On The Security of One-Witness Blind Signature Schemes

Foteini Baldimtsi and Anna Lysyanskaya

Computer Science Department, Brown University

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. 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, random oracle model, unforgeability.

1 Introduction

In a blind signature scheme, first introduced by Chaum in 1982 [13], a user can have a document signed without revealing the contents of the document 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 electronic cash (ecash) and anonymous credentials [9, 12, 11, 10, 3, 28]. For example, in the ecash application, a bank acts as the signer and signs a coin's serial number and some other useful information without seeing it, so that when the coin is later spent, the bank cannot link it to any specific user.

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 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, 37] have been developing, implementing and promoting cryptographic software tools that promise the best of both worlds: unforgeability for the banks and vendors, and privacy for the 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 [11, 2, 20, 24]; some of these have been adapted for practical use by IBM [11]. 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 remain of practical importance [1, 7, 4]. Some of the earliest proposed solutions [9, 36, 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 [9]. 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. These were formalized by Pointcheval and Stern [33, 35] who also proved that a variant of the Schnorr blind signature [27] satisfies these.

Recall the Schnorr signature scheme which is the most efficient of all the blind signature schemes proposed in the literature. 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's modeled as a random oracle in the security proof. A blind issuing protocol was proposed for this signature back in the 1980s [36], 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.

How secure is this blind signature? Note that, 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. Recently, Pass showed that the security of the Schnorr identification scheme cannot be proven 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 [5]). Perhaps an even stronger assumption may be reasonable. Can we prove it secure under even 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.

Pointcheval and Stern showed that we can prove the security of such 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. Their result does not apply to the original Schnorr blind signature, in which there is exactly one secret key corresponding to the public key. Other important blind signatures to which it does not apply are the Brands' blind signatures (the ones at the heart of Microsoft's UProve system), and the blind signatures based on the GQ signature [9, 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?

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. Our result is that, in fact, this type of oracle replay reductions cannot be used to prove security of generalized Schnorr blind signatures, no matter how strong an assumption we make. Put another way, any such reduction can be used in order to break the underlying assumption.

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

Related work Fischlin and Schröder [20] 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 random-oracle model, and in fact is technically very different from ours for that reason.

Recently, 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." His work is different from ours in that he 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; whereas we rule out, the existence of a very special type of reduction, irrespective of assumption even after just one signature was issued. 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).

Schnorr and Jakobsson [15] proved security of the Schnorr blind signature in the combined random oracle and generic group model. The generic group model is a very restricted setting in which the only way to sample group elements is by applying group operations; this does not correspond to any intractability assumption.

We study a very specific way of programming the random oracle reductions. Other variants of random oracle reductions have been considered in the literature and their relative strengths have been studied in prior work [19, 26]. In Section 3.1, we compare these variants with ours.

2 Preliminaries

Prior to presenting our main result, we are going to provide the necessary notation and definitions of the building blocks used in our proof. We will also explicitly define the class of blind signatures that our result applies to.

2.1 Unique witness relation

A witness relation for a language $L \in \mathcal{NP}$ is defined as [21]:

Definition 1 (Witness relation). *A witness relation for a language $L \in \mathcal{NP}$ is a binary relation R_L that is polynomially bounded (i.e., $(h,x) \in R_L$ implies $|x| \leq poly(|h|)$), is polynomial-time-recognizable and characterizes L by*

$$L = \{h : \exists x \ s.t. \ (h,x) \in R_L\}$$

For $h \in L$, any x satisfying $(h,x) \in R_L$ is called a *witness* (for the membership $h \in L$). By $R_L(h)$ we denote the set of witnesses for the membership $h \in L$; that is, $R_L(h) = \{x : (h,x) \in R_L\}$. If for each $h \in L$, there exists a unique $x \in R_L(h)$ then we say that R_L is a *unique-witness* relation.

An example of a unique witness relation is the discrete logarithm problem where $R_{G,g,q} = \{h, x \text{ s.t. } x \in \mathbb{Z}_q \text{ and } g^x = h\}$ and g, h are members of a cyclic group G of order q. Similarly, the RSA problem can be viewed as a unique witness relation where for a randomly chosen positive integer e, $R_N = \{h, x \text{ s.t. } x \in \mathbb{Z}_N^* \text{ and } x^e = h\}$ where $N = pq$ and p, q are distinct odd primes. For formal definitions of the DL and RSA problem refer to Appendix A.

2.2 Intractability Assumptions

We will use the definition given by Pass [32]: an intractability assumption is modeled as an interaction between a probabilistic machine C (the challenger) and an attacker A where they are both given as input 1^k (k is the security parameter). A's running time is measured as a function of k.[1] Once A halts, the challenger outputs 1 or 0. Any challenger C together with a threshold function $t(\cdot)$ intuitively corresponds to the assumption:

> *For every polynomial time adversary A there exists a negligible function ν such that for all k, the probability that C outputs 1 after interacting with A is bounded by $t(k) + \nu(k)$.*

We say that A breaks C with respect to t with advantage p if:

$$Pr[\langle A, C \rangle(1^k) = 1] \geq t(k) + p.$$

As Pass [32] notes, we can easily model all standard cryptographic assumptions as a challenger C and a threshold t. For example, the discrete logarithm assumption corresponds to the threshold $t(k) = 0$ and the 2-round challenger C who on input 1^k picks a random x and sends g^x to A. If the attacker responds with $x' = x$ then C outputs 1.

2.3 Σ-Protocols

Σ-protocols are a class of interactive proofs (see Appendix B) where (P, V) have a common input h and an x such that $(h,x) \in R_L$ is P's private input. Their main characteristic is that they have exactly 3 rounds of the following type: (1) P sends a message a to V, (2) V responds with a random challenge c chosen from a domain of size $\Theta(k)$ and (3) P resends a reply r. V decides whether to accept or not given the information he has seen: (h, a, c, r). Formally:

Definition 2 (Σ-Protocol). *A protocol $\mathcal{P}$ is said to be a Σ-protocol for a relation R_L if:*

- *$\mathcal{P}$ is of the above three rounds form, and if (P, V) follow the protocol, the verifier always accepts.*

[1] Pass also requires that there be a limit to the rounds of interaction between A and C: an r-bounded assumption is one in which there exists some polynomial $r(\cdot)$ such that C on input 1^k communicates with A for at most $r(k)$ rounds; in this paper, however, assumptions that do not bound the number of rounds are still meaningful.

– *From any h and any pair of accepting conversations on input h, (a, c, r), (a, c', r') where $c \neq c'$, one can efficiently compute x such that $(h, x) \in R_L$ (special soundness).*
– *There exists a polynomial-time simulator S, which on input h and a random c outputs an accepting conversation of the form (a, c, r), with the same probability distribution as conversations between the honest P, V on input h (special honest-verifier zero-knowledge).*

A Σ-protocol is said to be unique-witness *Σ-protocol ($UW\Sigma$) if R_L is a unique-witness relation.*

An example of a Σ-protocol is the Schnorr Identification scheme [36]. Let G be a group of prime order q with generator g, and let $\mathbb{Z}_q$ denote the field of integers modulo q. Schnorr's identification scheme works as follows:

$\text{Prover}(q, g, h = g^x)$		$\text{Verifier}(q, g, h)$
$y \leftarrow \mathbb{Z}_q$, $a = g^y$	$\xrightarrow{a}$	
	$\xleftarrow{c}$	$c \leftarrow \mathbb{Z}_q$
$r = y + cx \bmod q$	$\xrightarrow{r}$	$g^r \stackrel{?}{=} ah^c$

Σ-protocols are an essential building block for blind signatures and anonymous credentials. For example Brands [9] scheme is based on a Σ-protocol while CL anonymous credentials [11] uses ZK proofs which are based on Σ-protocols.

2.4 Fiat-Shamir Heuristic

Fiat and Shamir [18] proposed a method to transform any three-round interactive proof system with negligible soundness error, like Σ-protocols, into a digital signature scheme using a hash function, modeled as a random oracle.

