On the Security of One-Witness Blind Signature Schemes

Foteini Baldimtsi and Anna Lysyanskaya*

Department of Computer Science, Brown University, Providence, RI, USA
foteini@cs.brown.edu, anna@cs.brown.edu

Abstract. Blind signatures have proved an essential building block for applications that protect privacy while ensuring unforgeability, i.e., electronic cash and electronic voting. One of the oldest, and most efficient blind signature schemes is the one due to Schnorr that is based on his famous identification scheme. Although it was proposed over twenty years ago, its unforgeability remains an open problem, even in the random-oracle model. In this paper, we show that current techniques for proving security in the random oracle model do not work for the Schnorr blind signature by providing a meta-reduction which we call "personal nemesis adversary". Our meta-reduction is the first one that does not need to reset the adversary and can also rule out reductions to interactive assumptions. Our results generalize to other important blind signatures, such as the one due to Brands. Brands' blind signature is at the heart of Microsoft's newly implemented UProve system, which makes this work relevant to cryptographic practice as well.

Keywords: Blind signatures, meta-reduction technique, unforgeability, random oracle model.

1 Introduction

In a blind signature scheme, first introduced by Chaum in 1982 [16], a user can have a document signed without revealing its contents to the signer, and in such a way that the signer will not be able to recognize it later, when he sees the signature. Blind signatures have proven to be a very useful building block in applications requiring both anonymity and unforgeability, such as e-cash and anonymous credentials [12–15, 27].

Transactions that ensure unforgeability without violating privacy are of growing interest to cryptographic practice. The European Union E-Privacy Directive [31] limits the scope of the data that organizations are allowed to collect; so, to make sure that it is not in violation of this

* This work was supported by NSF grants 0964379 and 1012060.

directive, an online bank or vendor interacting with a user has an incentive to learn as little as possible about this user. Therefore, industry leaders such as Microsoft and IBM [30, 36] have been developing, implementing and promoting cryptographic software tools that promise the best of both worlds: unforgeability for banks and vendors, and privacy for users.

As a result, research on blind signatures has flourished, and provably secure solutions have been proposed based on well-established theoretical complexity assumptions in the standard model [14, 3, 22, 24] while some of these have been adapted for practical use by IBM [14]. However, schemes in the standard model either require exponentiation in the RSA group or bilinear pairings, which are typically considerably slower than, say, elliptic curve operations.

Thus, more efficient solutions that are provably secure in the random-oracle (RO) model [8] remain of practical importance [2, 9, 6]. Some of the earliest proposed schemes [12, 35, 23] do not have proofs of security even in the RO model; in fact, the security properties of the Schnorr blind signature is an important open problem. Moreover, Microsoft's UProve proposal [29, 30] is based on one of the unproven blind signatures, namely the one due to Brands [12]. UProve is currently part of a pilot project by NSTIC (National Strategy for Trusted Identities in the Cyberspace) that will be used quite extensively in a situation that will potentially affect millions of people [1]. Therefore, the security properties of these unproven but important blind signatures is a natural topic to look at.

In a nutshell, a blind signature scheme is secure if it satisfies two key properties: one-more unforgeability, which means that an adversary cannot produce more signatures than have been issued; and blindness, which means that an adversary cannot link a particular signature to a particular signing instance [33, 34].

The Schnorr blind signature scheme is the most efficient of all the blind signature schemes proposed in the literature given that it can be implemented using elliptic curves without pairings. It is constructed from the corresponding identification protocol via the Fiat-Shamir heuristic and some blinding operations. However, the security of this important scheme is an open problem. If the Schnorr identification scheme is not secure (i.e., after some number of interactions with the prover, the adversary can impersonate him), then the blind Schnorr signature is not one-more unforgeable. It is known that the Schnorr identification scheme cannot be proven secure under the discrete-logarithm assumption using black-box reductions in the standard model [32], so at the very least, it seems that Schnorr blind signatures require that we assume the security

of Schnorr identification (also studied by Bellare and Palacio [7]). Perhaps an even stronger assumption may be reasonable. Can we prove it secure under this or a stronger assumption?

To make this question more interesting, let us make it more general. Let us consider not just the Schnorr blind signature, but in general the blind variants of all Fiat-Shamir based signature schemes constructed along the lines described above: the signer acts as the prover in an identification protocol. And let us see if they can be proven secure under any reasonable assumption (by reasonable, we mean an assumption that is not obviously false), not just specific ones.

PS Reduction. Pointcheval and Stern showed that we can prove the security of blind signature schemes in the RO model when the underlying identification scheme is a witness-indistinguishable proof protocol for proving knowledge of a secret key, such that many secret keys are associated with the same public key [33, 34]. Their result does not apply to the original Schnorr blind signature, in which there is a single secret key corresponding to the public key. Other important blind signatures to which it does not apply are the ones due to Brands' (which are at the heart of Microsoft's UProve), and the ones based on the GQ signature [12, 23].

The idea of the Pointcheval-Stern reduction (also called "an oracle replay reduction") is to replay the attack polynomially many times with different random oracles in order to make the attacker successfully forge signatures. More precisely, we first run the attack with random keys, tapes and oracle f. Then, we randomly choose an index j and we replay with same keys and random tapes but with a new, different oracle f' such that the first $j-1$ answers are the same as before. We expect that, with non-negligible probability we will obtain two different signatures, σ, σ' of the same message m and we will be able to use them to solve a hard algorithmic problem (usually the one underlying the blind signature scheme) in polynomial time. This proof technique works for standard (i.e. not blind) versions of the Schnorr, Brands and GQ signatures. They also showed that it works for a modification of Schnorr blind signature which is less efficient than the original Schnorr's. A very natural question is: can it work for the original Schnorr blind signature and its generalizations, such as the Brands or GQ blind signatures?

Our results. Let us take a closer look at oracle replay reductions, as used by Pointcheval and Stern. Their reduction can be modeled as a Turing machine that has a special tape that is used specifically for answering random oracle queries; it always uses the next unused value when answering, afresh, the next random oracle query. We call this type of reductions:

Naive RO replay reductions and as we will discuss in Section 3.1 it can be used to model every known reduction for proving the security of digital signature schemes. Our result is that, in fact, naive RO replay reductions cannot be used to prove security of generalized Schnorr blind signatures, no matter how strong an assumption we make. Our result also holds for interactive assumptions or even if we assume the security of the blind signature scheme itself! Put another way, any such reduction can be used in order to break the underlying assumption.

