

On the Security Loss of Unique Signatures

Andrew Morgan
Cornell University
asmorgan@cs.cornell.edu

Rafael Pass*
Cornell Tech
rafael@cornell.edu

September 21, 2018

Abstract

We consider the question of whether the security of unique digital signature schemes can be based on game-based cryptographic assumptions using *linear-preserving* black-box security reductions—that is, black-box reductions for which the security loss (i.e., the ratio between “work” of the adversary and the “work” of the reduction) is some *a priori* bounded polynomial. A seminal result by Coron (Eurocrypt’02) shows limitations of such reductions; however, his impossibility result and its subsequent extensions all suffer from two notable restrictions: (1) they only rule out so-called “simple” reductions, where the reduction is restricted to only *sequentially* invoke “straight-line” instances of the adversary; and (2) they only rule out reductions to non-interactive (two-round) assumptions.

In this work, we present the first full impossibility result: our main result shows that the existence of *any* linear-preserving black-box reduction for basing the security of unique signatures on some *bounded-round* assumption implies that the assumption can be broken in polynomial time.

*Supported in part by NSF Award CNS-1561209, NSF Award CNS-1217821, AFOSR Award FA9550-15-1-0262, a Microsoft Faculty Fellowship, and a Google Faculty Research Award.

1 Introduction

Digital signature schemes, whereby a party can “sign” a message in a publicly verifiable yet still adversarially unforgeable way, are one of the most basic and important classes of cryptographic primitives; their security has been studied since the 1970s. While the earliest constructions of digital signatures [RSA78, Rab79, Sha78, DH06] were heuristic in nature, modern constructions have tight proofs of security against all computationally bounded adversaries based on certain underlying assumptions.

Specifically, in a provably secure construction, we have a reduction \mathcal{R} , which, given any adversary \mathcal{A} which breaks the security of a digital signature scheme Π , can break a certain underlying assumption \mathcal{C} ; hence, if the assumption holds, then the scheme must be secure. In this paper, we restrict our attention to *black-box* security reductions, where \mathcal{R} only interacts with \mathcal{A} as a “black box”.¹ As far as we know, all security proofs for digital signatures rely on black-box security reductions.

We are interested in the *security loss* of such a reduction (a concept originally proposed as “security preservation” in [Lub96]), or intuitively how “inefficient” it is in terms of running time and success probability compared to the adversary it runs. Informally, if, given a security parameter n , \mathcal{R} (including the instances of \mathcal{A} it runs) and \mathcal{A} run in time $\text{Time}_{\mathcal{R}^{\mathcal{A}}}(n)$ and $\text{Time}_{\mathcal{A}}(n)$ respectively, and have success probabilities of $\text{Success}_{\mathcal{R}^{\mathcal{A}}}(n)$ and $\text{Success}_{\mathcal{A}}(n)$, then the security loss is given by the maximum over all adversaries \mathcal{A} of:

$$\lambda_{\mathcal{R}}(n) = \frac{\text{Success}_{\mathcal{A}}(n) \text{Time}_{\mathcal{R}^{\mathcal{A}}}(n)}{\text{Success}_{\mathcal{R}^{\mathcal{A}}}(n) \text{Time}_{\mathcal{A}}(n)}$$

Intuitively, if we define the “work factor” of the adversary \mathcal{A} to be the ratio of its running time to its success probability, or $\text{Work}_{\mathcal{A}}(n) = \frac{\text{Time}_{\mathcal{A}}(n)}{\text{Success}_{\mathcal{A}}(n)}$ (and respectively for $\mathcal{R}^{\mathcal{A}}$), then we can think of the security loss as

$$\lambda_{\mathcal{R}} = \frac{\text{Work}_{\mathcal{R}^{\mathcal{A}}}(n)}{\text{Work}_{\mathcal{A}}(n)}$$

or how much “work” the reduction \mathcal{R} needs to do to break the underlying assumption compared to the amount of work that its adversary \mathcal{A} does to break the primitive Π . So the higher the security loss, the easier the primitive is to break compared to the underlying assumption. As such, having security reductions with low security loss is essential for proving practical security of cryptographic primitives, since the security loss has a significant effect on the security parameter (i.e., the bit length of a key, size of a large prime for RSA, etc.) which must be used for the underlying assumption to achieve a particular level of security for the primitive.

The most efficient possible reductions are those which have *constant* security loss $\lambda_{\mathcal{R}}(n) \leq c$; these are commonly called *tight* reductions [BR96]. These reductions prove that $\text{Work}_{\mathcal{A}}(n)$ is always directly proportional to $\text{Work}_{\mathcal{R}^{\mathcal{A}}}(n)$, and so increasing the security parameter will always have the same effect on the security of the primitive as on the security of the underlying assumption.

A weaker notion of efficiency—introduced by Luby [Lub96]—is that of a *linear-preserving* reduction, where the security loss is required to be bounded by some *a priori* fixed polynomial $p(\cdot)$ in just the security parameter; that is, $\lambda_{\mathcal{R}}(n) \leq p(n)$.

For instance, a security reduction that only runs the adversary \mathcal{A} a fixed polynomial number of times (independent of \mathcal{A} ’s running time and success probability) may not be tight, but is still

¹We note that Π need not be black-box itself in some underlying primitive; we only require the *reduction* \mathcal{R} to be black-box.

linear-preserving.² While, with a linear-preserving reduction, the concrete security of the primitive Π is only comparable to that of the assumption if we use an increased security parameter for Π , Π still retains the same “asymptotic” security as the underlying assumption: for instance, if Π can be broken in time $\text{poly}(n) \cdot (2^{n/3})$, then so can the underlying assumption.

Unique Signatures. While the original provably secure construction of digital signatures in [GMR88] was neither tight nor linear-preserving, more recent constructions [BR96, Ber08, AFLT12, HJ12, CW13, CGW17] with tight reductions have been exhibited. However, while these modern constructions are quite efficient, they sacrifice some arguably important features of the original constructions in achieving this. Most notably, the earliest construction in [RSA78] had the property that signatures were *unique*—that is, for every public key and every message, there exists at most one valid signature for that message. Whereas provably secure constructions of unique signatures exist [MVR99, Lys02], as well as constructions of verifiable random functions [MVR99, BGRV09, Jag15] (which [MVR99] shows imply unique signatures), none of these have linear-preserving, let alone tight, security reductions. And unfortunately, for many recent applications of digital signatures (e.g., the recent applications to blockchains [GHM⁺17, PS17]), this uniqueness property is in fact necessary.

Can Unique Signatures Have Linear-Preserving Reductions? A natural question, given the fact that no linear-preserving reductions have been discovered, is whether a certain degree of security loss is *required* when proving the security of unique signatures. This question was first addressed in 2002 by Jean-Sébastien Coron in his seminal paper [Cor02]. At a high level, Coron’s goal was to demonstrate that any unique signature scheme with a black-box security reduction must have a security loss of $O(\ell(n))$, where $\ell(n)$ is the number of signing queries made by the adversary. This, in particular, would rule out all linear-preserving reductions for unique signature schemes because $\ell(n)$ depends on the specific adversary \mathcal{A} and can be an arbitrarily large polynomial.

However, while Coron’s proof rules out many “natural” reductions, it does not fully answer the question. In particular, it applies only to a quite restricted class of “simple” reductions which run the adversary in a “sequential straight-line” fashion—that is, they can run many instances of the adversary, but must run them sequentially (such that each must finish before the next starts) and cannot rewind the adversary. Furthermore, Coron’s result applies only to reductions to the class of *non-interactive* (i.e., two-round) security assumptions (e.g., inverting a one-way function or breaking RSA). This latter restriction is necessary to some extent; if the security assumption may have arbitrarily many rounds, then it becomes trivial to base security on such an assumption (e.g., reducing the security of digital signatures to itself). However, there is still a large gap between non-interactive and “unbounded-round” security assumptions, leaving open the question of whether *bounded-round* assumptions [Pas11] (that is, security assumptions modeled as security games with an *a priori* bounded number of communication rounds) can be used to prove the security of unique signatures.

Since Coron’s seminal work, his result has been generalized to a number of related primitives [HJK12, BJLS16], improved and simplified [BJLS16, KK12], and extended to rule out other notions of security tightness [WMHT18]. However, despite these extensions, improvements, and

²The name “linear-preserving” comes from the fact that $\text{Work}_{\mathcal{R}, \mathcal{A}}$ is still linear in the quantity $\text{Work}_{\mathcal{A}}(n)$, although it may depend polynomially on the security parameter n .

generalizations, the above restrictions—to simple reductions, and to non-interactive assumptions—have not yet been surmounted, leaving open the question:

Does there exist a linear-preserving security reduction for basing the security of unique digital signatures on some natural hardness assumption?

Main Theorem. In this work, we settle this question, ruling out *all* linear-preserving reductions from unique signatures to *any* bounded-round assumption.

Theorem 1. (*Informal.*) There does not exist a linear-preserving black-box reduction from the security of some unique signature scheme Π to a bounded-round intractability assumption \mathcal{C} , unless \mathcal{C} can be broken in polynomial time.

More precisely, we show that, unless \mathcal{C} can be broken in polynomial time, the security loss of any black-box reduction from the security of some unique signature scheme Π to any bounded-round intractability assumption \mathcal{C} must be at least $O(\sqrt{\ell(n)})$, where $\ell(n)$ is the number of signature queries the adversary uses (and thus not a fixed polynomial independent of the adversary). Moreover, we observe (in Appendix C) that our main theorem, with minor alterations, can also be applied to the related notion of *rerandomizable* signatures, or non-unique signatures with the property that signatures can be efficiently “rerandomized”.

1.1 Proof Outline

In proving our main theorem, we follow the “meta-reduction” paradigm, originally pioneered in [BV98] (see also [BMV08, HRS09, FS10, AGO11, BBF13, BFW16] for related work concerning meta-reductions), though we note that work on black-box separations using other frameworks dates back much farther, to [IR89]. The core idea behind this approach is, given an arbitrary black-box reduction \mathcal{R} that breaks the assumption \mathcal{C} by using black-box access to some “ideal adversary” \mathcal{A} (which itself breaks the security of the primitive Π), to create an efficient adversary \mathcal{B} which breaks \mathcal{C} *without using \mathcal{A}* : \mathcal{B} will internally run \mathcal{R} and, roughly speaking, internally (and efficiently) “emulate” \mathcal{A} for \mathcal{R} . The implication then is that if such a reduction \mathcal{R} exists and breaks \mathcal{C} using the inefficient adversary \mathcal{A} , then \mathcal{B} would likewise break \mathcal{C} , but in polynomial time, proving that \mathcal{C} is not a secure assumption. See Figure 1 for an illustration of the mechanics of this paradigm.

Of course, we cannot prove complete non-existence (since, indeed, provably secure unique signature schemes exist); instead, we show that, unless \mathcal{R} makes many queries to \mathcal{A} —which already implies that the security loss is high—then we can efficiently emulate \mathcal{A} ’s responses with “high” (but not overwhelming) probability, which in turn will imply that \mathcal{B} breaks \mathcal{C} unless \mathcal{R} ’s success probability is relatively small (again implying that its security loss is high).

Coron’s Meta-Reduction. As already mentioned, Coron, in [Cor02], demonstrates how to employ this technique for a *restricted* class of reductions from the security of unique signatures. In particular, imagine an “ideal” inefficient adversary \mathcal{A} which requests signatures for $\ell(n)$ randomly chosen messages, next uses a *brute-force search* to find a signature (i.e., a forgery) on a new random message, and finally returns the forgery. (Note that this adversary \mathcal{A} is inefficient as it requires a brute-force search to recover the forgery.) In order to simulate \mathcal{R} ’s interaction with \mathcal{A} while running in polynomial time, \mathcal{B} will run \mathcal{R} normally and simulate \mathcal{A} by first requesting signatures for $\ell(n)$ random messages. However, in order to extract a forgery without using brute force, \mathcal{B} will

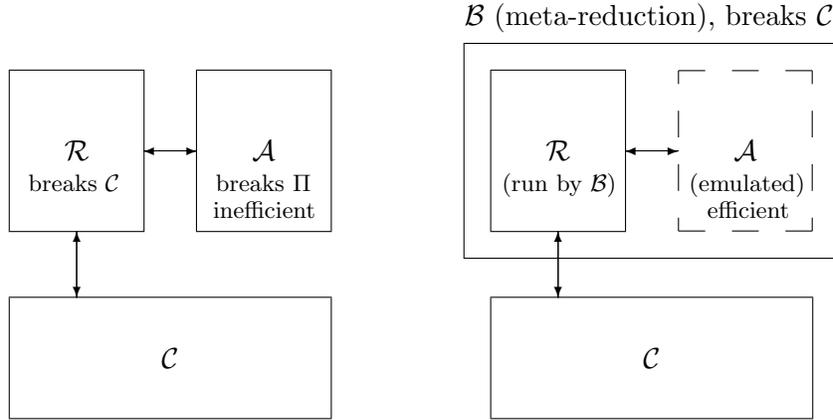


Figure 1: The meta-reduction paradigm. *Left:* \mathcal{R} breaks the assumption \mathcal{C} by using the “ideal” but inefficient adversary \mathcal{A} (against the signature scheme Π) as an oracle. *Right:* the meta-reduction \mathcal{B} runs \mathcal{R} (forwarding its communication with \mathcal{C}) and efficiently emulates \mathcal{A} to break \mathcal{C} with slightly less probability than in the left experiment.

pick a random message m^* , rewind the execution of \mathcal{R} to before a randomly selected query, and try querying \mathcal{R} for m^* instead of what it sent to \mathcal{R} during the “main” (i.e., non-rewound) thread. If \mathcal{R} returns a correct signature for m^* during this “rewinding”, then \mathcal{B} has succeeded in efficiently extracting a forgery and can return it to \mathcal{R} . In this case, \mathcal{B} has succeeded in *perfectly* emulating \mathcal{A} ; note that this relies on the fact that the signature scheme is unique and thus there exists at most one valid signature on m^* .

Of course, \mathcal{R} may not always return a correct response to \mathcal{A} ’s (or \mathcal{B} ’s) queries; however, if \mathcal{A} receives any incorrect responses in the “main” thread, it may simply return \perp to \mathcal{R} (as the security game for unique signatures only dictates that \mathcal{A} must return a valid forgery when its queries are correctly answered), and so, in that case, \mathcal{B} may also do so when emulating \mathcal{A} . The only time that \mathcal{B} will fail to emulate \mathcal{A} is, in fact, when \mathcal{R} responds correctly to the original $\ell(n)$ queries by \mathcal{B} during the “main” execution, but fails to respond to the rewind query of m^* in *any* rewinding (and so \mathcal{B} can neither return \perp or a forgery). Coron, through an elegant (yet complex) probabilistic argument, shows that the probability of this “bad event” is bounded above by $O(1)/\ell(n)$. Intuitively, the reason this holds is that, unless \mathcal{R} provides signatures to a fraction $O(1)/\ell(n)$ of random messages m^* (and thus the rewinding succeeds with probability $O(1)/\ell(n)$), it is unlikely that \mathcal{R} provides correct signatures to all the $\ell(n)$ signature requests on the “main” thread, and in this case \mathcal{B} does not need to provide a forgery to succeed in emulating \mathcal{A} . Of course, formalizing this argument is quite non-trivial, and Coron presents a sophisticated analysis which does so.

This argument rules out all reductions \mathcal{R} from the security of unique signatures to a non-interactive security assumption which break the assumption with probability greater than the failure probability of \mathcal{B} —that is, $O(1)/\ell(n)$ —assuming \mathcal{R} runs a single instance of its adversary. If \mathcal{R} runs multiple, $M(n)$, instances of its adversary in a sequential (i.e., non-interleaved) manner, then by the union bound over all instances, the failure probability bound becomes $O(M(n))/\ell(n)$. Furthermore, in this case we know that $\text{Time}_{\mathcal{R}\mathcal{A}}(n) \geq M(n)\text{Time}_{\mathcal{A}}(n)$, and so (given $\text{Success}_{\mathcal{A}}(n) = 1$) Coron’s argument achieves a bound of

$$\lambda_{\mathcal{R}}(n) \geq O(\ell(n))$$

[DNS98]. This connection was already noted in the earlier impossibility result for black-box reductions of [Pas11], where a “recursive rewinding strategy” (similar to [RK99, CLP10, DGS09, PV08]) is used to overcome this problem. The core idea behind this technique is to rewind *every* relevant query, but to “abort” the rewinding when “too many” nested queries take place during the rewinding. The limit on nested queries furthermore decreases by a factor of n with each recursive level of nesting, so, since the total number of messages is polynomial, the number of levels will always be bounded by a constant, providing the polynomial bound on running time.

