Centrum voor Wiskunde en Informatica
REPORT*RAPPORT*

An efficient electronic payment system withstanding parallel attacks

L.A.M. Schoenmakers

Computer Science/Department of Algorithmics and Architecture

CS-R9522 1995

Report CS-R9522
ISSN 0169-118X

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

An Efficient Electronic Payment System Withstanding Parallel Attacks

Berry Schoenmakers

CWI

P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

berry@cwi.nl

Abstract

A new blind signature protocol based on Schnorr signatures is presented that can be used in payment systems. Apart from its simplicity and efficiency, an important feature of the protocol is that it can be argued to imply a withdrawal protocol that is resistant to parallel attacks by a collusion of users. An essential property of the blind signature protocol is that the signer has complete knowledge of the receiver's secret key. Consequently, there's no protection whatsoever against framing by the bank in the proposed payment system. We therefore show how cryptographic protection against framing can be added again at the cost of introducing another trusted party, which is active during registration of the users only. A similar blind signature protocol can be based on Okamoto's variation of Schnorr's identification scheme, which is provably witness hiding.

AMS Subject Classification (1991): 94A60
CR Subject Classification (1991): D.4.6
Keywords & Phrases: Cryptography, Electronic Cash.

1. Introduction

The design of an electronic coin-based payment system comprises numerous aspects, many more than will be covered by this paper. As far as we will be concerned, a payment system involves three parties, viz. a bank, users (or payers), and shops (or payees), and three protocols, viz. withdrawal, payment, and deposit. Execution of these three protocols in sequence constitutes the life-cycle of an electronic coin, in which a coin "travels" from the bank to a user, from that user to a shop, and back again from that shop to the bank.

A natural requirement for this type of payment systems is that the payment protocol should be off-line, i.e. not involving on-line cooperation with the bank. In combination with the requirement that the users' privacy should be protected (untraceability and unlinkability), the need for a mechanism for tracing double-spenders is created. And here we arrive at the heart of the subject-matter: it should be ensured that each coin satisfies the property "once concealed, twice revealed" (courtesy [FY93]). This means that the identity of the user is guaranteed to be included in the coin somehow, and that any two payments using the same coin will reveal the identity to the bank. This basic property of privacy-protecting

Report CS-R9522
ISSN 0169-118X
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

off-line payment systems was first introduced in [CFN90] as "accountability after the fact", and we will refer to it by this name.

The hard nut to crack is therefore the design of a kind of blind signature protocol that can be used to issue such coins. In this paper we will build such a protocol starting from Schnorr's signature protocol [Sch91], which is based on the difficulty of computing discrete logarithms in groups of prime order. The goal is here to obtain a simple and efficient protocol that also makes the resulting payment system provably secure to some extent, thereby improving upon a similar system from [Bra93] (see also [Bra94a]).

The payment system is designed along the lines of [Bra93] and incorporates several ideas from that paper. Therefore, to avoid too much overlap with [Bra93], we will, for instance, not treat the case of "wallet with observers" [Cha92, CP93, CP94] as this can be added in a modular way. The payment system is designed independently of [Bra94b], however, which became available at the time of the CARDIS '94 conference (October 24–26, Lille, France) and later as technical report. A draft of the present paper [Sch94] has been circulated among members of the CAFE project in August 1994, but is independent of the results of that project. At that time a few characteristics of an improved system had already been announced by Stefan Brands, but the solution was not revealed whatsoever.

The above facts probably explain why the proposed systems are similar in structure and performance, yet with a crucial difference. The system of [Bra94b] is vulnerable to two types of so-called "parallel attacks", in which two users perform their withdrawals in parallel. These attacks enable two users to obtain a coin (or certificate) that contains neither of their identities, which they are able to spend either once (without breaking tamper-resistance) or more than once (if they are able to break tamper-resistance). In the latter case, the bank will be unable to trace double-spenders, which means that the crucial property of accountability after the fact is not achieved. The first type of attack is by Torben Pedersen [Ped94], and was in fact conceived to break an earlier version of the blind signature protocol from [Sch94]. The second type of attack is by Stefan Brands, and is reported in [Bra95a] and [Bra95b]; in the former paper he also describes an "immunization" to his attack (at the cost of a loss in performance), but this doesn't help against the first type of attack.

There are more reasons why the vulnerability to parallel attacks is undesirable, particularly when the protocols of [Bra94b] are applied to credential systems, as proposed in [Bra95b]. Needless to say that the straightforward way to prevent parallel attacks—by excluding parallel withdrawals (by different users) altogether—is undesirable. The point here is that it may be technically very difficult to ensure that parallel withdrawals are impossible, and moreover that it may degrade the capacity or flexibility of the bank too much when it is not allowed to serve customers in parallel. Using our proposed payment system there is no need to exclude parallel withdrawals by different users.

