Preventing Double-Spent Coins from Revealing User's Whole Secret

DaeHun Nyang and JooSeok Song

Department of Computer Science, Yonsei University
SeodaemunGu ShinchonDong 134, Seoul 120-749, Korea
{nyang,jssong}@emerald.yonsei.ac.kr

Abstract. Most digital cash systems have the identity revelation capability under the condition of doublespending, bu t the capability may be misused as a framing tool by Bank. We present a method that provides both the identity revelation capability and the framing prevention property.
Keywords : Cryptography, Digital Cash, Zero-Knowledge Interactive Proof, Blind Signature

1 Motivation

Digital cash is the most versatile tool for the electronic commerce, and it may be an alternative of the paper money. Chaum introduced quite a new concept of digital money in his paper [3]. Besides its conveniences in its usage, Chaum's digital cash provides its user anonimity. In 1993, Brands designed an efficient off-line cash system, which includes not only all merits that Chaum's system has, but also other good functionalities such as the framing prevention[1].

One of the most intriguing properties that Brands' scheme has is that User is computationally protected against Bank's framing, or Bank cannot compute a proof of double-spending if User follows the protocols and does not double-spend. Anonimoty control and concern about Bank's framing attempts has led researches such as [5], [6], [7] and [10]. But, they did not treat the following problem: what if User spends a coin more than twice? Does the fact guarantee that User is guilty for $n(> 2)$-spending? When Bank insists that $n(> 2)$-spending occurred in a digital cash system, there are actually two possibilities. One is Bank's framing of User. Since Bank knew already user's secret when double-spending occurred, it is possible to make another illegal copy of the double-spent coin, if needed, with the collusion of Shop. The other possibility is that User actually spent one coin more than twice. Besides $n(> 2)$-spending problem, u_1 revealed may be misused by Bank in various ways(e.g. Bank can withdraw digital coins using u_1).

This unavoidable problem comes from the lack of the user's secrecy that would not be revealed even when she double-spends a coin. If she has a key that would maintain its secrecy after the double-spending revelation and the key is required to perform the payment protocol, User cannot assert Bank's framing while she spent a coin more than twice. Owing to the existence of the secret that only User knows, Bank can make Judge sure that $n(> 2)$-spending has really occurred by User, not counterfeited by Bank itself.

JooSeok Song (Ed.): ICISC'99, LNCS 1787, pp. 30–37, 2000.

However, when double-spending occurs, the maintenance of secrecy or framing prevention looks incompatible with the identity revelation. In this paper, we solve the problem by introducing another secret into Brands' scheme and using the normal 3-move zero-knowledge interactive proof(ZKIP).

2 Features on Brands' Scheme

First, we briefly describe Brands' scheme. Refer to [1] for further understanding.

- The setup of the system: Bank generates at random a generator-tuple (g, g_1, g_2) and a number $x \in_R Z_q^*$. Also, she chooses two collision-intractable hash functions H, H_0. H is used for the signature generation/verification and H_0 is for the computation of challenges. The generator-tuple and two hash functions are her public-key, and x is her private-key.
- Opening an account: User's identity is $I = g_1^{u_1} g_2$(User is computationally protected from Bank's framing) or $I = g_1^{u_1} g_2^{u_2}$(the framing attempts of Bank have negligible probability of success, regardless of computing power). Because the case of $I = g_1^{u_1} g_2^{u_2}$ is easily adapted from the case of $I = g_1^{u_1} g_2$, we proceed for the latter case.
  User generates randomly $u_1 \in_R Z_q$ and computes $I = {g_1}^{u_1}$. If $Ig_2 \neq 1$, User transmits I to Bank, and keeps u_1 secret. Bank keeps I with User's identifying information in her database. Bank computes $z = (Ig_2)^x$, and transmits it to User.
- The withdrawal protocol: After proving ownership of her account, User perform the following withdrawal protocol.
  1. Bank generates $w \in_R Z_q$, and sends $a = g^w$ and $b = (Ig_2)^w$ to User.
  2. User Generates $s \in_R Z_q^*, x_1, x_2 \in_R Z_q$, and computes $A = (Ig_2)^s, B = g_1^{x_1} g_2^{x_2}$ and $z' = z^s$. User generates $u, v \in_R Z_q$ and computes $a' = a^u g^v, b' = b^{su} A^v$. User then computes the challenge $c' = H(A, B, z', a', b')$, and sends the blinded challenge $c = c'/u \bmod q$ to Bank.
  3. Bank sends the response $r = cx + w \bmod q$ to User, and debits the account of User.
  4. User accepts iff $g^r = h^c a$ and $(Ig_2)^r = z^c b$. If this holds, User computes $r' = ru + v \bmod q$.

