6 and we conclude with open problems in section 7.

2 Preliminaries

In [FTY96] it was shown what are the cryptographic assumptions needed for FOLC as summarized in the following Theorems.

Theorem 1 *(1) Unconditional unlinkability is impossible in FOLC even if only owner tracing or coin tracing is supported. (2) Further, any implementation of FOLC based on black box reduction from an arbitrary one-way permutation will separate P and NP (thus, it seems implausible, since it will yield a breakthrough in complexity theoretic proof techniques).*

Theorem 2 *Given off-line e-cash and public-key encryption, there exists a FOLC system in which anonymity is semantically secure (in the sense of secure encryption [GM84]).*

A semantically secure encryption which has homomorphic properties is the ElGamal encryption scheme [ElG85]:

Definition 1. (ElGamal public-key encryption scheme) *The* ElGamal public-key encryption *scheme is defined by a triplet* (G, E, D) *of probabilistic polynomial-time algorithms, with the following properties:*

- *The* system setup algorithm, S, *on input* 1^n, *where* n *is the security parameter, outputs the system parameters* (p, q, g), *where* (p, q, g) *is an instance of the DLP collection, i.e.,* p *is a uniformly chosen prime of length* $|p| = n + \delta$ *for a specified constant* δ, *and* g *is a uniformly chosen generator of the subgroup* G_q *of prime order* q *of* Z_p^*, *where* $q = (p-1)/\gamma$ *is prime and* γ *is a specified integer.*
- *The* key generating algorithm, G, *on input* (p, q, g), *outputs a public key,* $e = (p, q, g, y)$, *and a private key,* $d = (p, q, g, x)$, *where*
  - x *is a uniformly chosen element of* Z_q, *and*
  - $y \equiv g^x \bmod p$.
- *The* encryption algorithm, E, *on input* (p, q, g, y) *and a message* $m \in G_q$, *uniformly selects an element* k *in* Z_q *and outputs*

$$E((p, q, g, y), m) = (g^k \pmod p, my^k \pmod p) \ .$$

- *The* decryption algorithm, D, *on input* (p, q, g, x) *and a ciphertext* (y_1, y_2), *outputs*

$$D((p, g, x), (y_1, y_2)) = y_2 {y_1}^{x^{-1}} \pmod p \ .$$

For simplicity we write $E(m) = (g^k, my^k)$ for public key y.

Definition 2. (Decision Diffie-Hellman problem) *For security parameter* n, p *a prime with* $|p-1| = \delta + n$ *for a specified constant* δ, *for* $g \in Z_p^*$ *a generator of prime order* $q = (p-1)/\gamma$ *for a specified integer* γ *and for* $a, b \in_R Z_q$ *random, given* $[g^a, g^b, y]$ *output* 0 *if* $y \equiv g^{ab} \pmod p$ *and* 1 *otherwise, with probability better than* $1/2 + 1/n^c$ *for any constant* c *for large enough* n.

The decision Diffie-Hellman assumption states that it is infeasible to solve the decision Diffie-Hellman problem. In [TY98] a proof of the following is presented:

Theorem 3 *The ElGamal encryption scheme is semantically secure, if and only if there does not exist a p.p.t. TM that solves the decision Diffie-Hellman problem.*

We remark here that theorem 3 is true even for a modified **"inverted" ElGamal encryption,** i.e., when $E(m) = (y^k, mg^k)$ with y the public key.

3 Building Blocks

All off-line electronic cash schemes to date utilize a blinding protocol that allows the bank to verify that users embed their identity in the coin. In turn, all fair off-line e-cash schemes, employ a protocol for proving relations between committed values. We devote one subsection to each concept. In addition we show an implementation of "indirect discourse proofs" [FTY96,DFTY97] based on proofs of equality of logarithms.

3.1 The Blinding Protocol

There are several blind signature protocols in the literature which allow the signer to verify that some values are correctly embedded by the requester. The first was proposed by [CFN90] but here we will use protocols that avoid the costly (in terms of both speed and storage) "cut-and-choose" technique, such as the withdrawal protocols in [CP93a,BCC$^+$92,CP93b], the"restrictive blinding" in [Bra93b], the protocol "P" in [CMS96], or the "blind signature" protocol in [dST98]. Here we will demonstrate one particular such protocol, Brands' "restrictive blind signature," but it should be noted that the ideas presented are applicable to any of the other sub-protocols used as building blocks.

We now describe the blinding protocol in [Bra93b], between a signer $\mathcal{S}$ and a verifier $\mathcal{V}$.

Setup:

Let p and q be primes such that $|p-1| = \delta + k$ for a specified constant δ, and $p = \gamma q + 1$, for a specified integer γ. Define a unique subgroup G_q of prime order q of the multiplicative group Z_p^* and generators g, g_1, g_2 of G_q. Let $\mathcal{H}, \mathcal{H}_0, \mathcal{H}_1, \ldots$ be hash functions from a family of collision intractable hash functions.

Let $X_{\mathcal{S}} \in_R Z_q$ be the secret key of the signer. The signer publishes its public keys $h = g^{X_{\mathcal{S}}}$, $h_1 = g_1^{X_{\mathcal{S}}}, h_2 = g_2^{X_{\mathcal{S}}}$.

Let $u_1 \in G_q$ be the verifier's private key and $I = {g_1}^{u_1}$ his public identification information (knowledge of private keys should be verified as pointed out in [CFMT96], using e.g., a Schnorr proof of knowledge [Sch91]).

