A note on the n-spendable extension of Ferguson's single-term off-line coins

T. C. Lam
brianlam@tamu.edu
Department of Computer Science
Texas A & M Univeristy
College Station, TX 77843-3112

10th November 2005

Abstract

We show that an adversary can over-spend a coin $n \cdot (n+1)!$ times without being detected and identified in the n-spendable extension of Ferguson's single-term off-line coin, simply by permuting the witness messages in the three-move zero-knowledge proof payment protocol. We repair the detection scheme by adding a simple verification rule in the payment protocol. We repair the identification scheme by restricting the identity format.

1 Introduction

Single-term off-line coin is first proposed by Ferguson [1] to reduce the large number of challenge terms in *cut-and-choose* protocol. An *n-spendable extension* is also proposed [2] so that spending a coin n times does not expose the payer identity from the spent coins, but spending it more than n times does. Nevertheless, we show that an additional rule must be added to the original payment protocol in order to guarantee detection and identification of the *over-spending offender* who spends a coin more than n times.

In the *three-move zero-knowledge proof* payment protocol, the payer sends some *witness* messages to the shop, followed by a *challenge* replied by the shop, and then the payer answers the challenge by a *response*. The shop verifies the triple (*witness*, *challenge*, *response*) to decide whether or not the payment is valid. An over-spending event is detected if more than n triples with *identical* witness messages are deposited to the bank. The identity of the over-spending offender can be deciphered from $n + 1$ of these challenge-response pairs. In this note, we show that simply checking "identical" witness messages is not sufficient to detect all over-spending events, because an over-spending offender can permute the witness messages in $(n + 1)!$ different ways and still yields

the valid triple format. An over-spending offender can spend a coin n times for each permutation without being tagged over-spending, thus it can spend $n \cdot (n+1)!$ times without being identified. Among the several detection repairing schemes that we will propose shortly, the most efficient one is to regulate the prescribed order of witness message sequence used in payments. We also propose an identity format to repair the identification scheme from uncertainty arose from permutations.

2 Ferguson's scheme

In Fergusons scheme [2], an n-spendable coin is a triple (W, U, K), where

$$W = (c, a_0, a_1, ..., a_n) \tag{1}$$

is a collection of $n+2$ witness messages, U is the payer identity, and

$$K = (k_1, k_2, ..., k_n, S_0, S_1, ..., S_n) \tag{2}$$

is a collection of $2n+1$ *payer secrets* to encrypt U into the payment. Given that C and A_j are images of some one-way functions on the base numbers c and a_j, respectively, a valid coin must satisfy

$$S_0 = (C^U A_0)^{1/v}, \text{ and} \tag{3}$$

$$S_i = (C^{k_i} A_i)^{1/v}, \ 1 \leq i \leq n, \tag{4}$$

where $(v, 1/v)$ is the RSA public-private key pair of the bank. The payer makes a payment by sending W to the shop, followed by a challenge x replied by the shop. Then the payer computes the response (r, R) to the shop, where

$$r = U + \sum_{i=1}^{n} k_i x^i, \text{ and} \tag{5}$$

$$R = (S_0) \prod_{i=1}^{n} S_i^{x^i}. \tag{6}$$

The shop verifies the payment by checking that

$$R^v = C^r A_0 \prod_{i=1}^{n} A_i^{x^i}. \tag{7}$$

A payer who spends a coin more than n times can be identified from the challenge-response pairs of the spent coins, using Shamir's *secret sharing scheme* [3], which deciphers U and $k_1, k_2, ..., k_n$ by interpolating $n+1$ polynomials (5) of degree n.

3 Weakness and Repair

The weakness of the above scheme is that the payer can choose not to follow the sequences of witness messages and payer secrets suggested in (1) and (2). Instead, the payer can permute $(a_0, a_1, ..., a_n)$, $(U, k_1, k_2, ..., k_n)$, and $(S_0, S_1, ..., S_n)$ accordingly to produce valid responses that satisfy (7). For example, the payer can reorder the sequence of witness messages

$$W' = (c', a'_0, a'_1, ..., a'_n) \tag{8}$$

$$= (c, a_0, a_n, ..., a_1) \tag{9}$$

where the sub-sequence $(a_1, a_2, ..., a_n)$ in (1) is sent in the reverse order. After receiving x from the shop, the payer computes the response (r', R'), where,

$$r' = U + \sum_{i=1}^{n} k_{n-i+1} x^i, \text{ and} \tag{10}$$

$$R' = (S_0) \prod_{i=1}^{n} S_{n-i+1}^{x^i}, \tag{11}$$

which, again, reverses the orders of $(k_1, k_2, ..., k_n)$ and $(S_1, S_2, ..., S_n)$ from (5) and (6). Note that (7) is still satisfied based on the forged messages, because

$$(R')^v = (C')^{r'} A'_0 \prod_{i=1}^{n} (A'_i)^{x^i} = C^{r'} A_0 \prod_{i=1}^{n} A_{n-i+1}^{x^i} \tag{12}$$

