Efficient Blind Signatures without Random Oracles

Jan Camenisch[†] Maciej Koprowski[‡,⋆] Bodgan Warinschi[♯]

[†] IBM Research, Zurich Research Laboratory, CH–8803 Rüschlikon, jca@zurich.ibm.com
[‡] Intel Technology Poland, PL–80-298 Gdansk, maciejx.koprowski@intel.com
[♯]Computer Science Dept., UC Santa Cruz, Santa Cruz, CA 95064, USA, bogdan@soe.ucsc.edu

Abstract. The only known blind signature scheme that is secure in the standard model [20] is based on general results about multi-party computation, and thus it is extremely inefficient. The main result of this paper is the first provably secure blind signature scheme which is also efficient. We develop our construction as follows. In the first step, which is a significant result on its own, we devise and prove the security of a new variant for the Cramer-Shoup-Fischlin signature scheme. We are able to show that for generating signatures, instead of using randomly chosen *prime* exponents one can securely use randomly chosen *odd integer* exponents which significantly simplifies the signature generating process. We obtain our blind signing function as a secure and efficient two-party computation that cleverly exploits its algebraic properties and those of the Paillier encryption scheme. The security of the resulting signing protocol relies on the Strong RSA assumption and the hardness of decisional composite residuosity; we stress that it does not rely on the existence of random oracles.

1 Introduction

Provable security: standard vs. the random oracle model. Provable security is the defining paradigm of modern cryptography. Here, complex cryptographic constructs are designed starting from simpler ones, and the security of the former is *proved* exhibiting a reduction to the security of the latter. Although security proved this way is not unconditional, the guarantees that are obtained in this framework (known as the "standard model") typically rely only on a few widely studied and accepted assumptions.

The *random oracle* [3] model is a popular alternative to the above paradigm. Here, protocols are designed and proved secure under the additional assumption that publicly available functions that are chosen truly at random exist,[1] and concrete implementations are obtained by replacing the random oracles with cryptographic hash functions (such as SHA-1).

Although existence of random oracles enables very efficient cryptographic solutions for a large number of problems (digital encryption and signing, identification protocols etc.), in general, security proofs in this model are not sound

⋆ Work done at IBM Zurich Research Laboratory and BRICS, University of Aarhus, Denmark.

[1] These random oracles can only be accessed in a black-box way, by providing an input and obtaining the corresponding output.

with respect to the standard model: there exist constructions of various cryptographic schemes [6, 23, 18, 1] provably secure in the random oracle model, but for which no instantiation of the random oracle yields a secure scheme in the standard model. As a consequence, a central line of research in modern cryptography is designing efficient schemes provably secure in the standard model. We address this issue in the context of blind signature schemes.

Blind signatures. Since their introduction [7], blind signature schemes have been used in numerous applications, most prominently in anonymous voting schemes and anonymous e-cash.

Informally, blind signature schemes allow a user to obtain signatures from an authority on any document, in such a way that the authority learns nothing about the message that is being signed. A bit more formal, a signer S with public key pk and secret secret key sk, interacts with user U having as private input m. At the end of the interaction, the user obtains a signature σ on m. Two seemingly contradictory properties must be satisfied. The first property, termed *blindness*, requires that after interacting with various users, the signer S is not able to link a valid message-signature pair (m, σ) obtained by some user, with the protocol session during which σ was created. The second security property, termed *unforgeability*, requires that it be impossible for any malicious user that engages in k runs of the protocol with the signer, to obtain strictly more than k valid message-signature pairs. These security notions were formalized in [20] building on previous work [25, 27].

In contrast with the random oracle model where several very efficient schemes are already known [25, 26, 2], in the standard model only one such scheme has been designed [20]. The construction is based on general results regarding two-party computation and is thus extremely inefficient. In fact the authors themselves present their construction as an existence result.

Our results. Our main result is the design of an efficient blind signature scheme, provably secure in the standard model. The idea of the construction is similar to the one of [20]: consider the signing function $\mathsf{Sig}(\cdot, \cdot)$ of a signature scheme provably secure in the standard model, with input arguments a secret signing key sk and a message m. The output of the function is a signature σ on m which can later be verified using the public key pk associated to sk. We obtain a secure blind signature protocol by providing a secure two-party computation of this signing function in which the signer provides as input its secret key sk and the user provides the message m to be signed. In our implementation only the user learns the outcome of the computation, i.e., learns a signature $\sigma = \mathsf{Sig}(sk, m)$, and the signer learns nothing. Security of the resulting protocol is implied by standard properties of secure two-party computation: because S learns nothing about the message that it signed the protocol satisfies the blindness condition. Since after each interaction the user only learns a signature on a message of his choice, and nothing else, our blind signature scheme is also unforgeable.[2]

[2] A secure two-party computation is also used by Mackenzie and Reiter in [22] for generating DSA signatures. The problem they address is different and it does not seem possible to extend their solution to achieve blind signing.

We start with a more efficient and *provably secure* variant of the Cramer-Shoup signature scheme proposed by Fischlin [16]. Still, due to efficiency reasons, we do not implement this scheme directly; one of its essential ingredients is the use of a randomly chosen prime exponent each time a signature is created. In order to avoid this step, which seems to be a difficult and time consuming task, we further modify the Cramer-Shoup-Fischlin scheme by replacing the randomly chosen *prime* exponents with randomly chosen *odd integers*. An interesting result on its own, we show that the resulting scheme(mCSF) remains secure. We note that the same modification can be applied to the original Cramer-Shoup signature scheme, leading to a scheme which does not involve prime number generation. Next, we show how to implement the signing algorithm of the mCSF signature scheme as a secure two-party computation as discussed above. Efficiency is achieved by exploiting in a crucial way the algebraic properties of the mCSF signature scheme and those of Paillier's encryption scheme.

We prove the security of our scheme in a slightly weaker sense than the one captured by the the model of [20]. There, the setting that is considered involves an adversary interacting with the the honest party via multiple, possibly interleaved executions of the protocol. In contrast, we prove security of our scheme in a setting the signer executes the protocol sequentially only. The reason for this is that our proof of unforgeability requires rewinding of the user which, in the case of interleaved sessions, typically leads to an exponential blow-up of the reduction. This is similar to what Dwork et al. observed for rewinding w.r.t. to proving zero-knowledge for arbitrary proof systems [14]. We note that in fact, similar restrictions need to be applied to the scheme of Juels et al. [20], a point which until today has been overlooked. We postpone for the full version of the paper a discussion on the techniques that could potentially be used to achieve security of the protocol in such a concurrent setting.