The protocol creates a blind signature of I. $\mathcal{V}$ will end up with a Schnorr-type [Sch91] signature on $(Ig_2)^s$, where s is a random number (chosen by $\mathcal{V}$ and kept secret). The exact form of the signature is $sig(A, B) = (z, a, b, r)$ satisfying:

$$g^r = h^{\mathcal{H}(A,B,z,a,b)} a \quad \text{and} \quad A^r = z^{\mathcal{H}(A,B,z,a,b)} b \tag{1}$$

The **blinding protocol** (over an authenticated channel between $\mathcal{V}$ and $\mathcal{S}$) appears in figure 1.

This protocol produces a signed number A of the form ${I^s}{g_2}^s$, i.e., A is an unconditionally-hiding commitment of the verifier's identity I. There are no complete security proofs for such protocols, but they are used in every efficient e-cash scheme and have received continuous scrutiny in recent years. There do however exist security arguments under the random oracle model [PS96a] for the existential unforgeability of such signature schemes—but not for their "restrictive" properties (i.e., we cannot yet prove that A is a correct commitment on I).

3.2 Proving Equality of Logarithms

A basic tool for both owner and coin tracing is an efficient blind proof of equality of logarithms. Such proofs are used for FOLC either in isolation, or as a block in constructing non-interactive indirect discourse proofs, which can then provide some of the functionality needed for FOLC.

$\mathcal{S}$ blindly signs A, such that A embeds the identity I of $\mathcal{V}$

$\mathcal{V}$		$\mathcal{S}$
		$w \in_R Z_q$
$s \in_R Z_q$	$\xleftarrow{a',b'}$	$a' = g^w, b' = (Ig_2)^w$
$A = (Ig_2)^s$		
$z' = h_1^{u_1} h_2 [= (Ig_2)^{X_S}]$		
$z = z'^s$		
$x_1, x_2, u, v \in_R Z_q$		
$B = g_1^{x_1} g_2^{x_2}$		
$a = (a')^u g^v$		
$b = (b')^{su} A^v$		
$c = \mathcal{H}(A, B, z, a, b)$		
$c' = c/u$	$\xrightarrow{c'}$	
$r = r'u + v \bmod q$	$\xleftarrow{r'}$	$r' = c' X_S + w$

Fig. 1. *Blind signature protocol, embedding the verifier's identity. At the end of the protocol $\mathcal{V}$ verifies: $g^r \stackrel{?}{=} h^c a, A^r \stackrel{?}{=} z^c b$.*

Setup: A probabilistic polynomial-time *(p.p.t.)* prover $\mathcal{P}$ and a p.p.t. verifier $\mathcal{V}$.
Common **input** is $A, B, a, b, G_1, G_2, G_3$, with a, b, G_1, G_2, G_3 generators of G_q, a subgroup of prime order q of the multiplicative group Z_p^* for some large prime p. The prover is assumed to not know the relative discrete logarithms of a, b, G_1, G_2, G_3.
Secret input to $\mathcal{P}$ is x, v, w, such that $A \equiv a^x {G_1}^v \pmod p, B \equiv b^x {G_2}^w \pmod p$ (for simplicity we henceforth use the notation $A = a^x {G_1}^v$).
Notation: EqLog$[(A, a), G_1, (B, b), G_2]$ denotes that $A = a^x {G_1}^v$ and $B = b^x {G_2}^w$ for some $x \in G_Q$, and G_1, G_2 generators of G_Q. The reader may wish to think of that as $\log_a A = \log_b B$ for intuition (computations are always mod$P$).
The **proof** appears in Figure 2.

EqLog$[(A, a), G_1, (B, b), G_2]$
Input: $A, \quad B$
$\mathcal{P}$ proves that $A = a^x {G_1}^v, B = b^x {G_2}^w$, i.e., $\log_a A = \log_b B$:

$\mathcal{P}$		$\mathcal{V}$
$y, s_1, s_2, s_3 \in_R Z_q$		
$A' = a^y {G_1}^{s_1}, B' = b^y {G_2}^{s_2}$		
	$\xrightarrow{A',B'}$	$c \in_R Z_q$ *or*
	$\xleftarrow{c}$	$c = \mathcal{H}(A, A', a, G_1, B, B', b, G_2, \text{Date/Time, Info})$
$r = c \cdot x + y$		
$r_1 = c \cdot v + s_1, r_2 = c \cdot w + s_2$	$\xrightarrow{r, r_1, r_2}$	Verify: $a^r \cdot {G_1}^{r_1} \stackrel{?}{=} A^c \cdot A'$ and $b^r \cdot {G_2}^{r_2} \stackrel{?}{=} B^c \cdot B'$

Fig. 2. *Proof of equality of logarithms.*

The proof is essentially a set of parallel Schnorr knowledge proofs and can be used to prove equality of more than two logarithms (see "extensions" below). As is the case

in [Sch91], this minimal-knowledge proof can be made non-interactive and transferable under the random oracle model, with the challenge c being computed as a hash function of $\{A, A', a, G_1, B, B', b, G_2, \text{Date/Time, Info}\}$ and the hash function behaving like a random oracle.

We now discuss the correctness and zero-knowledge of the proof.

Correctness: It suffices to show that if a prover can answer to two challenges then s/he knows two representations as required (i.e., $A = a^x {G_1}^v$ and $B = b^x {G_2}^w$); then, if the prover cannot break the discrete log problem, s/he cannot know any other representations of A, B w.r.t. $(a, G_1), (b, G_2)$ respectively [Bra93b], since the relative logarithms of a w.r.t. G_1 and b w.r.t. G_2 are secret. Therefore there are only two possibilities: either the prover can answer to exactly one challenge (which depends on the construction of (A, B), i.e., it is "pre-selected" via the choice of (A, B)) or s/he knows the correct representations. But since the challenges are produced at random, the prover has only negligible probability of answering the "pre-selected" challenge without knowing the correct representations.

