Impossibility of Blind Signatures From One-Way Permutations

Jonathan Katz[1], Dominique Schröder[2]*, and Arkady Yerukhimovich[1]

[1] University of Maryland, USA
{jkatz, arkady}@cs.umd.edu
[2] Darmstadt University of Technology, Germany
schroeder@me.com

Abstract. A seminal result in cryptography is that signature schemes can be constructed (in a black-box fashion) from any one-way function. The minimal assumptions needed to construct *blind* signature schemes, however, have remained unclear. Here, we rule out black-box constructions of blind signature schemes from one-way functions. In fact, we rule out constructions even from a random permutation oracle, and our results hold even for blind signature schemes for 1-bit messages that achieve security only against honest-but-curious behavior.

1 Introduction

Blind signature schemes, introduced by Chaum [10], allow a signer to interactively issue signatures for a user in such a way that, roughly, the signer learns nothing about the message being signed (*blindness*) while the user cannot compute any additional signatures without the help of the signer (*unforgeability*). Classical applications of blind signatures include e-cash, where a bank blindly signs coins withdrawn by users, and e-voting, where an authority blindly signs public keys that voters later use to cast their votes.

Several constructions of blind signature schemes are known in either the random oracle model [23, 1, 6, 7, 3] or the standard model [19, 8, 22, 12, 15, 16, 20, 13, 2]. The *minimal* assumptions needed to construct blind signatures, however, are unclear. On the positive side, there exist constructions of blind signatures based on (doubly) enhanced trapdoor permutations [19, 12, 16]. Interestingly, these constructions are all *nonblack-box* even in the honest-but-curious setting, relying as they do on either generic secure two-party computation or non-interactive zero-knowledge proofs. (More recently, protocols for secure two-party computation making only black-box use of enhanced trapdoor permutations have been shown [18]; these could be used in conjunction with [16] to give a black-box construction of blind signatures from certified enhanced trapdoor permutations.) On the other hand, for standard signatures we know that one-way functions suffice [21, 25], and there is no reason *a priori* to believe that blind signatures cannot be constructed from one-way functions also.

* This work was done while visiting the University of Maryland.

Previous work of Camenisch, Neven, and Shelat [9] (see also [13]) shows that any *unique* blind signature scheme implies oblivious transfer. Combined with known results showing that oblivious transfer cannot be constructed in a black-box fashion from one-way permutations, this at first may appear to rule out black-box constructions of blind signatures from one-way permutations. The uniqueness requirement, however, is quite strong: Fiore and Schröder show that unique signatures (even without the blindness requirement) cannot be constructed in a black-box fashion even from trapdoor permutations [11]. More importantly, uniqueness is not a standard desideratum for blind signatures and the result of Camenisch et al. implies nothing for blind signatures without the uniqueness property. In another line of work, Fischlin and Schröder [14] show that *three-round* blind signature schemes with signature-derivation checks cannot be constructed in a black-box way from any non-interactive problem. Their result, however, says nothing about protocols with more rounds, or for schemes that do not have signature-derivation checks. We refer to the reader to [26] for a comprehensive survey of the above results.

As our main result, we show:

Theorem 1 (Main theorem). *There is no black-box construction of blind signature schemes from one-way functions.*

Our result imposes no restrictions on the blind signature scheme, and applies even to schemes with imperfect completeness. Our result is actually more general than the above theorem indicates; it also applies to constructions based on one-way permutations or random oracles, and even rules out constructions of blind signature schemes for 1-bit messages that achieve security only against honest-but-curious behavior.

The proof of our impossibility result requires a careful combination of prior techniques in the area of black-box separations. At a high level, our basic framework is similar to the one used by Barak and Mahmoody-Ghidary in studying black-box constructions of (standard) signature schemes from one-way functions [4]. Our setting introduces several additional difficulties, however, not least of which is that we must deal with the case of *interactive* protocols. Also, Barak and Mahmoody-Ghidary prove limits on the efficiency of constructions, whereas we are interested in proving impossibility. To deal with these complications, we also rely on techniques used in analyzing constructions of key-agreement protocols from one-way functions [17, 5]. A more detailed overview of our proof is given in Section 2.

Black-box separations. In cryptography, constructions are usually proven secure by reduction to the security of some "low-level" primitive. Most known constructions are *black-box*, in that they treat the underlying primitive as an oracle and do not use any internal structure of the primitive; see [24] for extensive discussion and formal definitions. Impagliazzo and Rudich [17] initiated work showing impossibility of black-box constructions; in their paper they showed impossibility of constructing key-exchange protocols in a black-box manner from one-way functions. It is important to bear in mind that several nonblack-box constructions are known; nevertheless, black-box impossibility are useful insofar as they rule out a particular approach to a problem. Nonblack-box constructions also tend to be orders of magnitude less efficient than black-box constructions.

Organization. We provide on overview of our proof in Section 2. In Section 3 we present definitions of blind signatures, and we prove our main result in Section 4. In Section 5 we discuss extensions of our result to handle schemes with imperfect completeness, and to rule out constructions from one-way permutations.

2 Overview of Our Techniques

We consider interactive signature-issue protocols between a *signer* and a *user*. The input of the signer is a private key *sk*, and the user's input is a public key *pk* and a message m; at the end of this protocol the user outputs a signature σ on the message m. The algorithms run by both the signer and the user are given black-box access to a one-way function (OWF); we allow the parties to be computationally unbounded, but require that they only query the one-way function a polynomial number of times. For our impossibility result, we assume that both parties follow the protocol and are just honest-but-curious (i.e., semi-honest). This assumption only strengthens our result.

In the setting of blind signatures, security demands that:

Unforgeability The user should not be able to output two valid signatures after interacting with the signer once. (More generally, the user should be unable to output $k+1$ valid signatures on distinct messages after interacting with the signer k times.)

Blindness If the user executes the signature-issue protocol twice, once using a message m_0 and once using a message m_1, then the signer should be unable to tell in which order these executions were run. This should hold even if the signer is given both of the resulting signatures.