The rest of the paper is organized as follows. In §2 we present some background on ingredients that go into our construction. §3 contains formal definitions of security for blind signatures. We then introduce and prove secure the mCSF signature scheme, §4. Finally we present a two party protocol computing the signing function of this scheme and prove that the resulting blind signature scheme is indeed secure.

2 Preliminaries

Statistically hiding commitment schemes. A building block, fundamental for our scheme, is a *statistically hiding commitment scheme* with an efficient statistical zero-knowledge proof of knowledge of the committed value. Consider a domain X. A commitment scheme to elements in X is given by a family $\{\mathsf{Comit}\}_{n\in\mathbf{N}}$, where $\mathsf{Comit}_n : X \times \{0,1\}^{r(n)} \to \{0,1\}^{l(n)}$; here $r(n)$ represents the number of random coins used to commit, and $l(n)$ is the bit-length of such a commitment. The security requirement that we need is that the scheme is statistically hiding, i.e., for any $x_0, x_1 \in X$, the distribution ensembles $\{\mathsf{Comit}(x_0, U(r(n))\}_n$ and $\{\mathsf{Comit}(x_1, U(r(n))\}_n$ are statistically indistinguishable, where $U(r(n))$ de-

notes the random variable of choosing an integer uniformly from $\{0,1\}^{r(n)}$. We are using essentially the scheme of [17, 11]: if G is a group of unknown order (for example $\mathbb{Z}_{\mathsf{n}}$ with n an RSA modulus with unknown factorization,) and $\mathfrak{g}$ and $\mathfrak{h}$ are random group elements then $\mathsf{Comit}(x)$ is defined by $\mathfrak{g}^x\mathfrak{h}^r$, where r is randomly chosen from a big enough domain.

Paillier encryption. Our protocol also makes use of the Paillier encryption scheme. Following [24], the algorithms defining the scheme, i.e., $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ are as follows. For a security parameter k, the key generation $\mathcal{K}$ algorithm picks two primes p and q of of bit-length k, sets n to be the product of the two primes and $\mathsf{h} := \mathsf{n} + 1$. The public key is (h, n), and the secret key is $\mathsf{d} = \mathsf{lcm}(\mathsf{p} - 1, \mathsf{q} - 1)$. A message $m \in [0, n-1]$ is encrypted by choosing $\mathsf{u} \in_R \mathbb{Z}_{\mathsf{n}^2}$ and computing the ciphertext $\mathsf{c} := \mathsf{h}^m\mathsf{u}^{\mathsf{n}} \bmod \mathsf{n}^2$. Given the secret key d, the clear-text $\hat{m}$ can be obtain from the cipher-text c as $\hat{m} := \tilde{m}\mathsf{d}^{-1} \bmod \mathsf{n}$ with $\tilde{m} := (\frac{(\mathsf{c}^{\mathsf{d}} \bmod \mathsf{n}^2)-1}{\mathsf{n}})$.

We will use the homomorphic properties of the Paillier encryption: if c_1 and c_2 are the encryptions of m_1 and m_2 respectively, then c_1^r is the encryption of $m_1 r$ and $\mathsf{c}_1\mathsf{c}_2$ is the encryption of $m_1 + m_2 \bmod \mathsf{n}$.

Efficient proof protocols. A Σ-protocol [9] is a protocol between a prover and a verifier, running on some common input y. Additionally, the prover has some additional input x. Such protocols are three move protocols: in the first move the prover sends the verifier a "commitment" message t, in the second move the verifier sends the prover a random "challenge" message c, and in the third move the prover sends the verifier a "response" message s.

Such a protocol is *special honest verifier zero knowledge* if there exists a simulator that, on input (y, c), outputs (t, s) such that the distribution of the triple (t, c, s) is is indistinguishable from that of an actual conversation, conditioned on the event that the verifier's challenge is c. This property implies (ordinary) honest verifier zero knowledge, and also allows the protocol to be *easily and efficiently* transformed into one satisfying much stronger notions of zero knowledge (e.g., using techniques in [10]).

Such a protocol is said to satisfy the *special soundness condition with respect to a property P* if it is computationally infeasible to find two valid conversations (t, c, s) and (t, c', s'), with $c \neq c'$, unless the input y satisfies P. Via standard rewinding arguments, this notion of soundness implies the more general notion of computational soundness.

We use notation introduced by Camenisch and Stadler [5] for the various zero-knowledge proofs of knowledge of discrete logarithms and proofs of the validity of statements about discrete logarithms. For instance,

$$PK\{(\alpha, \beta, \gamma) : y = g^\alpha h^\beta \ \wedge \ \mathfrak{y} = \pm\mathfrak{g}^\alpha\mathfrak{h}^\gamma \ \wedge \ (u \leq \alpha \leq v)\}$$

denotes a "*zero-knowledge* P*roof of* K*nowledge of integers α, β, and γ such that $y = g^\alpha h^\beta$, $\mathfrak{y} = \pm\mathfrak{g}^\alpha\mathfrak{h}^\beta$, and $u \leq \alpha \leq v$ holds*," where $y, g, h, \mathfrak{y}, \mathfrak{g}$, and $\mathfrak{h}$ are elements of some groups $G = \langle g \rangle = \langle h \rangle$ and $\mathfrak{G} = \langle \mathfrak{g} \rangle = \langle \mathfrak{h} \rangle$. The convention is that the elements listed in the round brackets denote quantities the knowledge of which is being proved (and are in general not known to the verifier), while all other parameters are known to the verifier. To make this distinction easier, we

use Greek letters to denote the quantities the knowledge of which is proved, and non-Greek letters for all quantities.