Now it is easy to see that given two answers to different challenges, $r = c{\cdot}x{+}y, r' = c'{\cdot}x{+}y$ and $[r_1 = c \cdot v + s_1, r_2 = c \cdot w + s_2], [r'_1 = c' \cdot v + s_1, r'_2 = c' \cdot w + s_2]$ one can solve the system of equations (where $r, r', r_1, r_2, r'_1, r'_2$ and c, c' are known values) to compute x, y, v, w, s_1 and s_2; thus if the prover can answer to two challenges, it knows (can compute) the correct representations.

Zero-knowledge: The proofs can be simulated w.r.t. an honest verifier. In the interactive setting this is done by the verifier selecting a random challenge c and random "responses" r, r_1, r_2, and computing $A' = A^{-c} a^r {G_1}^{r_1}, B' = B^{-c} b^r {G_2}^{r_2}$. Here we assume that the verifier is honest, i.e., that c is indeed randomly chosen (and can be learned in a simulation).

In the non-interactive setting the simulations are performed under the random oracle model, as in [PS96b]. Briefly, here the challenge is constructed using a hash function: $c = \mathcal{H}(A, A', a, G_1, B, B', b, G_2, \text{Date/Time, Info})$ where the hash function H is modeled as (i.e., assumed to act like) a "random oracle," or "perfect hash function". The simulator proceeds as previously; the random oracle assumption is used in the construction of c. I.e., we want to guarantee that after choosing c, r, r_1, r_2 and computing A', B', the equation $c = \mathcal{H}(A, A', a, G_1, B, B', b, G_2, \text{Date/Time, Info})$ still holds. For this we let the simulator "change" the output of the random oracle H, such that on input this particular vector it outputs c. Then the resulting "modified" random oracle cannot be distinguished from the original, since c was originally chosen as a random value. As the random value c is here substituted by the output of the random oracle, the "honest verifier" assumption is guaranteed; i.e., in the non-interactive version, and under the random oracle assumption, the equality of logarithm proofs are zero-knowledge. Full proof to appear in extended version.

Extensions: The same proof can be used for more than two values; thus we can define $\text{EqLog}[(A, a), G_1, (B, b), G_2, (C, c), G_3]$ to prove equality between the respective logarithms of A, B and C. The protocol and security proofs are similar; we omit description for conciseness. (Although it is simple to observe that two consecutive proofs of equality of logarithms for (A, B) and (B, C) respectively achieve the same result—but with slightly higher computation.) This extended version is used in section 5.

3.3 Indirect Discourse Proofs

We now show how proofs of equality of logarithms can be used to create indirect discourse proofs. These will be used for the protocols of section 4 but *not* for section 5; if interested only in simplicity and not backwards compatibility we encourage the reader to move directly to section 5.

In this particular example of indirect discourse proofs, tailored to our purposes, we will construct a proof which shows a specific construction for three numbers A, B, C. This is a more general construction than we actually need for section 4.

The proof appears in figure 3. The interactive form is shown, but the proof can be made non-interactive by computing the challenge using a random oracle $\mathcal{H}$: $c = \mathcal{H}(A, B, C, A', B', a, b, G_1, G_2, G_3, \text{Date/Time, Info})$, where "Info" is some transaction-related information (such as the identity of $\mathcal{V}$ or the transaction purpose/description/amount).
Notation: we use **IndPrf**$[(A, a), G_1, (B, C|G_3), G_2]$ to denote that $A = a^x {G_1}^v$, $B = C^x {G_2}^z {G_3}^t = b^{ex} {G_2}^w$.

IndPrf$[(A, a), G_1, (B, b, C|G_3), G_2]$
Input: A, $B = b^s {G_2}^w$, $C = b^e {G_3}^u$
$\mathcal{P}$ will prove to $\mathcal{V}$ that $A = a^x {G_1}^v, B = C^x {G_2}^z {G_3}^t = b^{ex} {G_2}^w$:

$\mathcal{P}$		$\mathcal{V}$
$y, s_1, s_2, s_3 \in_R Z_q$		
$A' = a^y {G_1}^{s_1}, B' = C^y {G_2}^{s_2} {G_3}^{s_3}$		
	$\xrightarrow{A', B'}$	$c \in_R Z_q$
	$\xleftarrow{c}$	
$r = c \cdot x + y$		
$r_1 = c \cdot v + s_1, r_2 = c \cdot w + s_2, r_3 = s_3 - r \cdot u$	$\xrightarrow{r, r_1, r_2, r_3}$	Verify:
		$a^r \cdot {G_1}^{r_1} \stackrel{?}{=} A^c \cdot A'$ and
		$C^r \cdot {G_2}^{r_2} \cdot {G_3}^{r_3} \stackrel{?}{=} B^c \cdot B'$

Fig. 3. *Indirect discourse proof.*

We omit the proof of security due to lack of space; its construction is similar to the correctness and zero-knowledge proof of the protocol for proving equality of logarithms in section 3.2 above.

4 Retaining Existing Infrastructure

Changing systems that have already been implemented sometimes requires a disproportionate amount of effort, compared to the changes required. Thus it is important to devise techniques that enhance functionality without affecting existing systems. In this section we show how modular additions to off-line electronic cash systems can be used to construct FOLC in a seamless manner. We elect to show our additions on the Brands' protocol, but similar solutions are possible in other blinding protocols. As mentioned earlier, our focus is primarily security (i.e., basing anonymity on the decision Diffie-Hellman assumption) and efficiency.

