Classical Proofs for the Quantum Collapsing Property of Classical Hash Functions

Serge Fehr[1,2](✉)

[1] CWI, Amsterdam, The Netherlands
serge.fehr@cwi.nl
[2] Mathematical Institute, Leiden University, Leiden, The Netherlands

Abstract. Hash functions are of fundamental importance in theoretical and in practical cryptography, and with the threat of quantum computers possibly emerging in the future, it is an urgent objective to understand the security of hash functions in the light of potential future quantum attacks. To this end, we reconsider the *collapsing property* of hash functions, as introduced by Unruh, which replaces the notion of *collision resistance* when considering quantum attacks. Our contribution is a formalism and a framework that offers significantly simpler proofs for the collapsing property of hash functions. With our framework, we can prove the collapsing property for hash domain extension constructions entirely by means of decomposing the iteration function into suitable elementary composition operations. In particular, given our framework, one can argue *purely classically* about the quantum-security of hash functions; this is in contrast to previous proofs which are in terms of sophisticated quantum-information-theoretic and quantum-algorithmic reasoning.

1 Introduction

Background. Given the threat of possible future quantum computing capabilities, it is an important and urgent objective to evaluate the security of classical cryptographic schemes against quantum attacks. There are different places where security can break down when using quantum computing techniques to attack a cryptographic scheme that was designed to withstand standard classical attacks. The most prominent place is the computational hardness *assumption*, which is typically well justified to hold for classical models of computation but may be false with respect to quantum computation. Another place is the security *proof*, which may use techniques that fail to work in the context of a quantum attacker, like proofs that rely on rewinding techniques. Finally, another place where things can go wrong is the security *definition*, which may not capture anymore what it is supposed to capture when allowing quantum attacks.

An example of the latter is the computational binding property of a commitment scheme. Our intuitive understanding of what a commitment should achieve is that once a commitment is "on the table" there should be no freedom left for the (computationally bounded) committer in choosing the value to which he can

A. Beimel and S. Dziembowski (Eds.): TCC 2018, LNCS 11240, pp. 315–338, 2018.
https://doi.org/10.1007/978-3-030-03810-6_12

open the commitment. The formal definition of the binding property expresses this requirement by demanding that no (computationally bounded) dishonest committer should be able to open a commitment in two distinct ways. While for classical committers this captures precisely what we want, it fails to do so for quantum committers. Indeed, a quantum committer can potentially open a commitment to *one* value that he *freely chooses* after he has put the commitment "on the table", without contradicting the requirement of being unable to produce *two* distinct openings; this is because producing the opening information may involve a destructive quantum measurement that can only be applied once.

We stress that being able to open a given commitment to an arbitrary value that one can freely choose renders a commitment scheme useless in essentially all applications. So, when considering the security of commitment schemes against quantum attacks, it is essential that one uses a *stronger notion of security* than the standard computational binding property extended to quantum attackers.

A similar and related example is the collision resistance of hash functions. Also here, in the presence of a quantum attacker, the standard formal requirement that it should be computationally hard to produce two colliding inputs does not capture our intuitive understanding of a hash value as acting as a "fingerprint" that removes any freedom in the message to which it fits. As such, also here, when considering security against quantum attacks, the standard security notion, i.e. collision resistance, needs to be replaced by something stronger.

The Collapsing Property. Unruh [5] proposed the notion of *collapsing*; in the context of commitment schemes as a counterpart for the computational binding property when considering quantum attacks, and in the context of hash functions as a counterpart for collision resistance. In essence, for hash functions, the collapsing property requires that for any computationally bounded adversary that output a hash value together with a *quantum superposition* of corresponding preimages, he should not be able to tell if the superposition gets measured or not. The details of the notion, and why it indeed restores the right security properties when considering quantum attacks, are not so important for the discussion here. In terms of achievability, Unruh proved that the random oracle is collapsing as a hash function, and thus that simple hash-function-based commitment schemes are collapsing in the random oracle model. In the context of hash functions, he proved in a follow-up work [6] that the *Merkle-Damgård* construction for hash functions is collapsing (under some mild restriction on the padding) if the underlying compression function is. Given that the random oracle is collapsing, this in particular implies that the Merkle-Damgård construction is collapsing in the random oracle model, and thus gives heuristic evidence that certain practical hash functions like SHA-2 are collapsing. Recently, Czajkowski *et al.* [2] showed a similar result for the *Sponge* construction [1], which for instance underlies the hash function standard SHA-3: the Sponge construction is collapsing if both parts of the underlying round function, i.e., the so-called *inner* and *outer* parts, are collapsing, and if the inner part is "zero-preimage resistant".[1]

Our Contribution. In this work, we introduce a new *formalism* and a new *framework* for arguing about the collapsing property of (hash) functions. The advantage of our new approach is that it allows for significantly simper proofs compared to the previous work above.

At the heart of our new *formalism* is a pseudo-metric that abstracts away computational aspects, and which allows for an "algebraic" formulation of the collapsing property. This in turn allows for simple proofs of basic *composability results* for the collapsing property. Some of those have already been claimed and proven in the work mentioned above; however, our proofs are much simpler. For instance, proving that the collapsing property is preserved under nested composition takes 2 full pages in [5] (see Lemma 27 in the full version of [5]), with various quantum circuits depicted; our proof (see Lemma 5) is a few lines. The main reason for this difference lies in the "algebraic" nature of our formulation, compared to the "algorithmic" approach used in prior work. This means that instead of specifying quantum reduction algorithms and arguing that they "do the job", our proofs are almost entirely by means of *term-manipulations*, where we manipulate the terms of interest by using a small set of basic rules that come along with our formalism. This not only results in very compact proofs, these proofs are also mathematically very clean in that in every term-manipulation step we can—and typically do—specify what basic rule was used.

