On the Impossibility of Three-Move Blind Signature Schemes

Marc Fischlin and Dominique Schröder

Darmstadt University of Technology, Germany
`www.minicrypt.de`

Abstract. We investigate the possibility to prove security of the well-known blind signature schemes by Chaum, and by Pointcheval and Stern in the standard model, i.e., without random oracles. We subsume these schemes under a more general class of blind signature schemes and show that finding security proofs for these schemes via black-box reductions in the standard model is hard. Technically, our result deploys meta-reduction techniques showing that black-box reductions for such schemes could be turned into efficient solvers for hard non-interactive cryptographic problems like RSA or discrete-log. Our approach yields significantly stronger impossibility results than previous meta-reductions in other settings by playing off the two security requirements of the blind signatures (unforgeability and blindness).

Keywords Blind signature scheme, black-box reduction, meta-reduction, random oracle, round complexity.

1 Introduction

Blind signatures [11] implement a carbon copy envelope allowing a signer to issue signatures for messages such that the signer's signature on the envelope is imprinted onto the message in the sealed envelope. In particular, the signer remains oblivious about the message (blindness), but at the same time no additional signatures without the help of the signer can be created (unforgeability).

Many blind signature schemes have been proposed in the literature, e.g., [1, 2, 6, 11, 12, 16, 17, 19, 20, 22–24, 26, 27, 29], with varying security and efficiency characteristics. The arguably most prominent examples are the schemes by Chaum [11] based on RSA and the ones by Pointcheval and Stern [27] based on the discrete logarithm problem, RSA and factoring. Both approaches admit a security proof in the random oracle model, in the case of Chaum's scheme the "best" known security proofs currently even requires the one-more RSA assumption [5].

Here we investigate the possibility of instantiating the random oracles in the schemes by Chaum and by Pointcheval and Stern, and of giving a security proof based on standard assumptions like RSA or discrete logarithm. Although both schemes are different in nature we can subsume them under a more general pattern of blind signature schemes where

- blindness holds in a statistical sense, i.e., where even an unbounded malicious signer cannot link executions of the issuing protocol to message-signature pairs,
- the interactive signature issuing has three (or less) moves, and
- one can verify from the communication between a possibly malicious signer and an honest user if the user is eventually able to derive a valid signature from the interaction.

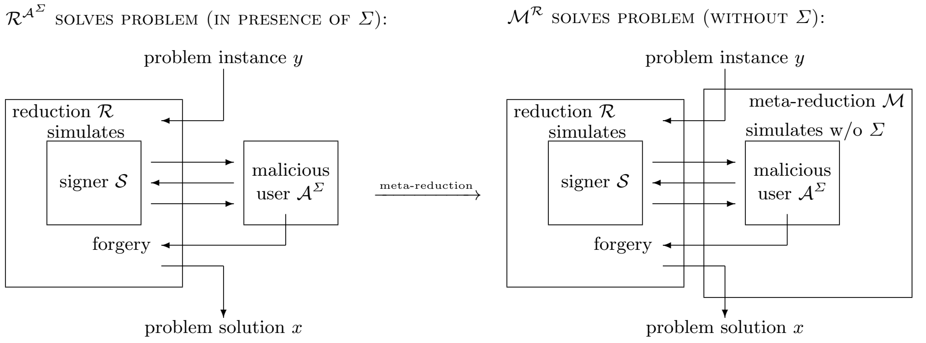
We note that the construction by Boldyreva [6] based on the one-more Gap Diffie-Hellman problem in the random oracle model also obeys these three properties such that any impossibility result immediately transfers to this scheme as well. The third property, which we coin signature derivation check, basically guarantees that blindness still holds if the user fails to produce a signature in the postprocessing step, after the actual interaction with the signer has been completed. Common notions of blindness do not provide any security guarantee in this case (see [13, 17] for further discussions).

1.1 The Idea Behind our Result

Given a blind signature scheme with the properties above we can show that for such schemes finding black-box reductions from successful forgers to an *arbitrary* non-interactive cryptographic problem (like RSA, discrete-log, or general one-wayness or collision-resistance) is infeasible. The key idea to our result is as follows. Assume that we are given a three-move blind signature scheme as above and a reduction $\mathcal{R}$ reducing unforgeability to a presumably hard problem (given only black-box access to an alleged forger). Vice versa, if the problem is indeed infeasbile, then the reduction therefore shows that the scheme is unforgeable.

Our approach is to show that the existence of a reduction $\mathcal{R}$ as above already violates the assumption about the hardness of the underlying problem. Our starting point is to design an oracle Σ with unlimited power and a "magic" adversary $\mathcal{A}^{\Sigma}$ breaking the unforgeability of the blind signature scheme with the help of Σ. By assumption, the reduction $\mathcal{R}$ with access to $\mathcal{A}^{\Sigma}$ is then able to break the underlying cryptographic problem (see the left part of Figure 1). Note that, at this point, we are still in a setting with an all-powerful oracle Σ and the non-interactive problem may indeed be easy relative to this oracle, without contradicting the presumed hardness in the standard model.

Now we apply meta-reduction techniques, as put forward for example in [7, 9, 14, 28], to remove the oracle Σ from the scenario. Given $\mathcal{R}$ we show how to build a meta-reduction $\mathcal{M}$ (a "reduction for the reduction") to derive an efficient solver for the problem, but now without any reference to the magic adversary and Σ (right part of Figure 1). To this end, the meta-reduction $\mathcal{M}$ fills in for adversary $\mathcal{A}^{\Sigma}$ and simulates the adversary's actions without Σ, mainly by resetting the reduction $\mathcal{R}$ appropriately. We have then eventually derived an algorithm $\mathcal{M}^{\mathcal{R}}$ solving the underlying non-interactive problem in the standard



$\mathcal{R}^{\mathcal{A}^\Sigma}$ solves problem (in presence of Σ):
problem instance y
reduction $\mathcal{R}$ simulates
signer $\mathcal{S}$
malicious user $\mathcal{A}^\Sigma$
forgery
problem solution x
meta-reduction
$\mathcal{M}^{\mathcal{R}}$ solves problem (without Σ):
problem instance y
reduction $\mathcal{R}$ simulates
meta-reduction $\mathcal{M}$ simulates w/o Σ
signer $\mathcal{S}$
malicious user $\mathcal{A}^\Sigma$
forgery
problem solution x

Fig. 1: Meta-reduction technique: The black-box reduction $\mathcal{R}$ on the left hand side uses the adversary $\mathcal{A}^\Sigma$ against unforgeability to solve an instance y of the non-interactive problem. The meta-reduction $\mathcal{M}$ on the right hand side then uses $\mathcal{R}$ to solve the problem from scratch, i.e., by simulating $\mathcal{A}^\Sigma$ without Σ. For this, the meta-reduction $\mathcal{M}$ exploits the blindness property of the scheme.

model, meaning that the problem cannot be hard. In other words, there cannot exist such a reduction $\mathcal{R}$ to a hard problem.[1]

At this point it seems as if we have not used the blindness property of the scheme and that the idea would paradoxically also apply to regular signature schemes (for which we know secure constructions based on any one-way function). This is not the case. The blindness subtly guarantees that the meta-reduction's simulation of the adversary is indistinguishable from the actual behavior of $\mathcal{A}^\Sigma$, such that the success probabilities of $\mathcal{R}^{\mathcal{A}^\Sigma}$ and of $\mathcal{M}^{\mathcal{R}}$ are close. For these two cases to be indistinguishable, namely $\mathcal{R}$ communicating with $\mathcal{A}^\Sigma$ or with $\mathcal{M}$, we particularly rely on the fact that blindness holds relative to the all-powerful oracle Σ used by $\mathcal{A}$, as in case of statistically-blind signature schemes.

