

# Anonymous Fingerprinting with Direct Non-repudiation

Birgit Pfitzmann and Ahmad-Reza Sadeghi

Universität des Saarlandes, Fachbereich Informatik,  
D-66123 Saarbrücken, Germany  
{pfitzmann, sadeghi}@cs.uni-sb.de

**Abstract.** Fingerprinting schemes support copyright protection by enabling the merchant of a data item to identify the original buyer of a redistributed copy. In asymmetric schemes, the merchant can also convince an arbiter of this fact. Anonymous fingerprinting schemes allow buyers to purchase digital items anonymously; however, identification is possible if they redistribute the data item.

Recently, a concrete and reasonably efficient construction based on digital coins was proposed. A disadvantage is that the accused buyer has to participate in any trial protocol to deny charges. Trials with direct non-repudiation, i.e., the merchant alone holds enough evidence to convince an arbiter, are more useful in real life. This is similar to the difference between “normal” and “undeniable” signatures.

In this paper, we present an equally efficient anonymous fingerprinting scheme with direct non-repudiation. The main technique we use, delayed verifiable encryption, is related to coin tracing in escrowed cash systems. However, there are technical differences, mainly to provide an unforgeable link to license conditions.

**Key words:** Fingerprinting, Digital Coin, Anonymity, Restrictiveness

## 1 Introduction

Protection of intellectual property in digital form has been a subject of research for many years and led to the development of various techniques. Fingerprinting schemes are an important class of these techniques. They are cryptographic methods applied to deter people from redistributing a data item by enabling the original merchant to trace a copy back to its original buyer. Dishonest buyers who redistribute the data item illegally are called traitors. The identifying information, called fingerprint, is embedded into copies of the original data item. The underlying watermarking techniques should guarantee that the embedded fingerprints are imperceptible and resistant to data manipulation as long as a traitor only uses one copy.

The first enhancement is collusion tolerance [BMP86, BS95, CKLS96], i.e., resistance even if traitors compare up to a certain number of different copies. A second addition is asymmetry [PS96a, PW97a, BM97]; here the merchant finds

an actual proof of the treachery in a redistributed copy, i.e., some data (similar to a signature “I redistributed”) that only the identified buyer could have computed. The third addition is anonymity where buyers can stay anonymous in purchasing a fingerprinted data item. Only if they redistribute the data item, the identity is revealed. We mean anonymity in the strong sense of the original definition in [PW97b], i.e., any coalition of merchants, central parties and other buyers should not be able to distinguish purchases of the remaining buyers. A weak form can easily be achieved by using any asymmetric fingerprinting scheme under a certified pseudonym instead of a real identity. In the context of fingerprinting a distinction can be made whether one fingerprints the actual data item or a key for decrypting it. The latter, introduced in [CFN94], is typically called traitor tracing. Here we deal with anonymous asymmetric data fingerprinting with collusion tolerance.<sup>1</sup>

Anonymous fingerprinting was introduced in [PW97b], but only a construction using general theorems like “every NP-language has a zero-knowledge proof system” was presented there. In [PS99], an explicit construction based on digital coins was shown. It is fairly efficient in the sense that all operations are efficient computations with modular multiplications and exponentiations; however, at least in the collusion-tolerant case, the code needed for embedding is so long that the overall system cannot be called practical.

A remaining problem with the coin-based construction is that it does not offer direct non-repudiation, i.e., in the case of a dispute, the accused buyer has to participate in the trial to deny the charges if possible. Direct non-repudiation, where the merchant alone has enough information to convince any arbiter, is more useful in real life. This is obviously true when the buyer is not reachable. But it holds even if the accused buyer has to be found in any case for reasons outside the cryptographic system, e.g., for punishment, or simply because real-life trials require the accused person to be notified. The buyer could rightly or wrongly claim to have lost the information needed for the trial or the password to it, or it could happen that a dissolved company did not leave such information to its legal successors. The difference is similar to that between normal digital signatures (direct non-repudiation) and undeniable signatures [CA90] (signer needed in trial).

In this paper we remedy this problem. Our new construction is coin-based again and equally efficient as the previous one. The new part is based on methods from coin tracing, concretely [FTY96], in particular a technique we call delayed verifiable encryption. However, on the one hand the similarity is only at the technical level: recall that we do not require a trusted third party.<sup>2</sup> On the other hand, we need a closer binding between this encryption and the coin than in coin tracing to provide an unforgeable link to the license conditions.

---

<sup>1</sup> Omitting the collusion tolerance automatically makes the schemes significantly more efficient.

<sup>2</sup> Otherwise we could use the simple solution (weak form) mentioned above.

## 2 Overview of the Model

In this section, we briefly review the model of anonymous fingerprinting proposed in [PW97b]. It involves merchants  $\mathcal{M}$ , buyers  $\mathcal{B}$ , registration centers  $\mathcal{RC}$  and arbiters  $\mathcal{A}$ . We assume that buyers can already digitally sign under their “real” identity  $ID_{\mathcal{B}}$ , i.e., that corresponding public keys  $pk_{\mathcal{B}}$  have been distributed. Before the buyers can purchase fingerprinted data items, they must register with a registration center  $\mathcal{RC}$ . Registration centers will enjoy the minimum possible trust, i.e., the most a dishonest  $\mathcal{RC}$  can do is to refuse a registration.<sup>3</sup> An arbiter  $\mathcal{A}$  represents an arbitrary honest party who should be convinced by a proof.