Smoothness of integers. Our proofs use several number theoretical facts related to the smoothness of randomly chosen integers. We will denote

$$\Psi(x,y) = \#\{0 < n \leq x : n_1 \leq y\}\,,\ \ \Psi(x,y,z) = \#\{0 < n \leq x : n_1 \leq y, n_2 \leq z\}\ \ .$$

where n_1 and n_2 are the first and the second biggest prime factors of n. Various bounds on these quantities are known from the existing literature (see for example [13, 12, 21, 19]) and these bounds are further used to derive concrete bounds on the probability that certain randomly chosen integers are (semi-)smooth.

3 Formal Model for Blind Signatures

In this section we recall the formal definition and the standard security notion for blind signature schemes introduced in [20].

Syntax. A blind signature scheme $\mathcal{BS} = (\mathsf{Kg}, \mathsf{Signer}, \mathsf{User}, \mathsf{Vf})$ is given by:

- the probabilistic key generation algorithm Kg takes as input security parameters params and outputs a pair (pk, sk) of public-secret keys; we write $(pk, sk) \in_R \mathsf{Kg}(\mathsf{params})$ for the process of running the key generation algorithm with fresh coins;
- Signer and User are two interactive probabilistic Turing machines that run in polynomial time. Each machine has a read-only input tape, a write-only output tape, a read/write work tape, and a read-only random tape. The machines communicate using a read-only and a write-only tape. Both machines have a common input that consists of a public key pk produced by the key generation algorithm. As private inputs, the Signer machine has the secret key sk corresponding to pk, and the User machine has a message m to be signed. The two parties interact, and, at the end of the interaction the expected local output is as follows. The Signer outputs one of the two messages *completed, not-completed*, and the User outputs either *fail* or a signature σ on m.
  We write $\sigma \in_R [\mathsf{User}(pk, m), \mathsf{Signer}(pk, sk)]$ for the process of producing signature σ on message m.
- the deterministic Vf verification algorithm takes as input the public key pk, a message m and a candidate signature σ and outputs 0/1, i.e., it either rejects or accepts.

It is required that for all (pk, sk) that have non-zero probability of being output by Kg, and all messages m, if $\sigma \in_R [\mathsf{Signer}(sk), \mathsf{User}(m)]$ then $\mathsf{Vf}(pk, (m, \sigma)) = 1$. The essential security properties for blind signatures, defined in [20] are unforgeability and blindness:

Unforgeability and strong unforgeability. Unforgeability is defined via an experiment parameterized by a security parameter k. The experiment involves an adversarial user user $\mathcal{U}$ and is as follows: First a public/secret key pair for the

signer is generated by running the key generation algorithm $(pk, sk) \in_R \mathsf{Kg}(k)$. Then, $\mathcal{U}$ engages in polynomially many runs of the protocol with the signer, interleaved at its own choosing. Finally $\mathcal{U}$ outputs a list of message-signature pairs $((m_1, \sigma_1), (m_2, \sigma_2), ..., (m_l, \sigma_l))$ with $m_i \neq m_j$. Let s be the number of runs successfully completed by the signer. We define the advantage of $\mathcal{U}$ by

$$\mathbf{Adv}^{\text{unforg}}_{\mathcal{BS},\mathcal{U}}(k) = \Pr\left[\,(\forall 1 \leq i \leq l, \mathsf{Vf}(pk, (m_i, \sigma_i)) = 1) \wedge (s < l)\,\right]$$

and say that blind signature scheme $\mathcal{BS}$ is unforgeable if $\mathbf{Adv}^{\text{unforg}}_{\mathcal{BS},\mathcal{U}}(\cdot)$ is negligible for any adversary $\mathcal{U}$. If $(m_i, \sigma_i) \neq (m_j, \sigma_j)$ instead of $m_i \neq m_j$ holds for message-signature pairs output by the adversary, the blind signature scheme is said to be *strongly* unforgeable[3].

Blindness. We define blindness via an experiment involving an adversarial signer $\mathcal{S}$. The experiment is parameterized by a bit b and security parameter k. It starts out by generating public/secret keys (pk, sk) by running the key generation algorithm on the security parameter. Then, the adversary outputs a pair of messages (m_0, m_1) lexicographically ordered. In the next stage of the experiment $\mathcal{S}$ engages in two (possibly correlated and interleaved) runs with two honest users, with inputs m_b and $m_{\bar{b}}$, respectively. If both users obtain valid signatures, on their respective message, $\mathcal{S}$ is also given these two signatures; otherwise there is no extra input to $\mathcal{S}$; in either case, $\mathcal{S}$ is required to output a bit d. We define the advantage of $\mathcal{S}$ by:

$$\mathbf{Adv}^{\text{blind}}_{\mathcal{BS},\mathcal{S}}(k) = 2 \cdot \Pr\left[\,b = d\,\right] - 1$$

and say that $\mathcal{BS}$ satisfies the blindness property, if for all polynomial time adversaries $\mathcal{S}$, the function $\mathbf{Adv}^{\text{blind}}_{\mathcal{BS},\mathcal{S}}(\cdot)$ is negligible (in the security parameter) for any polynomial time adversary $\mathcal{S}$.

4 A Modification of the Cramer-Shoup-Fischlin Signature Scheme

In this section we introduce the mCSF signature scheme. Recall that the original scheme [16] is parameterized by two security parameters k (the length of the RSA moduli) and l (the length of the hash function outputs), with $l < k$, and is strongly unforgeable under chosen message attack, assuming that the strong RSA assumption holds.

Before we provide our modified scheme, we discuss our motivation for the modifications. If we wanted to use the plain Cramer-Shoup-Fischlin signature scheme as a basis for our blind signatures scheme, we would have to implement the generation of a random prime exponent as a two party protocol. This would

[3] This distinction which is analogous to the case of standard signature schemes, was not explicitly made in [20]. We note that for the main application of blind signatures, i.e., electronic cash, unforgeability (rather than strong unforgeability) suffices.

be quite costly and thus not result in an efficient blind signature scheme. However, jointly generating a random integer can be done very efficiently, using a suitable commitment scheme. ¿From our analysis on smooth numbers, it turns out that one can indeed replace the random *prime* exponents by random sufficiently large integers. That is, if one considers a random interval I of size $2^{l'}$ of integers of size at least 2^{ul}, one finds that the probability that *all* integer in the interval have a prime factor that is bigger than $2^{l'}$ is negligible for suitably large u (and suitable l'). In fact, these prime factors will be unique: Assume the contrary, i.e., let $e_0 = pk_0$ and $e_1 = pk_1$, with p being a common factor larger than $2^{l'}$. Now $e_0 - e_1 = p(k_0 - k_1)$. As $p > 2^{l'}$, it follows that not both e_0 and e_1 can lie in I and hence any (prime) factor $p > 2^{l'}$ of an element in I is unique.