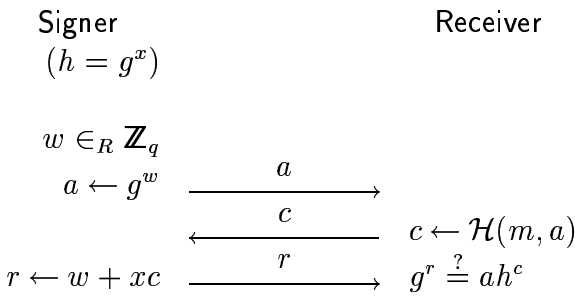

Figure 1: Issuing a Schnorr signature (c, r) on a blind message m

The contents of the paper is as follows. In Section 2 a new blind signature protocol is presented, followed by a description of a payment system based on this protocol in Section 3. In Section 4 it will be shown that the user's privacy is indeed protected for the proposed payment system. In Section 5 it will be argued informally why the payment system is secure against parallel attacks. Then, in Section 6 it is shown how cryptographic protection against framing by the bank can be achieved by introducing a trusted party; this trusted party is active during registration of new users only. Finally, in Section 7 we will summarize the differences between the proposed system and Brands' system, and it is shown that our approach is also applicable to Okamoto's variation of Schnorr's scheme [Oka93].

2. Blind signature protocol

In this section we present the blind signature protocol of [Sch94]. The protocol relies on several ideas, which we try to present kind of separately.

2.1 Schnorr signatures on a blind message

In [Sch91] Schnorr presented an efficient signature scheme based on the difficulty of the discrete log problem. In this scheme the public key of the signer consists of two large primes p, q, where $q \mid p-1$, and two generators g, h of order q in $\mathbb{Z}_p^*$. In addition an appropriate hash function $\mathcal{H}$ is part of the public key, which we assume here to map its input into (a substantial subset of) $\mathbb{Z}_q$. The secret key of the signer is the unique number $x \in \mathbb{Z}_q$ satisfying $h = g^x$ in $\mathbb{Z}_p^*$; that is, x is equal to the discrete log of h with respect to the base g. A Schnorr signature on a message m is now a pair (c, r) satisfying

$$c = \mathcal{H}(m, g^r h^{-c}).$$

Given a message m, the signer can easily compute a Schnorr signature (c, r) using the secret key x. In that case, however, the message m will be known to the signer afterwards. For our purposes, we will need to hide the message m from the signer. This can be done by generating the signature in an interactive process

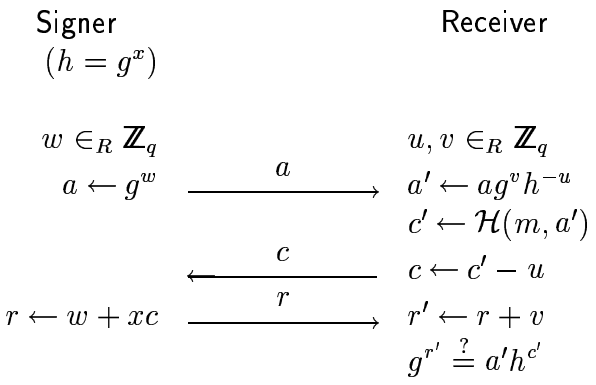

Figure 2: Blinded transcript to get a signature (c', r') on a blind message m

between signer and receiver. The signature protocol is displayed in Figure 1, and is identical to Schnorr's identification scheme [Sch91] with the modification that c is computed as a hash-value of the initial value a and the message m, instead of choosing c random.

2.2 Blinded transcripts

Although the protocol in Figure 1 has the property that the message signed remains unknown to the signer, the signer gains some information from each execution of the protocol. Indeed, if the receiver later releases the triple (m, c, r) to show that he holds a signature on a message m and this triple ever reaches the signer, then the signer will find out after all which message it signed (and in which execution of the protocol it did so) by looking at the value of c or r. To this end, the signer would keep a database of the transcripts of all executions of the signature protocol.