To transform a three-round proof system into a signature scheme, one could, instead of a random c, compute $c = H(a, m)$, where $H \rightarrow \{0,1\}^*$ is a hash function. Then, the Fiat-Shamir transformation uses c to create a signature $\sigma(m) = (a, r)$. Specifically:

– $\text{Gen}(1^k) : (h, x) \leftarrow R_L, \text{SK} = x, \text{PK} = (h, H)$.
– $\text{Sig}(m)$: produce a honestly using x, set $c = H(m, a)$, produce r honestly for this c, output $\sigma(m) = (a, r)$.
– $\text{Ver}(m, (a, r))$: $c = H(a, m)$ and check if (a, c, r) is valid.

Famous digital signatures that have been constructed from Σ-protocols using the Fiat-Shamir heuristic include Schnorr's [36] and GQ signatures [23] and they have been proven secure in the RO model [33].

Going back to our running example, Schnorr's identification scheme can be easily turned into a signature scheme using the Fiat-Shamir heuristic. The Signer has a secret/public key pair (h, x) and a message m. To sign m the following steps take place: (1) $y \leftarrow \mathbb{Z}_q$, (2) $a = g^y$, (3) $c = H(m, a)$, and (4) $r = y + cx \bmod q$. The signature on the message is $\sigma(m) = (c, r)$ and in order to verify the signature, one should check whether $w = H(m, g^r / y^c)$.

2.5 Blind Signatures

Blind signatures are a special case of digital signatures. They have a blind signature issuing protocol in which the signer doesn't learn anything about the message he is signing. Formally, a blind signature scheme is a four-tuple consisting of two interactive Turing machines, the Signer and the User, (S, U) and two algorithms (*Gen*, *Verify*) [25].

– $Gen(1^k)$: is a PPT key-generation algorithm which takes as an input a security parameter 1^k and outputs a pair (pk, sk) of public and secret keys.
– $S(sk, pk)$, $U(pk, m)$: S and U engage in an interactive protocol of some polynomial (in the security parameter) number of rounds. At the end of this protocol S outputs either "completed" or "not-completed" and U outputs either "fail" or $\sigma(m)$. The signing algorithm $Sign(sk, pk, m)$ is implicitly defined from S and R (i.e., in order to compute a signature σ on a message m, the signer simulates the interaction between $S(sk, pk)$ and $U(pk, m)$).

– *Verify*$(pk, m, \sigma(m))$: is a deterministic polynomial-time algorithm, which outputs "accept"/"reject" with the requirement that for any message m, and for all random choices of key generation algorithm, if both S and U follow the protocol then S always outputs "completed", and the output of U is always accepted by the verification algorithm.

A blind digital signature scheme is *secure* if (informally) for all the probabilistic polynomial time algorithms $\mathcal{A}$ there exists a security parameter $k_{\mathcal{A}}$ such that for all $k > k_{\mathcal{A}}$ the following two properties hold [25]:

- Blindness: the signer is unable to view the messages he signs (protection for the user). Furthermore, a malicious signer cannot link a $(m, \sigma(m))$ pair to any particular execution of the protocol.

-One-more Unforgeability: a user interacting with a signer S cannot output an additional, valid message/signature pair $(m, \sigma(m))$ no matter how many (ℓ) pairs of messages/signatures of S he has seen (protection for the signer).

For the full versions of the above definitions please refer to [25] or Appendix C.

Blind signature issuing protocols have been proposed for many traditional digital signature schemes constructed via the Fiat-Shamir heuristic. For the Schnorr signature, which is our running example, the proposed blind issuing protocol [15] would work as follows:

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

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

Ever since this protocol was proposed, its security properties were an open problem. Okamoto proposed a modification of the protocol [27]; Pointcheval and Stern proved security of this modification [33, 35]. Our work studies this blind signature and its generalizations, defined as follows:

Definition 3 (Generalized Blind Schnorr Signature). *A blind signature scheme* $(Gen, S, U, Verify)$ *is called Generalized Blind Schnorr Signature if:*

1. $(sk, pk) \in R_L$ *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* $(sk, pk) \in R_L$ *the prover's algorithm,* $P(sk, pk)$*, is identical to the signer's blind signing algorithm* $S(sk, pk)$.
3. *Let* $Sign(sk, pk, m)$ *be the signing algorithm implicitly defined by* (S, U)*. Then, there exists a* Σ*-protocol* $P(sk, pk)$, $V(pk)$ *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* (sk, pk) *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.)*

Let's go back to our running example of Schnorr's blind signature and see why it 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 [9] (for Brands also see Section 4).

3 Security of Blind Signatures

As we mentioned above, one-more unforgeability of generalized blind Schnorr signatures is an open problem. In this section we will first present a general class of RO reductions that can be used to model

all the known RO reductions in the literature used to prove the security of digital signatures. Then, we will 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 our result rules out.

Definition 4 (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's 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 $c = f(input, v)$ where input is the input to the reduction, and f is some function.*
$\mathcal{B}$ is a 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.*

Let's 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. PS reductions are *perfect* since they always create a signature. The same holds for the reduction given by Abe [1].

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 [14] 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 [7, 4, 6].

Programmability Let us compare naive RO replay reductions with other previously defined types. Non-programmable random-oracle reductions [26] 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. [19] also consider an intermediate notion of programmability, called "random re-programming reductions", which are incomparable to ours (but it would be interesting to extend our results to these reductions as well).

3.2 Theorem for perfect naive RO replay reduction

Our first result is that *perfect* naive RO replay reductions cannot be used to prove security of generalized blind Schnorr signature schemes. We will extend it to non-perfect reduction in Section 3.3.

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 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.*

We prove this theorem below. What are the consequences of this theorem for the Schnorr blind signatures, which is our running example? What we have shown is that, even if we assume security of the Schnorr identification scheme, and not just the hardness of the discrete logarithm problem, we still cannot exhibit a perfect naive RO replay reduction that will prove Schnorr blind signatures secure. In fact, somewhat oddly, even if we assume that the Schnorr blind signature scheme is secure, we still cannot find a perfect naive RO replay reduction $\mathcal{B}$ that will break this assumption should $\mathcal{A}$ be able to violate the unforgeability of the scheme. This is because a perfect naive reduction requires that the hash function queries be handled in a very specific way.

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 (as defined above), 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 $F_s : \{0,1\}^* \to \{0,1\}^k$ [2]. Additionally, it is parameterized by two messages m_1, m_2: signatures on these messages will be output in the end.

[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

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

Definition 5 (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.*
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 (2^{-k}) for $s\mathcal{A}$ to send the same c in another run.

The following 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.*

Next, we define the personal nemesis adversary, $p\mathcal{A}$. Similarly to $s\mathcal{A}$, it is parameterized 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 6 (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 it's 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) are the same as in some previous incarnation j, then make the same RO queries as the last time this incarnation was run.*
   – *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 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).*
   – *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. Next, compute σ_1 and σ_2 consistent with the RO queries from incarnation j, using property 4 of generalized blind Schnorr signatures.*
   – *If i is a repeat of j, and the value $c = F_s(trans) = c_j$, then fail (unsuccessful run).*

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 $p\mathcal{A}_{Rand} \approx p\mathcal{A}_s$.

Let's now 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). From the point of view of $\mathcal{B}$ 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 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 from the signer 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 $s\mathcal{A}_s \approx p\mathcal{A}_s$. □

3.3 Non-perfect naive RO replay reductions

Let's 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 on the number of times that the reduction can invoke the adversary.

Definition 7 (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:

- $p\mathcal{A}$ is also given write access to $\mathcal{B}$'s RO tape
- both $p\mathcal{A}$ and $s\mathcal{A}$ will be successful in producing a forgery with probability only $1/(\binom{L}{2} + L)$.

3.4 Theorem for *L*-naive RO replay reductions

Our second result is that L-naive RO replay reductions cannot be used to prove security of generalized blind Schnorr signature schemes.

