Cryptographically Strong Undeniable Signatures, Unconditionally Secure for the Signer

(Extended Abstract)

David Chaum[1], Eugène van Heijst[1], Birgit Pfitzmann[2]

Abstract

We present the first undeniable signature schemes where signers are unconditionally secure. In the efficient variants, the security for the recipients relies on a discrete logarithm assumption or on factoring; and in a theoretical version, on claw-free permutation pairs.

Besides, on the one hand, the efficient variants are the first practical cryptographically strong undeniable signature schemes at all. On the other hand, in many cases they are more efficient than previous signature schemes unconditionally secure for the signer.

Interesting new subprotocols are efficient collision-free hash functions based on a discrete logarithm assumption, efficient perfectly hiding commitments for elements of $\mathbb{Z}_p$ (p prime), and fairly practical perfect zero-knowledge proofs for arithmetic formulas in $\mathbb{Z}_p$ or $\mathbb{Z}_{2^\sigma}$.

1 Introduction

The signature schemes presented here combine, for the first time, two independent features that have recently been suggested as desirable for signatures in certain situations: unconditional security for the signer and "invisibility", the characteristic property of "undeniable signatures".

Each of these features will be described separately in §1.1 and §1.2, resp., together with reasons why one might want it and previous schemes realizing it.

In §1.3, we sketch the properties of our new schemes. In particular, for cases where just one of the features is of interest, we mention how the new schemes compare with the previous schemes realizing just this one feature. §1.4 lists interesting new subprotocols, §1.5 gives an overview over the rest of the paper.

1.1 Unconditional security for signers

In conventional digital signature schemes, i.e., those according to the idea in [DH], signers can be cheated with forged signatures if a cryptographic assumption, such as the hardness of factoring, turns out to be wrong. This holds even for provably secure signature schemes such as GMR [GMR]. (Recipients, however, are unconditionally secure.)

In contrast, symmetric authentication systems with unconditional, i.e., information-theoretical security for senders of messages (and for recipients, too) have existed for quite a while, e.g., [GMS, WC]. An exponentially small error probability is tolerated. (And this seems unavoidable: An attacker can always guess the complete secret information of the real sender.) Thus unconditional security in this context means that an adversary has no advantage over mere

[1] Centre for Mathematics and Computer Science (CWI), Kruislaan 413, NL-1098 SJ Amsterdam, The Netherlands

[2] Institut für Informatik, Universität Hildesheim, Samelsonplatz 1, W-3200 Hildesheim, FRG; Fax +49-5121-860475; Phone +49-5121-883-739; e-mail pfitzb@infhil.uucp (via unido.informatik.uni-dortmund.de)

J. Feigenbaum (Ed.): Advances in Cryptology - CRYPTO '91, LNCS 576, pp. 470-484, 1992.

guessing and cannot check locally whether a guess is correct. However, with symmetric authentication, disputes between a sender and a recipient cannot be solved.

Recently, unconditional security for signers has been considered with "non-undeniable" signature schemes.

This feature is interesting in practice, even if it is only combined with cryptographic security for recipients [PW2]. In particular, if two parties exchange signed messages, both were only computationally secure before. Now, if one party uses signatures unconditionally secure for the signer, and the other party conventional ones, the former party is unconditionally secure. This is particularly suitable if there is an asymmetry between the parties anyway: One would make the weaker party unconditionally secure.

For instance, one can make individuals unconditionally secure when they exchange signatures with a large organization, e.g., a bank in a payment system. The asymmetry is that the organization usually chooses the signature schemes and the security parameters. Thus it can provide for its own security, whereas many individuals may not even know what a security parameter is. The organization has no disadvantage due to the new scheme, since it had to trust a cryptographic assumption anyway; the individuals certainly have an advantage.

But even the organization may see advantages: If clients appreciate security, there may be a marketing advantage. Also, it may be easier to obtain a guarantee of legal significance for such a system, since the organization bears the whole risk. In particular, in the (hopefully) more likely case that the cryptographic assumption is not broken, the risk that courts believe dishonest clients who falsely claim that their signatures were forged should be much smaller, since such a forgery is (mathematically) impossible. Also, if any forgery ever occurs, the organization itself is sure about this and can stop the scheme or increase the security parameters, in contrast to the case where a client's signature is forged in a conventional scheme.

Previous schemes: In [WP, BPW, PW2], fail-stop signatures were introduced. They are cryptographically secure against forgeries in the sense of [GMR]. In addition, if a forgery occurs nevertheless, the signer can prove this (more precisely: the fact that the cryptograpic assumption has been broken) unconditionally (in the sense described above) to everyone, e.g., by showing the factors of a number that she was assumed not to be able to factor. In particular, if signatures become invalid once a proof of forgery has been shown, signers are unconditionally secure, and recipients cryptographically.

In [CR], a signature-like scheme where all parties are unconditionally secure was introduced.

However, so far, none of these schemes is efficient in all cases: Fail-stop signatures are efficient for one-bit messages, but their length grows linearly with the length of the message (up to a certain point, since messages can be hashed). (But note that for some important situations, protocols with one-bit messages exist [PW2].) Unconditional signatures have a complicated precomputation phase and grow linearly with the number of possible recipients.

1.2 Undeniable signatures: Invisibility

Undeniable (or perhaps rather "invisible") signatures were introduced in [CA] and further developed, e.g., in [C2, BCDP]. These are digital signatures providing more privacy: A recipient of a signature cannot show it to others without the help of the signer. If, however, the signer is forced to either deny or acknowledge a valid signature, e.g., in court, she cannot deny it.

Previous schemes: In the previous schemes, similar to conventional digital signatures, the signers' security relies on a cryptographic assumption. (Don't be confused by the fact that the verification protocols of some schemes are "perfect zero-knowledge": A forger who breaks the cryptographic assumption can compute the secret key from the public key directly.)

All the efficient schemes are based on a discrete logarithm assumption, and they are not cryptographically strong, i.e., proved to be as secure as the discrete logarithm. In particular, no security against active attacks has been shown (cf. [GMR]). There is, however, a cryptographically strong theoretical construction from any one-way function in [BCDP].