Meta-Reductions. In our proof we make use of the "meta-reduction" method [10]: a separation technique commonly used to show impossibility results in cryptography. Let $\mathcal{A}$ be an adversary who breaks the unforgeability of generalized Schnorr blind signatures with non-negligible probability. We will use a meta-reduction (which we call "personal nemesis adversary") to show that there *cannot* exist a naive RO replay reduction, $\mathcal{B}$, which turns $\mathcal{A}$ into a successful adversary for *any* hard assumption that may be considered. We do that by transforming $\mathcal{B}$ through the meta-reduction into an algorithm that breaks the underlying assumption, without relying on the existence of a successful adversary.

What makes our technique particularly interesting is that for the *first time* we introduce a meta-reduction (our personal nemesis adversary) that does not need to reset the reduction $\mathcal{B}$, as it is usually done when using the meta-reduction paradigm [19]. For example, our personal nemesis adversary could reset the reduction $\mathcal{B}$, get an additional signature and return this signature back to $\mathcal{B}$ as his forgery. However, this resetting makes things more complicated since the two executions are correlated. Our technique, instead, is much simpler: the personal nemesis adversary, $p\mathcal{A}$, will simply interact with the reduction $\mathcal{B}$ the way an actual adversary would (but taking advantage of powers not available to an adversarial algorithm, such as remembering its prior state if and when the reduction resets it, and having access to the reduction's random oracle tape), without resetting it at any time. When $\mathcal{B}$ halts, if it succeeded in breaking the assumption (as it should with non-negligible probability, or it wouldn't be a valid security reduction), $p\mathcal{A}$ has succeeded too — but *without* assuming the existence of an actual adversary that breaks the security of the underlying signature scheme.

What are the implications of our results on the security of Schnorr blind signatures and generalizations? We must stress that our results do not in fact constitute an attack, and so for all we know, these schemes might very well be secure. However, we have essentially ruled out all *known* approaches to proving their security. So in order to give any secu-

rity guarantee on these signature schemes, the cryptographic community would have to come up with radically new techniques.

Related work. Schnorr and Jakobsson [18] proved security of the Schnorr blind signature in the combined random oracle and generic group model which is very restricted. Fischlin and Schröder [22] show that proving security of a broad class of blind signature schemes (which, in particular, includes what we refer to as generalized Schnorr blind signatures) via black-box reductions in the standard model is as hard as solving the underlying hard problem. Their technique uses the meta-reduction paradigm to show that black-box reductions for this type of blind signatures can be turned into solvers for hard non-interactive assumptions. However, their result does not rule out reductions in the RO model, and is technically very different from ours for that reason.

Rafael Pass studied the assumptions needed for proving security of various cryptographic schemes [32]. In particular, relevant to our work, he considers the Schnorr identification scheme and variants, and a category of blind signatures called "unique blind signatures." Pass considers whether so-called *r-bounded-round* assumptions are strong enough to prove, in a black-box fashion in the standard model, the security of certain schemes when repeated more than r times. His results apply to Schnorr blind signatures (and their generalizations) in the following way: he shows that no so-called bounded-round assumption can imply secure composition of the Schnorr identification scheme using black-box reductions (and therefore the Schnorr blind signature).

Here is how our work goes beyond what was shown by Pass [32] for "unique blind signatures." First of all, we do not limit our consideration to *r-bounded-round* assumptions but we show that our result applies for every possible intractability assumption. Thus, we rule out the existence of a very special type of reduction, the naive RO replay one, that models all the known reductions for proving security of digital signatures, *irrespective of assumption*. As an example, consider the One More Discrete Logarithm assumption (OMDL) [6] which has been used to prove security of the Schnorr identification scheme against active attacks [7]. Our result directly implies that Schnorr blind signature cannot be proven secure under the OMDL assumption in the RO model. Finally, our result applies even after *just one signature was issued* whereas Pass' result questions the security of schemes when repeated more than r times.

The meta-reduction technique has been used to analyze security of Schnorr signatures. Paillier and Vergnaud [28] showed that the security of Schnorr signatures cannot be based on the difficulty of the one

more discrete logarithm problem in the standard model. Fischlin and Fleischhacker [20] extended their result by showing that the security of Schnorr signatures cannot be based to the discrete logarithm problem without programming the random oracle. Their work is also relevant to ours since the meta-reduction they define also doesn't need to reset the reduction[1]. However, their result holds only for reductions to the discrete logarithm problem and applies to non-programming reductions while our naive RO replay reductions fall somewhere in between the programmable and non-programmable setting (see Section 3.1 for a discussion about programmability). Finally, their result only holds for a very limited class of reductions: those that run a single copy of the adversary which makes our work much broader.

2 Generalized Blind Schnorr Signature

First we explicitly define the class of blind signatures that our result applies to. For a complete presentation of all the necessary building blocks please refer to our full version [5].

In the signature scheme described by Schnorr [35] the signer's secret key is an exponent x, while his public key is $h = g^x$. A signature on a message m is obtained, via the Fiat-Shamir heuristic, from the Schnorr identification protocol, i.e. the three-round proof of knowledge of x. Thus, a signature on a message m is of the form $\sigma = (a, r)$ such that $g^r = ah^{H(m,a)}$, where H is a hash function that is modeled as a random oracle in the security proof. A blind issuing protocol was proposed for this signature back in the 1980s [18], and, on a high level, it works by having the user "blind" the value a he receives from the signer into some unrelated a', then the user obtains $c' = H(m, a')$ and, again, "blinds" it into some unrelated c which he sends to the signer. The signer responds with r which the user, again, "blinds" into r' such that (a', r') are a valid signature on m.

$\text{Signer}(q, g, h = g^x)$		$\text{User}(q, g, h, m)$
$y \leftarrow \mathbb{Z}_q,\ a = g^y$	$\xrightarrow{a}$	
	$\xleftarrow{c}$	$\alpha, \beta \leftarrow \mathbb{Z}_q,\ a' = ag^\alpha h^\beta,\ c' = H(m, a'),\ c = c' + \beta$
$r = y + cx \bmod q$	$\xrightarrow{r}$	$g^r \stackrel{?}{=} ah^c,\ r' = r + \alpha,$ output r', c'

The signature is: $\sigma(m) = (a', c', r')$ and the verification checks whether $g^{r'} = a'h^{c'}$.

[1] This is a result that Fischlin and Fleischhacker [20] obtained after the first version of our manuscript appeared on eprint [5]; our result is in fact the first in which a meta-reduction works without resetting the reduction $\mathcal{B}$.

