# Redeeming Reset Indifferentiability and Applications to Post-Quantum Security

Mark Zhandry

NTT Research, USA & Princeton University, USA
mark.zhandry@ntt-research.com

**Abstract.** Indifferentiability is used to analyze the security of constructions of idealized objects, such as random oracles or ideal ciphers. Reset indifferentiability is a strengthening of plain indifferentiability which is applicable in far more scenarios, but has largely been abandoned due to significant impossibility results and a lack of positive results. Our main results are:

- Under *weak* reset indifferentiability, ideal ciphers imply (fixed size) random oracles, and domain shrinkage is possible. We thus show reset indifferentiability is more useful than previously thought.
- We lift our analysis to the quantum setting, showing that ideal ciphers imply random oracles under quantum indifferentiability.
- Despite Shor's algorithm, we observe that generic groups are still meaningful quantumly, showing that they are quantumly (reset) indifferentiable from ideal ciphers; combined with the above, cryptographic groups yield post-quantum *symmetric* key cryptography. In particular, we obtain a plausible post-quantum random oracle that is a subset-product followed by two modular reductions.

## 1 Introduction

The random oracle model [BR93] (ROM) has become a critical tool for justifying the security cryptosystems, both real-world and theoretical. In the ROM, all parties, including the cryptosystem and adversary, are given oracle access to a function $H$ sampled uniformly from the set of all functions. To actually implement the cryptosystem, $H$ is replaced with a concrete cryptographic hash function, with the hope that there is no way to exploit the structure of a well-designed $H$ to attack the cryptosystem. For many of the most efficient cryptosystems, the random oracle model is the only known justification for security, and constructions in the random oracle model tend to be simpler and require milder computational assumptions than those without random oracles.

Random oracles are members of a larger class of "idealized" objects, where an adversary is modeled as only having black box access. Ideal ciphers are idealizations of block ciphers, modeled as random keyed permutations. Generic groups are idealizations of cryptographic groups, modeled as random embeddings of $\mathbb{Z}_p$ into strings. Idealized objects have been used to design numerous cryptosystems (e.g. [RST01, Des00, BSW07, AY20, CLMQ20]) or justify the security

of new computational assumptions (e.g. Diffie-Hellman [Sho97] and its many variants [BBG05, BFF+14, DHZ14, BMZ19]). Ideal objects simplify the task of protocol design and analysis while providing meaningful heuristics for security.

### 1.1 Indifferentiability

Hash functions and other objects are usually built from lower-level building blocks. If one is not careful, such structure can be exploited in attacks [CDMP05], thus violating the random oracle assumption, even if the lower-level building block is treated ideally. The resolution is the indifferentiability framework of Maurer, Renner, and Holenstein [MRH04], a composable simulation-based definition which captures what it means for a construction to be "as good as" an ideal object, despite its structure, provided the underlying building block is treated ideally. Here, "as good as" applies to a wide array of settings called "single-stage games", capturing most standard cryptographic definitions. Indifferentiability has become a gold standard for analyzing hash function constructions, and numerous positive results are known such as domain extension and the equivalence of random oracles and ideal ciphers [CPS08, HKT11, DS16].

*Two Motivations for Reset Indifferentiability.* In the more general setting of "multi-stage" games, which capture cases where there are multiple distinct adversary parties, indifferentiability is insufficient [RSS11]. Such games include leakage resilience, deterministic encryption, key-dependent message security, and non-malleability, among others. In order to generically guarantee composition for multi-stage games including these critical applications, one needs a much stronger notion called *reset* indifferentiability, which is equivalent to requiring that the simulator be stateless. Given the limitations of plain indifferentiability, reset indifferentiability *should* be the gold standard, rather than plain indifferentiability.

Unfortunately, reset indifferentiability is subject to significant impossibility results [RSS11, LAMP12, DGHM13, BBM13]; in particular, any sort of domain extension is known to be impossible. Most prior work on reset indifferentiability focuses on a "strong" variant, which requires a single universal simulator to work for any distinguisher; under this variant, even stronger impossibilities are known. In particular, domain *shrinkage* is even impossible, which can in turn be used to prove other impossibilities such as constructing constant-sized ideal ciphers from infinite-sized random oracles, or vice versa [BBM13]. These are surprising and counter-intuitive results, and seem to have been interpreted as implying that reset indifferentiability is *too strong* to be useful. As such, reset indifferentiability seems to have been largely abandoned, with authors instead proposing milder notions of indifferentiability and showing that they apply to restricted classes of games [RSS11, DGHM13, Mit14]. However, reset indifferentiability is *exactly* characterized by general multi-stage games, meaning there will necessarily be applications where such restricted notions cannot be applied. Thus, under these weaker notions, security for a particular game has to be carefully analyzed.

However, we note that, beyond the impossibility of domain extension, not much is actually known about the "weak" variant of reset indifferentiability,

where the stateless simulator can depend on the distinguisher. This variant still captures general multi-stage games, meaning any weak reset indifferentiability result implies full applicability of the construction. Even though domain extension is still not possible, the notion may still be useful in many applications. For example, if one is considering public key encryption with fixed-sized messages, then domain extension may not be necessary.

An independent, perhaps unexpected, motivation for reset indifferentiability comes from the threat of quantum computing. The ability of a quantum algorithm to query the idealized object in superposition invalidates most classical results, and certain impossibilities are known [BDF+11, YZ20]. The difficulty is that even a single superposition query "views" the entire oracle; in order to ensure that the simulation of the ideal object is consistent and "looks like" the true ideal object, the approach employed by most works (e.g. [BDF+11, Zha12b, Unr15, TU16]) has been to simulate essentially statelessly, with the simulator usually depending on the distinguisher. In the context of indifferentiability, such an approach would correspond exactly to weak reset indifferentiability. We note that some recent techniques [Zha19, LZ19, CMSZ19, DFMS19, DFM20, KSS+20, YZ20] utilize stateful quantum simulators, and in particular [Zha19] proves the (non-reset) indifferentiability of domain extension for random oracles. However, these techniques are far more complex and require comparatively heavy quantum machinery, making the techniques more difficult to use.

We highlight the specific case of random permutations, which has been particularly challenging with few quantum results and techniques known for the setting where inverse queries are allowed. In fact, we are only aware of two such prior results: [AR16] considers the Even-Mansour cipher, but only considers adversaries with perfect success probability. [Zha16] constructs (non-indifferentiable) quantum-secure PRPs in such a model, but side-steps the issue of quantum queries entirely by having the entire oracle truth table be *statistically* close to a random permutation.

*Questions.* The prior discussion raises the following natural questions:

- Can weak reset indifferentiability be used to achieve *any* non-trivial result, even domain shrinkage?
- If so, how can one make non-black box use of the distinguisher to design an indifferentiability simulator?
- Can *fixed-size* random oracles be built from ideal ciphers, or vice versa?
- Can random oracles (fixed-size or infinite size) be built from ideal ciphers *quantumly*, even in the single-stage setting? In particular, can anything be said about the Sponge construction?

Making progress on these questions will be the focus of our work.

### 1.2 Our Results

*On Prior Impossibilities.* Essentially the main prior impossibility for weak reset indifferentiability is that of domain extension [RSS11, LAMP12, DGHM13,

BBM13], with other impossibilities [BBM13] relying crucially on the *strong* reset variant. We first observe that the techniques yielding the impossibility of domain extension apply even in the setting of *query-unbounded* simulators.

In contrast, we prove weak reset indifferentiability for random oracle domain shrinkage, ideal ciphers from random oracles, and vice versa, in such an unbounded simulator setting. More generally, we demonstrate that in *distinguisha*bility against query-unbounded attackers can usually be lifted to reset in *differentia*bility using query-unbounded simulators. The inefficient simulator makes these results rather un-useful for positive results. Nevertheless, it shows that the known techniques for negative results are unlikely to extend to a variety of interesting problems, in the weak reset setting. Combined with the lack of prior positive results for reset indifferentiability, this shows that weak reset indifferentiability is essentially completely open for any application that does not require domain extension. The question then is: how can we achieve an *efficient* simulator in these settings?

*Positive Results for Weak Reset Indifferentiability.* We first show that domain shrinking *is* possible, under weak reset indifferentiability with an *efficient* simulator. We thus see that random oracles with larger domain are *strictly stronger* that random oracles with smaller domain. This is in sharp contrast to the "duality" of *strong* reset indifferentiability, where any two objects are either *equivalent* or *incomparable*, with most examples being incomparable [BBM13].

We also show how to construct a (fixed-size) random oracle from an ideal cipher under weak reset indifferentiability, again with an efficient simulator. Specifically, we show that a natural pad-and-truncation of an ideal cipher—that is, the Sponge construction for a single-block message—gives a random oracle, for sufficient padding and truncation. An interesting feature of our analysis of pad-and-truncate is that the sum of the input and output sizes must be less than the width of the cipher. We show that this is tight: any larger input/output size will not be weakly reset indifferentiable, thus giving (to the best of our knowledge) the first negative result for weak reset indifferentiability that does not rely on domain extension. This is in contrast to the plain (non-reset) indifferentiability setting, where any non-trivial truncation gives indifferentiability [DRRS09]. Our result may help guide the design of Sponge-based hash functions.

These positive results are obtained by first proving reset indifferentiability in certain *shared randomness* models, which allows the simulator access to some consistent randomness, while still being stateless. We show that, for weak reset indifferentiability and for certain classes of "nice" ideal objects (including random oracles and ideal ciphers), the shared randomness can be removed to get a standard reset indifferentiability result.

*Quantum.* All of our results extend to the quantum setting. The simulators are identical to their classical counterparts. However, very few prior quantum results handle inverse queries, meaning a handful of new ideas are needed to lift our ideal cipher results to the quantum setting. We thus obtain the first proof of quantum indifferentiability (reset or otherwise) for a random oracle from an ideal cipher—and in particular the sponge construction for single-block messages. This

may give some evidence for the post-quantum (non-reset) indifferentiability of SHA3, which is based on the full sponge construction. While we cannot prove indifferentiability for the full Sponge construction[1], we can plug pad-and-truncate into the domain extension result of Zhandry [Zha19], obtaining the first quantum indifferentiability proof of an arbitrary-size random oracle from an ideal cipher, under (plain) indifferentiability.

*The Post-Quantum Generic Group Model.* We observe that Shor's algorithm, by virtue of being generic, is captured by the *generic group model* [Sho97] (GGM), albeit the quantum variant allowing quantum access to the group. Thus, despite Shor's algorithm, the GGM may remain a plausible heuristic in the quantum setting. Shor's algorithm, however, shows that the discrete-logarithm problem is easy in the quantum accessible GGM, so the question is then: what use is it?

We demonstrate that the quantum accessible GGM is equivalent to an ideal publicly-invertible injective function under (reset) indifferentiability. Our above positive results for ideal ciphers extend to the injective function case. In particular, by plugging in the above results, we obtain a quantum indifferentiable random oracle from the generic group model[2]. When instantiating with the multiplicative group over finite fields, the result is a plausible post-quantum hash function that is simply a subset-product, followed by two modular reductions.

### 1.3 Discussion

We significantly expand the set of techniques and results for reset indifferentiability, both classically and quantumly. We thus show that reset indifferentiability is more useful than suggested by prior works. Perhaps the main open question in the classical setting is whether ideal ciphers can be built from random oracles under reset indifferentiability.

We in particular expand the set of techniques available for analyzing quantum queries to permutation inverses, and in doing so expand the applicability of "old school" quantum simulation techniques, showing for the first time that stateless simulation is capable of achieving non-trivial indifferentiability results. Our hope is that our techniques can be combined with the sophisticated "new school" quantum techniques to aid in additional positive results. For example, can quantum indifferentiable ideal ciphers be built from random oracles?