These composability results for the collapsing property, together with a couple of basic features when "disallowing" certain inputs, form what we call our *framework*. With this framework, proving the collapsing property of hash domain extensions boils down to decomposing the iteration function under consideration into a few simple composition operations.

We demonstrate this new proof methodology on various examples. Applied to Merkle-Damgård, we obtain a proof of the collapsing property without any restriction on the padding as in [6], but with the additional assumption on the compression function to be "*iv*-preimage resistant" (which is satisfied in the random oracle model). We can also recover Unruh's original result, which requires a restriction on the padding but avoids the "*iv*-preimage resistance". By adding a counter and "salt" to the compression function but otherwise using the same kind of reasoning, we get a proof of the collapsing property of *HAIFA* [3], as proposed by Biham and Dunkelman. Applied to the *Sponge* construction, we recover the result from [2] up to an insignificant difference in the exact parameter.

The distinguishing feature of our proofs lies in their conceptual simplicity and low technical complexity. Our proofs are entirely in terms of decomposing the iteration function into elementary composition operations that are ensured to preserve the collapsing property. In particular, our proofs are purely classical. In contrast, the proofs provided in [2,6] are in terms of lengthy hybrid arguments

[1] This again implies security in the random oracle model, but a subtle issue here is that if the round function is efficiently invertible then the assumptions on the two parts are not satisfied. Hence, this is not so strong evidence yet that e.g. SHA-3 is collapsing.

that consider sequences of "quantum games" and in terms of quantum information theoretic arguments and quantum reduction algorithms for reasoning that every game in the sequence behaves similarly to its predecessor.

As such, even though the collapsing property of HAIFA is new, we consider our main contribution more in terms of offering a simple understanding of *why* certain hash function are collapsing, and in providing a tool to *easily* check if similar results also hold for other hash functions (as we demonstrate on HAIFA).

The Framework in Action. To give a better idea, we illustrate here on the various examples how our framework enables to argue for the collapsing property by means of decomposing the iteration function into suitable elementary decomposition operations, and thus in particular by means of purely classical reasoning. We challenge the reader to compare our proofs with those in [2,6].

Merkle-Damgård. The Merkle-Damgård hash of a message $x_1, \ldots, x_i$, consisting of i blocks, is given by $\mathit{IH}_i(x_1, \ldots, x_i)$, where IH_i is iteratively defined as

$$\mathit{IH}_i(x_1, \ldots, x_i) := f\big(\mathit{IH}_{i-1}(x_1, \ldots, x_{i-1}), x_i\big)$$

with $\mathit{IH}_0() = \mathsf{iv}$. The round function f is assumed to be collapsing. We observe that IH_i is the *nested composition* of f with the *concurrent composition* of IH_{i-1} with the identity $x_i \mapsto x_i$. Our framework ensures that these compositions preserve the collapsing property; thus, by recursive application, given that IH_0 is trivially collapsing, we get that IH_L is collapsing for every *fixed* L, and hence the Merkle-Damgård hash is collapsing when restricted to inputs of *fixed* size.

In order to deal with messages of variable size, we allow in the definition of $\mathit{IH}_i(x_1, \ldots, x_i)$ the left-most message blocks to be "empty", i.e., x_1 up to some x_j may be $\bot$, and we set $\mathit{IH}_i(\bot, \ldots, \bot) := \mathsf{iv}$ (for any i) and keep to recursive definition above if $x_i \neq \bot$. This extended version of IH_i is then the *disjoint union* of the trivial function $\{\bot^i\} \to \{\mathsf{iv}\}$ and the restriction of IH_i to inputs different than $\bot^i$, *if* we "disallow" non-$\bot^i$ inputs that are mapped to iv.[2] Thus, as long as we "disallow" such inputs (which is something our framework can capture), we still have that the recursive definition of IH_i decomposes into composition operations that are covered by our framework, and thus we can conclude that IH_L is collapsing for every fixed L, but now for inputs that may have $\bot$-prefixes, i.e., variable length. Finally, by the assumed "iv-preimage resistant" of f, inputs ($\neq \bot^i$) that IH_i maps to iv are hard to find, and therefore "disallowing" those has no noticeable effect.

HAIFA. The HAFIA hash function is a variant of Merkle-Damgård that includes a counter in the iteration function, and it uses a "salt" (which we though treat as ordinary input). Formally,

$$\mathit{IH}_i(\mathit{salt}, x_1, \ldots, x_i) := f\big(\mathit{salt}, \mathit{IH}_{i-1}(\mathit{salt}, x_1, \ldots, x_{i-1}), x_i, i\big)\,.$$

Here, we can reason exactly as above, except that now the iteration function is a *nested composition* of the function $f(\cdot, \cdot, \cdot, i)$, which is collapsing if f is, with

[2] The latter is because (our notion of) the disjoint union of two functions requires not only the two respective domains but also the two respective ranges to be disjoint.

the *parallel composition* of the projection function $(salt, x_1, \dots, x_i) \mapsto salt$ with the *concurrent composition* of IH_{i-1} with the identity function $x_i \mapsto x_i$. All these composition operations are covered by our framework, and so the collapsing property follows as for the original Merkle-Damgård construction, assuming again that f is "*iv*-preimage resistance" in case of arbitrary length messages.