One might be tempted to simply apply this technique directly to the problem at hand; unfortunately, due to a fundamental difference between the two results, this is not possible. Specifically, in [Pas11], the result proven is a complete impossibility (and not just a bound on the security loss): more precisely, emulation of \mathcal{A} can be shown to succeed with overwhelming probability. In our context, however, the probability that the emulation succeeds is some inverse polynomial (and inherently must be so, or else we would have shown a complete impossibility of black-box security reductions). The problem then with a recursive rewinding strategy is that the failure probabilities may “cascade”. Additionally, we cannot rely on the technique from [Pas11] of repeatedly rewinding until we obtain a correct response, since that would bias the distribution of the message m^* on which we output a forgery!

A Simple Rewinding Technique. We deal with the problem by using a different (and actually much simpler) rewinding strategy, inspired by a technique for “bounded-concurrent” zero-knowledge arguments originally introduced by Lindell [Lin03]. The key observation is that rewinding is only necessary when \mathcal{B} encounters an “end message” (i.e., \mathcal{R} requesting a forgery from an instance of \mathcal{A}). If, during the rewinding to extract a forgery for some instance of \mathcal{A} , \mathcal{B} avoids queries that contain an end message for a separate instance of \mathcal{A} (i.e., those that would cause recursive rewinding), then it becomes straightforward to bound the number of rewindings and show that \mathcal{B} will run in polynomial time; as an added advantage, this allows us to treat end messages very similarly to external communication in the analysis of our meta-reduction.

However, while this simulation strategy at first glance may seem fairly straightforward (and, indeed, in the context of bounded-concurrent zero-knowledge, the analysis is simple), our scenario presents multiple major differences that make it quite non-trivial. In particular (as already mentioned above), unlike for zero-knowledge, we can no longer rewind queries with arbitrary messages; instead, in order to generate a forgery of a uniformly random message, we must choose a single forgery target m^* and rewind every query *only once* using this *same* message m^* (otherwise, as mentioned above, we would bias the distribution of the message m^* on which we output a forgery). Thus, to not bias the distribution of m^* , we must rewind each query with the same message m^* and consequently, we no longer have any independence between the rewindings, which severely complicates the analysis. (Indeed, recall that even in the simplified setting of Coron, his argument is already quite non-trivial.)

Towards dealing with this, we present a new way of analyzing an ideal adversary which is quite similar to Coron’s ideal adversary. Our analysis relies on a “randomness-switching” argument similar in spirit to that of [PRS02, PTV14], where we demonstrate that any “bad” sequence of randomness which causes the meta-reduction to fail can be permuted into many “good” sequences for which the meta-reduction succeeds. In particular, recall from our above discussion of Coron’s meta-reduction that, if a sequence of messages is such that \mathcal{B} fails to emulate \mathcal{A} , then *all* of its rewindings with the forgery target m^* must fail to extract a signature for the query to m^* . Hence,

every rewind sequence beginning with a prefix $(m_1, \dots, m_{i-1}, m^*)$ will either contain nesting or external communication during the query for m^* , or is such that \mathcal{R} provides an incorrect signature for m^* . In the latter case, we can conclude as above that \mathcal{A} and \mathcal{B} , having received an invalid signature, will both accordingly return \perp (meaning that in fact \mathcal{B} will succeed) if any sequence with that prefix is given in the (non-rewind) execution; thus, any such sequence cannot itself be a “bad” sequence. This, combined with the fact that the amount of nesting and external communication (and hence the possible number of rewind sequences which do *not* fall into the latter category) is bounded, will allow us to derive an upper bound for the possible number of “bad” sequences of randomness and hence for the probability of one of these sequences occurring (and causing the meta-reduction \mathcal{B} to fail). This failure probability bound for \mathcal{B} will ultimately allow us to upper-bound the success probability, and hence lower-bound the security loss, of the reduction \mathcal{R} . (In Appendix A, as an independent contribution and a warm-up, we sketch how an even simpler randomness-switching approach can be used to provide a simplified proof of a generalization of Coron’s theorem to arbitrary reductions with *static* scheduling—where the order in which the reductions sends its sends messages is a-priori fixed).

Overview. In Section 2 we present key notation and definitions to be employed in our proof. We present and discuss our main result in Section 3, construct our “ideal” adversary \mathcal{A} in Section 3.1, construct the meta-reduction \mathcal{B} in Section 3.2, and complete our analysis and proof of the main theorem in Section 3.3. Lastly, we provide a synopsis of related work in Section 4.

2 Preliminaries and Definitions

2.1 Notation

Let \mathbb{N} denote the set of natural numbers (positive integers), and let $[n]$ denote the set of natural numbers at most n , or $\{1, 2, \dots, n\}$. For $n \in \mathbb{N}$, we denote by 1^n the string of n ones, which will be used to provide a security parameter as input to an algorithm (this is by convention, so that the input length is bounded below by the security parameter). Given a set $S = \{s_1, \dots, s_n\}$ of distinct elements, we shall let $|S|$ denote the number of elements n in S , and we refer to the set $\Pi_n(S)$ as the set of *permutations* of S , which contains any sequence which itself contains, in any order, each element of S exactly once.

When we say that a statement holds “for all sufficiently large $n \in \mathbb{N}$ ”, by this we indicate that there exists an $N \in \mathbb{N}$ such that, for any integer $n \geq N$, the statement holds for n .

We recall that a function $\epsilon(\cdot)$ is *negligible* if, for any polynomial $p(\cdot)$, $\epsilon(n) < 1/p(n)$ for all sufficiently large $n \in \mathbb{N}$ —that is, if $\epsilon(\cdot)$ is asymptotically smaller than any inverse polynomial. (For instance, an inverse exponential such as e^{-cn} is negligible in n for any constant $c > 0$.)

Lastly, we assume a basic level of familiarity with the concepts of probabilistic algorithms and interactive Turing machines [GMR85]. We will let $\mathcal{R}^{\mathcal{A}}(x)$ denote the probability distribution over the output of an oracle algorithm \mathcal{R} given oracle access to a probabilistic \mathcal{A} . If \mathcal{A} is a (deterministic) interactive algorithm, we instead assume \mathcal{R} has oracle access to the function that, given the current partial transcript of (i.e., all messages sent up to a certain point in) interaction between \mathcal{R} and \mathcal{A} , returns \mathcal{A} ’s next message to \mathcal{R} .

Furthermore, we shall refer by $\langle \mathcal{A}, \mathcal{C} \rangle(x)$ to the probability distribution over the output of \mathcal{C} after interaction between probabilistic interactive Turing machines \mathcal{A} and \mathcal{C} , both given common

input x (where the common input is also provided to any oracles, e.g., to \mathcal{O} if \mathcal{A} is an oracle machine given by $\mathcal{A}^{\mathcal{O}}$); the view of the respective experiment, or the transcript of all messages sent and all randomness consumed, shall be denoted as $[\mathcal{A} \leftrightarrow \mathcal{C}](x)$.

2.2 Unique Signatures

First, we define unique signature schemes. Recall that a signature scheme is a means by which a message can be signed with the signer's secret key and the signature can be verified using a public key. A unique signature scheme, then, is simply a signature scheme for which each message can only have one possible signature:

Definition 1. A **unique signature scheme** is a triple $(\text{Gen}, \text{Sign}, \text{Ver})$ of probabilistic polynomial-time algorithms such that, for every $n \in \mathbb{N}$:

- **Gen**, on input 1^n , produces a pair (pk, sk)
- **Sign**, on input (sk, m) for any $m \in \{0, 1\}^n$, produces a signature σ . (We write $\sigma \leftarrow \text{Sign}_{sk}(m)$.)
- **Ver**, on input (pk, m, σ) , produces either **Accept** or **Reject**. (We write $out \leftarrow \text{Ver}_{pk}(m, \sigma)$.)

and, in addition, the following properties hold:

- *Correctness*: For every $n \in \mathbb{N}$ and $m \in \{0, 1\}^n$:

$$\Pr [(pk, sk) \leftarrow \text{Gen}(1^n) : \text{Ver}_{pk}(m, \text{Sign}_{sk}(m)) = \text{Accept}] = 1$$

- *Uniqueness*: For every $m \in \{0, 1\}^*$, and $pk \in \{0, 1\}^*$, there exists at most one $\sigma \in \{0, 1\}^*$ for which $\text{Ver}_{pk}(m, \sigma) = \text{Accept}$.

We next turn to discussing what it means for such a scheme to be secure. A natural definition of security is the notion of *existential unforgeability against adaptive chosen-message attacks* [GMR88], which requires that an adversary knowing the public key, even if allowed to adaptively choose a bounded number of messages and observe their signatures, is unable to forge any signature for a message they have not yet queried. We formalize this by allowing the adversary access to an oracle for **Sign**, as follows:

Definition 2. We say that a signature scheme is **unforgeable** if, for every non-uniform probabilistic polynomial-time oracle-aided algorithm \mathcal{A} , there is some negligible function $\epsilon(\cdot)$ such that for all $n \in \mathbb{N}$:

$$\Pr \left[(pk, sk) \leftarrow \text{Gen}(1^n); (m, \sigma) \leftarrow \mathcal{A}^{\text{Sign}_{sk}(\cdot)}(1^n, pk) : \text{Ver}_{pk}(m, \sigma) = \text{Accept} \wedge \text{Valid} \right] \leq \epsilon(n)$$

where **Valid** is the event that none of \mathcal{A} 's queries were for the signature of the output message m .

We will define a weaker notion of a signature scheme being $\ell(\cdot)$ -**unforgeable** identically to the above, with the exception that **Valid** is the event that the following *two* conditions on \mathcal{A} are true:

- \mathcal{A} has queried its oracle at most $\ell(n)$ times.
- None of \mathcal{A} 's queries were for the signature of the output message m .

The bounded notion of $\ell(\cdot)$ -unforgeability is primarily useful to prove our concrete security loss bound, whereas our main result applies to the general notion of unforgeability. Furthermore, for the purposes of the impossibility result, we weaken the definition of unforgeability to a worst-case definition, as this will strengthen our main theorem (by showing that basing even this weak notion of security on standard assumptions will incur a security loss):

Definition 3. We say that a signature scheme is **weakly unforgeable** (respectively, weakly $\ell(\cdot)$ -unforgeable) if, for every non-uniform probabilistic polynomial-time oracle-aided algorithm \mathcal{A} and every $n \in \mathbb{N}$:

$$\Pr \left[(pk, sk) \leftarrow \text{Gen}(1^n); (m, \sigma) \leftarrow \mathcal{A}^{\text{Sign}_{sk}(\cdot)}(1^n, pk) : \text{Ver}_{pk}(m, \sigma) = \text{Accept} \wedge \text{Valid} \right] < 1$$

where **Valid** is defined as above (and respectively for $\ell(\cdot)$ -unforgeability). In particular, we say that a non-uniform probabilistic polynomial-time algorithm \mathcal{A} *breaks weak unforgeability* of a signature scheme $(\text{Gen}, \text{Sign}, \text{Ver})$ if the probability above is equal to 1.

2.3 Intractability Assumptions

Next, we define intractability assumptions in a manner originally proposed in [Nao03]. Formally, we can model an assumption as a “security game” involving an interaction between a probabilistic *challenger* \mathcal{C} and *adversary* \mathcal{A} , after which \mathcal{C} will output either **Accept** or **Reject**. We say that an adversary \mathcal{A} breaks the assumption if \mathcal{C} accepts with probability non-negligibly greater than a certain threshold.

For instance, an assumption that a function f is one-way could be modeled by a two-round interaction where \mathcal{C} sends \mathcal{A} the image $y = f(x)$ on a uniformly random input x , \mathcal{A} sends a message x' to \mathcal{C} , and \mathcal{C} accepts if and only if $f(x') = y$. In this case, \mathcal{A} breaks the assumption if it inverts f (i.e., \mathcal{C} accepts) with probability non-negligibly greater than zero.

As an example of an assumption that would have a non-zero threshold, an assumption that two distributions \mathcal{D}_0 and \mathcal{D}_1 are indistinguishable could be modeled by the two-round interaction where \mathcal{C} picks a random $b \in \{0, 1\}$, sends \mathcal{A} a sample from \mathcal{D}_b , receives b' from \mathcal{A} , and accepts if $b = b'$. Then \mathcal{A} would only break the assumption if \mathcal{C} accepts with probability non-negligibly greater than a threshold of $1/2$. Formally, we model these assumptions following [Pas11]:

Definition 4. For polynomial $r(\cdot)$, we denote an $r(\cdot)$ -**round intractability assumption** by a pair $(\mathcal{C}, t(\cdot))$, where $t(\cdot)$ is a function and \mathcal{C} is a probabilistic interactive algorithm with input 1^n and an *a priori* bound of $r(n)$ rounds of communication. We say that $(\mathcal{C}, t(\cdot))$ is **secure** if the following is true:

For any non-uniform probabilistic polynomial-time interactive algorithm \mathcal{A} , there exists a negligible function $\epsilon(\cdot)$ such that, for all $n \in \mathbb{N}$:

$$\Pr [\langle \mathcal{A}, \mathcal{C} \rangle(1^n) = \text{Accept}] \leq t(n) + \epsilon(n)$$

Furthermore, we say that a specific \mathcal{A} *breaks* the assumption if the above inequality is not true with respect to that \mathcal{A} ; in particular, for some polynomial $p(\cdot)$, we say that \mathcal{A} breaks $(\mathcal{C}, t(\cdot))$ with probability $1/p(\cdot)$ if, for infinitely many $n \in \mathbb{N}$,

$$\Pr [\langle \mathcal{A}, \mathcal{C} \rangle(1^n) = \text{Accept}] \geq t(n) + \frac{1}{p(n)}$$

We also call a pair $(\mathcal{C}, t(\cdot))$ a **bounded-round intractability assumption** if there exists some polynomial $r(\cdot)$ such that $(\mathcal{C}, t(\cdot))$ is an $r(\cdot)$ -round intractability assumption.

We note that any standard cryptographic security assumption can be modeled as a pair $(\mathcal{C}, t(\cdot))$ of this form, including our definitions above of the security of signature schemes. (In this case, the threshold $t(n)$ would be zero, and \mathcal{C} would have $r(n) = 2\ell(n) + 2$ rounds of communication, first generating (pk, sk) and sending pk to \mathcal{A} , then signing $\ell(n)$ messages for \mathcal{A} , and finally receiving (m, σ) and outputting the result of Ver , or Reject if it had already signed m .) In particular, this is why we require an *a priori* bound on the number of rounds $r(\cdot)$ of the assumption; it will allow us to avoid such trivial reductions as reducing unforgeability to itself (for which we could obviously not prove the impossibility result).

2.4 Black-Box Reductions

Finally, we briefly discuss what it means for one assumption to be based on another assumption. In particular, given two assumptions $(\mathcal{C}_1, t_1(\cdot))$ and $(\mathcal{C}_2, t_2(\cdot))$, basing the hardness of \mathcal{C}_1 on that of \mathcal{C}_2 in a black-box way would classically entail, given an arbitrary adversary \mathcal{A}_2 which can break $(\mathcal{C}_2, t_2(\cdot))$ constructing a polynomial-time procedure \mathcal{A}_1 that breaks $(\mathcal{C}_1, t_1(\cdot))$ through standard interactions with \mathcal{A}_2 (i.e., using \mathcal{A}_2 in a *black-box* manner).

Notably, there is no guarantee that \mathcal{A}_1 invoke \mathcal{A}_2 only once; it could be the case that there are polynomially many invocations, or even “nested” invocations (e.g., multiple concurrent invocations such that the rounds of communication may be interleaved or dependent on one another), of \mathcal{A}_2 during the execution of \mathcal{A}_1 . We can formalize this by imagining \mathcal{A}_1 as a polynomial-time *reduction* \mathcal{R} that has oracle access to the interactive algorithm \mathcal{A} (formerly \mathcal{A}_2):

Definition 5. We refer to a probabilistic polynomial-time oracle-aided algorithm \mathcal{R} as a **black-box reduction for basing the hardness of assumption $(\mathcal{C}_1, t_1(\cdot))$ on that of $(\mathcal{C}_2, t_2(\cdot))$** if, given any deterministic \mathcal{A} that breaks $(\mathcal{C}_1, t_1(\cdot))$, $\mathcal{R}^{\mathcal{A}}$ breaks $(\mathcal{C}_2, t_2(\cdot))$. We refer to such a reduction as **fixed-parameter** if, given common input 1^n , $\mathcal{R}^{\mathcal{A}}$ queries \mathcal{A} only on input 1^n .