The reason that our approach only applies to blind signature schemes with at most three moves originates from the resetting strategy of our meta-reduction. In a three-move scheme the user sends a single message only, such that resetting the reduction in such an execution allows our meta-reduction to choose independent user messages in each run. This is essential for our proof. In schemes with four or more moves the user sends at least two messages and the second message may

[1] We consider very general reductions running *multiple* instances of the adversary in a concurrent and *resetting* manner, covering all known reductions for blind signatures in the literature. Yet, since the meta-reduction itself uses rewinding techniques, we somewhat need to restrict the reduction in regard of the order of starting and finishing resetted executions of different adversarial instances (called resetting with restricted cross-resets). This saves us from an exponential running time for $\mathcal{M}$. For example, any resetting reduction running only a single adversarial instance at a time obeys our restriction.

then depend on the first one, e.g., the scheme may implement a commit-and-prove strategy with four moves.

1.2 The Essence of Our Meta-Reduction and Impossibility of Random Oracle Instantiations

There are essentially two approaches in the literature to derive black-box separations like ours. One class of black-box separation results (e.g., [21, 30, 31]) basically starts with an oracle Σ breaking any cryptographic primitive of type A, like a collision-resistant hash function, but adds an oracle Π implementing another primitive of type B like a one-way function (and which cannot be broken by Σ). Here, the cryptographic primitives in question are usually treated as black boxes.

The other approach uses meta-reductions [4, 7–9, 14, 28] and usually treats the adversary as a black box. In our case, we show that no black-box reduction to *arbitrary* (non-interactive) cryptographic problems can exist. This includes common assumptions like the RSA and discrete logarithm problem, but also more general notions of one-way functions and collision-resistant hash functions. Compared to oracle-based separations and previous meta-reduction techniques our result gives the following two advantages:

– Oracle separations involving a "positive" oracle Π implementing a primitive often do not allow to make statements about the possibility of deriving schemes based on concrete primitives such as RSA or discrete-log. The latter primitives have other properties which could potentially be exploited for a security proof, like homomorphic properties. This limitation does not hold for our results.
– Meta-reduction separations such as [4, 8, 28] consider the impossibility of reductions from secure encryption or signatures to a given RSA instance. Yet, they often fall short of providing any meaningful claim if other assumptions enter the security proof, e.g., the result in [28] does not hold anymore if two RSA instances are given or an additional collision-resistant hash function is used in the design. In comparison, our general approach covers such cases as we can easily combine non-interactive problems P_1, P_2 into more complex problems like $P_1 \vee P_2$ and $P_1 \wedge P_2$, requiring to break one of the two problems and both of them, respectively.

The latter advantage emerges because our meta-reduction plays off unforgeability against blindness. This idea may be useful in similar settings where two or more security properties are involved, to provide stronger separation results for meta-reductions.

The broader class of problems ruled out by our meta-reduction also allows to make meaningful claims when it comes to the possibility instantiating the random oracle in the blind signature schemes. Namely, our separation indicates the limitations of hash function options (assuming some restriction on the resets of the reductions, mentioned in the previous section):

Any hash function whose security can be proven by black-box reduction to hard non-interactive problems does not allow a black-box reduction from the unforgeability of the blind signature scheme to hard non-interactive problems, such as RSA or discrete-logarithm.

This can be seen as follows. Any reduction from the unforgeability either breaks the underyling non-interactive problem like RSA or discrete-log, or breaks some security property of the hash function. The latter, in turn, yields a nested reduction from the unforgeability of the blind signature scheme to the non-interactive problem on which the hash function is based. One only needs to ensure that this nested reduction falls within our admissible reset strategy. This is clearly true if the security property of the hash function is given by a hard non-interactive problem itself, like one-wayness or collision-resistance, or allows a suitable reduction to these problems or RSA, discrete-log etc.

1.3 Extension to Computational Blindness

In principle our result extends to computationally-blind signature schemes but the conditions are arguably more restrictive than in the statistical case. First, recall that blindness needs to hold relative to the forgery oracle Σ, i.e., the powerful forgery oracle must not facilitate the task of breaking blindness. While this comes "for free" in the statistical case, in the computational case one must assume that unforgeability and blindness of the scheme are somewhat independent. This is true for instance for Fischlin's scheme [16], but there are also examples where blindness and unforgeability are correlated, as in Abe's scheme [1] where unforgeability is based on the discrete-log problem and blindness on the DDH problem.

Second, given that the scheme is computationally-blind relative to Σ we still rely on the signature derivation check. One can easily design computationally-blind schemes infringing this property, say, by letting the user sent a public key and having the signer encrypt each reply (we are not aware of any counter example in the statistical case). On the other hand, these signature derivation checks are very common, e.g., besides the schemes above the ones by Okamoto [26] and by Fischlin [16] too have this property.

Third, since we have to change the forgery oracle Σ for the computational case, we also need a key-validity check which allows to verify if a public key has a matching secret key (i.e., if there is a key pair with this public key in the range of the key generating algorithm). For schemes based on discrete-logarithm this usually boils down to check that the values are group elements. Given that these three conditions are met we show that our techniques carry over to the computational case.

1.4 Related Work

In a sense, our results match the current knowledge about the round complexity of blind signature schemes. Nowadays, the best upper bound to build (non-concurrently) secure blind signatures are four moves for the standard model, i.e., neither using random oracles nor set-up assumptions like a common reference string. This is achieved by a protocol of Okamoto [26] based on the 2SDH bilinear Diffie-Hellman assumption. Any schemes with three moves or less either use the random oracle model [6, 11, 27] or a commom reference string [2, 16, 19].

We note that Lindell [25] rules out any concurrently secure blind signature scheme in the standard model, independently of any cryptographic assumption. Hence, it seems that two-move schemes —which are concurrently secure by nature— are impossible in the standard model. However, Lindell's impossibility result only refers to the stronger (black-box) *simulation-based* definition of blind schemes and can indeed be circumvented by switching to the common *game-based* definition, as shown by [20]. In contrast, our result holds with respect to game-based definitions and also covers three-move schemes, thus showing that such blind signature schemes may be hard to build even under this relaxed notion.

The recent results by Brown [8] and Bresson et al. [4] show meta-reduction based separations of the one-more RSA and one-more discrete-logarithm problem from their regular counterparts. The conclusion in [4] is that it should be hard to find a security proof for Chaum's scheme and the Pointcheval-Stern schemes using only these regular assumptions. As mentioned before, the meta-reductions in [4, 8] are limited in the sense that they either cannot rewind (as in [8]) or can only forward the input RSA or discrete log problem (as in [4]). Our approach, however, considers arbitrary hard non-interactive problems and is robust with respect to the combination of several underlying assumptions.

We also remark that the well-known three-move lower bound for non-trivial zero-knowledge [18] is not known to provide a lower bound for blind signature schemes. The intuitively appealing idea of using the blind signature scheme as a commitment scheme in such zero-knowledge proofs unfortunately results in proofs which require more than three moves. This is even true if we start with a two-move blind signature scheme where a "hidden" third move is required for the initial transmission of the signer's public key. In addition, the game-based notion of blind signatures is not known to yield appropriate zero-knowledge simulators.