4.1 Coin Tracing

Coin tracing can be performed efficiently using the techniques of [CMS96,DFTY97], modified to allow for provable security. To add it in a modular way we need a preliminary stage, in which the Trustee entity $\mathcal{T}$ is created, and an addition to the withdrawal protocol. The following steps are performed: during the withdrawal protocol an additional value $I' = I {g_3}^{s^{-1}} {g_4}^t$ is created (where g_4 is an additional generator of G_q) and its relationship to an ElGamal encryption $E_1 = (I {g_2} {g_4}^t)^s {f_1}^m, E_2 = {g_1}^m$ is proven using indirect discourse proofs; here $f_1 = {g_1}^{X_\tau}$ is a public key published by the Trustee. The coin then embeds I' instead of I and becomes $A = I^{s'} {g_2}^{s'} {g_3}^{s' \cdot s^{-1}} {g_4}^{t \cdot s^{-1}}$, where s' is the user's blinding factor.

At payment the verifier checks that the coin is of the form $A = {g_1}^x {g_2}^y {g_4}^z g_3$, thereby indirectly forcing the user to set $s' = s$.

Coin tracing is then performed by the trustee decrypting (E_1, E_2) to obtain the paid coin $\bar{A} = A/g_3 = (Ig_2{g_4}^t)^s$.

This method retains the anonymity of the user (based on the decision Diffie-Hellman assumption) with minimal computational overhead, while it requires no changes to the existing blind signature protocol. However we do not describe it in detail as (1) it can be derived in a straightforward manner from [DFTY97] and section 3.3 above, while (2) in section 5 we show a more efficient method achieving both owner and coin tracing.

4.2 Owner Tracing

An off-line coin by its nature has its owner's identity embedded in it. Thus for owner tracing all we need is an encryption of the user's identity using a public key encryption system, in such a way that the encryption is linked to the coin. Hence, Trustees can open the ciphertext to obtain the identity. An indirect discourse proof during payment assures the receiver that the encrypted identity is the same as the one embedded in the coin. The additions to the basic protocol are limited to a preliminary stage, in which the Trustee entity $\mathcal{T}$ is created, and to a modular addition to the payment protocol. Here we show the payment protocol that corresponds to the blinding protocol of section 3.1.

$\mathcal{T}$'s public information: Public key $f_2 = g_2^{X_\mathcal{T}}$ associated with private key $X_\mathcal{T} \in_R G_q$.

The new payment protocol:

$\mathcal{U}$		$\mathcal{R}$
$m \in_R Z_q$		
$D_1 = Ig_2^{X_\mathcal{T} m}, D_2 = g_2^m$		
$V_1 = \text{EqLog}[(D_1, f_2), g_1, (D_2, g_2), \text{nil}]$		
$V_2 = \text{IndPrf}[(\bar{A}, \{g_2, g_4\}), g_1, (\bar{A}, g_1, D_1 \mid f_2), g_2]$		
In $\mathbf{V_2}$, $\mathcal{U}$ uses $\mathbf{B} = \mathbf{g_1}^{\mathbf{x_1}}\mathbf{g_2}^{\mathbf{x_2}}$ from withdrawal, instead of random $\mathbf{A'}$		
	$\xrightarrow{D_1, D_2, V_1, V_2}$	$D_2 \stackrel{?}{\neq} 1$
		Verify V_1, V_2

This protocol proves to $\mathcal{R}$ that (D_1, D_2) is an ElGamal [ElG85] encryption of I, based on f_2, where I is the same identity as the one embedded in the coin A. In particular, first V_1 proves that $D_1 = {g_1}^x {g_2}^{X_\mathcal{T} m}, D_2 = {g_2}^m$ for some x, m. Then V_2 proves that $x \cdot s \equiv u \cdot s \pmod{Q}$ where $\bar{A} = {g_1}^{us}{g_2}^s{g_4}^{ts}$ is the user's coin; therefore, $x \equiv u \pmod{Q}$ and thus $D_1 = Ig_2^{X_\mathcal{T} m}$ as required.

Efficiency: The protocol poses minimal additional communication and computation requirements (on the order of 7 exponentiations for $\mathcal{U}$ and 9 for $\mathcal{R}$), while keeping $\mathcal{T}$ off-line in all cases.

5 Simplified FOLC

Although the protocols of the previous sections are efficient and secure, it turns out that if we can alter some design aspects of the basic e-cash system it is possible to perform coin and owner tracing in one step, thus effectively reducing in half the computational requirements, while remaining within the same security assumptions.

The idea is to combine the identity of the user with a "coin identifier" (as in [CMS96]) to an *unconditional* commitment. Then, this commitment is signed using the blinding protocol. The commitment is constructed such that the resulting coin is itself part of an ElGamal encryption of the user's identity (this idea has its root in [dST98]). Thus, one execution of the blinding protocol (which is the bulk of the computation at withdrawal) in effect performs two tasks at once: tracing the coin and encrypting the user's identity. The blinding protocol

used is a modification of the one appearing in section 3.1 that operates on 3 instead of 2 generators, but whose security is unchanged [Bra93a]. The fact that the commitment is unconditional allows us to prove anonymity under the decision Diffie-Hellman assumption.

For coin tracing an ElGamal encryption of the "coin identifier" is constructed at withdrawal, and its correct construction (with respect to the commitment) is verified using proofs of equality of logarithms. For owner tracing one additional value is constructed (at payment) to form an ElGamal encryption in conjuction with the coin; the correctness of this value requires proofs of equality of logarithms instead of indirect discourse proofs. We proceed with the details.