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 8 (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 $\frac{1}{\binom{L}{2}+L}$ and "unsuccessful" with probability $1 - \frac{1}{\binom{L}{2}+L}$.*
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.*
6. *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 following 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 $\mathcal{B}^{s\mathcal{A}(\cdot)}$ $(pk, system\ params)$ solves the assumption C.*

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

Definition 9 ($\mathcal{B}$'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 $\frac{1}{\binom{L}{2}+L}$. Specifically, in it's i^{th} incarnation [3], $p\mathcal{A}_{s,m_1,m_2,L}$ 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); 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 complete repeat of this incarnation. Then, $p\mathcal{A}$ randomly picks v_1, v_2, sets $c_1 = f(v_1), c_2 = f(v_2)$, picks r_1, r_2 and solves 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.*

[3] Recall that the terms "incarnation", "completed" run, "successful run were defined in Section 3.2

- *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 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 $s\mathcal{A}_{s,m_1,m_2}$.*

Proof. Similarly to the proof of Lemma 2, we first consider $p\mathcal{A}$ and $s\mathcal{A}$ that, instead of access to a pseudorandom function F_s have access to a truly random function *Rand*. Just as before, by pseudorandomness of F_s, $p\mathcal{A}_{s,m_1,m_2} \approx p\mathcal{A}_{Rand,m_1,m_2}$ and $s\mathcal{A}_{s,m_1,m_2} \approx s\mathcal{A}_{Rand,m_1,m_2}$; so it is sufficient to show that $p\mathcal{A}_{Rand,m_1,m_2} \approx s\mathcal{A}_{Rand,m_1,m_2}$. (We will omit the subscripts "$Rand, m_1, m_2$" in the rest of the proof.)

Consider $\mathcal{B}$'s view when interacting with $s\mathcal{A}$ for fixed (pk, a), i.e. in a given incarnation. Until $\mathcal{B}$ completes the incarnation by sending a valid response r, $\mathcal{B}$ does not know whether this incarnation is successful or not; thus $\mathcal{B}$'s view with $s\mathcal{A}$ is identical to his view with $s\mathcal{A}'$ defined as follows: $s\mathcal{A}'$ remembers previous times when $\mathcal{B}$ ran it. It is identical to $s\mathcal{A}$, except that it decides (at random) whether or not this incarnation is successful the first time that $\mathcal{B}$ correctly completes this incarnation by sending to $s\mathcal{A}'$ the correct r in Step 4. The way that $s\mathcal{A}'$ will determine whether this is a successful incarnation is by picking (i_1, i_2) the way that $p\mathcal{A}$ does, and then making the incarnation successful if it picked them correctly; note that $s\mathcal{A}'$ makes an incarnation successful if it picks the unique correct (i_1, i_2) out of $\binom{L}{2} + L$ possibilities ($\binom{L}{2}$ ways of picking $i_1 \neq i_2$, L ways to pick $i_1 = i_2$).

Next, let us compare $\mathcal{B}$'s view with $s\mathcal{A}'$ with his view with $p\mathcal{A}$. They make identically distributed queries to the random oracle; then they successfully produce forgeries whenever they have correctly guessed i_1 and i_2 (except if $p\mathcal{A}$ sends the same query c in both the first and the second complete run of this incarnation, which happens with only negligible probability). Therefore, the views that $\mathcal{B}$ receives when talking to $s\mathcal{A}'$ and $p\mathcal{A}$ are statistically indistinguishable, which completes the proof of the lemma.

4 Brands' Blind Signature Scheme

In this section we show that our results apply to the blind signature scheme given by Brands in [8] which constitutes the base for his well-known e-cash system.

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

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

Brands never gave a formal proof of security for his blind signatures. In this section, we argue that the security of his scheme cannot be proved via a perfect naive or an L-naive RO replay reduction.

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. Let's take a closer look at Brands blind signature and see why it is a Σ-protocol. We will do so by inspecting the three properties of Σ-protocols: (a) it's 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. [4]

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 the Brands' scheme (similarly to how the original Schnorr blind signature was modified to obtain the variant that Pointcheval and Stern proved secure). In this 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 (the full scheme and its proof of security appear in Appendix E). 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. (For the original Brands electronic cash, the best double-spending guarantee known, discovered by Cramer et al. [16], holds under the knowledge of exponent assumption [17], which is an assumption so strong that it cannot, for example, be captured by Pass's definition of an intractability assumption.)

Thus, although Brands' e-cash scheme has attractive efficiency and blindness properties, proving its unforgeability would require radically new techniques.

[4] Brands claimed that his scheme is secure under the existence of a Schnorr prover [9], though he never gave a proof on that. In Appendix D we define the DL problem under a Schnorr prover and show that it can be modeled as an intractability assumption.

References

1. Masayuki Abe. A secure three-move blind signature scheme for polynomially many signatures. In *EUROCRYPT 2001: International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001*, pages 136–151, 2001.
2. Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In *Advances in Cryptology - CRYPTO 2010*, pages 209–236. 2010.
3. Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Randomizable proofs and delegatable anonymous credentials. In *Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology*, pages 108–125, Berlin, Heidelberg, 2009. Springer-Verlag.
4. 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.
5. Mihir Bellare and Adriana Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, pages 162–177, 2002.
6. Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, CCS '93, pages 62–73. ACM, 1993.
7. Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography: Public Key Cryptography*, PKC '03. Springer-Verlag, 2003.
8. S. Brands. An efficient off-line electronic cash system based on the representation problem. In *CWI Technical Report CS-R9323*.
9. Stefan Brands. Untraceable off-line cash in wallets with observers. In *CRYPTO '93: 13th Annual International Cryptology Conference*, pages 302–318. Springer-Verlag, 1993.
10. Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In *In EUROCRYPT '05, volume 3494 of LNCS*, pages 302–321. Springer-Verlag, 2005.
11. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT '01, pages 93–118, London, UK, 2001. Springer-Verlag.
12. D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *Proceedings on Advances in cryptology*, CRYPTO '88, pages 319–327. Springer-Verlag New York, Inc., 1990.
13. David Chaum. Blind signatures for untraceable payment. In *Advances in cryptology,Crypto'82, Lect. Notes Computer Science*, pages 199–203, 1982.
14. Jean-Sébastien Coron. On the exact security of full domain hash. In *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '00, pages 229–235, London, UK, 2000. Springer-Verlag.
15. 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.
16. Ronald Cramer, Ivan Damgard, and Jesper Buus Nielsen. On electronic payment systems. In *http://www.daimi.au.dk/ ivan/ecash.pdf*, 2010.
17. Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In *CRYPTO '91*, pages 445–456. Springer-Verlag, 1992.
18. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology CRYPTO '86*, pages 186–194. Springer-Verlag, 1986.
19. Marc Fischlin, Anja Lehmann, Thomas Ristenpart, Thomas Shrimpton, Martijn Stam, and Stefano Tessaro. Random oracles with(out) programmability. In *ASIACRYPT*, pages 303–320, 2010.
20. Marc Fischlin and Dominique Schroder. Security of blind signatures under aborts. In *Public Key Cryptography - PKC 2009, 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings*, pages 297–316, 2009.
21. O. Goldreich. *Foundations of Cryptography, vol. 1: Basic Tools*. Cambridge University Press, 2001.
22. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18:186–208, February 1989.
23. Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In *Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings*, 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 *In TCC 2007*, 2007.
25. Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In *Crypto'97: Proceedings of the 17th annual international cryptology conference on advances in cryptology*, pages 150–164. Springer-Verlag, 1997.
26. Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *Advances in Cryptology - CRYPTO 2002*, pages 111–126, 2002.
27. Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *Advances in Cryptology - CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53. Springer Berlin / Heidelberg, 1993.
28. Tatsuaki Okamoto and Kazuo Ohta. Universal electronic cash. In *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pages 324–337, London, UK, 1992. Springer-Verlag.
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 proofs for signature schemes. In *EUROCRYPT '96*, pages 387–398. Springer-Verlag, 1996.
35. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal Of Cryptology*, 13:361–396, 2000.
36. Claus P. Schnorr. Efficient identification and signatures for smart cards. In *Proceedings on Advances in cryptology*, CRYPTO '89, pages 239–252. Springer-Verlag New York, Inc., 1989.
37. IBM Security Team. Specification of the identity mixer cryptographic library, version 2.3.0. In *IBM Research Report*, 2010.