Organization. We start with the definition of blind signature schemes in Section 2. In Section 3 we discuss our notion of black-box reductions to hard problems. Before presenting our main result in Section 5 where we show the hardness of finding black-box reductions from unforgeability to non-interactive problems we first discuss a simpler case for restricted reductions in Section 4 to provide some intuition about the general result. Due to the space restrictions, we have delegated the case of computational blindness, as well as most of the proofs, to the full version.

2 Blind Signatures

To define blind signatures formally we introduce the following notation for interactive execution between algorithms $\mathcal{X}$ and $\mathcal{Y}$. By $(a, b) \leftarrow \langle \mathcal{X}(x), \mathcal{Y}(y) \rangle$ we denote the joint execution, where x is the private input of $\mathcal{X}$, y defines the private input for $\mathcal{Y}$, the private output of $\mathcal{X}$ equals a, and the private output of $\mathcal{Y}$ is b. We write $\mathcal{Y}^{\langle \mathcal{X}(x), \cdot \rangle^\infty}(y)$ if $\mathcal{Y}$ can invoke an unbounded number of executions of the interactive protocol with $\mathcal{X}$ in sequential order. Accordingly, $\mathcal{X}^{\langle \cdot, \mathcal{Y}(y_0) \rangle^1, \langle \cdot, \mathcal{Y}(y_1) \rangle^1}(x)$ can invoke sequentially ordered executions with $\mathcal{Y}(y_0)$ and $\mathcal{Y}(y_1)$, but interact with each algorithm only once.

Definition 1 (Blind Signature Scheme). *A blind signature scheme consists of a tuple of efficient algorithms* $\mathsf{BS} = (\mathsf{KG}, \langle \mathcal{S}, \mathcal{U} \rangle, \mathsf{Vf})$ *where*

Key Generation. $\mathsf{KG}(1^n)$ *generates a key pair* (sk, pk).

Signature Issuing. *The joint execution of the algorithms* $\mathcal{S}(sk)$ *and* $\mathcal{U}(pk, m)$ *for message* $m \in \{0,1\}^n$ *generates an output* σ *of the user,* $(\perp, \sigma) \leftarrow \langle \mathcal{S}(sk), \mathcal{U}(pk, m) \rangle$, *where possibly* $\sigma = \perp$.

Verification. $\mathsf{Vf}(pk, m, \sigma)$ *outputs a bit.*

It is assumed that the scheme is complete, *i.e., for any* $(sk, pk) \leftarrow \mathsf{KG}(1^k)$, *any message* $m \in \{0,1\}^n$ *and any* σ *output by* $\mathcal{U}$ *in the joint execution of* $\mathcal{S}(sk)$ *and* $\mathcal{U}(pk, m)$ *we have* $\mathsf{Vf}(pk, m, \sigma) = 1$.

Security of blind signature schemes requires two properties, namely unforgeability and blindness [22, 27]. A malicious user $\mathcal{U}^*$ against unforgeability tries to generate $k+1$ valid message-signatures pairs after at most k completed interactions with the signer, where the number of interactions is adaptively determined by the user during the attack. The blindness condition says that it should be infeasible for a malicious signer $\mathcal{S}^*$ to decide upon the order in which two messages m_0 and m_1 have been signed in two executions with an honest user $\mathcal{U}$.

Definition 2 (Secure Blind Signature Scheme). *A blind signature scheme* $\mathsf{BS} = (\mathsf{KG}, \langle \mathcal{S}, \mathcal{U} \rangle, \mathsf{Vf})$ *is called secure if the following holds:*

Unforgeability. *For any efficient algorithm* $\mathcal{U}^*$ *the probability that experiment* $\mathsf{Forge}^{\mathsf{BS}}_{\mathcal{U}^*}$ *evaluates to* 1 *is negligible (as a function of* n*) where*

```
***Experiment*** $\mathsf{Forge}^{\mathsf{BS}}_{\mathcal{U}^*}$
    $(sk, pk) \leftarrow \mathsf{KG}(1^n)$
    $((m_1, \sigma_1), \ldots, (m_{k+1}, \sigma_{k+1})) \leftarrow \mathcal{U}^{*\langle \mathcal{S}(sk), \cdot \rangle^\infty}(pk)$
    Return 1 iff
        $m_i \neq m_j$ for $1 \leq i < j \leq k+1$, and
        $\mathsf{Vf}(pk, m_i, \sigma_i) = 1$ for all $i = 1, 2, \ldots, k+1$, and
        at most $k$ interactions with $\langle \mathcal{S}(sk), \cdot \rangle^\infty$ were completed.
```

Computational resp. Statistical Blindness. *For any (efficient resp. computationally unbounded) algorithm* $\mathcal{S}^*$ *working in modes* find, issue *and* guess, *the probability that the following experiment* $\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{S}^*}$ *evaluates to* 1 *is negligibly close to* $1/2$, *where*

```
***Experiment*** $\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{S}^*}$
    $(pk, m_0, m_1, st_{find}) \leftarrow \mathcal{S}^*(find, 1^n)$
    $b \leftarrow \{0,1\}$
    $st_{issue} \leftarrow \mathcal{S}^{*\langle\cdot,\mathcal{U}(pk,m_b)\rangle^1,\langle\cdot,\mathcal{U}(pk,m_{1-b})\rangle^1}(issue, st_{find})$
        and let $\sigma_b, \sigma_{1-b}$ denote the (possibly undefined) local outputs
        of $\mathcal{U}(pk, m_b)$ resp. $\mathcal{U}(pk, m_{1-b})$.
    set $(\sigma_0, \sigma_1) = (\perp, \perp)$ if $\sigma_0 = \perp$ or $\sigma_1 = \perp$
    $b^* \leftarrow \mathcal{S}^*(guess, \sigma_0, \sigma_1, st_{issue})$
    return 1 iff $b = b^*$.
```

We remark that, even if occassionally not mentioned, all algorithms in this paper receive the security parameter 1^n as additional input.

3 Hard Problems and Black-Box Reductions

In order to prove the security of a cryptographic protocol, usually reduction techniques are used. A reduction from a cryptographic protocol to an underlying problem shows that breaking the protocol implies breaking the underlying problem. A reduction is *black-box* if it treats the adversary and/or the underlying primitive as an oracle. Reingold et al. [30] call reductions which use both the adversary and the primitive merely as an oracle *fully-black-box*, whereas *semi-black-box* reductions work for any efficient adversaries (whose code the reduction may access) as long as the primitive is black-box.

In our case we only need the orthogonal requirement to semi-black-box reductions, namely that the reduction treats the adversary as an oracle but we do not make any assumption about the representation of the underlying primitive. The reduction we consider works for any kind of non-interactive primitive (i.e., in which one gets an instance as input and outputs a solution without further interaction):

Definition 3 (Hard Non-Interactive Problem). *A non-interactive (cryptographic) problem $P = (I, V)$ consists of two efficient algorithms:*

Instance generation $I(1^n)$**.** *The instance generation algorithm takes as input the security parameter 1^n and outputs an instance y.*

Instance Verification $V(x, y)$**.** *The instance verification algorithm takes as input a value x as well as an instance y of a cryptographic problem, and outputs a decision bit.*

We call a cryptographic problem P hard *if the following condition is fulfilled:*

Hardness. *We say that an algorithm $\mathcal{A}$* solves the cryptographic problem *P if the probability that $\mathcal{A}$ on input $y \leftarrow I(1^n)$ outputs x' such that $V(x', y) = 1$, is non-negligible. We say that the problem P is hard if no efficient algorithm solves it.*

Note that in the definition above we do not impose any completeness requirement on the cryptographic problem. The reason is that reductions from the security of blind signatures must work for arbitrary problems, and in particular to the ones with non-trivial completeness conditions.