  Now User has a coin that looks like $(A, B, Sign(A, B) = (z', a', b', r'))$.
- The payment protocol:
  1. User sends $(A, B, Sign(A, B))$ to Shop.
  2. If $A \neq 1$, then Shop computes and sends the challenge $d = H_0(A, B, I_S, date/time)$.
  3. User computes and sends the responses $r_1 = d(u_1 s) + x_1 \bmod q$ and $r_2 = ds + x_2 \bmod q$.
  4. Shop accepts iff $Sign(A, B)$ is valid, and $g_1^{r_1} g_2^{r_2} = A^d B$.
- The deposit protocol: Shop performs the same procedure as that in the payment protocol with Bank. Bank accepts iff $Sign(A, B)$ is valid, and $g_1^{r_1} g_2^{r_2} = A^d B$ and A has not been stored before. If A is already in the deposit database, Bank can find the double-spender's identity by computing $g_1^{(r_1 - r_1')/(r_2 - r_2')}$.

The role of u_1 in Brands' scheme is both to identify each user's coins when double-spending occurs and to prevent Bank from framing User on double-spending. Thus, the revelation of u_1 means not only double-spending detection, but also the possibility of Bank's framing. After Bank's double-spending detection, Bank comes to know u_1 and s. With u_1 and s, Bank can make another forged transcript of dealing with the double-spent coin. In this way, Bank can frame $n(> 2)$-spending on User for an arbitrary n. Besides that, u_1 revealed may be misused by Bank in various ways(e.g. Bank can withdraw digital coins using u_1). If needed, Bank may ask Shop to accept the forged transcript of dealing as true in front of Judge. Confronted with this case, User has no way to insist that she has spent the coin not $n(> 2)$ times, but only twice.

3 Protocol Design Concept

As mentioned before, u_1, the one secret that User has plays two roles in Brands' scheme. With only one secret, we cannot obtain both the identity revelation and the framing prevention when double-spending occurs. As a natural consequence, we separate u_1 into u_1 and u_2 such that each secret might play only one role. That is, u_1 reveals User's identity and u_2 prevents Bank from framing User on $n(> 2)$-spending when double-spending occurs. u_2 would not be revealed even when User double-spends a coin. Separation of u_1 into u_1 and u_2 according to their roles enables us to achieve both functionalities.

Note that Brands' payment protocol does not include the commitment move by the prover, whereas 3-move ZKIP systems such as [4,8] include the commitment stage. In that point of view, Brands' scheme separates the commitment stage from the payment protocol and includes it in the withdrawal protocol. As known, committed values must be different whenever the prover proves its knowledge in the 3-move ZKIP. If the same committed value is used again, the verifier can easily compute the prover's secret. The usage of the same committed value during the proof of the same knowledge in ZKIP corresponds exactly to the double-spending of the same coin in Brands' setting. And the exposure of the secret in ZKIP corresponds to the exposure of User's identity in Brands' scheme.

The knowledge of the first secret(u_1) is proved in the same way as that of Brands', whereas the proof of knowledge of the second secret(u_2) is performed by the normal 3-move ZKIP. This setting allows that u_1 will be revealed but u_2 will be kept secret whenever double-spending occurs. Thus, the payment protocol now, looks the normal ZKIP for u_2.