Sponge. The Sponge hash[3] of a message $x_1, \dots, x_i$ of i blocks is given by $S_i^0(x_1, \dots, x_i)$, where S_i^b is iteratively defined as

$$S_i^b(x_1, \dots, x_i) := f^b\big(S_{i-1}^0(x_1, \dots, x_{i-1}) \oplus x_i, S_{i-1}^1(x_1, \dots, x_{i-1})\big)$$

for $b \in \{0,1\}$, with $S_0^0() = 0 = S_0^1()$, and it is assumed that both components of the round function $f = (f^0, f^1)$ are collapsing. Here, S_i^b is the *nested composition* of f^b with a function that is yet another *composition* of the functions S_{i-1}^0 and S_{i-1}^1, and our framework immediately ensures that S_i^0 and S_i^1 stay collapsing as long as S_{i-1}^1 is. Thus, again, the iteration function decomposes into composition operations that are ensured to preserve the collapsing property, and so by recursive application we get that $S_1^1, \dots, S_{L-1}^1$ and eventually S_L^0 are collapsing. The only difference to above is that here, we have to set $S_i^b(\perp, \dots, \perp) := S_0^b() = 0$ to ensure that S_L^0 acts correctly on messages of smaller block size, i.e., that $S_j^b(x_1, \dots, x_j) = S_L^b(\perp, \dots, \perp, x_1, \dots, x_j)$. As a consequence, for the recursive reasoning, to have S_i^1 be the disjoint union of the trivial function $\{\perp^i\} \to \{0\}$ and the restriction of S_i^1 to non-$\perp^i$ inputs, we need to "disallow" inputs ($\neq \perp^i$) which S_i^1 maps to 0; this has no noticeable effect though if f^1 is "zero-preimage resistant".

2 Preliminaries

2.1 Basic Quantum Formalism

Knowledge of basic concepts of quantum information science is necessary in order to prove "correctness" of our framework (but not to apply the framework); we fix here some notation and conventions, which both are not fully standard.

Typically, the *state* of a quantum system with state space $\mathcal{H}$ is given by a density matrix ρ, i.e., by a trace-1 positive-semidefinite matrix that acts on $\mathcal{H}$, and a *quantum operation* is expressed by a CPTP map $\mathbb{T}$ which maps a state ρ to a new state $\mathbb{T}(\rho)$ over a possibly different state space. In this work, for technical reasons, we allow states to be *subnormalized*, and we consider the more general notion of completely-positive *trace-nonincreasing* (CPTN) maps, which are of the form $\mathbb{T} = \sum_i \mathbb{T}_i$ with $\mathbb{T}_i : \rho \mapsto T_i \rho T_i^\dagger$ and $\sum_i T_i^\dagger T_i \leq I$ (the identity on $\mathcal{H}$).[4]

For the purpose of this work, a *measurement* is a CPTN map $\mathbb{P} = \sum_i \mathbb{P}_i$ with $\mathbb{P}_i : \rho \mapsto P_i \rho P_i^\dagger$ as above, but with the restriction that the P_i's are mutually

[3] For simplicity, we consider *one* block of output only; multiple output blocks are argued by means of composition too.

[4] This can be understood in that quantum operations may "abort", and the trace $\mathrm{tr}(\rho) \leq 1$ expresses the probability that the process that produces ρ does not abort.

orthogonal Hermitian *projections* on $\mathcal{H}$. If $\mathbb{P}$ is in fact a CPTP map, i.e., $\sum_i P_i = I$, then we speak of a *total* measurement, and otherwise of a *partial* measurement. The individual "components" $\mathbb{P}_i$ of such a (partial or total) measurement are sometimes also referred to as *measurements with post-selection.*

We write $\Pr[\,\mathbb{P}(\rho) = i\,]$ for $\mathrm{tr} \circ \mathbb{P}_i(\rho) = \mathrm{tr}(P_i \rho P_i)$, i.e., the probability that "outcome i is observed". An elementary property of any (*projective*, as considered here) measurement $\mathbb{P}$, is Winter's "gentle-measurement lemma" [7], which captures that the measurement does not disturb the state much if the outcome is almost certain. Formally,[5] for any state ρ and any $\beta \geq 0$:

$$\exists\, i : \Pr[\,\mathbb{P}(\rho) = i\,] \geq \mathrm{tr}(\rho) - \beta \implies \delta\big(\mathbb{P}(\rho), \rho\big) \leq \sqrt{\beta} + \beta\,. \tag{1}$$

where δ is the *trace distance*, given by $\delta(\rho, \sigma) := \frac{1}{2}\|\rho - \sigma\|_{tr}$.

Different quantum systems are identified by means of "labels" X, Y etc., and we write ρ_X for the state of system X and $\mathcal{H}_X$ for its state space, etc. For a CPTN map $\mathbb{T}$, we may write $\mathbb{T}_X$ to emphasize that it acts on system X, and $\mathbb{T}_{X \to X'}$ to additionally emphasize that it maps into system X'. For simplicity, we tend to write $\rho_{\mathbb{T}(X)Y}$ rather than $\big(\mathbb{T}_X \otimes \mathbb{I}_Y\big)(\rho_{XY})$.