The four main protocols of an anonymous fingerprinting scheme are registration, fingerprinting, identification, and trial. Besides, there are three protocols for registration center key distribution, where  $\mathcal{RC}$  distributes specific parameters, data initialization, which a merchant carries out before the first sale of a specific data item, and enforced identification for the case where a merchant claims towards an arbiter that  $\mathcal{RC}$  refuses to cooperate in identification.

The main security requirements on an anonymous fingerprinting scheme are the following (for more details see [PW97b] and [PS00] the section on security):

1. An honest merchant must be able to identify a traitor and win in the corresponding trial for every illegally redistributed copy of the data item he finds, unless the collusion is larger than the tolerated limit. The identified traitor may be  $\mathcal{RC}$ , in particular if it wrongly refuses identification. Moreover, even if there are more traitors, the merchant may want to be protected from damaging his reputation by making accusations and losing the trial. Hence it is required that if identification succeeds at all, he should also win the trial.
2. No honest buyer  $\mathcal{B}$  or honest  $\mathcal{RC}$  should be found guilty by an honest arbiter, not even if there are more traitors than the limit used in the security of the merchant. In particular, as some redistributions may be legal, a proof of redistribution must be unambiguously linked to a value *text* used during fingerprinting and typically designating the terms and conditions.
3. Purchases of honest buyers should not be linkable even by a collusion of all merchants,  $\mathcal{RC}$ , and other buyers.

## 3 General Ideas of Coin-Based Fingerprinting

In this section we recall the coin-based fingerprinting from [PS99]. The basic idea for using digital cash systems with double-spender identification to construct an anonymous fingerprinting scheme is as follows: Registration corresponds to withdrawing a coin. (The “coins” only serve as a cryptographic primitive and have no

<sup>3</sup> One may ask why  $\mathcal{RC}$  is then needed, e.g., whether the merchants could not play this untrusted role themselves. However, buyers will only be anonymous among all people registered at the same registration center, and corresponding groups per merchant could be too small for meaningful anonymity.

monetary value.) During fingerprinting, the coin is given to the merchant, and in principle a first payment with this coin is made.<sup>4</sup> So far, the untraceability of the cash system should guarantee that the views of the registration center and the merchant are unlinkable. Then a second payment with the same coin is started. Now, instead of giving the buyer's response to the merchant, it is embedded in the data item. This embedding must be both secret and verifiable. After a redistribution, the merchant can extract the second response from the data item and carry out double-spender identification.

Apart from the efficient secret and verifiable embedding of the second payment response in the data, the main problem is the unambiguous link to a text describing the terms and conditions of the purchase that we required. Recall that in cash systems, double-spender identification has no such properties: the merchant simply obtains one fixed value  $i$ , called identity proof, independent of which coins were doublespent and how often. The first idea was to sign the text with a secret key whose corresponding public key  $pk_{text}$  is included in the coin. However, the registration center, as the signer of the coins, can forge coins even in such a way that they can be linked to a certain withdrawal (where the buyer may have signed the withdrawal data). Hence the real problem is how to show that the particular coin with  $pk_{text}$  is in fact one that the accused buyer has withdrawn. The solution idea in [PS99] was as follows: The buyer is able to repudiate an accusation with a wrong coin by presenting a different coin and the blinding elements that link it to the specific withdrawal from which this coin is supposed to come. For the case of Brands' payment system [Bra94], this was shown to be secure under a slightly stronger restrictiveness assumption than what would be needed for the pure payment system. Instead, we now want to give the merchant a direct proof that does not involve the buyer.

## 4 Ideas for Achieving Direct Non-repudiation

In this section we give an informal overview of the new construction with direct non-repudiation, i.e., where the merchant can convince an arbiter without participation of the accused buyer. As described in Section 3, we want to fix the actual terms and conditions  $text$  by signing them with respect to a key  $pk_{text}$  contained in the coin, and it remains to link this key unforgeably to a particular buyer after a redistribution.

The basic idea is to encrypt this coin key  $pk_{text}$  during the registration, and such that the identity proof  $i$  is the secret key needed for decryption. The buyer must sign this encryption  $enc$  under his real identity so that he is bound to it. Hence, once the merchant learns  $i$  due to a redistribution, it is possible to decrypt  $enc$  and verify which coin key  $pk_{text}$  the buyer planned to use. Note that the buyer is not needed in this step; this is essential for the direct non-repudiation.

---

<sup>4</sup> Actually the protocol is simpler, more like "zero-spendable" coins where the coin as such can be shown but any response to a challenge leads to identification. For intuitiveness, we nevertheless still call this response "second payment" in the informal part.

Each  $i$  is only used for one coin so that the link between the particular coin and the corresponding encryption  $enc$  will be clear. The next step is to force the buyer to encrypt the same  $pk_{text}$  in  $enc$  as he uses in the coin—clearly, if he can encrypt another value, his real coin will later not be attributed to him. Hence we need a kind of verifiable encryption. However, at this point there is nothing to verify the encryption against— $pk_{text}$  is deep inside the perfectly blinded coin. Here the ideas from coin tracing are applied, in particular from [FTY96] for Brands’ cash scheme, where a similar problem exists with an encryption  $enc^*$  for a trusted third party. The solution is to provide an additional specific encoding  $M$  of  $pk_{text}$  whose content is invariant under blinding. During registration (withdrawal), the buyer proves in zero-knowledge that  $enc$  and  $M$  have the same content. The registration center then blindly signs  $M$  and the buyer transforms it to  $M'$ . Later, in fingerprinting (a payment), the merchant sees the real  $pk_{text}$  used in the coin in clear. The buyer then opens the blinded encoding  $M'$ , which has the same content as  $M$ , and the merchant verifies that this content is really  $pk_{text}$ . Overall, this implies that also  $enc$  contained the correct  $pk_{text}$ .