Therefore, as a next step towards a blind signature protocol, we show how the transcript can be blinded in a simple way. Since the value of a is fixed once the values of c and r are fixed, it suffices to blind the latter two values of each transcript. To this end, the receiver uses two random numbers u and v from $\mathbb{Z}_q$ and adds these to c and r, respectively. To get the verification relation right again, it follows that a' should be computed as indicated in Figure 2:

$$g^{r'} = g^{r+v} = ah^c g^v = ah^{c'} h^{-u} g^v = a' h^{c'}.$$

As before, the pair (c', r') is accepted precisely when $c' = \mathcal{H}(m, g^{r'} h^{-c'})$. Note that this way of blinding may be considered as incorporating a randomly chosen simulation $(g^v h^{-u}, u, v)$ of Schnorr's identification protocol in the transcript $(a = g^r h^{-c}, c, r)$ of the signature protocol.

2.3 Blinded generators

Although individual signatures are now blinded, the fact that all signatures are verified with respect to the same public key still impairs the goal we have in mind.

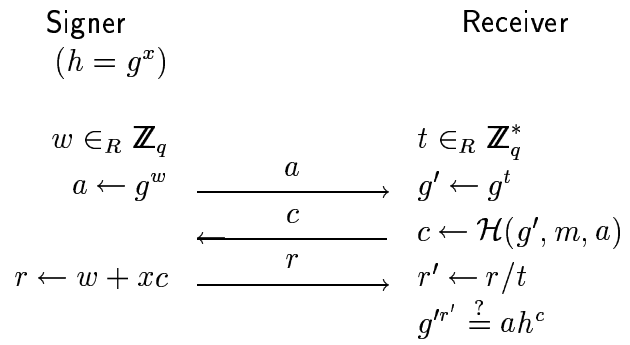

Figure 3: Blinded generator to get a signature (g', c, r') on a blind message m

What we need is a mechanism to make signatures issued to different receivers distinguishable from each other on the one hand, and a mechanism to blind these distinctions again on the other hand.

A general technique to make signatures distinguishable is to make (part of) the keys specific to a receiver. For this purpose, we consider the generators g and h. Now, in many applications of Schnorr's protocol, generator g is designated as the fixed generator (same for every receiver), and generator h is connected to the identity of the receiver. With this "standard" choice, however, we will run into problems later. The problem is that the resulting system would be vulnerable to parallel attacks.

A simple way around this problem is as follows. Instead of designating g as the fixed generator, we may as well take h as the fixed generator. This at first sight superficial interchange of the roles of g and h turns out be rather effective; as will be addressed in Section 5, the underlying reason is that receivers can control (to some extent) the exponent c of h in Schnorr's protocol, but cannot control the exponent r of g.

The next step is then to blind each use of generator g again, because different signatures by the same user would be linked when the same number g is used all the time. To this end, we let the receiver choose another random number t, say, which is used to blind g to $g' = g^t$, $t \neq 0$. The blinded generator g' is then included in the input to the hash function to ensure its validity, see Figure 3. Summarizing, we have a way to obtain blinded signatures, in which h is used as the fixed generator, and generator g is blinded to g'. A triple (g', c', r') is a valid signature on message m if

$$c' = \mathcal{H}(g', m, g'^{r'} h^{-c'}).$$

By itself this protocol is not very useful because the signatures can easily be forged by selecting g' and a as powers of h. This, however, is no problem in the way we use the protocol in payment systems; it is only of importance that it is

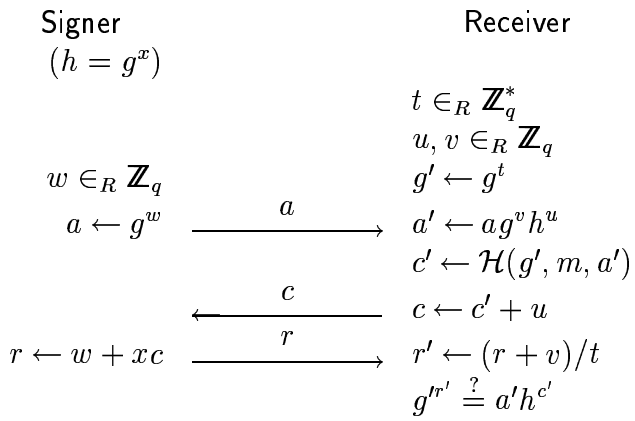

Figure 4: Protocol to get a blind signature (g', c', r') on a blind message m

guaranteed that g' is of the form g^t or of the form h^t, where the receiver knows t, $t \neq 0$.

2.4 Blind signature protocol

The combined protocol is given in Figure 4, where we have replaced u by $-u$ in comparison to Figure 2. In the next section, this protocol will be combined with a mechanism to trace double-spenders to obtain a payment system. Also, it will be guaranteed that the signature is useless if the user takes g' of the form h^t, with $t \neq 0$, which is possible as mentioned above.