For any state space we consider a fixed orthonormal basis, referred to as the *computational* basis. For state spaces $\mathcal{H}_X$ and $\mathcal{H}_Y$ with respective computational bases $\{|x\rangle\}_{x \in \mathcal{X}}$ and $\{|y\rangle\}_{y \in \mathcal{Y}}$, we associate to any function $f : \mathcal{X} \to \mathcal{Y}$ the CPTP "evaluation" map $\mathbb{E}[f]_{X \to XY} : \rho \mapsto V[f]\, \rho\, V[f]^\dagger$ given by the isometry $V[f] : |x\rangle \mapsto |x\rangle|f(x)\rangle$. Here, we also write $\rho_{Xf(X)Z}$ instead of $\rho_{\mathbb{E}[f](X)Z}$.[6] We note that $\mathbb{E}[f]$ admits a *left inverse*, i.e., a CPTP map $\mathbb{E}^{inv}[f]_{XY \to X}$ such that $\mathbb{E}^{inv}[f] \circ \mathbb{E}[f] = \mathbb{I}_X$.

The composition $\mathrm{tr}_Y \circ \mathbb{E}[f]_{X \to XY}$ of a CPTP evaluation map with the partial trace tr_Y equals the measurement $\mathbb{M}[f] = \sum_y \mathbb{M}[f = y]$, where $\mathbb{M}[f = y]$ is the CPTN map given by the projection into the span of $\{|x\rangle \mid f(x) = y\}$. To simplify notation, we may also write $\rho_{X^f Z}$ instead of $\rho_{\mathbb{M}[f](X)Z}$, and, similarly, $\rho_{X^{f=y}Z}$ instead of $\rho_{\mathbb{M}[f=y](X)Z}$.

The usual "measurement in the computational basis", given by the projections $|x\rangle\langle x|$, is simply denoted by $\mathbb{M}$. For lighter notation, we often use $\overline{(\cdot)}$ instead of $\mathbb{M}$ and write $\rho_{\bar{X}Y}$ instead of $\rho_{\mathbb{M}(X)Y}$. A quantum system X of a (possibly) joint state ρ_{XY} is called *classical* if $\rho_{\bar{X}Y} = \rho_{XY}$.

When the state is clear from the context, then we may do the "arithmetic" on the labels. For instance, using this convention, we can then say that any state ρ_{XZ} satisfies

$$\bar{X} f(X) Z = \bar{X} f(\bar{X}) Z = \bar{X}\,\overline{f(\bar{X})} Z = \bar{X}\,\overline{f(X)} Z\,, \tag{2}$$

to express that $\mathbb{M}_X$ and $\mathbb{E}[f]_X$ commute, and that $f(\bar{X})$ is classical given that $\bar{X}$ is. Similarly, we may then write $\Pr\big[\mathbb{M}[f](X) = y\big] = \Pr\big[\mathbb{M}\big(f(X)\big) = y\big] = \Pr\big[\,f(\bar{X}) = y\,\big]$, which may be interpreted differently but coincide.

[5] This bound can e.g. be derived from [8]. [2] claims the bound $\sqrt{\beta}$, but their proof has a small flaw; fixing it gives $\sqrt{2\beta}$ instead (but only works for *total* measurements). .

[6] A subtle issue with this notation is that $\mathrm{tr}_Y(\rho_{Xf(X)Z}) \neq \rho_{XZ}$, but rather $= \rho_{\mathbb{M}[f](X)Z}$ (see below).

2.2 Randomized Functions and States, and Their Complexity

In Appendix A, we offer a formal discussion of *randomized functions*, *randomized CPTN maps*, and *randomized quantum states.* As one would expect, these are simply functions, CPTN maps and states that depend on some *global randomness* r, which is randomly chosen once and for all from some finite set $\mathcal{R}$.

Informally, when considering randomized functions, one can make the following distinction. In one case, r is given as input to the function f (or to the algorithm that computes f, if you prefer); one then typically speaks of *keyed* or *seeded* functions. In the other case, f makes queries to an *oracle* that computes every reply dependent on r, in which case one refers to f as an *oracle function.* A similar distinction can be made for randomized CPTN maps, and thus for randomized states, which are simply randomized CPTN maps that act on the trivial state space $\mathbb{C}$.

Formally, the way the two variants differ is by the way *complexity* is captured: for keyed functions one consider the *computational complexity* of computing the function whereas for oracle functions one considers the *query complexity.*

Our results apply to both variants in that we consider an *abstract* complexity measure $\mathfrak{c}$ that assigns to every randomized function f a non-negative integer $\mathfrak{c}(f)$, also denoted $\mathfrak{c}_f$, and similarly for randomized CPTN maps, and which satisfies natural properties that one would expect from a complexity measure. The details of this are given in Appendix B. The computational complexity and the query complexity are then just specific instantiations.

2.3 The Distinguishing Advantage

The following parameterized indistinguishability measure, and our understanding of it as an abstract metric, is one of the central notions of our formalism.