Our results also show that cryptographic groups remain potentially useful in the quantum setting, just that they are limited to the symmetric key setting. While existing symmetric cryptography appears somewhat resilient to quantum attacks, we believe it is nevertheless important to study alternative techniques for building quantum-resistant symmetric cryptography.

[1] Our techniques work within the framework of reset indifferentiability, which cannot achieve domain extension, and therefore our techniques cannot apply to the full Sponge construction.

[2] [ZZ21] previously suggest building a random oracle from generic groups. Their result however is in the classical setting using stateful simulators, which does not translate to quantum. Our results are required to get a quantum indifferentiability proof.

### 1.4 Concurrent and Independent Work

Currently and independently of our work, Czajkowski [Cza21] prove the (plain) indifferentiability of the full Sponge construction in the quantum setting, necessarily using a stateful quantum simulation technique. In particular, this also justifies the *plain* quantum indifferentiability of the pad-and-truncate construction. The results and techniques are largely incomparable to ours, as we focus on reset indifferentiability.

## 2 Technical Overview and Discussion

*Indifferentiability.* Recall the usual notion of *indistinguishability* between two distributions over functions $F, G$, which says that the functions cannot be distinguished by oracle access. We will denote such indistinguishability as

$$F \approx G \ .$$

Indistinguishability is sufficient for settings like constructing a PRP from a PRF, as the underlying PRF building block is private and not directly accessible to the adversary. In the settings of length extension for hash functions, building ideal ciphers from random oracles, etc, indistinguishability is not sufficient since the adversary additionally can query the underlying building block, and indifferentiability [MRH04] is required instead. A construction $C$ making oracle queries to an ideal object $A$ (denoted $C^A$), is *indifferentiable* from an ideal object $B$ if there exists a simulator $S$ making queries to $B$ (denoted $S^B$) such that

$$(C^A, A) \approx (B, S^B) \ .$$

The above says that an adversary with two query interfaces—an "honest" interface to $B$ and "adversarial" interface to $A$—cannot distinguish the "Real Word" where $B$ is set to $C^A$ for ideal object $A$ from the "Ideal World" where $B$ is ideal and $A$ is simulated as $S^B$. For building an ideal cipher from a random oracle, $A$ represents a random oracle and $B$ an ideal cipher, with $C^A$ being a construction of a cipher from a hash function.

Note that, while the expression above appears symmetric between $A$ and $B$, for plain indifferentiability the notation hides the fact that $S$ can keep state between queries, whereas $C$ is usually considered to be stateless. Reset indifferentiability is a strengthening of indifferentiability to require $S$ to be stateless as well. As discussed above, reset indifferentiability is required in settings known as "multi-stage games." We disambiguate between strong and weak security, were strong requires a universal simulator that works for any potential distinguisher between $(C^A, A)$ and $(B, S^B)$, whereas weak allows for a distinguisher-dependent simulator. Weak reset indifferentiability is sufficient for composition and multi-stage games [RSS11]. Strong reset indifferentiability turns out to fully symmetric, with the roles of $C$ and $S$ being interchangeable [BBM13]. This means that any construction (resp. impossibility) of $B$ from $A$ immediately gives a construct (resp. impossibility) of constructing $A$ from $B$.

### 2.1 On Prior Impossibilities

We show that if one relaxes to query-*unbounded* simulation, then *indistinguishability* can be upgraded to weak reset indifferentiability, provided the indistinguishability holds against query-unbounded distinguishers. The idea is that the simulator can query the entire object $B$, and then sample $A$ conditioned on $C^A$ being functionally identical to $B$; such sampling is guaranteed by plain indistinguishability against unbounded queries. The difficulty is that there may be many $A$ such that $C^A$ is equivalent to $B$, and we must ensure that the simulator can consistently choose the same $A$ each time. For this, we show the simulator can basically have a choice of $A$ hard-coded for each separate $B$. The details are given in Section 4.

Query-unbounded indistinguishability follows from known results in various settings. For example, perfect shuffles [GP07] allow for constructing PRPs from random oracles. Indistinguishable domain shrinkage is also trivial. Our general theorem lifts these results to weak reset indifferentiability, albeit with inefficient simulators. Due to the above inefficient simulator, the result is not immediately useful. However, we observe that the impossibility of domain extension holds *even* under such inefficient simulators; for completeness, we give the result in the full version [Zha21]. Since domain extension is the main impossibility known to hold for *weak* reset indifferentiability, this shows that new techniques would be required to rule efficient simulation in settings where inefficient simulation is possible. We thus demonstrate that weak reset indifferentiability is largely open for settings that do not involve domain extension.

### 2.2 Shared Randomness Indifferentiability

We next discuss a model of indifferentiability, which we call shared randomness reset indifferentiability, that we will use as a stepping-stone to full reset indifferentiability. Here, the simulator $S$ is still stateless, but is allowed to query a random oracle $R$—independent from $A$ and $B$—in addition to querying $B$; we require that:

$$(C^A, A) \approx (B, S^{B,R}) \ .$$

Note that the random oracle breaks the symmetry between $A$ and $B$. In particular, we note that domain shrinking is trivial in this setting, as the simulator can use $R$ to simulate the parts of $A$ that are ignored by $C^A$.

In Section 6, we also show that shared randomness is sufficient for constructing a fixed-size random oracle $h$ from a (keyless) ideal cipher $P, P^{-1}$. The construction is the natural one based on truncation:

$$\mathsf{PadTrunc}_{c,d}^{P,P^{-1}}(x) = P(x||0^{(1-c)n})|_{[dn]} \ .$$

Here, $c, d \in (0,1)$ are constants, $P$ is an ideal cipher on $n$-bit inputs, $x$ is $cn$ bits and $y|_{[r]}$ is the first $r$ bits of $y$. Interestingly, we show that if $c + d > 1$, then the truncation-based construction is actually *not* reset indifferentiable:

**Theorem 1 (Informal).** *If $c + d > 1$, $\mathsf{PadTrunc}_{c,d}$ is* not *shared-randomness weakly reset indifferentiable from a random oracle.*

The proof of this theorem is as follows. Consider a distinguisher $D$ with query access to a function $H$ and permutation $P, P^{-1}$. It first chooses a random $x \in \{0,1\}^{cn}$ and queries $w||z \leftarrow P(x||0^{(1-c)n})$. It also queries $w' \leftarrow H(x)$, and checks that $w' = w$. Then it queries $x'||y' \leftarrow P^{-1}(w||z)$, and checks that $x' = x, y' = 0^{(1-c)n}$. $D$ outputs 1 if and only if all checks pass. Note that in the "Real world" where $H = \mathsf{PadTrunc}_{c,d}^{P,P^{-1}}$, $D$ outputs 1 always. However, in the "Ideal world" with $P, P^{-1}$ being supposedly simulated by a stateless simulator $S^H$, we argue that $D$ outputs 0 almost always. Indeed, a stateless simulator must have $w = w'$ to pass the distinguisher's first check. But then to answer the query $P^{-1}(w||z)$, it must somehow come up with the original pre-image $x$ of $w$. Since the simulator is stateless, it cannot remember $x$, and so computing $x$ would seem to require inverting $H$ on $w$, which is impossible for a random oracle $H$.

This intuition is not quite correct, as the simulator is also given $z$ as input, which can be seen as some side-information about $x$. However, for $c + d > 1$, $z$ is shorter than $x$, and therefore there must be some entropy left in $x$. Since random oracles remain hard to invert even for entropic sources, the inability for the simulator to output $x$ follows.

On the other hand, for $c + d \leq 1$, we show that $\mathsf{PadTrunc}_{c,d}$ actually *is* reset indifferentiable:

**Theorem 2 (Informal).** *If $c + d \leq 1$, $\mathsf{PadTrunc}_{c,d}$ is (strongly) reset indifferentiable from a random oracle in the shared randomness model.*

Inspired by the impossibility above, we devise a simulator that statelessly encodes $x$ into $z$ so that $x$ can be recovered from $z$ alone. It does this by setting $z$ to be the result of a random injection $I$ applied to $x$, in the case that $y = 0^{(1-c)n}$. For $I$ to indeed be a random injection, we must have $c + d \leq 1$. The problem is that $I$ represents state, which is not allowed in reset indifferentiability. Fortunately, for *shared randomness* reset indifferentiability, $S$ has access to a random oracle $R$; it can use this single random oracle to build $I$. Essentially, it follows typical approaches to building block ciphers from pseudorandom random functions, but instantiating the pseudorandom function using $R$.

In Section 5, we show that shared randomness reset security actually implies standard weak reset security, in many settings:

**Theorem 3 (Informal).** *Suppose a construction $C^A$ is shared randomness weakly reset indifferentiable from $B$, and that $B$ has certain nice "extraction" properties. Then $C^A$ is also weakly reset indifferentiable from $B$, without shared randomness.*

Combining with the above results shows that the ideal cipher model implies random oracles under weak reset indifferentiability.

The theorem is proved in two steps. First, we replace the shared randomness $R$ with a $q$-wise independent hash function $R_q$, where $q$ is set sufficiently large

relative to the number of queries made by the adversary. The result is perfectly indistinguishable from a truly random $R$. Next, we use a trick from [BBM13] to compute $R_q$ from the oracle $B$ itself, in a way such that $R_q$ is random and independent from the adversary's view.

We note that our simulator is almost black box, but requires knowledge of the number of queries made by the distinguisher, both to select $q$ and to apply the trick from [BBM13].

### 2.3 Quantum Distinguishers and Generic Groups

Reset indifferentiability is conveniently amenable to quantum proof techniques, and we show how to upgrade our positive results to the quantum setting. This is not trivial, but we show how to structure the classical proofs in such a way that they can be lifted to the quantum setting by plugging in known quantum query lower bounds in key steps. This requires care, since existing techniques mostly prohibit inverse queries to random permutations, whereas our results require such inverse queries. We thus must carefully embed prior inverse-query-less results into our setting to achieve our results. As a result, we obtain fixed-size random oracles from ideal ciphers quantumly. Generically plugging into the domain extension result of Zhandry [Zha19], we obtain the first proof of quantum indifferentiability of an (arbitrary) size random oracle from an ideal cipher:

**Corollary 1.** *There exists a construction $C$ of an (arbitrary-size) random oracle from an ideal cipher that is quantum (non-reset) indifferentiable.*

We note that our lower bound on the necessary truncation of ideal ciphers also trivially extend to the quantum setting, since a classical distinguisher is in particular a quantum distinguisher[3].

We next investigate the generic group model, quantumly. It is well known that Shor's quantum discrete log algorithm [Sho94] works on any cryptographic group; another interpretation is that Shor's algorithm works in the quantum-accessible generic group model. This interpretation of the generality of Shor's algorithm is usually seen as a negative, since it means that there is no hope of circumventing the algorithm by using alternate groups. But we interpret this as showing that Shor's algorithm does not fundamentally alter the validity of the generic group model quantumly. It just shows that discrete logarithms are now tractable.

The ability of Shor's algorithm to solve discrete log essentially shows that the generic group gives a random injection, quantumly, which we prove formally under reset indifferentiability. Our positive results from above readily apply to publicly invertible injections, and therefore give an quantum indifferentiable hash function from generic groups.

If we in particular focus on the case of finite fields, what we get is the hash function $H(x) = (g^x \bmod p) \bmod 2^n$, where $x \in \{0,1\}^n$ for $2n \leq \log p$. By

[3] There is a slight subtlety here, as quantum (reset) indifferentiability allows for a quantum simulator, whereas classical indifferentiability does not. Thus, quantum and classical indifferentiability are technically *incomparable*. Nevertheless, our impossibility results trivially adapt to the quantum simulator case.

pre-computing the various powers of 2, $g^x$ becomes a modular subset-product computation. The overall hash function is then a modular subset product followed by an additional modular reduction that can plausibly be used as a (quantum immune) random oracle.