The notion of a non-interactive cryptographic problem clearly covers such popular cases like the RSA problem, the discrete logarithm problem, or finding collisions for hash functions. It also comprises more elaborate combination of such problems, e.g., if P_0, P_1 are two non-interactive problems then so are $P_0 \wedge P_1$ and $P_0 \vee P_1$ (with the straightforward meaning requiring to solve both problems or at least one of them).

4 Warm Up: Impossibility Result for Vanilla Reductions

To give some intuition about our technique we first consider the simpler case of *vanilla* reductions. This type of reduction only runs a single execution with the adversary (without rewinding) and, if communicating with an honest user, makes the user output a valid signature with probability 1. This means that a vanilla reduction takes advantage of the magic adversary and its output, instead of solving the problem on its own. We then augment our result in the next section to deal with resetting reductions running multiple adversarial instances.

4.1 Preliminaries

For our impossibility result we need another requirement on the blind signature scheme, besides statistically blindness. This property says that one can tell from the public data and communication between a malicious signer and an honest user whether the user is able to compute a valid signature or not.

For instance, in Chaum's scheme the honest user sends a value y and receives z from the signer, and the user is able to compute a signature σ for an arbitrary message m if and only if $z^e = y \bmod N$. This is easily verifiable with the help of the public key and the communication. The scheme of Pointcheval and Stern implements the signature derivation check already in the user algorithm.[2] Analogous derivation checks occur in the schemes by Okamoto and by Fischlin. More formally:

Definition 4 (Signature-Derivation Check). *A blind signature scheme* BS *allows (computational resp. statistical) signature-derivation checks if there exists an efficient algorithm* SDCh *such that for any (efficient resp. unbounded) algorithm* $\mathcal{S}^*$ *working in modes* find *and* issue *the probability that the experiment* $\mathsf{SigDerCheck}^{\mathsf{BS}}_{\mathcal{S}^*,\mathsf{SDCh}}$ *evaluates to* 1 *is negligible, where*

[2] The signature derivation check is given by the user's local verification $a = g^R h^S y^e$, where the values a, r, R, S are exchanged during the signature issuing protocol and the values g, h, y are part of the public key.

```
***Experiment*** $\mathsf{SigDerCheck}^{\mathsf{BS}}_{\mathcal{S}^*,\mathsf{SDCh}}$
    $(pk, m, \mathit{st}) \leftarrow \mathcal{S}^*(\mathit{find}, 1^n)$
    $(\perp, \sigma) \leftarrow \langle \mathcal{S}^*(\mathit{issue}, \mathit{st}), \mathcal{U}(pk, m)\rangle$
        *where* $\mathit{trans}$ *denotes the communication between* $\mathcal{S}^*$, $\mathcal{U}$
    $c \leftarrow \mathsf{SDCh}(pk, \mathit{trans})$
    *return* 1 *if* $\sigma \neq \perp$ *and* $c = 0$, *or if* $\sigma = \perp$ *but* $c = 1$.
```

In the computational case, if the above holds even if $\mathcal{S}^*$ gets access to an oracle Σ then we say that the scheme has computational signature-derivation checks relative to Σ. (In the statistical case $\mathcal{S}^*$ could simulate Σ internally, such that granting access to Σ is redundant.)

The notion in some sense augments the blindness property of blind signature schemes to the case that the user algorithm fails to produce a valid signature in the final local step. The common notion of blindness does not provide any security in this case (because the malicious signer does not receive any of the signatures if the user fails only then). See [17] for more discussions and solutions. Here, the signature derivation check provides something stronger, as it can be efficiently performed by anyone and holds independently of the user's message.

Next we introduce a weaker notion than blindness which is geared towards our separation result. Informally, a blind signature scheme has so-called *transcript-independent signatures* if one cannot associate a transcript to a signature. This is formalized by comparing signatures generated via an execution with a malicious signer and signatures generated "magically" via an oracle Σ producing the signature for a message from the public key and the transcript of the first execution. The intuition behind the following experiment is that the malicious signer has to distinguish whether the second signature σ_b results from the signature issuing protocol, or if the oracle Σ derived the signature σ_b from the transcript of the signature issuing protocol where the honest user gets as input the message m_0.

Definition 5 (Transcript-Independent Signatures). *A blind signature scheme* BS *has (computationally resp. statistically) transcript-independent signatures with respect to Σ if for any (efficient resp. unbounded) algorithm $\mathcal{S}^*_{\mathit{trans}}$ the probability that the experiment* $\mathsf{trans\text{-}ind}^{\mathsf{BS}}_{\mathcal{S}^*_{\mathit{trans}},\Sigma}(n)$ *evaluates to 1 is negligibly close to 1/2, where*

```
***Experiment*** $\mathsf{trans\text{-}ind}^{\mathsf{BS}}_{\mathcal{S}^*_{\mathsf{trans}},\Sigma}(n)$:
    $b \leftarrow \{0, 1\}$
    $(pk, \mathit{st}_1, m_{-1}, m_0) \leftarrow \mathcal{S}^{*,\Sigma}_{\mathit{trans}}(\mathit{init}, 1^n)$
    $\mathit{st}_2 \leftarrow \mathcal{S}^{*,\Sigma,\langle\cdot,\mathcal{U}(pk,m_{-1})\rangle^1,\langle\cdot,\mathcal{U}(pk,m_0)\rangle^1}_{\mathit{trans}}(\mathit{issue}, \mathit{st}_1)$
        *let* $\sigma_{-1}$ *and* $\sigma_0$ *be the local outputs of the users in the two*
        *executions (possibly* $\sigma_{-1} = \perp$ *and/or* $\sigma_0 = \perp$*)*
        *and let* $\mathit{trans}_{-1}$ *be the transcript of the left execution*
    *set* $m_1 = m_0$ *and compute* $\sigma_1 \leftarrow \Sigma(pk, \mathit{trans}_{-1}, m_1)$
    *set* $(\sigma_{-1}, \sigma_0, \sigma_1) = (\perp, \perp, \perp)$ *if* $\sigma_{-1} = \perp$ *or* $\sigma_0 = \perp$ *or* $\sigma_1 = \perp$
    $b^* \leftarrow \mathcal{S}^{*,\Sigma}_{\mathit{trans}}(\mathit{guess}, \mathit{st}_2, m_{-1}, \sigma_{-1}, m_b, \sigma_b)$
    *return* 1 *iff* $b = b^*$.
```

To define our generic forgery oracle Σ allowing $\mathcal{A}$ to break unforgeability we first outline the idea for the case of Chaum's blind signature scheme. Assume that the adversary has already obtained a valid signature for some message m' by communicating with the signer. Let $\mathsf{trans} = (y, z)$ denote the transcript of this communication. Algorithm $\Sigma(pk, \mathsf{trans}, m)$ for $m \neq m'$ then searches some randomness r such that the user's first message for m and r matches y in the transcript, i.e., $H(m)r^e \bmod N = y$. Such an r exists by the perfect blindness and the signature derivation check.[3]

The above example can be generalized to any blind signature scheme and the following generic forgery oracle (which only depends on the blind signature scheme in question):

Definition 6 (Generic Forgery Oracle). *For a statistically-blind signature scheme* BS *the generic forgery oracle* $\Sigma(pk, \mathsf{trans}, m)$ *performs the following steps:*

```
enumerate all values $r$ such that
    the user algorithm $\mathcal{U}(pk, m)$ for randomness $r$ generates the same
    transcript $\mathsf{trans}$ when fed with the same signer messages as in $\mathsf{trans}$;
    also store all signatures $\sigma$ the user's algorithm generates in these executions.
select a value $r$ of the set at random and return the corresponding signature $\sigma$
(or return $\perp$ if there is no such $r$).
```

Proposition 1. *Every statistically blind signature scheme, which has statistical signature-derivation checks, also has statistical transcript-independent signatures with respect to the generic forgery oracle* Σ.