3. Payment system

Apart from the set-up of the system and the registration of users, a payment system mainly comprises a withdrawal protocol, a payment protocol, and a deposit protocol. By means of these three protocols, a coin "travels" along the triangle formed by the bank, user, and shop. These protocols are displayed in Figures 5, 6, and 7, respectively.

The same mechanism as in [Bra93] is used to deal with the problem of double-spenders. In order to identify these we take $g = g_1^U g_2$, where U denotes the user's secret key corresponding to the identity. Writing $g_1 = h^{x_1}$ and $g_2 = h^{x_2}$, we then have that $x = 1/(Ux_1 + x_2)$, since $h = g^x$. This explains most of the withdrawal protocol.

In the withdrawal protocol, m is chosen as a random value for which a representation w.r.t. g_1 and g_2 is known. The payment protocol thus consists of showing the blind signature plus an interactive part, in which the user proves knowledge of a representation of g' w.r.t. g_1 and g_2. Note that the values s_1 and s_2 blind the responses r_1 and r_2 in the payment protocol. The presence of the interactive part also serves to prevent so-called replay attacks. The variable `spec` contains information that is specific to the payment. For example, it may

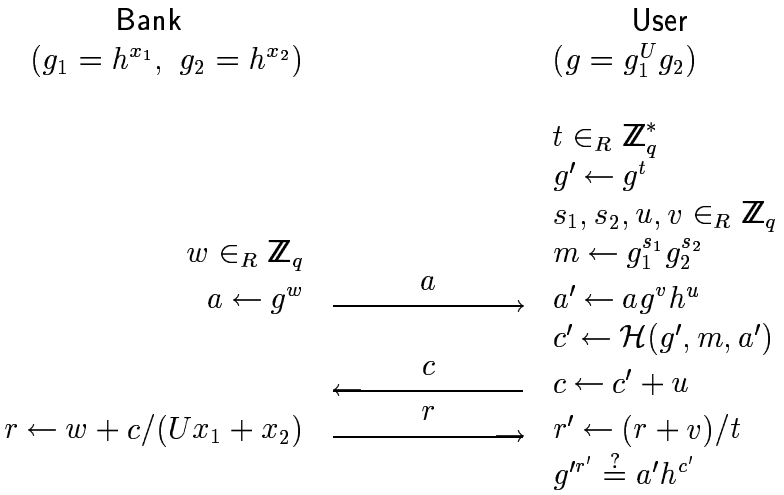

Figure 5: Withdrawal

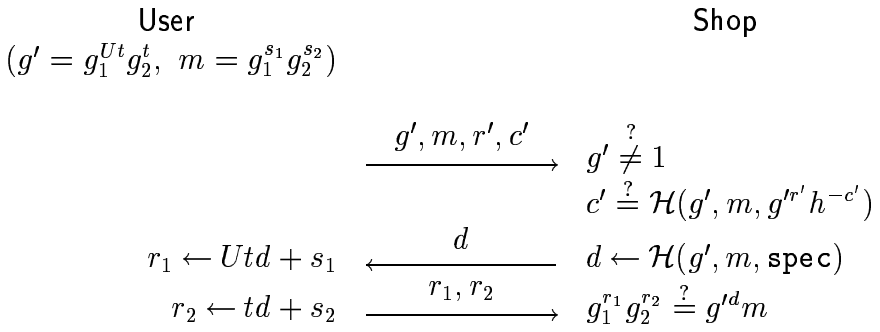

Figure 6: Payment

contain date and time of the payment, the payee's identity, and possibly some random bits to deal with the problem of "double-deposits".

Clearly, these protocols are rather simple. As for the efficiency, we note that the number of exponentiations is rather limited, and, more importantly, that the user can do all of the exponentiations required for the withdrawal protocol in preprocessing and in postprocessing. The number of exponentiations appears to be minimal, because each of the random numbers t, s_1, s_2, u, v, w is used in exactly one exponentiation (apart from the exponentiations in the verification at the end, which could be omitted). The fact that the random numbers t, s_1, s_2, u, v are indeed required and sufficient to protect the user's privacy is addressed in the next section.