We show that if we wish to satisfy both conditions above then OWFs are not sufficient. To illustrate the main idea why this is true, consider the setting where both the user and signer are given access to a random oracle. Let Q denote the oracle queries made by the signer in generating its public and private keys. Now consider two protocol executions in which the user first obtains a signature on the message m_0 and then obtains a signature on the message m_1. Correctness intuitively requires that in each interaction the user learns sufficiently many of the queries in Q in order to be abe to derive a valid signature. Unforgeability requires that the user does not learn "too many" of the queries in Q in each interaction; in particular, the user should not learn enough queries in the first interaction to derive a valid signature on m_1. Finally, blindness implies that, from the point of view of the signer, the queries the user learns in the first interaction should be distributed identically to the queries the user learns in the second interaction. We show that all these requirements are in conflict.

More formally, we rely on results of [17, 5] showing that for any two-party protocol there is an algorithm Find that takes as input a transcript of an execution of the protocol and outputs, with high probability, a set that contains every oracle query that was asked by both parties ("intersection queries"). Noting that the signer can run this algorithm, the blindness requirement thus implies that the set obtained by running Find on the signature-issue protocol for m_0 must contain a set of intersection queries that are sufficient to derive a signature on the message m_1. (Else the signer knows that the first execution could not possibly have been for m_1.) We use this to construct a forger, which is a more efficient version of the one given in [4]. Our forger runs a single protocol execution honestly to obtain a signature on m_0, and then runs Find to learn all the intersection queries. By what we have just said, this set will contain enough information to allow the forger to also compute a valid signature on m_1.

From a technical point of view, our proof technique can be viewed as following the general framework proposed by Barak and Mahmoody-Ghidary [4], who show a forger for any (standard) signature scheme constructed from one-way functions in a black-box fashion. Our work differs from theirs in the following respects:

– The obvious difference is that we consider an *interactive* signing protocol, whereas in [4] the signing algorithm was non-interactive. Moreover, Barak and Mahmoody-Ghidary assume that signing is deterministic. This assumption is without loss of generality for standard signatures, but is more subtle in the case of blind signatures.
– While we use the same "usefulness" property as in [4], our proof that usefulness holds is very different from the analogous proof in their work: they assume a large message and argue that usefulness occurs for some pair of messages with high probability, whereas in our case we rely on blindness and show (roughly) that usefulness holds for any two messages with all but negligible probability. This allows us to simplify the attack and obtain a forger that makes only polynomially many oracle queries regardless of how many oracle queries the construction uses. (In the work of Barak and Mahmoody-Ghidary the number of queries made by the forger depends exponentially on the number of queries made by the construction.)

3 Definitions

3.1 Blind Signatures

The notation $A^{\mathcal{O}}(x)$ refers to an algorithm A that on input x gets black-box access to an oracle $\mathcal{O}$. 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 $\mathcal{Y}^{\langle \mathcal{X}(x), \cdot \rangle}(y)$ if $\mathcal{Y}$ can invoke a *single* execution of the protocol with $\mathcal{X}$. Accordingly, $\mathcal{X}^{\langle \cdot, \mathcal{Y}(y_0) \rangle, \langle \cdot, \mathcal{Y}(y_1) \rangle}(x)$ denotes that $\mathcal{X}$ can invoke *one execution each* with $\mathcal{Y}(y_0)$ and $\mathcal{Y}(y_1)$.

We define blind signatures for 1-bit messages; since we are proving impossibility, this only makes our results stronger.

Definition 1 (Oracle blind signature scheme). *An oracle blind signature scheme is a tuple of polynomial-time algorithms* $\mathsf{BS} = (\mathsf{Gen}^{(\cdot)}, \mathcal{S}^{(\cdot)}, \mathcal{U}^{(\cdot)}, \mathsf{Vrfy}^{(\cdot)})$*, where for any* $\lambda \in \mathbb{N}$ *and any oracle* $\mathcal{O} : \{0,1\}^\lambda \to \{0,1\}^\lambda$ *we have:*

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

We assume[1] perfect completeness*: i.e., for any* $\lambda \in \mathbb{N}$ *and* $\mathcal{O} : \{0,1\}^\lambda \to \{0,1\}^\lambda$*, any* $(sk, pk) \leftarrow \mathsf{Gen}^{\mathcal{O}}(1^\lambda)$*, any* $m \in \{0,1\}$*, and any signature* σ *output by* $\mathcal{U}^{\mathcal{O}}$ *in the joint execution of* $\mathcal{S}^{\mathcal{O}}(sk)$ *and* $\mathcal{U}^{\mathcal{O}}(pk, m)$*, it holds that* $\mathsf{Vrfy}^{\mathcal{O}}(pk, m, \sigma) = 1$*.*

3.2 Security of Blind Signatures

Blind signatures must satisfy two properties: unforgeability and blindness. For unforgeability we require that a user who runs a single execution of the signature-issuing protocol should be unable to forge a valid signature on two messages. For blindness we require that in two executions of the protocol, in which the user obtains signatures on both possible messages, the signer should be unable to determine which message was signed in which execution. In both cases, we assume semi-honest behavior. Our definitions of security are weaker than those usually considered; since we show impossibility, this only strengthens our results.

In the definitions that follow we consider an execution of an oracle blind signature scheme BS relative to a random oracle $\mathcal{O}$. Since a random oracle is one-way with overwhelming probability, any construction of blind signatures from one-way functions must give an oracle blind signature scheme satisfying these definitions. We remark that our definitions consider unbounded adversaries who make polynomially many queries to $\mathcal{O}$; however, we could have stated our definitions in terms of polynomial-time adversaries given access to an **NP** oracle.