Apart from using the identity proof  $i$  as a key instead of a trusted third party’s key, we need another modification to this idea: In [FTY96], the coin and  $M$  are blindly signed in two different signatures. If we did this, traitors could successfully attack the scheme by combining wrong pairs of coins and  $M$ ’s. Hence we need a combined blind signature on the pair, where the pair can be uniquely decomposed both in the blinded and the unblinded form. Thus, while the coins and the encodings  $M$  in [FTY96] are constructed using the same pair of generators in a discrete-logarithm setting, we use four generators and construct coins and  $M$  using different pairs. The blind signature is made on the product. (More generators in conjunction with Brands’ system have been used several times in the past, e.g., in [Bra93, BGK95, FTY98].) Restrictiveness of the blind signature scheme, together with proofs of knowledge that the values are formed over the correct generators, guarantees that a buyer cannot decompose the product in two non-corresponding ways at both sides. Here is also where the specific restrictiveness assumption comes in: The security of  $\mathcal{RC}$  relies on the correct decomposition, and  $\mathcal{RC}$  cannot trust the merchants to verify zero-knowledge proofs in fingerprinting correctly. Hence one aspect of the decomposition, (the fact that the buyer knows the discrete logarithm of  $pk_{text}$  over the correct generator), is only substantiated by a Schnorr signature towards  $\mathcal{RC}$ . In our setting, even in the random oracle model we cannot easily define and prove this Schnorr signature to be a non-interactive proof of knowledge for lack of an initial common input and hence we have to accommodate for this immediately in the restrictiveness assumption, see Section 6.1. We believe that certain statements in papers on related coin systems must be formalized in the same way.

## 5 Construction

We now present the new construction step by step. There are no surprises given the informal description in the previous section. However, as there are no mod-

ular definitions for most components we use, and as we modify some of them internally, a concrete description of the overall system seems to be the easiest way to make everything precise and to get security proofs.

For simplicity, we assume that there is only one registration center. Once and for all, a group  $G_q$  from a family of groups of prime order and generators  $g, g_1, g_2, g_3, g_4 \in_R G_q \setminus \{1\}$  are selected. For concreteness, assume that  $G_q$  is the unique subgroup of order  $q$  of the multiplicative group  $\mathbb{Z}_p^*$ , where  $p$  is another prime with  $q|(p-1)$ . Even  $\mathcal{RC}$ , who will typically make this choice, should not be able to compute discrete logarithms in  $G_q$ , and the generators must be truly random.<sup>5</sup> Hash functions  $hash$  and  $hash'$  for the underlying protocols (Brands and Schnorr signatures) must also be fixed. Finally,  $\mathcal{RC}$  generates a secret signing key  $x \in_R \mathbb{Z}_q^*$  and publishes the public key  $h \equiv g^x \pmod p$ .

### 5.1 Registration

An overview of the registration protocol is given in Figures 1 and 2. In the following, we relate the figures to the informal description and explain the correctness proof.

**1. Opening a one-time account.**  $\mathcal{B}$  chooses the “identity proof”  $i \in_R \mathbb{Z}_q^*$  randomly and secretly and computes  $h_1 \equiv g_1^i$  (with  $h_1 g_2 \neq 1$ ), the “account number” from Brands’ system, and  $h_3 \equiv g_3^i$ , which we introduced specially as a public key for ElGamal encryption.

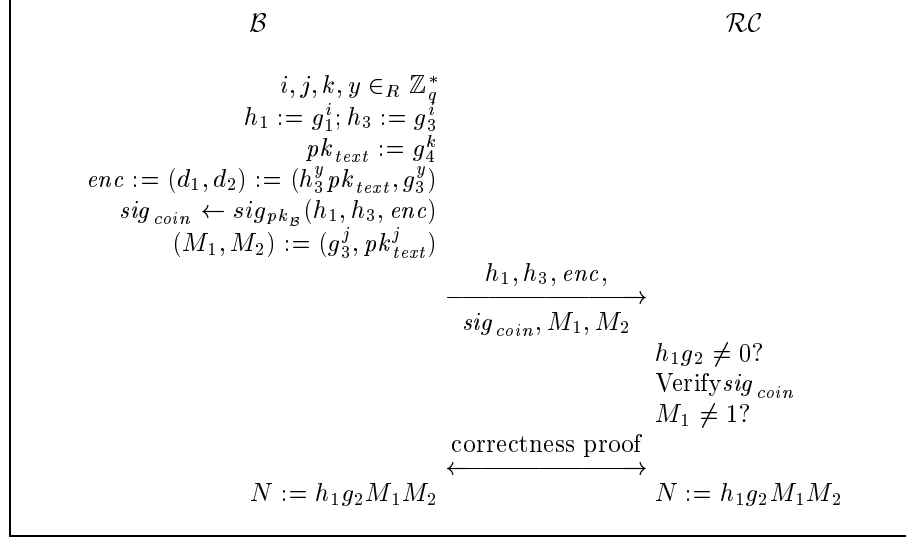
**2. Coin key and encryption.** The value  $k$ , also selected secretly and randomly by  $\mathcal{B}$ , serves as the secret coin key and  $pk_{text} \equiv g_4^k \pmod p$  as the corresponding public key.  $\mathcal{B}$  encrypts this public coin key into a ciphertext  $enc$  using  $h_3$  as the public key of ElGamal encryption. She computes a signature  $sig_{coin} \leftarrow sig_{pk_{\mathcal{B}}}(h_1, h_3, enc)$  under her normal identity and sends it to  $\mathcal{RC}$ , who verifies it. This signature later shows that  $\mathcal{B}$  is responsible for this “account” identified by the keys  $h_1$  and  $h_3$  and for the public key encrypted in  $enc$ .

