Round Optimal Blind Signatures

Sanjam Garg[1], Vanishree Rao[1], Amit Sahai[1], Dominique Schröder[2,⋆], and Dominique Unruh[3]

[1] University of California, Los Angeles, USA
[2] University of Maryland, USA
[3] University of Tartu, Estonia

Abstract. Constructing round-optimal blind signatures in the standard model has been a long standing open problem. In particular, Fischlin and Schröder recently ruled out a large class of three-move blind signatures in the standard model (Eurocrypt'10). In particular, their result shows that finding security proofs for the well-known blind signature schemes by Chaum, and by Pointcheval and Stern in the standard model via black-box reductions is hard. In this work we propose the first round-optimal, i.e., two-move, blind signature scheme in the standard model (i.e., without assuming random oracles or the existence of a common reference string). Our scheme relies on the Decisional Diffie Hellman assumption and the existence of sub-exponentially hard 1-to-1 one way functions. This scheme is also secure in the concurrent setting.

1 Introduction

Blind signature schemes [13, 14] provide the functionality of a carbon copy envelope: The user (receiver), puts his message into this envelope and hands it over to the signer (sender). The signer in return signs the envelope and gives it back to the user who uses the signed enelope to recover the original message together with a signature on it. The notion of security in this context entails (1) that the signer remains oblivious about the message (blindness), but at the same time, (2) the receiver cannot forge signatures for fresh messages (unforgeability).

Blind signatures are an important primitive, whose classical applications include e-cash, e-voting, and anonymous credentials [9, 10, 8]. Moreover, oblivious transfer can be built from *unique* blind signatures [12, 17]. The several known instantiations of blind signature schemes are based on security assumptions either in the random oracle model [35, 2, 6, 7, 5, 37], or in the standard model [11, 33, 24, 28, 3]. Constructions based on general assumptions are also known [25, 16, 23, 17].

One central measure of efficiency in these schemes is the round complexity of the signing protocol. This has been an explicit problem for at least a decade, since the work of Abe [2]. Currently, the best known blind signature scheme in terms of the round complexity in the standard model is due to Okamoto [33]. This scheme has *four* rounds.

⋆ Supported in part by a DAAD postdoctoral fellowship.

P. Rogaway (Ed.): CRYPTO 2011, LNCS 6841, pp. 630–648, 2011.

All round optimal solutions (the user sends a single message to the signer and gets a single response) rely either on the random oracle heuristic [14, 7], or they require a common reference string [16,3,22,31,5,4,19,27], and some instantiations even prove their security under an interactive assumption [6, 7, 22].

Many interesting impossibility results ruling out the existence of secure blind signatures have also been proposed. Most prominently, Fischlin and Schröder [18] provide a very general impossibility result for a large class of blind signature schemes. In their result, they rule out signing protocols of less than *four* rounds, but under some natural technical conditions on the protocols (motivated by existing blind signature schemes). Interestingly, most of the round optimal blind signature schemes known today have these properties [14, 7, 16]. In particular, this means that there is not much hope to instantiate one of the known schemes under weaker assumptions. Furthermore, Katz, Schröder and Yerukhimovich [26] rule out black-box constructions of blind signature schemes from one-way permutations. In light of these result, it seems clear that significant new ideas would be needed to construct round-optimal blind signatures.

Concurrently Secure Blind Signature Schemes. Another reason why round optimal blind signature schemes are desirable is that a solution would be concurrently secure. Concurrently secure blind signature schemes, however, are difficult to obtain. Juels, Luby, and Ostrovsky [25] explained why a straight forward approach does not work. The authors present a solution that is, according to Hazay et al. [23], only secure in the sequential setting. The reason is that the solution seems to require a *concurrently secure* protocol for two-party computation. Such a protocol, however, is a mayor open problem in the standard model [23].

Obtaining a concurrently secure protocol under simulation-based definition via black-box proofs is impossible as shown by Lindell [29]. Previous protocols overcome this impossibility result by assuming a common reference string and by relying on game-based definitions. The only exception is the protocol of Hazay et al. [23] that does not need a CRS. The authors build a blind signature scheme that uses the concurrent zero-knowledge protocol of Prabhakaran, Rosen, and Sahai [36] that has an (almost) logarithmic round complexity as a building block.

1.1 Our Contribution

In this work we give the *first* round-optimal, i.e., two-move, blind signature scheme in the standard model. Our scheme is also secure in the concurrent setting. This follows directly from the fact that any two round blind signature schemes is concurrently secure, as observed by [23]. In contrast to prior schemes, our solution does not need any setup assumption such as a common reference string.

This result is especially surprising in light of the recent impossibility result of Fischlin and Schröder [18]. They provide a very general impossibility result that rules out a large class of three (or less than three) round blind signature schemes in the setting of both statistical blindness and computational blindness. Specifically, they investigate the possibility of instantiating random oracles in the schemes by

Chaum [13] and by Pointcheval and Stern [35], and of giving a security proof based only on standard assumptions. Therefore, in order to make their Cproblem tractable they restrict themselves to those blind signature schemes which satisfy a few *technical conditions* which encompass most[1] known blind signature schemes. One of these conditions is that the reduction in the unforgeability proof needs to be efficient (since the reduction is transformed into an adversary against the blindness game). In fact, this is precisely the technical condition that we avoid in our scheme and overcome the impossibility result. We note that our scheme relies on the Decision Diffie Hellman (DDH) assumption[2] and the existence of sub-exponentially hard 1-to-1 one way functions. Further, we stress that our result is only a feasibility result, and it is not as efficient as the earlier constructions. However, our work opens doors to the possibility of constructing efficient round-optimal blind signature schemes in the standard model.

Besides being interesting in its own right, our construction is an example of a scenario in which known impossibility results for concurrently-secure 2-party computation [29, 30] can be avoided to achieve meaningful game-based security definitions. Finally, we note that in a recent result Pass [34] rules out the existence of unique blind signatures using super-polynomial reductions, as long as the blindness property holds for appropriately strong adversaries. In our case blindness holds against polynomial time adversaries only and hence our result is in some tight with respect this impossibility result.

The results presented in this paper are a merge between the following two publications [38, 20].

Notations. Before presenting our results we briefly recall some basic definitions. In what follows we denote by $\lambda \in \mathbb{N}$ the security parameter. We say that a function is *negligible* if it vanishes faster than the inverse of any polynomial. A function is non-negligible if it is not negligible. If S is a set, then $x \leftarrow S$ indicates that x is chosen uniformly at random over S (which in particular assumes that S can be sampled efficiently). We write $A(x; X)$ to indicate that A is an algorithm that takes as input a value x and uses randomness X. In general, we use capital letters for the randomness. W.l.o.g. we assume that X has bit length λ.

2 Blind Signatures and Their Security

By $(a, b) \leftarrow \langle \mathcal{X}(x), \mathcal{Y}(y) \rangle$ we denote interactive execution of algorithms $\mathcal{X}$ and $\mathcal{Y}$, where x (resp., y) is the private input of $\mathcal{X}$ (resp., $\mathcal{Y}$), and a (resp., b) is the private output of $\mathcal{X}$ (resp., $\mathcal{Y}$). We write $\mathcal{X}^{\langle \cdot, \mathcal{Y} \rangle^{\infty}}$ for $\mathcal{X}$ with oracle access two arbitrarily many interactions with $\mathcal{Y}$. And $\mathcal{X}^{\langle \cdot, \mathcal{Y} \rangle^{1}}$ for $\mathcal{X}$ with oracle access two arbitrarily a single interaction with $\mathcal{Y}$.

[1] Known blind signature schemes can indeed be modified in rather unnatural ways to construct blind signature schemes that do not satisfy their conditions.

[2] We also need a ZAP (a two round witness indistinguishable proof system) which can be constructed under the DDH assumption. We note that ZAPs can in fact be constructed from any one-way trapdoor permutation.

Experiment $\mathsf{Unforge}^{\mathsf{BS}}_{\mathcal{U}^*}(\lambda)$
$(sk, vk) \leftarrow \mathsf{Gen}(1^\lambda)$
$((m_1^*, \sigma_1^*), \ldots, (m_{k+1}^*, \sigma_{k+1}^*)) \leftarrow \mathcal{U}^{*\langle \mathcal{S}(sk), \cdot \rangle^\infty}(vk)$
Return 1 iff
$m_i^* \neq m_j^*$ for all i, j with $i \neq j$, and
$\mathsf{Vrfy}(vk, m_i^*, \sigma_i^*) = 1$ for all i, and
at most k interactions with $\mathcal{S}(sk)$
were completed.