## 3 Preliminaries

Unless otherwise noted, all functions, sets, algorithms, adversaries, distinguishers, simulators, and distributions are functions of a security parameter $\lambda$. We will often omit the security parameter; for example, when we say that $\mathcal{X}$ is a set, we mean that $\mathcal{X}$ is a family of sets $\{\mathcal{X}_\lambda\}_\lambda$. When we say that a function is polynomial or negligible, we mean polynomial or negligible in $\lambda$. When there are multiple functions of $\lambda$, we assume all functions use the same $\lambda$.

For an algorithm $A$ making queries to another (potentially stateful) algorithm $B$, we will denote their interaction by $A^B$.

*Ideal Objects.* For sets $\mathcal{X}, \mathcal{Y}$, a ideal object is a distribution over functions from $\mathcal{X}$ to $\mathcal{Y}$. Some idealized objects we will consider:

- **Random oracles.** A random oracle is just the uniform distribution over all functions $\mathsf{RO}$ from $\mathcal{X}$ to $\mathcal{Y}$. We denote this distribution by $\mathcal{Y}^{\mathcal{X}}$. Note that we will usually think of $\mathcal{X}, \mathcal{Y}$ as finite exponential size. It is also possible to consider an infinite random oracle, in which case $\mathcal{X}$ is infinite.
- **Ideal ciphers.** Let $\mathcal{X} = \{0,1\} \times \mathcal{K} \times \mathcal{Y}$ for exponential-size $\mathcal{Y}$, and $\mathcal{K}$ be another set. An ideal cipher is sampled by choosing a function $P : \mathcal{K} \times \mathcal{Y} \to \mathcal{Y}$, where for each $k$, the function $P(k, \cdot)$ is a uniformly random permutation. Let $P^{-1}(k, \cdot)$ be the inverse of $P(k, \cdot)$. The oracle is then $\mathsf{IC}(b, k, x) = \begin{cases} P(k,x) & \text{if } b = 0 \\ P^{-1}(k,x) & \text{if } b = 1 \end{cases}$. We note that ideal ciphers are typically modeled as being keyed, which corresponds to an exponential-sized family of independent ideal permutations. It is also possible to consider the keyless setting, where $\mathcal{K} = \{1\}$, and can be omitted.
- **(Keyed) Random Injections.** Let $\mathcal{Y} = \mathcal{Y}' \cup \{\perp\}$, $\mathcal{Z}$ an exponential-sized set such that $|\mathcal{Z}| \leq |\mathcal{Y}'|$, and $\mathcal{K}$ be another set. Then let $\mathcal{X} = (\{0\} \times \mathcal{K} \times \mathcal{Z}) \cup (\{1\} \times \mathcal{K} \times \mathcal{Y}')$. A keyed random injection is sampled by choosing a function $I : \mathcal{K} \times \mathcal{Z} \to \mathcal{Y}'$ where for each $k$, the function $I(k, \cdot)$ is a uniformly random injection. Let $I^{-1}(k, y)$ be the function that outputs $x$ such that $I(k, x) = y$ if it exists, and otherwise outputs $\perp$. Then $\mathsf{RI}(b, k, x) = \begin{cases} I(k,x) & \text{if } b = 0 \\ I^{-1}(k,x) & \text{if } b = 1 \end{cases}$.
- **Generic groups.** Let $p$ be an exponentially-large prime such that $|\mathcal{Y}| \geq p$, and let $L$ be a random injection from $\mathbb{Z}_p$ to $\mathcal{Y}$. The function $\mathsf{GG}$ then maps $x \mapsto L(x)$, and also $(\ell_1, \ell_2) \mapsto L(\quad L^{-1}(\ell_1) + L^{-1}(\ell_2) \quad)$. Here, if $L^{-1}$ is undefined on an input $\ell$, the entire expression outputs $\perp$. Note that the generic group model usually also allows for subtraction, but this is redundant since $p$ is known, and $-1 \equiv p - 1 \bmod p$ can be computed using just the addition operation.

*Quantum.* We will not need much quantum background in this work. In particular, all of our quantum results basically follow the classical proofs, but with key parts replaced with quantum equivalents.

### 3.1 Indifferentiability

Let $\mathcal{A}, \mathcal{B}$ be two distributions over functions, and $C$ a polynomial-time oracle-aided circuit. We write $C^{\mathcal{A}}$ to be the distribution over $C^A$ where $A \leftarrow \mathcal{A}$.

**Definition 1.** *$C^{\mathcal{A}}$ is (strong statistical classical plain) indifferentiable from $\mathcal{B}$ if there exists a polynomial-size, potentially stateful, oracle-aided simulator $S$ such that, for any probabilistic potentially unbounded oracle-added Turing machine $D$ making at most a polynomial number of queries, there is a negligible $\epsilon$ such that*

$$\left| \Pr_{A \leftarrow \mathcal{A}} \left[ D^{C^A, A}() = 1 \right] - \Pr_{B \leftarrow \mathcal{B}} \left[ D^{B, S^B}() = 1 \right] \right| \leq \epsilon \ .$$

*Variants.* We now discuss some variants of the indifferentiability definition:

- **Weak vs strong.** Weak indifferentiability allows for $S$ to depend on $D$, flipping the order of quantifiers.
- **Computational vs statistical vs perfect.** Computational indifferentiability only requires security to hold for polynomial-sized $D$. Note that in the statistical case, we still bound the number of queries made by $D$ to be polynomial. On the other hand, perfect indifferentiability requires security to hold for unbounded Turing machines, and for $\epsilon$ to be 0.
- **Quantum vs classical.** Quantum indifferentiability requires security to hold for quantum distinguishers $D$ which can make *quantum* queries to their oracles, but potentially allows for quantum simulators $S$ which can make quantum queries as well.
- **Reset vs plain.** Reset indifferentiability requires $S$ to be stateless. We note that [RSS11] define reset indifferentiability differently, allowing the simulator to be stateful but allowing the distinguisher to "reset" the simulator to its initial state at any point. The two versions are readily seen to be equivalent, and we prefer the stateless simulator definition for its simplicity.

We note that the four variants above are all orthogonal and any subset can be considered, giving a total of 24 possible notions of indifferentiability. Note that strong implies weak, reset implies plain, and perfect implies statistical implies computational, for any settings of the other variants. Quantum does not *necessarily* imply classical since it could be the case that a quantum simulator can fool a classical distinguisher, but no classical simulator can. However, in all cases we will consider in this work, if the scheme is quantum indifferentiable for some setting of the other variants, it will also be classical indifferentiable for the same variants. Thus, for our purposes, we will treat quantum indifferentiability as being stronger.

## 4 Lifting Indistinguishability to Indifferentiability in the Unbounded Setting

Here, we show how to lift query-unbounded in*distinguisha*bility into weak reset in*differentia*bility, albeit with query-unbounded simulation.

**Theorem 4.** *Let $\mathcal{A}, \mathcal{B}$ be distributions and $C$ a construction. Suppose the distributions of truth tables $B$ and $C^A$ for $A \leftarrow \mathcal{A}, B \leftarrow \mathcal{B}$ are statistically close. Suppose further that $\mathcal{B}$ has super-logarithmic min-entropy $H_\infty(\mathcal{B}) := \min_B \log 1/\Pr[B \leftarrow \mathcal{B}]$. Then for any (potentially query unbounded, classical or quantum) distinguisher $D$, there exists a query unbounded classical simulator $S$ and a negligible $\epsilon$ such that:*

$$\left| \Pr_{A \leftarrow \mathcal{A}} \left[ D^{C^A, A}() = 1 \right] - \Pr_{B \leftarrow \mathcal{B}} \left[ D^{B, S^B}() = 1 \right] \right| \leq \epsilon \ .$$

In other words, if $C^A$ is indistinguishable from $\mathcal{B}$ against unbounded distinguishers, then $C^A$ is also indifferentiable from $\mathcal{B}$, albeit using a query unbounded simulator.

*Proof.* Fix any distinguisher $D$. For any $B$, let $Q_B$ be the distribution over $A \leftarrow \mathcal{A}$, conditioned on $C^A$ being identical to $B$. Then, by the statistical closeness of $C^A$ and $B$, we have that there exists a negligible $\delta$ such that

$$\left| \Pr_{A \leftarrow \mathcal{A}} \left[ D^{C^A, A}() = 1 \right] - \Pr_{B \leftarrow \mathcal{B}, A \leftarrow Q_B} \left[ D^{B, A}() = 1 \right] \right| \leq \delta$$

Now consider the following distribution $\mathcal{J}$ over functions $J$: for each $B$, $J(B)$ is sampled from $Q_B$, independently from all other inputs. Then we have that

$$\Pr_{B \leftarrow \mathcal{B}, J \leftarrow \mathcal{J}} \left[ D^{B, J(B)}() = 1 \right] = \Pr_{B \leftarrow \mathcal{B}, A \leftarrow Q_B} \left[ D^{B, A}() = 1 \right]$$

We now describe our simulator $S$. $S$ will have a $J$ hard-coded. For every query, it will compute the truth table for $B$ in its entirety by making exponentially many queries, and then set $A = J(B)$. It will then answer the query using $A$. It remains to show how to select $J$. What we show is that, for any $D$, a *random* $J$ drawn from $\mathcal{J}$ will do. Concretely, consider the random variable $p := \Pr_{B \leftarrow \mathcal{B}} \left[ D^{B, J(B)}() = 1 \right]$, which depends on $J$. We observe that $p$ is identical to the random variable $\sum_B \Pr[B \leftarrow \mathcal{B}] p_B$, where the $p_B \in [0, 1]$ are independent random variables obtained by sampling $A \leftarrow Q_B$ and outputting $\Pr[D^{B, A}() = 1]$, where the last probability is over any random coins of $D$. Each $p_B$ is in $[0, 1]$, and the expectation of $p$ is exactly $q := \Pr_{B \leftarrow \mathcal{B}, A \leftarrow Q_B} \left[ D^{B, A}() = 1 \right]$.

We apply Hoeffding's inequality to the random variables $\Pr[B \leftarrow \mathcal{B}] p_B$, giving:

$$\begin{aligned} \Pr[|p - q| \geq \gamma] &\leq 2e^{-2\gamma^2 / \sum_B \Pr[B \leftarrow \mathcal{B}]^2} \\ &\leq 2e^{-2\gamma^2 2^{H_\infty(\mathcal{B})} / \sum_B \Pr[B \leftarrow \mathcal{B}]} = 2e^{-2\gamma^2 2^{H_\infty(\mathcal{B})}} \end{aligned} \tag{1}$$

Since $2^{H_\infty(\mathcal{B})}$ is super-polynomial, we can choose $\gamma$ negligible while still having Line 1 be less than 1. Thus, there is *some* value of $p_B$ for each $B$ (and hence

choice of $J$) such that $|\Pr_{B\leftarrow\mathcal{B}}\left[D^{B,J(B)}()=1\right]-p|\leq\gamma$. The simulator therefore uses this choice of $J$, and we have

$$\left|\Pr_{A\leftarrow\mathcal{A}}\left[D^{C^A,A}()=1\right]-\Pr_{B\leftarrow\mathcal{B}}\left[D^{B,S^B}()=1\right]\right|\leq\delta+\gamma$$

which is negligible. □

## 5 Shared Randomness Indifferentiability

In this section, we present shared randomness models of reset indifferentiability. In this model, the simulator has access to a source of randomness, and the same randomness is used in every invocation of the simulator. We will actually consider two variants, one where the shared randomness is simply a random string, and the other where the shared randomness is a random oracle.

*Shared Random String (SRS).* This model is equivalent to read-only indifferentiability [BDG20]. The simulator has access to an arbitrary-size random string.