Definition 1. *For randomized states ρ_X and ρ_Y (with randomness r) over a common Hilbert space $\mathcal{H}_X = \mathcal{H}_Y$, and for any non-negative integer q, we set*

$$\begin{aligned}\delta_q\big(\rho_X,\rho_Y\big) &:= \sup_{\mathbb{T}} \frac{1}{|\mathcal{R}|}\sum_r \big|\Pr\big[\mathbb{M}\big(\mathbb{T}(X)\big)=0\big] - \Pr\big[\mathbb{M}\big(\mathbb{T}(Y)\big)=0\big]\big| \\ &= \sup_{\mathbb{T}} \frac{1}{|\mathcal{R}|}\sum_r \delta\big(\mathbb{M}\circ\mathbb{T}(\rho_X),\mathbb{M}\circ\mathbb{T}(\rho_Y)\big)\,,\end{aligned}$$

where the supremum is over all randomized CPTN maps $\mathbb{T}$ (with randomness r) that map into the two-dimensional qubit state space and have complexity $\mathfrak{c}(\mathbb{T}) \le q$ and, by convention, $\mathbb{M}$ is the measurement in the computational basis.

Following the convention of doing the "arithmetic" on the labels, we typically write $\delta_q(X,Y)$ instead of $\delta_q\big(\rho_X,\rho_Y\big)$. Also, we write $\delta_q(X,Y|Z)$ as a short hand for $\delta_q\big(\rho_{XZ},\rho_{YZ}\big)$.

We emphasize that δ_q is a *pseudometric*: it is non-negative, symmetric, and satisfies triangle inequality, but it may potentially vanish for non-identical states.

Furthermore, δ_q is upper bounded by the trace distance δ, and it coincides with δ in case $q = \infty$, i.e., there is no restriction on $\mathfrak{c}(\mathbb{T})$. Finally, δ_q inherits several properties from the ordinary trace distance, which can easily be verified. For instance, it is *monotone* under randomized CPTN maps as

$$\delta_q\big(\mathbb{T}(X), \mathbb{T}(Y)\big) \leq \delta_{q+\mathfrak{c}(\mathbb{T})}(X, Y)\,,$$

and for any randomized CPTN map $\mathbb{T} = \sum_i \mathbb{T}_i$, we have *subadditivity* as

$$\delta_q\big(\mathbb{T}(X), \mathbb{T}(Y)\big) \leq \sum_i \delta_q\big(\mathbb{T}_i(X), \mathbb{T}_i(Y)\big)\,.$$

To simplify terminology, from now on we drop on the word "randomized" and take it as understood that functions, CPTN maps and states may be randomized, either in the form of keyed functions or as oracle functions, etc.

3 The Collapsing Property

We state here (a slight variation of) the definition of the collapsing property of functions, as proposed by Unruh [5], but using the formalism introduced above. In Sect. 3.2 we then discuss the straightforward extension to *partial* functions, which will turn out to be useful, and in Sect. 3.3 we show that the collapsing property behaves nicely under various composition operations. These composability results are all rather natural, and—with our formalism!—have simple short proofs. All together, this section then stands as "the framework" that we propose for arguing about the collapsing property of hash functions.

3.1 The Definition

The original formulation of the collapsing property for a function h is by means of two "games", where an "adversary" produces a (normalized) state ρ_{XYE} of a certain form, namely Y must be classical and equal to $h(X)$, and then in one game X is measured in the computation basis whereas in the other game it is left untouched instead, and the definition requires that it should be hard for any "distinguisher" to distinguish between the two games.

As for the notion of collision resistance, the collapsing property is meaningful only for randomized functions h.[7] In case of a *keyed* variant of such a function, one can aim for *conditional* results that state that h is collapsing (against computationally bounded adversaries) under some computational hardness assumption. In case of an *oracle* function and aiming for *unconditional* results, there is no exploitable effect in restricting the computational power of the parties, as long as the query complexity is limited. Our approach of using an abstract complexity notion allows us to cover *both* these settings simultaneously.

Our formal definition of the collapsing property is given below. Compared to the original definition by Unruh (which comes in a couple of different flavors,

[7] See the discussion in Appendix C for an exception to the rule.

which we discuss in Appendix C), we use a somewhat different terminology and formalism. For instance we do not explicitly speak of "games", and instead of quantifying over the possible adversaries we quantify over the states that may possibly be prepared by an adversary, and the quantification over the distinguishers is absorbed into the pseudometric δ_q. These modifications to the mathematical language have obviously no effect on the notion. There are a few more differences compared to the definition proposed by Unruh, but they all have no more than a small quantitative effect, as we discuss below.

Definition 2. *A function $h : \mathcal{X} \to \mathcal{Y}$ is called $\varepsilon(q)$-*collapsing *if*

$$\mathsf{cAdv}[h](q) := \sup_{\rho_{XYE}} \delta_q\big(X, \bar{X} \,|\, \bar{Y}E\big) \leq \varepsilon(q)$$

for all q, where the supremum is over all states[8] *$\rho_{XYE} = \rho_{Xh(X)E}$ with complexity $\mathfrak{c}(\rho_{XYE}) \leq q$. The measure $\mathsf{cAdv}[h]$ is called the* collapsing advantage *of h.*