The proof appears in the full version. The idea is that we can safely exchange the order of messages m_{-1}, m_0 in the transcript-independence experiment because of the blindness property. Then oracle Σ in this experiment simply computes another signature for $m_1 = m_0$ from the transcript for a run with the same message m_0 (instead of m_{-1}). By construction of Σ this is perfectly indistinguishable from the original signature derived from this transcript. We note that the signature derivation check and the statistical blindness ensure that failures of Σ do not interfere with the blindness definition (where there are only two executions with the user instances).

Given the generic forgery oracle Σ we can now define the "magic" adversary which first plays an honest users communicating with the signer once. If this single execution yields a valid signature (which is certainly the case when interacting with the genuine signer, but possibly not when interacting with the reduction), then the adversary generates another valid message-signature pair without interaction but using Σ as a subroutine instead.

[3] Note that blindness for Chaum's scheme is only guaranteed if the user can verify that the exponent e is relatively prime to $\varphi(N)$, say, if e is a prime larger than N; only then is guaranteed that the function $(\cdot)^e \bmod N$ really is a permutation.

Definition 7 (Magic Adversary). *The magic adversary $\mathcal{A}$ for input pk and with oracle access to the generic forgery oracle Σ and communicating with an oracle $\langle\mathcal{S}(sk),\cdot\rangle^1$ is described by the following steps:*

```
pick random messages $m'_0, m'_1 \leftarrow \{0,1\}^n$
run an execution $\langle\mathcal{S}(sk),\mathcal{U}(pk,m'_0)\rangle$ in the role of an honest user
    to obtain $\sigma'_0$ and let $\mathsf{trans}'_0$ be the corresponding transcript
if $\mathsf{Vf}(pk,m'_0,\sigma'_0)=1$ then let $\sigma'_1 \leftarrow \Sigma(pk,\mathsf{trans}'_0,m'_1)$ else set $\sigma'_1 \leftarrow \bot$
return $(m'_0,\sigma'_0,m'_1,\sigma'_1)$
```

By the completeness of the blind signature scheme the magic adversary, when attacking the honest signer, returns two valid message-signature pairs, with probability negligibly close to 1 (there is a probability of at most 2^{-n} that the adversary outputs identical pairs for $m'_0 = m'_1$). We also remark that the magic adversary, when attacking the actual scheme, applies the forgery oracle to derive a signature for the second message using the transcript of the first signature issuing protocol.

4.2 Impossibility Result

The following theorem states that vanilla black-box reductions to (non-interactive) cryptographic problems do not provide a meaningful security statement. That is, if there was such a reduction then the underlying problem would already be easy. Since we only deal with non-resetting reductions the claim even holds for schemes with arbitrary round complexity (instead of three-move schemes):

Theorem 1. *Let* BS *be a statistically blind signature scheme that allows statistical signature-derivation checks. Then there is no vanilla black-box reduction from unforgeability of* BS *to a hard non-interactive problem.*

Proof. For sake of readability we divide the reduction $\mathcal{R}$ into steps, according to the black-box simulation of the magic adversary in which $\mathcal{R}$ takes over the role of the signer: in mode init the reduction outputs the public key pk and in mode $\mathsf{msg}i$ the reduction creates the i-th protocol message msg$i$ of the signer. After getting the adversary's signatures σ_0,σ_1 in the post-processing step final the reduction outputs a putative solution x' for its input y. In each step the reduction also outputs some state information which is passed on to the next stage.

Analogously to the reduction $\mathcal{R}$ we denote by $\mathsf{msg}j$ the step of the honest user $\mathcal{U}$ which on input a public key pk, a message m and the previous message msg$i$ of the signer, outputs message msg$j$ sent to the signer. Likewise, in mode finish the user creates the signature from its state and the final message sent by the signer.

Description of the Meta-Reduction. The meta-reduction $\mathcal{M}$ works as follows (see Figure 2 for the case of three moves). It gets as input an instance y of the problem. It start to simulate the reduction $\mathcal{R}$ on y to derive a public key

```
**Meta-reduction** $\mathcal{M}(y)$
let $(pk, \mathsf{st_{init}}) \leftarrow \mathcal{R}(\mathsf{init}, y)$
let $(\mathrm{msg}_1, \mathsf{st_{msg1}}) \leftarrow \mathcal{R}(\mathsf{msg1}, \mathsf{st_{init}})$
choose $m_0 \leftarrow \{0,1\}^n$                                                        | choose $m_1 \leftarrow \{0,1\}^n$
let $(\mathrm{msg2}_0, \mathsf{st}^0_{\mathrm{msg2}}) \leftarrow \mathcal{U}(\mathsf{msg2}, pk, m_0, \mathrm{msg1})$      | let $(\mathrm{msg2}_1, \mathsf{st}^1_{\mathrm{msg2}}) \leftarrow \mathcal{U}(\mathsf{msg2}, pk, m_1, \mathrm{msg1})$
let $(\mathrm{msg3}_0, \mathsf{st}^0_{\mathrm{msg3}}) \leftarrow \mathcal{R}(\mathsf{msg3}, \mathsf{st}_{\mathrm{msg1}}, \mathrm{msg2}_0)$ | let $(\mathrm{msg3}_1, \mathsf{st}^1_{\mathrm{msg3}}) \leftarrow \mathcal{R}(\mathsf{msg3}, \mathsf{st}_{\mathrm{msg1}}, \mathrm{msg2}_1)$
let $\sigma_0 \leftarrow \mathcal{U}(\mathsf{finish}, \mathsf{st}^0_{\mathrm{msg2}}, \mathrm{msg3}_0)$                | let $\sigma_1 \leftarrow \mathcal{U}(\mathsf{finish}, \mathsf{st}^1_{\mathrm{msg2}}, \mathrm{msg3}_1)$
output $x' \leftarrow \mathcal{R}(\mathsf{final}, \mathsf{st}^0_{\mathrm{msg3}}, m_0, \sigma_0, m_1, \sigma_1)$
```

Fig. 2: Meta-Reduction for Vanilla Reduction (three moves), where $\mathsf{trans}_0 = (\mathrm{msg1}, \mathrm{msg2}, \mathrm{msg3})$ denotes the transcript of the first execution.

---

pk as well as the first message msg1 on behalf of the signer and a state $\mathsf{st}_{\mathrm{msg1}}$. Algorithm $\mathcal{M}$ first completes an instance of the signature issuing protocol with $\mathcal{R}$ using the program of the honest user on input a random message m_0 from $\{0,1\}^n$ and some randomness r. Afterwards, it selects another message m' from $\{0,1\}^n$ at random together with some independent randomness r' and resets the reduction to the point where $\mathcal{R}$ has returned the first message of the signature issuing protocol. As before, $\mathcal{M}$ executes the honest user algorithm on m' using the randomness r'.