There are many ways to implement these protocols in an efficient way. It is clear that, for instance, the bank doesn't really need to perform a division in the withdrawal protocol, as it can compute $1/(Ux_1 + x_2)$ once and store this for every user. Also, the bank doesn't need the value of g (which is different for each

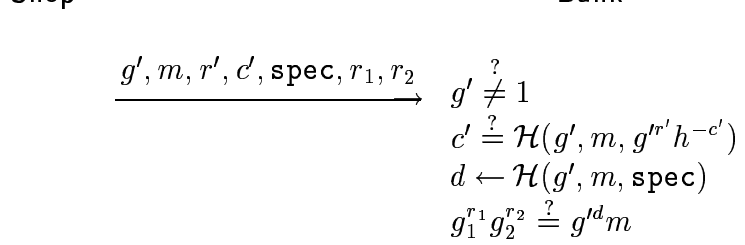

Figure 7: Deposit

user) for the withdrawal protocol: as the reader may verify, the bank may just as well compute a and r as h^w and $(w+c)/(Ux_1+x_2)$, respectively.

4. Privacy

In this section we show that the user's privacy is guaranteed for the proposed payment system. As shown in the next proposition, the bank is unable to trace any payment to the corresponding withdrawal (and hence, the corresponding user). Since payments by the same user are not linked either, it follows that both untraceability and unlinkability are guaranteed.

Proposition 1 *Accepted withdrawals and payments by the same user cannot be linked by the bank, even given the fact that the bank may try to encode linking information in the coin.*

Proof: Suppose the bank tries to identify a payment of user U. From its database, the bank extracts a, c, r from an arbitrary withdrawal of user U. The bank also picks the relevant information of an arbitrary accepted payment, say $g', m, r', c', \mathtt{spec}, r_1, r_2$. To show that there is no bias that might be used to link transactions, we show that there are unique s_1, s_2, t, u, v such that these views match. This shows that any payment can be linked to the coin considered by the bank in a unique way.

Let $g = g_1^U g_2$. We define, with $d = \mathcal{H}(g', m, \mathtt{spec})$:

$$\begin{aligned} t &= \log_g g' \\ s_1 &= r_1 - Utd \\ s_2 &= r_2 - Ut \\ u &= c - c' \\ v &= r't - r. \end{aligned}$$

Now we have to show that the views indeed match for these definitions. First note that $t \neq 0$, because $g' \neq 1$ holds for accepted payments. Further, assuming that the user correctly executed the protocols and that each payment has

been accepted, we conclude for the bank's view that $a = g^r h^{-c}$, and for the shop's view that $c' = \mathcal{H}(g', m, a')$ and $g_1^{r_1} g_2^{r_2} = g'^d m$, where $a' = g'^{r'} h^{-c'}$.

From the definitions of s_1 and s_2, it follows that all relations for the payment protocol hold. For the withdrawal protocol we need, firstly, that $g' = g^t$ and $m = g_1^{s_1} g_2^{s_2}$. The former holds on account of the definition of t. Since $g'^d m = g_1^{r_1} g_2^{r_2}$ and $g' = g^t$, the latter follows from:

$$m = g_1^{r_1} g_2^{r_2} / g'^d = g_1^{Utd+s_1} g_2^{td+s_2} g^{-td} = g_1^{s_1} g_2^{s_2},$$

using that $g = g_1^U g_2$. Secondly, we need that $a' = a g^v h^u$:

$$a' = g'^{r'} h^{-c'} = g^{tr'} h^{u-c} = g^{v+r} h^{-c} h^u = a g^v h^u,$$

using that $g' = g^t$ and $a = g^r h^{-c}$. □

We show more intuitively why each of the random variables is included. For instance, if we take $t = 1$, the views of a withdrawal and a payment match precisely when $g = g'$. Therefore, the random number t effectively blinds g. If we take $s_1 = 0$, views match just when $r_1 = Ud \log_g g'$, which can be evaluated as $g^{r_1} = g^{Ud} g'$ to avoid the computation of $\log_g g'$. Hence, s_1 blinds response r_1. Similar observations hold for the random numbers s_2, u, v, which blind the numbers r_2, c, r, respectively.

This intuitive explanation is however not sufficient. To prove that the minimal number of random variables is used, it must also be shown that any correlation between s_1, s_2, t, u, v is not allowed. For instance, if we take $u = v$, views match just when $c - c' = r' \log_g g' - r$, and the user's privacy is completely lost.

5. Security

We now argue, informally, why the proposed payment system provides more security than the system of [Bra94b]. As for forgeries by single users, the security seems to be at least as good as the security of Brands' system: the proofs in [Bra95a] can be adapted to our system. Since single-user attacks can thus be ignored at this point, the only thing that remains is that two or more users execute the withdrawal protocol in parallel. We limit our attention to a collusion between two users, as we see no reason why bigger collusions could be more successful.