Experiment $\mathsf{Unblind}^{\mathsf{BS}}_{\mathcal{S}^*}(\lambda)$
$(vk, m_0, m_1, \mathsf{st}_{\mathsf{find}}) \leftarrow \mathcal{S}^*(\mathsf{find}, 1^\lambda)$
$b \leftarrow \{0, 1\}$
$\mathsf{st}_{\mathsf{issue}} \leftarrow \mathcal{S}^{*\langle \cdot, \mathcal{U}(vk, m_b) \rangle^1, \langle \cdot, \mathcal{U}(vk, m_{\bar{b}}) \rangle^1}(\mathsf{issue}, \mathsf{st}_{\mathsf{find}})$
and let $\sigma_b, \sigma_{\bar{b}}$ denote the
(possibly undefined) local outputs
of $\mathcal{U}(vk, m_b)$ resp. $\mathcal{U}(vk, m_{\bar{b}})$.
set $(\sigma_0, \sigma_1) = (\bot, \bot)$ if $\sigma_0 = \bot$ or $\sigma_1 = \bot$
$b^* \leftarrow \mathcal{S}^*(\mathsf{guess}, \sigma_0, \sigma_1, \mathsf{st}_{\mathsf{issue}})$
return 1 iff $b = b^*$.

Fig. 1. Security games of blind signatures

Definition 1. *A* blind signature scheme BS *consists of PPT algorithms* Gen, Vrfy *along with interactive PPT algorithms* $\mathcal{S}, \mathcal{U}$ *such that for any* $\lambda \in \mathbb{N}$*:*

- $\mathsf{Gen}(1^\lambda)$ *generates a key pair* (sk, vk)*.*
- *The joint execution of* $\mathcal{S}(sk)$ *and* $\mathcal{U}(vk, m)$*, where* $m \in \{0,1\}^\lambda$*, generates an output* σ *for the user and no output for the signer. We write this as* $(\bot, \sigma) \leftarrow \langle \mathcal{S}(sk), \mathcal{U}(vk, m) \rangle$*.*
- *Algorithm* $\mathsf{Vrfy}(vk, m, \sigma)$ *outputs a bit* b*.*

We require completeness *i.e., for any* $m \in \{0,1\}^\lambda$*, and for* $(sk, vk) \leftarrow \mathsf{Gen}(1^\lambda)$*, and* σ *output by* $\mathcal{U}$ *in the joint execution of* $\mathcal{S}(sk)$ *and* $\mathcal{U}(vk, m)$*, it holds that* $\mathsf{Vrfy}(vk, m, \sigma) = 1$ *with overwhelming probability in* $\lambda \in \mathbb{N}$*.*

Note that it is always possible to sign messages of arbitrary length by applying a collision-resistant hash function to the message prior to signing.

Blind signatures must satisfy two properties: unforgeability and blindness [25, 35]. Notice that we can also achieve the stronger definition of unforgeability from [39] by applying their transformation which does not increase the round complexity.

For unforgeability we require that a user who runs k executions of the signature-issuing protocol should be unable to output $k + 1$ valid signatures on $k + 1$ distinct messages.

Definition 2. *A blind signature scheme* $\mathsf{BS} = (\mathsf{Gen}, \mathcal{S}, \mathcal{U}, \mathsf{Vrfy})$ *is* unforgeable *if for any PPT algorithm* $\mathcal{U}^*$ *the probability that experiment* $\mathsf{Unforge}^{\mathsf{BS}}_{\mathcal{U}^*}(\lambda)$ *defined in Figure 1 evaluates to* 1 *is negligible (as a function of* λ*).*

Blindness says that it should be infeasible for a malicious signer $\mathcal{S}^*$ to decide which of two messages m_0 and m_1 has been signed first in two executions with an honest user $\mathcal{U}$. This condition must hold, even if $\mathcal{S}^*$ is allowed to choose the public key maliciously [1]. If one of these executions has returned an invalid signature, denoted by $\bot$, then the signer is not informed about the other signature either.

Definition 3. *A blind signature scheme* $\mathsf{BS} = (\mathsf{Gen}, \mathcal{S}, \mathcal{U}, \mathsf{Vrfy})$ *satisfies* blindness *if for any efficient algorithm* $\mathcal{S}^*$ *(working in modes* find*,* issue*, and* guess*) the probability that experiment* $\mathsf{Unblind}^{\mathsf{BS}}_{\mathcal{S}^*}(\lambda)$ *defined in Figure 1 evaluates to* 1 *is negligibly bigger than* $1/2$*.*

A blind signature scheme is secure if it is unforgeable and blind.

3 Towards a Secure Construction

A central idea behind our work is to adapt techniques from secure two-party computation, despite the fact that we *cannot* achieve the traditional notions of secure two-party computation in the standard model with only 2 rounds. Indeed, unfortunately, very few two round protocol techniques are known at all.[3] The high-level idea of our construction is as follows: We employ a two-move secure function evaluation (SFE) protocol to let the signer and user compute a signature on a message chosen by the user. Using Yao's garbled circuits together with Naor-Pinkas OT [32], we get a two-move SFE protocol with the following properties: The user sends the first message, the signer replies, and only the user gets output. The user's input stays secret even in the case of an active malicious signer (as the user does not send any responses to the signer's messages, an active signer is no more powerful than a passive one). The signer's input stays secure against active malicious users. The correctness of the protocol's output is guaranteed against active users and against *passive* signers.

If we use this SFE protocol for signing, we face the following two problems:

(a) Although the signer does not learn the user's inputs, he could cheat in the SFE to make the signatures output by the two users in the blindness game depend on the message in different ways. Even if the SFE would have correctness against active signers, such cheating would still be possible by using different signing keys or randomnesses in the two interactions.
(b) To prove unforgeability of the blind signature scheme, we have to reduce it to the unforgeability of the underlying non-interactive signature scheme. To do so, we need to extract the messages sent by the user in order to feed them into a signing oracle. But in a two-round SFE protocol, we cannot use rewinding to extract the message.

To solve (a), we let the signer commit, as part of his public verification key, to the secret key, and to a random seed to be used when signing. In order to force the signer to actually use that key and randomness, we would like to use a zero-knowledge proof that the SFE was performed correctly and with the right inputs. Unfortunately, two-move zero-knowledge proofs do not exist in the standard model (without CRS). However, there are two-move *witness indistinguishable* proofs, so-called ZAPs. As these are not zero-knowledge, we cannot use them directly (witness indistinguishable proofs might still leak, e.g., the signing key). To make the ZAP "almost zero-knowledge", we introduce a trapdoor: We introduce the possibility of producing a fake proof by using the preimage under some one-way function of a value chosen by the user. A complexity-leveraged simulator can then use this trapdoor, and we can show that the ZAP does not reveal too much.

The solve (b), we again use complexity leveraging: Our SFE protocol only has computational security for the user, so the signer can extract the message m to

[3] We do know of two round witness indistinguishable protocols (ZAPs), which will be useful for us. But as we will see later, ZAPs by themselves are not sufficient for our purposes.

be signed in superpolynomial time T. Thus, we can transform the unforgeability game into one where the signer extracts m, sends it to a signing oracle, and re-inserts the resulting signature back into the SFE. This allows to reduce the unforgeability of the blind signature scheme against the unforgeability of the underlying non-interactive signature scheme. Note however, that the underlying scheme needs to be secure against T-time adversaries in this reduction. Also, standard properties of SFE do not seem to allow us to perform such an extraction and re-insertion. Thus, we define a new property called Alice-extraction-privacy for this purpose; fortunately, Yao's garbled circuits using Naor-Pinkas OTs have this property.

4 Required Primitives

Before presenting our generic construction, we review the required primitives. Most definitions are standard, except that in some cases we require security against superpolynomial adversaries. In these cases we write, e.g., T-one-wayness and mean one-wayness against adversaries running in time $T \cdot poly(\lambda)$. We will now review those primitives and security definitions that are not standard. Complete definitions are given in the full version [21].

ZAPs. A ZAP [15] is a two-round witness-indistinguishable proof system. That is, a ZAP for a language L consists of a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. The first invocation of $\mathcal{V}(1^\lambda)$ is independent of the statement to be proven and outputs a message msg. Given that message, the prover $\mathcal{P}(1^\lambda, \mathsf{msg}, s, w)$ outputs a proof π for statement s with witness w. Finally the verifier $\mathcal{V}(\mathsf{msg}, s, \pi)$ checks the proof π. Notice that this verification only uses msg but no private state from the first invocation of $\mathcal{V}$. In particular, the prover can check on his own whether verification succeeds.