Now, if the meta-reduction obtains two valid signatures σ_0, σ_1 from both executions, then it hands the pairs (m_0, σ_0), (m_1, σ_1) to the reduction which then outputs some x'. The meta-reduction returns x' and stops. For brevity we often write $\mathcal{R}^{\mathcal{M}}(y)$ for this interaction.

Analysis of the Meta-Reduction. The final step is to show that the reduction $\mathcal{R}$ successfully outputs a solution x', even if given the pairs from $\mathcal{M}$ instead of receiving them from the magic adversary. For this it suffices to show that

$$\mathrm{Prob}\left[y \leftarrow I(1^n), x' \leftarrow \mathcal{R}^{\mathcal{M}}(y) : V(x', y) = 1 \;\middle|\; \mathcal{M}\right]$$

is non-negligible. As outlined above, for this we exploit the transcript-independence of signatures.

Assume to the contrary that the reduction $\mathcal{R}$ outputs a valid solution x' with non-negligible probability if $\mathcal{R}$ receives two message-signature pairs (m_0, σ_0), (m_1, σ_1) from the magic adversary,

$$\mathrm{Prob}\left[y \leftarrow I(1^n), x' \leftarrow \mathcal{R}^{\mathcal{A}}(y) : V(x', y) = 1 \;\middle|\; \mathcal{A} \text{ magic}\right] \not\approx 0,$$

but succeeds only with negligible probability if the message-signature pairs are generated by $\mathcal{M}$:

$$\mathrm{Prob}\left[y \leftarrow I(1^n), x' \leftarrow \mathcal{R}^{\mathcal{M}}(y) : V(x', y) = 1 \;\middle|\; \mathcal{M}\right] \approx 0.$$

Then we construct an adversary $\mathcal{S}^*_{\mathsf{trans}}$ who breaks the transcript independence of signatures in experiment $\mathsf{trans\text{-}ind}^{\mathsf{BS}}_{\mathcal{S}^*, \Sigma}(n)$.

Description of Adversary $\mathcal{S}^*_{\mathsf{trans}}$. Informally, the adversary relays the first execution between the reduction and the external user instance and resets to reduction afterwards to answer the second execution. Afterwards $\mathcal{S}^*_{\mathsf{trans}}$ receives two message-signature pairs without knowing whether the second signature σ_0 has been derived from the signature issuing protocol or with the help of Σ. We then use the result of the reduction to distinguish this case.

More formally, the adversary $\mathcal{S}^*_{\mathsf{trans}}$ generates an instance $y \leftarrow I(n)$ of a cryptographic problem P. It simulates $\mathcal{R}$ in a black-box way, which for input y initially outputs a public key pk as well as the first message msg1 and some state information $\mathsf{st}_{\mathsf{msg1}}$. The algorithm $\mathcal{S}^*_{\mathsf{trans}}$ selects two random message $m_{-1}, m_0 \in \{0,1\}^n$ and outputs pk, m_{-1}, m_0 according to the transcript-independence experiment. It stores the first message (from $\mathcal{R}$ to $\mathcal{U}$) and relays the communication between the reduction $\mathcal{R}$ and the first external user instance $\mathcal{U}(pk, m_{-1})$. Then the adversary resets $\mathcal{R}$ to the point where $\mathcal{R}$ has returned msg1 and forwards the communication between $\mathcal{R}$ and $\mathcal{U}$.

After having finished both executions $\mathcal{S}^*_{\mathsf{trans}}$ receives two (valid) signatures (σ_{-1}, σ_0) and runs the reduction $\mathcal{R}$ in mode final on input $(\mathsf{st}^0_{\mathrm{msg3}}, m_{-1}, \sigma_{-1}, m_0, \sigma_0)$ to obtain a putative solution x' of the cryptographic problem P. The final output of the adversary is $b^* \leftarrow V(x', y)$.

Analysis of $\mathcal{S}^*_{\mathsf{trans}}$. For the analysis recall that the magic adversary, after a single interaction, outputs two message-signature pairs (with the help of Σ). In fact, taking the message-signature pairs (m_{-1}, σ_{-1}) of the first execution together with the message-signature pair (m_0, σ_0) derived from Σ in experiment $\mathsf{trans\text{-}ind}^{\mathsf{BS}}_{\mathcal{S}^*,\Sigma}(n)$ corresponds exactly to the behavior of the magic adversary $(b = 0)$. Here we take advantage of the fact that the second execution with the user cannot fail (and force the signatures to be undefined) by our assumption about the vanilla reduction always making the honest user derive a signature.

On the other hand, during the issuing protocol with the honest user $\mathcal{U}$, the adversary $\mathcal{S}^*_{\mathsf{trans}}$ resets $\mathcal{R}$ and uses in the second execution the prefix msg1 (obtained during the signature generation of (m_{-1}, σ_{-1})) in experiment $\mathsf{trans\text{-}ind}^{\mathsf{BS}}_{\mathcal{S}^*,\Sigma}(n)$. Therefore the message-signature pairs $(m_{-1}, \sigma_{-1}), (m_b, \sigma_b)$ are computed in the same way as the meta-reduction $\mathcal{M}$ does $(b = 1)$. Note that the additional run of Σ in the transcript-independence experiment cannot make the three signatures invalid (except with negligible probability), because of the statistical blindness and the signature derivation checks. More specifically, the statistical blindness guarantees that the transcript generated with $\mathcal{U}$ for message m_{-1} is (almost surely) also a potential transcript for $m_0 = m_1$ used by Σ. Furthermore, the signature derivation check tells us that, independently of the message, the transcript allows the user to derive a signature (such that Σ, too, will find a valid random string r for the simulated user with a valid signature). This fact is stated more formally in the full version. For simplicity we neglect the small error for Σ returning an invalid signature in the analysis below.

We obtain for the probability that $\mathcal{S}^*_{\mathsf{trans}}$ outputs the right bit $b^* = b$:

$$\mathrm{Prob}[\, b^* = b] = \tfrac{1}{2} + \tfrac{1}{2} \cdot (\mathrm{Prob}[\, b^* = 1 \mid b = 1] - \mathrm{Prob}[\, b^* = 1 \mid b = 0])$$

According to our construction, $b = 0$ corresponds to the case where the simulation mimics the behavior of the magic adversary, and $b = 1$ the setting involving the meta-reduction. Furthermore, the adversary $\mathcal{S}^*_{\mathsf{trans}}$ returns $b^* = 1$ in the case that the reduction $\mathcal{R}$ returns a valid solution x' of y. Hence,

$$\begin{aligned}\mathrm{Prob}\left[b^* = 1 \mid b = 1\right] &- \mathrm{Prob}\left[b^* = 1 \mid b = 0\right] \\ &= \mathrm{Prob}\left[y \leftarrow I(1^n), x' \leftarrow \mathcal{R}^{\mathcal{A}}(y) : V(x', y) = 1 \,\middle|\, \mathcal{A} \text{ magic}\right] \\ &\qquad - \mathrm{Prob}\left[y \leftarrow I(1^n), x' \leftarrow \mathcal{R}^{\mathcal{M}}(y) : V(x', y) = 1 \,\middle|\, \mathcal{M}\right].\end{aligned}$$

By assumption the difference is non-negligible (because the first probability is non-negligible and we have assumed that the second probability is negligible). This, however, contradicts the transcript independence of signatures. □

5 Impossibility Result for Statistically Blind Signature Schemes

Here we discuss more general reductions which may reset the adversary and run several nested executions with multiple copies of the adversary.