Bank's setup protocol: (performed once by $\mathcal{B}$)
Primes p and q are chosen such that $|p-1| = \delta + k$ for a specified constant δ, and $p = \gamma q + 1$, for a specified integer γ. Then a unique subgroup G_q of prime order q of the multiplicative group Z_p^* and generators g, g_1, g_2, g_3, g_4 of G_q are defined. Secret key $X_{\mathcal{B}} \in_R Z_q$ is created.[2] Hash functions $\mathcal{H}, \mathcal{H}_0, \mathcal{H}_1, \ldots$, from a family of collision intractable hash functions are also defined. $\mathcal{B}$ publishes $p, q, g, g_1, g_2, g_3, g_4, (\mathcal{H}, \mathcal{H}_0, \mathcal{H}_1, \ldots)$ and its public keys $h = g^{X_{\mathcal{B}}}$, $h_1 = {g_1}^{X_{\mathcal{B}}}, h_2 = {g_2}^{X_{\mathcal{B}}}, h_3 = {g_3}^{X_{\mathcal{B}}}$.
Trustee's setup protocol: (performed once by $\mathcal{T}$)
Public keys $f_2 = g_2^{X_{\mathcal{T}}}, f_3 = g_3^{X_{\mathcal{T}}}$ associated with private key $X_{\mathcal{T}} \in_R Z_q$ are published.
User's setup (account opening) protocol: (performed for each user $\mathcal{U}$)
The bank $\mathcal{B}$ associates user $\mathcal{U}$ with $I = g_1^{u_1}$ where $u_1 \in G_q$ is generated by $\mathcal{U}$ and $g_1^{u_1} g_2 \neq 1$. $\mathcal{U}$ also proves (using the Schnorr identification scheme [Sch91]) to $\mathcal{B}$ that he knows how to represent I w.r.t. g_1.

Withdrawal: (over an authenticated channel between $\mathcal{B}$ and $\mathcal{U}$)
An intermediate value $I' = {g_1}^{u_1 s^{-1}} {g_3}^{s^{-1}} {g_4}^{t}$ is created. The user constructs an ElGamal encryption $E_1 = {g_2}^{s} {f_3}^{m}, E_2 = {g_3}^{m}$ of ${g_2}^{s}$ and proceeds to prove its correct construction w.r.t. I'. The constructions of I', E_1, E_2 are proven using proofs of equality of logarithms. The blinding protocol of section 3.1 then proceeds with I' replacing I.

Note that during the payment protocol the user is expected to present a coin of a specific structure; this forces him to use the committed value s as the blinding factor. Thus the coin contains ${g_2}^{s}$ and can be traced by decrypting (E_1, E_2).
The withdrawal protocol results in a signature of the form appearing in equation (1) (see section 3.1):

$\mathcal{U}$		$\mathcal{B}$
$m, s, t \in_R Z_q$		
$I' = {g_1}^{u_1 s^{-1}} {g_3}^{s^{-1}} {g_4}^{t}$		
$E_1 = {g_2}^{s} {f_3}^{m}, E_2 = {g_3}^{m}$		
$V_1 = \mathrm{EqLog}[(E_1, f_3), g_2, (E_2, g_3), \mathrm{nil}]$		
$V_2 = \mathrm{EqLog}[(g_3, I'), (g_1, g_4),$ $(E_1, g_2), f_3, (I, I'), (g_3, g_4)]$		
	$\xrightarrow{I', E_1, E_2, V_1, V_2}$	$E_2 \stackrel{?}{\neq} 1$
		Verify V_1, V_2
		$w \in_R Z_q$
	$\xleftarrow{a', b', b''}$	$a' = g^w, b' = (I' g_2)^w, b'' = {g_4}^w$

[2] We assume, for simplicity, that only one denomination is used. A different key for each denomination is necessary.

$A = (I' {g_2}{g_4}^{t^{-1}})^s = {g_1}^{u_1} {g_2}^s g_3$
$z = {h_1}^{u_1} {h_2}^s h_3 \ [= A^{X_{\mathcal{B}}}]$
$x_1, x_2, u, v \in_R Z_q$
$B = g_1^{x_1} g_2^{x_2}$
$a = (a')^u g^v$
$b = (b' {b''}^{t^{-1}})^{su} A^v \ [= A^{wu+v}]$
$c = \mathcal{H}(A, B, z, a, b)$
$c' = c/u$ $\xrightarrow{c'}$
$\xleftarrow{r'}$ $r' = c' X_{\mathcal{B}} + w$
$r = r'u + v \bmod q$

At the end of the protocol $\mathcal{U}$ verifies: $g^r \stackrel{?}{=} h^c a, A^r \stackrel{?}{=} z^c b$.

Payment: (performed between $\mathcal{U}$ and $\mathcal{R}$ over an anonymous channel)
At payment time $\mathcal{U}$ supplies information to the receiver $\mathcal{R}$ (which is later forwarded to the bank) so that if a coin is double-spent the user $\mathcal{U}$ is identified.

The user provides the signature on the coin $A = {g_1}^x {g_2}^y g_3$ and uses $A_1 = A/g_3$ for the verifications of the payment protocol. I.e., the user is forced to use s as the blinding factor, in order to "neutralize" the exponent s^{-1} of g_3.

The user also provides the value $A_2 = {f_2}^s$ and proves that this, together with the coin, forms a (modified) ElGamal encryption of ${g_1}^x$ which, from the withdrawal protocol, can only be ${g_1}^{u_1} = I$, i.e., the user's identity. To prove the construction all that is needed is the proof of equality of logarithms $V_3 = \mathrm{EqLog}[(A_1, g_2), g_1, (A_2, f_2), \mathrm{nil}]$.