We notably restrict our attention to oracles that are deterministic (or have some fixed randomness), as this allows us to consider cases where the reduction \mathcal{R} can rewind or restart its oracle. We shall also restrict our attention, similarly to [Pas11], to the case of *fixed-parameter* reductions where \mathcal{R} invokes its adversary \mathcal{A} only using a single security parameter (i.e., \mathcal{A} must be the same algorithm in each instance); in particular, for security parameter n , we allow \mathcal{R} to run up to $M(n)$ instances of some parameterized adversary $\mathcal{A}(1^n)$. Lastly, we can also apply the above concept to our definition of weak $\ell(\cdot)$ -unforgeability (and define a fixed-parameter reduction identically for this case):

Definition 6. We say that a probabilistic polynomial-time oracle-aided algorithm \mathcal{R} is a *black-box reduction for basing weak unforgeability of a signature scheme $(\text{Gen}, \text{Sign}, \text{Ver})$ on the hardness of an assumption $(\mathcal{C}, t(\cdot))$* (resp. for weak $\ell(\cdot)$ -unforgeability) if, for every deterministic algorithm \mathcal{A} that breaks weak unforgeability of $(\text{Gen}, \text{Sign}, \text{Ver})$ (i.e., forges a signature with probability 1), there is a polynomial $p(\cdot)$ such that, for infinitely many $n \in \mathbb{N}$, $\mathcal{R}^{\mathcal{A}}$ breaks $(\mathcal{C}, t(\cdot))$ with probability $1/p(n)$.

Finally, we wish to formalize the *security loss* of such a reduction \mathcal{R} , or the loss in the reduction’s success probability proportionate to its time efficiency. We state this as follows:

Definition 7. Let \mathcal{R} be a black-box reduction for basing the hardness of assumption $(\mathcal{C}_1, t_1(\cdot))$ on that of $(\mathcal{C}_2, t_2(\cdot))$. Given any deterministic \mathcal{A} and for each $n \in \mathbb{N}$:

- Let $\text{Success}_{\mathcal{A}}(n) = \Pr[\langle \mathcal{A}, \mathcal{C}_1 \rangle(1^n) = \text{Accept}] - t_1(n)$ (that is, the probability with which \mathcal{A} breaks $(\mathcal{C}_1, t_1(\cdot))$, taken over all randomness of \mathcal{A} and \mathcal{C}_1).
- Let $\text{Query}_{\mathcal{A}}(n)$ denote the maximum, over all randomness of \mathcal{A} and \mathcal{C}_1 , of the possible number of messages sent from \mathcal{C}_1 to \mathcal{A} during the experiment $[\mathcal{A} \leftrightarrow \mathcal{C}_1](1^n)$.
- Let $\text{Success}_{\mathcal{R}^{\mathcal{A}}}(n) = \Pr[\langle \mathcal{R}^{\mathcal{A}}, \mathcal{C}_2 \rangle(1^n) = \text{Accept}] - t_2(n)$ (that is, the probability with which $\mathcal{R}^{\mathcal{A}}$ breaks $(\mathcal{C}_2, t_2(\cdot))$ taken over all randomness of \mathcal{A} , \mathcal{C}_2 , and \mathcal{R}).
- Let $\text{Query}_{\mathcal{R}^{\mathcal{A}}}(n)$ denote the maximum, over all randomness of \mathcal{A} , \mathcal{C}_2 , and \mathcal{R} , of the possible number of messages sent from \mathcal{R} to \mathcal{A} during the experiment $[\mathcal{R}^{\mathcal{A}} \leftrightarrow \mathcal{C}_2](1^n)$.

Then we say that the **security loss** of \mathcal{R} is given by:

$$\lambda_{\mathcal{R}}(n) = \max_{\mathcal{A}} \left(\frac{\text{Success}_{\mathcal{A}}(n)}{\text{Success}_{\mathcal{R}^{\mathcal{A}}}(n)} \frac{\text{Query}_{\mathcal{R}^{\mathcal{A}}}(n)}{\text{Query}_{\mathcal{A}}(n)} \right)$$

Furthermore, we say that \mathcal{R} is **linear-preserving** if its security loss is bounded above by a fixed polynomial independent of \mathcal{A} —that is, there is a polynomial $p(\cdot)$ for which, for all sufficiently large $n \in \mathbb{N}$ and every \mathcal{A} , $\lambda_{\mathcal{R}}(n) \leq p(n)$.

We note that, as we consider black-box reductions, we consider the ratio between the communication complexities of \mathcal{R} and \mathcal{A} as opposed to the running times when determining the security loss. While many other recent works (e.g., [HJK12, BJLS16]) use a definition which, though similar to the above, measures actual running time rather than rounds of communication, we note that our definition is at least as strong as time-based alternatives, and formally prove this fact in Appendix B.

3 Main Theorem

As our main theorem, we prove the following result:

Theorem 2. Let $\Pi = (\text{Gen}, \text{Sign}, \text{Ver})$ be a unique signature scheme, and let $(\mathcal{C}, t(\cdot))$ be some $r(\cdot)$ -round intractability assumption for polynomial $r(\cdot)$. If there exists some fixed-parameter black-box reduction \mathcal{R} for basing weak unforgeability of Π on the hardness of $(\mathcal{C}, t(\cdot))$, then either:

- (1) \mathcal{R} is not a linear-preserving reduction, or
- (2) there exists a polynomial-time adversary \mathcal{B} that breaks the assumption $(\mathcal{C}, t(\cdot))$.

We note that this result also applies to the slightly more general notion of *rerandomizable* signatures through an almost identical argument; we discuss this in more detail in Appendix C. Theorem 2 follows in a straightforward manner from the following lemma, which is a concrete security loss bound analogous to Coron’s in [Cor02], but generalized so that it handles arbitrary (i.e., not just simple) reductions:

Lemma 1. Let $\Pi = (\text{Gen}, \text{Sign}, \text{Ver})$ be a unique signature scheme, and let $(\mathcal{C}, t(\cdot))$ be some $r(\cdot)$ -round intractability assumption for polynomial $r(\cdot)$. If for some polynomial $\ell(\cdot)$ there exists some fixed-parameter black-box reduction \mathcal{R} for basing weak $\ell(\cdot)$ -unforgeability of Π on the hardness of $(\mathcal{C}, t(\cdot))$, then either \mathcal{R} 's security loss is at least

$$\lambda_{\mathcal{R}}(n) \geq \sqrt{\ell(n)} - (r(n) + 1)$$

for all sufficiently large $n \in \mathbb{N}$, or there exists a polynomial-time adversary \mathcal{B} that breaks the assumption $(\mathcal{C}, t(\cdot))$.

First, we show that Lemma 1 implies Theorem 2:

Proof. Given Lemma 1, assume for the sake of contradiction that $(\mathcal{C}, t(\cdot))$ is secure, and that the \mathcal{R} described in Theorem 2 is linear-preserving. Then there exists a polynomial $p(\cdot)$ such that $\lambda_{\mathcal{R}}(n) \leq p(n)$ for all sufficiently large n . Because \mathcal{R} by assumption is a reduction from (unbounded) weak unforgeability, it must likewise be a reduction from the strictly more specific definition of weak $\ell(\cdot)$ -unforgeability for any polynomial $\ell(\cdot)$, since every weak $\ell(\cdot)$ -unforgeability adversary will trivially break unbounded weak unforgeability. So, let $\ell(n) \triangleq (p(n) + r(n) + 2)^2$; hence, by Lemma 1, either there exists a polynomial-time adversary \mathcal{B} which breaks $(\mathcal{C}, t(\cdot))$ —which is false by assumption—or the security loss of \mathcal{R} is no less than

$$\sqrt{\ell(n)} - (r(n) + 1) = p(n) + 1 > p(n)$$

for all sufficiently large n , which is a contradiction and hence proves Theorem 2. \square

Hence, the remainder of the section is dedicated to proving Lemma 1. Our proof of Lemma 1 follows four major steps, which we shall describe here at a high level before beginning the full argument.

Constructing an Ideal Adversary. First, we describe an “ideal” adversary \mathcal{A} which is *guaranteed* to break the security of Π while sending $\ell(n)$ queries, but does so by brute force and hence does not run in polynomial time. Our objective then is to create a meta-reduction \mathcal{B} that *almost* always emulates the interaction $\mathcal{R}^{\mathcal{A}}$ between \mathcal{R} and \mathcal{A} . If it does so with, say, probability $1 - 1/p(n)$, then \mathcal{B} will break the assumption $(\mathcal{C}, t(\cdot))$ with probability at least $\text{Success}_{\mathcal{R}^{\mathcal{A}}}(n) - 1/p(n)$. However, this means that $\mathcal{R}^{\mathcal{A}}$ itself cannot have success probability non-negligibly greater than $1/p(n)$; otherwise, \mathcal{C} would be broken with non-negligible probability by \mathcal{B} .

The “ideal” \mathcal{A} will pick $\ell(n)$ messages (\vec{m}) at random and query \mathcal{R} for the signatures of each of these messages in turn, and will finally brute-force a secret key from the results and use that key to forge a signature for another random message m^* , which it will return. Crucially, \mathcal{A} will also verify \mathcal{R} 's responses to its queries and return \perp instead if not all are correct signatures. By construction \mathcal{A} breaks $\ell(\cdot)$ -weak unforgeability; however, due to the brute-force step, it (and consequently $\mathcal{R}^{\mathcal{A}}$) will not run in polynomial time.

Constructing a Meta-Reduction. Hence, to *efficiently* emulate $\mathcal{R}^{\mathcal{A}}$, we create the meta-reduction \mathcal{B} . \mathcal{B} will run \mathcal{R} and forward communicate with \mathcal{C} as normal; when \mathcal{R} would start a new instance of its adversary \mathcal{A} , \mathcal{B} will generate messages \vec{m} and m^* randomly (i.e., identically to \mathcal{A}) and forward queries to \mathcal{R} in the same manner as \mathcal{A} . However, when \mathcal{R} requires an instance to

provide a forged signature, \mathcal{B} will also “rewind” the simulated execution to the start of each query for that instance and try to query \mathcal{R} with m^* instead of the message it would normally query. If \mathcal{R} gives a response to the rewound query, then \mathcal{B} has (efficiently) found a forgery for m^* , which it can return to \mathcal{R} when it requests a forgery from the corresponding instance of \mathcal{A} .

\mathcal{B} , while rewinding, will abort (and try rewinding the next slot instead) if either \mathcal{R} would communicate externally with \mathcal{C} (which \mathcal{B} of course cannot rewind) or \mathcal{R} would request a forgery for some other simulated instance of \mathcal{A} during the simulated execution of $\mathcal{R}^{\mathcal{A}}$ (i.e., before responding to the rewound query). In particular, this strategy ensures that recursive rewinding as in [Pas11] will not be required, since \mathcal{B} will never attempt to start rewinding some instance while rewinding a different one.

Furthermore, \mathcal{B} will “verify” all of \mathcal{R} ’s responses to its signature queries (in the non-rewound part of the execution) in the same manner as \mathcal{A} , likewise returning \perp from the simulated instance of \mathcal{A} if not all responses are valid. So, whenever either \mathcal{R} gives a simulated instance one or more incorrect responses or \mathcal{B} successfully extracts a forgery (noting that, by the uniqueness property, the forgeries they return must be identical, which is crucial), \mathcal{A} and \mathcal{B} ’s simulations of \mathcal{A} will be identically distributed to one another.

Bounding the Failure Probability. So, to bound the probability with which \mathcal{B} does *not* successfully emulate some instance of \mathcal{A} , we must bound the probability that all of \mathcal{B} ’s queries to \mathcal{R} (\vec{m}) are correctly answered, yet the rewinding of *every one* of the queries fails due to either \mathcal{R} responding badly to m^* , \mathcal{R} communicating externally with \mathcal{C} , or a forgery request for another simulated instance of \mathcal{A} occurring before \mathcal{R} responds.

We bound this probability by using a counting argument similar to that exhibited in the introduction. In particular, we consider the messages \vec{m} and m^* for a particular instance, fixing the randomness outside of that instance arbitrarily. Then we show that, for any “bad” sequencing of these messages such that the non-rewound execution succeeds but every rewinding fails, many (though not all, because of the possibility for rewindings to fail due to an end message or external communication) of the rewindings of this sequence will correspond to “good” sequences where \mathcal{B} returns \perp due to receiving an incorrect response from \mathcal{R} .

What we intuitively show is that, in every set of $\ell(n) + 1$ sequences corresponding to a sequence and its various rewindings, at most $M(n) + r(n) + 1$ can be bad (since, informally, given a bad sequence, in expectation only $M(n) + r(n)$ of its rewindings can fail for reasons besides \mathcal{R} responding incorrectly, i.e., due to nested end messages or external communication), where $M(n)$ is the maximum number of instances of \mathcal{A} which \mathcal{R} executes (which we show by our construction of \mathcal{A} must be no less than the number of successful end messages). Hence we obtain a bound of

$$\frac{M(n) + r(n) + 1}{\ell(n) + 1}$$

on the failure probability for each instance, which by the union bound over all $M(n)$ instances sums to an overall failure probability of less than

$$M(n) \left(\frac{M(n) + r(n) + 1}{\ell(n)} \right)$$

Bounding the Security Loss. This does not immediately imply a bound on the security loss $\lambda_{\mathcal{R}}(n)$, since $M(n)$ can be arbitrarily large. However, as in the technical overview, we bound the

security loss by showing that, if $M(n)$ is large, this requires a large enough running time of \mathcal{R} that we still obtain a non-trivial lower bound on the security loss. Specifically, recalling that

$$\lambda_{\mathcal{R}}(n) \geq \frac{\text{Success}_{\mathcal{A}}(n)}{\text{Success}_{\mathcal{R}\mathcal{A}}(n)} \frac{\text{Query}_{\mathcal{R}\mathcal{A}}(n)}{\text{Query}_{\mathcal{A}}(n)}$$

we notice first that $\text{Query}_{\mathcal{R}\mathcal{A}}(n)/\text{Query}_{\mathcal{A}}(n) \geq M(n)$, which follows (with some subtleties which we defer to the main proof) from the fact that \mathcal{R} will in the worst case run $M(n)$ instances of \mathcal{A} . So, since $\text{Success}_{\mathcal{A}}(n) = 1$ by construction, and since, as we discussed previously, $\text{Success}_{\mathcal{R}\mathcal{A}}(n)$ cannot be non-negligibly larger than the failure probability of \mathcal{B} , we have

$$\lambda_{\mathcal{R}}(n) \geq \frac{\ell(n)}{M(n) + r(n) + 1}$$

which immediately implies the bound when $M(n) < \sqrt{\ell(n)} - (r(n) + 1)$ (and so $\lambda_{\mathcal{R}}(n) > \sqrt{\ell(n)}$). On the other hand, we also know that $\text{Success}_{\mathcal{R}\mathcal{A}}(n) \leq 1$ trivially, and so it is also the case that $\lambda_{\mathcal{R}}(n) \geq M(n)$, which implies the bound when $M(n) \geq \sqrt{\ell(n)} - (r(n) + 1)$, completing the proof of Lemma 1 and hence Theorem 2.

3.1 The “Ideal” Adversary

We now proceed to the formal proof of Lemma 1. First, we exhibit an inefficient adversary \mathcal{A} that will break weak $\ell(\cdot)$ -unforgeability, so that we can later construct an efficient \mathcal{B} to simulate it while running \mathcal{R} in order to break the assumption $(\mathcal{C}, t(\cdot))$.

Let $\Pi = (\text{Gen}, \text{Sign}, \text{Ver})$ be a unique signature scheme, and let $(\mathcal{C}, t(\cdot))$ be some $r(\cdot)$ -round intractability assumption for polynomial $r(\cdot)$. Assume that there exists some black-box reduction \mathcal{R} for basing weak $\ell(\cdot)$ -unforgeability of Π on the hardness of $(\mathcal{C}, t(\cdot))$ which, given an oracle breaking weak unforgeability, will break $(\mathcal{C}, t(\cdot))$ with probability $1/p(\cdot)$ for some polynomial $p(\cdot)$.

First, for any polynomial $\ell(n)$, we construct an inefficient but easily emulatable adversary \mathcal{A} which sends at most $\ell(n)$ queries and is guaranteed to break weak unforgeability of Π . Since we will require \mathcal{A} to be deterministic during execution yet generate random messages, we will assume that \mathcal{A} is formally given by a deterministic interactive $\mathcal{A}^{\mathcal{O}}$ which has access to a random oracle \mathcal{O} (of course, \mathcal{O} is not needed for our actual constructions, as we shall emulate \mathcal{A}), which, as in [Pas11], is given by a random variable which is uniformly distributed over functions $\{0, 1\}^* \rightarrow \{0, 1\}^{\infty}$. In particular, this ensures that the queries output by $\mathcal{A}^{\mathcal{O}}$ are uniformly distributed (i.e., over the randomness of \mathcal{O}), but are still preserved under rewinding.