Our approach has another merit that User does not need to perform the account opening procedure again. Unlike Brands' scheme, we can keep using the revealed u_1 together with u_2. Suppose that u_1 is known to Bank before the withdrawal protocol starts owing to the previous double-spending. After the withdrawal ends, Bank cannot compute $A = (Ig_2)^s$ without knowing s, which is randomly and secretly selected by User. That is, Bank's knowledge of u_1 does not help Bank to know what the blinded message A of $m = Ig_2$ is. Consequently, Bank cannot link User with a coin that is returned after the deposit protocol. Even if Bank tries to do exhaustive search for every user's identity with coins after the deposit protocol, she cannot link User's identity

with a one-spent coin $(r_1, r_2, A, B, sign(A, B), I_S, date/time)$. It can be easily seen as following:

$$r_1 = d(u_1 s) + x_1 \bmod q$$

$$r_2 = ds + x_2 \bmod q$$

In the above simultaneous equation, there are four unknown variables s, x_1, x_2, u_1. Thus, even when Bank correctly guesses u_1, there are infinitely many solutions of the equation and she has no way of validating of her guess. However, Bank may be able to frame User by making a forged transcript by the known u_1 unless User has the unknown u_2.

4 Privacy Enhaced Digital Cash System

Our digital cash system consists of system setup, opening an account, the withdrawal protocol, the payment protocol and the deposit protocol. All the conventions are the same as that of Brands' one. As mentioned already, we augment Brands' scheme such that it has two secrets for User and the added secret is treated as the secret of the normal 3-move ZKIP technique.

- System setup: This part is the same as that of Brands' one, but domains of two hash functions H and H_0 are corrected such that it may accommodate the added functionalities.
  $H : G_q \times G_q \times G_q \times G_q \times G_q \times G_q \times G_q \times G_q \rightarrow Z_q^*$
  $H_0 : G_q \times G_q \times G_q \times G_q \times SHOP - ID \times DATE/TIME \rightarrow Z_q$
- Opening an account: Besides u_1, User has another secret u_2 and corresponding account information I_2. Also, Bank must return account information restricted by x, the Bank's secret. Only adding another secret u_2 is different from Brands'.

$$I_1 = g_1^{u_1}, I_2 = g_1^{u_2}$$

$$z_1 = (I_1 g_2)^x, z_2 = (I_2 g_2)^x$$

- The withdrawal protocol: User blinds account information by s_1 and s_2 that is related to u_1 and u_2, respectively. However, the number that will be used as a commitment for u_2 in the payment protocol is not prepared in this phase. Instead, the commitment will be generated during each payment. After the completion of the withdrawal protocol, User has a coin $(A, B, C, Sign(A, B, C))$.
  1. Bank generates $w \in_R Z_q$, and sends $a = g^w$ and $b_1 = (I_1 g_2)^w, b_2 = (I_2 g_2)^w$to User.
  2. User Generates $s_1, s_2 \in_R Z_q^*, x_1, x_2 \in_R Z_q$, and computes

     $$A = (I_1 g_2)^{s_1}, B = g_1^{x_1} g_2^{x_2}, C = (I_2 g_2)^{s_2} \text{ and } z_1' = z_1^{s_1}, z_2' = z_2^{s_2}.$$

     User generates $u, v \in_R Z_q$ and computes

     $$a' = a^u g^v, b_1' = b_1^{s_1 u} A^v, b_2' = b_2^{s_2 u} C^v$$

User then computes the challenge

$$c' = H(A, B, C, z_1', z_2', a', b_1', b_2'),$$

and sends the blinded challenge $c = c'/u \bmod q$ to Bank.