**3. Encoding for delayed verifiable encryption.** The additional encoding of  $pk_{text}$  is the pair  $(M_1, M_2) = (g_3^j, pk_{text}^j)$  whose content is invariant under the following blinding operation.  $\mathcal{RC}$  will verify that  $M_1 \neq 1$ . The content is uniquely defined because  $M_1 \neq 1$  uniquely defines  $j \neq 0$ , and then  $M_2$  and  $j$  uniquely define  $pk_{text}$ .

**4. Correctness proofs.** Now  $\mathcal{B}$  sends the public values to  $\mathcal{RC}$  and gives certain correctness proofs. Intuitively, this is in particular that  $h_1$  and  $h_3$  contain the same identity proof  $i$ , and that the content of the encryption (which is uniquely defined given  $h_3$ ) equals the content of the pair  $(M_1, M_2)$  as defined above. Formally,  $\mathcal{B}$  has to give a zero-knowledge proof of knowledge of the values  $i, j, k, y$

<sup>5</sup> The randomness of the generators can be verified if  $\mathcal{RC}$  proceeds as follows: Select a non-secret string  $r$  of a certain length uniformly and randomly, e.g., by using an old random number table. Using  $r$ , generate primes  $q$  and  $p$  and elements  $e_i \in \mathbb{Z}_p^*$  deterministically. Compute the generators as  $g_i \equiv e_i^{(p-1)/q}$ . If a  $g_i$  is not a generator, repeat its choice.

such that the public values, i.e.,  $h_1, h_3, enc, M_1, M_2$  fulfill the prescribed equations.

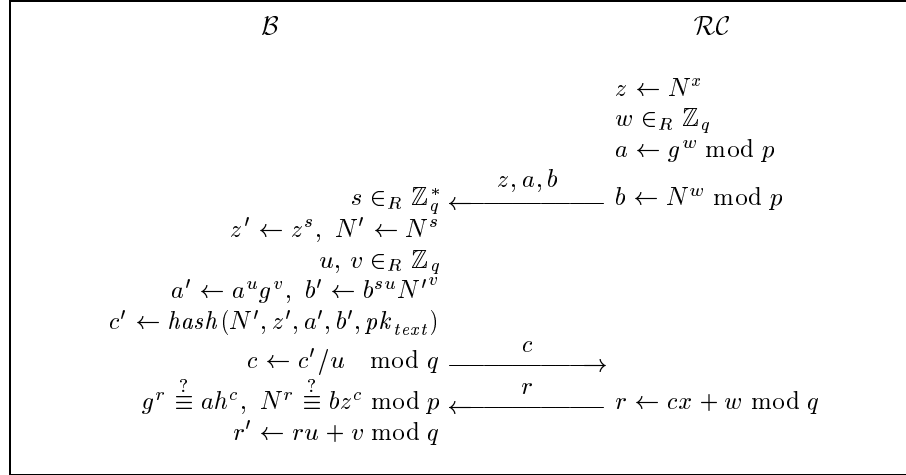


**Fig. 1.** The registration protocol before the blind signature

This can be done by using a simple protocol from [CEG88] for  $i$  and the specific “indirect discourse proof” from [FTY96] for the remaining parameters. However, there is also a general efficient technique for proving low-degree polynomial relations in exponents [Cam98], Section 3.5, which comprises this and many similar situations. The protocol from [CEG88] for showing that  $h_1$  and  $h_3$  are correct is shown in [PS00]. Exactly the same type of proof is not possible for the other values because one equation is  $M_2 = g_4^{jk}$ , where neither  $g_4^j$  nor  $g_4^k$  can be public. Here is where the techniques for polynomials come in (e.g., Camenisch uses blinded versions of the required intermediate values, e.g.,  $g_5^r g_4^k$  to get back to the linear situation.).

**5. Withdrawal.** Now  $\mathcal{RC}$  gives a blind signature on the combination of a coin and the encoding  $(M_1, M_2)$ . Let  $m \equiv g_1^i g_2 = h_1 g_2$  be the value typically signed in Brands’ scheme,  $M \equiv M_1 M_2$ , and  $N \equiv m M$ . This  $N$  is the common input to the blind signing protocol (essentially from [CP93]). In [Bra94], an additional value is included in the hashing; we use  $pk_{text}$  in that place. The resulting protocol is shown in Figure 2. As a result,  $\mathcal{B}$  obtains the “coin”  $coin' = (N', pk_{text}, \tau')$ , where  $N' \equiv (m M)^s$  and  $\tau' = (z', a', b', r')$  is called the signature on  $(N', pk_{text})$ .<sup>6</sup> We denote the blinded versions of  $m$  and  $M$  by  $m' \equiv m^s \equiv g_1^{is} g_2^s$  and  $M' \equiv M^s \equiv g_3^{s'} pk_{text}^{s'}$ , where  $s' = sj$ .

<sup>6</sup> In the sense of Section 4 this is not only the coin, but also still contains the blinded specific of  $pk_{text}$ . However, in the following, it is simpler to call this unit a coin.



**Fig. 2.** The blind signature part of the registration protocol

## 5.2 Fingerprinting

The main common input in fingerprinting is the value *text* typically used to refer to the license conditions. We assume that each *text* is fresh for both buyer and merchant in this protocol, i.e., neither of them uses a value *text* twice. This can be achieved by a number of standard techniques.