Definition 2 (Unforgeability). *Oracle blind signature scheme* $\mathsf{BS} = (\mathsf{Gen}, \mathcal{S}, \mathcal{U}, \mathsf{Vrfy})$ *is* unforgeable *if for any semi-honest algorithm* $\mathcal{U}^*$ *that makes at most* $\mathsf{poly}(\lambda)$ *queries to* $\mathcal{O}$*, the probability that experiment* $\mathsf{Forge}^{\mathsf{BS}}_{\mathcal{U}^*}(\lambda)$ *evaluates to* 1 *is negligible (in* λ*), where*

Experiment $\mathsf{Forge}^{\mathsf{BS}}_{\mathcal{U}^*}(\lambda)$*:*

```
    Oracle $\mathcal{O} : \{0,1\}^\lambda \to \{0,1\}^\lambda$ is chosen at random
    $(sk, pk) \leftarrow \mathsf{Gen}^{\mathcal{O}}(1^\lambda)$
    $(\sigma_0, \sigma_1) \leftarrow \mathcal{U}^{*\langle \mathcal{S}(sk), \cdot \rangle, \mathcal{O}}(pk)$ (where $\mathcal{U}^*$ runs an honest execution
        of $\mathcal{U}(pk, 0)$ with $\mathcal{S}$ and then outputs signatures of its choice)
    Return 1 iff $\mathsf{Vrfy}^{\mathcal{O}}(pk, 0, \sigma_0) = 1$ and $\mathsf{Vrfy}^{\mathcal{O}}(pk, 1, \sigma_1) = 1$.
```

[1] We relax this requirement in Section 5.1.

Definition 3 (Blindness). *Oracle blind signature scheme* $\mathsf{BS} = (\mathsf{Gen}, \mathcal{S}, \mathcal{U}, \mathsf{Vrfy})$ *satisfies* blindness *if for any semi-honest algorithm* $\mathcal{S}^*$ *that makes at most* $\mathsf{poly}(\lambda)$ *queries to* $\mathcal{O}$*, the probability that experiment* $\mathsf{Unblind}_{\mathcal{S}^*}^{\mathsf{BS}}(\lambda)$ *evaluates to* 1 *is negligibly close to* $1/2$*, where*

```
Experiment $\mathsf{Unblind}_{\mathcal{S}^*}^{\mathsf{BS}}(\lambda)$
    Oracle $\mathcal{O} : \{0,1\}^\lambda \to \{0,1\}^\lambda$ is chosen at random
    $r \leftarrow \{0,1\}^\lambda$; $b \leftarrow \{0,1\}$
    $(sk, pk) \leftarrow \mathsf{Gen}^{\mathcal{O}}(1^\lambda; r)$
    $\mathsf{st} \leftarrow \mathcal{S}^{*\langle \cdot, \mathcal{U}(pk,b)\rangle, \langle \cdot, \mathcal{U}(pk,\bar{b})\rangle, \mathcal{O}}(sk, pk, r)$ (where $\mathcal{S}^*$ runs
        an honest execution of the protocol with each instance of $\mathcal{U}$)
    Let $\sigma_b, \sigma_{1-b}$ denote the local outputs of each instance of $\mathcal{U}$
    $b' \leftarrow \mathcal{S}^{*\mathcal{O}}(\mathsf{st}, \sigma_0, \sigma_1)$
    Return 1 iff $b = b'$.
```

Throughout this work we make the simplifying assumption that the signing algorithm $\mathcal{S}$ is deterministic. This is without loss of generality when we consider the above definitions of security, as a blind signature scheme with randomized signer $\mathcal{S}$ can always be converted to a scheme with deterministic signer $\mathcal{S}'$ by (1) including a key for a pairwise-independent hash function as part of the signer's private key; (2) having the user send a random nonce as its first message in the signing protocol; and then (3) having $\mathcal{S}'$ apply the hash function to the user's first message to generate random coins that it then uses to run $\mathcal{S}$.

4 Attacking Black-Box Constructions of Blind Signatures

In this section we show that there is no black-box construction of blind signatures from one-way functions. To this end, we show that any oracle blind signature scheme $\mathsf{BS}^{(\cdot)}$ fails to satisfy either blindness or unforgeability when instantiated with a random oracle $\mathcal{O} : \{0,1\}^\lambda \to \{0,1\}^\lambda$.

4.1 Preliminaries

We begin by reviewing a lemma from previous work [17, 5] that we utilize in our proof. Informally, it states that for any two-party protocol Π where each party has access to a random oracle there exists an algorithm that, upon observing the transcript of an interaction, finds with high probability all the intersection queries (queries to the oracle that have been asked by both parties).

Lemma 1 ([5]). *Let* Π *be a two-party (randomized) protocol where each party asks at most* q *queries to an oracle. Then for every* $\delta \in (0,1)$*, there is an algorithm* Find_δ *that makes* $\mathcal{O}((q/\delta)^2)$ *oracle queries, such that when* Find_δ *is given the transcript of an execution of the protocol between the parties in the presence of a random oracle, the queries made by* Find_δ *contain all the intersection queries of the two parties with probability at least* $1 - \delta$*. (The probability is taken over the coins of* Find_δ *and the parties, as well as choice of the random oracle.)*

We apply this in our setting in the following way. Corresponding to any oracle blind signature scheme $\mathsf{BS}^{(\cdot)}$, define the following two-party protocol Π between a signer $\mathcal{S}$ and a user $\mathcal{U}$:

1. $\mathcal{S}$ runs $(sk, pk) \leftarrow \mathsf{Gen}^{\mathcal{O}}(1^\lambda)$ and sends pk to $\mathcal{U}$.
2. $\mathcal{U}$ and $\mathcal{S}$ then run the signature-issuing protocol on the message 1, at the end of which $\mathcal{U}$ obtains a signature σ_1.
3. $\mathcal{U}$ runs $\mathsf{Vrfy}^{\mathcal{O}}(pk, 1, \sigma_1)$.