Considering the security proof of the Cramer-Shoup-Fischlin scheme, one finds that it requires the exponents to have a unique prime factor that is bigger than the outputs of the hash function. So, if we set l' to be bigger than the output length l of the hash function used, we can indeed replace the random primes by random integers from the interval I. However, it turns out that this required to choose rather large integers. Fortunately, we can do better: A closer inspection of the signature scheme's security proof shows that it is sufficient that (1) every integer in the interval has a unique prime factor larger than $2^{l'}$ and (2) that the integers the signer uses to sign have either a prime factor larger than 2^l or two prime factors larger than $2^{l'}$ with $2l' > l$. This facts allow us to choose much smaller intergers. We will give a detailed concrete treatement of the security of the resulting scheme in the full version of this paper.

We are now ready to describe our modification of the Cramer-Shoup-Fischlin signature scheme. Apart from using random integer exponents instead of prime ones, we operate two further modifications. The reason for both of them is purely technical. The first one takes care of the problem of when doing proofs of knowledge modulo an RSA modulus that is safe-prime product. That is, we introduce an extra squaring in the verification equations which will allow us later in the blind signature generation protocol to square the "blinded message" to cast it into the group of squares modulo n. The second one is splitting the signing algorithm in two stages: To sign message m, the algorithm first outputs some random data and then, in the second stage, outputs the remaining part of the signature on m, deterministically determined by the message and the output of the first stage. This modification will allow us to reduce the security of our blind signature scheme to the security of the mCSF scheme.

The mCSF Signature Scheme. The scheme uses parameters k, l, l', and u where $l/2 \leq l' < l$ and u is a real number. The parameters l', and u are as above. The parameter k denotes here the bit-length of the prime factor of the RSA modulus, l is the bit-length of the output of a public, collision resistant hash function $\mathcal{H}() : \{0,1\}^* \rightarrow \{0,1\}^l$. Also, let t be the maximal number of messages to be signed.

The parameters l' and u strictly depend on l and can be chosen such that the discussed above smoothness probabilities are sufficiently small. Also, we note

that for practical purposes it is possible to choose u based on concrete bounds. We will show how to derive these bounds in the full version of our paper.

The algorithms defining the mCSF signature scheme are the following:

- The key generation algorithm KGen generates two random safe primes $p = 2p' + 1$ and $q = 2q' + 1$ of bit-length k and sets $N := pq$. It also draws at random $x, h_1, h_2 \in QR_N$ and a random integer $f_0 \in_R]0, 2^{lu-l'}[$ and sets $f := 2^{l'} f_0 + 1$. Then, it chooses a public collision-resistant hash function $\mathcal{H}()$. The public key is $(N, h_1, h_2, x, \mathcal{H}(), f, l')$, and the corresponding secret key is $\phi = (p-1)(q-1) = 4p'q'$.
- Signing a message m is done as follows. Pick a random l-bit string a and a random odd number e from the interval $[f, f + 2^{l'}[$ and output it. This completes the first stage of the signing process. Then, on a further request (not necessarily executed immediately afterwards) compute y such that $y^e = (x h_1^a h_2^{(a+\mathcal{H}(m) \bmod 2^l)})^2$. The signature on m is $\sigma = (e, a, y)$.
- A signature (e, a, y) on message m is valid if e is odd and the following two relations are valid

$$f \leq e < f + 2^{l'} \quad \text{and} \quad y^e = (x h_1^a h_2^{(a+\mathcal{H}(m) \bmod 2^l)})^2 \ . \tag{1}$$

Security analysis. The security of the mCSF signature scheme is captured by the following theorem:

Theorem 41 *Let $\mathcal{H}()$ be a collision-resistant hash function, let $l > l' \geq l/2$ and u be such that $\Psi\left(2^{lu}, 2^{l'}\right) 2^{-lu+l'}$ and $\Psi\left(2^{lu}, 2^{l}, 2^{l'}\right) 2^{-lu+4l'/5}$ are negligible. Then* mCSF *is strongly unforgeable under adaptive chosen message attack provided that the strong RSA assumption holds.*

Moreover, the signature scheme is secure under the more general attack, where the adversary is allowed to query for pairs (e, a) and then, at some later point, ask for signatures on a hash of a message w.r.t. some of these pairs (but only once per pair).

We postpone the proof for the full version of this paper.

Notice that if we are interested in signing only short messages, we do not need to assume a collision resistant hash functions.

Two issues are raised in comparing the original scheme to our modified version. On the one hand, our scheme has a more efficient signing algorithm: the prevailing cost for signing in the original scheme is to find the prime exponent e. In our scheme the signer just needs to choose a random number which, for the computation of y, the signer can further reduce it modulo $\phi(n)$. On the other hand, our verification protocol is less efficient, as the verifier has to perform a computation with an exponent e that is much larger in our case (and the verifier cannot reduce it modulo $\phi(n)$).

5 Our Blind Signature Protocol

In this section we give the construction of our blind signature scheme. As we have anticipated, its signing protocol is a two-party computation of the signing function of the mCSF signature scheme, while the verification algorithms is the same. The two parties, henceforth a user U and a signer S, provide as private inputs to the signing protocol a message m and a secret key ϕ, respectively, and jointly compute an mCSF signature $\sigma = (a, e, y)$ on m. The properties of the joint computation are such that S learns absolutely no information (in a strong, information-theoretic sense), and the user U learns the signature σ, but no information on ϕ (in a computational sense.)

We start by discussing the main ideas behind our construction. Recall that given the public key $(N, h_1, h_2, x, \mathcal{H}(), f, l')$, the signer, which has the corresponding secret key ϕ, and the user, having some private input m, need to compute values (e, a, y) such that $y^e \equiv (xh_1^a h_2^{(a+\mathcal{H}(m) \bmod 2^l)})^2 \pmod N$. This is done as follows.