We shall henceforth denote by \mathcal{A} the specific adversary $\mathcal{A}^{\mathcal{O}}$ which, on input 1^n , behaves as described in Figure 3. Informally, \mathcal{A} makes $\ell(n)$ signature queries, generating the message for each query by applying \mathcal{O} to the current partial transcript. Finally, after receiving responses for each query, \mathcal{A} returns a brute-forced forgery, but only if it successfully “verifies” the transcript by ensuring that each query’s response is valid and that each query in the transcript was generated in the correct manner (i.e., by \mathcal{O} applied to the prior partial transcript).

It is straightforward to see that \mathcal{A} , given any fixed oracle \mathcal{O} , will break weak $\ell(\cdot)$ -unforgeability; given an honest signing oracle (which will always send the correct partial transcript), it will always return some (m, σ) such that $\text{Ver}_{pk}(m, \sigma) = \text{Accept}$, m was not queried (as m^* is not equal to any of the queries m_i), and only $\ell(n)$ queries were made.

- Initially, receive a message pk , the public key; respond (i.e., generate m_1) according to the next step for $i = 1$.
- On receiving a message consisting of a partial transcript $\tau = (pk, m_1, \sigma_1, \dots, m_{i-1}, \sigma_{i-1})$ for some $i \in [\ell(n)]$, do the following:
 - Generate m_i by taking the first n bits resulting from applying the oracle \mathcal{O} to τ .
 - Return the new partial transcript $\tau || m_i$.
- On receiving a message consisting of a complete transcript $\tau = (pk, m_1, \sigma_1, \dots, m_{\ell(n)}, \sigma_{\ell(n)})$ (we shall refer to such a message as a “forgery request” or “end message”), do the following:
 - Verify that, for each signature σ_i , $\text{Ver}_{pk}(m_i, \sigma_i) = \text{Accept}$. If not true for all i , return \perp .
 - Verify that, for each message m_i , m_i is equal to the first n bits resulting from applying the oracle \mathcal{O} to the prefix transcript $\tau_{<i} = (pk, m_1, \sigma_1, \dots, m_{i-1}, \sigma_{i-1})$. If not true for all i , then return \perp .
 - Finally, generate a random message m^* (distinct from each m_i in τ) by applying \mathcal{O} to the transcript τ , use brute force to find a signature σ^* for which $\text{Ver}_{pk}(m^*, \sigma^*) = \text{Accept}$, and return the forgery (m^*, σ^*) .

Figure 3: Formal description of the “ideal” adversary $\mathcal{A}^{\mathcal{O}}$.

However, when interacting with \mathcal{R} , which is not bound by the rules of an honest oracle, the transcript verification is necessary to prevent \mathcal{R} from “cheating” in certain ways during its interaction. First, we wish to ensure that \mathcal{R} will return valid signatures to queries as often as possible. Also, we wish to ensure that \mathcal{R} is actually required to answer $\ell(n)$ signature queries generated randomly by \mathcal{A} and cannot, for instance, immediately send \mathcal{A} an end message with an artificially generated transcript; this is done by using the oracle \mathcal{O} to generate \mathcal{A} ’s messages and ensuring that the transcript is consistent with the oracle. Formally, we make the following claim, which will be useful later:

Claim 1. There exists a negligible function $\nu(\cdot)$ such that, for all $n \in \mathbb{N}$, the probability, over all randomness in the experiment $[\mathcal{R}^{\mathcal{A}^{\mathcal{O}}} \leftrightarrow \mathcal{C}](1^n)$, that some instance of \mathcal{A} returns a forgery (i.e., something besides \perp) to \mathcal{R} without having received $\ell(n)$ different responses to its signature queries from \mathcal{R} , is less than $\nu(n)$.

Proof. Assume that \mathcal{A} returns a forgery to \mathcal{R} after receiving fewer than $\ell(n)$ responses to its signature queries; we shall demonstrate that the only way this can happen is if \mathcal{R} correctly predicts the output of the random oracle \mathcal{O} , which can happen with only negligible probability (specifically, no greater than $\nu(n) \triangleq \ell(n)2^{-n}$) due to the uniformly random choice of \mathcal{O} .

Let $i \in [\ell(n) - 1]$ (we exclude $\ell(n)$ because \mathcal{R} by assumption makes a forgery request) be the index of the first signature query “skipped” by \mathcal{R} —that is, the first index for which \mathcal{R} does *not* send \mathcal{A} a partial transcript $\tau_{\leq i} = (pk, m_1, \sigma_1, \dots, m_i, \sigma_i)$. Of course, there must exist such an i by the assumption that \mathcal{R} sends \mathcal{A} fewer than $\ell(n)$ messages. In order to receive a forgery from \mathcal{A} (and not \perp), \mathcal{R} must send a complete transcript $\tau = (pk, m_1, \sigma_1, \dots, m_{\ell(n)}, \sigma_{\ell(n)})$, notably having the property that each message m_j is equivalent to the first n bits of the result of applying the random oracle \mathcal{O} to the prior transcript $m_{\leq j-1}$.

In particular, since by assumption \mathcal{R} did not send \mathcal{A} any partial transcript $\tau_{\leq i}$, it is impossible for \mathcal{R} to have received from \mathcal{A} the correct message m_{i+1} , which can, by construction of \mathcal{A} , only be retrieved by sending \mathcal{A} the (unique) partial transcript $\tau_{\leq i}$ for which all m_j are generated according to \mathcal{O} and all σ_j are the (unique) accepting signature for the respective m_j .

By our assumption, then, \mathcal{R} will only be able to send the correct message m_{i+1} in its forgery request, and hence receive a forgery from \mathcal{A} , with probability 2^{-n} —that is, the chance that the output of the (uniformly distributed) random oracle \mathcal{O} matches the sent m_{i+1} in the first n bits. And as, by our assumption, \mathcal{R} sends fewer than $\ell(n)$ messages to \mathcal{A} , it cannot attempt to “guess” the correct m_{i+1} more than $\ell(n)$ times; thus the probability that \mathcal{A} will return a forgery, taken over the randomly distributed \mathcal{O} , is by the union bound no more than $\nu(n) = \ell(n)2^{-n}$, which as desired is negligible. \square

Furthermore, this construction of \mathcal{A} (using the oracle \mathcal{O}) allows us to assume, without loss of generality, that the reduction \mathcal{R} will never rewind an instance of \mathcal{A} —this is without loss of generality because there is a single accepting transcript for each choice of the oracle \mathcal{O} . Namely, given an oracle \mathcal{O} , if \mathcal{R} always provides correct signatures, then \mathcal{A} ’s messages (including the forgery it returns) and \mathcal{R} ’s responses are fully determined by \mathcal{O} and the uniqueness property of Π . Meanwhile, if \mathcal{R} does not provide correct signatures, \mathcal{A} will not return a forgery.

Because \mathcal{A} breaks unforgeability, and by the assumed properties of \mathcal{R} and the determinism of $\mathcal{A}^{\mathcal{O}}$ for any fixed oracle \mathcal{O} , it must be true that there exists polynomial $p(\cdot)$ such that

$$\Pr [\langle \mathcal{R}^{\mathcal{A}^{\mathcal{O}}}, \mathcal{C} \rangle(1^n) = \text{Accept}] \geq t(n) + \frac{1}{p(n)}$$

for any oracle \mathcal{O} . As such, by the fact that \mathcal{R} is fixed-parameter, we can observe that, for any n , averaging this probability over all possible oracles \mathcal{O} , we likewise have

$$\Pr [\langle \mathcal{R}^{\mathcal{A}}, \mathcal{C} \rangle(1^n) = \text{Accept}] \geq t(n) + \frac{1}{p(n)}$$

even though \mathcal{A} over a randomly-chosen \mathcal{O} is not deterministic.

Of course, \mathcal{A} is inefficient, so, in order to break the assumption $(\mathcal{C}, t(\cdot))$, we must construct an efficient \mathcal{B} that is able to run \mathcal{R} while emulating its interactions with \mathcal{A} most of the time. Hence, the remainder of the proof will be dedicated to constructing this meta-reduction and analyzing the probability with which it succeeds in emulating the “ideal” \mathcal{A} . Intuitively, if \mathcal{B} successfully emulates \mathcal{A} at least $1 - 1/p'(n)$ of the time for some function $p'(\cdot)$, then:

$$|\Pr [\langle \mathcal{R}^{\mathcal{A}}, \mathcal{C} \rangle(1^n) = \text{Accept}] - \Pr [\langle \mathcal{B}, \mathcal{C} \rangle(1^n) = \text{Accept}]| \leq \frac{1}{p'(n)}$$

$$\Pr [\langle \mathcal{B}, \mathcal{C} \rangle(1^n) = \text{Accept}] \geq t(n) + \frac{1}{p(n)} - \frac{1}{p'(n)}$$

meaning that \mathcal{B} must break \mathcal{C} with probability at least $1/p(n) - 1/p'(n)$, as desired. Hence, what we shall effectively show in the subsequent steps is that, unless the security loss of \mathcal{R} is large, $1/p'(n)$ will be non-negligibly smaller than $1/p(n)$, and thus \mathcal{B} will break the security of $(\mathcal{C}, t(\cdot))$.

Slots. As a notational aside, we shall for simplicity henceforth refer to the pair of a query made by \mathcal{A} (or something, such as \mathcal{B} , which emulates \mathcal{A}) and its corresponding response by \mathcal{R} as a *slot* (v_{open}, v_{close}) . Such a slot is determined by two views: the “opening” of the slot, or the view v_{open} of the execution of \mathcal{R} immediately before \mathcal{A} ’s query to \mathcal{R} , and the “closing” of the slot, or the view v_{close} of the execution immediately after \mathcal{R} responds to the respective query. (We will also often refer to the view of \mathcal{R} immediately *after* the opening query of a message m , which we shall denote by the concatenation $v_{open}||m$.)

3.2 The Meta-Reduction

We next construct the meta-reduction \mathcal{B} which will efficiently emulate \mathcal{A} . Let \mathcal{B} be as described formally in Figure 4; informally, \mathcal{B} will run \mathcal{R} internally, forwarding communication to \mathcal{C} as \mathcal{R} would while also internally simulating instances of \mathcal{A} interacting with \mathcal{R} . The primary difference between \mathcal{B} and the “ideal” execution of \mathcal{A} interacting with \mathcal{R} is that \mathcal{B} , being restricted to polynomial time, cannot brute-force forgeries as \mathcal{A} does; instead, while simulating each instance of \mathcal{A} , \mathcal{B} will select at random a message m^* for which to forge a signature and attempt to rewind each slot for that instance, substituting m^* for the original message.⁴

If \mathcal{R} ever returns a valid signature σ^* for m^* , then \mathcal{B} may store that signature and finally return (m^*, σ^*) when \mathcal{R} requests a forgery for that instance. However, if one of the following “bad events” occurs:

- \mathcal{R} fails to return a valid signature of m^* .
- \mathcal{R} asks for a forgery for another instance before returning a signature.
- \mathcal{R} requires external communication with \mathcal{C} (which cannot be rewound) before returning a signature.

then \mathcal{B} will abort and try the next slot. In this way we circumvent the issue of having to recursively rewind nested end messages as in [Pas11].

First, we can show that \mathcal{B} , unlike \mathcal{A} , is efficient:

Claim 2. There exists a polynomial $t(n)$ such that, for all $n \in \mathbb{N}$, $\text{Real}(1^n)$ is guaranteed to run in time at most $t(n)$.

Proof. It suffices to show that \mathcal{B} sends a polynomial number of simulated messages to \mathcal{R} , as \mathcal{R} and \mathcal{C} run in polynomial time by definition and all other operations performed by \mathcal{B} are clearly polynomial-time. To that end, we observe that each instance I contains $O(\ell(n) + r(n))$ rounds of communication (including both communication with \mathcal{R} and \mathcal{R} ’s external communication with \mathcal{C}). Furthermore, there are at most a polynomial number of instances of \mathcal{A} started by \mathcal{R} —call this number $M(n)$ —and so disregarding rewinding we know that there must be $O((\ell(n) + r(n))M(n))$ total messages.

\mathcal{B} will rewind \mathcal{R} as described above; however, we know that there can be at most $\ell(n)M(n)$ rewind slots (as each slot is rewind once and \mathcal{B} will never rewind recursively), each of which might contain $O((\ell(n) + r(n))M(n))$ nested messages; hence, the total number of messages is

⁴That is, when “rewinding” a slot (v_{open}, v_{close}) , \mathcal{B} will simulate interaction with \mathcal{R} starting from the view $v_{open}||m^*$.

- Set initial view $v \leftarrow \perp$ and set $k \leftarrow 1$. Execute \mathcal{R} , updating the current view v according to the following rules.
- When \mathcal{R} begins a new instance of \mathcal{A} and sends a public key pk , label this instance as instance k . Generate and store $\ell(n)$ random queries $\vec{m}_k = (m_{k,1}, \dots, m_{k,\ell(n)})$ and a target forgery m_k^* . (Abort and return **Fail** if m_k^* is equal to a message in \vec{m}_k .^a) Also let $pk_k \leftarrow pk$ and initialize the forgery $f_k \leftarrow \{\}$. Lastly, respond with $\tau_k^* = pk_k || m_{k,1}$ and increment k .
- When \mathcal{R} attempts to communicate externally with \mathcal{C} , forward the message, return \mathcal{C} 's response to \mathcal{R} , and update v accordingly.
- When \mathcal{R} sends a transcript $\tau = (pk, m_{I,1}, \sigma_{I,1}, \dots, m_{I,j}, \sigma_{I,j})$ to some simulated instance I of \mathcal{A} , store the signature $\sigma_{I,j}$ and do the following:
 - If $j = \ell(n)$ (i.e., this is an end message), then do the following:
 - * If τ is an inconsistent transcript (i.e., $m_{I,i}$ or $\sigma_{I,i}$ in τ is different from the stored $m_{I,i}$ or $\sigma_{I,i}$ (respectively) for some $i \in [\ell(n)]$, or not all $\sigma_{I,i}$ have been stored) or \mathcal{R} 's response to some signature query j was invalid (i.e., $\text{Ver}_{pk_I}(m_{I,i}, \sigma_{I,i}) = \text{Reject}$ for some $i \in [\ell(n)]$), then return \perp .
 - * Otherwise, if f_I is still empty (i.e., not \perp), run the procedure **Rewind** detailed below for the instance I .
 - * If, at this point, there is a stored forgery $f_I = (m_I^*, \sigma_I^*)$, then return it and continue executing \mathcal{R} as above. Otherwise, abort the entire execution of \mathcal{B} and return **Fail**.
 - If the response is an invalid signature (i.e., $\text{Ver}_{pk_I}(m_{I,j}, \sigma_{I,j}) = \text{Reject}$), then store $f_I \leftarrow \perp$.
 - Lastly, respond with $\tau || m_{I,j+1}$ and continue the execution of \mathcal{R} .

Rewind procedure:

- Given instance I , for $j \in [\ell(n)]$ let $(v_{open}^j, v_{close}^j)$ denote the slot corresponding to the j^{th} signature query for instance I .
- For each $j \in [\ell(n)]$, “rewind” the slot $(v_{open}^j, v_{close}^j)$ as follows: Let $k' \leftarrow k$, and begin executing \mathcal{R} from the view $v' = v_{open}^j || m_I^*$ as in the main routine, with the following exceptions:
 - When \mathcal{R} begins a new instance of \mathcal{A} , label this instance as instance k' and increment k' . (That is, continue creating new instances, but preserve the counter k in the outer execution for after the rewinding.)
 - When \mathcal{R} attempts to communicate externally with \mathcal{C} , abort the rewinding and continue to the next j .
 - When \mathcal{R} sends an end message for an instance $I' \neq I$ of \mathcal{A} , abort the rewinding and continue to the next j , *unless* \mathcal{R} has not sent responses to $\ell(n)$ signature queries for that instance (in which case respond with \perp).
 - If v' ever contains a message whose transcript contains a response σ_I^* to the query for m_I^* , then, if it is the case that $\text{Ver}_{pk_I}(m_I^*, \sigma_I^*) = \text{Accept}$, store $f_i \leftarrow (m_I^*, \sigma_I^*)$ and end the **Rewind** procedure (i.e., return to the outer execution); otherwise, if $\text{Ver}_{pk_I}(m_I^*, \sigma_I^*) = \text{Reject}$, store nothing to f_I and continue to the next j .

^aNote that in the case where this happens, then \mathcal{B} will fail to output a successful forgery regardless; however, this can only happen with probability negligible in n , and hence will not affect the bound we show.

Figure 4: Formal description of the meta-reduction \mathcal{B} .

bounded by $O(\ell(n)(\ell(n) + r(n))M(n)^2)$, which is still polynomial. The claim follows from defining $t(n)$ accordingly.