1.3 The new schemes

We present the first three undeniable signature schemes were signers are unconditionally secure. The security of the recipients is cryptographically strong. In the two efficient variants, this means provably as secure as factoring or a discrete logarithm assumption, resp. In a theoretical variant, the security of the recipients relies on arbitrary claw-free permutation pairs.

If one cares about invisibility only, but not about which of the parties is unconditionally secure: Our schemes are not quite as efficient as the most efficient previous ones. However, if one wants the remaining party's security to be at least cryptographically strong, one must use ours.

If one cares about unconditional security for signers only, but not about invisibility: In many cases, the new schemes are much more efficient than fail-stop and unconditionally secure signatures are so far. However, if one relaxes the requirements on fail-stop signatures like they are in the schemes presented here, i.e., omits the fail-stop property and allows interaction between signer and recipient, one can construct variants of those which are even more efficient [PW2, Pf].

1.4 Interesting new subprotocols

Several subprotocols may be interesting in their own right:

We present cryptographically collision-free hash functions based on a discrete logarithm assumption, which need about one multiplication per message bit only. So far, this was only possible on the factoring assumption, whereas in the discrete logarithm case, about one exponentiation per bit was needed [D1].

We also construct efficient perfectly hiding commitments for elements of $\mathbb{Z}_p$, where the unchangeability relies on a discrete logarithm assumption. The commitment is only as long as the message. So far, perfectly hiding commitments where normally made bitwise, which induces a large message expansion. The only efficient version for larger messages was based on factoring [BPW].

The commitments can be added and subtracted locally, like those of [BPW]. We present efficient inequality proofs and procedures to multiply them. This makes fairly practical perfect zero-knowledge proofs (computationally convincing) for arithmetic formulas in $\mathbb{Z}_p$ or $\mathbb{Z}_{2^\sigma}$ possible.

1.5 Overview

We first present the basic idea in an informal way (§2). We then desribe the basic parts and security proofs of the discrete logarithm scheme (§3) and the theoretical version (§4) and sketch the factoring scheme (§5). Finally, we sketch the remaining parts, which are quite similar in the three schemes (§6).

2 Basic Idea

Our scheme combines ideas of previous undeniable signatures and of fail-stop signatures.

The basic idea to achieve the invisibility characterizing undeniable signatures is that a signature can only be verified by an interactive protocol between the signer and the recipient, preferably in zero-knowledge.

The basic idea of fail-stop signatures is that, given a public key and perhaps previous signatures, each new message has many acceptable signatures s. On a cryptographic assumption, however, the signer can compute just one of them, say s^*. If the cryptographic assumption is broken and someone forges a signature, still with high probability they will not use s^*, since all acceptable signatures look equally likely to them in the information-theoretical sense. Thus the signer now knows two different signatures for one message. This counts as a proof of forgery.

In combination with invisibility, the latter idea is changed a bit: When the real signer receives the forged signature, say sf, she cannot prove that s^* and sf are two different signatures for the same message, since she cannot carry out the verification protocol for sf. Instead, she will only prove (in zero-knowledge) that her real signature is not sf. (However, this is not a proof that the cryptographic assumption has been broken, since it can be done for any value sf. Thus this scheme has no fail-stop property.)

3 The Discrete Logarithm Scheme

Most things described in §§3.1-3.3 and 3.5-3.6 can be found in more detail in [CHP].

3.1 Assumption and notation

The cryptographic assumption, needed for the security of the recipients, is that we have an infinite sequence of groups $\mathcal{G}_p$ of known prime orders p, where one can perform the group operations and choose random elements efficiently, but the discrete logarithm is hard. (The security of both signers and recipients depends on the primality of p.) This is the same assumption as in [CA, C2].

An efficient proposal from [CA] is $\mathcal{G}_p = \mathbb{Z}_q^*/\{\pm 1\}$ where $q = 2p+1$ and both p and q are prime. $\mathcal{G}_p$ can be represented by $\{1, \ldots, p\}$. Note that if the discrete logarithm in a group is hard, it is also hard in a large subgroup (i.e., of logarithmic index). Hence this is just the normal discrete logarithm assumption for $\mathbb{Z}_q^*$, restricted to primes q of the form $2p+1$. These are usually considered as particularly hard cases.

Groups $\mathcal{G}_p = \mathrm{GF}(2^n)^*$ for Mersenne-primes $p = 2^n-1$ or large subgroups on elliptic curves are also possible.

For $g = (g_1, \ldots, g_n) \in \mathcal{G}_p{}^n$ and $x = (x_1, \ldots, x_n) \in \mathbb{Z}_p{}^n$, let

$$g^x := g_1{}^{x_1} \cdot \ldots \cdot g_n{}^{x_n}.$$

(Since the order of $\mathcal{G}_p$ is p, exponents only need to be defined modulo p.)

For $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in \mathbb{Z}_p{}^n$, denote the inner product by $x * y := x_1y_1 + \ldots + x_ny_n$.

For a message $m \in \mathbb{Z}_p$, let $\underline{m}$ denote its extension to a vector $\underline{m} := (1, 1, m)$.

We will call the signer Sibyl and the recipient Rick.

3.2 System structure for one message of fixed length

To give a better idea of the new features of the scheme, we first assume that just one message of fixed length for a given recipient is to be signed. Efficient extensions to many messages of arbitrary length and more recipients, plus extended security definitions, are quite canonical and sketched in §6.

Key exchange:

0. All the participants agree on a group $\mathcal{G}_p$ of prime order p and a parameter L. (L determines that the probability of successful cheating in the following zero-knowledge proofs should be at most 2^{-L}.)

1. The recipient Rick chooses a triple
$$g = (g_1, g_2, g_3)$$
of generators of $\mathcal{G}_p$, i.e., any elements of $\mathcal{G}_p^*$, with $g_1 \neq g_2$, randomly and publishes it.

The public values p, L, and g are parameters of the following algorithms, but will be omitted for simplicity.