A Standard Assumptions

We start by describing two standard cryptographic assumptions useful in the rest of the paper.

Definition 10 (Discrete Logarithm Assumption). *Let k be the security parameter. Let G be an abelian group of order q (k-bit prime) and generator g. Then, for every polynomial time algorithm $\mathcal{A}$ it holds that:*

$$Pr[h \leftarrow G; x \leftarrow A(h) : x = \log_g h] \leq \nu(k)$$

where $\nu(k)$ is a negligible function.

Definition 11 (RSA Assumption). *Let k be the security parameter. Let $N = pq$ where p and q are k-bit, distinct odd primes. Let e be a randomly chosen positive integer less than and relatively prime to $\phi(N) = (p-1)(q-1)$. Then, for every polynomial time algorithm $\mathcal{A}$ it holds that:*

$$Pr[h \leftarrow \mathbb{Z}_N^*; x \leftarrow A(N, e) : x^e \equiv h \bmod N] \leq \nu(k)$$

where $\nu(k)$ is a negligible function.

B Interactive Proofs

Informally speaking, an interactive proof system for a language L is a two-party protocol between a prover and a probabilistic polynomial-time verifier satisfying the following two conditions with respect to the common input, denoted h. If $h \in L$, then, with very high probability, the verifier is "convinced" of this fact, after interacting with the prover (completeness). If $h \notin L$, then no matter what the prover does, with very high probability, he fails to convince the verifier that "h is in L"(soundness).

Definition 12 (Interactive Proof System [22]). *An interactive proof system with soundness error $s \in [0,1]$for a language L with witness relation R_L is a pair of algorithms (P, V) where v is probabilistic polynomial time and the following properties hold:*

1. *Completeness. For every $h \in L$ and every $x \in R_L(h)$,*

$$Pr[\langle P(x), V_L\rangle(h) = 1] = 1.$$

2. *s-Soundness. For every $h \notin L$, every $z \in \{0,1\}^*$ and every interactive algorithm P^**

$$Pr[\langle P^*(z), V\rangle(h) = 0] \geq 1 - s.$$

A useful property in the above setting would be if the verifier V wouldn't learn anything useful from P about the witness x besides the fact that P knows x. This property is called *zero knowledge*. If soundness error s is negligible, then this interactive proof system has *strong soundness*.

Definition 13 (Honest Verifiable Zero Knowledge (HVZK)). *An interactive proof system (P, V) for a language L is said to be* honest verifiable zero knowledge *if there exists a probabilistic polynomial time algorithm S (the Simulator) such that for all $h \in L$:*

$$view_V[P(h) \leftrightarrow V(h)] \approx S(h),$$

where view_V is the view of the honest verifier V of the interaction between V and P on input h.

C Blind Signatures Definitions

We present the formal definitions for blind signatures as they were described in [25]. A blind signature scheme is a four-tuple consisting of two interactive Turing machines, the Signer and the User, (S, U) and two algorithms (Gen, $Verify$).

- $Gen(1^k)$: is a probabilistic polynomial time key-generation algorithm which takes as an input a security parameter 1^k and outputs a pair (pk, sk) of public and secret keys.
- $S(pk, sk)$, $U(pk, m)$: are polynomially- bounded probabilistic Interactive Turing machines who have the following (separate) tapes: read-only input tape, write-only output tape, a read/write work tape, a read-only random tape, and two communication tapes, a read-only and a write-only tape. They are both given (on their input tapes) as a common input a pk produced by a key generation algorithm. Additionally, S is given on her input tape a corresponding private key sk and U is given on her input tape a message m, where the length of all inputs must be polynomial in the security parameter 1^k of the key generation algorithm. Both S and U engage in an interactive protocol of some polynomial (in the security parameter) number of rounds. At the end of this protocol S outputs either "completed" or "not-completed" and U outputs either "fail" or $\sigma(m)$.
- $Verify(pk, m, \sigma(m))$: is a deterministic polynomial-time algorithm, which outputs "accept"/"reject" with the requirement that for any message m, and for all random choices of key generation algorithm, if both S and U follow the protocol then S always outputs "completed", and the output of U is always accepted by the verification algorithm.

A blind digital signature scheme is *secure* if for all the probabilistic polynomial time algorithms $\mathcal{A}$ there exists a security parameter $k_{\mathcal{A}}$ such that for all $k > k_{\mathcal{A}}$ the following two properties hold [25]:

- Blindness: the signer is unable to view the messages he signs (protection for the user). Furthermore, a malicious signer cannot link a $(m, \sigma(m))$ pair to any particular execution of the protocol. In order to define blindness formally consider the following experiment. Let $\mathcal{A}$ control the signer but not the user and $b \in \{0, 1\}$ is a randomly chosen bit which is kept secret from $\mathcal{A}$. $\mathcal{A}$ will try to guess the value of b by performing the following steps:

1. $(pk, sk) \leftarrow Gen(1^k)$
2. $\{m_0, m_1\} \leftarrow \mathcal{A}(1^k, pk, sk)$ (i.e. $\mathcal{A}$ produces two documents, polynomial in 1^k, where $\{m_0, m_1\}$ are by convention lexicographically ordered and may even depend on pk and sk).
3. We denote by $\{m_b, m_{1-b}\}$ the same two documents $\{m_0, m_1\}$, ordered according to the value of bit b, where the value of b is hidden from $\mathcal{A}$. $\mathcal{A}(1^k, pk,\ sk, m_0, m_1)$ engages in two parallel (and arbitrarily interleaved) interactive protocols, the first with $U(pk, m_b)$ and the second with $U(pk, m_{1-b})$.
4. If the fist User outputs on her private tape $\sigma(m_b)$ (i.e. does not output fail) and the second user outputs on her private tape $\sigma(m_{1-b})$ (i.e., also does not output fail) then $\mathcal{A}$ is given as an additional input $\{\sigma(m_0), \sigma(m_1)\}$.(We remark that we do not insist that this happens, and either one or both users may output fail).
5. $\mathcal{A}$ outputs a bit b' (given her view of steps 1 through 3, and if conditions are satisfied on step 4 as well).

Then the probability, taken over the choice of b, over coin-flips of the key-generation algorithm, the coin-flips of $\mathcal{A}$, and (private) coin-flips of both users (from step 3), that $b' = b$ is at most $\frac{1}{2} + \nu(k)$, where $\nu(k)$ is a negligible function.

-One-more Unforgeability: a user interacting with a signer S cannot output an additional, valid message/signature pair $(m, \sigma(m))$ no matter how many pairs of messages/ signatures of S he has seen (protection for the signer). To define that formally, consider an adversary $\mathcal{A}$ who controls the user but not the signer and executes the following experiment in order to get "one-more" signature (this is also called "one-more" forgery).

1. $(pk, sk) \leftarrow Gen(1^k)$
2. $\mathcal{A}(pk)$ engages in polynomially many (in k) adaptive, parallel and arbitrarily interleaved interactive protocols with polynomially many copies of $S(pk, sk)$, where $\mathcal{A}$ decides in an adaptive fashion when to stop. Let ℓ be the number of executions, where the Signer outputted "completed" in the end of Step 2.

3. $\mathcal{A}$ outputs a collection $\{(m_1, \sigma(m_1)), \ldots, (m_j, \sigma(m_j))\}$ subject to the constraint that $(m_i, \sigma(m_i))$ for $1 \leq i \leq j$ are all accepted by $Verify(pk, m_i, \sigma(m_i))$, and all m_i's are distinct.