First the parties jointly generate a random e and a in such a way that only U learns their values. For this, U first commits to random shares for e and a (via a statistically hiding commitment scheme), and sends these commitments to the signer. The signer replies with his own shares, allowing U to compute the resulting e and a as sum of the corresponding shares. At this point, U computes the value $xh_1^a h_2^{(a+\mathcal{H}(m) \bmod 2^l)}$ by himself, blinds it using a mechanism similar to the one in Chaum's RSA-based blind signature scheme [8], and sends the resulting value to the signer.

The signer and the user together compute the value $\hat{e} = e\bar{e} + r\phi$, which is statistically independent from e and thus reveals no information about e to the signer. At this point the signer can compute an $e\bar{e}$-th root modulo N of the blinded message which he then returns to the user. Finally, the user eliminates the blinding factor and obtains (e, a, y), a signature on m. A key element of our protocol are *efficient* zero-knowledge proofs that all messages of the user follow the protocol as outlined above. We note that the signer does not need to prove that it behaves according to the protocol: the signer can only cheat in the computation of the last message it sends which will result in an invalid signature therefore the user will note cheating here. We now proceed with the detailed description of our scheme.

Key generation. The key generation algorithm Kg takes as input four security parameters k, k', l, l' (k, l, l' are as in Theorem 41 and $k' \approx 100$ is a parameter that controls the statistical blindness of e) and is as follows:

Algorithm $\mathsf{Kg}(k, k', l, l')$:

1. Select keys for the Paillier encryption scheme, i.e., pick two primes $\mathsf{p}, \mathsf{q} > 2^{(lu+2k+k'+1)/2}$; set $\mathsf{n} := \mathsf{pq}$, $\mathsf{h} := (1 + \mathsf{n})$, $\mathsf{d} := \mathrm{lcm}(\mathsf{p} - 1, \mathsf{q} - 1)$
2. Choose $u > 1$ as described in §4.
3. Select keys for the mCSF signature scheme, i.e., pick two safe prime p and q of bit-length k, set $N := pq$, and $\phi := (p - 1)(q - 1)$.

4. Pick $\mathcal{H}()$ at random from a collision resistant hash function family with an output size of l bits;
5. Select a random f_0 from the interval $]0, 2^{lu-l'}[$; set $f := 2^{l'} f_0 + 1$;
6. Select random $x, h_1, h_2, h_3 \in_R QR_N$.
7. Select $\mathsf{v} \in_R \mathbb{Z}_{\mathsf{n}}^*$ and set $\mathsf{c} := \mathsf{h}^{\phi}\mathsf{v}^{\mathsf{n}} \bmod \mathsf{n}^2$.
8. Select auxiliary keys for the proof-protocols, i.e., pick two safe prime $\mathfrak{p}, \mathfrak{q}$ of bit-length k, set $\mathfrak{n} := \mathfrak{pq}$. Select two generators $\mathfrak{g}$ and $\mathfrak{h}$ of $QR_{\mathfrak{n}}$.
9. Set the public key of S to $(N, h_1, h_2, h_3, x, f, \mathfrak{n}, \mathfrak{h}, \mathfrak{g}, \mathsf{n}, \mathsf{c}, \mathsf{h}, k, k', l, l', u)$, set the secret key of S to (ϕ, d).

We imagine that this key generation algorithm is either run by a trusted third party which then hands over the secret key to the signer or, alternatively, the signer runs the key generation by itself, and then proves to a trusted party that it had followed the protocol. Specifically, it needs to prove that 1) the moduli N and $\mathfrak{n}$ are indeed products of safe-primes, 2) that $h_1, h_2 \in \langle h_3 \rangle$ and $\mathfrak{g} \in \langle \mathfrak{h} \rangle$, and 3) that c is a Paillier encryption of $\phi(N)$. Proving these statements can be done via standard protocols: Showing that N and $\mathfrak{n}$ are products of two safe prime can be done as in Camenisch and Michels [4]. In order to prove that $h_1, h_2 \in \langle h_3 \rangle$, it is sufficient to show that the h_i's are squares [15] and to check that $\gcd(h_3 \pm 1, N) = 1$. Similarly, one can show that $\mathfrak{g} \in \langle \mathfrak{h} \rangle$. Finally, the statement that c is an encryption of ϕ can be proved, given an auxiliary modulus $\hat{\mathfrak{n}}$ that is the product of two safe primes of bit-size k and two generators $\hat{\mathfrak{g}}$ and $\hat{\mathfrak{h}}$ of $QR_{\hat{\mathfrak{n}}}$ such that the signer does not knows the factorization of $\hat{\mathfrak{n}}$ or an integer $\hat{\mathfrak{u}}$ such that $\hat{\mathfrak{h}} = \hat{\mathfrak{g}}^{\hat{\mathfrak{u}}}$. (Such parameters could be generated either by a trusted third party, or by the party verifying the proof). Because this proof is less known, we give its details in the sequel: The signer chooses $v_1, v_2 \in_R [1, \lfloor \frac{\hat{\mathfrak{n}}}{4} \rfloor]$, computes $\Phi := \hat{\mathfrak{g}}^{\phi}\hat{\mathfrak{h}}^{v_1}$ and $P := \hat{\mathfrak{g}}^{p-1}\hat{\mathfrak{h}}^{v_2}$, sends Φ and P to the verifier and runs the protocol

$$\textit{Proof 0} = PK\{(\alpha, \beta, \gamma, \delta, \xi_1, \ldots, \xi_4) : \ \Phi = \pm\hat{\mathfrak{g}}^{\alpha}\hat{\mathfrak{h}}^{\xi_1} \ \wedge \ P = \pm\hat{\mathfrak{g}}^{\beta}\hat{\mathfrak{h}}^{\xi_2} \ \wedge$$
$$\Phi = \pm P^{\gamma}\hat{\mathfrak{h}}^{\xi_3} \ \wedge \ \frac{\hat{\mathfrak{g}}^N}{\hat{\mathfrak{g}}P} = \pm(\hat{\mathfrak{g}}P)^{\gamma}\hat{\mathfrak{h}}^{\xi_4} \ \wedge \ \mathsf{c} = \mathsf{h}^{\alpha}\delta^{\mathsf{n}} \pmod{\mathsf{n}^2} \ \wedge \ 1 \leq \beta \leq N-2\}$$

with the verifier. (See Theorem 51 for an analysis of this protocol.) The signer could prove these statements just once to the certification authority (or to some other representative trusted by the users in this respect) or to each user individually.