(Note that, while we assume without loss of generality that \mathcal{R} will never rewind \mathcal{A} , such rewinding even if it did occur could at most multiply the running time by a polynomial factor since \mathcal{R} is restricted to polynomial time.) \square

Next, to reason about the failure probability of \mathcal{B} (and through it the success probability of $\mathcal{R}^{\mathcal{A}}$), let us define the following experiments:

- Let $\mathbf{Ideal}(1^n)$ denote $[\mathcal{R}^{\mathcal{A}} \leftrightarrow \mathcal{C}](1^n)$ —that is, the experiment where $\mathcal{R}(1^n)$, using the “ideal” adversary $\mathcal{A}(1^n)$ as a black box, communicates with $\mathcal{C}(1^n)$.
 - When we refer to probabilities in the context of this experiment, they are taken over a uniform distribution over random oracles \mathcal{O} (which results in uniformly distributed messages \vec{m}_I and m_I^*) for each instance I of \mathcal{A} started by \mathcal{R} .
 - When we wish to fix the randomness of a particular execution of \mathbf{Ideal} , we will denote this with the notation $\mathbf{Ideal}_{\{\mathcal{O}_I\}_{I \in [M(n)]}, \mathcal{O}_{ext}}(1^n)$, or, for more clarity, $\mathbf{Ideal}_{\{\vec{m}_I, m_I^*\}_{I \in [M(n)]}, \mathcal{O}_{ext}}(1^n)$. \vec{m}_I and m_I^* are the messages and forgery generated for instance I by each oracle \mathcal{O}_I given the (deterministic) prefix; \mathcal{O}_{ext} is a random variable representing the random coins used by \mathcal{R} and \mathcal{C} , containing a number of bits equal to the maximum number of coins needed (which must be polynomially many since \mathcal{R} and \mathcal{C} are polynomial-time). When all \vec{m}_I and m_I^* are fixed, and \mathcal{O}_{ext} is fixed, note that the execution of \mathbf{Ideal} is deterministic for each instance.
- Let $\mathbf{Real}(1^n)$ denote $[\mathcal{B} \leftrightarrow \mathcal{C}](1^n)$ —that is, the “real” experiment where $\mathcal{B}(1^n)$ communicates directly with $\mathcal{C}(1^n)$ by attempting to simulate the interaction between $\mathcal{A}(1^n)$ and $\mathcal{R}(1^n)$ while forwarding any external communications.
 - When we refer to probabilities in the context of this experiment, they are taken over uniformly distributed messages \vec{m}_I and m_I^* for each simulated instance I of \mathcal{A} started by \mathcal{R} in the context of \mathcal{B} .
 - When we wish to fix the randomness of a particular execution of \mathbf{Real} , we will again denote this with the notation $\mathbf{Real}_{\{\vec{m}_I, m_I^*\}_{I \in [M(n)]}, \mathcal{O}_{ext}}(1^n)$, where \vec{m}_I and m_I^* are the messages and forgery for each simulated instance I of \mathcal{A} and \mathcal{O}_{ext} is again a random variable representing the random coins used by \mathcal{R} and \mathcal{C} . When all \vec{m}_I and m_I^* are fixed, and \mathcal{O}_{ext} is fixed, note that the execution of \mathbf{Real} is deterministic for each instance, just as with \mathbf{Ideal} .
 - Furthermore, we may opt to isolate a particular simulated instance k by fixing all randomness *except for* that instance’s; we denote this by $\mathbf{Real}_{\{\vec{m}_I, m_I^*\}_{-k}, \mathcal{O}_{ext}}^*(1^n)$ and note that the probability space in this altered experiment is over uniformly distributed \vec{m}_k and m_k^* . Further note that in experiment \mathbf{Real}^* the execution up to the start of the isolated instance k is deterministic, as is the execution for any choice of \vec{m}_k and m_k^* .
- In all experiments, we will denote the *view*, or *execution*, as the transcript of all messages sent between, and all randomness consumed by, real or simulated machines (i.e., between \mathcal{R} , \mathcal{A} or \mathcal{B} ’s simulation of \mathcal{A} , and \mathcal{C}). We will notate this using just the notation for the experiment, e.g., $\mathbf{Real}(1^n)$.

- In all experiments, we will denote the *result*, or *output*, as either the final output of \mathcal{C} (either Accept or Reject) if \mathcal{C} finishes, or as Fail if \mathcal{C} does not finish (i.e., when \mathcal{B} aborts returning Fail). This will be notated by Output, e.g., $\text{Output}[\text{Real}(1^n)] = \text{Accept}$.

3.3 Analyzing the Meta-Reduction

Using this terminology, we wish to show that $\text{Real}(1^n)$ is identically distributed to $\text{Ideal}(1^n)$ with high probability. To that end, we make the following claim:

Claim 3. For all $n \in \mathbb{N}$:

$$|\Pr[\text{Output}[\text{Real}(1^n)] = \text{Accept}] - \Pr[\text{Output}[\text{Ideal}(1^n)] = \text{Accept}]| \leq \Pr[\text{Output}[\text{Real}(1^n)] = \text{Fail}]$$

Proof. The probability of each possible assignment of the random variables $\{\vec{m}_I, m_I^*\}_{I \in [M(n)]}$ and \mathcal{O}_{ext} occurring is clearly identical between the Real and Ideal experiments due to the choice of uniformly random oracles \mathcal{O} for the Ideal experiment. Thus, it suffices to show the following lemma:

Lemma 2. For all $n \in \mathbb{N}$ and for any assignment of $\{\vec{m}_I, m_I^*\}_{I \in [M(n)]}$ and \mathcal{O}_{ext} such that

$$\text{Output}[\text{Real}_{\{\vec{m}_I, m_I^*\}_{I \in [M(n)]}, \mathcal{O}_{ext}}(1^n)] \neq \text{Fail}$$

it holds that:

$$\text{Ideal}_{\{\vec{m}_I, m_I^*\}_{I \in [M(n)]}, \mathcal{O}_{ext}}(1^n) = \text{Real}_{\{\vec{m}_I, m_I^*\}_{I \in [M(n)]}, \mathcal{O}_{ext}}(1^n)$$

and thus

$$\text{Output}[\text{Ideal}_{\{\vec{m}_I, m_I^*\}_{I \in [M(n)]}, \mathcal{O}_{ext}}(1^n)] = \text{Output}[\text{Real}_{\{\vec{m}_I, m_I^*\}_{I \in [M(n)]}, \mathcal{O}_{ext}}(1^n)]$$

Proof. Intuitively, the lemma follows from the uniqueness property of Π ; for any message m^* , there is only a single possible signature σ^* . Thus, unless the extraction fails for some m^* , the simulation in \mathcal{B} will output the same forgery as in the actual execution of \mathcal{A} .

Formally, denote the experiments $\text{Real}_{\{\vec{m}_I, m_I^*\}_{I \in [M(n)]}, \mathcal{O}_{ext}}(1^n)$ and $\text{Ideal}_{\{\vec{m}_I, m_I^*\}_{I \in [M(n)]}, \mathcal{O}_{ext}}(1^n)$ by Real and Ideal respectively. Because \mathcal{A} 's, \mathcal{C} 's, and \mathcal{R} 's randomness are fixed by assumption, it must be the case that the views of Ideal and Real are identical up to the first forgery request. At this point, if \mathcal{B} in Real does not return Fail (indicating that \mathcal{B} either found a valid forgery or will return \perp due to \mathcal{R} returning an incorrect signature), the forgeries (m_I^*, σ_I^*) returned by \mathcal{A} in Ideal and by \mathcal{B} in Real must also be identical. This holds because (1) the situations in which \mathcal{A} or \mathcal{B} return \perp are the same, and (2) if, given certain randomness, \mathcal{A} does not return \perp and \mathcal{B} does not return Fail, both \mathcal{A} and \mathcal{B} must return a correct signature of an identical message m_I^* , which by uniqueness must also be identical.

From here, we can inductively apply the same logic; if we assume that the first k forgery requests' responses are identical (i.e. that \mathcal{B} has not yet failed), then, for each possible assignment of these responses (which must by assumption be equally likely between \mathcal{A} and \mathcal{B}), the execution with fixed randomness proceeds identically up to the $(k+1)^{\text{st}}$ forgery request; hence, if \mathcal{B} does not return Fail for that request as well, its responses are once again distributed identically in Real to \mathcal{A} 's in Ideal, meaning that the executions up to the $(k+1)^{\text{st}}$ forgery requests' responses must likewise be identical.

So, if we fix $\{\vec{m}_I, m_I^*\}_{I \in [M(n)]}$ and \mathcal{O}_{ext} for both experiments, and if \mathcal{B} does not return Fail, then all messages sent by \mathcal{A} to \mathcal{R} in Ideal must be identical to the simulated (non-rewind) messages from \mathcal{B} to \mathcal{R} in Ideal, ensuring that (since \mathcal{R} and \mathcal{C} are executed identically and with the same randomness in both experiments) the executions will likewise be identical. \square

Thus, by this lemma, the probability that the outputs of the two experiments differ must be at most the probability that some assignment of the random variables occurs which causes $\text{Real}(1^n)$ to return Fail (as in all other cases the views and outputs are identical), as desired. \square

Next, we use this to bound $\mathcal{R}^{\mathcal{A}}$'s success probability by bounding the probability that \mathcal{B} will return Fail for some simulated instance I of \mathcal{A} . Let $M(n)$ be the maximum, over all randomness of \mathcal{A} , \mathcal{C} , and \mathcal{R} , of the number of instances of \mathcal{A} that \mathcal{R} runs to completion (i.e., for which it responds to all $\ell(n)$ queries) during the experiment $\text{Real}(1^n)$. Then we show the following:

Proposition 1. There exists a negligible function $\epsilon(\cdot)$ such that, for all $n \in \mathbb{N}$:

$$\Pr[\text{Output}[\text{Real}(1^n)] = \text{Fail}] \leq M(n) \left(\frac{M(n) + r(n) + 1}{\ell(n) + 1} \right) + \epsilon(n)$$

Proof. We first prove the following claim for any execution $\text{Real}_{\{\vec{m}_I, m_I^*\}_{-k}, \mathcal{O}_{ext}}^*(1^n)$ (i.e., for any fixed setting of all randomness aside from \vec{m}_k and m_k^*), and notice that, since it applies to arbitrarily fixed randomness, it must thus apply over all possible randomness of the experiment $\text{Real}(1^n)$:

Claim 4. There exists a negligible $\nu(\cdot)$ such that, given any experiment $\text{Real}_{\{\vec{m}_I, m_I^*\}_{-k}, \mathcal{O}_{ext}}^*(1^n)$, the probability, over the uniformly chosen messages $m_{k,1}, \dots, m_{k,\ell(n)}, m_k^*$, that the simulated instance k will return Fail is, for all $n \in \mathbb{N}$, at most

$$\frac{M(n) + r(n) + 1}{\ell(n) + 1} + \nu(n)$$

Proof. Let us begin by assuming that other simulated instances (besides k) of \mathcal{A} in the experiment $\text{Real}_{\{\vec{m}_I, m_I^*\}_{-k}, \mathcal{O}_{ext}}^*(1^n)$ will never return Fail (i.e., that they will “magically” produce a correct forgery in the case where they otherwise would return Fail). Clearly, this can only increase the probability that instance k will return Fail by ensuring that the experiment never aborts early.

Now let us consider the messages $\vec{m}_k^* \triangleq (m_{k,1}, \dots, m_{k,\ell(n)}, m_k^*)$ in instance k ; note that by the definition of $\text{Real}_{\{\vec{m}_I, m_I^*\}_{-k}, \mathcal{O}_{ext}}^*(1^n)$ the execution is fully determined by \vec{m}_k^* . Let us also define for $i \in [\ell(n) - 1]$ the “rewound” sequence

$$\rho(\vec{m}_k^*, i) \triangleq (m_{k,1}, \dots, m_{k,i-1}, m_k^*)$$

that is, the rewinding of \vec{m}_k where the message in slot i is replaced by m_k^* to attempt to extract a forgery.

In order for \mathcal{B} to return Fail, one of the following “bad events” must occur for each $i \in [\ell(n)]$:

- $E_1(\rho(\vec{m}_k^*, i))$: \mathcal{R} fails to return a valid signature of m_I^* in the rewinding of the last slot in the sequence (slot i).

- $E_2(\rho(\vec{m}_k^*, i))$: During the rewinding of the last slot in the sequence (slot i), \mathcal{R} asks for a forgery for another instance $k' \neq k$ before returning a signature for m_I^* , or \mathcal{R} requires external communication with \mathcal{C} (which cannot be rewound) before returning a signature for m_I^* .

In addition, for k to fail, the non-rewound execution of the instance must succeed, in that the event $E_1(\vec{m}_{k, \leq i})$ (where \mathcal{R} fails to return a valid signature) *cannot* occur for any prefix $\vec{m}_{k, \leq i} = (m_{k,1}, \dots, m_{k,i})$, where $i \in [\ell(n)]$.

Since, as we have noted, the behavior of k in the execution of $\text{Real}_{\{\vec{m}_I, m_I^*\}_{-k}, \mathcal{O}_{ext}}^*(1^n)$ is fully determined by \vec{m}_k^* , every sequence \vec{m}_k^* will deterministically either result in instance k returning something (either a forgery or \perp) or aborting and returning Fail; we shall refer to the former type of sequence (where k succeeds) as a “good” sequence, and the latter type as a “bad” sequence. To describe the relationship between these “good” and “bad” sequences, we first introduce the following terminology:

For any $k > 0$ and any arbitrary set \vec{m} of k distinct messages in $[2^n]$, let $\Pi_k(\vec{m})$ denote the set of ordered permutations of the elements of \vec{m} . Given a sequence $\pi = (m_1, \dots, m_{k-1}, m^*) \in \Pi_k(\vec{m})$, we let the “rewinding” operator ρ for $i \in [k-1]$ be defined as before—that is:

$$\rho(\pi, i) \triangleq (m_1, \dots, m_{i-1}, m^*)$$

(Note that this is a sequence of length i .) We shall say that a sequence $a = (a_1, \dots, a_{k-1}, a^*) \in \Pi_k(\vec{m})$ *blocks* a sequence $b = (b_1, \dots, b_{k-1}, b^*) \in \Pi_k(\vec{m})$ with respect to some $i \in [k-1]$ if

$$\rho(a, i) = (b_1, \dots, b_i)$$

that is, if a has a rewinding equivalent to a prefix of b . If we wish to denote that a blocks b with respect to a particular i , we shall say that a blocks b *in slot* i . Furthermore, we will say that sequences $a^1, a^2, \dots, a^c \in \Pi_k(\vec{m})$ *c-block* a sequence $b \in \Pi_k(\vec{m})$ if there exist *distinct* $i_1, \dots, i_c \in [k-1]$ such that, for any $j \in [c]$, a^j blocks b in slot i_j .

We next formalize the relationship between the “blocking” property and good/bad sequences that will allow us to use this property to bound the number of bad sequences that may occur. Specifically, we prove the following lemma:

Lemma 3. Any sequence that is $(M(n) + r(n) + 1)$ -blocked by bad sequences must be a good sequence.

Proof. Consider a sequence \vec{m}'_k which is $(M(n) + r(n) + 1)$ -blocked by bad sequences. This means that \vec{m}'_k must have $M(n) + r(n) + 1$ distinct slots i_j for which E_1 or E_2 occurs in its (non-rewound) execution, as the execution is at each of those points identical to a rewinding of one of the bad sequences by the fact that the sequence blocks \vec{m}'_k in slot i_j . However, because at most $M(n)$ end messages (to completed instances of \mathcal{A} , note that others are answered with \perp) and at most $r(n)$ rounds of external communication can occur in any given execution, we observe that E_2 can happen for at most $M(n) + r(n)$ of these slots, and thus that E_1 must happen for at least one slot. In this case, we can deduce that \vec{m}'_k must be a *good* sequence, because it must contain some slot for which \mathcal{R} fails to return a correct response (meaning that \mathcal{B} can successfully emulate \mathcal{A} by returning \perp)—that is, the event $E_1(\vec{m}_{k, \leq i})$ must occur for that slot, which we have previously stated cannot be the case for bad \vec{m}_k^* . \square

Consider, then, a set S of “bad” sequences \vec{m}_k^* which are permutations of any set of $\ell(n) + 1$ distinct messages (i.e., an unordered set containing $\vec{m}_{k,i}$ and m_k^*). The following lemma, combined with Lemma 3, allows us to bound the size of such a set S :

Lemma 4. Let \vec{m} be an arbitrary set of $\ell + 1$ distinct messages in $[2^n]$, and let $S \subset \Pi_{\ell+1}(\vec{m})$ be a set of permutations of \vec{m} . If it is the case that, for some $B \in \mathbb{N}$, any member of $\Pi_{\ell+1}(\vec{m})$ which is $(B + 1)$ -blocked by a subset of S cannot itself lie in S , then $|S| \leq (B + 1)\ell!$

Proof. We begin with the following crucial claim:

Subclaim 1. No member of $\Pi_{\ell+1}(\vec{m})$ is $(B + 2)$ -blocked by a subset of S .