Beyond the change in mathematical language, another difference is that in the original definition the system Y of the state ρ_{XYE}, as produced by the adversary, is required to be classical, whereas in Definition 2 we allow it to be non-classical but then "make it classical" by measuring it; this is obviously equivalent (given that measuring has zero complexity). A slightly more substantial difference is that we allow the state ρ_{XYE} to be *subnormalized*; i.e., we allow the adversary to abort. However, the collapsing advantage $\delta_q\big(X, \bar{X} \,|\, \bar{Y}E\big)$ of any subnormalized state ρ_{XYE} is the same as of the normalized state $\tilde{\rho}_{XYE} := \rho_{XYE} + (1 - \mathrm{tr}(\rho_{XYE}))|x_\circ\rangle\langle x_\circ| \otimes |h(x_\circ)\rangle\langle h(x_\circ)| \otimes |0\rangle\langle 0|$ for an arbitrary choice of $x_\circ \in \mathcal{X}$ on which h is defined. Since $\mathfrak{c}(\tilde{\rho}_{XYE}) \leq \mathfrak{c}(\rho_{XYE}) + \mathfrak{c}(h)$, this has only a small quantitative effect that is insignificant if $\mathfrak{c}(h)$ is insignificant compared to q. In other words, we can easily transform an adversary that aborts into one that does not abort but outputs $x_\circ$ and $y_\circ = h(x_\circ)$ instead.

Finally, in the original definition, the complexity of the adversary and the distinguisher *together* is bounded (by q), whereas we bound the individual complexities (both by q). This is merely for simplicity, and has only a factor-2 quantitative effect.

3.2 *Partial* versus *Total* Functions

In Definition 2, we implicitly considered the function $h : \mathcal{X} \to \mathcal{Y}$ to be a *total* function, i.e., a function that is defined on its entire domain $\mathcal{X}$. However, it will be useful to extend the definition to *partial* functions, which are defined only on a subset $\mathcal{X}_{\text{eff}} \subseteq \mathcal{X}$ of the domain.[9] In the context of *randomized* functions, as considered here, we allow $\mathcal{X}_{\text{eff}}$ to depend on the global randomness r; this is

[8] We recall that the requirement $\rho_{XYE} = \rho_{Xh(X)E}$ is a shorthand for asking ρ_{XYE} to be equal to a state obtained by applying $\mathbb{E}[h]$ to system X.

[9] This may be understood in that the computation of h "fails" on inputs not in $\mathcal{X}_{\text{eff}}$.

what distinguishes such a partial function from a total function with a smaller domain, since the domain $\mathcal{X}$ of a function is declared fixed and independent of r.

Definition 2 applies directly to such partial functions as well, given that the definition of the evaluation map $\mathbb{E}[h]$ is naturally extended to partial functions h by having the defining operator $V[h]$ map $|x\rangle$ to 0 for any $x \notin \mathcal{X}_{\text{eff}}$. The effect of this is that the requirement $\rho_{XYE} = \rho_{Xh(X)E}$ enforces X to contain no inputs from outside of $\mathcal{X}_{\text{eff}}$. Hence, considering partial functions in Definition 2 serves as a convenient way to "disallow" certain inputs.

Formally, consider a function $h : \mathcal{X} \to \mathcal{Y}$ (which may be partial but let us think of it as a total function for now), and let $\pi : \mathcal{X} \to \{0,1\}$ be a predicate, which will always be understood to be a total function. Then, we define $h|_\pi$ to be the partial function $h|_\pi : \mathcal{X} \to \mathcal{Y}$ that is undefined for $x \in \mathcal{X}$ with $\pi(x) = 0$, and that coincides with h for the remaining $x \in \mathcal{X}$. The collapsing advantage of $h|_\pi$ then coincides with the collapsing advantage of h modified in that the quantification over ρ_{XYE} is restricted to states for which $\Pr[\,\pi(\bar{X})=0\,] = 0$.

Below, in Lemmas 1 and 2, we show how $\mathsf{cAdv}[h]$ and $\mathsf{cAdv}[h|_\pi]$ relate to each other. Lemma 1 follows trivially from the above observation, i.e., that $\rho_{XYE} = \rho_{Xh|_\pi(X)E}$ implies $\rho_{XYE} = \rho_{Xh(X)E}$.

Lemma 1. *If h is $\varepsilon(q)$-collapsing then so is $h|_\pi$, i.e.,* $\mathsf{cAdv}[h|_\pi] \le \mathsf{cAdv}[h]$.

Applied to h of the form $h|_\tau$, and noting that $(h|_\tau)|_\pi = h|_{\pi\wedge\tau}$, we get the following, which captures that disallowing more inputs can only decrease the collapsing advantage.

Corollary 1. *For any predicates π and τ, it holds that* $\mathsf{cAdv}[h|_{\pi\wedge\tau}] \le \mathsf{cAdv}[h|_\tau]$. *In particular, if π implies τ, i.e. $\pi(x)=1 \Rightarrow \tau(x)=1$, then* $\mathsf{cAdv}[h|_\pi] \le \mathsf{cAdv}[h|_\tau]$.

For the other direction, disallowing some inputs has little effect if those are hard to find. For the formal statement, we need the following definition.