Blind signing protocol. We give the details of the blind signature protocol in Figure 1. The protocol is as follows:

The user U sends to the signer S commitments E_U, A_U, and M to random values e_U and a_U and to $\mathcal{H}(m)$, respectively, and proves to the signer that she knows how to open these commitments by running the interactive protocol

$$\textit{Proof 1} = PK\{(\alpha, \beta, \gamma, \delta, \varepsilon, \varrho) : \ E_U = \pm\mathfrak{g}^{\alpha}\mathfrak{h}^{\beta} \ \wedge \ M = \pm\mathfrak{g}^{\gamma}\mathfrak{h}^{\delta} \ \wedge \ A_U = \pm\mathfrak{g}^{\varepsilon}\mathfrak{h}^{\varrho}\}$$

with the signer. If the proof succeeds, the signer S responds with random values a_S and e_S with which U to computes e and a. Thus both parties are assured

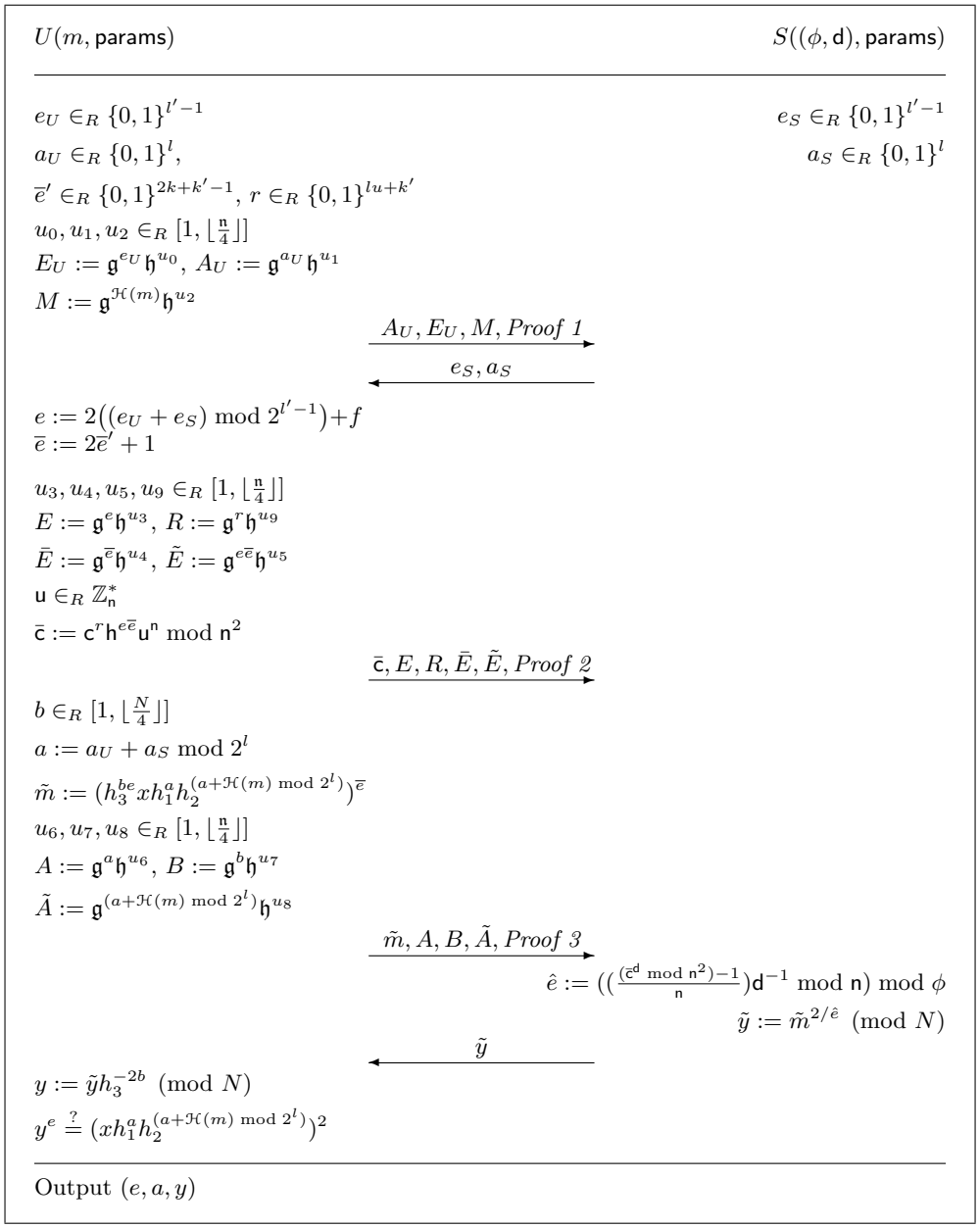

Fig. 1. A two-party protocol for producing mCSF signatures. The signer does not learn any information about the signed message. Here, the common parameters params are $((N, h_1, h_2, h_3, x, f, \mathsf{n}, \mathfrak{h}, \mathfrak{g}, \mathsf{n}, \mathsf{c}, \mathsf{h}, k, k', l, l', u))$

that a and e are chosen randomly but S does not learn anything more about a and e.