Ever since this protocol was proposed, its security properties were an open problem. Okamoto proposed a modification [26]; Pointcheval and Stern proved security of this modification [33, 34]. Our work studies this blind signature and its generalizations, defined as follows:

Definition 1 (Generalized Blind Schnorr Signature). *A blind signature scheme* $(Gen, S, U, Verify)$ *is called Generalized Blind Schnorr Signature if:*

1. $(pk, sk) \in R_L$ *is a unique witness relation for a language* $L \in \mathcal{NP}$.
2. *There exists a* Σ*-protocol* (P, V) *for* R_L *such that for every* $(pk, sk) \in R_L$ *the prover's algorithm,* $P(pk, sk)$*, is identical to the signer's blind signing algorithm* $S(pk, sk)$.
3. *Let* $Sign(pk, sk, m)$ *be the signing algorithm implicitly defined by* (S, U). *Then, there exists a* Σ*-protocol* $P(pk, sk)$, $V(pk)$ *such that, in the random oracle (RO) model, a signature* $\sigma = (a, c, r)$*, where* $c = H(m, a)$ *is distributed identically to a transcript of the* Σ*-protocol.*
4. *There exists an efficient algorithm that on input* (pk, sk) *a "valid tuple"* (a, c, r) *and a value* c'*, computes* r' *s.t.* (a, c', r') *is a valid tuple. (By "valid tuple" we mean a signature for which the verification equation holds.) Note that no additional information about* a *is required, such as, e.g. its discrete logarithm.*

Let us now see why Schnorr's blind signature falls under the generalized blind Schnorr signature category. (1) The secret/public key pair is an instance of the DL problem which is a unique witness relation; (2) the signer's side is identical to the prover's side of the Schnorr identification scheme, which is known to be a Σ-protocol; (3) the signature $\sigma(m) = (a', c', r')$ is distributed identically to the transcript of the Schnorr identification protocol since a' comes uniformly at random from G; c' is truly random in the RO model, and r' is determined by α (4) finally, for a tuple (a, c, r) and a value c' one can compute $r' = r - cx + c'x$ so that (a, c', r') is still a valid tuple.

The definition also captures other well-known blind signature schemes, such as the blind GQ [23] and Brands [12] (for Brands also see Section 4).

3 Security of Generalized Blind Schnorr Signatures

We first define a general class of RO reductions and then prove that generalized blind Schnorr signature schemes cannot be proven unforgeable, and thus secure, using these reductions.

3.1 Naive RO replay reductions

We first explicitly describe the type of reductions that we rule out.

Definition 2 (Naive RO replay reduction). *Let $\mathcal{B}$ be a reduction in the random-oracle model that can run an adversary $\mathcal{A}$, and may also reset $\mathcal{A}$ to a previous state, causing $\mathcal{A}$ to forget $\mathcal{B}'s$ answers to its most recent RO queries. We assume, without loss of generality, that if $\mathcal{A}$ has already queried the RO on some input x, and hasn't been reset to a state that is prior to this query, then $\mathcal{A}$ does not make a repeat query for x.*
We say that $\mathcal{B}$ is a naive RO replay reduction *if: $\mathcal{B}$ has a special random tape for answering the RO queries as follows: when $\mathcal{A}$ queries the RO, $\mathcal{B}$ retrieves the next value v from its RO tape, and replies with $c = f(b, v)$ where b is the input to the reduction, and f is some efficiently computable function.*

Discussion Let us now take a closer look at known reductions for proving security of signatures in the RO model and see whether they fall under the naive RO replay reduction category. We first look at the reduction given by Pointcheval and Stern [33] for proving security of blind signatures. Their reduction could be easily modeled as a naive RO replay reduction with f being the identity function on its second input. PS reductions are *perfect* since they always create a signature. The same holds for the reduction given by Abe and Okamoto [4]. To convince the reader that our way of modeling reductions in the RO model is a very natural one, let us also look at the reduction given by Coron [17] proving the security of full domain hash (FDH) RSA signature. Coron's reduction works as follows: the reduction, $\mathcal{B}$, gets as input (N, e, y) where (N, e) is the public key and y is a random element from $\mathbb{Z}_N^*$ and tries to find $x = y^d \bmod n$. $\mathcal{B}$ runs an adversary $\mathcal{A}$, who can break the signature, with input the public key. As usual, $\mathcal{A}$ makes RO and signing queries which $\mathcal{B}$ answers. Whenever $\mathcal{A}$ makes an RO query, $\mathcal{B}$ picks a random $r \in \mathbb{Z}_n^*$ and either returns $h = r^e \bmod N$ with probability p or returns $h = yr^e \bmod N$ with probability $1 - p$. So, it is pretty straightforward that we could model Coron's reduction as a naive RO replay reduction by interpreting the contents of an RO tape as r and the output of a p-biased coin flip (return either r^e or yr^e). Other well-known reductions used in the literature to prove security of digital signatures in the RO model can be modeled as naive RO replay reductions as well [9, 6, 8].

Programmability Let us compare naive RO replay reductions with other previously defined types. Non-programmable random-oracle reductions [25]

do not give the reduction the power to set the answers to the RO queries; instead these answers are determined by some truly random function. Naive RO replay reductions can be more powerful than that: they can, in fact, answer the adversary's queries in some way they find convenient, by applying the function f to the next value of their RO tape. However, they are not as powerful as the general programmable RO reductions: naive RO replay reductions are not allowed, for example, to compute an answer to an RO query as a function of the contents of the query itself. Fischlin et al. [21] also consider an intermediate notion of programmability, called "random re-programming reductions", which are incomparable to ours.

3.2 Theorem for perfect naive RO replay reduction

Our first result is on a simpler class of reductions called "perfect". We will extend it to non-perfect reductions in Section 3.3.

Definition 3 (Perfect-Naive RO replay reduction). *A naive RO replay reduction $\mathcal{B}$ is called* perfect naive RO replay reduction *if $\mathcal{B}$ always gives valid responses to $\mathcal{A}$, i.e. its behavior is identical to that of the honest signer.*

We show that *perfect* naive RO replay reductions cannot be used to prove security of generalized blind Schnorr signature schemes.