Then the probability, taken over coin-flips of key-generation algorithm, the coin-flips of $\mathcal{A}$, and over the (private) coin-flips of the Signer, that $j > \ell$ is at most $\nu(k)$.

D DLP with Schnorr Prover

Brands argues that his blind signatures construction is secure under the existence of a Schnorr prover [9]. In this section we formally define DLP with access to a Schnorr's prover and show that this can be modeled as an intractability assumption. Thus, even if we give a perfect or an L- naive RO replay reduction access to the prover side of the Schnorr identification scheme (we will call him Schnorr prover from now on) and an adversary $\mathcal{A}$ that performs a one-more forgery attack on the Brands signature scheme, the reduction cannot solve the discrete logarithm problem unless the DLP is easy in this setting. Let us now define the discrete logarithm problem in the setting with a Schnorr prover.

Definition 14 (DLP with the Schnorr prover). *Let $\mathcal{A}^{(\cdot)}(\cdot)$ be an oracle Turing machine that takes as input a discrete logarithm instance (G, q, g, h) and has oracle access to the Schnorr prover $Schnorr(G, q, g, x)$ for $h = g^x$. (That is to say, $\mathcal{A}$ may act as the verifier in the Schnorr protocol, as many times as it wishes.) Upon termination, $\mathcal{A}$ outputs a value x'. We say that $\mathcal{A}$* solves *the discrete logarithm problem with the Schnorr prover if $x' = x$.*

Assumption 1 (Security of DLP with Schnorr) *For any probabilistic poly-time family of oracle Turing machines, $\mathcal{A}^{(\cdot)}(\cdot)$, there exists a negligible function $\nu(k)$ such that*

$$Pr[(G, q, g, g^x) \leftarrow S(1^k); x' \leftarrow \mathcal{A}^{Schnorr(g,x)}(g, g^x) : x' = x] = \nu(k).$$

where $S(1^k)$ samples Discrete Logarithm instances of size k.

DLP with Schnorr can easily be modeled as an intractability assumption similar to the standard DLP. It corresponds to threshold $t(k) = 0$ and a 2-round challenger C who on input 1^k picks a random x and sends g^x to the adversary $\mathcal{A}$ which works as defined in Definition 10. $\mathcal{A}$, having oracle access to the Schnorr prover responds with x' to the challenger. If $x' = x$ then C outputs 1.

E Modifying Brands

We will modify Brands' scheme in such a way that the Bank will use more that one secret key to compute her public key. We will be using the modular representation of e-cash systems provided by Cramer et al. in [16] to present our scheme and explain the building blocks besides e-cash systems.

Setup The setup is similar as the one of the commitment scheme. There exists a group G of prime order q and three generators $G_1, G_2, g_2 \in G_q$. No one in the system should be able to compute $\log_{G_1} G_2$. The Bank samples uniformly at random $w_1, w_2 \in \mathbb{Z}_q$ and computes

$$H = G_1^{w_1} G_2^{w_2}.$$

The values (H, G_1, G_2) are public, and used as the Bank's verification key while the values (w_1, w_2) are kept secret.

User Registration When a User opens an account at the Bank the following take place. The User picks $U \in \mathbb{Z}_q$ uniformly at random and secretly stores U which represents his identity. Then, he defines

$$g_1 = G_1^U G_2$$

and sends g_1 to the Bank. As a part of the registration, the User also needs to prove knowledge of $U \in \mathbb{Z}_q$ such that $g_1 = G_1^U G_2$. The Bank first verifies the proof of knowledge sent by the User, then picks $w_3 \in \mathbb{Z}_q$ uniformly at random and computes

$$h = g_1^{w_1} g_2^{w_3}$$

which sends to the User. The User stores (g_1, h) and U. The Bank records that this particular user registered under the public identifier g_1.

Withdrawal In each withdrawal, User and Bank use the instance

$$X = ((H, G_1, G_2), (h, g_1, g_2)).$$

The User and the Bank are going to run a blind signature scheme so that the user will receive a signed coin by the Bank. The withdrawal proceeds as follows;

1. The Bank picks $v_1, v_2, v_3 \in \mathbb{Z}_q$ uniformly at random and sends to the User:
$$V_1 = G_1^{v_1} G_2^{v_2}$$
$$V_2 = g_1^{v_1} g_2^{v_3}.$$
2. The User selects uniformly random r', z_1', z_2', z_3' and computes the simulated values:
$$V_1' = G_1^{z_1'} G_2^{z_2'} H^{-r'}$$
$$V_2' = g_1^{z_1'} g_2^{z_3'} h^{-r'}.$$
Alternative we could say that the User uses a NIZK simulator and samples a valid conversation $((H, G_1, G_2), (h, g_1, g_2), (V_1', V_2'), r', (z_1', z_2', z_3'))$.
3. Then, the User selects $s, k \in \mathbb{Z}_q$ uniformly at random and computes:
$$\tilde{h} = h^s g_2^k$$
$$\tilde{g_1} = g_1^s$$
$$\tilde{V_1} = V_1 V_1'$$
$$\tilde{V_2} = (V_2 V_2')^s.$$
Note that $\tilde{g_1} = G_1^{Us} G_2^s$, where we can write $W_1 = Us$ and $W_2 = s$ and the User is actually proving that he knows a witness (W_1, W_2) such that $\tilde{g_1} = G_1^{W_1} G_2^{W_2}$.
4. As a next step, User picks $b_1, b_2 \in \mathbb{Z}_q$ and computes
$$m = G_1^{b_1} G_2^{b_2}.$$
5. Now, the User is going to compute the hash:
$$\tilde{r} = \mathcal{H}(m, (H, G_1, G_2), (\tilde{h}, \tilde{g_1}, g_2), (\tilde{V_1}, \tilde{V_2}))$$
and sends to the Bank the blinded value $r = \tilde{r} - r'$.
6. Upon receiving r, the Bank computes:
$$z_i = rw_i + v_1, \quad i \in \{1, \ldots, 3\}$$
and forwards the z_i's to the User.
7. The User first checks:
$$V_1 H^r \stackrel{?}{=} G_1^{z_1} G_2^{z_2}$$
$$V_2 h^r \stackrel{?}{=} g_1^{z_1} g_2^{z_3}$$
and if the equations verify computes:
$$\tilde{z_1} = z_1 + z_1'$$
$$\tilde{z_2} = z_2 + z_2'$$
$$\tilde{z_3} = (z_3 + z_3')s + k\tilde{r}.$$
The blind signature on $(m, \tilde{g_1})$ is
$$\sigma_B(m, \tilde{g_1}) = ((H, G_1, G_2), (\tilde{h}, \tilde{g_1}, g_2), (\tilde{V_1}, \tilde{V_2}), (\tilde{z_1}, \tilde{z_2}, \tilde{z_3}))$$

.

In order for somebody to verify the signature, he needs to check:
$$\tilde{V_1} H^{\tilde{r}} \stackrel{?}{=} G_1^{\tilde{z_1}} G_2^{\tilde{z_2}}$$
$$\tilde{V_2} \tilde{h}^{\tilde{r}} \stackrel{?}{=} \tilde{g_1}^{\tilde{z_1}} g_2^{\tilde{z_3}}.$$

The registration phase and the withdrawal protocol can be found in Table 1.