**Definition 2.** *$C^{\mathcal{A}}$ is (strong statistical classical) reset indifferentiable from $\mathcal{B}$* in the SRS model *if there exists set $\mathcal{R}$ and a polynomial-sized stateless oracle-aided simulator $S$ such that, for any probabilistic potentially unbounded oracle-added Turing machine $D$ making at most a polynomial number of queries, there exists a negligible $\epsilon$ such that*

$$\left|\Pr_{A\leftarrow\mathcal{A}}\left[D^{C^A,A}()=1\right]-\Pr_{B\leftarrow\mathcal{B},r\leftarrow\mathcal{R}}\left[D^{B,S^B(\cdot\,;\,r)}()=1\right]\right|\leq\epsilon\ .$$

*Above, $S^B(\cdot\,;\,r)$ means that queries $x$ to $S$ are answered as $S^B(x\,;\,r)$.*

*Remark 1.* [DGHM13] consider a notion of *resource restricted* indifferentiability, where the simulator's space is bounded but potentially non-zero. While the SRS model can be seen as a form of storage, the model is incomparable: SRS allows for unbounded length random string, but the string must be read-only.

*Shared Random Oracle (SRO).* Here, the simulator has access to an arbitrary-sized random *oracle*.

**Definition 3.** *$C^{\mathcal{A}}$ is (strong statistical classical) reset indifferentiable from $\mathcal{B}$ in the SRO model if there exists sets $\mathcal{X},\mathcal{Y}$ and a polynomial-sized stateless oracle-aided simulator $S$ such that, for any oracle-aided Turing machine $D$ making at most a polynomial number of queries, there exists a negligible $\epsilon$ such that*

$$\left|\Pr_{A\leftarrow\mathcal{A}}\left[D^{C^A,A}()=1\right]-\Pr_{B\leftarrow\mathcal{B},H\leftarrow\mathcal{Y}^{\mathcal{X}}}\left[D^{B,S^{B,H}}()=1\right]\right|\leq\epsilon\ .$$

*Above, $\mathcal{Y}^{\mathcal{X}}$ is the uniform distribution over the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$.*

When contrasting SRS or SRO indifferentiability from Definition 1, we call Definition 1 the *standard* model. Strong vs weak, computational vs statistical vs perfect, and quantum vs classical are defined analogously to the setting without shared randomness. Note that the definitions also makes sense in the plain (non-reset) setting. However, the SRS and SRO models are redundant in the plain setting, as shown in the following:

**Lemma 1.** *Let $\Phi \in \{\textit{strong}, \textit{weak}\}$, $\Gamma \in \{\textit{computational}, \textit{statistical}, \textit{perfectly}\}$ and $\Delta \in \{\textit{classical}, \textit{quantum}\}$. If $C^{\mathcal{A}}$ is $\Phi$ $\Gamma$ $\Delta$ plain indifferentiable from $\mathcal{B}$ in either the of the SRS or SRO models, then it is also $\Phi$ $\Gamma$ $\Delta$ plain indifferentiable from $\mathcal{B}$ in the standard model.*

*Proof.* All 12 settings of $\Phi, \Gamma, \Delta$ are essentially identical. We first show the SRS case. Given a simulator $S$ for SRS indifferentiability, we can simply create a new simulator which chooses a random string $r$ at the first query, and answers all queries using $S(\cdot\ ;\ r)$. For the SRO case, we can simulate the shared random oracle on the fly. In the classical case, this is done via lazy sampling; in the quantum case, this is done using Zhandry's compressed oracles [Zha19]. □

We note that shared randomness is *not* necessarily redundant in the reset setting since there is no explicit ability to store $r$ in order to maintain consistency between the different executions. Looking forward, our results imply that shared randomness is an extra resource in the strong reset setting (in the sense that it makes the notion weaker), but it is usually redundant in the weak reset setting.

### 5.1 Domain Shrinkage

To illustrate the utility of the shared randomness models, we show that the SRO model is sufficient for domain shrinkage, even with reset indifferentiability. This is in contrast to strong reset indifferentiability without shared randomness, where [BBM13] show that domain extension *and* shrinkage are impossible.

Our domain shrinker is the obvious one, which just ignores part of the domain. Let $\mathcal{X}, \mathcal{Y}$ be sets with $A : \mathcal{X} \to \mathcal{Y}$. Let $\mathcal{X}' \subset \mathcal{X}$. Then $\mathsf{Shrink}^A : \mathcal{X}' \to \mathcal{Y}$ is simply defined as $\mathsf{Shrink}^A(x) = A(x)$.

**Theorem 5.** $\mathsf{Shrink}^{\mathsf{RO}}$ *is strong perfectly quantum and classical reset indifferentiable from a random oracle, in the SRO model.*

*Proof.* Let $B : \mathcal{X}' \to \mathcal{Y}$ and $H : \mathcal{X} \to \mathcal{Y}$. Let

$$S^{B,H}(x) = \begin{cases} B(x) & \text{if } x \in \mathcal{X}' \\ H(x) & \text{if } x \notin \mathcal{X}' \end{cases}.$$

First, note that $\mathsf{Shrink}^{S^{B,H}}(x) = B(x)$. Also note that if $B, H$ are random functions, then $S^{B,H}(\cdot)$ is a random function. Thus, for any distinguisher $D$ (quantum or classical, computationally unbounded), we have that $\Pr\left[D^{\mathsf{Shrink}^A, A}() = 1\right] = \Pr\left[D^{B, S^{B,H}}() = 1\right]$. □

In the next few subsections, we will show how to remove the SRO model in the setting of weak reset indifferentiability, ultimately achieving domain shrinkage in the standard model with weak reset indifferentiability.

### 5.2 SRO Implies Weak SRS

Here, we show that indifferentiability with shared random oracles implies indifferentiability with shared random strings, in the weak indifferentiability setting. The idea is to simulate the random oracle using a $k$-wise independent hash function, which can be set as the shared random string. We note that [BDG20] employ a similar technique, but use a PRF instead, meaning their results require computational assumptions. Our Theorem 6 shows that such computational assumptions are unnecessary.

**Theorem 6.** *Let $\Gamma \in \{comp., stat., perfect\}, \Delta \in \{classical, quantum\}$. If $C^{\mathcal{A}}$ is weak $\Gamma$ $\Delta$ reset indifferentiable from $\mathcal{B}$ in the SRO model, then it is also weak $\Gamma$ $\Delta$ reset indifferentiable from $\mathcal{B}$ in the SRS model.*

*Proof.* The computational, statistical, and perfect settings are identical, and will be proved together. We first prove the classical case, the quantum case being a small modification that we describe at the end.

Let $D$ be a supposed distinguisher for reset indifferentiability, which we will interpret as a potential distinguisher in both the SRS and SRO models. By SRO indifferentiability, there exists sets $\mathcal{X}$, $\mathcal{Y}$ and a simulator $S^{B,H}$ satisfying Definition 3, meaning there exists a negligible $\epsilon$ such that

$$\left| \Pr_{A \leftarrow \mathcal{A}} \left[ D^{C^A, A}() = 1 \right] - \Pr_{B \leftarrow \mathcal{B}, H \leftarrow \mathcal{Y}^{\mathcal{X}}} \left[ D^{B, S^{B,H}}() = 1 \right] \right| \leq \epsilon \ .$$

Now, let $q_0$ be an upper bound on the number of queries $D$ makes, and $q_1$ an upper bound on the number of queries $S$ makes to $H$ on any call to $S$. Then $D^{B,S^{B,H}}()$ makes at most $k = q_0 q_1$ calls to $H$. Let $\mathcal{F}$ be a family of $k$-wise independent functions. Then

$$\Pr_{B \leftarrow \mathcal{B}, H \leftarrow \mathcal{Y}^{\mathcal{X}}} \left[ D^{B, S^{B,H}}() = 1 \right] = \Pr_{B \leftarrow \mathcal{B}, f \leftarrow \mathcal{F}} \left[ D^{B, S^{B,f}}() = 1 \right]$$

Our new simulator therefore sets $\mathcal{F}$ as the space of random strings, and $f$ the shared randomness. SRS security immediately follows.

For the quantum case, we just set $\mathcal{F}$ to be a family of $2k$-wise independent functions, and security follows from the following Lemma of Zhandry [Zha12b]:

**Lemma 2 ([Zha12b]).** *Let $\mathcal{F}$ to be a family of $2q$-wise independent functions from $\mathcal{X}$ to $\mathcal{Y}$. Then for any algorithm $D$ making at most $q$ quantum queries, $\Pr_{f \leftarrow \mathcal{F}}[D^f() = 1] = \Pr_{f \leftarrow \mathcal{Y}^{\mathcal{X}}}[D^f() = 1]$.*

This completes the proof of Theorem 6. □

### 5.3 SRS Often Implies Standard Weak Indifferentiability

Here, we show that SRS (and therefore SRO) indifferentiability often gives weak indifferentiability in the standard model. The intuition is to use the idealized object $\mathcal{A}$ itself to simulate the random string.

*Extractable Distributions.* Here, we define a notion of *extractability* for a distribution, which captures the ability to extract randomness from the function.

**Definition 4.** *A distribution $\mathcal{A}$ over functions $A : \mathcal{X} \to \mathcal{Y}$ is statistically classically extractable if, for any polynomial $\ell$ and any computationally unbounded distinguisher $D$ making a polynomial number of classical queries, there exists a deterministic polynomial time oracle-aided Turing machine $\mathsf{Ext}^A()$ which outputs $\ell$ bit strings, and a negligible function $\epsilon$ such that:*

$$\left| \Pr_{A\leftarrow\mathcal{A}, r\leftarrow\{0,1\}^\ell}\left[D^A(r)=1\right] - \Pr_{A\leftarrow\mathcal{A}}\left[D^A(\mathsf{Ext}^A())=1\right]\right| \leq \epsilon \ .$$

*In other words, $D$ cannot distinguish the output of $\mathsf{Ext}^A$ from random. We define computational, perfect, and quantum extractability analogously.*

We expect most idealized models of interest to be extractable. In particular, we demonstrate that random oracles are extractable, as is any idealized model that can build random oracles under *plain* (non-reset) indifferentiability.

**Theorem 7.** *Random oracles are perfectly classically and quantumly extractable.*

*Proof.* Our proof follows ideas from [BBM13], who show how to remove ephemeral (per query) randomness from "pseudo-deterministic" simulators. We generate randomness in the same way, but with a different application and additionally prove the quantum case. First, we will assume for simplicity that $A$ has $\ell$-bit outputs, which is without loss of generality since we can always trade off input and output length in a random oracle, the result potentially multiplying the number of queries by up to $\ell$ while being perfectly indifferentiable.

Then we have $\mathsf{Ext}^A()$ work as follows. For a parameter $k$ to be chosen latter, $\mathsf{Ext}$ arbitrarily (but deterministically) chooses $k$ distinct points $(x_i)_{i\in[k]}$, and outputs $r = \oplus_{i\in[k]} A(x_i)$. Since we require random oracles to have exponential-sized domains, there will always exist $k$ distinct points. To prove extractability, we first consider the classical case. We set $k = q+1$. Then any $q$-query algorithm $D$ cannot possibly query all the $x_i$. As such, at least one of the $A(x_i)$ values will be information-theoretically hidden from $D$, meaning $r = \oplus_{i\in[k]} A(x_i)$ is information-theoretically hidden. As such, $D$ cannot distinguish $r$ from random.

For the quantum case, more care is required since the distinguisher can query on superpositions of potentially all $x_i$, meaning we cannot argue any particular $A(x_i)$ is hidden. Instead, we use the following result of Zhandry [Zha15b]:

**Lemma 3 ([Zha15b], Theorem 5.1).** *Let $Q$ be a $q$-quantum query algorithm to $A$. Then $\Pr[Q^A() = \oplus_{i\in[k]} A(x_i)] \leq \lfloor k/(k-q)\rfloor/2^\ell$. In particular, if $q < k/2$, then the probability is at most $2^{-\ell}$.*

We now turn the very strong intractability of computing $r$ into the desired indistinguishability. Let $k = 2q + 1$ and let $D$ be a $q$-query distinguisher. Let $p_0$ be the probability $D$ outputs 1 when given $\oplus_{i\in[k]} A(x_i)$, and let $p_1$ be the probability $D$ outputs 1 when given a random $r \neq \oplus_{i\in[k]} A(x_i)$ as input. Suppose $p_0 \neq p_1$. In this case, assume without loss of generality that $p_0 > p_1$, by flipping the output bit of $D$ if necessary.