For the remainder of Section 4, fix some δ and define Find_δ (as per Lemma 1) relative to the above protocol Π. Say the above protocol is run in the presence of a random oracle $\mathcal{O}$. If we let $Q(\mathcal{S}_\Pi)$ and $Q(\mathcal{U}_\Pi)$ denote the $\mathcal{O}$-queries made by each party during an execution of the above protocol that resulted in transcript trans, then Lemma 1 guarantees that, with high probability,

$$Q(\mathcal{S}_\Pi) \cap Q(\mathcal{U}_\Pi) \subseteq \mathsf{Find}_\delta^{\mathcal{O}}(\mathsf{trans}).$$

4.2 From Blindness to Usefulness

In this section we study the question of what blindness implies with regard to the set of queries $\mathcal{I}$ output by the Find algorithm. The main observation is that due to blindness the set $\mathcal{I}$ (that contains all intersection queries with high probability) must be somehow "independent" of the actual message being signed. Recall that in the blindness game the semi-honest signer interacts with two honest user instances in a random order. The task for the attacker is to guess which instance used which message. Now, consider two protocol executions and suppose that the set of intersection queries depended on the message being used. Then just by looking at those queries it would be possible to determine the order of the messages.

To formalize this intuition, we first define some notation. Consider an execution of the blindness experiment. We write $Q(\mathsf{Gen})$ to represent the set of $\mathcal{O}$-queries made during key generation. In the interaction between $\mathcal{S}$ and $\mathcal{U}(pk, 0)$, let $Q(\mathcal{S}_0)$ denote the $\mathcal{O}$-queries made by $\mathcal{S}$; let trans_0 denote the resulting transcript; let σ_0 be the signature that $\mathcal{U}$ outputs; and let $Q(\mathsf{Vrfy}_0)$ be the set of $\mathcal{O}$-queries made by the verification algorithm $\mathsf{Vrfy}^{\mathcal{O}}(pk, 0, \sigma_0)$. Define $Q(\mathcal{S}_1)$, trans_1, and $Q(\mathsf{Vrfy}_1)$ analogously for the interaction between $\mathcal{S}$ and $\mathcal{U}(pk, 1)$. (Note that by perfect completeness and the assumption of semi-honest behavior by $\mathcal{S}$, both user instances always obtain a valid signature on their message.)

Consider a (semi-honest) signer $\mathcal{S}^*$ in the blindness game. Say the adversary runs Find using trans_1. It follows from Lemma 1 and the definition of Π in the previous section that, with high probability,

$$Q(\mathsf{Vrfy}_1) \cap (Q(\mathsf{Gen}) \cup Q(\mathcal{S}_1)) \subseteq \mathsf{Find}(\mathsf{trans}_1). \tag{1}$$

$\mathcal{S}^*$ can check whether Equation (1) holds by computing $\mathsf{Vrfy}^{\mathcal{O}}(pk, 1, \sigma_1)$ itself. But then blindness implies that Equation (1) must hold with high probability even when Find is run on the "wrong" interaction; i.e.,

$$Q(\mathsf{Vrfy}_1) \cap (Q(\mathsf{Gen}) \cup Q(\mathcal{S}_0)) \subseteq \mathsf{Find}(\mathsf{trans}_0).$$

In the language of [4], this means that the message '0' is "useful" for the message '1' with high probability.

We now give the formal proof.

Lemma 2. *Let* BS *be an oracle blind signature scheme satisfying blindness. Consider an execution of the blindness experiment (cf. Definition 3), and let* $Q(\mathsf{Gen})$*,* $Q(\mathcal{S}_b)$*,* trans_b*, and* $Q(\mathsf{Vrfy}_b)$ *be as defined above. Then with probability at least* $1 - \delta - \mathsf{negl}(\lambda)$ *over the random coins of the experiment it holds that*

$$Q(\mathsf{Vrfy}_1) \cap (Q(\mathsf{Gen}) \cup Q(\mathcal{S}_0)) \subseteq \mathsf{Find}_\delta(\mathsf{trans}_0).$$

Proof. We first observe that with probability at least $1 - \delta$ we have

$$Q(\mathsf{Vrfy}_1) \cap (Q(\mathsf{Gen}) \cup Q(\mathcal{S}_1)) \subseteq \mathsf{Find}_\delta(\mathsf{trans}_1).$$

This follows immediately from Lemma 1 and our definition of protocol Π in the previous section.

Consider now the following adversary $\mathcal{S}^*$:

1. $\mathcal{S}^*$ runs the honest key-generation algorithm to obtain (sk, pk). It records the $\mathcal{O}$-queries $Q(\mathsf{Gen})$ made during this step.
2. $\mathcal{S}^*$ then runs the honest signing protocol with the first user instance. Let trans denote the transcript of this execution, and let $Q(\mathcal{S})$ denote the $\mathcal{O}$-queries made during this step.
3. $\mathcal{S}^*$ then runs the honest signing protocol with the second user instance.
4. $\mathcal{S}^*$ is given signatures σ_0, σ_1 on the messages 0 and 1, respectively. (By perfect completeness, both user instances always obtain valid signatures.) $\mathcal{S}^*$ verifies σ_1 and records the $\mathcal{O}$-queries $Q(\mathsf{Vrfy}_1)$ made in doing so.
5. Finally, $\mathcal{S}^*$ outputs 1 iff $Q(\mathsf{Vrfy}_1) \cap (Q(\mathsf{Gen}) \cup Q(\mathcal{S})) \subseteq \mathsf{Find}_\delta(\mathsf{trans})$.

If $b = 1$, and so the first user instance represents an interaction with $\mathcal{U}(pk, 1)$, then $\mathsf{trans} = \mathsf{trans}_1$ and $Q(\mathcal{S}) = Q(\mathcal{S}_1)$ and so $\mathcal{S}^*$ outputs 1 with probability at least $1-\delta$. The blindness property thus implies that $\mathcal{S}^*$ outputs 1 with probability at least $1 - \delta - \mathsf{negl}(\lambda)$ when $b = 0$ (and the first user instance represents an interaction with $\mathcal{U}(pk, 0)$). This concludes the proof.

4.3 Forging a Signature

Before presenting our forger, we begin by discussing the ideas behind our attack. The main observation is that due to the blindness of the signature scheme the intersection queries between the signer and user are somehow "independent" of the message. This was formalized in Lemma 2, where we showed that (with high probability)

$$Q(\mathsf{Vrfy}_1) \cap (Q(\mathsf{Gen}) \cup Q(\mathcal{S}_0)) \subseteq \mathsf{Find}(\mathsf{trans}_0).$$

Intuitively, this means that all the "important" queries needed to verify a signature on the message '1' must already be contained in the set of queries that are found when signing and verifying the message '0'. Thus, in the language of

Barak and Mahmoody-Ghidary [4], we have shown that 0 is "useful" for 1 with high probability. As in that paper, we use this property to show an attack.