Theorem 1. *Let $(Gen, S, U, Verify)$ be a generalized blind Schnorr signature scheme. Assume that there exists a polynomial-time perfect naive RO replay reduction $\mathcal{B}$ such that $\mathcal{B}^{\mathcal{A}}$ breaks an interactive intractability assumption C for every $\mathcal{A}$ that breaks the unforgeability of the blind signature (S, U). Then, C can be broken in polynomial time.*

Proof of theorem for perfect naive RO replay reduction We start by introducing some terminology. Note that the reduction $\mathcal{B}$ is given black-box access to $\mathcal{A}$ and is allowed to run $\mathcal{A}$ as many times as it wishes, and instead of running $\mathcal{A}$ afresh every time, it may reset $\mathcal{A}$ to some previous state. At the same time, $\mathcal{B}$ is interacting with its own challenger C; we do not restrict C in any way.

Consider how $\mathcal{B}$ runs $\mathcal{A}$. $\mathcal{B}$ must give to $\mathcal{A}$ some public key pk for the signature scheme as input. Next, $\mathcal{B}$ runs the blind signing protocol with $\mathcal{A}$; recall that a generalized blind Schnorr signing protocol always begins with a message a from the signer to the user. When $\mathcal{B}$ runs $\mathcal{A}$ again, it can choose to give it the same (pk, a) or different ones. It is helpful for

the description of the adversary we give, as well as for the analysis of the interaction, to somehow organize various calls that $\mathcal{B}$ makes to $\mathcal{A}$.

Every time that $\mathcal{B}$ runs $\mathcal{A}$, it either runs it "anew", providing a new public key pk and first message a, or it "resets" it to a previous state, in which some pk and a have already been given to $\mathcal{A}$. In the latter case, we say that $\mathcal{A}$ has been "reincarnated", and so, an *incarnation* of $\mathcal{A}$ is defined by (pk, a). Note that $\mathcal{B}$ may reincarnate $\mathcal{A}$ with the same (pk, a) several times. In this case, we say that this incarnation is *repeated*. Thus, if this is the i^{th} time that $\mathcal{A}$ has been reset to a previous state for this specific (pk, a), then we say that this is the i^{th} repeat of the (pk, a) incarnation. Without loss of generality, $\mathcal{B}$ never runs $\mathcal{A}$ anew with (pk, a) that it has used (i.e., if $\mathcal{B}$ has already created an incarnation for (pk, a), it does not create another one).

Let us consider what happens once $\mathcal{A}$ receives (pk, a). The signing protocol, in which $\mathcal{A}$ is acting as the user, expects $\mathcal{A}$ to send to $\mathcal{B}$ the challenge c. Additionally, $\mathcal{A}$ is free to make any random oracle queries it chooses. Once $\mathcal{B}$ receives c, the signing protocol expects it to send to $\mathcal{A}$ the response r. After that, the security game allows $\mathcal{A}$ to either request another signature, or to output a one-more signature forgery, i.e., a set of signatures (one more than it was issued); also, again, $\mathcal{A}$ can make RO queries. The adversaries that we consider in the sequel will not request any additional signatures, but will, at this point, output two signatures (or will fail).

Note that, if $\mathcal{B}$ is a perfect naive RO replay reduction, then it will always provide to $\mathcal{A}$ a valid response r to the challenge c; while if it is not perfect, then it may, instead, provide an invalid response, or stop running $\mathcal{A}$ at this point altogether. Thus, a particular run can be:

- Uncompleted: no valid response, r, was given by $\mathcal{B}$ at the end of the protocol (cannot happen if $\mathcal{B}$ is perfect).
- Completed but unsuccessful: a valid r was given but $\mathcal{A}$ was not able to output a forgery.
- Completed and successful: a valid r was given and $\mathcal{A}$ did output a forgery.

The technique we follow to prove our theorem is the following. We first define a special adversary which we call the *super adversary*, $s\mathcal{A}$, who exists if it is easy to compute the signing key for this signature scheme from the corresponding verification key. We do not show how to construct such an adversary (because we do not know how to infer the signing key for generalized blind Schnorr, and in fact we generally assume that it is impossible to do so in polynomial time); instead, we construct another

adversary, the *personal nemesis adversary*, $p\mathcal{A}$, whose behavior, as far as the reduction $\mathcal{B}$ can tell, will be identical to $s\mathcal{A}$.

Note that, generally, an adversary is modeled as a deterministic circuit, or a deterministic non-uniform Turing machine: this is because, inside a reduction, its randomness can be fixed. Thus, we need $s\mathcal{A}$ to be deterministic. Yet, we need to make certain randomized decisions. Fortunately, we can use a pseudorandom function for that. Thus, $s\mathcal{A}$ is parametrized by s, a seed to a pseudorandom function[2] $F_s : \{0,1\}^* \to \{0,1\}^k$. Additionally, it is parametrized by two messages m_1, m_2: signatures on these messages will be output in the end.

Consider $s\mathcal{A}_{s,m_1,m_2}$ that interacts with a signer as follows:

Definition 4 (Perfect super adversary $s\mathcal{A}_{s,m_1,m_2}$). *On input the system parameters:*

1. *Begin signature issue with the signer and receive* (pk, a).
2. *Find* sk.
3. *Use* sk *to compute the signatures: pick* a_1, a_2 *and make two RO queries* (m_1, a_1) *and* (m_2, a_2). *Produce two forged signatures for* m_1, m_2, *denote them as* σ_1 *and* σ_2 *(remember that* $s\mathcal{A}$ *is deterministic so if reincarnated he makes the same RO queries).*
4. *Resume the signature protocol with the signer: send to the signer the value* $c = F_s(trans)$ *where* $trans$ *is the current transcript between* $s\mathcal{A}_{s,m_1,m_2}$, *the RO and the signer, and receive from the signer the value* r *in response (which will always be valid for the perfect naive RO reduction* $\mathcal{B}$*).*
5. *Output the two message-signature pairs,* (m_1, σ_1) *and* (m_2, σ_2).

Note that when $s\mathcal{A}$ executes the signature issue protocol with the signer it computes c as a pseudorandom function of its current transcript with the RO and the signer. Thus, there is only a very small probability (of about 2^{-k}) for $s\mathcal{A}$ to send the same c in another run.

The next lemma follows directly from the definition of a reduction $\mathcal{B}$:

Lemma 1. *If a perfect naive RO replay reduction* $\mathcal{B}$ *exists, then* $\mathcal{B}^{s\mathcal{A}(\cdot)}$ $(pk, system\ params)$ *solves the assumption* C.

Lemma 1 works even if the assumption C is an interactive one. That is why, $s\mathcal{A}$ and $p\mathcal{A}$ are defined in such a way that they do not reset the reduction $\mathcal{B}$.