Two-party-computation. We will need a two-move two-party secure function evaluation protocol. Syntactically, such a protocol is described by three PPT algorithms $\mathsf{SFE}_1, \mathsf{SFE}_2, \mathsf{SFE}_3$. The intended use is as follows: Assume that Alice holds a circuit C and Bob holds an input m to that circuit. Then Bob first computes $(\mathsf{sfe}_1, \mathsf{sfest}) \leftarrow \mathsf{SFE}_1(1^\lambda, m)$ and sends sfe_1 to Alice (sfest is Bob's state). Then Alice computes $\mathsf{sfe}_2 \leftarrow \mathsf{SFE}_2(1^\lambda, \mathsf{sfe}_1, C)$ and sends sfe_2 to Bob. Finally, Bob computes the result σ of the computation via $\sigma \leftarrow \mathsf{SFE}_3(1^\lambda, \mathsf{sfe}_2, \mathsf{sfest})$. We require two standard properties, perfect completeness (an honest execution gives the right result with probability 1) and Bob-privacy (from Alice's point of view, different Bob-inputs are computationally indistinguishable). We also require one non-standard property for Alice's security:

Instead of requiring that Bob does not learn anything about the circuit C except for $C(m)$, we require the following: If Alice knows m (which we model by a superpolynomial-time extraction algorithm SFEExt), then instead of applying SFE_2 with circuit C, she can instead compute $\sigma := C(m)$ directly and hardcode the result into the function evaluation using a function $\mathsf{SFEFake}_2$. (This property will be needed so that we can outsource the evaluation of C to a signing oracle later in our proofs.)

Definition 4 (Non-uniform T-Alice-extraction-privacy). *There is a $(T \cdot poly(\lambda))$-time probabilistic algorithm* SFEExt *and a polynomial-time algorithm* $\mathsf{SFEFake}_2$ *such that the following holds:*

For an adversary $\mathcal{A}$, consider the following experiments:

Experiment 1	***Experiment*** 2
$(\mathsf{sfe}_1, C, \mathsf{sfest}) \leftarrow \mathcal{A}(1^\lambda)$	$(\mathsf{sfe}_1, C, \mathsf{sfest}) \leftarrow \mathcal{A}(1^\lambda)$
	$m \leftarrow \mathsf{SFEExt}(1^\lambda, \mathsf{sfe}_1)$
	$\sigma \leftarrow C(m)$
$\mathsf{sfe}_2 \leftarrow \mathsf{SFE}_2(1^\lambda, \mathsf{sfe}_1, C)$	$\mathsf{sfe}_2 \leftarrow \mathsf{SFEFake}_2(1^\lambda, \mathsf{sfe}_1, \sigma)$
$b \leftarrow \mathcal{A}(\mathsf{sfe}_2, \mathsf{sfest})$	$b \leftarrow \mathcal{A}(\mathsf{sfe}_2, \mathsf{sfest})$

Then for any PPT $\mathcal{A}$, $|\Pr[b = 1 : \textit{Experiment 1}] - \Pr[b = 1 : \textit{Experiment 2}]|$ is negligible in λ.

In the fullversion [21], we show that Yao's garbled circuits using Naor-Pinkas OTs [32] satisfies these conditions for any T such that the DDH problem can be decided in time T.

5 Construction and Security Proofs

In this section, we present our construction and show its security.

5.1 Construction

We proceed to define our blind signature scheme. Fix superpolynomial functions T and T' with $T' < T$. In our construction, we assume a one-way function f, a pseudorandom function F, commitment schemes $\mathcal{C}_R, \mathcal{C}_X$, a signature scheme $\mathsf{Sig} = (\mathsf{SigGen}, \mathsf{Sign}, \mathsf{SigVrfy})$, a ZAP $Z = (\mathcal{P}, \mathcal{V})$, and a two-message two-party secure function evaluation protocol $(\mathsf{SFE}_1, \mathsf{SFE}_2, \mathsf{SFE}_3)$. We then define the blind signature scheme $(\mathsf{Gen}, \mathcal{S}, \mathcal{U}, \mathsf{Vrfy})$ as follows:

Key Generation. $\mathsf{Gen}(1^\lambda)$ peforms the following steps:

- $R, S, T \leftarrow \{0,1\}^\lambda$
- $(ssk, svk) \leftarrow \mathsf{SigGen}(1^\lambda; S)$
- $\mathsf{com}_R \leftarrow \mathsf{Com}_R((R, ssk); T)$
- set $sk \leftarrow (svk, ssk, R, S, T)$ and $vk \leftarrow (svk, \mathsf{com}_R)$

Signing. The signature issue protocol between the signer $\mathcal{S}$ and the user $\mathcal{U}$ is as follows:

- $\mathcal{U}$ generates the first message of the SFE protocol $(\mathsf{sfe}_1, \mathsf{sfest}) \leftarrow \mathsf{SFE}_1(1^\lambda, m)$ and the challenge $\mathsf{msg} \leftarrow \mathcal{V}(1^\lambda)$ for the ZAP Z. It picks $x \leftarrow \{0,1\}^\lambda$ uniformly at random, sets $y \leftarrow f(x)$, and sends $(\mathsf{sfe}_1, \mathsf{msg}, y)$ to $\mathcal{S}$.
- $\mathcal{S}$ receives $(\mathsf{sfe}_1, \mathsf{msg}, y)$ from the user. If y is not a valid image of f, then $\mathcal{S}$ sends $\perp$. Otherwise, denote by $C_{ssk,R}(m)$ the circuit computing $\mathsf{Sign}(ssk, m; F_R(m))$. The signer $\mathcal{S}$ picks two random values V, X each of

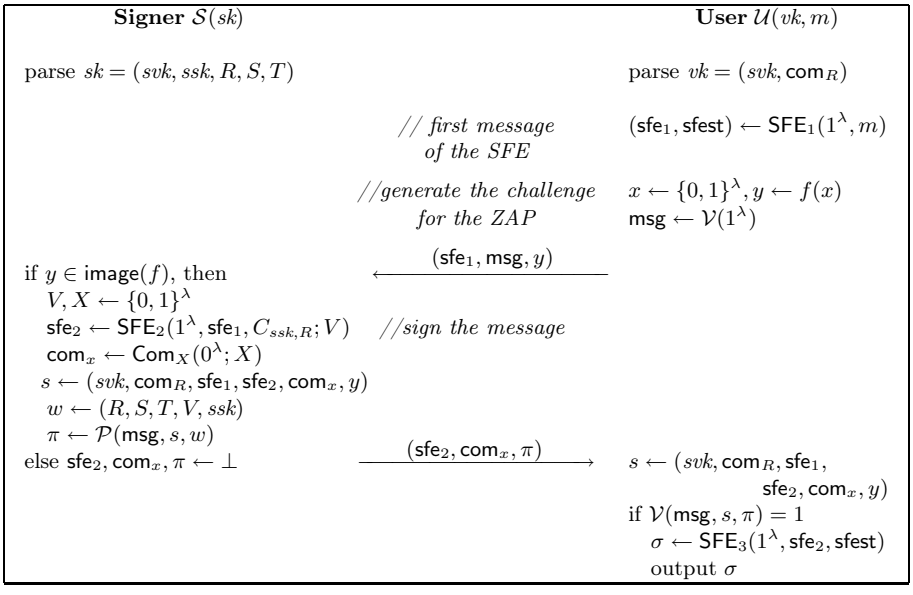

Fig. 2. Issue protocol of the two move blind signature scheme

bit length λ, it computes $\mathsf{sfe}_2 \leftarrow \mathsf{SFE}_2(1^\lambda, \mathsf{sfe}_1, C_{ssk,R}; V)$ and $\mathsf{com}_x \leftarrow \mathsf{Com}_X(0^\lambda; X)$. Then, it generates a proof π (with respect to msg) for the statement $(svk, \mathsf{com}_R, \mathsf{sfe}_1, \mathsf{sfe}_2, \mathsf{com}_x, y) \in L$, where L contains tuples for which there exists either a witness $\omega_1 = (R, S, T, V, ssk)$ such that:

$$\mathsf{sfe}_2 = \mathsf{SFE}_2(1^\lambda, \mathsf{sfe}_1, C_{ssk,R}; V) \bigwedge$$

$$\mathsf{com}_R = \mathsf{Com}_R((R, ssk); T) \bigwedge (ssk, svk) = \mathsf{SigGen}(1^\lambda; S)$$

or there exists a witness $\omega_2 = (x, X)$ such that

$$\mathsf{com}_x = \mathsf{Com}_X(x; X) \bigwedge f(x) = y.$$

$\mathcal{S}$ then sends $(\mathsf{sfe}_2, \mathsf{com}_x, \pi)$ to $\mathcal{U}$.