The above condition seems to suggest that the set of intersection queries for '0' is sufficient to generate a signature on '1'. However, this is not quite true. The problem is that there may be queries that the user makes with high probability when generating and verifying a signature for 1 that are not in the set $\mathsf{Find}(\mathsf{trans}_0)$; this could cause technical problems because our forger must get the answers to these queries right when constructing a forged signature. For a concrete example, consider a blind signature scheme where the user, on input a message b, always queries $y = \mathcal{O}(b)$ and includes y as part of the signature; verification checks whether $\mathcal{O}(b) = y$ (among other things). In such a case the query $\mathcal{O}(1)$ may not be in the set $\mathsf{Find}(\mathsf{trans}_0)$.

As in [4], we handle this issue by introducing a phase in which the forger makes any "heavy" queries that are made by the user with high probability. If the forger knows the correct answers to all these high-probability queries then it is very unlikely that it will incorrectly answer some query asked during the verification of the forged signature.

Given this intuition we now present the details of the attack. The main structure of the attack is based on [4] with necessary changes to adapt the proof to our setting. In particular, our attack makes only polynomially many oracle queries (regardless of the number of queries the scheme itself makes).

Theorem 2. *Let* BS *be an oracle blind signature scheme satisfying blindness. Then there exists an adversary* $\mathcal{U}^*$ *for which* $\mathsf{Forge}_{\mathcal{U}^*}^{\mathsf{BS}^{\mathcal{O}}}(\lambda)$ *(cf. Definition 2) is not negligible.*

Proof. Consider the following adversary $\mathcal{U}^*$:

Setup. The input of $\mathcal{U}^*$ is a public key pk.

Step 1: Requesting a signature. $\mathcal{U}^*$ runs the honest signing protocol (using message '0') with the signer, eventually obtaining a valid signature σ_0. Let trans_0 be the transcript (i.e., the messages exchanged) for this execution. $\mathcal{U}^*$ verifies the received signature and records the oracle queries $Q(\mathsf{Vrfy}_0)$ made. $\mathcal{U}^*$ then computes $\mathsf{Find}_\delta(\mathsf{trans}_0)$ with $\delta = 1/10$.

Denote by T_0 the complete transcript of the entire experiment run so far; i.e., T_0 contains the entire views of both the signer and $\mathcal{U}^*$. Note that $\mathcal{U}^*$ has only partial knowledge about T_0.

Step 2: Learning query/answer pairs. Let L_0 be the information that $\mathcal{U}^*$ has about T_0 and the oracle $\mathcal{O}$ following Step 1. Let q be an upper bound on the total number of queries asked when running each of the algorithms in BS once. Let $\epsilon = \delta/q$ and $M = q/\epsilon\delta = 100q^2$. For $i = 1, \ldots, M$ do the following:

1. Let $\mathbf{D}_{i-1}$ be the distribution of T_0, the transcript of the first step, conditioned on L_{i-1}.
2. Denote by $Q(L_{i-1})$ the oracle queries that appear in L_{i-1}. If a query $x \in \{0,1\}^\lambda \backslash Q(L_{i-1})$ appears with probability at least ϵ in $\mathbf{D}_{i-1}$, then

$\mathcal{U}^*$ queries $\mathcal{O}(x)$ and adds the query/answer pair to L_{i-1} to obtain L_i. (If there is more than one such x, then $\mathcal{U}^*$ adds the lexicographically first one.)

Step 3: Sampling a possible transcript. $\mathcal{U}^*$ samples a random transcript $\widetilde{T}_0$ according to the distribution $\mathbf{D}_M$. Observe that $\widetilde{T}_0$ also defines a secret key $\widetilde{sk}$ that may be distinct from the real secret key sk. Moreover, $\widetilde{T}_0$ may include some new mappings that were not defined in L_M. We let $\widetilde{\mathcal{O}}$ be the following oracle: If a query x appears in $\widetilde{T}_0$ then $\widetilde{\mathcal{O}}(x)$ returns the value contained in $\widetilde{T}_0$; otherwise, $\widetilde{\mathcal{O}}(x) = \mathcal{O}(x)$.

Step 4: Forging. $\mathcal{U}^*$ runs the interactive signing protocol for the message '1' locally, playing the role of both the signer and the user, using $\widetilde{sk}$ and $\widetilde{\mathcal{O}}$; that is, it computes $\sigma_1 \leftarrow \left\langle \mathcal{S}^{\tilde{\mathcal{O}}}(\widetilde{sk}), \mathcal{U}^{\tilde{\mathcal{O}}}(pk, 1) \right\rangle$. For technical reasons, we also have $\mathcal{U}^*$ verify σ_1 (using $\mathcal{O}$). Finally, $\mathcal{U}^*$ outputs the two signatures σ_0, σ_1.

Analysis. It is easy to see that $\mathcal{U}^*$ makes polynomially many queries to $\mathcal{O}$. Since $\mathcal{U}^*$ runs the honest user protocol in its execution with the signer (in step 1), σ_0 is always a valid signature on '0'. In the remainder of the proof, we show that σ_1 is a valid signature on the message '1' with probability at least $4/5 - \delta - \mathsf{negl}(\lambda)$.

In the following we show that, with high probability, verification of σ_1 never asks a query on which oracles $\widetilde{\mathcal{O}}$ and $\mathcal{O}$ disagree. Assuming this to be the case, it follows (by the perfect completeness of the signature scheme) that σ_1 is a valid signature on '1' with respect to the true oracle $\mathcal{O}$.