Consider two users i and j, say, with corresponding generators $g_i = g_1^{U_i} g_2$ and $g_j = g_1^{U_j} g_2$. Suppose they want to obtain a valid signature $(\tilde{g}, \tilde{c}, \tilde{r})$, such that they know a representation of $\tilde{g}$ with respect to g_1 an g_2, while $\tilde{g}$ is neither a power of g_i nor a power of g_j. Then they would be able to use $(\tilde{g}, \tilde{c}, \tilde{r})$ in a payment, while the identity of neither user is involved.

In a parallel attack, the order of execution is as follows. We ignore the problem of blinding the transcripts in the attack; this can be added afterwards.

1. User i receives value a_i and user j receives value a_j.
2. The users determine $\tilde{g}$ and $\tilde{a}$.
3. The value $\tilde{c} = \mathcal{H}(\tilde{g}, \tilde{a})$ is computed.
4. The users determine c_i and c_j.
5. User i sends value c_i and user j sends value c_j.
6. User i receives value r_i and user j receives value r_j.
7. The users determine $\tilde{r}$ such that $\tilde{g}^{\tilde{r}} = \tilde{a}h^{\tilde{c}}$.

Note that it is essential that c_i and c_j are determined after $\tilde{c}$ has been computed in order that this attack can be considered as a parallel attack: otherwise, if either of these values is computed independently of $\tilde{c}$, we could have used simulations (a_i, c_i, r_i) or (a_j, c_j, r_j) instead, and the attack would be a single-user attack.

Now, restricting the attacks to "algebraic" ones, we consider the following multiplicative property[1]:

$$g_i^{r_i} = a_i h^{c_i} \ \wedge \ g_j^{r_j} = a_j h^{c_j} \quad \Rightarrow \quad g_i^{sr_i} g_j^{r_j} = a_i^s a_j h^{sc_i + c_j},$$

for any ratio s, $s \neq 0$ (for $s = 0$, nothing is gained).

To exploit this property we could take $\tilde{a} = a_i^s a_j$ and $\tilde{g} = g_i^s g_j$. But then the only constraint on c_i and c_j is that $sc_i + c_j = \tilde{c}$, hence one of these values can be considered as independent of $\tilde{c}$. So, this doesn't correspond to a parallel attack of the above form. What is more, even if we use that $g_i = g_1^{U_i} g_2$ and $g_j = g_1^{U_j} g_2$, and we set $\tilde{g} = g_1^{\tilde{U}} g_2$ for an arbitrary value $\tilde{U}$, we do not obtain an additional constraint on c_i and c_j, for the only thing that can be done is to combine the factors h^{c_i} and h^{c_j} into $h^{\tilde{c}}$: putting powers of h into $\tilde{a}$ doesn't help in this respect, and putting powers of h into $\tilde{g}$ doesn't make sense, since we do not know how to represent h in terms of g_1 and g_2.

In addition we have that the values of r_i and r_j cannot be controlled by choosing proper values for c_i and c_j, respectively. If this were the case, additional

[1] A similar multiplicative property was used in the parallel attack [Ped94] on a previous version of [Sch94]. Later, this attack turned out to be applicable to [Bra94b] as well. Using the notation of that paper, it allows two users i and j, say, to obtain a certificate $(h'_{ij}, \ldots)$, where h'_{ij} is of the form $h_i^s h_j$. A direct consequence of this attack is that the bank won't be able to trace either user in case they are able to coerce their smart cards into double-spending certificates. In other words, the crucial property of accountability after the fact is not achieved. Depending on the way this type of certificates is used, there are more undesirable consequences, even when the users are not able to break the tamper-resistance of their smart cards (or observers). For instance, the second way to deal with currency exchange rates proposed in Section 5 of [Bra94b] is completely insecure in the sense that users are able to obtain any rate they like without breaking the tamper-resistance of their smart cards.

constraints on c_i and c_j might follow. That it is indeed impossible in Schnorr's protocol (see Figure 1) to control the value of r by choosing challenge c in a "clever" way seems to be a basic property of this protocol. If this property does not hold, it follows that information about several responses for different challenges for the same a value is available; this seems to reveal too much information about the secret x, given the fact that the soundness of Schnorr's protocol relies on the property that two responses r and r' for two different challenges c and c', respectively, for the same a value reveal x.

Summarizing, we know of no way to attack the system in parallel, for in all cases that we considered there's only one constraint on c_i and c_j, which means that either user i or user j may use a simulation. We have also considered choices for $\tilde{a}$ like $\tilde{a} = a_i^{s_i} a_j^{s_j} g_i^{t_i} g_j^{t_j} g_1^{u_i} g_2^{u_j}$, and similarly for $\tilde{g}$, but none of these choices led to additional constraints on c_i and c_j.