The payment protocol ($\mathcal{U}$ and $\mathcal{R}$ agree on date/time, to be used as input to the non-interactive challenge):

$\mathcal{U}$		$\mathcal{R}$
$A_1 = {g_1}^{u_1} {g_2}^s \ [= A/g_3]$		
$A_2 = {f_2}^s$		
$V_3 = \mathrm{EqLog}[(A_1, g_2), g_1, (A_2, f_2), \mathrm{nil}]$		
$\mathcal{U}$ uses $\mathbf{B}$ instead of $\mathbf{A}'$ in the construction of $\mathbf{V_3}$		
	$\xrightarrow{A_1, A_2, A, B, (z,a,b,r)}$	$A_1 \stackrel{?}{\neq} 1, A_1 g_3 = A$
		$sig(A, B) \stackrel{?}{=} (z, a, b, r)$
		Verify V_3

Deposit: (performed between $\mathcal{R}$ and $\mathcal{B}$ over an authenticated channel)
$\mathcal{R}$ sends a transcript of the payment protocol to $\mathcal{B}$ who verifies the (non-interactive) proofs.

Owner tracing: (performed between $\mathcal{B}$ and $\mathcal{T}$ over an authenticated channel)
The bank simply sends the deposited coin to the trustee $\mathcal{T}$. $\mathcal{T}$ uses the private key to decrypt the ElGamal encryption (A_1, A_2) and sends the decrypted value (i.e., $I = {g_1}^{u_1}$) to $\mathcal{B}$. The bank indexes this against its account database to find the coin's owner.
Coin tracing: (performed between $\mathcal{B}$ and $\mathcal{T}$ over an authenticated channel)
The bank sends a withdrawal transcript to $\mathcal{T}$. The trustee decrypts the ElGamal encryption (E_1, E_2) to obtain the value ${g_2}^s$; the bank then searches its deposit databases for the coin $A = I {g_2}^s g_3$, where I is the user's identity.

Efficiency: The protocols require around 8 and 11 exponentiations for the user and bank at withdrawal[3] and 4 and 2 for the user and receiver at payment.

6 Security

The security of FOLC can be described in three parts: (1) security for the payees and bank (i.e., unreusability, unforgeability, and unexpandability of coins; see [FY93] for a precise model), (2) security of the extensions (i.e., the ability of the trustees to trace), and (3) security (anonymity) for the legitimate users. Our protocols guarantee the following:

- (1) above is unchanged from the underlying basic off-line e-cash protocol. This can be seen since the blinding protocol is either unmodified (section 4) or (in section 5) the modifications do not impair its security [Bra93a]. See appendix A for a sketch of the proof.
- (2) above is based on the correctness property of the proof of equality of logarithms, i.e., it is guaranteed based on the existence of hash functions that behave like random oracles. The proof here is straightforward (verify that the user is constrained in the construction of the ElGamal encryptions, based on the proofs of equality of logarithms) but relatively lengthy. See appendix A for more details.
- Finally, (3) above is based on the semantic security of the (inverted) ElGamal encryption, i.e., on the decision Diffie-Hellman assumption. Intuitively, note that the disclosed values do not reveal any information; a sketch of an actual proof which shows that if anonymity is broken then the decision D-H problem does not hold, is given in appendix B.

7 Discussion and Open Problems

We have constructed a simple solution for fair off-line electronic cash, utilizing recent security proofs for homomorphic encryption schemes [TY98]. We believe that the biggest open problem is to prove security under even more strict assumptions while keeping the efficiency of our constructions. A first step to this direction may be a recently proposed encryption scheme with homomorphic properties, whose semantic security is equivalent to factoring [OU98]. Similarly, we would like to see blinding protocols whose restrictive properties can be proven secure.

References

BCC+92. S. Brands, D. Chaum, R. Cramer, N. Ferguson, and T. Pedersen. Transaction systems with observers, August 13 1992. Unpublished manuscript.

Bra93a. S. Brands. An efficient off-line electronic cash system based on the representation problem. Technical Report CS–R9323, CWI (Centre for Mathematics and Computer Science), Amsterdam, 1993. anonymous ftp: ftp.cwi.nl:/pub/CWIreports/AA/CS-R9323.ps.zip.

Bra93b. S. Brands. Untraceable off-line cash in wallets with observers. In *Advances in Cryptology — Crypto '93, Proceedings (Lecture Notes in Computer Science 773)*, pages 302–318. Springer-Verlag, 1993. Available at http://www.cwi.nl/ftp/brands/crypto93.ps.Z.

CFMT96. A. Chan, Y. Frankel, P. MacKenzie, and Y. Tsiounis. Mis-representation of identities in e-cash schemes and how to prevent it. In *Advances in Cryptology — Proceedings of Asiacrypt '96 (Lecture Notes in Computer Science 1163)*, pages 276–285, Kyongju, South Korea, November 3–7 1996. Springer-Verlag. Available at http://www.ccs.neu.edu/home/yiannis/pubs.html.

[3] Here we count the additional burden over basic e-cash that is required for the "fairness" extensions.

CFN90. D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *Advances in Cryptology —Crypto '88 (Lecture Notes in Computer Science)*, pages 319–327. Springer-Verlag, 1990.

CMS96. J. Camenisch, U. Maurer, and M. Stadler. Digital payment systems with passive anonymity-revoking trustees. In *Esorics '96*, (Lecture Notes in Computer Science 1146), pages 33–43. Springer-Verlag, Italy, 1996. Available at http://www.inf.ethz.ch/personal/camenisc/publications.html.