[2] We know that if $\mathcal{B}$ exists then secure signatures exist which imply one way functions existence and PRFs existence, so this is not an extra assumption.

Next, we define the personal nemesis adversary, $p\mathcal{A}$. Similarly to $s\mathcal{A}$, it is parametrized by (s, m_1, m_2); and so we denote it $p\mathcal{A}_{s,m_1,m_2}$. To the reduction $\mathcal{B}$, $p\mathcal{A}_{s,m_1,m_2}$ will look exactly the same as $s\mathcal{A}_{s,m_1,m_2}$, even though $p\mathcal{A}_{s,m_1,m_2}$ cannot compute sk. Instead, $p\mathcal{A}_{s,m_1,m_2}$ looks inside the reduction $\mathcal{B}$ itself; this is why we call $p\mathcal{A}_{s,m_1,m_2}$ "$\mathcal{B}$'s personal nemesis":

Definition 5 (Perfect $\mathcal{B}$'s personal nemesis adversary $p\mathcal{A}_{s,m_1,m_2}$). *On input the system parameters, $p\mathcal{A}_{s,m_1,m_2}$ performs a "one-more" forgery attack, using the following special powers: (1) $p\mathcal{A}_{s,m_1,m_2}$ has full access to $\mathcal{B}$'s random oracle tape; (2) in case $p\mathcal{A}_{s,m_1,m_2}$ is rewound, he remembers his previous state.*

$p\mathcal{A}_{s,m_1,m_2}$ performs the one-more forgery for $\ell = 1$. Thus, he runs one signature issuing session with the signer and then outputs two valid signatures. Specifically, in his ith incarnation, $p\mathcal{A}$ does the following:

1. *Begin signature issue with the signer, and receive (pk, a).*
2. *Do nothing ($p\mathcal{A}$ cannot find sk).*
3. – *If (pk, a) is the same as in some previous incarnation j then make the same RO queries as the last time this incarnation was run ($s\mathcal{A}$ remembers the previous RO queries; obviously it will receive different c_1, c_2 than before).*
   – *If (pk, a) is a new tuple, then this is a new incarnation; do the following:*
     - *If $p\mathcal{A}$ has already computed the sk for this pk, then use this power to forge two signatures on (m_1, m_2); call the resulting signatures σ_1 and σ_2,*
     - *else (if sk not already known), $p\mathcal{A}$ computes two signatures using its special access to $\mathcal{B}$ by looking in advance what the next c_1, c_2 are going to be, then picking random [3] r_1, r_2 and solving for a_1, a_2 using the third property of generalized blind Schnorr signatures and the simulator from the underlying Σ-protocol. $p\mathcal{A}$ makes two RO queries of the form $(m_1, a_1), (m_2, a_2)$ and gets c_1, c_2 in response. Call the resulting signatures σ_1 and σ_2.*
4. *Resume the signature issue protocol with the signer: send to the signer the value $c = F_s(trans)$ where trans is the current transcript between $p\mathcal{A}$, the RO and the signer, and receive from the signer the value r in response (which will be valid for the perfect naive RO reduction $\mathcal{B}$).*
5. – *If this is the first time for this incarnation, then output the two message-signature pairs, (m_1, σ_1) and (m_2, σ_2) (completed and successful run).*

[3] Recall that $p\mathcal{A}$ uses a PRF that takes as input its current state in order to make each random choice.

- *If this is a repeat of some incarnation j, and the value $c = F_s(trans) \neq c_j$, where c_j is the corresponding value from incarnation j, then using r and r_j, property 3 of generalized blind Schnorr signatures and the extractability of the Σ-protocol, compute sk (if you don't already know it for this pk). Next, compute σ_1 and σ_2 consistent with the RO queries from incarnation j, using property 4 of generalized blind Schnorr signatures (completed and successful run).*
- *If i is a repeat of j, and the value $c = F_s(trans) = c_j$, then fail (completed and unsuccessful run).*

Given the definition above it becomes clear why our naive RO reductions are not allowed to compute answers to the RO queries as a function of the query itself. It is important that the personal nemesis adversary has full access to the reduction's special RO tape and he should able to see what the next answer would be *before* forming his query. In particular, on the second case of Step 3 in Definition 5, $p\mathcal{A}$ first looks into $\mathcal{B}$'s RO tape to see what is the next c_1, c_2 and then formulates his RO query which depends on c_1, c_2. In this case, our analysis would break if the answer to the query was computed as a function of the content of the query itself.

Lemma 2. *If $\mathcal{B}$ is a perfect naive RO replay reduction, then $\mathcal{B}$'s view in interacting with $p\mathcal{A}_{s,m_1,m_2}$ is indistinguishable from its view when interacting with $s\mathcal{A}_{s,m_1,m_2}$.*

Proof. In order to prove this, we will analyze the behavior of $s\mathcal{A}$ and $p\mathcal{A}$ step by step, as they were defined, and we will show that $\mathcal{B}$ receives indistinguishable views when interacting with $s\mathcal{A}_s$ or $p\mathcal{A}_s$ with all but negligible probability (to simplify notation we will omit writing the messages m_1, m_2 to the parameters given to the adversaries). We begin by defining $s\mathcal{A}_{Rand}$ and $p\mathcal{A}_{Rand}$ who behave exactly as $s\mathcal{A}_s$ and $p\mathcal{A}_s$ do but using a truly random source instead of the pseudorandom function F_s. We will use the following hybrid argument: $s\mathcal{A}_s \approx s\mathcal{A}_{Rand} \approx p\mathcal{A}_{Rand} \approx p\mathcal{A}_s$.

Let us first argue that $s\mathcal{A}_s \approx s\mathcal{A}_{Rand}$. This follows by a straightforward reduction that contradicts the pseudorandomness of F_s. Similarly, it holds that $p\mathcal{A}_{Rand} \approx p\mathcal{A}_s$. We prove that $s\mathcal{A}_{Rand} \approx p\mathcal{A}_{Rand}$ by examining step by step the behavior of $s\mathcal{A}_{Rand}$ and $p\mathcal{A}_{Rand}$.

1. In the first step, both $s\mathcal{A}_{Rand}$ and $p\mathcal{A}_{Rand}$ begin the signature issuing with the Signer and wait for him to respond with (pk, a). For $\mathcal{B}$ there is no difference whether talking to $s\mathcal{A}_{Rand}$ or $p\mathcal{A}_{Rand}$.
2. In the second step there is no interaction with $\mathcal{B}$.