6. Protection against framing

An essential property of the payment system in Section 3 is that the bank has complete knowledge of the user's secret key U. For this reason, the bank is able to accuse a user falsely of double-spending. This is called "framing by the bank", and the bank can do this, for instance, by constructing a coin with the user's identity and then spending it twice. To prevent the bank from framing users, a trusted party may be introduced in the following way.

We first note that it makes no sense to assume that the bank doesn't know the secrets x_1 and x_2 and the value U for each user, but merely the value of $Ux_1 + x_2$ for each user; knowledge of this value is sufficient for the withdrawal protocol. In that case, the bank wouldn't be able to do a payment, as this requires knowledge of the value of U. However, in cooperation with two users with identities U_i and U_j, say, the bank may find the secrets x_1 and x_2 anyway from the values of $U_i x_1 + x_2$ and $U_j x_1 + x_2$.

Yet, by using an additional generator g_0, say, it is possible to achieve cryptographic protection against framing in much the same way as in [Bra93]. For registration each user chooses two random numbers U_0 and U_1, and computes $g = g_0^{U_0} g_1^{U_1} g_2$. Subsequently, the user proves to the bank that he knows a representation of g/g_2 w.r.t. g_0 and g_1. In order that the bank is able to sign coins in the withdrawal protocol, the bank needs to know the value of $U' = \log_h(g/g_2)$. Since $U' = U_0 x_0 + U_1 x_1$, with $x_0 = \log_h g_0$, this value may be computed by a trusted party to which the user hands over the numbers U_0 and U_1. The trusted party must know x_0 and x_1 (which may also be known to the bank). When the bank receives the value U' from the trusted party, it checks whether $h^{U'} = g/g_2$. After registration of the users the role of the trusted party is over; disputes between the bank and a user can be solved without active involvement of the trusted party.

The extended withdrawal protocol is shown in Figure 8. The protection against

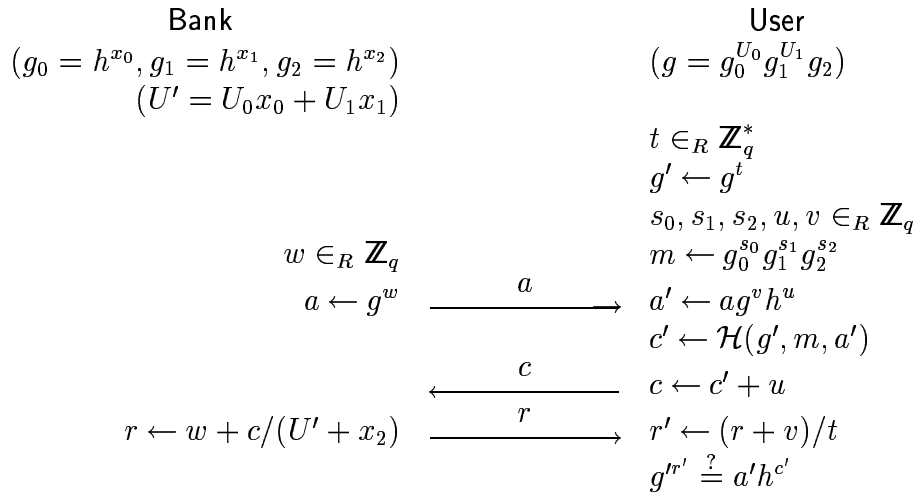

Figure 8: Extended withdrawal

framing requires only one additional exponentiation (for the computation of m). The payment protocol is adapted accordingly and extended with an additional response r_0. The verification relation for the shop becomes:

$$g_0^{r_0} g_1^{r_1} g_2^{r_2} \stackrel{?}{=} g'^d m.$$

To trace a double-spender the procedure from [Bra93] can be used.

7. Comparison with Brands' system

We have not presented our payment system for the "wallet with observer" setting, since this can be done in exactly the same way as in [Bra93, Bra94b]. Therefore, we will not compare the proposed systems in this respect. Also, we would like to stress that the blind signature protocol proposed in this paper is not intended as a building block for credential systems—in contrast with the intention of, e.g., [Bra95b].

The most important difference is the payment system of [Bra94b] is vulnerable to parallel attacks, which is considered highly undesirable (also for credential systems, of course), while our system seems to be resistant against this kind of attacks. As explained in Section 2, this is due to the fact that the roles of the generators in Schnorr's protocol are interchanged. In this way, the protection against parallel attacks is achieved at virtual no extra cost.