Lemma 3. *Let $Q(\mathsf{Vrfy}_1)$ denote the set of oracle queries made when $\mathcal{U}^*$ verifies the signature σ_1. Let $\widetilde{Q}(\mathsf{Gen})$ and $\widetilde{Q}(\mathcal{S}_0)$ denote the set of oracle queries made by the key-generation and signing algorithms, respectively, in the sampled transcript $\widetilde{T}_0$. Then with probability at least $\frac{4}{5} - \delta - \mathsf{negl}(\lambda)$ it holds that*

$$Q(\mathsf{Vrfy}_1) \cap \left(\widetilde{Q}(\mathsf{Gen}) \cup \widetilde{Q}(\mathcal{S}_0)\right) \subseteq \mathsf{Find}_\delta(\mathsf{trans}_0).$$

Lemma 3 implies Theorem 2. To see this, note that $\mathsf{Vrfy}^{\tilde{\mathcal{O}}}(pk, 1, \sigma_1) = 1$ by perfect completeness of the signature scheme. But the only queries on which $\tilde{\mathcal{O}}$ and $\mathcal{O}$ can possibly differ are queries in $\left(\widetilde{Q}(\mathsf{Gen}) \cup \widetilde{Q}(\mathcal{S}_0)\right) \setminus \mathsf{Find}_\delta(\mathsf{trans}_0)$. If verification makes no such queries, then

$$\mathsf{Vrfy}^{\mathcal{O}}(pk, 1, \sigma_1) = \mathsf{Vrfy}^{\tilde{\mathcal{O}}}(pk, 1, \sigma_1) = 1.$$

Let E denote the event considered in Lemma 3. The proof of Lemma 3 follows the proof in [4]: we define a sequence of hybrid distributions, and analyze the probability of E in each of them. The biggest difference between the proof in [4] and the proof here is when we analyze the probability that E happens in the final hybrid distribution.

Definition of hybrid distributions. We define four hybrid distributions $\mathbf{H}^0$, $\mathbf{H}^1$, $\mathbf{H}^2$, and $\mathbf{H}^3$ as follows:

$\mathbf{H}^0$. The first hybrid is the distribution $(\widetilde{T}_0, T_1)$, where $\widetilde{T}_0$ is the transcript sampled by $\mathcal{U}^*$ in Step 3, and T_1 is the transcript of Step 4 (i.e., generation and verification of σ_1). Note that $\widetilde{T}_0$ includes the queries of the key-generation algorithm.

$\mathbf{H}^1$. The second hybrid is defined identically to $\mathbf{H}^0$, except that we use $\widetilde{\mathcal{O}}$ to verify σ_1. (In $\mathbf{H}^0$, oracle $\mathcal{O}$ was used when verifying σ_1.)

$\mathbf{H}^2$. The third hybrid has the same distribution as $\mathbf{H}^1$, except that we change the definition of $\widetilde{\mathcal{O}}$ as follows. Recall that L_M is the set of $\mathcal{O}$ query/answer pairs that $\mathcal{U}^*$ knows after the learning queries step (Step 2). We define $\widetilde{\mathcal{O}}$ to answer any query contained in L_M with the answer stored there and all other queries with a randomly chosen value. This modification results in an oracle $\widetilde{\mathcal{O}}$ that agrees with $\mathcal{O}$ on all the queries $\mathcal{U}^*$ has queried to $\mathcal{O}$ until the end of Step 2; all other queries are answered completely at random.

$\mathbf{H}^3$. The distribution of the last hybrid is the same as $\mathbf{H}^2$ except that $\widetilde{T}_0$ is replaced with T_0. Thus the output of this hybrid is (T_0, T_1) which describes the experiment where we first compute $(sk, pk) \leftarrow \mathsf{Gen}$; then run $\sigma_0 \leftarrow \langle \mathcal{S}^{\mathcal{O}}(sk), \mathcal{U}^{\mathcal{O}}(pk, 0) \rangle$ and $\sigma_1 \leftarrow \langle \mathcal{S}^{\mathcal{O}}(sk), \mathcal{U}^{\mathcal{O}}(pk, 1) \rangle$; and finally verify both signatures. Note that all algorithms here use the "real" oracle $\mathcal{O}$ and thus verification succeeds for both signatures.

The distributions considered in each hybrid are taken over random choice of the oracle and random coins of the key-generation algorithm, the signer, and the adversary. We prove Lemma 3 by showing that (1) event E occurs with high probability in $\mathbf{H}^3$ and (2) the probability that event E occurs in $\mathbf{H}^0$ is not much smaller than its probability in $\mathbf{H}^3$.

We first show that E occurs with high probability in $\mathbf{H}^3$. The following is an immediate consequence of Lemma 2.

Claim. $\Pr_{\mathbf{H}^3}[E] \geq 1 - \delta - \mathsf{negl}(\lambda)$.

We next show that the probability of E remains unchanged when we move from $\mathbf{H}^3$ to $\mathbf{H}^2$.

Claim. $\mathbf{H}^2 \equiv \mathbf{H}^3$. Thus, $\Pr_{\mathbf{H}^2}[E] = \Pr_{\mathbf{H}^3}[E]$.

Proof. The proof here is the same as in [4]. We can view $\mathbf{H}^3$ as being sampled as follows: first, fix L_M; then choose the transcript T_0 at random from $\mathbf{D}_M$. This, however, is exactly the same distribution as $\mathbf{H}^2$ where L_M is fixed and we then choose $\widetilde{T}_0$ from $\mathbf{D}_M$.

For the next claim, we need the following definition.

Definition 4 (Statistical distance). *If X, Y are two random variables taking values in a finite set A, then* $\mathsf{SD}(X, Y) = 1/2 \cdot \sum_{a \in A} |\Pr[X = a] - \Pr[Y = a]|$.

We now show that $\mathbf{H}^1$ and $\mathbf{H}^2$ are "close".

Claim. $\mathsf{SD}(\mathbf{H}^1, \mathbf{H}^2) \leq \frac{1}{5}$. Thus, $\Pr_{\mathbf{H}^1}[E] \geq \Pr_{\mathbf{H}^2}[E] - \frac{1}{5}$.