**1. Text signing and coin verification.**  $\mathcal{B}$  selects an unused coin  $coin' = (N', pk_{text}, \tau')$ . He uses the corresponding secret key  $k$  to make a Schnorr signature  $sig_{text}$  on *text* (where we include  $pk_{text}$  in the hashing) and sends  $(coin', m', M', s', sig_{text})$  to  $\mathcal{M}$ . Now  $\mathcal{M}$  first verifies the blind signature: He computes  $c' \equiv \text{hash}(N', z', a', b', pk_{text}) \bmod q$  and tests whether  $g^{r'} \equiv a'h^{c'}$  and  $N'^{r'} \equiv b'z'^{c'} \bmod p$  hold. We say that a coin is valid if and only if it passes these tests. He then verifies  $sig_{text}$  using  $pk_{text}$  from  $coin'$ .

**2. Verification of decomposition.**  $\mathcal{M}$  first verifies that  $N' \equiv m'M'$ ,  $N' \neq 1$  and  $m' \neq 1$ . Then  $\mathcal{B}$  proves to  $\mathcal{M}$  in zero-knowledge that he knows a representation of  $m'$  with respect to  $(g_1, g_2)$  and of  $pk_{text}$  with respect to  $g_4$  [CEG88].

**3. Delayed part of verifiable encryption.**  $\mathcal{M}$  verifies whether  $M' \equiv g_3^{s'} pk_{text}^{s'}$  holds. (Details why this verification is sufficient can be seen in the proof of the security of the registration center, see [PS00].)

**4. Embedding.**  $\mathcal{B}$  takes the representation  $(is, s)$  of  $m' \equiv g_1^{is} g_2^s$  as the value *emb* to be embedded secretly and verifiably in the data item. This is the identical task as in [PS99] and thus from here on we can reuse the old protocol.

For the overall security considerations later, note that in this protocol, additional commitments on  $(is, s)$  are made. These are information-theoretically hiding discrete-logarithm commitments using generators chosen by the merchant



and quadratic-residue commitments with respect to a number  $n$  chosen by the buyer specially for this embedding. The rest are zero-knowledge protocols. Finally, the buyer decrypts quadratic-residue commitments provided by the merchant with respect to the buyer's  $n$ .

### 5.3 Identification

**1. Merchant-side retrievals.**  $\mathcal{M}$  extracts a value  $emb = (r_1, r_2)$  from the redistributed data item using the same extraction algorithm (consisting of a watermarking part and a decoding part) as in [PS99]. This pair should be  $(is, s)$  with  $s \neq 0$ ; thus he sets  $s = r_2$  and  $i = r_1/r_2$ . He computes  $m' \equiv g_1^{is} g_2^s \pmod{p}$  and uses it to retrieve  $coin'$ ,  $M'$ ,  $text$  and  $sig_{text}$  from the corresponding purchase record of the given data item. If any of these steps do not succeed, he gives up. (The collusion tolerance of the underlying code may be exceeded.) Otherwise he sends to  $\mathcal{RC}$  the triple  $proof_0 = (i, text, sig_{text})$ .

**2. Registration center retrieval.** On input  $proof_0$ , the registration center searches in its registration database for a buyer who has registered the one-time account number  $h_1 \equiv g_1^i$  and retrieves the values  $(pk_{\mathcal{B}}, enc, sig_{coin})$ , where  $pk_{\mathcal{B}}$  corresponds to a real identity  $ID_{\mathcal{B}}$ .  $\mathcal{RC}$  refuses identification if it is clear from  $text$  that the redistribution was legal. Otherwise  $\mathcal{RC}$  decrypts  $enc$  using  $i$  to obtain  $pk_{text}$  and verifies that  $sig_{text}$  is a valid signature on  $text$  for this public key  $pk_{text}$  with respect to the generator  $g_4$ . If positive,  $\mathcal{RC}$  sends the retrieved values to  $\mathcal{M}$ .

**3. Merchant verification.** If  $\mathcal{M}$  gets an answer  $(pk_{\mathcal{B}}, enc, sig_{coin})$  from  $\mathcal{RC}$ , he first verifies that  $sig_{coin}$  is a valid signature with respect to  $pk_{\mathcal{B}}$  on the triple  $(h_1 \equiv g_1^i, h_3 \equiv g_3^i, enc)$ . He also verifies that  $enc$  correctly decrypts to the value  $pk_{text}$  contained in  $coin'$  with respect to the secret key  $i$  and the generator  $g_3$ . If one of these tests fails or  $\mathcal{M}$  receives no answer, he starts enforced identification.

### 5.4 Enforced Identification

If  $\mathcal{M}$  has to enforce the cooperation of  $\mathcal{RC}$ , he sends  $proof_1 = (coin', s', i, s, text, sig_{text})$  to an arbiter  $\mathcal{A}$ .  $\mathcal{A}$  verifies the validity of  $coin'$  and calls its components  $(N', pk_{text}, \tau')$  as usual. Then she verifies that  $N' \equiv m'M'$  for  $m' \equiv g_1^{is} g_2^s \pmod{p}$  and  $M' \equiv g_3^{s'} pk_{text}^{s'}$ . Finally, she verifies that  $sig_{text}$  is a valid signature on  $text$  for the public key  $pk_{text}$  with respect to the generator  $g_4$ .<sup>7</sup>

If any of these tests fails,  $\mathcal{A}$  rejects  $\mathcal{M}$ 's claim. Otherwise she sends  $proof_0 = (i, text, sig_{text})$  to  $\mathcal{RC}$  and requires values  $(pk_{\mathcal{B}}, enc, sig_{coin})$ . Then  $\mathcal{A}$  verifies them as  $\mathcal{M}$  does in Step 3 of identification.