Another difference with Brands' system is that the generators g_1 and g_2 used in the interactive part of the payment protocol are different from the generators g and h used during withdrawal. In the system of [Bra94b] there is a generator g_0, which is used both during withdrawal and during the interactive part of a payment. This double use of g_0 is a potential disadvantage.

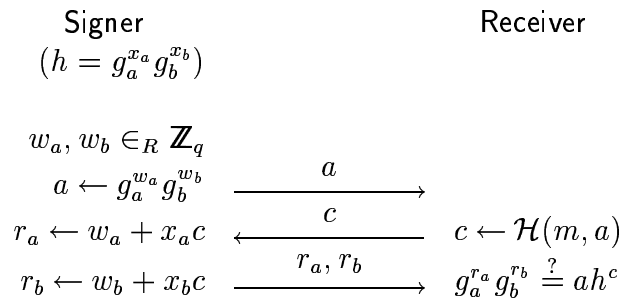

Figure 9: Issuing an Okamoto signature (c, r_a, r_b) on a blind message m

In [Bra94b, Bra95a, Bra95b] it is claimed that the approach applies to other signature schemes than Schnorr's scheme as well. The resulting payment systems are all subject to parallel attacks—as far as we can judge from these papers. Therefore, it is interesting to extend our approach to other signature schemes as well. As a first result in this direction we show how to obtain an efficient payment system that resists parallel attacks based on Okamoto's Identification Scheme 1 [Oka93]. The advantage of this scheme, which is a variation of Schnorr's scheme, is that it has been proven to be witness hiding (while this is only conjectured for Schnorr's scheme). Virtually the same approach can be followed, where g_a plays the role of g, see Figure 9. That is, $g_a = g_1^U g_2$ and x_a are made user-dependent, while g_b is a fixed generator like h, and x_b is also fixed. The corresponding blind signature protocol can be derived in the same way as in Section 2.

Acknowledgements
It is a pleasure to thank Torben Pedersen [Ped94] for his instantaneous attack on an earlier version, to thank Ronald Cramer for his observations on the consequences of the existence of parallel attacks (e.g., the problem with the exchange rates mentioned in the footnote on page 10), and to thank both for inspiring discussions about the material for this paper.

REFERENCES

[Bra93] S. Brands. An efficient off-line electronic cash system based on the representation problem. Report CS-R9323, Centrum voor Wiskunde en Informatica, March 1993.

[Bra94a] S. Brands. Untraceable off-line cash in wallet with observers. In *Advances in Cryptology—CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 302–318, Berlin, 1994. Springer-Verlag.

[Bra94b] S. Brands. Off-line cash transfer by smart cards. In V. Cordonnier and J.-J. Quisquater, editors, *Proceedings First Smart Card Research and Advanced Application Conference*, pages 101–117, 1994. Also as report CS-R9455, Centrum voor Wiskunde en Informatica.

[Bra95a] S. Brands. Off-line electronic cash based on secret-key certificates.

Report CS-R9506, Centrum voor Wiskunde en Informatica, 1995. To appear in the proceedings of the Second International Symposium of Latin American Theoretical Informatics (LATIN '95).

[Bra95b] S. Brands. Restrictive blinding of secret-key certificates. Report CS-R9509, Centrum voor Wiskunde en Informatica, 1995. To appear in the proceedings of EUROCRYPT '95.

[CFN90] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *Advances in Cryptology—CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 319–327, Berlin, 1990. Springer-Verlag.

[Cha92] D. Chaum. Achieving electronic privacy. *Scientific American*, pages 96–101, August 1992.

[CP93] D. Chaum and T. P. Pedersen. Wallet databases with observers. In *Advances in Cryptology—CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105, Berlin, 1993. Springer-Verlag.

[CP94] R. Cramer and T. P. Pedersen. Improved privacy in wallets with observers. In *Advances in Cryptology—EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 329–343, Berlin, 1994. Springer-Verlag.

[FY93] M. Franklin and M. Yung. Secure and efficient off-line digital money. In *Automata, Languages and Programming, ICALP 93*, volume 700 of *Lecture Notes in Computer Science*, pages 265–276, Berlin, 1993. Springer-Verlag.

[Oka93] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *Advances in Cryptology—CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53, Berlin, 1993. Springer-Verlag.

[Ped94] T. P. Pedersen, August 1994. Personal communication.

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

[Sch94] B. Schoenmakers. A new blind signature protocol. Internal CAFE document, Centrum voor Wiskunde en Informatica, August 8, 1994.