2. The signer Sibyl checks that $g_1 \neq 1$ and $g_1 \neq g_2$. Then she chooses a secret key
$$SK = (x_1, x_2, x_3) \in \mathbb{Z}_p^3$$
randomly and computes and publishes the public key
$$PK = pub(SK) := g^{SK} = g_1^{x_1} \cdot g_2^{x_2} \cdot g_3^{x_3}.$$

Signing: To sign a message $m \in \mathbb{Z}_p$, Sibyl forms the signature
$$sign(SK, m) := SK * \underline{m} = x_1 + x_2 + m \cdot x_3.$$

Verification: To accept a value $s \in \mathbb{Z}_p$ as a signature on m, Rick requires Sibyl to give a perfect zero-knowledge proof of knowledge for the relation corresponding to the following statement (where PK, m, and s are common inputs):
$$\text{I know } SK\text{: } PK = pub(SK) \ \wedge\ sign(SK, m) = s. \qquad (1)$$

Disavowal: To disavow a value $sf \in \mathbb{Z}_p$ as a signature on m, the signer gives a perfect zero-knowledge proof of knowledge of the statement (again, PK, m, and sf are common inputs):
$$\text{I know } SK\text{: } PK = pub(SK) \ \wedge\ sign(SK, m) \neq sf. \qquad (2)$$

The description of efficient zero-knowledge proofs for these two purposes is postponed to §3.6 and §3.7.

3.3 Security of the signer

We must prove that however a forger Felix (using all the information that he could obtain from Sibyl) forges a signature, the probability that Sibyl can disavow it is exponentially close to 1. We first show this without the information obtained from verifications and disavowals:

Lemma 1: $\forall$ PK, $m \neq m^*$, in the probability space defined by the random choice of the secret key SK: If a forger Felix knows $PK = pub(SK)$ and $s = sign(SK, m)$, then for any forged signature sf for m^*, the probability that Sibyl can disavow sf is $1 - p^{-1}$.

Proof: Because of the completeness of the disavowal protocol, any signature on m^* other than $sign(SK, m^*)$ can be disavowed with probability 1. Thus it suffices to prove for all sf:

$$\Pr(sf \neq sign(SK, m^*) \mid PK = pub(SK) \wedge s = sign(SK, m)) = 1 - p^{-1}. \qquad (3)$$

Assume $g_2 = g_1^{\alpha}$, $g_3 = g_1^{\beta}$, and $PK = g_1^{\gamma}$. (This representation is possible, since g_1 is a generator.)

Then $$PK = pub(SK) \Leftrightarrow g_1^{\gamma} = g_1^{x_1 + \alpha x_2 + \beta x_3} \Leftrightarrow \gamma = x_1 + \alpha x_2 + \beta x_3 \text{ in } \mathbb{Z}_p,$$

$$s = sign(SK, m) \Leftrightarrow s = x_1 + x_2 + m \cdot x_3,$$

and $$sf = sign(SK, m^*) \Leftrightarrow sf = x_1 + x_2 + m^* \cdot x_3.$$

The matrix of these three equations can be transformed by row operations into

$$\begin{pmatrix} 1 & \alpha & \beta \\ 0 & 1-\alpha & m-\beta \\ 0 & 0 & m^*-m \end{pmatrix}$$

We have $m \neq m^*$, and we explicitly required $g_1 \neq g_2$, i.e., $\alpha \neq 1$. Thus the rank of this matrix is 3. (Since p is prime, a rank is defined.) Hence exactly p secret keys fulfil the condition of the probability in (3), and $sf = sign(SK, m^*)$ holds for just one of them. This proves (3). □

Note that Sibyl's security just depends on $g_1 \neq g_2$, not on the randomness of the generators. Thus it does not harm her if Rick chooses them incorrectly. Also, we have considered an adaptive chosen message attack for this simple case, since "$\forall\ m, m^*$" means that Felix can choose m and m^* in any order.

Of course, Felix does not just see PK and s, but he may ask Sibyl to verify s and to disavow other signatures. Here we need one restriction: There are only p possible signatures for m^*, thus we cannot allow Felix to try them all. For example, we restrict him to $\sqrt{p}$ attempts. This does not contradict unconditional security, since it is a restriction not on Felix's computing abilities, but on the number of disavowals Sibyl or a court are willing to perform. In practice, for a realistic size of p, $\sqrt{p}$ disavowals are impossible anyway.

Theorem 1 (Sibyl's security): $\forall\ PK, m$: Assume a forger Felix knows $PK = pub(SK)$ and $s = sign(SK, m)$ and can ask Sibyl to verify s arbitrarily often and disavow up to $\sqrt{p}$ adaptively chosen other signatures, i.e., pairs $(m^*, sf) \neq (m, s)$. Then the probability that Sibyl can disavow them all is at least $1 - \sqrt{p}^{-1}$.

Proof (Sketch): Verification and disavowal are perfect zero-knowledge. Hence a verification gives Felix no information about SK at all, and a disavowal tells him at most $sign(SK, m^*) \neq sf$ for one pair (m^*, sf). By the proof of Lemma 1, this excludes exactly one secret key. Thus no strategy gives him a better chance than $\sqrt{p}$ guesses for SK. The probability that he guesses right at least once is at most $\sqrt{p}/p = \sqrt{p}^{-1}$. □

3.4 Invisibility

Invisibility of signatures, i.e., that they cannot be recognized without the help of the signer, was not defined formally in the first publications about undeniable signatures. A definition of computational invisibility (developed independently of the abstract-version of this paper, and probably earlier) ought to appear before this in the final version of [BCDP], a newer one is contained in [CBDP]. Here, however, we need perfect invisibility, corresponding to the unconditional security of the signer.