3. Bank sends the response $r = cx + w \bmod q$ to User, and debits the account of User.
4. User accepts iff

$$g^r = h^c a \text{ and } (I_1 g_2)^r = z_1^c b_1, (I_2 g_2)^r = z_2^c b_2$$

If this holds, User computes

$$r' = ru + v \bmod q$$

Proposition 1. *A and C(the blinded numbers of $I_1 g_2$ and $I_2 g_2$, respectively) and their signatures are unconditionally untraceable to any specific execution of the withdrawal protocol.*

Proof Sketch: Given any $h \neq 1$, for each pair $(A, B, C, Sign(A, B, C) = (z_1', z_2', a', b_1', b_2', r'))$ and the information that Bank gets during the execution of any protocol in which User accepts, there are exactly q possible random choices of sets (s_1, s_2, t, u, v) that could have been made by User that bring about the link. □

Proposition 2. *Under the discrete logarithm assumption, there is no way that has noneglibible probability of success for User such that she ends up with a pair $(A, B, C,\ sign(A, B, C))$ for which she knows two different representations of (A, B, C) with respect to (g_1, g_2).*

Proof Sketch: Refer to Proposition 12 in [2]. □

Proposition 2 means that User cannot double-spend a coin without changing the internal structure of the coin, that is the account information. With the propositions, we obtain the following assumption.

Assumption 1.*The withdrawal protocol is a restrictive blind signature protocol (blinded numbers are A and C) with blinding invariant functions IV_1 and IV_2 with respect to (g_1, g_2) defined by $IV_1(a_1, a_2) = IV_2(a_1, a_2)a_1/a_2 \bmod q$.*

Now, Bank's attempt to link a coin with User is always frustrated by the following proposition.

Proposition 3. *Under the discrete logarithm assumption, Bank's prior knowledge of u_1 does not help her compute the coin.*

Proof: It's trivial since s_1, s_2, x_1, x_2 are chosen secretly to make a coin $(A = (I_1 g_2)^{s_1}, B = g_1^{x_1} g_2^{x_2}, C = (I_2 g_2)^{s_2}, sign(A, B, C))$ by User. Bank cannot compute them. □

– The payment protocol: User can prove the knowledge of the representation of A, B, C with respect to $g_1 g_2$ with the payment protocol. A, B is for the knowledge proof of u_1, and C is for that of u_2. In the protocol, $D = g_1^{deal_1} g_2^{deal_2}$ corresponds to the commitment numbers of the normal 3-move ZKIP.

1. User generates $deal_1, deal_2 \in_R Z_q^*$ and computes

$$D = g_1^{deal_1} g_2^{deal_2}$$

   User sends $(A, B, C, D, Sign(A, B, C))$ to Shop.
2. If $A \neq 1$, then Shop computes and sends the challenge

$$d = H_0(A, B, C, D, I_S, date/time)$$

3. User computes and sends the responses

$$r_1 = d(u_1 s_1) + x_1 \bmod q, r_2 = ds_1 + x_2 \bmod q$$

$$r_3 = d(u_2 s_2) + deal_1 \bmod q, r_4 = ds_2 + deal_2 \bmod q$$

4. Shop accepts iff $Sign(A, B, C)$ is valid, and

$$g_1^{r_1} g_2^{r_2} = A^d B, g_1^{r_3} g_2^{r_4} = C^d D$$

Proposition 4. *During the payment protocol, User does not reveal more information than Brands' does.*

Proof Sketch: The information for the proof of knowledge of representation of A with respect to $g_1 g_2$ is the same as that for Brands'. Added information is for the the proof of knowledge of representation of C, but it is Schnorr's identification scheme and does not reveal any information on u_2. □

– The deposit protocol: First, Bank checks the validity of the coin by scrutinizing the signature and the transcript submitted by Shop. Double-spending is examined by comparing (A, C) with entries in Bank's database. If found match, Bank can extract account information by computing

$$I_1 = g_1^{(r_1 - r_1')/(r_2 - r_2')}.$$

Since $deal_1 \neq deal_1'$ and $deal_2 \neq deal_2'$, $g_1^{(r_3 - r_3')/(r_4 - r_4')}$ does not reveal I_2 even if double-spending occurs. Because Bank cannot compute any forged transcript of dealing with the double-spent coin without knowing u_2, User is computationally protected from Bank's framing attempt after the double-spending.