In fact, duplicating the elements in W, such as $(a_0, a_0, ..., a_0)$, and the corresponding payer secrets, can also yield responses that satisfy (7). However, we do not consider these cases because these duplicates can easily be detected by the shop who receives the coin. In contrast, permutation of W appears normal to the shop. Different permutations of W can only be detected after they are deposited to the bank. Ferguson's scheme cannot guarantee detection of over-spending under the permutation attack because it detects identical W only. We can repair the detection scheme by taking all permutations into consideration, but then its computation cost increases greatly. An alternative is to use a signature on an agreed sequence of witness messages, blindly signed by the bank at withdrawal. The payer has to present this signature as well in the payment for the shop to verify. The agreed sequence may not be the same as the suggested sequence in (1), due to the blindness of the bank. But it does not matter because the bank signs only once, which implies that the payer must follow the same agreed sequence to produce responses in different payment instances of this coin. As a result, the bank only needs to consider identical witness messages in the detection of over-spending, without worrying other permutation derivatives. Yet, the delivery and the verification of this bank signature still impose extra communication and computation costs in payments. Another more

efficient way, without using bank signatures, is to add a system rule that, among all permutations of $(a_0, a_1, ..., a_n)$, a payer can only use the one with monotonic increasing order (or other prescribed order.) In other words, if (8) is the witness message received, then the shop accepts payment only if $a'_0 < a'_1 < ... < a'_n$. Since there is only one valid permutation for each coin, over-spending detection by checking identical witness messages is sufficient.

Note that the three methods proposed above only solve the detection problem. The identification problem is still opened because we only know that the $(n+1)$-vector solved by the interpolations of (5) is *some* permutation of $(U, k_1, k_2, ..., k_n)$, but we do not know which element in the vector is corresponding to U. We solve this problem by restricting the identity format when U is registered to the bank. For example, let the format of identity be $U = U_1||U_2||U_3$, where the symbol "||" denotes a concatenation operator between two binary strings $hash(\cdot)$ is a collision-free hash function, $U_2 = hash(U_3)$, and $U_1 = hash(U_2||U_3)$. Then the element in the vector which satisfies this identity format is U except negligible probability.

4 Related Work

The n-spendable extension of Brands' scheme [5] is proposed by Tsiounis [4] to illustrate the notion of *ordered* n-spendable coin. But its *unordered* counterpart, such as Ferguson's scheme [2], is not well addressed. The term "order" in [4] is not about the order of witness messages, but the ability to label the i^{th} instance of a coin. For example, the witness messages in Brands' extension include an $(n+1)$-vector

$$(A, B_1, B_2, ..., B_n) \tag{13}$$

and a bank signature on (13). The i^{th} instance is verified by B_i and the bank signature, so this is labeled by the $(i+1)^{st}$ element in (13).

In spite of the similar construct between (1) and (13), Brands' extension is safe from our permutation attack, because the order of its witness messages is well restricted by the bank signature, as many other ordered schemes do in the same way. In short, we suggest to pay extra attention on the order of witness messages, when similar techniques, such as the polynomial based secret sharing [3], are used to construct the n-spendable extension of an existing scheme, especially when an unordered scheme is considered.

5 Conclusion

We address the weakness on the n-spendable extension of Ferguson's scheme due to permutations of witness messages. We propose several ways to repair the over-spending detection scheme and the identification scheme. We focus our discussions on Ferguson's scheme, but our analysis is also useful to similar extensions of n-spendable coins.

References

[1] N. Ferguson, "Single Term Off-line Coins." In *Advances in Cryptology - EUROCRYPT'93* (1993), 318-328.

[2] N. Ferguson, "Extenstions of Single-term Coins." In *Advances in Cryptology - CRYPTO'93* (1993), 292-301.

[3] A. Shamir, "How to Share a Secret." *Communications ACM*, **22**(11) (1979), 612-613.

[4] Y. S. Tsiounis, *Efficient Electronic Cash: New Notations and Techniques*, PhD Thesis, Department of Computer Sciecne, Northeasten University, June 1997.

[5] S. Brands, "Untraceable Off-line Cash in Wallets with Obserers." In *Advacnces in Cryptology - CRYPTO'93* (1993), 302-318.