Next, the two parties compute (an encryption $\bar{\mathsf{c}}$ of) $\hat{e} = e\bar{e} + r\phi(N)$ from the encryption c of $\phi(N)$: U picks random blinding factors $\bar{e}$ and r, computes $\bar{\mathsf{c}} := \mathsf{c}^r \mathsf{h}^{e\bar{e}} \mathsf{u}^{\mathsf{n}} \bmod \mathsf{n}^2$, and sends $\bar{\mathsf{c}}$ to S. By means of the auxiliary commitments E, $\bar{E}$, and $\tilde{E}$ and the Σ-protocol

$$\begin{aligned}\mathit{Proof\ 2} = PK\{(\alpha,\beta,\gamma,\delta,\varepsilon,\xi_1,\ldots,\xi_8) : \frac{E}{\mathfrak{g}^f} = \pm(\mathfrak{g}^2)^{\alpha}\mathfrak{h}^{\xi_1} \ \wedge \\ \frac{E}{E_U^2\mathfrak{g}^{(2e_S+f)}} = \pm(\mathfrak{g}^{2^{l'}})^{\xi_2}\mathfrak{h}^{\xi_3} \ \wedge \ \bar{E}/\mathfrak{g} = \pm(\mathfrak{g}^2)^{\beta}\mathfrak{h}^{\xi_4} \ \wedge \ \tilde{E} = \pm\mathfrak{g}^{\gamma}\mathfrak{h}^{\xi_6} \ \wedge \\ \tilde{E}/E = \pm(E^2)^{\beta}\mathfrak{h}^{\xi_5} \ \wedge \ R = \pm\mathfrak{g}^{\delta}\mathfrak{h}^{\xi_8} \ \wedge \ \bar{\mathsf{c}} = \mathsf{c}^{\delta}\mathsf{h}^{\gamma}\xi_7^{\mathsf{n}} \pmod{\mathsf{n}^2} \ \wedge \\ \alpha \in \{0,1\}^{l'-1} \ \wedge \ \beta \in \{0,1\}^{2k+k'-1} \ \wedge \ \delta \in \{0,1\}^{lu+k'}\}\end{aligned}$$

U convinces S that $\bar{\mathsf{c}}$ was correctly computed, that r and $\bar{e}$ have the required length, that e was computed as $2(e_U + e_S \bmod 2^{l'-1}) + f$, and that $\bar{e}$ is odd (cf. Theorem 51). It is not hard to show that $e\bar{e} + r\phi(N)$ is statistically independent of e.

Next, U computes the "blinded message" $\tilde{m} = (h_3^{be} x h_1^a h_2^{(a+\mathcal{H}(m) \bmod 2^l)})^{\bar{e}}$, where b is the randomly chosen blinding factor, and sends $\tilde{m}$ to S. Using the auxiliary commitments A, B, and M and the Σ-protocol

$$\begin{aligned}\mathit{Proof\ 3} = PK\{(\alpha,\beta,\gamma,\delta,\varepsilon,\mu,\varphi,\rho,\omega,\nu,\kappa,\xi_1,\ldots,\xi_{11}) : \quad A = \pm\mathfrak{g}^{\alpha}\mathfrak{h}^{\xi_1} \ \wedge \\ B = \pm\mathfrak{g}^{\beta}\mathfrak{h}^{\xi_2} \ \wedge \ \tilde{A} = \pm\mathfrak{g}^{\gamma}\mathfrak{h}^{\xi_3} \ \wedge \ \tilde{E} = \pm\mathfrak{g}^{\delta}\mathfrak{h}^{\xi_4} \ \wedge \ \bar{E} = \pm\mathfrak{g}^{\varepsilon}\mathfrak{h}^{\xi_5} \ \wedge \\ M = \pm\mathfrak{g}^{\mu}\mathfrak{h}^{\xi_6} \ \wedge \ \tilde{m} = \pm h_3^{\varphi} x^{\varepsilon} h_1^{\sigma} h_2^{\rho} \ \wedge \ 1 = \pm A^{\varepsilon}(\frac{1}{\mathfrak{g}})^{\sigma}(\frac{1}{\mathfrak{h}})^{\xi_7} \ \wedge \\ 1 = \pm B^{\delta}(\frac{1}{\mathfrak{g}})^{\varphi}(\frac{1}{\mathfrak{h}})^{\xi_8} \ \wedge \ 1 = \pm\tilde{A}^{\varepsilon}(\frac{1}{\mathfrak{g}})^{\rho}(\frac{1}{\mathfrak{h}})^{\xi_9} \ \wedge \ A_U = \pm\mathfrak{g}^{\omega}\mathfrak{h}^{\xi_{12}} \ \wedge \\ \frac{A}{A_U\mathfrak{g}^{a_S}} = \pm(\mathfrak{g}^{2^l})^{\nu}\mathfrak{h}^{\xi_{10}} \ \wedge \ \frac{\tilde{A}}{MA} = \pm(\mathfrak{g}^{2^l})^{\kappa}\mathfrak{h}^{\xi_{11}} \ \wedge \ \alpha,\mu,\gamma \in \{0,1\}^l\}\end{aligned}$$

she convinces S that $\tilde{m}$ is correctly formed, that a is computed as $a_U + a_S \bmod 2^l$, and that $\mathcal{H}(m) \in \{0,1\}^l$ (cf. Theorem 54).

Next, the signer decrypts $\bar{\mathsf{c}}$ to obtain $\hat{e}(\equiv e\bar{e} + r\phi(N) \pmod{\phi(N)})$, calculates $\tilde{y} := \tilde{m}^{2/\hat{e}}$, and sends $\tilde{y}$ to U. It is immediate that

$$\tilde{y} = (h_3^{be} x h_1^a h_2^{(a+\mathcal{H}(m) \bmod 2^l)})^{2\bar{e}/(e\bar{e}+r\phi(N))} = h_3^{2b}(x h_1^a h_2^{(a+\mathcal{H}(m) \bmod 2^l)})^{2/e}).$$

Finally, U removes the blinding factor by computing $y := \tilde{y} h_3^{-2b}$ and thereby obtains the signature (e, a, y) on m.

5.1 Security of the Scheme for Sequential Sessions

Proving security of our protocol requires analyzing the Σ-subprotocols. Their security is captured by Theorems 51, 52, 53, and 54 whose proofs are postponed for the full version of this paper.

Theorem 51 *Under the strong RSA assumption and provided that N is a product of two primes, Proof 0 constitutes a statistical zero-knowledge argument that* c *is an encryption of $\phi(N)$.*

Theorem 52 *Under the strong RSA assumption Proof 1 constitutes a statistical zero-knowledge proof of knowledge of the integers committed to by E_U, A_U, and M. Also, the proof is witness indistinguishable w.r.t. the integers committed to by E_U, A_U, and M.*

The proof of this theorem follows from the explanations in §2 and the fact the commitments E_U, A_U, and M are statistically hiding.