Proof. Assume for the sake of contradiction that there exists some $\pi \in \Pi_{\ell+1}(\vec{m})$, $B + 2$ sequences $\pi^1, \dots, \pi^{B+2} \in S$, and $B + 2$ distinct integers $i_1, \dots, i_{B+2} \in [\ell]$ such that each partial sequence $\rho(\pi^j, i_j)$ is equivalent to the first i_j elements of π .

Assume without loss of generality that the integers i_j are in strictly ascending order. Consider the last sequence $\pi^{B+2} = (\pi_1^{B+2}, \dots, \pi_*^{B+2})$; we shall show that π^{B+2} is $(M + 1)$ -blocked by sequences in S , leading to a contradiction because by definition no element of S can be $(M + 1)$ -blocked by other members of S .

We know that, since by assumption the first i_{B+2} elements of π are equivalent to

$$\rho(\pi^{B+2}, i_{B+2}) = (\pi_1^{B+2}, \dots, \pi_{i_{B+2}-1}^{B+2}, \pi_*^{B+2})$$

then the first $i_{B+2} - 1$ elements of π^{B+2} and π must be identical. However, notice that, for any $j < B + 2$, we have that $\rho(\pi^j, i_j)$ is identical to π in the first i_j elements, which, since by assumption $i_j \leq i_{B+2} - 1$, also indicates that $\rho(\pi^j, i_j)$ is identical to π^{B+2} in the first i_j elements.

This in turn implies that π^{B+2} is $(B + 1)$ -blocked by the sequences $\pi^1, \dots, \pi^{B+1} \in S$, contradicting that $\pi^{B+2} \in S$ by the requisite property of S . \square

So, we know that any member of S can be at most B -blocked by a subset of S , while any non-member can be at most $(B + 1)$ -blocked by a subset of S . We will combine this fact (an effective upper bound on the number of blocked sequences) with the subsequent claim (a respective lower bound) to derive our final bound on $|S|$.

Subclaim 2. For each $i \in [\ell]$, there exist $|S|$ distinct sequences blocked in slot i by sequences in S .

Proof. Beginning with $i = 1$, we observe that sequences with at least $|S|/\ell!$ different last elements m^* must occur in S (as there are only $\ell!$ sequences with any given last element). Furthermore, any sequence in S with a certain last element m^* must block in slot 1 a total of $\ell!$ different sequences (i.e., anything beginning with m^*), and different m^* will produce disjoint sets of sequences blocked. Thus, we conclude that the sequences in S will block in slot 1 at least $(|S|/\ell!)\ell! = |S|$ distinct sequences.

For the remaining slots $i > 1$, we can apply the same logic to the distinct arrangements of the elements (m_1, \dots, m_{i-1}) and m^* . Among the sequences in S there must be a minimum of $|S|/(\ell + 1 - i)!$ such arrangements (since, given a fixed $(m_1, \dots, m_{i-1}, m^*)$, there are $(\ell + 1 - i)!$ sequences possible), and sequences with each arrangement will block in slot i a total of $(\ell + 1 - i)!$ distinct sequences (i.e., any sequence beginning with $(m_1, \dots, m_{i-1}, m^*)$). Hence, the sequences in S will block in slot i at least $(|S|/(\ell + 1 - i)!)(\ell + 1 - i)! = |S|$ distinct sequences. \square

In total, we notice that at least $|S|$ distinct sequences are blocked in slot i for any $i \in [\ell]$, and so there are at least $|S|\ell$ distinct pairs (π, i) such that the sequence π is blocked in slot i by sequences in S . Furthermore, we recall that the sequences in S are each blocked in slot i by sequences in S for at most B different i , while the remaining $(\ell + 1)! - |S|$ elements are each blocked in slot i by sequences in S for at most $B + 1$ different i . This provides an upper bound of $B|S| + ((\ell + 1)! - |S|)(B + 1)$ on the number of “blocking” pairs (π, i) . We lastly combine these lower and upper bounds (noting that, if the lower bound exceeded the upper bound, there would be a contradiction) to bound $|S|$:

$$|S|\ell \leq B|S| + ((\ell + 1)! - |S|)(B + 1) = B(\ell + 1)! + (\ell + 1)! - |S|$$

$$|S|(\ell + 1) \leq (B + 1)(\ell + 1)!$$

$$|S| \leq (B + 1)\ell!$$

□

Recall that, if S is the set of all bad sequences which are permutations of some set \vec{m}^* of $\ell(n) + 1$ distinct messages, we have by Lemma 3 that any sequence which is $(M(n) + r(n) + 1)$ -blocked by bad sequences in S must be good and thus lie outside of S . Hence, by Lemma 4, S has size at most

$$(M(n) + r(n) + 1) (\ell(n))!$$

Given any set of $\ell(n) + 1$ distinct messages, then, the above applies to show that at most an

$$\frac{(M(n) + r(n) + 1) (\ell(n))!}{(\ell(n) + 1)!} = \frac{M(n) + r(n) + 1}{\ell(n) + 1}$$

fraction of the sequences defined by the permutations of this set can be bad, and the remainder must be good. Applying this to every possible set of $\ell(n) + 1$ distinct messages, we get that at most the same fraction of *all* sequences of distinct messages can be bad. While the property that messages are distinct is not necessarily guaranteed, we note that the probability that they are not over uniformly randomly chosen messages is negligible—specifically, we notice that the probability of any pair of elements colliding is 2^{-n} , and so, by the union bound, the probability that any of the $\frac{\ell(n)(\ell(n)+1)}{2}$ pairs of elements can collide is smaller than $\nu(n) \triangleq \ell(n)^2 2^{-n}$ (which is negligible in n because $\ell(\cdot)$ is polynomial).

Hence, the chance that a sequence chosen at random is bad, which by definition is equal to the probability that a randomly chosen sequence of messages $\vec{m}_k^* = (m_{k,1}, \dots, m_{k,\ell(n)}, m_k^*)$ will result in instance k returning Fail, can be at most the fraction of sequences without repeated elements which are bad plus the fraction of sequences with repeated elements, or:

$$\frac{M(n) + r(n) + 1}{\ell(n) + 1} + \nu(n)$$

as desired. □

Recall that, because this result holds for any execution of $\text{Real}_{\{\vec{m}_I, m_I^*\}_{-k}, \mathcal{O}_{ext}}^*(1^n)$, it must also hold over a *random* such execution—i.e., the actual execution $\text{Real}(1^n)$ of \mathcal{B} , where the messages for all instances are chosen uniformly at random. Furthermore, it holds for any instance k of \mathcal{A} .

To conclude the proof of the proposition, by Claim 1 we know that \mathcal{R} must send a total of at least $\ell(n)$ messages to each instance of \mathcal{A} in order for the failure probability of \mathcal{B} to emulate that

instance to be more than negligible; if not, then \mathcal{B} will always respond with \perp (having not received $\ell(n)$ signature query responses), while the claim shows that \mathcal{A} will return \perp with all but negligible probability. Recall that $M(n)$ is an upper bound to the number of instances of \mathcal{A} to which \mathcal{R} sends $\ell(n)$ messages. By Claim 4 and the union bound over all $M(n)$ completed instances of \mathcal{A} , the failure probability of \mathcal{B} for those instances is at most

$$M(n) \left(\frac{M(n) + r(n) + 1}{\ell(n) + 1} + \nu(n) \right)$$

for negligible $\nu(\cdot)$, and the failure probability for all other instances (of which there can only be a polynomial number by the time constraint on \mathcal{R}) is negligible by the union bound applied to Claim 1. Hence the overall failure probability of **Real** (i.e., the execution of \mathcal{B}) must be bounded above by

$$\begin{aligned} \Pr[\text{Output}[\text{Real}(1^n)] = \text{Fail}] &\leq M(n) \left(\frac{M(n) + r(n) + 1}{\ell(n) + 1} + \nu(n) \right) + \nu'(n) \\ &< M(n) \left(\frac{M(n) + r(n) + 1}{\ell(n) + 1} \right) + \epsilon(n) \end{aligned}$$

for some negligible functions $\nu'(\cdot)$ and $\epsilon(\cdot)$. □

Completing the Proof of Lemma 1. Finally, in order to bound the security loss, we note that, if the probability $\text{Success}_{\mathcal{R}\mathcal{A}}(n)$ (which specifically is by definition a lower bound to the probability $\Pr[\text{Output}[\text{Ideal}(1^n)] = \text{Accept}] - t(n)$; recall that $t(\cdot)$ is the threshold for the underlying assumption \mathcal{C}) is non-negligibly greater than the failure probability of **Real**, there exists a polynomial $p(\cdot)$ such that:

$$\Pr[\text{Output}[\text{Ideal}(1^n)] = \text{Accept}] - t(n) \geq \Pr[\text{Output}[\text{Real}(1^n)] = \text{Fail}] + \frac{1}{p(n)}$$

But, by Claim 3, this would imply that

$$\Pr[\text{Output}[\text{Real}(1^n)] = \text{Accept}] - t(n) \geq \frac{1}{p(n)}$$

that is, that \mathcal{B} breaks the security of $(\mathcal{C}, t(\cdot))$. So, by Proposition 1, unless \mathcal{B} breaks the security of $(\mathcal{C}, t(\cdot))$, the above cannot be the case—that is, there must exist negligible $\epsilon(\cdot), \epsilon'(\cdot)$ such that, for sufficiently large n :

$$\begin{aligned} \text{Success}_{\mathcal{R}\mathcal{A}}(n) &\leq \Pr[\text{Output}[\text{Real}(1^n)] = \text{Fail}] + \epsilon'(n) \\ &< M(n) \left(\frac{M(n) + r(n) + 1}{\ell(n) + 1} \right) + (\epsilon(n) + \epsilon'(n)) < M(n) \left(\frac{M(n) + r(n) + 1}{\ell(n)} \right) \end{aligned}$$

Of course, $\text{Success}_{\mathcal{R}\mathcal{A}}(n)$, being a probability, is also trivially bounded above by 1. Furthermore, by the definition of $M(n)$, we know that $\text{Query}_{\mathcal{R}\mathcal{A}}(n) \geq M(n)\ell(n)$. Lastly, we consider two cases to derive our bound on the security loss.

Case 1. If $M(n) \geq \sqrt{\ell(n)} - (r(n) + 1)$, then:

$$\begin{aligned} \lambda_{\mathcal{R}}(n) &\geq \frac{\text{Success}_{\mathcal{A}}(n)}{\text{Success}_{\mathcal{R}\mathcal{A}}(n)} \frac{\text{Query}_{\mathcal{R}\mathcal{A}}(n)}{\text{Query}_{\mathcal{A}}(n)} \geq \frac{1}{1} \frac{M(n)\ell(n)}{\ell(n)} \\ &= M(n) \geq \sqrt{\ell(n)} - (r(n) + 1) \end{aligned}$$

Case 2. Otherwise, if $M(n) < \sqrt{\ell(n)} - (r(n) + 1)$ we have $M(n) + r(n) + 1 < \sqrt{\ell(n)}$, and so:

$$\begin{aligned} \lambda_{\mathcal{R}}(n) &\geq \frac{\text{Success}_{\mathcal{A}}(n)}{\text{Success}_{\mathcal{R}^{\mathcal{A}}}(n)} \frac{\text{Query}_{\mathcal{R}^{\mathcal{A}}}(n)}{\text{Query}_{\mathcal{A}}(n)} \geq \frac{1}{M(n)} \frac{M(n)\ell(n)}{\binom{M(n)+r(n)+1}{\ell(n)}} \\ &= \frac{\ell(n)}{M(n) + r(n) + 1} > \frac{\ell(n)}{\sqrt{\ell(n)}} = \sqrt{\ell(n)} \end{aligned}$$

Either way, we observe that $\lambda_{\mathcal{R}}(n) \geq \sqrt{\ell(n)} - (r(n) + 1)$, thus completing the proof of both Lemma 1 and Theorem 2.

4 Related Work

Lastly, we present a brief survey of, and comparison of our results with, related work concerning unique signatures and black-box impossibility results.

Definitions and techniques. The notion of security loss dates back to literature on “security-preserving reductions”, formalized originally by Luby in 1996 [Lub96]; early works on security-preserving reductions include Naor and Reingold’s efficient constructions of pseudo-random functions in [NR97, NR99], and there has been a considerable amount of more recent research concerning efficient and tight security reductions [BR96, BBM00, BR09, BJLS16]. Black-box separations, or techniques for proving the impossibility of certain black-box reductions, date back to the work of Impagliazzo and Rudich in 1989 [IR89]; the meta-reduction framework (which we employ here) was originally proposed by Boneh and Venkatesan in 1998 [BV98], and has since been applied to prove impossibilities for “one-more” problems [BMV08], witness-hiding of interactive proofs [HRS09], blind signatures [FS10], structure-preserving signatures [AGO11], and CCA-security of certain signature schemes [BFW16]. Lastly, Pass, in [Pas11], provides a generalization of several of the above results (analogous in some ways to our generalization of Coron), drawing the connection with concurrent zero-knowledge upon which we build here and presenting a powerful meta-reduction technique used to rule out a wide class of reductions (specifically, reductions to any standard bounded-round assumption from sequential witness-hiding of any protocol which is constant-round, unique-witness, and computationally special-sound).

Impossibilities for Unique Signatures. Focusing in particular on black-box impossibilities for unique signatures, literature dates back to the original result of Coron in 2002 [Cor02]. Since then, there have been many refinements and generalizations of this elegant result. For instance, Hofheinz et al. [HJK12] demonstrate that the bound also extends to rerandomizable signatures. Kakvi and Kiltz [KK12] demonstrate a subtlety in Coron’s argument which allows for more efficient reductions for full-domain hash signatures (and in general for signatures where uniqueness is not certifiable), amending Coron’s result while also demonstrating such a reduction. Bader et al. [BJLS16] refine Coron’s result to demonstrate a modernized version of the bound that applies to efficiently rerandomizable primitives and to decisional complexity assumptions. Recent works by Wang et al. [WMHT18] have applied analysis similar to Coron’s to rule out *memory-tight* reductions from unique signatures to general security assumptions in partially restricted settings.

Lastly, there are impossibility results unrelated to Coron’s bound which indirectly demonstrate complete impossibility for basing unique signatures on specific cryptographic primitives using *black-box constructions*: Brakerski et al. [BGRV09] show that black-box constructions of verifiable random functions (which imply unique signatures) from one-way permutations are impossible; subsequently, Fiore and Schröder [FS12] demonstrate that black-box constructions from trapdoor permutations are likewise impossible. In contrast, note that we here (just like Coron) consider arbitrary (potentially non-black-box) *constructions*—we only restrict to black-box *reductions*.

Constructions of Unique Signatures. Of course, all of the upper bounds for reductions based on generic assumptions, including those which we present here, merely rule out *efficient* (i.e., linear-preserving) black-box reductions; indeed, there are many well-known constructions of unique signature schemes from standard assumptions. Unique signature schemes, along with the related notion of verifiable random functions, were first presented by Micali et al. in 1999 [MVR99]; the authors defined these primitives and also proved that they could be constructed from the RSA assumption. Since then, there have been several black-box constructions of both unique and rerandomizable signatures (e.g. [Lys02, Wat05]) and, more recently, of verifiable random functions (e.g. [BGRV09, Jag15, HJ16]) from various assumptions. More recently still, Guo et al. [GCS⁺17] devised a non-black-box reduction proving security of a unique signature scheme in the random oracle model; while such a reduction is not subject to Coron’s impossibility or variants thereof, it is still not linear-preserving (since its security loss depends on the number of adversarial oracle queries).

Tightly Secure Constructions of Non-Unique Signatures. We lastly remark on the existence of several constructions of non-unique and non-rerandomizable digital signature schemes, as well as of identity-based encryption (IBE) schemes (which, as described in [BF01], imply signature schemes with the same security loss via a transformation devised by Naor), that exhibit tight or linear-preserving security reductions. Bellare and Rogaway [BIN97] and, later, Bernstein [Ber08] and Abdalla et al. [AFLT12], demonstrated tightly secure reductions for digital signature schemes in the random-oracle model. Hofheinz and Jager [HJ12] constructed a signature scheme which is tightly secure based only on the Decision Linear assumption. Chen and Wee [CW13] proposed the first IBE scheme with a linear-preserving reduction based on standard assumptions, and Chen et al. [CGW17] later proposed another IBE scheme which, while not technically tight or linear-preserving, is tighter than any previous construction in a concrete sense (as its security loss is proportional to the logarithm of the number of secret keys and challenge ciphertexts revealed to the adversary).