- $\mathcal{U}$ verifies that π is a valid proof for the statement $(svk, \mathsf{com}_R, \mathsf{sfe}_1, \mathsf{sfe}_2, \mathsf{com}_x, y) \in L$ with respect to msg and the ZAP Z. If this proof fails, then $\mathcal{U}$ outputs $\perp$. Otherwise, it computes the signature $\sigma \leftarrow \mathsf{SFE}_3(1^\lambda, \mathsf{sfe}_2, \mathsf{sfest})$ and outputs σ.

Verification. $\mathsf{Vrfy}(vk, \sigma, m)$ returns $\mathsf{SigVrfy}(svk, \sigma, m)$.

5.2 Security

Theorem 5. *Assume that* Sig *is complete;* $(\mathsf{SFE}_1, \mathsf{SFE}_2, \mathsf{SFE}_3)$ *is correct;* $\mathcal{C}_R, \mathcal{C}_X$ *are complete; and* $Z = (\mathcal{P}, \mathcal{V})$ *is complete. Then the protocol from Section 5.1 is complete.*

Theorem 6. *Assume that f is invertible in time T and has efficiently decidable images; F is a T-pseudorandom function; $(\mathsf{SFE}_1, \mathsf{SFE}_2, \mathsf{SFE}_3)$ is non-uniformly T-Alice-extraction-private; Sig is T-unforgeable; $\mathcal{C}_R$ is T-hiding; $\mathcal{C}_X$ is non-uniformly hiding; the ZAP $Z = (\mathcal{P}, \mathcal{V})$ is non-uniformly computationally witness-indistinguishable.*

Then the blind signature scheme from Section 5.1 is unforgeable.

Theorem 7. *Assume that f is T'-one-way; $(\mathsf{SFE}_1, \mathsf{SFE}_2, \mathsf{SFE}_3)$ is perfectly correct and has Bob-privacy; $\mathcal{C}_R, \mathcal{C}_X$ are perfectly binding; $\mathcal{C}_X$ is T-extractable; ZAP $Z = (\mathcal{P}, \mathcal{V})$ is adaptively sound.*

Then the scheme from Section 5.1 is blind.

Theorem 5 is immediate from the construction of the protocol, and Theorems 6 and 7 will be shown in the next section.

Instantiating the Primitives. The primitives needed in our construction (see Theorems 5, 6, and 7) can be instantiated if the DDH problem is non-uniformly hard, and if subexponentially-hard[4] 1-1 one-way functions with efficiently decidable range exist: Given the one-way functions, we can construct a perfectly hiding non-uniformly T-hiding commitment $\mathcal{C}_R$, a T-unforgeable signature scheme Sig, and a T-pseudorandom function (for some $T \in 2^{\eta^{\Omega(1)}}$). By scaling of the security parameter (such that the used randomness is $< \log T$), we can produce a T-extractable non-uniformly commitment $\mathcal{C}_R$. By similar rescaling, we get a T'-one-way T-time invertible one-way function f. Non-uniform ZAPs can be instantiated from non-uniform DDH [15]. Finally, $(\mathsf{SFE}_1, \mathsf{SFE}_2, \mathsf{SFE}_3)$ can be instantiated given non-uniform one-way functions (for Yao's circuits) and DDH (for Naor-Pinkas OT), see the full version [21], as long as the underlying DDH can be broken in time T. This can again be achieved by rescaling the security parameter.

5.3 Security Proofs

Proof of Unforgeability. The proof idea is the following. We start with a game that corresponds to the unforgeability game of blind signatures and we then gradually change this game such that at the end we can build an adversary against the unforgeability of the underlying signature scheme. The main steps of the proof are the following:

- We apply a complexity leveraging argument. This technique allows us to extract the message m out of the first message of the secure function evaluation protocol. We also use this technique in order to invert the one-way function f and obtain $f^{-1}(y)$.
- We use the external signing oracle in the unforgeability game of the underlying signature scheme to sign the message m.

[4] By subexponentially-hard, we mean that a function $T \in 2^{\eta^{\Omega(1)}}$ exist, such that the primitive is hard against T-time adversaries.

– Instead of computing the second message of the SFE protocol honestly, we use the $\mathsf{SFEFake}_2$ algorithm. These modifications do not change the success probability of the adversary (against the unforgeability of the blind signature scheme) because:
  - the SFE protocol is extraction-private and thus, the attacker cannot tell the difference;
  - the ZAP remains valid as it now uses the previously computed preimage x of f as a witness.

Due to the complexity leveraging, we have to be careful which primitives need superpolynomial-hardness and which do not. In some cases, non-uniform security turns out to be sufficient, even though we use the security of a primitive in a game involving superpolynomial computations.

Proof (of Theorem 6). Assume towards contradiction that the construction from Section 5.1 is not unforgeable. Then there exists a PPT algorithm $\mathcal{U}^*$ that outputs $(\ell+1)$ message/signature pairs (m_i, σ_i) after ℓ executions of the signature issue protocols. This adversary wins if all messages are distinct and all signatures verify under vk. Now, consider the following sequence of games, where the first game Game-0 is the unforgeability game in which we run the game with the forger $\mathcal{U}^*$. Within all games, the first digit of the line numbers is the number of the game (i.e., line 107 in Game-1 corresponds to 007 in Game-0).

Game-0

000 $R, S, T, V_i, X_i \leftarrow \{0,1\}^\lambda$
001 $(ssk, svk) \leftarrow \mathsf{SigGen}(1^\lambda; S)$
002 $\mathsf{com}_R \leftarrow \ldots$
003 $\mathsf{st}_0 \leftarrow \mathcal{U}^*(svk, \mathsf{com}_R)$
004 for $i = 1, \ldots, \ell$
005 $\quad(\mathsf{sfe}_{1,i}, \mathsf{msg}_i, y_i) \leftarrow \mathcal{U}^*(\mathsf{st}_{i-1})$
006 $\quad$if $y_i \in \mathsf{image}(f)$ then
007 $\quad\quad\mathsf{sfe}_{2,i} \leftarrow \mathsf{SFE}_{2,i}(1^\lambda, \mathsf{sfe}_{1,i}, C_{ssk,R})$
008 $\quad\quad\mathsf{com}_{x,i} \leftarrow \mathsf{Com}_X(0^\lambda; X_i)$
009 $\quad\quad s_i \leftarrow (svk, \mathsf{com}_R, C, \mathsf{sfe}_{1,i}, \mathsf{sfe}_{2,i}, \mathsf{com}_{x,i}, y_i)$
010 $\quad\quad w_i \leftarrow (R, S, T, V_i, ssk)$
011 $\quad\quad \pi_i \leftarrow \mathcal{P}(\mathsf{msg}_i, s_i, w_i)$
012 $\quad$else
013 $\quad\quad\mathsf{sfe}_{2,i}, \mathsf{com}_i^x, \pi_i \leftarrow \perp$
014 $\quad\mathsf{st}_i \leftarrow \mathcal{U}^*(\mathsf{sfe}_{2,i}, \mathsf{com}_{x,i}, \pi_i, \mathsf{st}_i)$
015 end for
016 $(m_1, \sigma_1, \ldots, m_{\ell+1}, \sigma_{\ell+1}) \leftarrow \mathcal{U}^*(\mathsf{st}_\ell)$
017 Return 1 iff $\mathsf{SigVrfy}(svk, m_i, \sigma_i) = 1$ for all $i = 1, \ldots, \ell+1$ and $m_i \neq m_j$ for all $i \neq j$

Game-0 $\Rightarrow$ Game-1. We now modify the above game by letting the signer (after Step 005) extract the message $m_i \leftarrow \mathsf{SFEExt}(1^\lambda, \mathsf{sfe}_{1,i})$ from the first message

of the SFE protocol $\mathsf{sfe}_{1,i}$ according to Definition 4. Analogously, $f^{-1}(y)$ is the algorithm that inverts the one-way function f. Both algorithms are running in time T.