CP93a. D. Chaum and T.P. Pedersen. Wallet databases with observers. In E. Brickell, editor, *Advances in Cryptology — Crypto '92, Proceedings (Lecture Notes in Computer Science)*, pages 90–106. Springer-Verlag, New York, 1993. Santa Barbara, California.

CP93b. R. Cramer and T. Pedersen. Improved privacy in wallets with observers. In *Advances in Cryptology: Eurocrypt '93, Proceedings (Lecture Notes in Computer Science 765)*, pages 329–343. Springer-Verlag, 1993.

DFTY97. G. Davida, Y. Frankel, Y. Tsiounis, and M. Yung. Anonymity control in e-cash. In *Proceedings of the 1st Financial Cryptography conference (Lecture Notes in Computer Science 1318)*, Anguilla, BWI, February 24-28 1997. Springer-Verlag. To appear. Available at http://www.ccs.neu.edu/home/yiannis/pubs.html.

dST98. A. de Solages and J. Traore. An efficient fair off-line electronic cash system with extensions to checks and wallets with observers. In *Proceedings of the 2nd Financial Cryptography conference*, Anguilla, BWI, February 1998. Springer-Verlag. To appear.

ElG85. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory*, 31:469–472, 1985.

FTY96. Y. Frankel, Y. Tsiounis, and M. Yung. Indirect discourse proofs: achieving fair off-line e-cash. In *Advances in Cryptology, Proc. of Asiacrypt '96 (Lecture Notes in Computer Science 1163)*, pages 286–300, Kyongju, South Korea, November 3–7 1996. Springer-Verlag. International patent pending. Available at http://www.ccs.neu.edu/home/yiannis/pubs.html.

FY93. M. Franklin and M. Yung. Secure and efficient off-line digital money. In *Proceedings of the twentieth International Colloquium on Automata, Languages and Programming (ICALP 1993), (Lecture Notes in Computer Science 700)*, pages 265–276. Springer-Verlag, 1993. Lund, Sweden, July 1993.

GM84. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.

JY96. M. Jakobson and M. Yung. Revokable and versatile e-money. In *3rd ACM Symp. on Computer and Communication Security*, March 1996.

NR97. M. Naor and O. Reingold. On the construction of pseudo-random permutations: Luby-Rackoff revisited. In *38th Annual Symp. on Foundations of Computer Science (FOCS)*, 1997.

OU98. T. Okamoto and S. Uchiyama. An efficient public-key cryptosystem. In *Eurocrypt 98*, Espoo, Finland, May 31–June 4 1998. Springer-Verlag. To appear. Preliminary announcement in Workshop on Public Key Cryptography, Feb. 5-6 1998, Yokohama, Japan.

PS96a. D. Pointcheval and J. Stern. Provably secure blind signature schemes. In *Advances in Cryptology, Proc. of Asiacrypt '96 (Lecture Notes in Computer Science)*, Kyongju, South Korea, November 3–7 1996. Springer-Verlag. To appear. Available at http://www.ens.fr/dmi/equipes_dmi/grecc/pointche/pub.html.

PS96b. D. Pointcheval and J. Stern. Security proofs for signature schemes. In U. Maurer, editor, *Advances in Cryptology, Proc. of Eurocrypt '96*, pages 387–398, Zaragoza, Spain, May 11–16, 1996. Springer-Verlag. Available at http://www.ens.fr/dmi/equipes_dmi/grecc/pointche/pub.html.

Sch91. C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

TY98. Y. Tsiounis and M. Yung. On the security of El Gamal-based encryption. In *International workshop on Public Key Cryptography (PKC '98)*, Yokohama, Japan, February 5-6 1998. Springer-Verlag. To appear. Available at http://yiannis.home.ml.org.

A Security for the Bank and Receivers (Shops)

We show a sketch of the proof for (1) and (2) of section 6; for shortness we limit the discussion to the protocol of section 5.

At payment, V_3 proves that $A_1 = g_2{}^x g_1{}^y$ and $A_2 = f_2{}^x$ for some x, y; i.e., that (A_1, A_2) forms an ElGamal encryption of $g_1{}^y$ based on the Trustee's public key f_2. Also notice that V_3 is always carried out with the same randomness $B = g_1{}^{x_1} g_2{}^{x_2}$, therefore if it is executed twice it will reveal x (which, as we will see, is the user's private key).

Then, at withdrawal, V_1 proves that $E_1 = g_2{}^s f_3{}^m, E_2 = g_3{}^m$ for some s, m, i.e., that (E_1, E_2) forms an ElGamal encryption of $g_2{}^s$. Also, V_2 proves that $g_3 = (I')^v g_1{}^w g_4{}^t, E_1 = g_2{}^v f_3{}^u, I = (I')^v g_3{}^r g_4{}^z$ for some v, w, t, u, r, z. But from V_1 we have that $E_1 = g_2{}^s f_3{}^m$, thus $v = s, u = m$ and therefore $g_3 = (I')^s g_1{}^w g_4{}^t, E_1 = g_2{}^s f_3{}^m, I = (I')^s g_3{}^r g_4{}^z$. By rearranging the equations involving I' we get $I' = g_3{}^{s^{-1}} g_1{}^{w'} g_4{}^{t'}$ and $I' = I^{s^{-1}} g_3{}^{r'} g_4{}^{z'}$, where $w' = -ws^{-1}, t' = -ts^{-1}$, etc. Also at user setup it has been proven that $I = g_1{}^{u_1}$, hence we have that $I' = (g_1{}^{u_1})^{s^{-1}} g_3{}^{r'} g_4{}^{z'} = g_1{}^{u_1 s^{-1}} g_3{}^{r'} g_4{}^{z'}$, and (from the first equation on I'), $I' = g_1{}^{u_1 s^{-1}} g_3{}^{s^{-1}} g_4{}^{t'}$, where t' is unknown to the bank, and s is the same as in E_1.