References

- [AFLT12] Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-Secure Signatures from Lossy Identification Schemes. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 572–590, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [AGO11] Masayuki Abe, Jens Groth, and Miyako Ohkubo. Separating Short Structure-Preserving Signatures from Non-interactive Assumptions. In *Proceedings of the 17th In-*

- ternational Conference on The Theory and Application of Cryptology and Information Security, ASIACRYPT'11*, pages 628–646, Berlin, Heidelberg, 2011. Springer-Verlag.
- [BJS16] Christoph Bader, Tibor Jager, Yong Li, and Sven Schäge. On the Impossibility of Tight Cryptographic Reductions. In *Proceedings, Part II, of the 35th Annual International Conference on Advances in Cryptology — EUROCRYPT 2016 - Volume 9666*, pages 273–304, New York, NY, USA, 2016. Springer-Verlag New York, Inc.
- [BBF13] Paul Baecher, Christina Brzuska, and Marc Fischlin. Notions of Black-Box Reductions, Revisited. In *Part I of the Proceedings of the 19th International Conference on Advances in Cryptology - ASIACRYPT 2013 - Volume 8269*, pages 296–315, Berlin, Heidelberg, 2013. Springer-Verlag.
- [BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements. In Bart Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, pages 259–274, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.
- [BIN97] Mihir Bellare, Russell Impagliazzo, and Moni Naor. Does Parallel Repetition Lower the Error in Computationally Sound Protocols? In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science, FOCS '97*, pages 374–, Washington, DC, USA, 1997. IEEE Computer Society.
- [BR09] Mihir Bellare and Thomas Ristenpart. Simulation without the Artificial Abort: Simplified Proof and Improved Concrete Security for Waters' IBE Scheme. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, pages 407–424, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [BR96] Mihir Bellare and Phillip Rogaway. The Exact Security of Digital Signatures—How to Sign with RSA and Rabin. In *Proceedings of the 15th Annual International Conference on Theory and Application of Cryptographic Techniques, EUROCRYPT'96*, pages 399–416, Berlin, Heidelberg, 1996. Springer-Verlag.
- [BFW16] David Bernhard, Marc Fischlin, and Bogdan Warinschi. On the Hardness of Proving CCA-Security of Signed ElGamal. In *Proceedings, Part I, of the 19th IACR International Conference on Public-Key Cryptography — PKC 2016 - Volume 9614*, pages 47–69, Berlin, Heidelberg, 2016. Springer-Verlag.
- [Ber08] Daniel J. Bernstein. Proving Tight Security for Rabin-Williams Signatures. In Nigel Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, pages 70–87, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [BF01] Dan Boneh and Matt Franklin. Identity-Based Encryption from the Weil Pairing. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, pages 213–229, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
- [BV98] Dan Boneh and Ramarathnam Venkatesan. Breaking RSA may not be equivalent to factoring. In Kaisa Nyberg, editor, *Advances in Cryptology — EUROCRYPT'98*, pages 59–71, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

- [BGRV09] Zvika Brakerski, Shafi Goldwasser, Guy N. Rothblum, and Vinod Vaikuntanathan. Weak Verifiable Random Functions. In *Proceedings of the 6th International Conference on Theory of Cryptography*, TCC '09, pages 558–576, Berlin, Heidelberg, 2009. Springer-Verlag.
- [BMV08] Emmanuel Bresson, Jean Monnerat, and Damien Vergnaud. Separation Results on the “One-More” Computational Problems. In Tal Malkin, editor, *Topics in Cryptology – CT-RSA 2008*, pages 71–87, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [CLP10] Ran Canetti, Huijia Lin, and Rafael Pass. Adaptive Hardness and Composable Security in the Plain Model from Standard Assumptions. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 541–550, Oct 2010.
- [CGW17] Jie Chen, Junqing Gong, and Jian Weng. Tightly Secure IBE Under Constant-Size Master Public Key. In Serge Fehr, editor, *Public-Key Cryptography – PKC 2017*, pages 207–231, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.
- [CW13] Jie Chen and Hoeteck Wee. Fully, (Almost) Tightly Secure IBE and Dual System Groups. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, pages 435–460, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [Cor02] Jean-Sébastien Coron. Optimal Security Proofs for PSS and Other Signature Schemes. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT '02, pages 272–287, London, UK, UK, 2002. Springer-Verlag.
- [DGS09] Yi Deng, Vipul Goyal, and Amit Sahai. Resolving the Simultaneous Resettability Conjecture and a New Non-Black-Box Simulation Strategy. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '09, pages 251–260, Washington, DC, USA, 2009. IEEE Computer Society.
- [DH06] Witfield Diffie and Martin Hellman. New Directions in Cryptography. *IEEE Trans. Inf. Theor.*, 22(6):644–654, September 2006.
- [DNS98] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent Zero-Knowledge. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 409–418, New York, NY, USA, 1998. ACM.
- [FS12] Dario Fiore and Dominique Schröder. Uniqueness is a Different Story: Impossibility of Verifiable Random Functions from Trapdoor Permutations. In *Proceedings of the 9th International Conference on Theory of Cryptography*, TCC'12, pages 636–653, Berlin, Heidelberg, 2012. Springer-Verlag.
- [FS10] Marc Fischlin and Dominique Schröder. On the Impossibility of Three-move Blind Signature Schemes. In *Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'10, pages 197–215, Berlin, Heidelberg, 2010. Springer-Verlag.

- [GHM⁺17] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles, SOSP '17*, pages 51–68, New York, NY, USA, 2017. ACM.
- [GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing, STOC '85*, pages 291–304, New York, NY, USA, 1985. ACM.
- [GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM J. Comput.*, 17(2):281–308, April 1988.
- [GCS⁺17] Fuchun Guo, Rongmao Chen, Willy Susilo, Jianchang Lai, Guomin Yang, and Yi Mu. Optimal Security Reductions for Unique Signatures: Bypassing Impossibilities with a Counterexample. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017: 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20–24, 2017, Proceedings, Part II*, pages 517–547, Cham, 2017. Springer International Publishing.
- [HRS09] Iftach Haitner, Alon Rosen, and Ronen Shaltiel. On the (Im)possibility of Arthur-Merlin Witness Hiding Protocols. In Omer Reingold, editor, *Theory of Cryptography*, pages 220–237, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [HJ12] Dennis Hofheinz and Tibor Jager. Tightly Secure Signatures and Public-Key Encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, pages 590–607, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [HJ16] Dennis Hofheinz and Tibor Jager. Verifiable Random Functions from Standard Assumptions. In *Proceedings, Part I, of the 13th International Conference on Theory of Cryptography - Volume 9562, TCC 2016-A*, pages 336–362, New York, NY, USA, 2016. Springer-Verlag New York, Inc.
- [HJK12] Dennis Hofheinz, Tibor Jager, and Edward Knapp. Waters Signatures with Optimal Security Reduction. In *Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography, PKC'12*, pages 66–83, Berlin, Heidelberg, 2012. Springer-Verlag.
- [IR89] Russell Impagliazzo and Steven Rudich. Limits on the Provable Consequences of One-way Permutations. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing, STOC '89*, pages 44–61, New York, NY, USA, 1989. ACM.
- [Jag15] Tibor Jager. Verifiable Random Functions from Weaker Assumptions. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography: 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23–25, 2015, Proceedings, Part II*, pages 121–143, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
- [KK12] Saqib A. Kakvi and Eike Kiltz. Optimal Security Proofs for Full Domain Hash, Revisited. In *Proceedings of the 31st Annual International Conference on Theory and*

- Applications of Cryptographic Techniques*, EUROCRYPT'12, pages 537–553, Berlin, Heidelberg, 2012. Springer-Verlag.
- [Lin03] Yehuda Lindell. Bounded-Concurrent Secure Two-party Computation Without Setup Assumptions. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 683–692, New York, NY, USA, 2003. ACM.
- [Lub96] Michael Luby. *Pseudorandomness and Cryptographic Applications*. Princeton University Press, Jan. 1996.
- [Lys02] Anna Lysyanskaya. Unique Signatures and Verifiable Random Functions from the DH-DDH Separation. In *Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '02, pages 597–612, London, UK, UK, 2002. Springer-Verlag.
- [MVR99] Silvio Micali, Salil Vadhan, and Michael Rabin. Verifiable Random Functions. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS '99, pages 120–, Washington, DC, USA, 1999. IEEE Computer Society.
- [Nao03] Moni Naor. On Cryptographic Assumptions and Challenges. *Advances in Cryptology - CRYPTO 2003*, pages 96–109, 2003.
- [NR97] Moni Naor and Omer Reingold. Number-Theoretic Constructions of Efficient Pseudo-Random Functions. *J. ACM*, 51:231–262, 1997.
- [NR99] Moni Naor and Omer Reingold. Synthesizers and Their Application to the Parallel Construction of Pseudo-Random Functions. *J. Comput. Syst. Sci.*, 58(2):336–375, April 1999.
- [Pas11] Rafael Pass. Limits of Provable Security from Standard Assumptions. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 109–118, New York, NY, USA, 2011. ACM.
- [PS17] Rafael Pass and Elaine Shi. The Sleepy Model of Consensus. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 380–409, Cham, 2017. Springer International Publishing.
- [PTV14] Rafael Pass, Wei-Lung Tseng, and Muthuramakrishnan Venkatasubramanian. Concurrent Zero Knowledge, Revisited. *J. Cryptol.*, 27(1):45–66, January 2014.
- [PV08] Rafael Pass and Muthuramakrishnan Venkatasubramanian. On Constant-Round Concurrent Zero-Knowledge. In *Proceedings of the Fifth International Conference on Theory of Cryptography*, TCC'08, pages 553–570, 2008.
- [PRS02] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent Zero Knowledge with Logarithmic Round-Complexity. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, FOCS '02, pages 366–375, Washington, DC, USA, 2002. IEEE Computer Society.
- [Rab79] Michael Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, Cambridge, MA, USA, 1979.

- [RK99] Ransom Richardson and Joe Kilian. On the Concurrent Composition of Zero-knowledge Proofs. In *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT’99, pages 415–431, Berlin, Heidelberg, 1999. Springer-Verlag.
- [RSA78] Ronald Rivest, Adi Shamir, and Leonard Adleman. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.
- [Sha78] Adi Shamir. A fast signature scheme. Technical report, Cambridge, MA, USA, 1978.
- [WMHT18] Yuyu Wang, Takahiro Matsuda, Goichiro Hanaoka, and Keisuke Tanaka. Memory Lower Bounds of Reductions Revisited. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*, pages 61–90, 2018.
- [Wat05] Brent Waters. Efficient Identity-based Encryption Without Random Oracles. In *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT’05, pages 114–127, Berlin, Heidelberg, 2005. Springer-Verlag.

A A Stronger Bound for Statically Scheduled Reductions

As a high-level demonstration of our proof techniques, we present a brief simplified version of our argument assuming static scheduling (i.e., that \mathcal{R} queries instances of \mathcal{A} in the same order irrespective of the messages sent by \mathcal{A}) which in fact achieves a security loss bound asymptotically equivalent to the original result of Coron in a more general setting. We emphasize that our actual proof, as it must deal with dynamic scheduling, is significantly more complicated.

Assume that $M(n)$ instances of \mathcal{A} are run to completion by \mathcal{R} , and each instance makes at most $\ell(n)$ signature queries; in that case, in any one instance of \mathcal{A} , there can be at most $M(n)$ queries for which \mathcal{R} requires a forgery for some other instance of \mathcal{A} before its response, and an additional $r(n)$ which can contain (non-rewindable) external communication between \mathcal{R} and \mathcal{C} before \mathcal{R} ’s response. As such, the minimum possible number of queries in some instance of \mathcal{A} which are “rewindable” in that they do not have one of these occurrences is

$$\ell^*(n) \triangleq \ell(n) - (M(n) + r(n))$$

(which for brevity we shall abbreviate as ℓ^* , and similarly for ℓ , m , and r). To analyze the probability of \mathcal{B} failing to emulate \mathcal{A} , then, we must consider the probability with which \mathcal{R} will respond incorrectly to all of the rewindings of the ℓ^* rewindable queries while responding correctly to all of \mathcal{B} ’s original queries.

As we restrict to static scheduling, we can bound the failure probability of \mathcal{B} by noticing an important relationship between sequences of messages for which \mathcal{B} fails to emulate \mathcal{A} correctly and sequences for which \mathcal{B} succeeds in emulating \mathcal{A} . In particular, consider a context where we (arbitrarily) fix all randomness aside from the messages and forgery generated by a certain specific

instance of \mathcal{A}^5 , and assume we have some sequence $(m_1, \dots, m_\ell; m^*)$, defined by the messages (m_1, \dots, m_ℓ) and forgery message m^* for that instance, that is “bad” in the sense that \mathcal{B} will fail to emulate \mathcal{A} due to every rewinding failing to extract a forgery for m^* yet every (non-rewound) query receiving a valid response from \mathcal{R} . In that case, given some message m_i whose query is rewindable (in the sense that, given the static schedule, it does not contain a forgery request or external communication), we can deduce that running the same instance using the queries $(m_1, \dots, m_{i-1}, m^*, m_{i+1}, \dots, m_\ell)$, as the execution is identical to one of the rewindings of \mathcal{B} which by assumption does not receive a valid signature for m^* , must result in \mathcal{R} failing to respond to m^* and thus \mathcal{A} (and likewise \mathcal{B}) returning \perp . However, in this latter case, we notice that \mathcal{B} in fact *will* successfully emulate \mathcal{A} , and so the latter sequence of messages cannot be “bad” as defined above.

In other words, for each sequence $(m_1, \dots, m_\ell; m^*)$ which is “bad” in that it causes \mathcal{B} to return \perp while \mathcal{A} would return a forgery, the ℓ^* sequences given by $(m_1, \dots, m_{i-1}, m^*, m_{i+1}, \dots, m_\ell; m_i)$ for each rewindable query i must be “good” in that they cause \mathcal{B} to successfully emulate \mathcal{A} (by both returning \perp). Furthermore, note that each of the “good” sequences we consider are mapped *uniquely* to the “bad” sequence from which they originate by switching the forgery message (i.e., m_i) with the *first* query where \mathcal{R} fails to respond. Note that this query *must* be m^* , since, because $(m_1, \dots, m_\ell; m^*)$ is “bad”, the non-rewound queries (m_1, \dots, m_ℓ) receive a valid response, and so the prefix (m_1, \dots, m_{i-1}) must likewise receive valid responses to all queries, and yet we know that placing m^* in the i^{th} query must fail to procure a correct response to m^* .

Hence, we notice that each “bad” sequence implies that, out of itself and ℓ^* of its rewindings (a total of $\ell^* + 1$ sequences), ℓ^* of those must be “good” (and that these sets of implied “good” sequences are disjoint between different “bad” sequences); hence, at most an

$$\frac{1}{\ell^* + 1} = \frac{1}{\ell - (M + r) + 1}$$

fraction of *all* sequences can be bad, which by definition corresponds precisely to the probability that \mathcal{B} fails to emulate that particular instance of \mathcal{A} taken over a randomly chosen sequence of messages and forgery message. (And, as this bound holds given any fixed external randomness, it must hold in general when the randomness is not fixed.) Applying the union bound over M different instances gives us an overall failure probability bound for \mathcal{B} of

$$\frac{M}{\ell - (M + r) + 1}$$

To bound the security loss, we must of course consider the success probability of $\mathcal{R}^{\mathcal{A}}$ rather than the success probability of \mathcal{B} . However, we know by the security assumption $(\mathcal{C}, t(\cdot))$ that \mathcal{B} can break \mathcal{C} with probability at most negligible in n (that is, it causes \mathcal{C} to accept with probability no greater than $t(n) + \nu(n)$ for some negligible $\nu(\cdot)$), and, since $\mathcal{R}^{\mathcal{A}}$ and \mathcal{B} are identically distributed with all but $\frac{M}{\ell - (M + r) + 1}$ probability, this means that $\mathcal{R}^{\mathcal{A}}$ can only break \mathcal{C} with probability

$$\text{Success}_{\mathcal{R}^{\mathcal{A}}}(n) \leq \frac{M}{\ell - (M + r) + 1} + \nu(n) < \frac{M}{\ell - (M + r)}$$

(for sufficiently large n). We can separate into two cases from here.

⁵We shall, as in the main result, show that our failure probability bound holds for *any* setting of such external randomness.