We construct $Q$ as follows: $Q^A()$ chooses a random $r$, and runs $b \leftarrow D^A(r)$. If $b = 1$, it outputs $r$; otherwise it chooses a new random $r'$ and outputs $r'$.

We now compute the probability $Q^A()$ outputs $\oplus_{i\in[k]} A(x_i)$. Conditioned on $r = \oplus_{i\in[k]} A(x_i)$, then $Q^A()$ outputs $r$ (and is hence correct) with probability $p_0$; otherwise it outputs a random $r'$, which is correct with probability $2^{-\ell}$. Conditioned on $r \neq \oplus_{i\in[k]} A(x_i)$, $Q^A()$ is only correct if it outputs $r'$ (which happens with probability $1 - p_1$) and $r'$ is correct (which has probability $2^{-\ell}$). Over, the probability $Q^A()$ is correct is then

$$\begin{aligned}
\Pr[Q^A() = \oplus_{i\in[k]} A(x_i)] &= \frac{1}{2^\ell}\left(p_0 + (1-p_0)\frac{1}{2^\ell}\right) + \frac{2^\ell - 1}{2^\ell}(1-p_1)\frac{1}{2^\ell} \\
&> \frac{1}{2^\ell}\left(p_0 + (1-p_0)\frac{1}{2^\ell}\right) + \frac{2^\ell - 1}{2^\ell}(1-p_0)\frac{1}{2^\ell} \\
&= \frac{1}{2^\ell}p_0 + \frac{1}{2^\ell}(1-p_0) = \frac{1}{2^\ell}
\end{aligned}$$

thus contradicting Lemma 3. □

Though not needed for our main results, we would also like to show that ideal ciphers are extractable. Classically, the same $\mathsf{Ext}$ from the proof of Theorem 7 also works for ideal ciphers. Quantumly, however, the situation is more difficult, in particular because do not know a suitable analog of Lemma 3 for the ideal cipher setting. While it is possible to directly prove that ideal ciphers are quantum extractable by carefully adapting known techniques, we will prove a more general theorem which shows that any ideal model which implies random oracles under indifferentiability is also extractable.

**Theorem 8.** *Let $\Gamma \in \{$comp., stat., perfect$\}$, $\Delta \in \{$classical, quantum$\}$. Suppose $\mathcal{A}$ is a distribution over functions such that there exists a construction $C^{\mathcal{A}}$ that is* strong $\Gamma$ $\Delta$ plain *indifferentiable from a random oracle. Then $\mathcal{A}$ is $\Gamma$ $\Delta$ extractable.*

*Proof.* We prove the classical statistical case, the quantum, perfect, and computational cases being essentially identical. Let $\ell$ be a polynomial and $D$ a potential distinguisher for the extractability of $\mathcal{A}$. Let $S$ be the universal simulator guaranteed by the *strong* (plain) indifferentiability of $C^{\mathcal{A}}$. Then consider the distinguisher $D_0^{\mathcal{B}} = D^{S^{\mathcal{B}}}$ for the extractability of the random oracle $\mathcal{B}$. By Theorem 7, there must exist an extraction procedure $\mathsf{Ext}_0^{\mathcal{B}}$ and negligible $\epsilon$ with

$$\left| \Pr_{B\leftarrow\mathcal{B}, r\leftarrow\{0,1\}^\ell}\left[D_0^B(r) = 1\right] - \Pr_{B\leftarrow\mathcal{B}}\left[D_0^B(\mathsf{Ext}_0^B()) = 1\right]\right| = 0 \ .$$

Remembering that $D_0^{\mathcal{B}} = D^{S^{\mathcal{B}}}$, we interpret $D^A(\mathsf{Ext}_0^B)$ and $D^A(r)$ as indifferentiability distinguishers for $C^{\mathcal{A}}$, meaning there exists a negligible $\epsilon, \epsilon'$ and

$$\left| \Pr_{B \leftarrow \mathcal{B}} \left[ D_0^B(\mathsf{Ext}_0^B()) = 1 \right] - \Pr_{A \leftarrow \mathcal{A}} \left[ D^A(\mathsf{Ext}_0^{C^A}()) = 1 \right] \right| \leq \epsilon$$
$$\left| \Pr_{B \leftarrow \mathcal{B}, r \leftarrow \mathcal{X}} \left[ D_0^B(r) = 1 \right] - \Pr_{A \leftarrow \mathcal{A}, r \leftarrow \mathcal{X}} \left[ D^A(r) = 1 \right] \right| \leq \epsilon' \ .$$

We now let $\mathsf{Ext}^{\mathcal{A}}() = \mathsf{Ext}^{C^{\mathcal{A}}}()$, and we conclude that

$$\left| \Pr_{A \leftarrow \mathcal{A}, r \leftarrow \{0,1\}^\ell} \left[ D^{\mathcal{A}}(r) = 1 \right] - \Pr_{A \leftarrow \mathcal{A}} \left[ D^A(\mathsf{Ext}^A()) = 1 \right] \right| < \epsilon + \epsilon' \ .$$

Thus $\mathsf{Ext}$ satisfies Definition 4. □

Looking ahead, in Section 6, we will prove that ideal ciphers can be used to construct random oracles that are sufficiently indifferentiable to apply Theorem 8. This means that ideal ciphers are extractable.

*Removing shared randomness for extractable sources.* We now show that, if the source is extractable, we can remove shared randomness in the weak indifferentiability setting.

**Theorem 9.** *Let $\Gamma \in \{comp., stat., perfect\}, \Delta \in \{classical, quantum\}$. If $C^{\mathcal{A}}$ is weak $\Gamma$ $\Delta$ reset indifferentiable from $\mathcal{B}$ in the SRS model, and if $\mathcal{B}$ is $\Gamma$ $\Delta$ extractable, then $C^{\mathcal{A}}$ is also weak $\Gamma$ $\Delta$ reset indifferentiable from $\mathcal{B}$ in the standard model.*

*Proof.* All six settings are essentially identical, so we prove the statistical classical case. Let $D$ be a supposed distinguisher for reset indifferentiability, which we will interpret as both a potential distinguisher in both the SRS and standard models. By SRS indifferentiability, there exists a set $\mathcal{X}$ and a simulator $S^B$ satisfying Definition 2, meaning there exists a negligible $\epsilon$ such that

$$\left| \Pr_{A \leftarrow \mathcal{A}} \left[ D^{C^A, A}() = 1 \right] - \Pr_{B \leftarrow \mathcal{B}, r \leftarrow \mathcal{X}} \left[ D^{B, S^B(\cdot\,;\,r)}() = 1 \right] \right| \leq \epsilon \ .$$

Consider the extractability distinguisher $E^B(r) := D^{B, S^B(\cdot\,;\,r)}()$ for $\mathcal{B}$. By the assumed extractability of $\mathcal{B}$, there exists an extraction procedure $\mathsf{Ext}$ and negligible $\delta$ such that

$$\left| \Pr_{B \leftarrow \mathcal{B}, r \leftarrow \mathcal{X}} \left[ D^{B, S^B(\cdot\,;\,r)}() = 1 \right] - \Pr_{B \leftarrow \mathcal{B}} \left[ D^{B, S^B(\cdot\,;\,r)}() = 1 : r = \mathsf{Ext}^B() \right] \right| \leq \delta \ .$$

We therefore define a new standard-model simulator $T^B(x) = S^B(x\ ;\ \mathsf{Ext}^B())$. The result is that

$$\left| \Pr_{A \leftarrow \mathcal{A}} \left[ D^{C^A, A}() = 1 \right] - \Pr_{A \leftarrow \mathcal{A}} \left[ D^{B, T^B}() = 1 \right] \right| \leq \epsilon + \delta$$

Thus establishing reset indifferentiability in the standard model. □

As an immediate corollary, we have:

**Corollary 2.** *For any $\mathcal{X}' \subseteq \mathcal{X}$, $\mathsf{Shrink}^{\mathsf{RO}}$ is weak statistical (classical and quantum) reset indifferentiable from a random oracle, in the standard model.*

*Remark 2.* It may seem odd that we can use extractability to prove *reset* indifferentiability, when Theorem 8 only needs *plain* indifferentiability to justify extractability. Note, however, that the actual indifferentiability simulator uses $\mathsf{Ext}$, which is indeed stateless. The simulator used to justify extractability only comes up as a hybrid in the security analysis, where it is okay to keep state.

### 5.4 Extensions

Here, we consider shared randomness beyond random oracles, namely a generalization to oracle distributions are *constructible* from random oracles.

**Definition 5.** *We say a distribution $\mathcal{F}$ is statistically classically* constructible *from $\mathcal{G}$ if there is a deterministic polynomial-time oracle-aided Turing machine $C$ such that, for any computationally unbounded distinguisher $D$ making a polynomial number of classical queries, there exists a negligible $\epsilon$ such that*

$$|\Pr_{F\leftarrow\mathcal{F}}[D^F()=1] - \Pr_{G\leftarrow\mathcal{G}}[D^{C^G}()=1]| \leq \epsilon$$

*We analogously define computational, perfect, and quantum constructibility.*

Note that constructibility does not give the distinguisher access to $G$, meaning plain indistinguishability suffices. Let $\Gamma \in \{\text{computational}, \text{statistical}, \text{perfectly}\}$ and $\Delta \in \{\text{classical}, \text{quantum}\}$. We note that constructibility has some basic composition properties:

- If $\mathcal{F}$ is $\Gamma$ $\Delta$ constructible from $\mathcal{G}$, and $\mathcal{G}$ is $\Gamma$ $\Delta$ constructible from $\mathcal{H}$, then $\mathcal{F}$ is $\Gamma$ $\Delta$ constructible from $\mathcal{H}$.
- Let $\mathcal{F}_1, \ldots, \mathcal{F}_n$ be distributions, and denote $(\mathcal{F}_1, \ldots, \mathcal{F}_n)$ denote the distribution on functions $(i, x) \to F_i(x)$ where $F_i \leftarrow \mathcal{F}_i$. If each $\mathcal{F}_i$ is $\Gamma$ $\Delta$ constructible from $\mathcal{G}_i$ for $i = 1, \ldots, n$, then $(\mathcal{F}_1, \ldots, \mathcal{F}_n)$ is $\Gamma$ $\Delta$ constructible from $(\mathcal{G}_1, \ldots, \mathcal{G}_n)$
- Let $\mathsf{RO}_1, \ldots, \mathsf{RO}_n$ be independent random oracles. Then $(\mathsf{RO}_1, \ldots, \mathsf{RO}_n)$ is perfectly classical and quantum constructible from appropriately-sized random oracles, by simple domain separation.

Next, we observe that existing results imply the constructibility of ideal ciphers from random oracles:

**Lemma 4.** *Ideal ciphers are perfectly quantumly and classically constructible from appropriately-sized random oracles.*

*Proof.* In the classical statistical case, we can use Luby-Rackoff [LR86]. Quantum Luby-Rackoff unfortunately is unknown since we need to handle inversion queries. Instead, we follow [Zha16], and use perfect shuffles. In particular, [GP07] shows the existence of a perfect random permutation from a random oracle, which therefore achieves perfect constructibility, even under quantum queries. □

**Corollary 3.** *Keyed random injections are perfectly quantumly and classically constructible from appropriately-sized random oracles.*

*Proof.* Keyed random injections are perfectly classically and quantumly constructible from keyed ideal ciphers, by simply padding the input. Then composition gives the desired result. □

*Generalizing Shared Randomness.* We now give our general definition.