(Note that the zero-knowledge property in [C2] concerns just verification and disavowal, not the act of issuing the signature. Issuing the signature cannot be zero-knowledge, since it enables the recipient to do something that he could not have done before, i.e., show a signature that Sibyl cannot disavow. More formally: Perfect zero-knowledge (pZKP) implies that the distribution of the prover's output is independent of her secret knowledge. This is not the case with the signatures.) However, with respect to everybody except Sibyl, perfect zero-knowledge seems just what we need. The information that these outsiders have about Sibyl's secret knowledge is modelled by a probability distribution. Thus, as a basic weaker version of perfect zero-knowledge, we define (in the notation of [TW]):

Definition: Let R be a relation, and for each x with $R_x := \{y \mid (x, y) \in R\} \neq \emptyset$, let $p_x(y)$ denote a probability distribution on R_x. An interactive TM P is **perfect zero-knowledge against outsiders** (pZKO) on $(R, (p_x)_x)$ iff for all probabilistic polynomial interactive TMs V^*, there is a simulator M_{V^*} so that $M_{V^*}(x, s)$ is polynomial in $|x|$, and for all x with $R_x \neq \emptyset$ and all s, z:

$$\sum_y (p_x(y) \cdot \Pr((P(y),\ V^*(s))(x) = z)) = \Pr(M_{V^*}(x, s) = z). \qquad \blacklozenge$$

(A computational analogue of this definition can be identified in the final version of [BCDP] if one omits the active attacks by the distinguisher and notices that the probabilities there must also be taken over the key choice.)

For the most basic part of invisibility, i.e., defining that the protocol of just issuing one signature is invisible, we can directly apply this definition: P is the TM that, on input m, issues $z := sign(SK, m)$ once, R is $\{(PK, SK) \mid PK = pub(SK)\}$, and the distributions $p_{PK}(SK)$ are naturally defined by Sibyl's choice of SK.

For the particular system of §3.2, we can easily see that this definition is fulfilled, since one can simulate the signer by choosing z randomly.

If P additionally carries out perfect zero-knowledge verifications, one can show that it is still pZKO by proving that the concatenation of a pZKO and pZKPs is always pZKO. (Although intuitively clear, this is not trivial formally, but there should be no other difficulties than in similar proofs for ZKPs [O, TW].)

For adding disavowals, one may generally relax the requirement to statistical zero-knowledge against outsiders (since before each disavowal, the public information x is slightly increased by revealing $sign(SK, m_i) \neq sf_i$).

Also, for the case where the person to whom the recipient wants to show the signature has information I about y secret from the recipient, one can extend the notion to partial outsiders by computing the expected value of the probability of the outputs z based on the corresponding probability distribution $p_{(x, I)}(y)$. In the simple case considered in this §3, I can only be the little information from disavowals.

This definition, like its computational counterpart, does not consider cooperating verifiers who try to get a signature verified simultaneously, as described in [C2, DY]. (That is, practical cases covered by this model are that a cheating recipient tries to show a signature on private information to third parties who are fairly honest, but might not look away when being shown a conventional signature, or tries to sell a received signature afterwards.) Measures against cooperating verifiers exist [C2, CBDP]: Firstly, the verification protocol should not only be zero-knowledge, but start with verifier-commitments. This is the case in all previous undeniable signature schemes and excludes the practical attack described in [DY], where the verifiers just use a coin-flipping protocol to choose their challenges. Secondly, against more complicated uses of multi-party computations, one can take measures involving timing. However, they are difficult to formalize.

3.5 Security of the recipient

We first show that *pub* is cryptographically collision-free, i.e., Sibyl cannot find two secret keys fitting the same public key. A bit more generally, we prove the following lemma:

Lemma 2 (Collision-freeness of n-tuple exponentiation): On the discrete logarithm assumption of §3.1, and for fixed n: For any probabilistic polynomial-time algorithm $\mathcal{A}_n$, any polynomial Q, and sufficiently large p: The probability that $\mathcal{A}_n$, on input a random n-tuple $G_n = (g_1, ..., g_n)$ of generators of $\mathcal{G}_p$, finds a G_n-collision, i.e.,

$$X_n \neq X_n' \in \mathbb{Z}_p^n \text{ with } G_n^{X_n} = G_n^{X_n'},$$

is smaller than $1/Q(\log(p))$.

Proof (Sketch): The proof is by induction on n. The case $n = 2$ is quite easy. For $n > 2$, we assume that an algorithm $\mathcal{A}_n$ contradicts the lemma and show that the following algorithm would then contradict the lemma for $n-1$:

$\mathcal{A}_{n-1}$: On input $G_{n-1} = (g_1, ..., g_{n-1})$:

1. Choose $e_1, ..., e_{n-1}$ from $\mathbb{Z}_p^*$ and $r_1, ..., r_{n-1}$ from $\mathbb{Z}_p$ randomly, and let
$$E := (e_1, ..., e_{n-1}) \text{ and } R := (r_1, ..., r_{n-1}).$$
2. Define an n-tuple
$$G_n := (g_1^{e_1}, \ldots, g_{n-1}^{e_{n-1}}, g_1^{r_1} \cdot \ldots \cdot g_{n-1}^{r_{n-1}}).$$
(If the last component of G_n is not a generator, repeat the choice of r_{n-1} until it is.)
3. Run $\mathcal{A}_n$ on G_n and call the result C_n. If C_n is a collision (X_n, X_n'), output
$$C_{n-1} := ((e_1x_1 + r_1x_n, \ldots, e_{n-1}x_{n-1} + r_{n-1}x_n), (e_1x'_1 + r_1x'_n, \ldots, e_{n-1}x'_{n-1} + r_{n-1}x'_n)).$$

One easily sees that if C_n is a collision and the two components X_{n-1}, X_{n-1}' of C_{n-1} are different, then C_{n-1} is a collision, too. Next one shows that for fixed G_{n-1}, G_n, and C_n, the equation $X_{n-1} = X_{n-1}'$ can be true for just one of the many possible underlying choices of R. Finally, one formalizes the following idea: When $\mathcal{A}_n$ is called, it has no information about R except G_n. Thus no matter how $\mathcal{A}_n$ chooses C_n, with high probability R is not the one for which $X_{n-1} = X_{n-1}'$ for the given G_{n-1}. □

Theorem 2 (Rick's security): On the discrete logarithm assumption of §3.1, it is infeasible for Sibyl to prove an s to be a valid signature for a message m and to disavow it later.