3. Here we have two different cases on $p\mathcal{A}_{Rand}$'s behavior depending on whether the current incarnation is *repeated* or not. In both cases the interaction between $p\mathcal{A}_{Rand}$ and $\mathcal{B}$ consists of $p\mathcal{A}_{Rand}$ making two RO queries where $p\mathcal{A}_{Rand}$ either makes two RO queries on fresh values that it computed on the current step or makes the same RO queries as in the *repeated* incarnation (so, there is no difference for $\mathcal{B}$). Thus, in Step 3, no matter who $\mathcal{B}$ is talking to, $\mathcal{B}$ receives two RO queries distributed identically.
4. Step 4 is identical for both $s\mathcal{A}_{Rand}$ and $p\mathcal{A}_{Rand}$. Just send $c = R(trans)$, where R is a random function and receive the value r in response.
5. Since r will always be a valid response (recall that $\mathcal{B}$ is perfect), $s\mathcal{A}_{Rand}$ will always output two message-signature pairs, (m_1, σ_1) and (m_2, σ_2). $p\mathcal{A}_{Rand}$ will also output (m_1, σ_1) and (m_2, σ_2), which are distributed identically to the ones output by $s\mathcal{A}_{Rand}$ unless it is the case that the incarnation is a repeat of j and $c = R(trans) = c_j$. In that case $p\mathcal{A}_{Rand}$ fails. The probability that $c = R(trans) = c_j$ is only $2^{-\Theta(k)}$. Thus, with probability $1 - 2^{-\Theta(k)}$ $\mathcal{B}$'s view is identical no matter whether he is talking to $s\mathcal{A}_{Rand}$ or $p\mathcal{A}_{Rand}$.

So, by the hybrid argument we defined at the beginning of the proof, it holds that $s\mathcal{A}_s \approx p\mathcal{A}_s$. □

Remark: we don't explicitly exploit blindness and in fact our result would go through even if a signature could be linkable to an issuing instance. For example, including the first message of the signer into the RO query would produce a contrived scheme in which the resulting signatures are linkable to the issuing instance; yet it would not affect our negative result.

3.3 Theorem for Non-perfect naive RO replay reductions

Let us apply our result to a broader class of reductions by removing the requirement that our reduction be perfect, i.e. always outputs valid responses. Instead, we will require an upper bound L on the number of times that the reduction can invoke the adversary which is independent of $\mathcal{A}$'s success probability. Note that, of course, $\mathcal{B}$'s success probability needs to depend on $\mathcal{A}$'s success probability. However, the number of times it invokes $\mathcal{A}$ need not; in fact known reductions (such as Coron or Pointcheval and Stern) as a rule only invoke the adversary a constant number of times.

Definition 6 (L-Naive RO replay reduction). *A naive RO replay reduction $\mathcal{B}$ is called* L-naive RO replay reduction *if there is a polynomial upper bound L on how many time $\mathcal{B}$ resets $\mathcal{A}$; this upper bound is a*

function of the number of RO queries that $\mathcal{A}$ makes, but otherwise is independent of $\mathcal{A}$, in particular, of $\mathcal{A}$'s success probability.

Our previous analysis wouldn't work for the *L-naive RO replay reduction.* Think of the scenario where $p\mathcal{A}$ receives a message a from $\mathcal{B}$ for the first time but is not given a valid r at the end. Then in the repeat of this incarnation, $p\mathcal{A}$ will have to make the same two RO queries he did before and output forgeries if given a valid r at the end. But, given the definitions of $\mathcal{B}$ and $p\mathcal{A}$ we gave before, $p\mathcal{A}$ will now get different c_1 and c_2 for his RO queries and thus he will not be able to output the same forgeries he had prepared before.

What changes in our new analysis is that: (a) $p\mathcal{A}$ is also given write access to $\mathcal{B}$'s RO tape, and (b) both $p\mathcal{A}$ and $s\mathcal{A}$ will be successful in producing a forgery with probability only $1/(\binom{L}{2} + L)$.

Theorem 2. *Let $(Gen, S, U, Verify)$ be a generalized blind Schnorr signature scheme. Suppose that there exists a polynomial-time L-naive RO replay reduction $\mathcal{B}$ such that $\mathcal{B}^{\mathcal{A}}$ breaks an intractability assumption C for every $\mathcal{A}$ that breaks the unforgeability of the blind signature (S, U). Then, C can be broken in polynomial time.*

This theorem rules out a broader class of security reductions. If we look back to our running example of Schnorr blind signatures, this theorem shows that under any assumption (DL, security of Schnorr identification, etc.) we cannot find an L-naive RO replay reduction to prove its security.

Proof of theorem for *L*-naive RO replay reduction Similar to what we did before, we first define the *super adversary* $s\mathcal{A}_{s,m_1,m_2,L}$ who knows L and works as follows:

Definition 7 (Super adversary $s\mathcal{A}_{s,m_1,m_2,L}$). *On input the system parameters:*

1. *Begin signature issue with the signer and receive (pk, a). Decide whether this is going to be a successful incarnation: choose "successful" with probability $1/(\binom{L}{2}+L)$ and "unsuccessful" with probability $1-1/(\binom{L}{2}+L)$.*
2. *Find sk.*
3. *Use sk to compute the signatures: pick a_1, a_2 and make two RO queries (m_1, a_1) and (m_2, a_2). Produce two forged signatures for m_1, m_2, denote them as σ_1 and σ_2.*

4. *Resume the signature protocol with the signer: send to the signer the value $c = F_s((trans))$ where trans is the current transcript between $s\mathcal{A}$, the RO and the signer, and receive from the signer the value r in response.*
5. – *If r is not valid, then this was an uncompleted run, then fail.*
   – *If r valid (completed run) and in Step 1 it was decided that this is a successful incarnation, output the two message-signature pairs, (m_1, σ_1) and (m_2, σ_2). Otherwise fail.*

The next lemma (similar to Lemma 1) follows from the definition of $\mathcal{B}$:

Lemma 3. *If an L-naive RO replay reduction $\mathcal{B}$ exists, then $\mathcal{B}^{s\mathcal{A}(\cdot)}$ $(pk, \text{system params})$ solves the assumption C.*

Now we are going to define the personal nemesis adversary, $p\mathcal{A}_{s,m_1,m_2,L}$.