Game-1
105a $\boldsymbol{x_i \leftarrow f^{-1}(y_i), m_i \leftarrow \mathsf{SFEExt}(1^\lambda, \mathsf{sfe}_{1,i})}$
108 $\mathsf{com}_{x,i} \leftarrow \mathsf{Com}_X(\boldsymbol{x_i}; X_i)$

The difficulty in showing that the adversary's success probability in both games is the same arises from the point that Step 105a cannot be computed efficiently. Nevertheless, we solve this issue by applying the following (standard) technique. The idea is to consider primitives that are secure against *non-uniform* adversaries. This allows us to perform computations *in advance* that are not feasible in polynomial time. More precisely, consider the commitment scheme $\mathcal{C}_X$ that is non-uniformly hiding. The adversary is allowed to be unbounded as long as it has not received the commitment (since the unbounded computation is captured by the auxiliary input). Once the attacker has obtained the commitment, it runs in polynomial time. The basic idea behind this approach is that it is possible to wire an advice into the circuit that is hard to compute. This observation allows us to perform a computation that is not feasible in polynomial time *before* seeing the commitment. During this step, we extract the message out of the encryption $m_i \leftarrow \mathsf{SFEExt}(1^\lambda, \mathsf{sfe}_{1,i})$ and we invert the one-way function $x_i \leftarrow f^{-1}(y_i)$. Then, we commit to x_i (instead of 0^λ). Note that this is only possible because Step 108 happens after Step 105. This, however, is not quite true because this step happens in a loop. Thus, at some point Step 108 happens before Step 105. To handle this issue, we refine our argument as follows: let Game-$\tilde{1}_i$ be the game where we applied the modifications during the first i iterations but not in iterations $i+1, \ldots, \ell$. Now, the same argument as above shows that Game-$\tilde{1}_i$ and Game-$\tilde{1}_{i+1}$ are indistinguishable for any i (even if i depends on the security parameter). This, however, also implies that Game-$\tilde{1}_0$ and Game-$\tilde{1}_\ell$ are indistinguishable. Furthermore Game-$\tilde{1}_0$ = Game-0 and Game-$\tilde{1}_\ell$ = Game-1, hence Game-0 $\approx$ Game-1 where $\approx$ indicates that the probability that both games output 1 is the same (except for a negligible amount).

Game-1 $\Rightarrow$ Game-2 $\Rightarrow$ Game-3. In the next game, Game-2, we first change the witness of the ZAP Z. That is, we use as a witness the pre-image x_i of the one-way function f that we have inverted in the previous step. Afterwards, in Game-3, we sign the message that was extracted in Game-1, and then use $\mathsf{SFEFake}_2$ in order to make the function evaluation output that signature σ_i.

Game-2
209 $s_i \leftarrow (svk, \mathsf{com}_R, \mathsf{sfe}_{1,i}, \mathsf{sfe}_{2,i}, \mathsf{com}_{x,i}, y_i)$
210 $w_i := \boldsymbol{(x_i, X_i)}$
211 $\pi_i \leftarrow \mathcal{P}(\mathsf{msg}, s_i, \boldsymbol{w_i})$

Game-3
307 $\boldsymbol{\sigma_i \leftarrow \mathsf{Sign}(ssk, m_i; F_R(m_i))}$
307a $\mathsf{sfe}_{2,i} \leftarrow \boldsymbol{\mathsf{SFEFake}_2(1^\lambda, \mathsf{sfe}_{1,i}, \sigma_i)}$

Now, we argue that both modifications do not change the success probability of the adversary $\mathcal{U}^*$ by more than a negligible amount and therefore, Game-0 $\approx$ Game-3. This should follow from the following two observations

- The one-way function f has efficiently decidable images and the signer checks if y_i is a valid image under f in step 006. Thus, according to our construction the witness w_i is a valid witness. Note that according to our construction the witness $w_i := (R, S, T, V_i, ssk)$ used in Game-1 is also valid. Since both witnesses are a valid witness and because we have assumed that the ZAP Z is non-uniformly witness-indistinguishable, it follows that the success probability of $\mathcal{U}^*$ in both games is the same (except for a negligible amount).
- The secure function evaluation scheme is T-Alice-extraction-private. Thus, the adversary $\mathcal{U}^*$ does not notice the difference in the computation of $\mathsf{sfe}_{2,i}$.

We elaborate more on the second point: The only difference between Game-2 and Game-3 is that in the i-th iteration of the loop, in Game-3 we replace $\mathsf{sfe}_{2,i} \leftarrow \mathsf{SFE}_{2,i}(1^\lambda, \mathsf{sfe}_{1,i}, C_{ssk,R})$ (Step 207) by $\mathsf{sfe}_{2,i} \leftarrow \mathsf{SFEFake}_2(1^\lambda, \mathsf{sfe}_{1,i}, \sigma_i) = C_{ssk,R}(m_i)$, where $\sigma_i \leftarrow \mathsf{Sign}(ssk, m_i; F_R(m_i))$ (Steps 307, 307a). By definition of T-Alice-extraction-privacy, it follows that the games are indistinguishable. (Notice that we use *non-uniform* T-Alice-extraction-privacy, the fact that there are superpolynomial computations occurring before the SFE does not limit the applicability of T-Alice-extraction-privacy.) Unfortunately, we cannot apply the arguments justifying the transformations Game-1 $\Rightarrow$ Game-2 $\Rightarrow$ Game-3 directly. The reason is that these arguments are only applicable as long as the games run in polynomial time (or at least those steps of the games occurring after the modifications). In the previous step, however, we have inverted the one-way function and we have extracted the message from the first message of the SFE protocol. Both steps, however, are not computable in polynomial time. We handle this issue by carefully applying a hybrid argument. Now, we apply carefully a hybrid argument over all three games. We omit the details of this hybrid argument.

Game-3 $\Rightarrow$ Game-4. This game is identical to the prior one, but instead of committing to R and ssk, we commit to an all zero string.

Game-4	
401	$\mathsf{com}_R \leftarrow \mathsf{Com}_R(\mathbf{0^\lambda}; T)$

Since the commitment scheme $\mathcal{C}_R$ is T hiding, and since the commitment's randomness T is not used anywhere else, this modification changes the success probability of $\mathcal{U}^*$ only by a negligible amount and thus, Game-3 $\approx$ Game-4 and therefore Game-0 $\approx$ Game-4. (Remember that both f^{-1} and SFEExt and thus all steps in the game run in time T.)

Game-4 $\Rightarrow$ Game-5. In this game, we do not generate the signing key locally, but we build a forger $\mathcal{B}$ against the signature scheme Sig. The difference to the above described games is that it uses its external signing oracle in order to obtain the signature σ_i on the message m_i.

Game-5	
500	$x, T, V_i, X_i \leftarrow \{0,1\}^\lambda$ **(removed R, S)**
501	$\boldsymbol{svk} \leftarrow \widehat{\mathbf{SigGen}}(\mathbf{1^\lambda})$, $\mathsf{com}_R \leftarrow \mathsf{Com}_R(R, ssk; T)$
507	$\boldsymbol{\sigma_i \leftarrow \widehat{\mathbf{Sign}}(m_i)}$

Here $\widehat{\mathsf{SigGen}}$ and $\widehat{\mathsf{Sign}}$ constitute a signing oracle. $\widehat{\mathsf{SigGen}}$ produces a verification key and $\widehat{\mathsf{Sign}}$ signs messages, but whenever a message is submitted that was already signed, $\widehat{\mathsf{Sign}}$ returns the previously produced signature again.

Since F is a T-pseudorandom function, and since R and S are used in Game-4 only in the arguments of SigGen and Sign, it follows that Game-4 $\approx$ Game-5 and thus Game-0 $\approx$ Game-5.

Now, assume that the adversary $\mathcal{U}^*$ wins the unforgeability game Game-0 with non-negligible probability. Then, since Game-0 $\approx$ Game-5, $\mathcal{U}^*$ also wins with non-negligible probability in Game-5.

Then it returns $\ell + 1$ pairs (m_i, σ_i) such that $m_i \neq m_j$ for all $i \neq j$ and $\mathsf{SigVrfy}(vk, m_i, \sigma_i) = 1$ for all $i = 1, \ldots, \ell + 1$. We denote by $Q = (\hat{m}_1, \ldots, \hat{m}_\ell)$ the set of messages that have been asked to the external signing oracle $\widehat{\mathsf{Sign}}$. Since all messages m_i are distinct there exists at least one message $m_j \notin Q$. The forger $\mathcal{B}$ outputs (m_j, σ_j). Since $\mathsf{SigVrfy}(vk, m_i, \sigma_i) = 1$ for all i, we have in particular that the pair (m_j, σ_j) verifies and thus $\mathcal{B}$ succeeds with non-negligible probability. Since $\mathcal{B}$ runs in time $T \cdot poly(\lambda)$, this contradicts the assumption that Sig is T-unforgeable. This concludes the proof.