Signer$((H, G_1, G_2), (w_1, w_2))$		Receiver(H, G_1, G_2)
		$U \in \mathbb{Z}_q$
		$g_1 = G_1^U G_2,\ \pi =$ proof of knowledge of U
	$\xleftarrow{g_1, \pi}$	
verify π		
$w_3 \in \mathbb{Z}_q$		
$h = g_1^{w_1} g_2^{w_3}$		
	$\xrightarrow{h}$	
$v_1, v_2.v_3 \in_R \mathbb{Z}_q$		
$V_1 = G_1^{v_1} G_2^{v_2}$		
$V_2 = g_1^{v_1} g_2^{v_3}$		
	$\xrightarrow{V_1, V_2}$	
		$r', z_1', z_2', z_3' \in_R \mathbb{Z}_q$
		$V_1' = G_1^{z_1'} G_2^{z_2'} H^{-r'}$
		$V_2' = g_1^{z_1'} g_2^{z_3'} h^{-r'}$
		$s, k \in_R \mathbb{Z}_q$
		$\tilde{h} = h^s g_2^k$
		$\tilde{g_1} = g_1^s$
		$\tilde{V_1} = V_1 V_1'$
		$\tilde{V_2} = (V_2 V_2')^s$
		$b_1, b_2 \in_R \mathbb{Z}_q$
		$m = G_1^{b_1} G_2^{b_2}$
		$\tilde{r} = \mathcal{H}(m, (H, G_1, G_2), (\tilde{h}, \tilde{g_1}, g_2), (\tilde{V_1}, \tilde{V_2}))$
		$r = \tilde{r} - r'$
	$\xrightarrow{r}$	
$z_i = r w_i + v_i,\ i \in \{1, \ldots, 3\}$		
	$\xrightarrow{z_1, z_2, z_3}$	
		$V_1 H^r \stackrel{?}{=} G_1^{z_1} G_2^{z_2}$
		$V_2 h^r \stackrel{?}{=} g_1^{z_1} g_2^{z_3}$
		$\tilde{z_1} = z_1 + z_1'$
		$\tilde{z_2} = z_2 + z_2'$
		$\tilde{z_3} = (z_3 + z_3')s + k\tilde{r}$

Table 1. Brands' blind signature modified

Spending Assume that the User wants to spent the coin in a Shop, with unique identity I_s. The protocol begins with the Shop sending to the user *pid* where:

$$pid = \mathcal{H}(I_s, date/\ time).$$

Then, the user is going to "sign" *pid* by sending to the bank:

$$p_1 = pid\ W_1 + b_1$$
$$p_2 = pid\ W_2 + b_2$$

plus the coin $(m, \tilde{g_1})$ and σ_B. The Shop verifies the validity of the coin and checks whether:

$$m\tilde{g_1}^{pid} \stackrel{?}{=} G_1^{p_1} G_2^{p_2}.$$

If both verifications hold, the Shop accepts the payment and stores $((m, \tilde{g_1}),\ \sigma_B, (p_1, p_2),\ pid)$.

Deposit When the Shop needs to deposit a coin in the Bank just needs to send the spending transcript $((m, \tilde{g_1}), \sigma_B, (p_1, p_2),\ pid)$ and the date/time of the transaction. The Bank needs to first check that *pid* encodes the identity of the Shop and the date/time in order to make sure that the Shop is not trying to deposit the same coin again. If something is wrong the bank doesn't accept the coin.

If everything was all right then the Bank needs to verify the signature on the coin and that (p_1, p_2) are valid (by checking whether $m\tilde{g_1}^{pid} \stackrel{?}{=} G_1^{p_1} G_2^{p_2}$. If so, then the bank stores $(m, \tilde{g_1}), \sigma_B, (p_1, p_2),\ sid)$ if something doesn't verify, the bank aborts.

In the case that a user is trying to double spend, then the Bank will receive the same $((m, \tilde{g_1}), \sigma_B)$ but with new $(p_1', p_2'), /sid'$. Then, the Bank is able to catch the User who is double spending by computing the following:

$$G_1^{\frac{p_1 - p_1'}{pid - pid'}} G_2^{\frac{p_2 - p_2'}{pid - pid'}}$$

which is equal to

$$G_1^{W_1} G_2^{W_2}$$

and by knowing W_1 and W_2 which are equal to $W_1 = Us$ and $W_2 = s$, the Bank can compute the secret key of the User who was double spending.

E.1 Unforgeability of Withdrawal Protocol

By modifying Brands e-cash scheme so that the bank uses more than secret keys to compute the public key, we can apply the technique proposed in [33] to prove the unforgeability of the modified withdrawal protocol. We will prove the following Theorem.

Theorem 3. *Consider the modified Brands' withdrawal/blind signature scheme in the random oracle model. If there exists a probabilistic polynomial time Turing machine which can perform a "one - more" forgery, with non-negligible probability, even under a parallel attack, then the discrete logarithm can be solved in polynomial time.*

Proof. We first describe an outline of the proof, then we will simplify the notations and finally we will complete the proof.
Outline of the proof Let $\mathcal{A}$ be the attacker who can be described as a probabilistic polynomial time Turing machine with random tape ω. Thus, there exists an integer ℓ such that after ℓ interactions with the bank $(v_{1,i}, v_{2,i}, r_i, z_{1,i}, z_{2,i}, z_{3,i})$ for $i \in \{1, \ldots, \ell\}$, and a polynomial number Q of queries asked to the random oracle, $\mathcal{Q}_1, \ldots, \mathcal{Q}_Q$, $\mathcal{A}$ returns $\ell + 1$ valid signatures (coins), $(m_i, \tilde{g}_{1,i}), \tilde{h}_i, (\tilde{V}_{1,i}, \tilde{V}_{2,i}), (\tilde{z}_{1,i}$, $\tilde{z}_{2,i}, \tilde{z}_{3,i}$, for $i = 1, \ldots, \ell + 1$ (to verify the signature (coin) you would first need to compute $\tilde{r}_i = \mathcal{H}(m_i, (H, G_1, G_2), (\tilde{h}_i, \tilde{g}_{1,i}, g_2), (\tilde{V}_{1,i}, \tilde{V}_{2,i}))$).

The bank possesses a secret key (w_1, w_2) associated to a public key $H = G_1^{w_1} G_2^{w_2}$, and a random tape Ω. The secret key is stored on the knowledge tape of the Turing machine.

Through a collision of the Bank and the attacker, we want to compute the discrete logarithm of G_1 relatively to G_2. The technique used is the one described in [34] as technique of "oracle replay". We first

run the attack with random keys, tapes and oracle f (which answers the hash queries). We randomly choose an index j and then replay with same keys and random tapes, but a different oracle f' such that the first $j-1$ answers remain unchanged. We expect that, with non-negligible probability, both executions output a common $\tilde{V}_1$ and $\tilde{V}_2$ coming from the j^{th} query having two distinct representations relatively to G_1 and G_2. Specifically, we expect to get the same $\tilde{V}_1$ for the two different sets $(\tilde{r}, \tilde{z}_1, \tilde{z}_2, \tilde{z}_3)$ and $(\hat{\tilde{r}}, \hat{\tilde{z}}_1, \hat{\tilde{z}}_2, \hat{\tilde{z}}_3)$, where $\hat{}$ denotes the second execution. We could also choose to work with $\tilde{V}_2$ but it wouldn't make any difference. So, we would have:

$$\tilde{V}_1 = H^{-\tilde{r}} G_1^{\tilde{z}_1} G_2^{\tilde{z}_2}$$
$$\tilde{V}_1 = H^{-\hat{\tilde{r}}} G_1^{\hat{\tilde{z}}_1} G_2^{\hat{\tilde{z}}_2}$$

and

$$\log_{G_1} G_2 = \frac{r'_1 - w_1 c' - \hat{r'_1} + w_1 \hat{c'}}{\hat{r'_2} - w_2 \hat{c'} - r'_2 + w_2 c'}$$

where the reduction knows the secret key of the Bank, (w_1, w_2).

Cleaning up Notations Before proceeding to the actual proof we will clean up some notation issues. Without loss of generality, we assume that all the $(m_i, (H, G_1, G_2), (\tilde{h}, \tilde{g}_{1,i}, g_2), (\tilde{V}_{1,i}, \tilde{V}_{2,i}))$ are queries which have been asked during the attack (otherwise, the probability of success would be negligible due to the randomness of the random oracle outputs). Then, we can assume that the indices, $(Ind_1, \ldots, Ind_{\ell+1})$, of $(m_1, (H, G_1, G_2), (\tilde{h}, \tilde{g}_{1,1}, g_2), (\tilde{V}_{1,1}, \tilde{V}_{2,1}), \ldots, (m_{\ell+1}, (H, G_1, G_2), (\tilde{h}, \tilde{g}_{1,\ell+1}, g_2), (\tilde{V}_{1,\ell+1}, \tilde{V}_{2,\ell+1}))$ in the list of queries is constant. As a result, the probability of success decreases from ε to $\rho \approx \varepsilon / Q^{\ell+1}$ (where Q the number of queries asked to the random oracle).