Proof. Let $Q(T_0)$ be the queries contained in the transcript T_0. Let B be the event that $\mathcal{U}^*$ ever asks a query in $Q(T_0) \setminus Q(L_M)$. It is clear that $\mathbf{H}^1 = \mathbf{H}^2$ as long as event B does not occur in either of them, since in both distributions any queries outside of $Q(T_0)$ are answered randomly. This implies that $\Pr_{\mathbf{H}^1}[B] = \Pr_{\mathbf{H}^2}[B]$, and $\mathsf{SD}(\mathbf{H}^1, \mathbf{H}^2) \leq \Pr_{\mathbf{H}^2}[B]$. We now show that $\Pr_{\mathbf{H}^2}[B] \leq \frac{1}{5}$. (In the following, all probabilities are in $\mathbf{H}^2$.)

Recall that in Step 2 of the attack, we set $\epsilon = \delta/q$ and $\mathcal{U}^*$ learns at most $M = 100q^2$ query/answer pairs from $\mathcal{O}$. Let $\mathbf{D}_i$ be the distribution of T_0 sampled in this step by $\mathcal{U}^*$ given the set L_i of known query/answer pairs. Let C be the event that there are more than M queries that become likely during the attack. That is, C is the event that there exists a query $x \notin Q(L_M)$ such that x is asked in $\mathbf{D}_M$ with probability at least ϵ. Below, we show that $\Pr[C] \leq \delta = \frac{1}{10}$ and $\Pr[B \mid \neg C] \leq \delta = \frac{1}{10}$. This completes the proof, since then

$$\begin{aligned}\Pr[B] &= \Pr[C] \cdot \Pr[B \mid C] + \Pr[\neg C] \cdot \Pr[B \mid \neg C] \\ &\leq \Pr[C] + \Pr[B \mid \neg C] \leq 2\delta = \frac{1}{5}\,.\end{aligned}$$

The following two claims complete the proof that $\mathbf{H}^1$ and $\mathbf{H}^2$ are close.

Claim. Let C be the event defined in the proof of the previous claim. Then $\Pr_{\mathbf{H}^2}[C] \leq \delta$.

Proof. All probabilities here are in $\mathbf{H}^2$. Consider an arbitrary query x and let hit_x be the event that x is queried to $\mathcal{O}$ by the signer and then by the user when generating the signature on '0'. Let $q_x = \Pr[\mathsf{hit}_x]$. Finally, let $A_x(i)$ be the event that x is asked in the ith iteration of Step 2; let $p_x(i) = \Pr[A_x(i)]$; and let $p_x = \Pr[\cup_i A_x(i)]$. Note that $\sum_x q_x \leq q$ since q is an upper bound on the total number of queries asked when running each algorithm of the blind signature scheme. Furthermore, $q_x \geq \epsilon p_x$ because

$$q_x = \Pr[\mathsf{hit}_x] \geq \sum_i \Pr[\mathsf{hit}_x \mid A_x(i)] \cdot \Pr[A_x(i)]\,,$$

and $\mathcal{U}^*$ adds a query to its list only if the probability that this query is asked is at least ϵ. Thus, $\Pr[\mathsf{hit}_x \mid A_x(i)] \geq \epsilon$ and so $q_x \geq \epsilon \sum_i \Pr[A_x(i)] = \epsilon p_x$.

Assume for the sake of contradiction that $\Pr[C] > \delta$. Since C is the event that M queries are learned in Step 2, this implies that the expected number of queries asked, $\sum_x p_x$, is larger than δM. But this would imply

$$\delta M < \sum_x p_x \leq \sum_x q_x/\epsilon \leq q/\epsilon,$$

contradicting the fact that $M = q/\delta\epsilon$.

Claim. Let B and C be as defined earlier. Then $\Pr_{\mathbf{H}^2}[B \mid \neg C] \leq \delta$.

Proof. Recall that in Step 4 $\mathcal{U}^*$ relies only on the mappings stored in L_M, and all queries from $Q(T_0)\backslash Q(L_M)$ are answered at random. But then $\mathbf{H}^2$ is independent of T_0 conditioned on L_M (whereas L_M has the distribution $\mathbf{D}_M$). This means that we can imagine defining $\mathbf{H}^2$ by choosing L_M first, then running $\mathcal{U}^*$ (using L_M) to sample $\mathbf{H}^2$, and then choosing T_0 conditioned on L_M and $\mathbf{H}^2$. Recall that event C is determined by L_M, and assume that L_M is such that event $\neg C$ occurs. This implies that every query asked by $\mathcal{U}^*$ that is not in $Q(L_M)$ must appear in $\mathbf{D}_M$ with probability less than ϵ. Since $\mathcal{U}^*$ asks at most q queries in Step 4, the probability that $Q(T_0)\backslash Q(L_M)$ contains one of these queries is at most $\epsilon q = \delta$.

Finally, we show that E occurs with the same probability in $\mathbf{H}^0$ and $\mathbf{H}^1$.

Claim. $\Pr_{\mathbf{H}^0}[E] = \Pr_{\mathbf{H}^1}[E]$.

Proof. This claim follows easily if both hybrid distributions $\mathbf{H}^0$ and $\mathbf{H}^1$ use the same oracle $\mathcal{O}$ and if they are sampled using the same random coins for key generation and the adversary (note that the randomness of the adversary fully determines the randomness used to run the honest user algorithm during the signature-issue protocol). But then it follows that event E occurs in $\mathbf{H}^0$ if and only if it also occurs in $\mathbf{H}^1$.

This completes the proof of Lemma 3, and thus the proof of Theorem 2.

5 Extensions

In this section we briefly discuss how to extend our impossibility result to the case of blind signature schemes with imperfect completeness, and to constructions from one-way permutations.