Proof: The soundness of the interactive proofs implies that if Sibyl can prove s to be a valid signature for m and later disavow it, she can compute secret keys SK and SK^* with

$$PK = pub(SK) \wedge sign(SK, m) = s \wedge PK = pub(SK^*) \wedge sign(SK^*, m) \neq s.$$

Since *sign* is deterministic, this implies $SK \neq SK^*$, i.e., Sibyl has found a collision of the function *pub*. This contradicts Lemma 2. □

3.6 Efficient verification

A signature is verified by use of the following protocol:

Sibyl (Prover)		Rick (Verifier)
Repeat L times:		
Choose $R = (r_1, r_2, r_3) \in \mathbb{Z}_p{}^3$ randomly.		
Compute $SK' = SK + R$, $PK' = pub(SK')$,		
$PK' = pub(SK')$, and $s' = sign(SK', m)$	$\xrightarrow{PK',\, s'}$	Choose $b \in \{0,1\}$ randomly.
	$\xleftarrow{b}$	
(If b=0, open the blinding R.	$\xrightarrow{R \text{ or } SK'}$	
If b=1, prove the knowledge	if $b = 0$ or 1, resp.	Check:
for the blinded problem instance.)		If b=0: $PK' = PK \cdot g^R \wedge s' = s + R * \underline{m}$?
		If b=1: $PK' = pub(SK') \wedge s' = sign(SK', m)$?

Protocol 1 Verification of a signature s

Lemma 3 (Verification): Protocol 1 is a perfect zero-knowledge interactive proof of knowledge for the relation defined by Formula (1).

Proof (Sketch): One can easily show that this is a special case of the proof system for random self-reducible relations of [TW, Th. 4]; the self-reduction is just the addition of a random R to SK. (More systematically, one can see that for $\phi_{p,m}(SK) = (pub_p(SK), sign_p(\cdot, m))$ is a homomorphism for all p, m, and construct a similar protocol to prove the knowledge of a preimage under any efficient homomorphism.) □

One can add verifier-commitments on the challenges b like in the protocol in [Br] (also sketched in [BCLL]), in order to achieve the same robustness against cooperating verifiers as other undeniable signatures have (see §3.4), and probably also to parallelize the rounds. (One would base the commitments on g_1, and a simulator could use a cheating verifier to compute α, β, and γ from Lemma 1 in order to cheat, too.)

3.7 Efficient disavowal

The basic structure of our disavowal is similar to that in [C1]:

In an ideal version, there would be a number L' of rounds, with Rick allowed to choose one of three challenges in each round. We now consider one round: First Sibyl chooses R randomly and computes blinded values $SK' = SK + R$, $PK' = pub(SK')$, $s' = sign(SK', m)$, and $sf' = sf + R * \underline{m}$. She prepares commitments on PK', s', and sf' (cf. Fig. 1). Rick can choose among the following three challenges:

C1 Sibyl must open the two left commitments and reveal R.

C2 Sibyl must open the two right commitments and reveal SK'.

C3 Sibyl must prove inequality of the values in the lower two commitments, without opening the commitments. (If they were both opened, Rick could compute the correct signature s.)

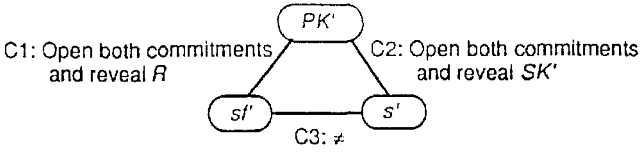
PK'
C1: Open both commitments and reveal R
C2: Open both commitments and reveal SK'
sf'
s'
C3: ≠

Fig. 1 Idea of the disavowal protocol; enclosures represent commitments

This idea can be implemented with bit commitments, similar to [C1]. However, bit commitments make the protocol far less efficient than verification. Thus we present a new variant, almost as efficient as the verification, using commitments on complete numbers instead of bits. The main problem will be the inequality proof.

We present the protocol in a generalized version, so that it can also be used in the factoring case. (A different and slightly simpler protocol, with a complete proof, can be found in [CHP], but it is slightly less efficient and could not be generalized to the factoring case.)

Definition (Sketch): A **homomorphic perfectly hiding commitment scheme for a family of abelian groups** consists of an algorithm to choose a key K, a test that is passed by all correctly chosen keys, sequences (G_K), (H_K), and (D_K) of abelian groups where all standard operations are efficiently computable, efficient homomorphisms h_K: $G_K \rightarrow H_K$ and π_K: $G_K \rightarrow D_K$, and an efficient algorithm to choose elements of $\pi_K^{-1}(a)$ randomly.

If $h_K(\alpha) = A$ and $\pi_K(\alpha) = a$, we call A a commitment to a, and say A is opened by showing α.

The two security requirements are: If K passes the test, then for each commitment, all contents are equally probable. If K is chosen correctly, then h is cryptographically collision-free. ♦

For the concrete system, we keep the commitment scheme from [CHP], which has independently been proposed in [Pe], too (but without the inequality proof):

Lemma 4 (Commitments): The following parameters define a homomorphic perfectly hiding commitment scheme for a family of abelian groups:

- K consists of a group $\mathcal{G}_p$ and two generators g_1 and g_2 of $\mathcal{G}_p$, with $g_1 \neq g_2$.
- $G_K = \mathbb{Z}_p^2$, $H_K = \mathcal{G}_p$, $D_K = \mathbb{Z}_p$.
- If $\alpha = (\alpha_1, \alpha_2)$, then $h_K(\alpha) = g_1^{\alpha_1} \cdot g_2^{\alpha_2}$ and $\pi_K(\alpha) = \alpha_1$.
- Given a, choose $\alpha := (a, \alpha_2)$ with random $\alpha_2 \in \mathbb{Z}_p$.