<sup>7</sup> This is necessary for the security of  $\mathcal{RC}$  by guaranteeing that the division of  $N'$  into  $m'$  and  $M'$  is correct, even if  $\mathcal{RC}$  is supposed to identify all redistributors independent of  $text$ .

## 5.5 Trial

Now  $\mathcal{M}$  tries to convince an arbiter  $\mathcal{A}$  that  $\mathcal{B}$  redistributed the data item bought under the conditions described in *text*. The values  $pk_{\mathcal{B}}$  and *text* are common inputs. Note that in the following no participation of  $\mathcal{B}$  is required in the trial.

**1. Proof string.**  $\mathcal{M}$  sends to  $\mathcal{A}$  the proof string

$$proof = (coin', s', i, s, sig_{text}, enc, sig_{coin}).$$

**2. Verification of  $i$ .**  $\mathcal{A}$  computes  $h_1 \equiv g_1^i$  and  $h_3 \equiv g_3^i \pmod{p}$  and verifies that  $sig_{coin}$  is a valid signature on  $(h_1, h_3, enc)$  with respect to  $pk_{\mathcal{B}}$ . If yes, it means that  $i$ , the discrete logarithm of an account number  $h_1$  for which  $\mathcal{B}$  was responsible, has been recovered by  $\mathcal{M}$  and thus, as we will see,  $\mathcal{B}$  has redistributed some data item. It remains to verify the link to the terms and conditions described by *text*.

**3. Verification of  $text$ .**  $\mathcal{A}$  verifies the validity of  $coin'$  and calls its components  $(N', pk_{text}, \tau')$ . She then verifies that  $N' \equiv m' M'$  for  $m' \equiv g_1^{is} g_2^s \pmod{p}$  and  $M' = g_3^{s'} pk_{text}^{s'}$ .<sup>8</sup> She also verifies the signature  $sig_{text}$  on the disputed text with respect to  $pk_{text}$  and the generator  $g_4$ . These verifications imply that if the accused buyer owned this coin, he must have spent it in the disputed purchase on *text*. Finally,  $\mathcal{A}$  verifies that this coin belongs to  $\mathcal{B}$ : She tests whether *enc* correctly decrypts to  $pk_{text}$  if one uses  $i$  as the secret key. If all verifications are passed,  $\mathcal{A}$  finds  $\mathcal{B}$  guilty of redistribution, otherwise  $\mathcal{M}$  should be declared as the cheating party.

## 6 Security

Due to space restrictions we leave out the proofs and only highlight the security aspects which in our belief are of more theoretical importance. Hence we omit the security for buyers and merchants and sketch security for  $\mathcal{RC}$  and buyer's anonymity. For analysis of all security issues we refer the interested reader to [PS00].

### 6.1 Security for the Registration Center

The security requirement is that if the registration center is honest, an honest arbiter will never output that  $\mathcal{RC}$  is guilty.

For this, we need the restrictiveness of the underlying blind signature scheme for showing that the value  $m'$  used in fingerprinting “contains” the same value  $i$  as the original  $m$ , and also that the delayed verification of  $pk_{text}$  works. In [Bra94], Brands only works with two generators  $g_1, g_2$ , while we use four. However, in the underlying report [Bra93] the same assumptions are made and heuristically explained for any number of generators  $g_1, \dots, g_n$ , and coin systems with more than two generators have also been presented in [BGK95, FTY98]. The exact assumption we need is the following:

<sup>8</sup> The latter verification is not essential, but otherwise  $\mathcal{M}$  must include  $M'$  in *proof*.

**Assumption 1** (*Restrictiveness with Schnorr signature*). Let  $A$  be a probabilistic polynomial-time adversary that can interact with a Brands signer as in Figure 2 several times for messages  $N$  of its choice.  $A$  also has to output representations of all these messages, i.e., quadruples  $(i_1, \dots, i_4)$  such that

$$N = g_1^{i_1} \cdots g_4^{i_4}.$$

At the end,  $A$  has to output a message (coin)  $N'$  with a valid signature and a representation of  $N'$ , except that it need not show  $i'_4$ , but only values  $(h'_4, i'_4, msg, \sigma)$  such that  $N' = g_1^{i'_1} \cdots h'_4 i'_4$  and  $\sigma$  is a valid Schnorr signature on  $msg$  for the public key  $h'_4$  and the generator (with  $h'_4$  included in the hashing). We then define  $i'_4$  as  $i'_4 \log_{g_4}(h'_4)$ .

Then the probability that  $A$  fulfills all the conditions and that the vector  $(i'_1, \dots, i'_4)$  is not a scalar multiple of one of the vectors  $(i_1, \dots, i_4)$  is negligible. (The probability is taken over the random choices of the signer and  $A$ .)