(w_1, w_2) is the secret key used by the Bank. The random tape of the Bank, Ω, determines the pairs $(v_{1,i}, v_{2,i}, v_{3,i})$ such that $V_{1,i} = G_1^{v_{1,i}} G_2^{v_{2,i}}$ and $V_{2,i} = g_1^{v_{1,i}} g_2^{v_{3,i}}$ for $i = 1, \ldots, \ell$. The distribution of (w_1, w_2, H) where w_1 and w_2 are random and $H = G_1^{w_1} G_2^{w_2}$, is the same as the distribution (w_1, w_2, H) where w_1 and H are random and w_2 is the unique element such that $H = G_1^{w_1} G_2^{w_2}$. Accordingly, we replace (w_1, w_2) by (w_1, H) and, similarly, each $(v_{1,i} v_{2,i})$ by $(v_{1,i}, V_{1,i})$ and $(v_{1,i} v_{3,i})$ by $(v_{1,i}, V_{2,i})$.

For the rest of the proof, we will group $(\omega, H, (V_{1,1}, V_{1,2}), \ldots, (V_{1,\ell}, V_{1,\ell}))$ under variable ν, and $(v_{1,i}, \ldots, (v_{1,\ell})$ under the variable τ. $\mathcal{S}$ will denote the set of all successful data, i.e. quadruples (ν, w_1, τ, f) such that the attack succeeds. Then,

$$\Pr_{\nu, x_1, \tau, f}[(\nu, w_1, \tau, f) \in \mathcal{S}] \geq \rho.$$

Before continuing with the proof we state a well - know probabilistic lemma:

Lemma 5. ***(The probabilistic lemma)****. Let A be a subset of $X \times Y$ such that $\Pr[A(x, y)] \geq \varepsilon$, then there exists $\Omega \subset X$ such that*

1. $\Pr[x \in \Omega] \geq \varepsilon/2$
2. *whenever $a \in \Omega$, $\Pr[A(a, y)] \geq \varepsilon/2$.*

The probabilistic lemma is useful to split a set X in two subsets, a non-negligible subset Ω consisting of "good" x's which provide a non-negligible probability of success over y, and its complement, consisting of "bad" x's.

Lemma 6. ***(The forking lemma.)*** *Randomly choose an index j, the keys and the random tapes. Run the attack twice with the same random tapes and two different random oracles, f and f', providing identical answers to the $j-1$ first queries. With non-negligible probability, the different outputs reveal two different representations of some $\tilde{V}_{1,i}$, relatively to G_1 and G_2.*

Proof. By proving this lemma we basically prove Theorem 3. What we want to show is, that after a replay, we can obtain a common $\tilde{V}_{1,i}$ such that:

$$\tilde{V}_{1,i} = G_1^{\tilde{z}_{1,i}} G_2^{\tilde{z}_{2,i}} H^{-\tilde{r}_i} = G_1^{\tilde{z}_{1,i} - w_1 \tilde{r}_i} G_2^{\tilde{z}_{2,i} - w_2 \tilde{r}_i}$$
$$= G_1^{\hat{\tilde{z}}_{1,i}} G_2^{\hat{\tilde{z}}_{2,i}} H^{-\hat{\tilde{r}}_i} = G_1^{\hat{\tilde{z}}_{1,i} - w_1 \hat{\tilde{r}}_i} G_2^{\hat{\tilde{z}}_{2,i} - w_2 \hat{\tilde{r}}_i}$$

where, $\tilde{z}_{1,i} - w_2\tilde{r_i} \neq \hat{\tilde{z}}_{1,i} - w_1\hat{\tilde{r}}_i$. We can remark that, for each i, $\tilde{V}_{1,i}$ only depends on (ν, w_1, τ) and the first $Ind_i - 1$ answers of f. What is left to study is whether or not the random variable $\chi_i = r'_{1i} - x_1 c'_i$ is sensitive to queries asked at steps Ind_i, $Ind_i + 1$, etc. We expect the answer to be yes. We can consider the most likely value taken by χ_i when (ν, w_1, τ) and the $Ind_i - 1$ first answers of f are fixed. Then, we are led to consider a function $e_i(\nu, w_1, \tau, f_i)$, where f_i ranges over the set of answers to the first $Ind_i - 1$ possible queries. Set

$$\lambda_i(\nu, w_1, \tau, f_i, e) = \Pr_f\left[(\chi_i(\nu, w_1, \tau, f) = e) \ \&\ ((\nu, w_1, \tau, f) \in \mathcal{S}) \,|\, f \text{ extends } f_i\right].$$

We define $e_i(\nu, w_1, \tau, f_i)$ as any value e such that $\lambda_i(\nu, w_1, \tau, f_i, r)$ is maximal. We then define the "good" subset $\mathcal{G}$ of $\mathcal{S}$ whose elements satisfy, for all i, $\chi_i(\nu, w_1, \tau, f) = e_i(\nu, w_1, \tau, f_i)$, where f_i denotes the restriction of f to queries of index strictly less than Ind_i, and the "bad" $\mathcal{B}$ its compliments in $\mathcal{S}$.

Definition 15. *We denote by Φ the transformation which maps any quadruple (ν, w_1, τ, f) to $(\nu, w_1 + 1, \tau - c, f)$, where $\tau - r = (v_{1,1} - r_1, \ldots, v_{1,\ell} - r_\ell)$.*

Lemma 7. *Both executions corresponding to (ν, w_1, τ, f) and $\Phi(\nu, w_1, \tau, f)$ are totally identical with respect to the view of the attacker. Especially, outputs are the same.*

Proof. Let (ν, w_1, τ, f) be an input for the collusion. Replay with $\hat{w_1} = w_1 + 1$ and $\hat{\tau} = \tau - r$, the same ν and the same oracle f. The answers of the oracle are unchanged and the interactions with the bank become

$$\hat{z_{1,i}}(\hat{w_1}, \hat{v}_{1,i}, r_i) = \hat{v}_{1,i} + \hat{w_1} r_i = (v_{1,i} - r_i) + r_i(w_1 + 1) = v_{1,i} + r_i w_1 = z_{1,i}(w_1, v_{1,i}, r_i).$$

Thus, everything remains the same.

Corollary 2. *Φ is a one-to-one mapping from $\mathcal{S}$ onto $\mathcal{S}$.*

Lemma 8. *For fixed (ν, w_1, τ), the probability*

$$\Pr_f[((\nu, w_1, \tau, f) \in \mathcal{G}) \ \&\ (\Phi(\nu, w_1, \tau, f) \in \mathcal{G}) \leq 1/q.$$

Which means that Φ sends the set $\mathcal{G}$ into $\mathcal{B}$, except for a negligible part.