Proof: Obviously, h_K and π_K are homomorphisms. The content is unconditionally hidden since for each $A \in \mathcal{G}_p$, $a \in \mathbb{Z}_p$, there is exactly one α_2 with $h(a, \alpha_2) = A$. Lemma 2 implies that h is collision-free. □

The inequality proof is based on the following ideas:

1. Since contents of commitments can be subtracted, it suffices to show that the value a in a commitment A is not zero.
2. Our groups D_K are actually rings $\mathbb{Z}_p$ or $\mathbb{Z}_{2^\sigma}$. In $\mathbb{Z}_p$, we can prove $a \neq 0$ in zero-knowledge by proving that it has an inverse b, and in $\mathbb{Z}_{2^\sigma}$, by proving that there exists b such that $a \cdot b = 2^{\sigma-1}$. We unify this by saying that D_K has an element $c \neq 0$ such that $\forall\ a \neq 0\ \exists\ b$: $a \cdot b = c$, and this b can be computed efficiently.
3. Thus we finally need a protocol to prove that the product of the contents a and b of two commitments is c. We adapt an idea for shared secrets from [Be]: The factors are blinded as a'

$= a + y$, $b' = b + z$, and either a multiplication $a' \cdot b' = d$ is opened, or the correct connection between the original and the blinded multiplication must be shown. The idea is that, once the blinding factors y and z are opened, $a' \cdot b' = a \cdot b + a \cdot z + y \cdot b + y \cdot z$ is a linear equation connecting c and d and can therefore be tested on the unopened commitments. We only need to be able to multiply commitments and the values used to open them by contents. Again, we present a unification of the factoring case and the discrete logarithm case:

Definition: A semi-homomorphic perfectly hiding commitment scheme for a family of commutative rings is a homomorphic perfectly hiding commitment scheme for a family of abelian groups if additionally

1. the D_K's are commutative rings with an efficient multiplication algorithm, and
2. there are efficient "multiplications" of elements of G_K and H_K by those of D_K, which commute with the homomorphisms (i.e., $h_K(d \cdot \alpha) = d \cdot h_K(\alpha)$ and $\pi_K(d \cdot \alpha) = d \cdot \pi_K(\alpha)$). ♦

Whenever D_K is a ring $\mathbb{Z}_x$, like in our cases, we can obtain suitable multiplications as follows: If $d \in \mathbb{Z}_x$ is represented by $z \in \{0, \ldots, x-1\}$, let $d \cdot g := g + g + \ldots + g$, z times.

In order to manage with just L rounds overall, the protocol from Figure 1 and the multiplication protocol are joined more closely.

About *pub* and *sign*, we only need that they are group homomorphisms, e.g., *pub*: $G^* \rightarrow G'$, *sign*$(\bullet, m)$: $G^* \rightarrow G''$, and that the results are imbedded into the domain of the commitment scheme, e.g., by injective functions ι^*: $G' \rightarrow D_K$, ι: $G'' \rightarrow D_K$. In our discrete logarithm case, $G'' = \mathbb{Z}_p$, hence the same group $\mathcal{G}_p$ can be used here as for the signature scheme itself. For $G' = \mathcal{G}_p$, we need an efficient embedding ι^*: $\mathcal{G}_p \rightarrow \mathbb{Z}_{p'}$ for a possibly larger prime p'. However, e.g., for the first choice of $\mathcal{G}_p$ in §3.1, we can use $p' = p$ and ι only needs to change p into 0.

We obtain the following Protocol 2:

Repeat L times:

Sibyl prepares commitments: (Remember: c is a fixed nonzero value with $\forall\ a \neq 0\ \exists\ b$: $a \cdot b = c$.)

Choose R, y, z randomly.

Let $SK' = SK + R$, $PK' = pub(SK')$, $s' = sign(SK', m)$, $sf' = sf + sign(R, m)$,

$a = \iota(sf') - \iota(s')$, b such that $a \cdot b = c$, $a' = a + y$, $b' = b + z$, and $d = a' \cdot b'$.

Choose commitments P on $\iota^*(PK')$, SF' on sf', S' on s', B on b, Y on y, Z on z, D on d.

She sends the commitments to Rick, and they both compute locally:

$A = SF' - S'$, $A' = A + Y$, $B' = B + Z$

Rick can choose among 2 possibilities:

	Sibyl:	Rick tests if the commitments are opened correctly and if:
C1	Opens P, S', A', B', D, shows PK' and s', and reveals SK'.	$PK' = pub(SK')$, $s' = sign(SK', m)$ $a' \cdot b' = d$.
C2	Opens P', SF', Y, Z and shows PK' and sf', reveals R, and opens $D - z \cdot A - y \cdot B$ by showing $\chi := \delta - z \cdot \alpha - y \cdot \beta$.	$PK' = PK \cdot pub(R)$ $sf' = sf + sign(R, m)$ $\pi(\chi) = c + y \cdot z$.

Protocol 2 Efficient disavowal

Lemma 5: Protocol 2 is a perfect zero-knowledge interactive proof of knowledge that Sibyl either knows a satisfying assignment for the disavowal formula or can break the commitment scheme (cf. [BP]).

Proof: Omitted in this extended abstract. □

It is easy to see that the proof of Theorem 2 is not much changed by the occurrence of the commitment scheme in Lemma 5.

Remark: The multiplication protocol can easily be adapted to the case where c is hidden in a commitment, too (actually, this is closer to the protocol from [Be]). Thus we can compute arbitrary arithmetic terms of commitments, i.e., we can build perfect ZKPs of atomic formulas in $\mathbb{Z}_p$ or $\mathbb{Z}_{2^\sigma}$. Since logical operations can be substituted by arithmetic ones, we can generalize this to all arithmetic formulas in these rings.

4 A Theoretical Construction from Claw-free Permutation Pairs

For theoretical purposes, we sketch a construction based on an arbitrary family of claw-free permutation pairs (not necessarily with a trap-door) [GMR, D1]. This seems a sensible assumption, since it is the same one on which collision-free hash functions can be constructed, and we require a similar property from our function *pub*.

Key exchange:

0. All participants agree on a family of claw-free permutation pairs and parameters k for cryptographic security, σ for information-theoretical security, and L for the ZKPs.
1. Rick chooses a claw-free pair (d, f_0, f_1) and publishes it (cf. [GMR]). (f_0, f_1 are the permutations; d is an algorithm to choose a random element from their common domain D.) He proves Sibyl in (computational) zero-knowledge that his choice is correct.

   From the claw-free pair, a hiding function h: $\{0, 1\}^\sigma \times D \rightarrow D$ is defined as
   $$h((b_1,\ldots,b_\sigma), x) = f_{b_1}(\ldots(f_{b_\sigma}(x))\ldots). \tag{4}$$
2. Now Sibyl uses d to choose a secret key
   $$SK = (sk_1, sk_2) = ((a, x), (b, y)) \in (\{0, 1\}^\sigma \times D)^2$$
   randomly and computes and publishes the public key $PK = pub(SK) = (h(sk_1), h(sk_2)) \in D^2$.