Proof of Blindness. Before proving the blindness property, we discuss the central points. In our protocol, the privacy of the user (blindness) is preserved by the fact that the secure function evaluation does not reveal the message to be signed (Bob-privacy). We cannot, however, directly apply Bob-privacy: Bob-privacy only guarantees that the users are unlinkable as long as the outputs of the SFE are not revealed. In our setting, however, the outputs are revealed. Thus, we first have to transform the blindness game into one where the signer does not get the signatures output by the users. To achieve this, we use a rewinding argument: Instead of using the signatures produced in the main execution of the blindness game, we rewind the blindness game and use the signatures from the rewound execution (called look-ahead threads). The ZAP sent by the signer guarantees that the signatures are always produced using the same randomness, hence the signatures in the look-ahead threads equal those of the main thread. Finally, we show that the signer can simulate the look-ahead threads on his own. Thus we have a game in which the output of the SFE in the main thread is not used, and we can apply Bob-privacy.

Care needs to be taken with the ZAPs: In our ZAP, one can fake the proof by committing to the preimage $f^{-1}(y)$. Since a ZAP is not a proof of knowledge, the mere fact that the signer does not know $f^{-1}(y)$ is not sufficient to show that the signer cannot fake the ZAP. A complexity-leveraging argument between Games 3 and 4 solves this issue.

Proof (of Theorem 7). We prove this theorem via a sequence of games. In order to facilitate notation, we assume that the malicious signer $\mathcal{S}^*$ is given by a deterministic, stateless algorithm $\mathcal{S}^*$. That is, in its first invocation, $\mathcal{S}^*$ expects an explicit random tape as argument and returns its new internal state st.

In further invocations, $\mathcal{S}^*$ expects the previous internal state st and returns a new internal state st'.

In the blindness game, $\mathcal{S}^*$ has the liberty to schedule the instances of the user in an arbitrary order. In case of a two-move scheme, however, we can fix the ordering. Since $\mathcal{S}^*$ does not receive any response to the message it sends to the user, we can assume that $\mathcal{S}^*$ sends these messages after having gathered all incoming messages from the user. Thus, without loss of generality, $\mathcal{S}^*$ first outputs the public key vk and the challenge messages m_0, m_1, then expects the two incoming blinded messages from the two user instances, and then outputs its responses to the user messages.

With these assumptions about $\mathcal{S}^*$, the blindness game can be reformulated as follows:

Game-0

000 $\mathsf{rand}_{\mathcal{S}^*} \leftarrow \{0,1\}^\infty$, $b \leftarrow \{0,1\}$
001 $(\mathsf{st}, vk, m_0, m_1) \leftarrow \mathcal{S}^*(1^\lambda, \mathsf{rand}_{\mathcal{S}^*})$
002 $\mathsf{usermsg}_0 \leftarrow \mathcal{U}(vk, m_b) \mid \mathsf{usermsg}_1 \leftarrow \mathcal{U}(vk, m_{\bar{b}})$
003 $(\mathsf{st}', \mathsf{signermsg}_0, \mathsf{signermsg}_1) \leftarrow \mathcal{S}^*(\mathsf{st}, \mathsf{usermsg}_0, \mathsf{usermsg}_1)$
004 if fail then $b' \leftarrow \mathcal{S}^*(\mathsf{st}', \bot, \bot)$
005 else $b' \leftarrow \mathcal{S}^*(\mathsf{st}', \sigma_b, \sigma_{\bar{b}})$

In Game-0 and in the following, we use the abbreviation fail for $\sigma_0 = \bot$ or $\sigma_1 = \bot$. Blindness is then equivalent to requiring that $\left|\Pr[b' = b : \text{Game-0}] - \frac{1}{2}\right|$ is negligible. For contradiction, we assume that blindness does not hold, i.e., that $\left|\Pr[b' = b : \text{Game-0}] - \frac{1}{2}\right|$ is non-negligible. Then there exists a polynomial $p = p(\lambda)$ such that $\Pr[b' = b : \text{Game-0}] \geq \frac{1}{2} + 1/p$ for infinitely many λ. (Or $\leq \frac{1}{2} - 1/p$, but in this case we can replace $\mathcal{S}^*$ by an adversary that outputs the negated guess $1 - b'$.)

Game-0 $\Rightarrow$ Game-1. Our first transformation is to make the definition of the user explicit in the blindness game. That is, we replace invocations to $\mathcal{U}$ by its program code. For notational convenience later on, we split the description of Game-0 into the actual game and a subroutine Thread that executes the interaction between $\mathcal{S}^*$ and the users.

Thread$(vk, m_0, m_1, b, \mathsf{st})$

T00 $K_0, K_1, X_0, X_1, E_0, E_1, x_0, x_1 \leftarrow \{0,1\}^\lambda$
T01 $(\mathsf{sfe}_{1,0}, \mathsf{sfest}_0) \leftarrow \mathsf{SFE}_1(1^\lambda, m_b)$ | $(\mathsf{sfe}_{1,1}, \mathsf{sfest}_1) \leftarrow \mathsf{SFE}_1(1^\lambda, m_{\bar{b}})$
T07 $y_0 \leftarrow f(x_0)$ | $y_1 \leftarrow f(x_1)$
T08 $\mathsf{msg}_0 \leftarrow \mathcal{V}(1^\lambda)$ | $\mathsf{msg}_1 \leftarrow \mathcal{V}(1^\lambda)$
T09 $(\mathsf{st}', (\mathsf{sfe}_{2,0}, \mathsf{com}_{x,0}, \pi_0), (\mathsf{sfe}_{2,1}, \mathsf{com}_{x,1}, \pi_1)) \leftarrow \mathcal{S}^*(\mathsf{st}, (\mathsf{sfe}_{1,0}, \mathsf{msg}_0, y_0), (\mathsf{sfe}_{1,1}, \mathsf{msg}_1, y_1))$
T10 $s_0 \leftarrow (svk, \mathsf{com}_R, \mathsf{sfe}_{1,0}, \mathsf{sfe}_{2,0}, \mathsf{com}_{x,0}, y_0)$ | $s_1 \leftarrow (svk, \mathsf{com}_R, \mathsf{sfe}_{1,1}, \mathsf{sfe}_{2,1}, \mathsf{com}_{x,1}, y_1)$
T11 if $\mathcal{V}(\mathsf{msg}_0, s_0, \pi_0) = 1$ then | if $\mathcal{V}(\mathsf{msg}_0, s_1, \pi_1) = 1$ then
T12 $\quad \sigma_b \leftarrow \mathsf{SFE}_3(1^\lambda, \mathsf{sfe}_{2,0}, \mathsf{sfest}_0)$ else $\sigma_b \leftarrow \bot$ | $\quad \sigma_{\bar{b}} \leftarrow \mathsf{SFE}_3(1^\lambda, \mathsf{sfe}_{2,1}, \mathsf{sfest}_1)$ else $\sigma_{\bar{b}} \leftarrow \bot$

Return$(\sigma_0, \sigma_1, \mathsf{st}')$

Game-1

100 $b \leftarrow \{0,1\}$, $\mathsf{rand}_{\mathcal{S}^*} \leftarrow \{0,1\}^\infty$
101 $(\mathsf{st}, vk, m_0, m_1) \leftarrow \mathcal{S}^*(1^\lambda; \mathsf{rand}_{\mathcal{S}^*})$ with $vk = (svk, \mathsf{com}^R)$
102 $(\sigma_0, \sigma_1, \mathsf{st}') \leftarrow \mathsf{Thread}(vk, m_0, m_1, b, \mathsf{st})$
103 if fail then $b' \leftarrow \mathcal{S}^*(\mathsf{st}', \bot, \bot)$
104 else $b' \leftarrow \mathcal{S}^*(\mathsf{st}', \sigma_0, \sigma_1)$

Since we have only restructured the game, we have $\Pr[b' = b : \mathsf{Game\text{-}0}] = \Pr[b' = b : \mathsf{Game\text{-}1}]$.

Game-1 ⇒ Game-2. Game-2 is identical to Game-1 except for the following modifications. Once both user instances have computed their signatures, i.e., right after Step 202, we reset the malicious signer $\mathcal{S}^*$ to the point where it has returned the two messages, i.e., to after Step 201. We repeat this procedure p times. Since $\mathcal{S}^*$ is deterministic and stateless, this can be modeled by running the subroutine Thread p times using the same initial state st for $\mathcal{S}^*$ in each thread. Each thread uses a fresh random bit $\hat{b}$ to assign the messages to the user instances. We refer to the first execution as the *main thread* (representing the original blindness game) and to the other p executions as *look-ahead threads*. Observe that this rewinding *only* involves Step T01 to T12. The other steps are part of the blindness game. In particular, $\mathcal{S}^*$ gets in Step 204 the signatures σ_0 and σ_1 from the main thread.