**Definition 6.** *Let $\mathcal{F}$ be a distribution over functions. $C^{\mathcal{A}}$ is (strong statistical classical) reset indifferentiable from $\mathcal{B}$ in the Shared-$\mathcal{F}$ model if there exists a polynomial-time stateless oracle-aided simulator $S$ such that, for any oracle-aided Turing machine $D$ making at most a polynomial number of queries, there exists a negligible $\epsilon$ such that*

$$\left|\Pr_{A\leftarrow\mathcal{A}}\left[D^{C^A,A}()=1\right]-\Pr_{B\leftarrow\mathcal{B},f\leftarrow\mathcal{F}}\left[D^{B,S^{B,f}}()=1\right]\right|\leq\epsilon\ .$$

*We similarly define weak, computational, perfect, and quantum Shared-$\mathcal{F}$ models.*

**Lemma 5.** *Let $\Phi\in\{\text{strong},\text{weak}\}$, $\Gamma\in\{\text{computational},\text{statistical},\text{perfectly}\}$ and $\Delta\in\{\text{classical},\text{quantum}\}$. If $C^{\mathcal{A}}$ is $\Phi\ \Gamma\ \Delta$ reset indifferentiable from $\mathcal{B}$ in the Shared-$\mathcal{F}$ model, and $\mathcal{F}$ is $\Gamma\ \Delta$ constructible from $\mathcal{G}$, then $C^{\mathcal{A}}$ is also $\Phi\ \Gamma\ \Delta$ reset indifferentiable from $\mathcal{B}$ in the Shared-$\mathcal{G}$ model.*

## 6 Random Oracles from Ideal Ciphers

Here, we show how to build random oracles from ideal ciphers using weak reset indifferentiability. Concretely, we prove that an ideal cipher gives a random oracle with *strong* reset indifferentiability in the shared random oracle (SRO) model:

**Theorem 10.** *Let $\mathcal{A}$ be an ideal cipher. There exists a construction $C^{\mathcal{A}}$ that is strong statistical (classical and quantum) reset indifferentiable from a random oracle in the SRO model.*

We prove Theorem 10 in Section 6.1, but first show two corollaries:

**Corollary 4.** *Ideal ciphers are statistical (classical and quantum) extractable.*

*Proof.* By Lemma 1, $C^{\mathcal{A}}$ is strong statistical quantum *plain* indifferentiable in the *standard* model. The result then follows from Theorem 8. □

**Corollary 5.** *Let $\mathcal{A}$ be an ideal cipher. There exists a construction $C^{\mathcal{A}}$ that is* weak *statistical (classical and quantum) reset indifferentiable from a random oracle in the standard model.*

*Proof.* We apply Theorem 6 to Theorem 10 to get that $C^{\mathcal{A}}$ is weak statistical (classical and quantum) reset indifferentiable in the *SRS* model. Then we use the extractability of random oracles and Theorem 9 to conclude weak statistical (classical and quantum) reset indifferentiability in the standard model. □

### 6.1 The Pad-and-Truncate Construction

Our construction can be seen as the Sponge construction for 1-block messages. Fix real numbers $c, d \in (0,1)$. Let $A : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a keyed injection with inverse $A^{-1}$. Let $\mathcal{X}' \subseteq \mathcal{X}$ and $\mathcal{Y}' \subseteq \mathcal{Y}$ such that $|\mathcal{X}'| \leq |\mathcal{Y}|^c$ and $|\mathcal{Y}'| \leq |\mathcal{Y}|^d$. Assume for simplicity that $|\mathcal{Y}'|$ divides $|\mathcal{Y}|$, interpret $\mathcal{Y} = \mathcal{Y}' \times \mathcal{Z}$, and define $\mathsf{Proj}(y,z) = y$. Then define $\mathsf{PadTrunc}_{c,d}^{A,A^{-1}} : \mathcal{K} \times \mathcal{X}' \to \mathcal{Y}'$ as $\mathsf{PadTrunc}_{c,d}^{A,A^{-1}}(x) = \mathsf{Proj}(A(x))$. We now restate Theorem 10, using $\mathsf{PadTrunc}$:

**Theorem 10.** *For any constants $c, d \in (0,1)$ such that $c + d \leq 1$, $\mathsf{PadTrunc}_{c,d}^{\mathsf{IC}}$ is strongly shared randomness statistically (classically and quantumly) reset indifferentiable from a random oracle.*

### 6.2 The Simulator

In order to be consistent with $\mathsf{PadTrunc}_{c,d}$, our simulator needs to answer queries to $A(k,x)$ with $(B(k,x),z)$ for some $z$. At the same time, it needs to be able to answer queries to $A^{-1}(k,\ (B(k,x),z)\ )$ with $x \in \mathcal{X}'$. For all other queries, the simulator needs to answer in a way that "looks like" a random keyed injection.

The central difficulty is that, by virtue of having a stateless simulator, we cannot answer these queries lazily, and we cannot "remember" how previous queries were answered. This particularly represents a problem for answering $A^{-1}(k,\ (B(k,x),z)\ )$ queries, since we somehow have to recover $x$, even though $B$ is a random oracle which would hide $x$. Our solution is to do the following. Following Lemma 5, it suffices to have our simulator work in the Shared-$(\mathsf{RI},\mathsf{RI})$ model, having access to random keyed injections $I : \mathcal{K} \times \mathcal{X}' \to \mathcal{Z}, Q : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, and their inverses $I^{-1}, Q^{-1}$. The simulator $S^{B,I,Q}$ answers $A$ and $A^{-1}$ queries as $P$ and $P^{-1}$ respectively, where:

$$P(k,x) = \begin{cases} (B(k,x), I(k,x)) & \text{if } x \in \mathcal{X}' \\ Q(k,x) & \text{otherwise} \end{cases} \tag{2}$$

$$P^{-1}(k,(w,z)) = \begin{cases} x & \text{if } w = B(k,x) \text{ where } x = I^{-1}(k,z) \\ Q^{-1}(k,(w,z)) & \text{otherwise} \end{cases} \tag{3}$$

### 6.3 Indifferentiability Proof

We now need to prove that this simulator is indistinguishable from the case where $A, A^{-1}$ are uniformly random permutations, and $B = \mathsf{PadTrunc}_{c,d}^{A,A^{-1}}$.

First, we show that without loss of generality we can focus on the key-less case ($|\mathcal{K}| = 1$). This follows immediately from a generalization of a result of Zhandry [Zha12a], which we prove in the full version [Zha21]:

**Lemma 6.** *Let $D_0, D_1$ be distributions over oracles from $\mathcal{X}$ to $\mathcal{Y}$. Let $O_1, O_2$ be distributions on oracles from $\mathcal{K} \times \mathcal{X}$ to $\mathcal{Y}$, where for each $k$, $O_b(k,\cdot)$ is sampled from $D_b$. Suppose there exists a $q$ quantum query algorithm $A$ with access to an oracle $O_0$ or $O_1$ such that $|\Pr[A^{O_0}() = 1] - \Pr[A^{O_1}() = 1]| = \epsilon$. Then there is a quantum algorithm $B$ such that $|\Pr[B^{D_0} = 1] - \Pr[B^{D_1} = 1]| \geq \Omega(\epsilon^2/q^3)$.*

Now let $D$ be a (potentially quantum) distinguisher making polynomially-many queries in the keyless case, and define several hybrid experiments:

- **Hybrid 0.** This is the "Ideal World" where $B$ is a random oracle and $A, A^{-1}$ are set to $P, P^{-1}$ as defined in our simulator in Lines 2 and 3, with $I, Q$ being random (key-less) injections. Let $p_0$ be the probability $D$ outputs 1.
- **Hybrid 1.** This is the same as **Hybrid 0**, except that we replace $D$'s queries to $B(x)$ with $\mathsf{PadTrunc}_{c,d}^{P,P^{-1}}(x)$. Let $p_1$ be the probability $D$ outputs 1. Note that $\mathsf{PadTrunc}$ only makes $A$ queries on inputs $x \in \mathcal{X}'$, which $S$ answers as $(B(x), I(x))$. Thus $\mathsf{PadTrunc}_{c,d}^{P,P^{-1}}(x) = B(x)$, and therefore the distribution of oracles seen by $D$ is identical in **Hybrids 0 and 1**. Thus $p_0 = p_1$.
- **Hybrid 2.** This is the "Real World", where $A, A^{-1}$ are a random (keyed) injection and its inverse, and $B(k, x) = \mathsf{PadTrunc}_{c,d}^{A,A^{-1}}(x)$. Equivalently, **Hybrid 2** is the same as **Hybrid 1**, except that $P, P^{-1}$ in Equations 2 and 3 are replaced by a random keyed injection $A$ and its inverse $A^{-1}$. Let $p_2$ be the probability $D$ outputs 1.

It remains to show that $|p_2 - p_1|$ is negligible, which constitutes the bulk of the indifferentiability proof. For this, the following claim suffices:

**Lemma 7.** *For any distinguisher $E$ making at most a polynomial number of classical or quantum queries, we have that $|\Pr[E^{P,P^{-1}}() = 1] - \Pr[E^{A,A^{-1}}() = 1]|$ is negligible, where $A, A^{-1}$ are a random (keyless) injection and its inverse, and $P, P^{-1}$ are as in Equations 2 and 3, with $I, Q$ are random keyed injections.*

Lemma 7 proves Theorem 10 by letting $E^{A,A^{-1}}() = D^{\mathsf{PadTrunc}_{c,d}^{A,A^{-1}},A,A^{-1}}()$. We now prove Lemma 7.

*Proof.* Classically, proving this is possible using lazy sampling. However, ultimately we will also want to prove the indistinguishability under quantum queries. This is somewhat more challenging, and requires a more careful proof, given limitations of known techniques. We will therefore structure the proof in a way that allows us to prove both classical and quantum indistinguishability.

Let $E$ be a potential distinguisher. We prove the indistinguishability through another sequence of hybrids:

- **Hybrid $\alpha$.** Here we give $E$ the oracles $A, A^{-1}$ that are a uniformly random (keyless) injection and its inverse. Define $p_\alpha$ as the probability $E$ outputs 1.
- **Hybrid $\beta$.** Here, we sample a uniformly random injection $J : \mathcal{X}' \to \mathcal{Y}$. We give $E$ the oracles $A_\beta, A_\beta^{-1}$, where

$$A_\beta(x) = \begin{cases} A(\, J^{-1}(\, A(x)\,)\,) & \text{if } A(x) \in \mathsf{Img}(J), x \notin \mathcal{X}' \\ A(x) & \text{otherwise} \end{cases}$$

$$A_\beta^{-1}(y) = \begin{cases} A^{-1}(J^{-1}(A^{-1}(y))) & \text{if } y \in \mathsf{Img}(J), A^{-1}(y) \notin \mathcal{X}' \\ A^{-1}(y) & \text{otherwise} \end{cases}$$

Here, $\mathsf{Img}(J)$ is the set of images of $J$. Note the $J^{-1}$ in both the definition of $A_\beta$ and $A_\beta^{-1}$. Let $p_\beta$ be the probability $E$ outputs 1.

Note that $A_\beta, A_\beta^{-1}$ are identical to $A, A^{-1}$, except on points determined by the sparse image of $J$. Since $J$ is random, these points should be hidden from the view of $E$. Indeed, it is straightforward that, in the classical case, such points will only be queried with negligible probability, and in the absence of querying these points the distributions are identical.