1. Shop sends the transcript

$$(A, B, C, D, date/time, Sign(A, B, C), r_1, r_2, r_3, r_4)$$

2. Bank accepts iff $Sign(A, B, C)$ is valid, and

$$g_1^{r_1} g_2^{r_2} = A^d B, g_1^{r_3} g_2^{r_4} = C^d D$$

and (A, C) has not been stored before. If (A, C) is already in the deposit database, Bank can find the double-spender's identity by computing

$$g_1^{(r_1 - r_1')/(r_2 - r_2')}$$

After the double-spending is detected, there are possibly two choices regarding the use of (u_1, u_2). One is that User throws away (u_1, u_2) and performs the whole procedure of account-opening again, and the other is that User keeps using (u_1, u_2). As mentioned before, the revelation of u_1 does not give any hint for Bank to link a coin with User's identity. Thus, (u_1, u_2) can be re-used without any modification. This feature is very useful, since it eliminates the need of re-opening an account and updating the database of Bank.

5 Conclusion

We presented Bank's framing problem that might arise after double-spending in digital cash systems, and propose a digital cash system that satisfies both the double-spending detection capability and the framing prevention property. The idea is that our digital cash system uses two secrets: one is for double-spending detection and the other would not be revealed even when double-spending occurs. Though we present a digital cash system based on Brands' scheme, the idea may be applied to existing and forth-coming digital cash systems.

In summary, our digital cash system has the following properties:

1. If $n(\geq 2)$-spending occurs, User cannot deny it.
2. Bank cannot frame User on $n(> 2)$-spending, while User only double-spends($n = 2$).
3. One-spent coin does not reveal User's identity.
4. After the double-spending detection, account-reopening procedure is not needed.

Though we did not present the observer-based cash protocol, it is easy to augment the proposed protocol such that it might have that pre-restriction capability on double-spending. Other than that, our technique can be easily embeded to recently published digital cash system in [5].

References

1. S. Brands, Untraceable off-Line cash in wallets with observers, Advances in Cryptology : Proceedings of Crypto '93, Lecture Notes in Computer Science, Springer-Verlag, (1994) pp. 302-318.
2. S. Brands, An Efficient Off-line Electronic Cash System Based on the representation problem, CWI Technical Report CS-R9323, (1993).

3. D. Chaum, Security Without Identification Transaction Systems to Make Big Brother Obsolete, *Communications of the ACM*, Vol. 28, No. 10, 1985, pp. 1030–1044.
4. Amos Fiat and Adi Shamir, How To Prove Yourself: Practical Solutions to Identification and Signature Problems, Advances in Cryptology : Proceedings of Crypto 86, Lecture Notes in Computer Science, Springer-Verlag, (1987) pp. 186–194.
5. Y. Frankel, Y. Tsiounis and M. Yung, Indirect discourse proof: achieving efficient fair off-line e-cash, Advances in Cryptology : Proceedings of AsiaCrypt 96, Lecture Notes in Computer Science, Springer-Verlag, (1996) pp. 286–300.
6. M. Jakobsson and M. Yung, Revokable and versatile electronic money, The 3rd ACM Conference on Computer and Communications Security, (1996) pp. 76–87.
7. S. Miyazaki and K. Sakurai, A more efficient untraceable e-cash system with partially blind signatures based on the discrete logarithm problem, Financial Cryptography, (1998).
8. J.J. Quisquater and L.S. Guillou, A Paradoxical Identity Based Signature Scheme Resulting from Zero Knowledge, Advances in Cryptology : Proceedings of EuroCrypt 88, Lecture Notes in Computer Science, Springer-Verlag, (1988) pp. 77–86.
9. C.P. Schnorr, Efficient Identification and Signatures for Smart cards, Advances in Cryptology : Proceedings of Crypto 89, Lecture Notes in Computer Science, Springer-Verlag, New York, (1989) pp. 239–251.
10. B. Schoenmakers, An efficient electronic payment system withstanding parallel attacks, Technical Report, CWI, (1995), Available in "`http://www.cwi.nl/cwi/publications/`".