**Discussion of the assumption.** In a simpler restrictiveness assumption, the adversary has to output complete representations of both the blinded and unblinded values, i.e., also  $i'_4$ . In our case, he only outputs a factor  $i'_4$  of  $i'_4$  and, instead of the other factor  $k := i'_4/i''_4$ , a Schnorr signature with respect to the corresponding public key  $h'_4 = g_4^k$ . The intuitive idea why this should be secure is that a Schnorr signature should be a non-interactive proof of the knowledge of the secret key. Such arguments are mentioned, e.g., in [Bra94] (Corollary 9) and [FTY96, FTY98]. However, really trying to prove our assumption from the simpler one, even in the random oracle model, leads to problems. First, the given situation does not fall under the most obvious way to define Schnorr signatures to be non-interactive zero-knowledge proofs in the random oracle model: One would take  $g, h$  as common inputs and an extractor, allowed to simulate the random oracle (in a way indistinguishable for the adversary) would have to extract the secret  $x$  with  $h = g^x$ . Under this definition, it is easy to prove that Schnorr signatures are proofs of knowledge. However, in our situation and many others where a non-interactive proof is needed,  $h$  is not a common input, but chosen by the adversary in the same step as the signature serving as proof. Hence as to definitions, it is not clear what  $x$  the extractor should extract—simply producing pairs  $(x, h)$  with  $h = g^x$  is trivial. The definition must therefore be made with respect to a scenario, i.e., in a joint probability space together with other variables. We can, e.g., define that the extractor must output pairs  $(x, h)$  where  $h$  has the same joint distribution with the other variables as the values  $h$  output by the adversary.

Now, if the scenario is non-interactive, one can still prove the desired theorem by using the forking lemma from [PS96b] if one includes  $h$  into the hashing in the Schnorr signature. However, in our scenario the adversary interacts with the bank as blind signer, in addition to the random oracle. This gives the same problems with exponential rewinding as in [PS96c] and [SG98], Section 2.4. It may be interesting to investigate how to modify either the proof techniques or the scheme so that some proof of this type goes through, but for the moment we had to make the stronger assumption.

In our scheme,  $\mathcal{RC}$  could only be found guilty in enforced identification, because in a trial an honest arbiter  $\mathcal{A}$  only finds either  $\mathcal{B}$  or  $\mathcal{M}$  guilty. Under Assumption 1 we can prove, as shown in [PS00] in detail, the security for an honest  $\mathcal{RC}$  with an honest arbiter.

## 6.2 Anonymity

We assume that  $\mathcal{RC}$  and  $\mathcal{M}$  collude and both may deviate from their protocols, hence we call them  $\mathcal{RC}^*$  and  $\mathcal{M}^*$ . We want to show that they learn nothing about the purchase behaviour of honest buyers, except for facts that can simply be derived from the knowledge of who registered and for what number of purchases, and at what time protocols are executed. This should even hold for the remaining purchases of a buyer if  $\mathcal{RC}^*$  and  $\mathcal{M}^*$  obtain some data items this buyer bought.

In our construction, the only information common to all registrations of a buyer is her global key pair  $(sk_{\mathcal{B}}, pk_{\mathcal{B}})$  (recall that we use each  $i$  only once). She only uses it to generate the signature  $sig_{coin}$ , and uses neither the keys nor this signature in fingerprinting. Thus other fingerprintings and possible redistributions of a buyer are statistically independent of one registration and the corresponding fingerprinting. Hence we focus on the question whether  $view_{reg}$  and  $view_{fing}$  from such a pair of corresponding protocols are linkable. For this, we let an adversary carry out two registrations and then the two corresponding fingerprintings in random order. The adversary is considered successful if it can guess with probability significantly better than  $1/2$  which views correspond to each other.

More precisely, first the global parameters are generated (the group and generators in our construction), given a security parameter  $l$ . Then the two buyers generate their global keys. Next, the registration protocol  $reg$  is run where  $\mathcal{RC}^*$  inputs the buyer's public key and the buyer  $\mathcal{B}$  her secret key. The outputs are  $\mathcal{RC}^*$ 's view and  $\mathcal{B}$ 's view  $view_{\mathcal{B}}$ . For  $\mathcal{RC}^*$ 's view we write  $(traf_{reg}, aux_i)$ , where  $traf_{reg}$  ("traffic" in slight abuse of the term) denotes the messages from  $\mathcal{B}$  to  $\mathcal{RC}^*$ , while the variables  $aux_i$  model the adversary's entire memory between protocol executions. Now a bit  $b$  is uniformly chosen; it denotes on which registration the first execution of fingerprinting is based, assuming that the registrations succeeded from the buyers' point of view. The notation for the fingerprinting protocol  $fing$  is similar to that for  $reg$ . Finally, the adversary algorithm  $A_{Link}$  outputs a guess  $b^*$  for  $b$  based on the adversary's memory, which may of course contain the traffic. The values sent by  $\mathcal{B}$  are (for simplicity we included  $pk_{\mathcal{B}}$  in  $traf_{reg}$ ):

$$\begin{aligned} traf_{reg,0} &= (pk_{\mathcal{B},0}, h_{1,0}, h_{3,0}, M_{1,0}, M_{2,0}, enc_0, sig_{coin,0}, c_0, traf_{reg,0}^{ZKP}), \\ traf_{fing,b} &= (coin'_b, m'_b, M'_b, s'_b, sig_{text,b}, traf_{embed,b}, traf_{fing,b}^{ZKP}), \end{aligned}$$

and similarly for  $traf_{reg,1}$  and  $traf_{fing,\bar{b}}$ . Here  $c_0$  is the only value sent in the withdrawal subprotocol,  $coin'_b = (N'_b, pk_{text,b}, \tau'_b)$  the coin,  $traf_{embed,b}$  the traffic from Step 4 of fingerprinting and  $traf_{reg,0}^{ZKP}$ ,  $traf_{fing,b}^{ZKP}$  that from all zero-knowledge

protocols in registration and fingerprinting. The texts to be signed may be chosen adaptively by  $\mathcal{M}^*$  in *fing*.