In the quantum case, we have to work slightly harder. We prove the following in the full version [Zha21], which follows from known quantum techniques:

**Lemma 8.** *Let $D$ be a distribution over subsets $V$ of $\mathcal{X}$ such that each element in $\mathcal{X}$ is placed in $V$ with probability $\epsilon$ (not necessarily independently). Consider any quantum algorithm $E$ making $q$ queries to an oracle $O$ with domain $\mathcal{X}$, and let $p_0$ be the probability $E^O()$ outputs 1. Let $O'$ that is identical to $O$, except that on a set $V$ sampled from $D$, $O'$ is changed arbitrarily. Let $p_1$ be the probability $E^{O'}()$ outputs 1. Then $|p_0 - p_1| < O(q\sqrt{\epsilon})$.*

The random injection $J$ defines such a set $V$ where each input to $A$ or $A^{-1}$ is placed in the changed set with probability $|\mathcal{X}'|/|\mathcal{Y}| = |\mathcal{Y}|^{-(1-c)}$. Therefore $|p_\beta - p_\alpha| < O(q|\mathcal{Y}|^{-(1-c)/2})$, which is negligible.

– **Hybrid $\gamma$.** Here, we sample $J, A, A^{-1}, A_\beta, A_\beta^{-1}$ as in **Hybrid** $\beta$. Let $K : \mathcal{X}' \to \mathcal{Y}$ be the restriction of $A$ to $\mathcal{X}'$: $K(x) = A(x)$. Also define $Q(x) = A_\beta(x)$ for $x \notin \mathcal{X}'$. The values $Q(k, x)$ for $x \in \mathcal{X}'$ are random and distinct values from the set $\mathcal{Y} \setminus \{A_\beta(x) : x \notin \mathcal{X}'\}$. Plugging in the definition of $A_\beta, K$, this gives

$$Q(x) = \begin{cases} K(J^{-1}(A(x))) & \text{if } A(x) \in \mathsf{Img}(J), x \notin \mathcal{X}' \\ A(x) & \text{if } A(x) \notin \mathsf{Img}(J), x \notin \mathcal{X}' \end{cases}.$$

We then give the adversary the oracles $A_\gamma, A_\gamma^{-1}$ defined as

$$A_\gamma(x) = \begin{cases} K(x) & \text{if } x \in \mathcal{X}' \\ Q(x) & \text{otherwise} \end{cases}$$

$$A_\gamma^{-1}(y) = \begin{cases} K^{-1}(y) & \text{if } y \in \mathsf{Img}(K(\cdot)) \\ Q^{-1}(y) & \text{otherwise} \end{cases}$$

Let $p_\gamma$ be the probability $E$ outputs 1. Plugging in the definitions of $Q, K$, we see that $A_\gamma = A_\beta, A_\gamma^{-1} = A_\beta^{-1}$. Therefore, $p_\gamma = p_\beta$.

Note that in **Hybrid** $\gamma$, $Q$ is a uniformly random keyless permutation, and $K$ is a uniformly random keyless injection.

– **Hybrid $\delta$.** Now give $E$ the oracles $A_\gamma, A_\gamma^{-1}$, except where $K$ is chosen as $K(x) = (B(x), I(x))$, $B$ is a random function, and $I(x)$ is a random keyless injection. Note that the result is equivalent to the oracles $P, P^{-1}$ defined as in Equations 2 and 3. Let $p_\delta$ be the probability $E$ outputs 1.

It remains to show that $p_\gamma$ is close to $p_\delta$. Since $Q$ is identically distributed in both hybrids, it suffices to prove that the distribution over $K$ in the two hybrids is indistinguishable:

**Lemma 9.** *Fix $c, d \in (0,1)$, and let $\mathcal{X}', \mathcal{Y}', \mathcal{Z}, \mathcal{Y}$, $\mathcal{Y} = \mathcal{Y}' \times \mathcal{Z}$, be sets such that $|\mathcal{X}| \leq |\mathcal{Y}|^c$ and $|\mathcal{Y}'| \leq |\mathcal{Y}|^d$. Write $K : \mathcal{X}' \to \mathcal{Y}$ as $K(x) = (B(x), I(x))$ for $B : \mathcal{X}' \to \mathcal{Y}'$ and $I : \mathcal{X}' \to \mathcal{Z}$. Then for any adversary making $q$ classical or quantum queries to $K$ and its inverse, the following two distributions are indistinguishable:*

- *$K$ is chosen as a random keyless injection*
- *$I$ is a random keyless injection, and $B$ is a random function.*

*Proof.* In the classical case, this is straightforward: the only way an adversary can distinguish is by finding $x_0, x_1$ such that $I(x_0) = I(x_1)$, which cannot happen in the case where $I$ is injective. To prove that such tuples are infeasible to find, we rely on the fact that the adversary cannot make inverse queries on valid images (whp), except on values that were the result of prior forward queries.

In the quantum setting, what makes proving this non-trivial is that the attacker has query access to both $K$ *and* $K^{-1}$, whereas the vast majority of the quantum literature does not consider inversion queries. In order to prove security, then, we carefully embed an instance of a problem that does *not* use inversion queries, and then rely on known quantum complexity techniques to prove the hardness of the inversion-less problem.

We first consider the case where $c < d$. The reason this case is easier is that we can switch from using $I(x)$ to recover $x$ to using $B(x)$ to recover $x$. Then since we do not need to query $I^{-1}$, we can rely on known quantum query lower bound techniques to switch to $I$ being random.

To prove indistinguishability in the $c < d$ case, we define a few more hybrids.

- **Hybrid i.** This hybrid sets $K : \mathcal{X}' \to \mathcal{Y}$ to be a uniformly random (keyless) injection. Let $p_i$ be the probability of outputting 1.
- **Hybrid ii.** This hybrid sets $K$ to be a random *function*. The problem with $K$ being a uniformly random function is that there might be collisions, meaning the inverse is not well defined. We define $K^{-1}(y)$ to be $x$ is there is a *unique* $x$ such that $K(x) = y$. Otherwise, if there are 0 or $\geq 2$ solutions, $K^{-1}(y) = \perp$. Let $p_{ii}$ be the probability of outputting 1.
  Since $c < d$ and $c + d \leq 1$, we have that $2c < 1$. As such, a random function is an injection with overwhelming probability by a union bound. Thus $|p_i - p_{ii}| \leq O(|\mathcal{Y}|^{-(1-2c)})$.
  Note here that if we write $K(x) = (B(x), I(x))$, then $B, I$ are independent uniform random functions.
- **Hybrid iii.** Here, we change how we answer $K^{-1}(w, z)$ queries. Rather than directly computing the inverse (supposing it exists and is unique), we instead compute $L_w := \{x : B(x) = w\}$, and then for each $x \in L_w$, we check if $I(x) = z$ by querying $I$. To bound the number of queries to $I$, we abort if $|L_w| > r$, for some parameter $r$. Let $p_{iii}$ be the probability of outputting 1.
  By standard balls-and-bins arguments, for each $w \in \mathcal{Y}'$, $L_w$ is at most $r$, except with probability $\binom{|\mathcal{X}'|}{r}|\mathcal{Y}'|^{-r} \leq |\mathcal{Y}|^{-(d-c)}$. Union bounding over all $w$ gives that $\max_w |L_w| \leq r$ except with probability $\leq |\mathcal{Y}|^{d-(d-c)r}$. Setting $r = O(1)$, this bound becomes $|\mathcal{Y}|^{-1}$. In the case all $L_w$ have size at most

$r$, there are no aborts and inverse procedure outputs the same value as in **Hybrid ii**. Thus $|p_{ii} - p_{iii}| \leq |\mathcal{Y}|^{-1}$. Moreover, the number of queries made to $I$ for each $K^{-1}$ query is at most a polynomial.
- **Hybrid iv.** Here, we change $I$ to be a keyless injection, and let $p_{iv}$ be the probability of outputting 1. If the adversary makes $q$ queries, we ultimately make $O(q)$ queries to $I$ (and no queries to $I^{-1}$). We can use the indistinguishability of random functions from random injections [AS04, Zha15a] to bound $|p_{iii} - p_{iv}| \leq O(q^3/|\mathcal{Z}|) = O(q^3/|\mathcal{Y}|^{1-d})$, which is negligible.

This completes the case $c < d$. We now extend to all $c, d > 0$ such that $c + d \leq 1$. The problem with the above proof is that the output of $B$ is no longer large enough to uniquely decode $x$. Nevertheless, we show how to embed an instance of the problem for $c' < d'$ into the general case, thereby proving security.

Let $c', d' \in (0, 1)$ be constants to be chosen later. Write $\mathcal{X}' = \mathcal{W} \times \mathcal{X}''$ and $\mathcal{Z} = \mathcal{W} \times \mathcal{Z}'$ for $|\mathcal{X}''| = |\mathcal{Y}|^{c'd/d'}, |\mathcal{W}| = |\mathcal{Y}|^{c-c'd/d'}, |\mathcal{Z}'| = |\mathcal{Y}|^{d/d'-d}$. Since $\mathcal{Z} = \mathcal{W} \times \mathcal{Z}'$, we must have $d' = d(1-c')/(1-c)$. Moreover, for the sizes of the sets involved to be non-negative, we must have $c' \leq c$, which implies $d' \geq d$.

We will sample $K$ as follows:

- First choose random keyless permutations $W, W' : (\mathcal{W} \times \mathcal{Z}') \to (\mathcal{W} \times \mathcal{Z}')$.
- Next, choose a *keyed* function $K' : \mathcal{W} \times \mathcal{X}'' \to \mathcal{Y}' \times \mathcal{Z}'$
- Set $K(x)$ to be the following: Let $x' = W'(x)$ and write $x' = (\eta, \mu) \in \mathcal{W} \times \mathcal{X}''$. Then compute $(\zeta, \tau) \leftarrow K'(\eta, \mu) \in \mathcal{Y}' \times \mathcal{Z}'$. Then output $(\zeta, W(\eta, \tau))$.

It is straightforward that, if $K'$ is a random keyed injection, then $K$ is a random keyed injection. On the other hand, suppose for any $\eta$, the mapping under $K'$ of $\mu \mapsto \tau$ was a random injection whereas the mapping $\mu \mapsto \zeta$ was a random function. Then it is straightforward that $K$ satisfies the distribution for **Hybrid iv**. Thus, proving the indistinguishability for the two cases of $K$ reduces to proving the indistinguishability for the two cases of $K'$. By applying Lemma 6, we can further reduce to the keyless case and ignore $\eta$. Since the range of $K'$ has size $|\mathcal{Y}|^{d/d'}$, we have that $K'$ is an instance of Lemma 9 with parameters $c', d'$. Choose an arbitrary $c' \leq c$ such that $d' = d(1-c')/(1-c) > c'$, which is equivalent to $c' < d/(1+d-c)$. We can then invoke the $c < d$ case of Lemma 9 as proved above on $K'$, obtaining the indistinguishability of the two settings. □

This completes the proof of Lemma 7. Putting everything together, this completes the proof of Theorem 10. □

### 6.4 On Necessary Shrinkage

Our positive result works for any $c + d \leq 1$. Here, we show that this is tight.

**Theorem 11.** *For any constants $c, d > 0$ such that $c + d > 1$, if $A$ is a random permutation, then $\mathsf{PadTrunc}_{c,d}^{A,A^{-1}}$ is not even weak computational (classical or quantum) reset indifferentiable from a random oracle.*

*Proof.* For simplicity, we focus on the keyless case ($s = 0$), which is without loss of generality. The intuition behind the proof is that the simulator, when answering queries of the form $A^{-1}(B(x), z)$, cannot invert $B$ to recover $x$. It must therefore recover $x$ from $z$. But this is only possible if $|z| \geq |x|$.