5.1 Imperfect Completeness

Let $\mathsf{BS}^{(\cdot)}$ be an oracle blind signature scheme for which correctness holds with all but negligible probability; i.e., for any $\mathcal{O}$ and any $m \in \{0,1\}$, we have

$$\Pr\left[\begin{array}{c}(sk, pk) \leftarrow \mathsf{Gen}^{\mathcal{O}}(1^\lambda);\\ (\perp, \sigma) \leftarrow \left\langle \mathcal{S}^{\mathcal{O}}(sk), \mathcal{U}^{\mathcal{O}}(pk, m)\right\rangle\end{array} : \mathsf{Vrfy}^{\mathcal{O}}(pk, m, \sigma) = 1\right] \geq 1 - \mathsf{negl}(\lambda).$$

Our results from the previous section can be easily extended to such schemes. The proof of Lemma 2 is largely identical, with the only modification being to explicitly consider what happens if either of the signatures computed by the two user instances are invalid.

The forgery attack also proceeds just as in the previous section. Since the probability that one of the signatures is invalid is negligible, this only affects the forgery probability by a negligible amount.

5.2 One-Way Permutations

We now discuss how to extend our impossibility result to also rule out constructions from one-way permutations. As noted in [5], the Find algorithm can be modified to work in the random permutation model with a polynomial blow-up in the number of queries. It follows that an analogue of Lemma 2 holds when $\mathcal{O}$ is chosen as a random permutation. (Again, a random permutation oracle is one-way with all but negligible probability.) For the forgery attack we modify the proof of Theorem 2 as in [4]. We omit the details here.

Acknowledgments

The authors thank Marc Fischlin, Dov Gordon, and Mohammad Mahmoody-Ghidary for helpful discussions. We also thank the anonymous reviewers for valuable comments. Dominique Schröder is supported by the Emmy Noether Program Fi 940/2-1 of the German Research Foundation (DFG) and by the German Academic Exchange Service (DAAD).

References

1. M. Abe. A secure three-move blind signature scheme for polynomially many signatures. In *Advances in Cryptology — Eurocrypt 2001*, volume 2045 of *LNCS*, pages 136–151. Springer, 2001.
2. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *Advances in Cryptology — Crypto 2010*, volume 6223 of *LNCS*, pages 209–236. Springer, 2010.
3. M. Abe and M. Ohkubo. A framework for universally composable non-committing blind signatures. In *Advances in Cryptology — Asiacrypt 2009*, volume 5912 of *LNCS*, pages 435–450. Springer, 2009.
4. B. Barak and M. Mahmoody-Ghidary. Lower bounds on signatures from symmetric primitives. In *48th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 680–688. IEEE, 2007.
5. B. Barak and M. Mahmoody-Ghidary. Merkle puzzles are optimal — an $O(n^2)$-query attack on key exchange from a random oracle. In *Advances in Cryptology — Crypto 2009*, volume 5677 of *LNCS*, pages 374–390. Springer, 2009.
6. M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, 2003.
7. A. Boldyreva. Threshold signatures, multisignatures, and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In *6th Intl. Workshop on Theory and Practice in Public Key Cryptography — PKC 2003*, volume 2567 of *LNCS*, pages 31–46. Springer, 2003.
8. J. Camenisch, M. Koprowski, and B. Warinschi. Efficient blind signatures without random oracles. In *4th Intl. Conf. on Security in Communication Networks — SCN 2004*, volume 3352 of *LNCS*, pages 134–148. Springer, 2004.
9. J. Camenisch, G. Neven, and A. Shelat. Simulatable adaptive oblivious transfer. In *Advances in Cryptology — Eurocrypt 2007*, volume 4515 of *LNCS*, pages 573–590. Springer, 2007.

10. D. Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology — Crypto '82*, pages 199–203. Plenum Press, 1983.
11. D. Fiore and D. Schröder. Uniqueness is a different story: Impossibility of verifiable random functions from trapdoor permutations, 2010. Cryptology ePrint Archive, Report 2010/648. `http://eprint.iacr.org/2010/648`
12. M. Fischlin. Round-optimal composable blind signatures in the common reference string model. In *Advances in Cryptology — Crypto 2006*, volume 4117 of *LNCS*, pages 60–77. Springer, 2006.
13. M. Fischlin and D. Schröder. Security of blind signatures under aborts. In *12th Intl. Conference on Theory and Practice of Public Key Cryptography — PKC 2009*, volume 5443 of *LNCS*, pages 297–316. Springer, 2009.
14. M. Fischlin and D. Schröder. On the impossibility of three-move blind signature schemes. In *Advances in Cryptology — Eurocrypt 2010*, volume 6110 of *LNCS*, pages 197–215. Springer, 2010.
15. C. Hazay, J. Katz, C.-Y. Koo, and Y. Lindell. Concurrently-secure blind signatures without random oracles or setup assumptions. In *4th Theory of Cryptography Conference — TCC 2007*, volume 4392 of *LNCS*, pages 323–341. Springer, 2007.
16. O. Horvitz and J. Katz. Universally-composable two-party computation in two rounds. In *Advances in Cryptology — Crypto 2007*, volume 4622 of *LNCS*, pages 111–129. Springer, 2007.
17. R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 44–61. ACM Press, 1989.
18. Y. Ishai, E. Kushilevitz, Y. Lindell, and E. Petrank. Black-box constructions for secure computation. In *38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 99–108. ACM Press, 2006.
19. A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures. In *Advances in Cryptology — Crypto '97*, volume 1294 of *LNCS*, pages 150–164. Springer, 1997.
20. A. Kiayias and H.-S. Zhou. Equivocal blind signatures and adaptive UC-security. In *5th Theory of Cryptography Conference — TCC 2008*, volume 4948 of *LNCS*, pages 340–355. Springer, 2008.
21. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 33–43. ACM Press, 1989.
22. T. Okamoto. Efficient blind and partially blind signatures without random oracles. In *3rd Theory of Cryptography Conference — TCC 2006*, volume 3876 of *LNCS*, pages 80–99. Springer, 2006.
23. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
24. O. Reingold, L. Trevisan, and S. P. Vadhan. Notions of reducibility between cryptographic primitives. In *1st Theory of Cryptography Conference — TCC 2004*, volume 2951 of *LNCS*, pages 1–20. Springer, 2004.
25. J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 387–394. ACM Press, 1990.
26. D. Schröder. *On the Complexity of Blind Signatures.* PhD thesis, Darmstadt University of Technology, 2010.