Game-2

200 $b \leftarrow \{0,1\}, \mathsf{rand}_{\mathcal{S}^*} \leftarrow \{0,1\}^\infty, \boldsymbol{\hat{b}_1, \ldots, \hat{b}_p \leftarrow \{0,1\}}$
201 $(vk, m_0, m_1, \mathsf{st}) \leftarrow \mathcal{S}^*(1^\lambda; \mathsf{rand}_{\mathcal{S}^*})$ with $vk = (svk, \mathsf{com}^R)$
202 $(\sigma_0, \sigma_1, \mathsf{st}') \leftarrow \mathsf{Thread}(vk, m_0, m_1, b, \mathsf{st})$
202a $\boldsymbol{(\sigma^{\mathsf{la}}_{0,i}, \sigma^{\mathsf{la}}_{1,i}, \mathsf{st}'_i) \leftarrow \mathsf{Thread}(vk, m_0, m_1, \hat{b}_i, \mathsf{st})}$ **for** $\boldsymbol{i = 1, \ldots, p}$
203 if fail then $b' \leftarrow \mathcal{S}^*(\mathsf{st}', \bot, \bot)$
204 else $b' \leftarrow \mathcal{S}^*(\mathsf{st}', \sigma_0, \sigma_1)$

The success probability of $\mathcal{S}^*$ in both games is clearly the same, because the results of the look-ahead threads are not used and Thread has no side effects. That is, $\Pr[b' = b : \mathsf{Game\text{-}1}] = \Pr[b' = b : \mathsf{Game\text{-}2}]$.

Now, we bound the probability that both user instances in the main thread return valid signatures, but at least one user algorithm in *each* look-ahead threads fails. Observe that this includes aborting parties, decryption attempts that fail, or the case where a certain ZAP may be invalid. Recall that $\mathsf{fail} = 1$ if $\sigma_0 = \bot$ or $\sigma_1 = \bot$. We define $\mathsf{fail}^{\mathsf{la}}_i$ analogously to fail, i.e., $\mathsf{fail}^{\mathsf{la}}_i = 1$ if $\sigma^{\mathsf{la}}_{0,i} = \bot$ or $\sigma^{\mathsf{la}}_{1,i} = \bot$.

Lemma 8. *Denote by* bad *the event that* $\mathsf{fail}^{\mathsf{la}}_i$ *holds for all* $i = 1, \ldots, p$ *but that* fail *does not hold. The probability that* bad *happens in* Game-2 *is less or equal* $\frac{1}{p+1}$.

Game-2 ⇒ Game-3. In this game, we set $b' = 0$ whenever the the main thread produces valid signatures ($\neg$fail) but the look-ahead threads do not (fail_1 and $\ldots$ and fail_p).

Game-3
303 if fail then $b' \leftarrow \mathcal{S}^*(\mathsf{st}', \perp, \perp)$
303a **else if** $\mathsf{fail}_1^{\mathsf{la}}$ **and ... and** $\mathsf{fail}_p^{\mathsf{la}}$ **then** $b' \leftarrow 0$
304 else $b' \leftarrow \mathcal{S}^*(\mathsf{st}', \sigma_0, \sigma_1)$

Notice that the else-branch in line 303a is only taken if the event bad occurs. This happens with probability at most $\frac{1}{p+1}$ by Lemma 8. Thus, $\mathcal{S}^*$'s avantage reduces at most by $\frac{1}{p+1}$, i.e., $\Pr[b' = b : \text{Game-3}] \geq \Pr[b' = b : \text{Game-2}] - \frac{1}{p+1}$.

Game-3 $\Rightarrow$ Game-4. In this hybrid, we do not give the adversary $\mathcal{S}^*$ the signatures from the main thread, but the ones from one of the successful look-ahead threads. That is, we choose a random g with $\neg\mathsf{fail}_g^{\mathsf{la}}$ and we give the signatures $(\sigma_{0,g}, \sigma_{1,g})$ to $\mathcal{S}^*$. Notice that we only need to do this if $\mathsf{fail}_1^{\mathsf{la}} \wedge \cdots \wedge \mathsf{fail}_p^{\mathsf{la}}$ does not hold (otherwise line 404 would not have been reached), so there is always at least one such g.

Game-4
404 else $\boldsymbol{g \leftarrow \{i : \neg\mathsf{fail}_i^{\mathsf{la}}\}}$, $b' \leftarrow \mathcal{S}^*(\mathsf{st}', \boldsymbol{\sigma_{0,g}}, \boldsymbol{\sigma_{1,g}})$

We first argue that all messages $\sigma_0, \sigma_{0,g}$ have the same value (if defined), and analogously for $\sigma_1, \sigma_{1,g}$: Due to the adaptive soundness of the ZAP, we have that with overwhelming probability the statements proven by the ZAPs are true. That is, for each thread it holds that $\mathsf{com}_{x,0}$ contains a preimage of y under f or that the message $\mathsf{sfe}_{2,0}$ is constructed correctly. The first occurs only with negligible probability, otherwise by using the T'-extractability of $\mathcal{C}_X$ we could build a T'-time inverter for f, breaking the T'-onewayness of f. Thus $\mathsf{sfe}_{2,0}$ is constructed correctly in all thread with overwhelming probability. Analogously for $\mathsf{sfe}_{2,1}$. Due to the perfect correctness of SFE (and the perfect binding property of $\mathcal{C}_R$), this implies that all signatures $\sigma_{i,g}$ are produced by applying the same circuit $C_{ssk,R}$ with the same ssk and R to the message m_i. Thus, giving the malicious signer $\mathcal{S}^*$ the signatures returned from the gth look-ahead thread does not change its success probability by more than a negligible amount. That is, $|\Pr[b' = b : \text{Game-3}] - \Pr[b' = b : \text{Game-4}]|$ is negligible.

Game-4 $\Rightarrow$ Game-5. Now, we modify Game-4 to Game-5 by returning $(\perp, \perp)$ to $\mathcal{S}^*$ only if one of the ZAPs in the main thread failed. Formally, we check this by validating the ZAP π. In what follows, we denote by $\mathsf{msg}_0(\mathsf{st}'), s_0^{u'}(\mathsf{st}'), \pi_0(\mathsf{st}')$ the messages that are needed to verify the ZAP as seen by $\mathcal{S}^*$. We assume, w.l.o.g., that these messages are stored in the state st'.

Game-5
503 if $\boldsymbol{\mathcal{V}(\mathsf{msg}_0(\mathsf{st}'), s_0^u(\mathsf{st}'), \pi_0(\mathsf{st}')) = 0}$ **or** $\boldsymbol{\mathcal{V}(\mathsf{msg}_1(\mathsf{st}'), s_1^u(\mathsf{st}'), \pi_1(\mathsf{st}')) = 0}$ then $b' \leftarrow \mathcal{S}^*(\mathsf{st}', \perp, \perp)$

Due to the soundness of the ZAP, this modification does not change the success probability of $\mathcal{S}^*$ by more than a negligible amount.

Game-5 $\Rightarrow$ Game-6. In this game, we build an attacker $\mathcal{B}$ that simulates the look-ahead threads and the malicious signer $\mathcal{S}^*$ locally.