Consider the distinguisher $D$, which chooses a random $x \in \mathcal{X}'$, and runs $(w, z) \leftarrow A(x) \in \mathcal{Y}' \times \mathcal{Z}$. Then it runs $x' \leftarrow A^{-1}(w, z)$ and $w' \leftarrow B(x')$ (assuming $x' \in \mathcal{X}'$), and outputs 1 if and only if $w' = w, x' = x$. Consider a supposed simulator $S^B$ for $D$, where we write $S_0^B, S_1^B$ for the simulator's responses to $A$ and $A^{-1}$ queries, respectively. We have that there exists a negligible $\epsilon$ such that

$$\Pr\left[D^{B,S_0^B,S_1^B}() = 1\right] \geq 1 - \epsilon \ .$$

We turn $S^B$ into an algorithm $U^B(w)$, which finds an $x$ such that $B(x) = w$. $U^B(w)$ works as follows: choose a random $z^* \in \mathcal{Z}$, and output $x \leftarrow S_1^B(w, z^*)$.

*Claim.* For a random $x \in \mathcal{X}'$, $\Pr[U^B(B(x)) = x] \geq (1 - \epsilon)/|\mathcal{Y}|^{1-\max(c,d)}$.

*Proof.* Imagine running $D$ on a random $x \in \mathcal{X}'$. We therefore know that, with probability at least $1 - \epsilon$, the following are both true: (1) $S_0^B(x)$ outputs $(B(x), z)$ for some $z$, and (2) $S_1^B(z, w) = x \in \mathcal{X}'$. We will therefore say that $x$ is "good" if the above both hold; there are at least $(1 - \epsilon)|\mathcal{X}'|$ good $x$. In the case $c \leq d$, suppose that $x$ is good. Then $U^B(B(x))$ will successfully invert provided $z^* = z$, which occurs with probability $|\mathcal{Y}|^{-(1-d)}$.

In the case $c > d$, then there will be multiple good $x$ for each $w$. Consider the set of good $x' \in \mathcal{X}'$ such that $B(x') = w$, and let $z'$ be the associated value outputted by $S_0^B(x')$. Let $p_w$ be the number of such $x'$. Then as long as $z^*$ is equal to *any* $z'$ for a good $x'$, $U^B(w)$ will output $x'$, a pre-image of $w$. Thus, the probability of success for a given $w$ is at least $p_w|\mathcal{Y}|^{-(1-d)}$. Since the total number of good $x'$ is $(1 - \epsilon)|\mathcal{X}'|$, the expectation of $p_w$ is $(1 - \epsilon)|\mathcal{X}'|/|\mathcal{Y}'| = (1 - \epsilon)|\mathcal{Y}|^{c-d}$, meaning $B$ succeeds with probability $(1 - \epsilon)|\mathcal{Y}|^{-(1-c)}$. □

We now contrast Claim 6.4 with the (quantum) hardness of pre-image search:

**Lemma 10 ([BBBV97]).** *For any $q$ quantum query algorithm $A$ making queries to a random function $O : |\mathcal{X}| \to |\mathcal{Y}|$, $\Pr_{x \leftarrow \mathcal{X}}[O(\ A^O(\ O(x)\ )\ ) = O(x)] \leq O(q^2/\min(|\mathcal{X}|, |\mathcal{Y}|))$. In other words, a random oracle is quantum one-way* [4].

This shows that no $q$-query (quantum) algorithm can invert $B$ except with probability at most $O(q^2|\mathcal{Y}|^{-\min(c,d)})$. We thus have $q^2 \geq \Omega(|\mathcal{Y}|^{\min(c,d)+\max(c,d)-1}) = \Omega(|\mathcal{Y}|^{c+d-1}) = |\mathcal{Y}|^{\Omega(1)}$ (since $c + d > 1$), which is exponential. □

## 7 Post-Quantum Groups

Here, we demonstrate that generic groups are strongly reset indifferentiability from random injections in the quantum setting.

[4] Note that [BBBV97] phrase their result as finding a marked item in a list. Nevertheless, the statement of their result and its proof can be rephrased as in Lemma 10.

**Theorem 12.** *Let* GG *be a generic group of order* $p$ *and label space* $\{0,1\}^n$. *Then the labeling function for* GG*, namely* $L$*, is strongly statistical quantum reset indifferentiable from a (keyless) random injection* $I : \{0,1\}^{\log p} \to \{0,1\}^n$.

*Proof.* We use Shor's algorithm [Sho94] to invert the labeling function. We can simulate the group operations by inverting the labeling function, performing the group operation in $\mathbb{Z}_p$, and then re-applying the labeling function. □

### 7.1 Instantiations and Applications

We can instantiate the generic group using either subgroups of the multiplicative group of finite fields, or over elliptic curves. Then, applying the pad-and-truncate construction, we obtain a plausible post-quantum random oracle. We briefly discuss the case of finite fields. Let $q$ be a prime and $g$ an element generating a large subgroup of $\mathbb{Z}_q^*$. As we do not need discrete logarithms to be hard, the order of $g$ does not seem to matter, and $g$ can even be a generator of $\mathbb{Z}_q^*$. Let $g_i = g^{2^i} \bmod q$. Then $g^a \bmod q = \prod_{i=0}^{n-1} g_i^{a_i}$, where $a_i$ is the $i$th binary bit of $a$. Our pad-and-truncate construction is then $a \mapsto (\prod_{i=0}^{n-1} g_i^{a_i} \bmod q) \bmod r$, for some sufficiently small $r$, giving a simple plausible a post-quantum random oracle.

*Key-less classical permutations.* One limitation of the above is that the generic group is only *quantumly* equivalent to a key-less injection, requiring Shor's algorithm to perform inverses. However, an easy fix is to make the discrete log classically easy, by having the group order be smooth. Let $q$ be such that $q-1$ has all small prime factors. Then computing discrete logs in $\mathbb{Z}_q^*$ is even classically easy by solving discrete log mod each of the factors of $q-1$, and then Chinese Remaindering. Our labeling function maps $\mathbb{Z}_{q-1} \to \mathbb{Z}_q^*$; this can be turned into a permutation by simply subtracting 1 from the final result.

## References


AR16. Gorjan Alagic and Alexander Russell. Quantum-secure symmetric-key cryptography based on hidden shifts. Cryptology ePrint Archive, Report 2016/960, 2016. http://eprint.iacr.org/2016/960.

AS04. Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, Jul 2004.

AY20. Shweta Agrawal and Shota Yamada. Optimal broadcast encryption from pairings and LWE. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 13–43. Springer, Heidelberg, May 2020.

BBBV97. Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, Oct 1997.

BBG05. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer, Heidelberg, May 2005.

BBM13. Paul Baecher, Christina Brzuska, and Arno Mittelbach. Reset indifferentiability and its consequences. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 154–173. Springer, Heidelberg, December 2013.

BDF+11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011.

BDG20. Mihir Bellare, Hannah Davis, and Felix Günther. Separate your domains: NIST PQC KEMs, oracle cloning and read-only indifferentiability. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 3–32. Springer, Heidelberg, May 2020.

BFF+14. Gilles Barthe, Edvard Fagerholm, Dario Fiore, John C. Mitchell, Andre Scedrov, and Benedikt Schmidt. Automated analysis of cryptographic assumptions in generic group models. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 95–112. Springer, Heidelberg, August 2014.

BMZ19. James Bartusek, Fermi Ma, and Mark Zhandry. The distinction between fixed and random generators in group-based assumptions. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 801–830. Springer, Heidelberg, August 2019.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

BSW07. John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy*, pages 321–334. IEEE Computer Society Press, May 2007.

CDMP05. Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård revisited: How to construct a hash function. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 430–448. Springer, Heidelberg, August 2005.

CLMQ20. Yilei Chen, Alex Lombardi, Fermi Ma, and Willy Quach. Does fiat-shamir require a cryptographic hash function? Cryptology ePrint Archive, Report 2020/915, 2020. https://eprint.iacr.org/2020/915.

CMSZ19. Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. Quantum lazy sampling and game-playing proofs for quantum indifferentiability. Cryptology ePrint Archive, Report 2019/428, 2019. https://eprint.iacr.org/2019/428.

CPS08. Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 1–20. Springer, Heidelberg, August 2008.

Cza21. Jan Czajkowski. Quantum indifferentiability of sha-3. Cryptology ePrint Archive, Report 2021/192, 2021. https://eprint.iacr.org/2021/192.

Des00. Anand Desai. The security of all-or-nothing encryption: Protecting against exhaustive key search. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 359–375. Springer, Heidelberg, August 2000.

DFM20. Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In Daniele Micciancio

and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 602–631. Springer, Heidelberg, August 2020.

DFMS19. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 356–383. Springer, Heidelberg, August 2019.

DGHM13. Gregory Demay, Peter Gaži, Martin Hirt, and Ueli Maurer. Resource-restricted indifferentiability. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 664–683. Springer, Heidelberg, May 2013.

DHZ14. Ivan Damgård, Carmit Hazay, and Angela Zottarel. Short paper on the generic hardness of ddh-ii. 2014.

DRRS09. Yevgeniy Dodis, Leonid Reyzin, Ronald L. Rivest, and Emily Shen. Indifferentiability of permutation-based compression functions and tree-based modes of operation, with applications to MD6. In Orr Dunkelman, editor, *FSE 2009*, volume 5665 of *LNCS*, pages 104–121. Springer, Heidelberg, February 2009.

DS16. Yuanxi Dai and John P. Steinberger. Indifferentiability of 8-round Feistel networks. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 95–120. Springer, Heidelberg, August 2016.

GP07. Louis Granboulan and Thomas Pornin. Perfect block ciphers with small blocks. In Alex Biryukov, editor, *FSE 2007*, volume 4593 of *LNCS*, pages 452–465. Springer, Heidelberg, March 2007.

HKT11. Thomas Holenstein, Robin Künzler, and Stefano Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 89–98. ACM Press, June 2011.

KSS+20. Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shifeng Sun. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 703–728. Springer, Heidelberg, May 2020.

LAMP12. Atul Luykx, Elena Andreeva, Bart Mennink, and Bart Preneel. Impossibility results for indifferentiability with resets. Cryptology ePrint Archive, Report 2012/644, 2012. http://eprint.iacr.org/2012/644.

LR86. Michael Luby and Charles Rackoff. How to construct pseudo-random permutations from pseudo-random functions (abstract). In Hugh C. Williams, editor, *CRYPTO'85*, volume 218 of *LNCS*, page 447. Springer, Heidelberg, August 1986.

LZ19. Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 326–355. Springer, Heidelberg, August 2019.

Mit14. Arno Mittelbach. Salvaging indifferentiability in a multi-stage setting. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 603–621. Springer, Heidelberg, May 2014.

MRH04. Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle

methodology. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer, Heidelberg, February 2004.

RSS11. Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 487–506. Springer, Heidelberg, May 2011.

RST01. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, Heidelberg, December 2001.

Sho94. Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994.

Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997.

TU16. Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 192–216. Springer, Heidelberg, October / November 2016.

Unr15. Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 755–784. Springer, Heidelberg, April 2015.

YZ20. Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. Cryptology ePrint Archive, Report 2020/1270, 2020. https://eprint.iacr.org/2020/1270.

Zha12a. Mark Zhandry. How to construct quantum random functions. In *53rd FOCS*, pages 679–687. IEEE Computer Society Press, October 2012.

Zha12b. Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, Heidelberg, August 2012.

Zha15a. Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Info. Comput.*, 15(7-8):557–567, May 2015.

Zha15b. Mark Zhandry. Quantum oracle classification - the case of group structure, 2015.

Zha16. Mark Zhandry. A note on quantum-secure PRPs. Cryptology ePrint Archive, Report 2016/1076, 2016. http://eprint.iacr.org/2016/1076.

Zha19. Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, Heidelberg, August 2019.

Zha21. Mark Zhandry. Redeeming reset indifferentiability and applications to post-quantum security. Cryptology ePrint Archive, Report 2021/288, 2021. https://eprint.iacr.org/2021/288.

ZZ21. Mark Zhandry and Cong Zhang. The relationship between idealized models under computationally bounded adversaries. Cryptology ePrint Archive, Report 2021/240, 2021. https://eprint.iacr.org/2021/240.