5.1 Preliminaries

To build our meta-reduction we will reset the reduction continuously. That is, whenever the reduction expects a forgery from an instance of the magic adversary, we freeze the scenario and branch into a loop in which the meta-reduction seeks a second valid message-signature pair. In order to avoid an exponential blow-up in the running time of such rewinding executions [15], we consider slightly restricted reductions.

Resetting Reductions with Restricted Cross-Resets. Any reduction in our case is allowed to run $q = q(n)$ concurrent executions with the copies of the adversary, each copy resetting at most q times, except that the reduction has to finish the interaction in the order according to the arrival of the second messages of the signature issue protocol. That is, consider a three-move signature issuing run of the reduction with a copy of the adversary playing the honest user. Assume that the reduction receives the second message in this execution (which has been sent by the adversary resp. user), and call this execution pending from then on. We say that the reduction successfully finishes this pending execution if it sends the third message of the protocol such that the user is able to derive a valid signature.

The cross-reset restriction now demands that, if the reduction ever finishes a pending execution successfully, then there is no other execution which has become pending and has been finished successfully meanwhile. In other words, between the pending state of an execution and its completion the reduction may not receive the second message and complete any other execution (for which the user can compute a signature). We remark that the reduction may decide to entirely abort a pending execution and is still allowed to finish other pending executions, as long as the user is unable to produce a signature from that interaction. A formal definition appears in the full version.

Note that the scheduling of reductions with restricted cross-resets is related to so-called bounded concurrent (and resettable) executions [3]. In m-bounded concurrent executions the number of instances running simultaneously is bounded by some fixed function $m = m(n)$ where the bound itself is known by the protocol. We do not put any a-priori bound on the number of concurrently running executions, because the number q of such instances depends on the reduction and is not bounded by a fixed polynomial. We merely restrict the way successful executions are finished. We also note that we can extend our proof below to allow a constant number of successfully finished executions between pending runs, but state and prove the simpler version for sake of readability.

q-wise Independent functions. An adequate measure to thwart reset attacks are usually pseudorandom functions (e.g., as in [10]). The idea is to make the randomness of the adversary depend on the communication by computing it as the output of the pseudorandom function for the communication. In this case, resetting the adversary essentially yields runs with independent random choices. Here, we use the same idea but can fall back to the weaker requirement of q-wise independent functions in order to avoid the additional assumption that pseudorandom functions exist.

We note that using q-wise independent functions instead of pseudorandom functions makes the adversary now depend on the reduction. Namely, below we use q as the number of maximal resets per row. However, since we deal with black-box reductions this is admissible. We also remark that we can overcome this dependency by using pseudorandom functions instead of q-wise independent function.

The New Magic Adversary. We use again the generic forgery oracle from the vanilla case. But here we augment our "new" magic adversary through a q-wise independent function (i.e., the random hash function h is given by parts of the adversary's randomness). Informally, the adversary again runs the issuing protocol with the signer in the role of the honest user once. However, it now generates the message (and the user's randomness) as the result of the q-wise independent function applied to the public key and the first message of the signer. Again, in the case that the single execution yields a valid signature, then the magic adversary here also creates another valid signature.

As we will later view Σ to be an integral part of the magic adversary and thus let the adversary provide the randomness $s \in \{0,1\}^{\psi(n)}$ required by oracle Σ. We denote this augmented (deterministic) oracle with Σ^{aug} which now takes pk, trans, m and randomness s as input and returns σ. This randomness is also derived through the q-wise independent function, ensuring consistent answers for the same data $(pk, \text{msg1})$. We note that the length $\psi(n)$ of this randomness is only polynomial by construction of the generic forgery oracle:

Definition 8 (Magic Adversary). *The magic adversary $\mathcal{A} = \mathcal{A}_q$ (with parameter q) for input pk and access to the generic forgery oracle Σ^{aug} and communicating with an oracle $\langle \mathcal{S}(sk), \cdot \rangle^1$ works as described in the following steps:*

```
select a hash function $h$ from the family of $q$-wise independent functions $\mathcal{H}$
run an execution $\langle \mathcal{S}(sk), \mathcal{U}(pk, m'_0; r'_0)\rangle$ in the role of an honest user, where
    $(m'_0, m'_1, r'_0, s'_0) \leftarrow h(pk, msg1)$ is generated as the result of the
    $q$-wise independent function applied to the public key $pk$ and
    the first message $msg1$ of $\mathcal{S}$; let $\sigma'_0$ denote the resulting signature and
    $\mathsf{trans}'_0$ the corresponding transcript.
if $\mathsf{Vf}(pk, m'_0, \sigma'_0) = 1$ then let $\sigma'_1 \leftarrow \Sigma^{aug}(pk, \mathsf{trans}'_0, m'_1; s'_0)$ else set $\sigma'_0, \sigma'_1 \leftarrow \perp$
return $(m'_0, \sigma'_0, m'_1, \sigma'_1)$
```

It follows again from the completeness of BS together with the construction of the generic forgery oracle that the magic adversary succeeds in the unforgeability experiment with probability negligibly close to 1.

5.2 Impossibility Result

In the following we extend our result to restricted-cross resets.

Theorem 2. *Let BS be a three-move blind signature scheme, which is statistically blind and has statistical signature-derivation checks. Then there is no resetting (with restricted cross-resets) black-box reduction from unforgeability of the blind signature scheme BS to a hard non-interactive problem.*

The proof appears in the full version. The idea is similar to the vanilla case. Only here we use the q-wise independent hash function to ensure independent randomness for runs with the adversary, and we need to take care of the fact that the meta-reduction now loops to find the second message-signature pair. The latter can be done in (expected) polynomial time by the assumption about the restricted resets. Appropriate truncation then yields a meta-reduction running in fixed polynomial time.

Transcript-independence again guarantees that the redcution cannot distinguish answers from the magic adversary from the ones of the meta-reduction. Formally, one first reduces the case of at most q instances, each with at most q resets, to a single run by a standard hybrid argument. Then one injects the data from the transcript-independence experiment into this single run. The signature derivation check allows to verify (without the help of Σ) if one has successfully inserted the data in a "good" execution (and not in a run in which the magic adversary would have failed to produce a forgery).

6 Conclusion

We have shown that for the blind signature schemes of Chaum [11] and of Pointcheval-Stern [27] finding security reductions to any non-interactive cryptographic problem in the standard model is hard. This class of cryptographic problems is very broad in the sense that it contains candidates like RSA and collision-resistant hash functions, and also any combination thereof. This also allows us to make stronger infeasibility claims compared to previous results using meta-reductions in other areas.

Concerning optimality of our results we remark that:

- Our result can be transfered to the computational blindness case (under additional stipulations), thus also ruling out many approaches to revert to computationally blindness to circumvent the results for the statistical schemes.
- Enlarging the class of cryptographic problems to interactive ones is too demanding: unforgeability of any blind signature scheme can indeed be securely reduced to an interactive problem in the standard model by simply assuming that the scheme is unforgeable. It is, however, unclear if and how decisional problems can be subsumed under our class of non-interactive (computational) problems.
- Extending the result to protocols with more moves is impossible in light of Okamoto's scheme [26] with four moves in the standard model, based on a non-interactive assumption.

Hence, our result fits well into our current knowledge about constructing blind signatures and shows close boundaries for potential improvements on the efficiency or assumptions.