Theorem 53 *Under the strong RSA assumption and provided that* c *encrypts a value μ such that $0 \leq \mu \leq 2^{2k}$, Proof 2 constitutes a statistical zero-knowledge argument that*

1. $\bar{\mathsf{c}}$ *is an encryption of the integer $e\bar{e}+r\mu$, where r is an integer in $\{0,1\}^{lu+k'}$.*
2. $e = 2\big((e_U+e_S) \bmod 2^{l'-1}\big)+f$ *and $\bar{e}$ is odd, where e is the integer committed by E, $\bar{e}$ is the one committed by $\bar{E}$, and e_U is the integer committed by E_U.*

Moreover, the protocol is witness indistinguishable w.r.t. all $\bar{e}$, e, and r such that $e\bar{e}+r\mu$ equals the value encrypted in $\bar{\mathsf{c}}$.

Theorem 54 *Under the strong RSA assumption Proof 3 constitutes a statistical zero-knowledge argument that*

$$\tilde{m} = \pm h_3^{b\tilde{e}}(x h_1^a h_2^{(a+\dot{m} \bmod 2^l)})^{\bar{e}} \ , \quad a = a_U + a_S \bmod 2^l \ , \quad \textit{and} \quad \tilde{e} = e\bar{e} \qquad (2)$$

holds, where a is the integer committed to by A, a_U is the integer committed to by A_U, e the integer committed to by E, $\bar{e}$ the integer committed to by $\bar{E}$, $\tilde{e}$ the integer committed to by $\tilde{E}$, and $\dot{m}$ the integer committed to by M. Also, the protocol is witness indistinguishable w.r.t. all a, a_U, b, e, $\bar{e}$, $\tilde{e}$, and $\dot{m}$ such that the Equations (2) hold.

Our main results is the following theorem.

Theorem 55 *Under the strong RSA and the decisional n-residuosity assumptions the blind signature scheme depicted in Figure 5 is blind and strongly unforgeable under an adaptive chosen message attack, if executed sequentially polynomially many times.*

References

1. M. Bellare, A. Boldyreva, and A. Palacio. An un-instantiable random-oracle-model scheme for a hybrid-encryption problem. In *EUROCRYPT 2004*, LNCS, 2004.
2. M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The power of RSA inversion oracles and the security of Chaum's RSA-based blind signature scheme. In *Proceedings of Financial Cryptography 01*. Springer-Verlag, 2001.

3. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS* , pp. 62–73, 1993.
4. J. Camenisch and M. Michels. Proving in zero-knowledge that a number n is the product of two safe primes. In *EUROCRYPT '99*, vol. 1592 of *LNCS*.
5. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In *CRYPTO '97*, vol. 1296 of *LNCS*. Springer Verlag, 1997.
6. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *Proceedings of the 13th Annual ACM STOC*, pp. 209–218, 1998.
7. D. Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology — Proceedings of CRYPTO '82*, pp. 199–203. Plenum Press, 1983.
8. D. Chaum. Blind signature systems. In *Advances in Cryptology — CRYPTO '83*, p. 153. Plenum Press, 1984.
9. R. Cramer. *Modular Design of Secure yet Practical Cryptographic Protocol*. PhD thesis, University of Amsterdam, 1997.
10. I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *EUROCRYPT 2000*, vol. 1807 of *LNCS*. Springer Verlag, 2000.
11. I. Damgård and E. Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *ASIACRYPT 2002*, vol. 2501 of *LNCS*.
12. N. de Bruijn. On the number of positive integers $\leq x$ and free of prime factors $> y$. *Nederl. Akad. Wetensch. Proceedings*, 53:813–821, 1950.
13. K. Dickman. On the frequency of numbers containing prime factors of a certain relative magnitude. *Arkiv för Matematik, Astronomi och Fysik*, 22A(10), 1930.
14. C. Dwork, M. Naor, and A. Sahai. Concurrent zero-knowledge. In *Proceedings of the 30th Annual STOC*, pp. 409–418, Dallas, TX, USA, May 1998. ACM Press.
15. A. Fiat and A. Shamir. How to prove yourself: Practical solution to identification and signature problems. In *CRYPTO '86*, vol. 263 of *LNCS*. Springer Verlag, 1987.
16. M. Fischlin. The Cramer-Shoup Strong-RSA signature scheme revisited. In *PKC 2003*, vol. 2567 of *LNCS*, pp. 116–129. Springer-Verlag, 2003.
17. E. Fujiski and T. Okamoto. Statistical zero-knowledge protocols to prove modular polynomial relations. In *CRYPTO '97*, vol. 1294 of *LNCS*, pp. 16–30, 1997.
18. S. Goldwasser and Y. Tauman. On the (in)security of the Fiat-Shamir transform. In *Proceedings of Foundations of Computer Science*, 2003.
19. A. Hildebrand. On the number of positive integers $\leq x$ and free of prime factors $> y$. *Journal of Number Theory*, 22:289–307, 1986.
20. A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures. In *Proceedings of Crypto'97*, vol. 1294 of *LNCS*, pp. 150–164. Springer-Verlag, 1997.
21. D. E. Knuth and L. T. Pardo. Analysis of a simple factorization algorithm. *Theoretical Computer Science*, 3(3):321–348, Dec. 1976.
22. P. MacKenzie and M. K. Reiter. Two-party generation of DSA signatures. *International Journal of Information Security*, 2(3), 2004.
23. J. B. Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *CRYPT0 2002*, vol. 2442 of *LNCS*.
24. P. Paillier. Public-key cryptosystem based on composite degree residuosity classes. In *Proceedings of EUROCRYPT'99*, vol. 1592 of *LNCS*, pp. 223–238, 1999.
25. D. Pointcheval and J. Stern. Provably secure blind signature schemes. In *Advances in Cryptology – ASIACRYPT '96*. LNCS, Springer-Verlag, 1996.
26. D. Pointcheval and J. Stern. New blind signatures equivalent to factorization. In *ACM CCS*, pp. 92–99. ACM Press, 1997.
27. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.