Signing: Now we assume that $\{0, 1\}^\sigma$ is interpreted as $GF(2^\sigma)$. The message space is $GF(2^\sigma)$, and
$$sign(SK, m) = a + b{\cdot}m \text{ in } GF(2^\sigma).$$

Verification and **disavowal** are perfect zero-knowledge proofs of knowledge for the relations defined by Formulas (1) and (2).

The three required ZKPs exist on our assumption (e.g., [GMW, BCC, D1]).

Security (Sketch): We use that h is cryptographically collision-free and hides its first argument unconditionally [BPW, PW1], and that for each PK, the family $(sign(SK, \bullet))_{SK \in pub^{-1}(PK)}$ is strongly universal$_2$ [WC].

5 A Practical Scheme Based on Factoring

An inefficient scheme based on factoring can be obtained as a special case of §4 by using the special claw-free permutation pairs from [GMR]. However, to make the scheme efficient, we need special zero-knowledge proofs. We only sketch this scheme in this extended abstract.

The proof of Lemma 3 implies that one can use an analogue of Protocol 1 for efficient verification if $sign(\bullet, m)$ is a homomorphism on the same group as *pub*. Thus we must change *sign* from §4 to

$$sign(SK, m) = a + b \cdot m \text{ in } \mathbb{Z}_{2^\sigma}.$$

To prove an analogue of Lemma 1 with a probability of $1 - 2^{-\tau}$ (and thus Sibyl's security), we now restrict the message space to $\{0, \ldots, 2^{\sigma-\tau}-1\}$.

To be able to use the efficient disavowal of Protocol 2 and Lemma 5, we use the following commitments (where h_K is the same function as h in §4):

Lemma 6 (Commitments): The following parameters define a homomorphic perfectly hiding commitment scheme for a family of abelian groups:

- K should be a Blum-integer n. (This defines a GMR claw-free pair.) The test checks $n \equiv 5$ mod 8 and $n = p^s \cdot q^t$ with $p \equiv 3$ mod 4 for odd s, t, using the efficient proof-system from [GP].
- $D_n = \mathbb{Z}_{2^\sigma}$, $H_n = \pm \mathrm{QR}_n / \{\pm 1\}$, $G_n = \mathbb{Z}_{2^\sigma} \times H_n$ with an operation $\circ$ defined by
  $$(a, x) \circ (b, y) := ((a + b) \bmod 2^\sigma, |x \cdot y \cdot 4^{(a+b) \operatorname{div} 2^\sigma}|).$$
- If $\alpha = (a, x)$, then $h_K(\alpha) = \pm (4^a \cdot x^{2^\sigma})$ and $\pi_K(\alpha) = a$.
- Given a, choose $\alpha := (a, x)$ with random $x \in H_n$.

Proof: The main parts follow from [BPW]. It only remains to show that the GMR claw-free pairs are still permutations when n is not a Blum-integer, but of the form that Sibyl checks. This is quite easy. □

6 Efficient Extension to Many Long Messages

Everything in this section is only sketched in this extended abstract.

Definitions: First, of course, the definition of the security of the signer (in Theorem 1) must be extended to more than one message. This means introducing the possibility for *sign* to need memory, and a real adaptive chosen message attack, i.e., an additional "$\forall\ m_1, m_2 \ldots$: assume Sibyl signed $m_1, m_2 \ldots$ in this order ...". (Thus an active attack by an unrestricted attacker is easier to formalize than a normal one, see [GMR].)

The security of the recipient can still be defined as in Theorem 2.

In the definition of invisibility, one must include that both the recipient and the (partial) outsider to whom the signature is shown can have received more signatures. (For the computational variant, one can see this in detail in the future versions of [BCDP]. An additional complication is that our schemes are not memory-less; however, like in all "perfect" definitions, we need not model the outsider explicitly as a distinguisher, but only in the "$\forall$" over the public and private information (x, I) that he can obtain.)

Prekeys: If there are many participants, of course each recipient Rick publishes just one triple g, which can then be used by all signers when signing a message for Rick. The participants can also jointly choose one triple g whose randomness they all trust. In the discrete logarithm case, they just need a coin-flipping protocol; this is feasible. (In this case, one might even let a center make the choice alone, since no way of choosing a trap-door is known yet.) If participants disrupt, the computation is repeated without them. Although a bias results, security can still be proved for g's chosen this way. In the following, we consider the case of one g only.

Tree-authentication: As usual, one can extend the scheme to many signatures, without augmenting the public key, by two versions of tree-authentication. First, the previous public keys can be used as leaves of a hash tree [M1]. The new public key is just the root. (However, for invisibility, the signer must tell the recipient the leaves of the tree, and must use the leaves in random order.) A collision for the hash function *hash* used counts as disavowal, and for the recipient's security, *hash* must be cryptographically collision-free. There are such functions based on claw-free permutation pairs, and an efficient one based on factoring [D1]. For an efficient *hash* based on the discrete logarithm, we can use Lemma 2 directly, since we only need to hash messages of fixed length.

Generating the complete secret key in advance is the most efficient possibility; however, if one wants the scheme to go on "polynomially forever", one can use some signatures to sign new "public" keys in a tree-like fashion [M2, NY].

Message hashing: Similarly, long messages can be hashed before signing. For this, we can use the general construction of computationally collision-free hash functions *hash** for messages of arbitrary length from hash functions *hash* for messages of fixed length from [D2], starting with *hash* from the previous paragraph.

Hence, public keys and signatures are as short as in conventional signature schemes, such as GMR. The information exchanged during verification or disavowal is about L times the length of a signature.

Acknowledgements

We are happy to thank *Manfred Böttger* for initializing this cooperation and for saving us a lot of time by quickly spotting a principle problem in an early version of this scheme, *Matthijs Coster* for his hospitality, *Ivan Damgård, Eiji Okamoto, Torben Pedersen,* and *Andreas Pfitzmann* for helpful discussions, and *Michael Waidner* for helpful discussions and tedious proof-reading.