Proof. We will prove the above lemma by contradiction. Assume that $\Pr_f[(\nu, w_1, \tau, f) \in \bigcup_{r_1,\ldots,r_\ell} Y(r_1, \ldots, r_\ell)] > 1/q$, where the set $Y(r_1, \ldots, r_\ell)$ is defined by the conditions $(\nu, w_1, \tau, f) \in \mathcal{G}$, $\Phi(\nu, w_1, \tau, f) \in \mathcal{G}$ and $(r_1, \ldots, r_\ell)$ are the successive questions asked to the authority. Then, there exists a ℓ-tuple $(r_1, \ldots, r_\ell)$ such that $\Pr_f[Y(r_1, \ldots, r_\ell)] > 1/(q^{\ell+1})$. Thus, there exist two oracles f and f' in $Y(r_1, \ldots, r_\ell)$ which provide distinct answers for some queries $\mathcal{Q}_{Ind_j} = (m_j, (H, G_1, G_2), (\tilde{h}_j, \tilde{g}_{1,j}, g_2), (\tilde{V}_{1,j}, \tilde{V}_{2,j}))$ to the oracle, for some $j \in 1, \ldots, \ell + 1$, and are such that answers to queries not of the form of $\mathcal{Q}_{Ind_j}$ are similar. We will denote by i the smallest such index j. Then $f_i = f'_i$ and $\tilde{r}_i \neq \hat{\tilde{r}}_i$. Also, we have $(\nu, w_1, \tau, f) \in \mathcal{G}$, $\Phi(\nu, w_1, \tau, f) \in \mathcal{G}$ and similarly $(\nu, w_1, \tau, f') \in \mathcal{G}$, $\Phi(\nu, w_1, \tau, f') \in \mathcal{G}$. Because of the property of Φ (see lemma 3), and by definition of $\mathcal{G}$,

$$\begin{aligned}
e_i(\nu, w_1, \tau, f_i) &= z_{1,i}(\nu, w_1, \tau, f) - w_1\tilde{r}_i \\
&= z_{1,i}(\Phi(\nu, w_1, \tau, f)) - w_1\tilde{r}_i \\
&= e_i(\nu, w_1 + 1, \tau - r, f_i) + ((w_1 + 1) - w_1)\tilde{r}_i \\
e_i(\nu, w_1, \tau, f'_i) &= z_{1,i}(\nu, w_1, \tau, f') - xw1\,\hat{\tilde{r}}_i \\
&= z_{1,i}(\Phi(\nu, w_1, \tau, f')) - w_1\,\hat{\tilde{r}}_i \\
&= e_i(\nu, w_1 + 1, \tau - \hat{r}, f'_i) + ((w_1 + 1) - w_1)\,\hat{\tilde{r}}_i
\end{aligned}$$

The equality $f_i = f'_i$ implies $e_i(\nu, w_1, \tau, f_i) = e_i(\nu, w_1, \tau, f'_i)$. Since we have assumed $(r_1, \ldots, r_\ell) = (\hat{r_1}, \ldots, \hat{r_\ell})$, then $e_i(\nu, w_1 + 1, \tau - r, f_i) = e_i(\nu, w_1 + 1, \tau - \hat{r}, f'_i)$. Thus, $\tilde{r}_i = \hat{\tilde{r}}_i$ which contradicts the hypothesis.

Lemma 4 says that for any (ν, w_1, τ),

$$\Pr_f[((\nu, w_1, \tau, f) \in \mathcal{G}) \ \& \ (\Phi(\nu, w_1, \tau, f) \in \mathcal{G})] \leq 1/q.$$

By making the sum over all the triplets (ν, w_1, τ), and using the bijectivity of Φ (corollary 1), we obtain

$$\begin{aligned}\Pr[\mathcal{G}] &= \Pr_{\nu,w_1,\tau,f}[((\nu, w_1, \tau, f) \in \mathcal{G}) \ \& \ (\Phi(\nu, w_1, \tau, f) \in \mathcal{G})] \\ &\quad + \Pr_{\nu,w_1,\tau,f}[((\nu, w_1, \tau, f) \in \mathcal{G}) \ \& \ (\Phi(\nu, w_1, \tau, f) \in \mathcal{B})] \\ &\leq \frac{1}{q} + \Pr_{\nu,w_1,\tau,f}[\Phi(\nu, w_1, \tau, f) \in \mathcal{B}] \leq \frac{1}{q} + \Pr[\mathcal{B}]\end{aligned}$$

Then, $\Pr[\mathcal{B}] \geq (\Pr[\mathcal{S}] - 1/q)/2$. Since $1/q$ is negligible w.r.t $\Pr[\mathcal{S}]$, for enough large keys, we have, $\Pr[\mathcal{B}] \geq \Pr[\mathcal{S}]/3 \geq \rho/3$.

Conclusion We will use this probability to show the success of forking.

$$\begin{aligned}\frac{\rho}{3} \leq \Pr[\mathcal{B}] &= \Pr_{\nu,w_1,\tau,f}[\mathcal{S} \ \& \ ((\exists i)\chi_i(\nu, w_1, \tau, f) \neq e_i(\nu, w_1, \tau, f_i))] \\ &\leq \sum_{i=1}^{\ell+1} \Pr_{\nu,w_1,\tau,f}[\mathcal{S} \ \& \ (\chi_i(\nu, w_1, \tau, f) \neq e_i(\nu, w_1, \tau, f_i))].\end{aligned}$$

There exists k such that $\Pr[\mathcal{S} \ \& \ (\chi_k(\nu, w_1, \tau, f) \neq e_i(\nu, w_1, \tau, f_k))] \geq \rho/3\ell + 1$. Let us randomly choose the forking index i. With probability greater than $1/(\ell + 1)$, we have guessed $i = k$. The probabilistic lemma ensures that there exists a set X such that

1. $\Pr_{\nu,w_1,\tau,f}[(\nu, w_1, \tau, f_i) \in X] \geq \rho/6(\ell + 1)$
2. for all $(\nu, w_1, \tau, f_i) \in X$, $\Pr_f[(\nu, w_1, \tau, f) \in \mathcal{S} \ \& \ (\chi_i \neq e_i)|$ extends $f_i] \geq \rho/6(\ell + 1)$.

Let us choose a random quadruple (ν, w_1, τ, f). With probability greater than $(\rho/6(\ell+1))^2$, $(\nu, w_1, \tau, f) \in \mathcal{S}$, $(\nu, w_1, \tau, f_i) \in X$ and $\chi_i(\nu, w_1, \tau, f) \neq e_i(\nu, w_1, \tau, f_i)$. We will denote by α the value $\chi_i(\nu, w_1, \tau, f)$ and by β the value $e_i(\nu, w_1, \tau, f_i)$. Then, two cases appear relatively to $\lambda_i(\nu, w_1, \tau, f_i, \alpha)$:

- if $\lambda_i(\nu, w_1, \tau, f_i, \alpha) \geq \rho/12(\ell+1)$, then, by definition of e_i, we know that $\lambda_i(\nu, w_1, \tau, f_i, \beta) \leq \rho/12(\ell+1)$.
- otherwise,

$$\begin{aligned}&\lambda_i(\nu, w_1, \tau, f_i, \alpha) + \Pr_{f'}[\mathcal{S} \ \& \ (\chi_i(\nu, w_1, \tau, f_i, \alpha') \neq \alpha)|f' \text{ extends } f_i] \\ &= \Pr_{f'}[S|f' \text{ extends } f_i] \\ &\geq \Pr_{f'}[\mathcal{S} \ \& \ (\chi_i(\nu, w_1, \tau, f') \neq \beta)|f' \text{ extends } f_i] \geq \rho/6(\ell + 1).\end{aligned}$$

Both cases lead to $\Pr_{f'}[\mathcal{S} \ \& \ (\chi_i(\nu, w_1, \tau, f') \neq \alpha)|f' \text{ extends } f_i] \geq \rho/12(\ell+1)$. Thus, if we replay with the same keys and random tapes but another random oracle f' such that $f'_i = f_i$, we obtain, with probability at least $\rho/12(\ell + 1)$, a new success with $\chi_i(\nu, w_1, \tau, f') \neq \alpha$. Then, both executions provide two different representations of a_i with respect to G_1 and G_2.

Global Complexity of the Reduction By using a replay oracle technique with a random forking index, the probability of success is greater than

$$\frac{1}{\ell + 1} \times \left(\frac{\rho}{6(\ell + 1)}\right)^2 \times \frac{\rho}{12(\ell + 1)} \times \left(\frac{1}{6(\ell + 1)} \times \frac{\varepsilon}{\mathcal{Q}^{\ell+1}}\right)^3$$

where ε is the probability of success of an $\ell, \ell + 1$-forgery and Q the number of queries asked to the random oracle.