Case 1: If $M < \ell/2 - r$, then the security loss is given by

$$\lambda_{\mathcal{R}}(n) \geq \frac{\text{Success}_{\mathcal{A}}(n)}{\text{Success}_{\mathcal{R}\mathcal{A}}(n)} \frac{\text{Query}_{\mathcal{R}\mathcal{A}}(n)}{\text{Query}_{\mathcal{A}}(n)} \geq \frac{\ell - (M + r)}{M} \left(\frac{M\ell}{\ell} \right) = \ell - (M + r) \geq \frac{\ell}{2}$$

(notice that $\text{Success}_{\mathcal{A}}(n) = 1$ and \mathcal{R} must send ℓ messages to each instance of \mathcal{A}).

Case 2: Otherwise, if $M \geq \ell/2 - r$, we can observe that, even without regarding the failure probability of \mathcal{B} (i.e., only comparing the running times of \mathcal{R} and \mathcal{A}), we obtain:

$$\lambda_{\mathcal{R}}(n) > \frac{1}{1} \left(\frac{\left(\frac{\ell}{2} - r\right) \ell}{\ell} \right) = \frac{\ell}{2} - r$$

Either way, we notice that $\lambda_{\mathcal{R}}(n) \geq \frac{\ell(n)}{2} - r(n)$.

Thus, we have demonstrated a basic bound on the security loss of such a reduction \mathcal{R} . And, assuming for the sake of contradiction that \mathcal{R} is linear-preserving with $\lambda_{\mathcal{R}}(n) \leq p(n)$ for some polynomial $p(\cdot)$, setting $\ell(n) > 2(p(n) + r(n))$ (since the adversary \mathcal{A} may make any polynomial number of signature queries) gives $\lambda_{\mathcal{R}}(n) > p(n)$, which is a contradiction.

B Defining Security Loss using Queries

Here we prove that our definition of security loss using the number of queries instead overall running time is in fact without loss of generality; in particular, we can always construct an adversary \mathcal{A}' where the maximum running time is proportional to the maximum number of queries, showing that the query-based security loss is a lower bound for the time-based security loss. As a result of this, our results, while proven using our query-based definition, apply to time-based definitions of security loss as well.

Lemma 5. For any reduction \mathcal{R} which breaks some assumption $(\mathcal{C}, t(\cdot))$ and any deterministic adversary \mathcal{A} , there exists a deterministic adversary \mathcal{A}' such that, letting $\text{Time}_{\mathcal{A}'}(n)$ and $\text{Time}_{\mathcal{R}\mathcal{A}'}(n)$ denote the worst-case running time of \mathcal{A}' and $\mathcal{R}\mathcal{A}'$, respectively:

$$\frac{\text{Success}_{\mathcal{A}'}(n)}{\text{Success}_{\mathcal{R}\mathcal{A}'}(n)} \frac{\text{Time}_{\mathcal{R}\mathcal{A}'}(n)}{\text{Time}_{\mathcal{A}'}(n)} \geq \frac{\text{Success}_{\mathcal{A}}(n)}{\text{Success}_{\mathcal{R}\mathcal{A}}(n)} \frac{\text{Query}_{\mathcal{R}\mathcal{A}}(n)}{\text{Query}_{\mathcal{A}}(n)}$$

Proof. Given some \mathcal{A} , let $\tau(n)$ denote for each $n \in \mathbb{N}$ the maximum time taken by \mathcal{A} to send a message after receiving a message from \mathcal{R} during any experiment $[\mathcal{R}^{\mathcal{A}} \leftrightarrow \mathcal{C}](1^n)$ —that is, the maximum is taken over all randomness of \mathcal{R} , \mathcal{A} , and \mathcal{C} .

Let \mathcal{A}' be an adversary that proceeds identically to \mathcal{A} , except that, on input 1^n , it takes $\tau(n)$ time (after receiving a message from \mathcal{R}) to send every message in its interaction with \mathcal{R} . So we have:

$$\frac{\text{Success}_{\mathcal{A}'}(n)}{\text{Success}_{\mathcal{R}\mathcal{A}'}(n)} = \frac{\text{Success}_{\mathcal{A}}(n)}{\text{Success}_{\mathcal{R}\mathcal{A}}(n)}$$

and, since by construction $\text{Time}_{\mathcal{R}\mathcal{A}'}(n) \geq \tau(n)\text{Query}_{\mathcal{R}\mathcal{A}}(n)$ and $\text{Time}_{\mathcal{A}'}(n) = \tau(n)\text{Query}_{\mathcal{A}}(n)$, we also have:

$$\frac{\text{Time}_{\mathcal{R}\mathcal{A}'}(n)}{\text{Time}_{\mathcal{A}'}(n)} \geq \frac{\tau(n)\text{Query}_{\mathcal{R}\mathcal{A}}(n)}{\tau(n)\text{Query}_{\mathcal{A}}(n)} = \frac{\text{Query}_{\mathcal{R}\mathcal{A}}(n)}{\text{Query}_{\mathcal{A}}(n)}$$

These facts taken together complete the argument. \square

C Rerandomizable Signatures

Our impossibility result for unique signatures also applies to the more general class of *rerandomizable* signatures, which are signature schemes that need not be unique but are “efficiently rerandomizable” in the sense that there exists some algorithm **Rerand** which can select a rerandomization of a signature uniformly randomly from the set of possible signatures of a message. Formally:

Definition 8. A **rerandomizable signature scheme** is a tuple $(\text{Gen}, \text{Sign}, \text{Ver}, \text{Rerand})$ of probabilistic polynomial-time algorithms such that, for every $n \in \mathbb{N}$:

- **Gen**, on input 1^n , produces a pair (pk, sk)
- **Sign**, on input (sk, m) for any $m \in \{0, 1\}^n$, produces a signature σ . (We write $\sigma \leftarrow \text{Sign}_{sk}(m)$.)
- **Ver**, on input (pk, m, σ) , produces either **Accept** or **Reject**. (We write $out \leftarrow \text{Ver}_{pk}(m, \sigma)$.)
- **Rerand**, on input (pk, m, σ) , produces a signature σ' . (We write $\sigma' \leftarrow \text{Rerand}_{pk}(m, \sigma)$.)

and, in addition, the following properties hold:

- *Correctness*: For every $n \in \mathbb{N}$ and $m \in \{0, 1\}^n$:

$$\Pr[(pk, sk) \leftarrow \text{Gen}(1^n) : \text{Ver}_{pk}(m, \text{Sign}_{sk}(m)) = \text{Accept}] = 1$$

- *Rerandomizability*: For any $pk \in \{0, 1\}^*$, $m \in \{0, 1\}^n$, and $\sigma \in \{0, 1\}^*$ such that $\text{Ver}_{pk}(m, \sigma) = \text{Accept}$, the output of $\text{Rerand}_{pk}(m, \sigma)$ is uniformly distributed over the set $\Sigma_{pk, m}$ of all signatures σ' where $\text{Ver}_{pk}(m, \sigma') = \text{Accept}$.

We note that this is of course a strict generalization of unique signatures, as we may take $\text{Rerand}_{pk}(m, \sigma) = \sigma$ to trivially transform any unique signature scheme into a rerandomizable signature scheme.

The impossibility results presented in [HJK12, BJLS16] are generalized to the case of rerandomizable signatures, but are notably still restricted to the case of straight-line or “simple” reductions. Through a slight alteration to the meta-reduction we use to prove Theorem 2 for the case of unique signatures, we can generalize our result (for arbitrary fixed-parameter reductions and any bounded-round assumption) to rerandomizable signatures by proving the following analogue of Theorem 2:

Theorem 3. Let $\Pi = (\text{Gen}, \text{Sign}, \text{Ver}, \text{Rerand})$ be a rerandomizable signature scheme, and let $(\mathcal{C}, t(\cdot))$ be some $r(\cdot)$ -round intractability assumption for polynomial $r(\cdot)$. If there exists some fixed-parameter black-box reduction \mathcal{R} for basing weak unforgeability of Π on the hardness of $(\mathcal{C}, t(\cdot))$, then either:

- (1) \mathcal{R} is not a linear-preserving reduction, or
- (2) there exists a polynomial-time adversary \mathcal{B} that breaks the assumption $(\mathcal{C}, t(\cdot))$.

While it is tempting to believe that we could directly apply the ideal adversary \mathcal{A} and meta-reduction \mathcal{B} from Theorem 2 to show this, while simply using the rerandomizability property as a substitute for uniqueness (that is, have \mathcal{A} and \mathcal{B} respectively rerandomize their forgeries to guarantee that the results, and hence transcripts, are identically distributed), there is a subtle

- Initially, receive a message pk , the public key; respond (i.e., generate m_1) according to the next step for $i = 1$.
- On receiving a message consisting of a partial transcript $\tau = (pk, m_1, \sigma_1, \dots, m_{i-1}, \sigma_{i-1})$ for some $i \in [\ell(n)]$, do the following:
 - Generate m_i by taking the first n bits resulting from applying the oracle \mathcal{O} to the input $(pk, i, \text{Valid}(\mathcal{O}, \tau))$.
 - Return the new partial transcript $\tau || m_i$ which results from appending m_i to the transcript τ .
- On receiving a message consisting of a complete transcript $\tau = (pk, m_1, \sigma_1, \dots, m_{\ell(n)}, \sigma_{\ell(n)})$ (we shall refer to such a message as a “forgery request” or “end message”), do the following:
 - If $\text{Valid}(\mathcal{O}, \tau)$ returns 0, then return \perp .
 - Otherwise, generate a random message m^* (distinct from each m_i in τ) by applying \mathcal{O} to the transcript τ' , use brute force to find a signature σ^* for which $\text{Ver}_{pk}(m^*, \sigma^*) = \text{Accept}$, and return the forgery $(m^*, \text{Rand}_{pk}(m^*, \sigma^*))$.

Let $\text{Valid}(\mathcal{O}, \tau)$ be defined as follows:

- Parse $\tau = (pk, m_1, \sigma_1, \dots, m_k, \sigma_k)$.
- Verify that, for each signature σ_i , $\text{Ver}_{pk}(m_i, \sigma_i) = \text{Accept}$. If not true for all i , return 0.
- Verify that, for each message m_i , m_i is equal to the first n bits resulting from applying the oracle \mathcal{O} to the input $(pk, i, 1)$. If not true for all i , then return 0.
- Otherwise, return 1.

Figure 5: Formal description of the “ideal” adversary $\mathcal{A}^{\mathcal{O}}$ for rerandomizable signatures.

issue that precludes doing so. In particular, it is no longer without loss of generality that \mathcal{R} is unable to rewind \mathcal{A} ; instead, since we no longer have the uniqueness property, \mathcal{R} may attempt to influence the signature queries generated by \mathcal{A} by rewinding to a prior query and offering a different (valid) signature for the respective message, which will change the random transcript to which the oracle \mathcal{O} is applied and thus influence subsequent queries from \mathcal{A} .

Hence, we instead require the subtly altered ideal adversary and meta-reduction presented in Figures 5 and 6, respectively. The key difference is that the input to the oracle \mathcal{O} depends not on the prior responses from \mathcal{R} , but only on whether the prior responses were valid; in this way, we can once again assert that \mathcal{R} is without loss of generality unable to rewind \mathcal{A} (as, for a particular \mathcal{O} , every sequence of valid responses from \mathcal{R} will produce the same sequence of queries from \mathcal{A} , irrespective of the responses themselves), while also preserving Claim 1 in that \mathcal{R} must still give a valid response to each of \mathcal{A} ’s queries in order to receive the correct subsequent queries from \mathcal{A} (and so it must still answer every query correctly, or guess the output of the random oracle, in order to receive a forgery).

Using this, we can now proceed to prove Theorem 3 in an identical manner to Theorem 2, with the aforementioned exception that, for the purposes of showing that \mathcal{A} ’s and \mathcal{B} ’s forgeries (now rerandomized before returning) are identically distributed in the case where \mathcal{B} succeeds (i.e., Claim 3), we replace the uniqueness property with the fact that the rerandomizations of any two valid

- Set initial view $v \leftarrow \perp$ and set $k \leftarrow 1$. Execute \mathcal{R} , updating the current view v according to the following rules.
- When \mathcal{R} begins a new instance of \mathcal{A} and sends a public key pk , label this instance as instance k . Generate and store $2\ell(n)$ random queries $\vec{m}_k^0 = (m_{k,1}^0, \dots, m_{k,\ell(n)}^0)$ and $\vec{m}_k^1 = (m_{k,1}^1, \dots, m_{k,\ell(n)}^1)$ and a target forgery m_k^* . (Abort and return Fail if m_k^* is equal to a message in either \vec{m}_k^* .) Also let $pk_k \leftarrow pk$ and initialize the forgery $f_k \leftarrow \{\}$. Lastly, respond with $\tau_k^* = pk_k || m_{k,1}$ and increment k .
- When \mathcal{R} attempts to communicate externally with \mathcal{C} , forward the message, return \mathcal{C} 's response to \mathcal{R} , and update v accordingly.
- When \mathcal{R} sends a transcript $\tau = (pk, m_{I,1}, \sigma_{I,1}, \dots, m_{I,j}, \sigma_{I,j})$ to some simulated instance I of \mathcal{A} :
 - If $\text{Valid}(\tau, \vec{m}_I^1) = 1$ then store the signature $\sigma_{I,j}$.
 - If $j = \ell(n)$ (i.e., this is an end message), then do the following:
 - * If $\text{Valid}(\tau, \vec{m}_I^1) = 0$ or if there exists some $j \in [\ell(n)]$ for which no response $\sigma_{I,j}$ has been stored^a, return \perp .
 - * Otherwise, run the procedure **Rewind** detailed below for the instance I .
 - * If, after running **Rewind**, there is a stored forgery $f_I = (m_I^*, \sigma_I^*)$, then return $(m_I^*, \text{Rerand}_{pk}(m_I^*, \sigma_I^*))$ and continue executing \mathcal{R} as above. Otherwise, abort the entire execution of \mathcal{B} and return Fail.
 - Lastly, respond with $\tau || m_{I,j+1}^{\text{Valid}(\tau, \vec{m}_I^1)}$ and continue the execution of \mathcal{R} .

Rewind procedure:

- Given instance I , for $j \in [\ell(n)]$ let $(v_{open}^j, v_{close}^j)$ denote the slot corresponding to the j^{th} signature query for instance I (i.e., the first instance where \mathcal{B} sends to \mathcal{R} the message $m_{I,j}^1$; note that there must be such an instance for \mathcal{B} to have stored the respective signature).
- For each $j \in [\ell(n)]$, “rewind” the slot $(v_{open}^j, v_{close}^j)$ as follows: Let $k' \leftarrow k$, and begin executing \mathcal{R} from the view $v' = v_{open}^j || m_I^*$ as in the main routine, with the following exceptions:
 - Replace any other instances of the message $m_{I,j}^1$ with m_I^* (e.g., if \mathcal{R} attempts to rewind and make a query to the same slot).
 - When \mathcal{R} begins a new instance of \mathcal{A} , label this instance as instance k' and increment k' . (That is, continue creating new instances, but preserve the counter k in the outer execution for after the rewinding.)
 - When \mathcal{R} attempts to communicate externally with \mathcal{C} , abort the rewinding and continue to the next j .
 - When \mathcal{R} sends an end message for an instance $I' \neq I$ of \mathcal{A} , abort the rewinding and continue to the next j , *unless* \mathcal{R} has not sent responses to $\ell(n)$ signature queries for that instance (in which case respond with \perp).
 - If v' ever contains a message whose transcript contains a response σ_I^* to any query for m_I^* , then, if it is the case that $\text{Ver}_{pk_I}(m_I^*, \sigma_I^*) = \text{Accept}$, store $f_i \leftarrow (m_I^*, \sigma_I^*)$ and end the **Rewind** procedure (i.e., return to the outer execution); otherwise, if $\text{Ver}_{pk_I}(m_I^*, \sigma_I^*) = \text{Reject}$, store nothing to f_I and continue to the next j .

Let $\text{Valid}(\tau, \vec{m}^1)$ be defined as follows:

- Parse τ as $(pk, m_1, \sigma_1, \dots, m_j, \sigma_j)$. If this fails, return 0.
- If, for any j , $\text{Ver}_{pk}(m_j, \sigma_j) = \text{Reject}$ or $m_j \neq m_j^1$, then return 0. Otherwise, return 1.

^aThis can occur with at most negligible probability if $\text{Valid} = 1$, as in that case \mathcal{R} would have to guess some element of \vec{m}^1 .

Figure 6: Formal description of the meta-reduction \mathcal{B} for rerandomizable signatures.

signatures of the same message are by definition identically distributed. As the proof is identical aside from this detail, we omit it.