Acknowledgments

We thank the participants of the Dagstuhl Workshop on Cryptography, in particular Oded Goldreich for bringing up the connection to m-bounded concurrency and Salil Vadhan for pointing out the usage of q-wise independent functions instead of pseudorandom functions. We also thank the anonymous reviewers for valuable comments. Both authors are supported by the Emmy Noether Grant Fi 940/2-1 of the German Research Foundation (DFG), and by CASED (`www.cased.de`).

References

1. Masayuki Abe. *A Secure Three-Move Blind Signature Scheme for Polynomially Many Signatures.* Advances in Cryptology — Eurocrypt2001, Volume 2045 of Lecture Notes in Computer Science, pages 136–151. Springer-Verlag, 2001.
2. Masayuki Abe and Miyako Ohkubo. *A Framework for Universally Composable Non-Committing Blind Signatures.* Advances in Cryptology — Asiacrypt 2009, Volume 5912 of Lecture Notes in Computer Science, pages 435–450. Springer-Verlag, 2009.
3. Boaz Barak. *How to Go Beyond the Black-Box Simulation Barrier.* Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS) 2001, pages 106–115. IEEE Computer Society Press, 2001.

4. Emmanuel Bresson, Jean Monnerat, and Damien Vergnaud. *Separation Results on the "One-More" Computational Problems.* Topics in Cryptology — Cryptographer's Track, RSA Conference (CT-RSA) 2008, Volume 4964 of Lecture Notes in Computer Science, pages 71–87. Springer-Verlag, 2008.
5. Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. *The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. Journal of Cryptology*, 16(3):185–215, 2003.
6. Alexandra Boldyreva. *Efficient Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme.* Public-Key Cryptography (PKC)2003, Volume 2567 of Lecture Notes in Computer Science, pages 31–46. Springer-Verlag, 2003.
7. Daniel Brown. *What Hashes Make RSA-OAEP Secure?* Number 2006/223 in Cryptology eprint archive. `eprint.iacr.org`, 2006.
8. Daniel Brown. *Irreducibility to the One-More Evaluation Problems: More May Be Less.* Number 2007/435 in Cryptology eprint archive. `eprint.iacr.org`, 2007.
9. Dan Boneh and Ramarathnam Venkatesan. *Breaking RSA May Be Easier Than Factoring.* Advances in Cryptology — Eurocrypt'98, Lecture Notes in Computer Science, pages 59–71. Springer-Verlag, 1998.
10. Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. *Resettable Zero-Knowledge.* Proceedings of the Annual Symposium on the Theory of Computing (STOC) 2000, pages 235–244. ACM Press, 2000.
11. David Chaum. *Blind Signatures for Untraceable Payments.* Advances in Cryptology — Crypto'82, pages 199–203. Plemum, New York, 1983.
12. Jan Camenisch, Maciej Koprowski, and Bogdan Warinschi. *Efficient Blind Signatures Without Random Oracles.* Security in Communication Networks, Volume 3352 of Lecture Notes in Computer Science, pages 134–148. Springer-Verlag, 2004.
13. Jan Camenisch, Gregory Neven, and Abhi Shelat. *Simulatable Adaptive Oblivious Transfer.* Advances in Cryptology — Eurocrypt'07, Lecture Notes in Computer Science, pages 573–590. Springer-Verlag, 2007.
14. Jean-Sebastien Coron. *Optimal Security Proofs for PSS and Other Signature Schemes.* Advances in Cryptology — Eurocrypt2002, Volume 2332 of Lecture Notes in Computer Science, pages 272–287. Springer-Verlag, 2002.
15. Cynthia Dwork, Moni Naor, and Amit Sahai. *Concurrent zero-knowledge. J. ACM*, 51(6):851–898, 2004.
16. Marc Fischlin. *Round-Optimal Composable Blind Signatures in the Common Reference String Model.* Advances in Cryptology - Crypto 2006, Volume 4117 of Lecture Notes in Computer Science, pages 60–77. Springer-Verlag, 2006.
17. Marc Fischlin and Dominique Schröder. *Security of Blind Signatures under Aborts.* Public Key Cryptography, Volume 5443 of Lecture Notes in Computer Science, pages 297–316. Springer-Verlag, 2009.
18. Oded Goldreich and Hugo Krawczyk. *On the composition of Zero-Knowledge Proof Systems. SIAM Journal on Computing*, 25(1):169–192, 1996.
19. Omer Horvitz and Jonathan Katz. *Universally-Composable Two-Party Computation in Two Rounds.* Advances in Cryptology — Crypto 2007, Lecture Notes in Computer Science, pages 111–129. Springer-Verlag, 2007.
20. Carmit Hazay, Jonathan Katz, Chiu-Yuen Koo, and Yehuda Lindell. *Concurrently-Secure Blind Signatures Without Random Oracles or Setup Assumptions.* Theory of Cryptography Conference (TCC) 2007, Volume 4392 of Lecture Notes in Computer Science, pages 323–341. Springer-Verlag, 2007.

21. Russell Impagliazzo and Steven Rudich. *Limits on the Provable Consequences of One-Way Permutations.* Proceedings of the Annual Symposium on the Theory of Computing (STOC) 1989, pages 44–61. ACM Press, 1989.
22. Ari Juels, Michael Luby, and Rafail Ostrovsky. *Security of Blind Digital Signatures.* Advances in Cryptology — Crypto'97, Volume 1294 of Lecture Notes in Computer Science, pages 150–164. Springer-Verlag, 1997.
23. Aggelos Kiayias and Hong-Sheng Zhou. *Concurrent Blind Signatures Without Random Oracles.* SCN, Volume 4116 of Lecture Notes in Computer Science, pages 49–62. Springer-Verlag, 2006.
24. Aggelos Kiayias and Hong-Sheng Zhou. *Equivocal Blind Signatures and Adaptive UC-Security.* Theory of Cryptography Conference (TCC), Volume 4948 of Lecture Notes in Computer Science, pages 340–355. Springer-Verlag, 2008.
25. Yehuda Lindell. *Bounded-Concurrent Secure Two-Party Computation Without Setup Assumptions.* Proceedings of the Annual Symposium on the Theory of Computing (STOC)2003, pages 683–692. ACM Press, 2003.
26. Tatsuaki Okamoto. *Efficient Blind and Partially Blind Signatures Without Random Oracles.* Theory of Cryptography Conference (TCC)2006, Volume 3876 of Lecture Notes in Computer Science, pages 80–99. Springer-Verlag, 2006.
27. David Pointcheval and Jacques Stern. *Security Arguments for Digital Signatures and Blind Signatures. Journal of Cryptology*, 13(3):361–396, 2000.
28. Pascal Paillier and Jorge Luis Villar. *Trading One-Wayness Against Chosen-Ciphertext Security in Factoring-Based Encryption.* Advances in Cryptology — Asiacrypt'06, Lecture Notes in Computer Science. Springer-Verlag, 2006.
29. Markus Rückert. *Lattice-based Blind Signatures.* Cryptology ePrint Archive, Report 2008/322, 2008. http://eprint.iacr.org/.
30. Omer Reingold, Luca Trevisan, and Salil P. Vadhan. *Notions of Reducibility between Cryptographic Primitives.* TCC 2004, Lecture Notes in Computer Science, pages 1–20. Springer-Verlag, 2004.
31. Daniel Simon. *Finding Collisions on a One-Way Street: Can Secure Hash Functions Be Based on General Assumptions?* Advances in Cryptology — Eurocrypt'98, Volume 1403, pages 334–345. Springer-Verlag, 1998.