Definition 3. *A predicate $\pi : \mathcal{X} \to \{0,1\}$ is called* $\beta(q)$-almost-certain *if it holds that $\Pr[\pi(\bar{X})=0] \le \beta(q)$ for any state ρ_X with complexity q.*

Lemma 2. *If π is $\beta(q)$-almost-certain then*

$$\mathsf{cAdv}[h](q) \le \mathsf{cAdv}[h|_\pi](q + \mathfrak{c}_\pi) + \sqrt{\beta(q)} \cdot \min\big\{\sqrt{2}, 1+\sqrt{\beta(q)}\,\big\}\,.$$

Proof. Let $\rho_{XYZ} = \rho_{Xh(X)E}$ be with complexity q. Consider the measurement $\mathbb{P} = \mathbb{P}_0 + \mathbb{P}_1$ given by $\mathbb{P}_0 := \mathbb{M} \circ \mathbb{M}[\pi = 0]$ and $\mathbb{P}_1 := \mathbb{M}[\pi = 1]$.[10] By triangle inequality and since $\mathbb{M} = \mathbb{P} \circ \mathbb{M}$, we have

$$\begin{aligned}\delta_q\big(X,\bar{X}\,|\,\bar{Y}E\big) &\le \delta_q\big(X,\mathbb{P}(X)\,|\,\bar{Y}E\big) + \delta_q\big(\mathbb{P}(X),\mathbb{P}(\bar{X})\,|\,\bar{Y}E\big)\\ &\le \delta\big(X,\mathbb{P}(X)\,|\,\bar{Y}E\big) + \delta_q\big(X^{\pi=1},\bar{X}^{\pi=1}\,|\,\bar{Y}E\big) + \delta_q\big(\bar{X}^{\pi=0},\bar{X}^{\pi=0}\,|\,\bar{Y}E\big)\\ &\le \sqrt{\beta(q)} + \beta(q) + \mathsf{cAdv}[h|_\pi](q + \mathfrak{c}_\pi)\,,\end{aligned}$$

[10] I.e., $\mathbb{P}$ first performs the measurement $\mathbb{M}[\pi]$, and then measures the resulting state in the computation basis if (and only if) the measurement outcome was 0.

where the second inequality is because $\delta_q \leq \delta$, and by subadditivity and choice of $\mathbb{P}$, and the last inequality is by the "gentle-measurement lemma" (1), plus footnote 5, given that $\Pr\big[\mathbb{P}(X)=1\big] = \Pr\big[\pi(\bar{X})=1\big] \geq \operatorname{tr}(\rho_X) - \beta(q)$, plus the observation that $\rho_{X^{\pi=1}YE}$ has complexity $q + \mathfrak{c}_\pi$. □

We conclude with the following simple observation, which follows from the fact that under the given assumptions, $\Pr[\tau(\bar{X})=0] \leq \Pr[\pi \circ \lambda(\bar{X})=0] \leq \beta(q+\mathfrak{c}(\lambda))$ for any state ρ_X with complexity q.

Lemma 3. *Consider predicates $\pi : \mathcal{X}' \to \{0,1\}$ and $\tau : \mathcal{X} \to \{0,1\}$ and a total function $\lambda : \mathcal{X}' \to \mathcal{X}$ such that $\pi \circ \lambda$ implies τ, i.e., $\pi(\lambda(x))=1 \Rightarrow \tau(x)=1$. If π is $\beta(q)$-almost-certain then τ is $\beta(q+\mathfrak{c}(\lambda))$-almost-certain.*

3.3 Composability Properties

We show composability of the collapsing property under different means of composing functions. In one or another form, some of these composability properties are also present in previous work (see e.g. Lemma 27 in the full version of [5] for the corresponding claim on nested composition); we cover them here for completeness and since our notion differs in minor ways, but also in order to demonstrate how succinctly these composability properties can be *phrased* and *proven* using our formalism.

We take it as understood that for partial functions g and h, the considered composition is defined whenever g and h are both defined on their respective inputs.

Lemma 4 (Concurrent composition). *For $g : \mathcal{X} \to \mathcal{Y}$ and $h : \mathcal{W} \to \mathcal{Z}$, the concurrent composition $g\|h : \mathcal{X} \times \mathcal{W} \to \mathcal{Y} \times \mathcal{Z}$, $(x,w) \mapsto \big(g(x), h(w)\big)$ satisfies*

$$\mathsf{cAdv}[g\|h] \leq \mathsf{cAdv}[g] + \mathsf{cAdv}[h]\,.$$

Proof. Let $\rho_{XWYZE} = \rho_{XWg(X)h(W)E}$ be with complexity q. Then, by triangle inequality,

$$\begin{aligned}\delta_q\big(XW, \bar{X}\bar{W}|\bar{Y}\bar{Z}E\big) &\leq \delta_q\big(XW, X\bar{W}|\bar{Y}\bar{Z}E\big) + \delta_q\big(X\bar{W}, \bar{X}\bar{W}|\bar{Y}\bar{Z}E\big)\\ &= \delta_q\big(W, \bar{W}|\bar{Z}X\bar{Y}E\big) + \delta_q\big(X, \bar{X}|\bar{Y}\bar{W}\bar{Z}E\big)\\ &\leq \mathsf{cAdv}[g](q) + \mathsf{cAdv}[h](q)\,.\end{aligned}$$

□

Lemma 5 (Nested composition). *For $g : \mathcal{X} \to \mathcal{Y}$ and $h : \mathcal{Y} \to \mathcal{Z}$, the nested (or sequential) composition $h \circ g : \mathcal{X} \to \mathcal{Z}$, $x \mapsto h\big(g(x)\big)$ satisfies*

$$\mathsf{cAdv}[h \circ g](q) \leq \mathsf{cAdv}[g](q + \mathfrak{c}_g) + \mathsf{cAdv}[h](q + \mathfrak{c}_g)\,.$$