$\mathcal{B}(\mathsf{st}, \mathsf{st}')$

B00 $\hat{b}_1, \ldots, \hat{b}_p \leftarrow \{0,1\}$
B01 $(\sigma^{\mathsf{la}}_{0,i}, \sigma^{\mathsf{la}}_{1,i}, \mathsf{st}'_i) \leftarrow \mathsf{Thread}(vk, m_0, m_1, \hat{b}_i, \mathsf{st})$ for $i = 1, \ldots, p$
B02 if $\mathcal{V}(\mathsf{msg}_0(\mathsf{st}'), s_0^{u'}(\mathsf{st}'), \pi_0(\mathsf{st}')) = 0$ or
$\mathcal{V}(\mathsf{msg}_1(\mathsf{st}'), s_1^{u}(\mathsf{st}'), \pi_1(\mathsf{st}')) = 0$ then $b' \leftarrow \mathcal{S}^*(\mathsf{st}', \bot, \bot)$
B03 else if fail_1 and ... and fail_p then $b' \leftarrow 0$
B04 else $g \leftarrow \{i : \neg\mathsf{fail}_i\}$, $b' \leftarrow \mathcal{S}^*(\mathsf{st}', \sigma_{0,g}, \sigma_{1,g})$

The game looks as follows (here, we show the whole Game-6, not just the lines changed with respect to Game-5):

Game-6
600 $b \leftarrow \{0,1\}$, $\mathsf{rand}_{\mathcal{S}^*} \leftarrow \{0,1\}^\infty$
601 $(vk, m_0, m_1, \mathsf{st}) \leftarrow \mathcal{S}^*(1^\lambda; \mathsf{rand}_{\mathcal{S}^*})$ with $vk = (svk, \mathsf{com}^R)$
602 $(\sigma_0, \sigma_1, \mathsf{st}) \leftarrow \mathsf{Thread}(vk, m_0, m_1, b, \mathsf{st})$
603 $\boldsymbol{b' \leftarrow \mathcal{B}(\mathsf{st}, \mathsf{st}')}$

Clearly, the success probability of $\mathcal{B}$ in this game is equal to the success probability of $\mathcal{S}^*$ in the previous game since we have just moved some of the computations of Game-5 into $\mathcal{B}$.

Game-6 $\Rightarrow$ Game-7. Now, we modify the algorithm Thread only for the main thread (recall that all other threads are computed by $\mathcal{B}$ locally). Our modification removes all dependencies on the input message m. That is, we let the user algorithm compute the first message of the SFE protocol on 0^λ instead of m. Additionally, we remove the computation of the signatures σ_0, σ_1 that are never used.

Thread'

T'01	$(\mathsf{sfe}_{1,0}, \mathsf{sfest}_0) \leftarrow \mathsf{SFE}_1(1^\lambda, \boldsymbol{0^\lambda})$	$(\mathsf{sfe}_{1,1}, \mathsf{sfest}_1) \leftarrow \mathsf{SFE}_1(1^\lambda, \boldsymbol{0^\lambda})$
T'11–T'12	**(removed)**	**(removed)**

Since the SFE scheme is Bob-private, the success probability of $\mathcal{B}$ remains the same (except for a negligible amount). This, however, means that the entire transcript is independent of the message.

Now, we obtain the following contradiction concluding that advantage of $\mathcal{S}^*$ in Game-0 is less or equal $\frac{1}{p+1} + \mathsf{negl}(\lambda)$, where $\mathsf{negl}(\lambda)$ is a negligible function. In Game-0, however, we assumed that $\mathcal{S}^*$ wins the blindness game with advantage at least $\nu \geq 1/p$ (infinitely often). Since $1/p > 1/(p+1) + \mathsf{negl}$ for sufficiently large λ we obtain a contradiction. Thus, our initial assumption that $\mathcal{S}^*$ succeeds with non-negligible probability was wrong and therefore, our construction is blind.

Acknowledgments. We thanks the anonymous reviewers for valuable comments and Masayuki Abe for his useful feedback on this merged paper. Dominique Unruh was supported by European Social Fund's Doctoral Studies and Internationalisation Programme DoRa. Dominique Schröder is also supported by the Emmy Noether Program Fi 940/2-1 of the German Research Foundation (DFG). Part of this work was supported by CASED (www.cased.de). Amit Sahai is supported in part from a DARPA/ONR PROCEED award, NSF grants 0916574 and 0830803, a

Xerox Foundation Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant.

References

1. Abdalla, M., Namprempre, C., Neven, G.: On the (im)possibility of blind message authentication codes. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 262–279. Springer, Heidelberg (2006)
2. Abe, M.: A secure three-move blind signature scheme for polynomially many signatures. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 136–151. Springer, Heidelberg (2001)
3. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-preserving signatures and commitments to group elements. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 209–236. Springer, Heidelberg (2010)
4. Abe, M., Haralambiev, K., Ohkubo, M.: Signing on elements in bilinear groups for modular protocol design. IACR ePrint 2010/133 (2010)
5. Abe, M., Ohkubo, M.: A framework for universally composable non-committing blind signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 435–450. Springer, Heidelberg (2009)
6. Bellare, M., Namprempre, C., Pointcheval, D., Semanko, M.: The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. Journal of Cryptology 16(3), 185–215 (2003)
7. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 31–46. Springer, Heidelberg (2002)
8. Brands, S., Paquin, C.: U-prove cryptographic specification v1.0 (March 2010), `http://connect.microsoft.com/site642/Downloads/DownloadDetails.aspx?Dow%nloadID=26953`
9. Brands, S.A.: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press, Cambridge (2000)
10. Camenisch, J., Groß, T.: Efficient attributes for anonymous credentials. In: ACM CCS 2008, pp. 345–356. ACM Press, New York (2008)
11. Camenisch, J., Koprowski, M., Warinschi, B.: Efficient blind signatures without random oracles. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 134–148. Springer, Heidelberg (2005)
12. Camenisch, J., Neven, G., Shelat, A.: Simulatable adaptive oblivious transfer. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 573–590. Springer, Heidelberg (2007)
13. Chaum, D.: Blind signatures for untraceable payments. In: CRYPTO 1982, pp. 199–203. Plenum Press, New York (1983)
14. Chaum, D.: Blind signature system. In: CRYPTO 1983, p. 153. Plenum Press, New York (1984)
15. Dwork, C., Naor, M.: Zaps and their applications. SIAM J. Comput. 36(6), 1513–1543 (2007)
16. Fischlin, M.: Round-optimal composable blind signatures in the common reference string model. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 60–77. Springer, Heidelberg (2006)
17. Fischlin, M., Schröder, D.: Security of blind signatures under aborts. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 297–316. Springer, Heidelberg (2009)

18. Fischlin, M., Schröder, D.: On the impossibility of three-move blind signature schemes. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 197–215. Springer, Heidelberg (2010)
19. Fuchsbauer, G.: Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. IACR ePrint 2009/320 (2009)
20. Garg, S., Rao, V., Sahai Round, A.: optimal blind signatures in the standard model (2011) (manuscript)
21. Garg, S., Rao, V., Sahai, A., Schröder, D., Unruh, D.: Round optimal blind signatures. IACR ePrint (2011)
22. Ghadafi, E., Smart, N.: Efficient two-move blind signatures in the common reference string model. IACR ePrint 2010/568 (2010)
23. Hazay, C., Katz, J., Koo, C.Y., Lindell, Y.: Concurrently-secure blind signatures without random oracles or setup assumptions. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 323–341. Springer, Heidelberg (2007)
24. Horvitz, O., Katz, J.: Universally-composable two-party computation in two rounds. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 111–129. Springer, Heidelberg (2007)
25. Juels, A., Luby, M., Ostrovsky, R.: Security of blind digital signatures (extended abstract). In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 150–164. Springer, Heidelberg (1997)
26. Katz, J., Schröder, D., Yerukhimovich, A.: Impossibility of blind signature from one-way permutation. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 615–629. Springer, Heidelberg (2011)
27. Kiayias, A., Zhou, H.S.: Concurrent blind signatures without random oracles. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 49–62. Springer, Heidelberg (2006)
28. Kiayias, A., Zhou, H.S.: Equivocal blind signatures and adaptive UC-security. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 340–355. Springer, Heidelberg (2008)
29. Lindell, Y.: Bounded-concurrent secure two-party computation without setup assumptions. In: STOC 2003, pp. 683–692. ACM Press, New York (2003)
30. Lindell, Y.: Lower bounds for concurrent self composition. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 203–222. Springer, Heidelberg (2004)
31. Meiklejohn, S., Shacham, H., Freeman, D.M.: Limitations on transformations from composite-order to prime-order groups: The case of round-optimal blind signatures. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 519–538. Springer, Heidelberg (2010)
32. Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: SODA 2001, pp. 448–457 (2001)
33. Okamoto, T.: Efficient blind and partially blind signatures without random oracles. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 80–99. Springer, Heidelberg (2006)
34. Pass, R.: Limits of provable security from standard assumptions. In: STOC 2011. ACM Press, New York (to appear, 2011)
35. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology 13(3), 361–396 (2000)
36. Prabhakaran, M., Rosen, A., Sahai, A.: Concurrent zero knowledge with logarithmic round-complexity. In: FOCS 2002, pp. 366–375. IEEE, Los Alamitos (2002)
37. Rückert, M.: Lattice-based blind signatures. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 413–430. Springer, Heidelberg (2010)
38. Schröder, D., Unruh, D.: Round optimal blind signatures. IACR ePrint (2011)
39. Schröder, D., Unruh, D.: Security of blind signatures revisited. IACR ePrint (2011)