We can prove, as shown in detail in [PS00], that given a successful adversary as defined above, there are also successful adversaries in successive scenarios where the “buyer” sends fewer and fewer values. This finally leads to a contradiction. The anonymity of our scheme is based on the following assumption and the random oracle model for the hash function used in the blind signature protocol:

**Assumption 2** (*Strong Decisional Diffie-Hellman Assumption*). No probabilistic polynomial-time algorithm  $A_{SDDH}$ , on inputs of the form

$$(g, g^x, g^y, g^{y^{-1}}, u)$$

where  $u$  is either  $g^{xy}$  or a random group element, can distinguish the two cases with probability significantly better than  $1/2$ .

## Acknowledgment

We thank Mihir Bellare, Victor Shoup, Michael Steiner and Michael Waidner for interesting discussions.

## References

- [BGK95] Ernest Brickell, Peter Gemmell, David Kravitz: Trustee-based Tracing Extensions to Anonymous Cash and the Making of Anonymous Change; SODA 1995, ACM Press, New York 1995, 457-466.
- [BM97] Ingrid Biehl, Bernd Meyer: Protocols for Collusion-Secure Asymmetric Fingerprinting; STACS 97, LNCS 1200, Springer-Verlag, Berlin 1997, 399-412.
- [BMP86] G. R. Blakley, Catherine Meadows, George B. Purdy: Fingerprinting Long Forgiving Messages; Crypto'85, LNCS 218, Springer-Verlag, Berlin 1986, 180-189.
- [Bra93] Stefan Brands: An Efficient Off-line Electronic Cash System Based On The Representation Problem; Centrum voor Wiskunde en Informatica, Report CS-R9323, March 1993.
- [Bra94] Stefan Brands: Untraceable Offline Cash in Wallet with Observers; Crypto'93, LNCS 773, Springer-Verlag, Berlin 1994, 302-318.
- [BS95] Dan Boneh, James Shaw: Collusion-Secure Fingerprinting for Digital Data; Crypto'95, LNCS 963, Springer-Verlag, Berlin 1995, 452-465.
- [CA90] David Chaum, Hans van Antwerpen: Undeniable Signatures; Crypto'89, LNCS 435, Springer-Verlag, Berlin 1990, 212-216.
- [Cam98] Jan Camenisch: Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem; Hartung-Gorre Verlag, Konstanz 1998.
- [CEG88] David Chaum, Jan-Hendrik Evertse, Jeroen van de Graaf: An improved protocol for demonstrating possession of discrete logarithms and some generalizations; Eurocrypt'87, LNCS 304, Springer-Verlag, Berlin 1988, 127-141.

- [CFN94] Benny Chor, Amos Fiat, Moni Naor: Tracing traitors; Crypto'94, LNCS 839, Springer-Verlag, Berlin 1994, 257-270.
- [CKLS96] Ingemar Cox, Joe Kilian, Tom Leighton, Talal Shamoon: A Secure, Robust Watermark for Multimedia; Information Hiding, LNCS 1174, Springer-Verlag, Berlin 1996, 185-206.
- [CP93] David Chaum, Torben Pryds Pedersen: Wallet Databases with Observers; Crypto'92, LNCS 740, Springer-Verlag, Berlin 1993, 89-105.
- [FTY96] Yair Frankel, Yiannis Tsiounis, Moti Yung: "Indirect Discourse Proofs": Achieving Efficient Fair Off-Line E-cash; Asiacrypt'96, LNCS 1163, Springer-Verlag, Berlin 1997, 287-300.
- [FTY98] Yair Frankel, Yiannis Tsiounis, Moti Yung: Fair Off-Line e-Cash Made Easy; Asiacrypt'98, LNCS 1514, Springer-Verlag, Berlin 1998, 257-270.
- [PS96a] Birgit Pfitzmann, Matthias Schunter: Asymmetric Fingerprinting; Eurocrypt'96, LNCS 1070, Springer-Verlag, Berlin 1996, 84-95.
- [PS96b] David Pointcheval, Jacques Stern: Security proofs for signature schemes; Eurocrypt'96, LNCS 1070, Springer-Verlag, Berlin 1996, 387-398.
- [PS96c] David Pointcheval, Jacques Stern: Provably Secure Blind Signature Schemes; Asiacrypt'96, LNCS 1163, Springer-Verlag, Berlin 1996, 252-265.
- [PS99] Birgit Pfitzmann, Ahmad-Reza Sadeghi: Coin-Based Anonymous Fingerprinting; Eurocrypt'99, LNCS 1592, Springer-Verlag, Berlin 1996, 150-164.
- [PS00] Birgit Pfitzmann, Ahmad-Reza Sadeghi: Anonymous Fingerprinting with Direct Non-Repudiation; Cryptology ePrint Archive, Report 2000/032, <http://eprint.iacr.org/>, 2000.
- [PW97a] Birgit Pfitzmann, Michael Waidner: Asymmetric Fingerprinting for Larger Collusions; 4th ACM CCS, ACM Press, New York 1997, 151-160.
- [PW97b] Birgit Pfitzmann, Michael Waidner: Anonymous Fingerprinting; Eurocrypt'97, LNCS 1233, Springer-Verlag, Berlin 1997, 88-102.
- [Sch91] Claus-Peter Schnorr: Efficient Signature Generation by Smart Cards; Journal of Cryptology 4/3 (1991) 161-174.
- [SG98] Victor Shoup, Rosario Gennaro: Securing threshold cryptosystems against chosen ciphertext attack; Eurocrypt'98, LNCS 1403, Springer-Verlag, Berlin 1998, 1-16.