Definition 8 ($\mathcal{B}$'s personal nemesis adversary $p\mathcal{A}_{s,m_1,m_2,L}$). *On input the system parameters, $p\mathcal{A}_{s,m_1,m_2,L}$ performs a "one-more" forgery attack, using the following special powers: (1) $p\mathcal{A}_{s,m_1,m_2,L}$ has full read and write access to $\mathcal{B}$'s random oracle tape; (2) in case $p\mathcal{A}_{s,m_1,m_2,L}$ is rewound, it does remember his previous state.*

$p\mathcal{A}_{s,m_1,m_2,L}$ performs the one-more forgery for $\ell = 1$. Thus, it runs one signature issuing session with the signer and then outputs two valid signatures with probability $1/(\binom{L}{2} + L)$. Specifically, in his i^{th} incarnation, $p\mathcal{A}_{s,m_1,m_2,L}$ does the following:

1. *Begin signature issue with the signer, and receive (pk, a).*
2. *Do nothing.*
3. – *If (pk, a) is received for the first time, then this is a new incarnation; do the following:*
     - *If $p\mathcal{A}$ has already found sk for this pk, then use this power to forge two signatures on (m_1, m_2) (still required to make two RO queries); call these signatures σ_1 and σ_2,*
     - *else, $p\mathcal{A}$ guesses (i_1, i_2) where $i_1 (\leq i_2)$ denotes the repeat where c_1 will be given in response to $p\mathcal{A}$'s next RO query; and i_2 is $p\mathcal{A}$'s guess for the first completed repeat of this incarnation. Then, $p\mathcal{A}$ randomly picks v_1, v_2, computes $c_1 = f(v_1), c_2 = f(v_2)$, picks r_1, r_2, solves for a_1, a_2 using the third property of generalized blind Schnorr signatures and the simulator from the underlying Σ-protocol and computes two signatures σ_1, σ_2.*
   – *$p\mathcal{A}$ makes two RO queries of the form $(m_1, a_1), (m_2, a_2)$ (the two RO queries are always the same for a specific incarnation).*

- *If this is the repeat incarnation i_1, and $\mathcal{B}$ wants a fresh answer to the query (m_1, a_1) then write v_1 on $\mathcal{B}$'s RO tape; else (if this isn't repeat i_1) write a random v_1'.*
- *If this is the repeat incarnation i_2 then write v_2 on $\mathcal{B}$'s RO tape; else (if this isn't repeat i_2) write a random v_2'.*

4. *Resume the signature issue protocol with the signer: send to the signer the value $c = F_s(trans)$ where F_s is a PRF and trans is the current transcript between $p\mathcal{A}$, the RO and the signer, and wait to receive the value r as a response from the signer.*
5. - *If r is valid (completed run):*
     - *If already know the secret key, sk, then output (m_1, σ_1) and (m_2, σ_2) with probability $1/(\binom{L}{2} + 2)$ or else fail.*
     - *If this is the first time for this incarnation, then output the two message-signature pairs, (m_1, σ_1) and (m_2, σ_2).*
     - *If this is the second successful repeat for this incarnation and the value $c = F_s(trans) \neq c_j$, where c_j is the corresponding value from the j^{th} run of this incarnation, then using r and r_j solve for sk using property 4 of generalized Schnorr signatures. Next, compute σ_1 and σ_2 consistent with the RO queries from this incarnation.*
     - *If this is the second successful repeat for this incarnation but $c = F_s(trans) = c_j$, then fail (unsuccessful run).*
     - *If the guess (i_1, i_2) was correct (that is, this is repeat i_2 of this incarnation, it was successful, and $\mathcal{B}$'s answer to (m_1, a_1) was the same as in incarnation i_1; and in incarnation i_1, $\mathcal{B}$ wanted a fresh answer to the (m_1, a_1) RO query) then output the two message-signature pairs, (m_1, σ_1) and (m_2, σ_2).*
     - *If the guess (i_1, i_2) was wrong then fail (unsuccessful run).*
   - *If r is not valid or r was not received then fail.*

Lemma 4. *If $\mathcal{B}$ is an L-naive RO replay reduction, then $\mathcal{B}$'s view in interacting with $p\mathcal{A}_{s,m_1,m_2}$ is indistinguishable from its view when interacting with $s\mathcal{A}_{s,m_1,m_2}$.*

The proof is similar to the one of Lemma 2 and can be found in the full version of the paper [5].

4 Brands' Blind Signature Scheme

Here we show that our results apply to the blind signature scheme given by Brands [11]. Let us first describe his construction. G is a group of order

q, where q a k-bit prime, and g is a generator of the group. The signer holds a secret key $x \leftarrow \mathbb{Z}_q$ and the corresponding public key $h = g^x$, while the user knows signer's public key h as well as g, q. $\mathcal{H}$ is a collision resistant hash function. The signature issuing protocol works as follows:

Signer (g, h, x)		User(g, h)
		$m = g^\alpha$
$w \in_R \mathbb{Z}_q$, $z \leftarrow m^x$, $a \leftarrow g^w$, $b \leftarrow m^w$	$\xleftarrow{\alpha}$	
	$\xrightarrow{z, a, b}$	
		$s, t \in_R \mathbb{Z}_q$, $m' \leftarrow m^s g^t$, $z' \leftarrow z^s h^t$
		$u, v \in_R \mathbb{Z}_q$, $a' \leftarrow a^u g^v$, $b' \leftarrow a^{ut} b^{us} (m')^v$
	$\xleftarrow{c}$	$c' \leftarrow \mathcal{H}(m', z', a', b')$, $c \leftarrow c'/u \bmod q$
$r \leftarrow w + cx \bmod q$	$\xrightarrow{r}$	$h^c a \stackrel{?}{=} g^r$, $z^c b \stackrel{?}{=} m^r$, $r' \leftarrow ur + v \bmod q$

A signature on m' is $\sigma(m') = (z', a', b', c', r')$. Anyone can verify a signature by first computing $c' = \mathcal{H}(m', z', a', b')$ and then checking whether the following equations hold: $h^{c'} a' \stackrel{?}{=} g^{r'}$, $(z')^{c'} b' \stackrel{?}{=} (m')^{r'}$.