References

(All references can, if nowhere else, be obtained from the third author.)

[BCC] Gilles Brassard, David Chaum, Claude Crépeau: Minimum Disclosure Proofs of Knowledge; Journal of Computer and System Sciences 37 (1988) 156-189.

[BCDP] Joan Boyar, David Chaum, Ivan Damgård, Torben Pedersen: Convertible Undeniable Signatures; Crypto '90, Abstracts, 195-208.

[BCLL] Gilles Brassard, Claude Crépeau, Sophie Laplante, Christian Léger: Computationally Convincing Proofs of Knowledge; STACS '91, LNCS 480, Springer-Verlag, Berlin 1991, 251-262.

[Be] Donald Beaver: Multiparty Protocols Tolerating Half Faulty Processors; Crypto '89, LNCS 435, Springer-Verlag, Berlin 1990, 560-572.

[BP] Joan Boyar, René Peralta: On the concrete complexity of zero-knowledge proofs; Crypto '89, LNCS 435, Springer-Verlag, Heidelberg 1990, 507-525.

[BPW] Gerrit Bleumer, Birgit Pfitzmann, Michael Waidner: A Remark on a Signature Scheme where Forgery can be Proved; Eurocrypt '90, LNCS 473, Springer-Verlag, Berlin 1991, 441-445.

[Br] Gilles Brassard: Efficient constant-round perfect zero-knowledge; Département d'informatique et de R.O., Université de Montréal, C.P. 6128, Succ. "A", Montréal, Québec Canada H3C 3J7, Dec. 1990. (Manuscript available from the author.)

[C1] David Chaum: Zero-Knowledge Undeniable Signatures; Eurocrypt '90, Abstracts, Århus 1990, 419-426.

[C2] David Chaum: Zero-Knowledge Undeniable Signatures; Eurocrypt '90, LNCS 473, Springer-Verlag, Berlin 1991, 458-464.

[CA] David Chaum, Hans van Antwerpen: Undeniable signatures; Crypto '89, LNCS 435, Springer-Verlag, Heidelberg 1990, 212-216.

[CBDP] David Chaum, Joan Boyar, Ivan Damgård, Torben Pedersen: Undeniable Signatures: Applications and Theory; July 1, 1991. (Manuscript available from Ivan Damgård.)

[CHP] David Chaum, Eugène van Heijst, Birgit Pfitzmann: Cryptographically Strong Undeniable Signatures, Unconditionally Secure for the Signer; Fakultät f. Informatik, Univ. Karlsruhe, Internal Report 1/91, February 1991.

[CR] David Chaum, Sandra Roijakkers: Unconditionally secure digital signatures; Crypto '90 Abstracts, 209-217.

[D1] Ivan Damgård: Collision free hash functions and public key signature schemes; Eurocrypt '87, LNCS 304, Springer-Verlag, Berlin 1988, 203-216.

[D2] Ivan Damgård: A design principle for hash functions; Crypto '89, LNCS 435, Springer-Verlag, Heidelberg 1990, 416-427.

[DH] Whitfield Diffie, Martin Hellman: New Directions in Cryptography; IEEE Transactions on Information Theory 22/6 (1976) 644-654.

[DY] Yvo Desmedt, Moti Yung: Weaknesses of Undeniable Signature Schemes; Eurocrypt '91, Brighton, 8-11 April 1991, Abstracts, 111-116.

[GMR] Shafi Goldwasser, Silvio Micali, Ronald L. Rivest: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks; SIAM J. Comput. 17/2 (1988) 281-308.

[GMS] E. N. Gilbert, F. J. Mac Williams, N. J. A. Sloane: Codes which detect deception; The Bell System Technical Journal 53/3 (1974) 405-424.

[GMW] Oded Goldreich, Silvio Micali, Avi Wigderson: Proofs that Yield Nothing But their Validity and a Methodology of Cryptographic Protocol Design; 27th FOCS, IEEE Computer Society, 1986, 174-187.

[GP] Jeroen van de Graaf, René Peralta: A simple and secure way to show the validity of your public key; Crypto '87, LNCS 293, Springer-Verlag, Berlin 1988, 128-134.

[M1] Ralph Merkle: Protocols for Public Key Cryptosystems; Proceedings of the 1980 Symposium on Security and Privacy, April 14-16, 1980 Oakland, California, 122-134.

[M2] Ralph Merkle: A digital signature based on a conventional encryption function; Crypto '87, LNCS 293, Springer-Verlag, Berlin 1988, 369-378.

[NY] Moni Naor, Moti Yung: Universal One-way Hash Functions and their Cryptographic Applications; 21st STOC, ACM, New York 1989, 33-43.

[O] Yair Oren: On the cunning power of cheating verifiers: some observations about zero-knowledge proofs; 28th FOCS, IEEE Computer Society, 1987, 462-471.

[Pe] Torben Pedersen: Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing; Crypto '91, Santa Barbara, CA, 11.-15. August 1991, Abstracts 3.12-3.17.

[Pf] Birgit Pfitzmann: Fail-stop Signatures: Making Signers Unconditionally Secure; invited for Compsec, London 30.10.-1.11.1991, Proc. to be published by Elsevier.

[PW1] Birgit Pfitzmann, Michael Waidner: Formal Aspects of Fail-stop Signatures; Fakultät f. Informatik, Univ. Karlsruhe, Internal Report 22/90, 1990.

[PW2] Birgit Pfitzmann, Michael Waidner: Fail-stop Signatures and their Application; Securicom 91, Paris 1991, 145-160.

[TW] Martin Tompa, Heather Woll: Random self-reducibility and zero knowledge proofs of possession of information; 28th FOCS, IEEE Computer Society, 1987, 472-482.

[WC] Mark Wegman, Lawrence Carter: New Hash Functions and Their Use in Authentication and Set Equality"; Journal of Computer and System Sciences 22 (1981) 265-279.

[WP] Michael Waidner, Birgit Pfitzmann: Unconditional Sender and Recipient Untraceability in spite of Active Attacks – Some Remarks; Fakultät f. Informatik, Univ. Karlsruhe, Interner Bericht 5/89, 1989.