Now, if we assume that the withdrawal protocol is a restrictive blind signature protocol (an assumption initially made and argued for in [Bra93a]), i.e., under the terminology of [FY93] it satisfies unforgeability and unexpandability, then the signed number A must be of the form $A = (I' g_2)^u g_4{}^v$, for some u, v, i.e., $A = g_1{}^{u_1 s^{-1} u} g_3{}^{s^{-1} u} g_2{}^u g_4{}^{t' u} g_4{}^v$. From the payment above we have seen that $A = A_1 g_3 = g_2{}^x g_1{}^y g_3$. Therefore, it must be that $s^{-1}u \equiv 1 \pmod q$ and $t'u + v \equiv 0 \pmod q$; in particular, $u \equiv s \pmod q$. Putting these values in A we get $A = g_1{}^{u_1} g_2{}^s g_3$, and therefore $A_1 = g_1{}^{u_1} g_2{}^s, A_2 = f_2{}^s$, as required for tracing.

Thus we have shown that if unforgeability and unexpandability are satisfied for the starting scheme (in this case [Bra93a]) then traceability and bank/shop security also hold for FOLC.

B Anonymity

For anonymity (i.e., untraceability as defined in [FY93]) we want to prove that given a pair of withdrawal protocols and the corresponding paid coins, a collaboration of bank and shops cannot decide which coin came from which withdrawal. Again we limit the discussion to the protocol of section 5. The data that is available for this linking is the following[4] (we omit the values I'_0, I'_1 since they are unconditionally blinded by the random t_0, t_1):

At withdrawal: $\left[V_1^0, V_2^0, E_1^0 = g_2{}^{s_0} f_3{}^{m_0}, E_2^0 = g_3{}^{m_0}, c'^0\right]$ and $\left[V_1^1, V_2^1, E_1^1 = g_2{}^{s_1} f_3{}^{m_1}, E_2^1 = g_3{}^{m_1}, c'^1\right]$.

At payment: $\left[A_1^i = g_1{}^{u_i} g_2{}^{s_i}, A_2^i = f_2{}^{s_i}, V_3^i, z^i, a^i, b^i, r^i, B^i\right]$ and $\left[A_1^{\bar{i}} = g_1{}^{u_{\bar{i}}} g_2{}^{s_{\bar{i}}}, A_2^{\bar{i}} = f_2{}^{s_{\bar{i}}}, V_3^{\bar{i}}, z^{\bar{i}}, a^{\bar{i}}, b^{\bar{i}}, r^{\bar{i}}, B^{\bar{i}}\right]$, $i, \bar{i} \in \{0, 1\}, i \neq \bar{i}$.

The linking problem is to determine whether i is 0 or 1.

Suppose now we have a machine $\mathcal{M}$ which given the above information can find i. Then we can use this machine to break the ElGamal encryption in the sense of indistinguishability, i.e., break the decision D-H assumption [TY98], as follows (sketch):

Let $\mu_i = g_2{}^{s_i}, \mu_{\bar{i}} = g_2{}^{s_{\bar{i}}}$ be two messages, and let $(E_1^0, E_2^0), (E_1^1, E_2^1)$ be the encryptions of μ_0, μ_1 respectively. Then we feed $\mathcal{M}$ with these encryptions, plus $(A_1^i, A_2^i), (A_1^{\bar{i}}, A_2^{\bar{i}})$, which we can construct for a random $u_i, u_{\bar{i}}$, since we know $s_i, s_{\bar{i}}$. We then simulate V_j^l, for $j = \{1, 2\}, l = \{0, 1\}$ and $V_3^i, V_3^{\bar{i}}$; the simulations for V_3 require random values to be chosen for $B^i, B^{\bar{i}}$. Then the signatures of the coins are simulated, i.e., random values $R_i, c^i, R_{\bar{i}}, c^{\bar{i}}$ are chosen and $a^i = g^{R^i}, r^i = c^i X_{\mathcal{B}} + R^i, z^i = (A^i)^{X_{\mathcal{B}}}, b^i = (A^i)^{R^i}, a^{\bar{i}} = g^{R^{\bar{i}}}, r^{\bar{i}} = c^{\bar{i}} X_{\mathcal{B}} + R^{\bar{i}}, z^{\bar{i}} =$

[4] Here X^i or X_i denotes value X at protocol i.

$(A^{\bar{i}})^{X_{\mathcal{B}}}, b^{\bar{i}} = (A^{\bar{i}})^{R^{\bar{i}}}$ are calculated. Finally ${c'}^0, {c'}^1$ are chosen at random (it is easy to verify that for any choice of c, c', R, setting $u = c/c', v = R - wu$ satisfies both $c' = c/u$ and the values of a, b, as calculated using R; thus the simulation is perfect). These values are then inserted into $\mathcal{H}(A^i, B^i, z^i, a^i, b^i), \mathcal{H}(A^{\bar{i}}, B^{\bar{i}}, z^{\bar{i}}, a^{\bar{i}}, b^{\bar{i}})$ and the values of the hash function at these points are changed so that the results are $c^i, c^{\bar{i}}$ respectively.

The whole output of the simulator (consisting of the above values) is then fed to $\mathcal{M}$, which returns the value of i, and thus breaks the semantic security of ElGamal encryption.

Finally, the above problem of distinguishing between two ciphertexts can be embedded in a context of polynomially many withdrawals using standard methods.