4.1 Security of Brands' Blind Signatures

Corollary 1. *If there exists a perfect or an L-naive RO replay reduction $\mathcal{B}$ that solves any intractability assumption C using an adversary $\mathcal{A}$ that breaks the unforgeability of Brands' signature, then assumption C can be solved in polynomial time with non-negligible probability.*

In order for this corollary to hold we need to show that Brands' blind signature is a generalized blind Schnorr signature. We can show this by inspecting one by one the needed requirements: (1) Brands public/secret key pair is $(h = g^x, x)$, which is a unique witness relation for $L = \{h : g^x = h\} \in \mathcal{NP}$, (2) the signer's side of Brands blind signature is the same as the prover's side in Schnorr's identification scheme, which is known to be a Σ-protocol, (3) Brands blind signature is of the form $\sigma(m') = ((z', a', b'), c', r')$ which has identical distribution to a transcript of a Σ-protocol, as we will explain below (4) given the secret key x and a valid transcript of Brands scheme: $(\hat{a}, c'_1, r'_1)$, where $\hat{a} = (z', a', b')$, then $\forall\ c'_2$ we can compute r'_2 as: $r'_2 = r'_1 - c'_1 x + c'_2 x$ so that $(\hat{a}, c'_2, r'_2)$ is still a valid transcript. Brands blind signature is indeed a Σ-protocol: (a) it is a three-round protocol, (b) for any h and any pair of accepting conversations $(\hat{a}, c'_1, r'_1)$ and $(\hat{a}, c'_2, r'_2)$ where $c'_1 \neq c'_2$ one can efficiently compute x such that $h = g^x$ and (c) there exists a simulator S who on input h and a random c' picks r', m and z, solves for a', b', so he can output an accepting conversation of the form $((z', a', b'), c', r')$.

Thus, by applying Theorems 1 and 2, we rule out perfect and L-naive RO replay reductions for Brands' blind signatures.

Pointcheval and Stern [33] suggest that for their proof approach to work, the public key of the scheme should have more than one secret key associated with it. One could modify Brands' scheme similarly to how the original Schnorr blind signature was modified to obtain the variant that Pointcheval and Stern proved secure. In the full version [5] we propose such a modification; the public key of the signer will be of the form $H = G_1^{w_1} G_2^{w_2}$ where (H, G_1, G_2) are public and (w_1, w_2) are the secret key. As a blind signature, the resulting signature scheme is inferior, in efficiency, to the provably secure variant of the Schnorr blind signature. As far as its use in an electronic cash protocol is concerned, it is still an open problem whether provable guarantees against double-spending can be given for our modification of Brands.

References

1. http://www.nist.gov/nstic/pilot-projects2012.html. 2012.
2. Masayuki Abe. A secure three-move blind signature scheme for polynomially many signatures. In *EUROCRYPT'01*, pages 136–151, 2001.
3. Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In *CRYPTO'10*, pages 209–236. 2010.
4. Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In *CRYPTO '00*, pages 271–286. Springer-Verlag, 2000.
5. Foteini Baldimtsi and Anna Lysyanskaya. On the security of one-witness blind signature schemes. Cryptology ePrint Archive, Report 2012/197, 2012.
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*, 16:185–215, 2003.
7. Mihir Bellare and Adriana Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In *CRYPTO'02*, pages 162–177, 2002.
8. Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *ACM-CCS'93*, pages 62–73. ACM, 1993.
9. Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *PKC'03*, 2003.
10. Dan Boneh and Ramarathnam Venkatesan. Breaking rsa may not be equivalent to factoring. In *EUROCRYPT'98*, pages 59–71, 1998.
11. S. Brands. An efficient off-line electronic cash system based on the representation problem. In *CWI Technical Report CS-R9323*.
12. Stefan Brands. Untraceable off-line cash in wallets with observers. In *CRYPTO '93*, pages 302–318. Springer-Verlag, 1993.
13. Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In *EUROCRYPT'05*, pages 302–321. Springer-Verlag, 2005.
14. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT '01*, pages 93–118. Springer-Verlag, 2001.

15. D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *Proceedings on Advances in cryptology*, CRYPTO '88, pages 319–327. Springer-Verlag, 1990.
16. David Chaum. Blind signatures for untraceable payment. In *CRYPTO'82*, pages 199–203, 1982.
17. Jean-Sébastien Coron. On the exact security of full domain hash. In *CRYPTO '00*, pages 229–235. Springer-Verlag, 2000.
18. C.P.Schnorr and M.Jakobsson. Security of discrete log cryptosystems in the random oracle + generic model. In *The Mathematics of Public-Key Cryptography, The Fields Institute*, 1999.
19. Marc Fischlin. Black-box reductions and separations in cryptography. In *AFRICACRYPT'12*, pages 413–422, 2012.
20. Marc Fischlin and Nils Fleischhacker. Limitations of the meta-reduction technique: The case of schnorr signatures. *EUROCRYPT'13*, 2013.
21. Marc Fischlin, Anja Lehmann, Thomas Ristenpart, Thomas Shrimpton, Martijn Stam, and Stefano Tessaro. Random oracles with(out) programmability. In *ASIACRYPT'10*, pages 303–320, 2010.
22. Marc Fischlin and Dominique Schröder. On the impossibility of three-move blind signature schemes. In *EUROCRYPT'10*, pages 197–215, 2010.
23. Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In *EUROCRYPT '88*, pages 123–128, 1988.
24. Carmit Hazay, Jonathan Katz, Chiu yuen Koo, and Yehuda Lindell. Concurrently-secure blind signatures without random oracles or setup assumptions. In *TCC 2007*, 2007.
25. Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *CRYPTO'02*, pages 111–126, 2002.
26. Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53. Springer Berlin / Heidelberg, 1993.
27. Tatsuaki Okamoto and Kazuo Ohta. Universal electronic cash. In *CRYPTO '91*, pages 324–337. Springer-Verlag, 1992.
28. Pascal Paillier and Damien Vergnaud. Discrete-log-based signatures may not be equivalent to discrete log. In *ASIACRYPT*, pages 1–20, 2005.
29. Christian Paquin. U-prove cryptographic specification v1.1. In *Microsoft Technical Report, http://connect.microsoft.com/site1188*, February 2011.
30. Christian Paquin. U-prove technology overview v1.1. In *Microsoft Technical Report, http://connect.microsoft.com/site1188*, February 2011.
31. European Parliament and Council of the European Union. Directive 2009/136/ec. In *Official Journal of the European Union*, 2009.
32. Rafael Pass. Limits of provable security from standard assumptions. In *STOC*, pages 109–118, 2011.
33. David Pointcheval and Jacques Stern. Provably secure blind signature schemes. In *Asiacrypt '96, LNCS 1163*, pages 252–265. Springer-Verlag, 1996.
34. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal Of Cryptology*, 13:361–396, 2000.
35. Claus P. Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO '89*, pages 239–252. Springer-Verlag, 1989.
36. IBM Security Team. Specification of the identity mixer cryptographic library, version 2.3.0. In *IBM Research Report*, 2010.