Proof. Let $\rho_{XZE} = \rho_{X(h\circ g)(X)E}$ be with complexity q. Then, $\rho_{XYZE} = \rho_{Xg(X)ZE}$ has complexity at most $q + \mathfrak{c}_g$. Recalling that ρ_{XZE} is recovered from ρ_{XYZE} by applying $\mathbb{E}^{inv}[g]_{XY\to X}$, we get

$$\begin{aligned}\delta_q\big(X, \bar{X}|\bar{Z}E\big) &\le \delta_{q+\mathfrak{c}_g}\big(XY, \bar{X}Y|\bar{Z}E\big) && \text{(monotonicity)}\\ &\le \delta_{q+\mathfrak{c}_g}\big(XY, X\bar{Y}|\bar{Z}E\big) + \delta_{q+\mathfrak{c}_g}\big(X\bar{Y}, \bar{X}Y|\bar{Z}E\big) && (\triangle \text{ inequality})\\ &\le \delta_{q+\mathfrak{c}_g}\big(Y, \bar{Y}|\bar{Z}XE\big) + \delta_{q+\mathfrak{c}_g}\big(X, \bar{X}|\bar{Y}\bar{Z}E\big) && (\bar{X}Y = \bar{X}\bar{Y} \text{ by (2)})\\ &\le \mathsf{cAdv}[g](q+\mathfrak{c}_g) + \mathsf{cAdv}[h](q+\mathfrak{c}_g)\,. \end{aligned}$$

□

Lemma 6. *For $g : \mathcal{X} \to \mathcal{Y}$ and $h : \mathcal{W} \times \mathcal{X} \to \mathcal{Z}$, where the latter function is such that $h(\cdot, x)$ is injective for any $x \in \mathcal{X}$, the composition $f : \mathcal{W}\times\mathcal{X} \to \mathcal{Y}\times\mathcal{Z}$, $(w, x) \mapsto \big(g(x), h(w, x)\big)$ satisfies*

$$\mathsf{cAdv}[f] \le \mathsf{cAdv}[g]\,.$$

We emphasize that the statement includes the special case where $\mathcal{W}$ is empty, i.e., $h : \mathcal{X} \to \mathcal{Z}$, in which case the the injectivity requirement becomes void, so that in particular the following holds.

Corollary 2 (Parallel composition). *For $g : \mathcal{X} \to \mathcal{Y}$ and $h : \mathcal{X} \to \mathcal{Z}$, the parallel composition $(g, h) : \mathcal{X} \to \mathcal{Y} \times \mathcal{Z}$, $x \mapsto \big(g(x), h(x)\big)$ satisfies*

$$\mathsf{cAdv}[(g, h)] \le \min\big\{\mathsf{cAdv}[g], \mathsf{cAdv}[h]\big\}\,.$$

Proof (of Lemma 6). Let $\rho_{WXYZE} = \rho_{WXg(X)\,h(W,X)E}$ be with complexity q. Then, using that $\bar{W}\bar{X}\bar{Z} = W\bar{X}\bar{Z}$, which holds by (2) because w is a function of x and $z = h(w, x)$,

$$\delta_q\big(WX, \bar{W}\bar{X}|\bar{Y}\bar{Z}E\big) = \delta_q\big(X, \bar{X}|\bar{Y}\bar{Z}WE\big) \le \mathsf{cAdv}[g](q)\,.$$

□

Lemma 7 (Disjoint union). *For $g : \mathcal{X} \to \mathcal{Y}$ and $h : \mathcal{W} \to \mathcal{Z}$ with disjoint domains and images, the* disjoint union *$g \sqcup h : \mathcal{X} \cup \mathcal{W} \to \mathcal{Y} \cup \mathcal{Z}$, which maps $x \in \mathcal{X}$ to $g(x)$ and $w \in \mathcal{W}$ to $h(w)$, satisfies*

$$\mathsf{cAdv}[g \sqcup h] \le \mathsf{cAdv}[g] + \mathsf{cAdv}[h]\,.$$

Proof. Let $\rho_{UVE} = \rho_{U(g\sqcup h)(U)E}$, and consider the "distinguishing function" $dis : \mathcal{X} \cup \mathcal{W} \to \{0, 1\}$ that maps $x \in \mathcal{X}$ to 1 and $w \in \mathcal{W}$ to 0. By our convention on function domains being recognizable, dis has zero complexity. Furthermore, $\rho_{U^{dis}VE} = \rho_{\mathbb{M}[dis](U)VE}$ is of the form

$$\rho_{U^{dis}VE} = \rho_{U^{dis=0}VE} + \rho_{U^{dis=1}VE} = \rho_{Xg(X)E} + \rho_{Wh(W)E}$$

and, by the disjointness of the images, $\rho_{U\bar{V}E} = \rho_{U^{dis}\bar{V}E}$, and so it follows from subadditivity that

$$\delta_q\big(U, \bar{U}|\bar{V}E\big) = \delta_q\big(U^{dis}, \bar{U}^{dis}|\bar{V}E\big) \le \delta_q\big(X, \bar{X}|\bar{Y}E\big) + \delta_q\big(W, \bar{W}|\bar{Z}E\big)$$

which is bounded by $\mathsf{cAdv}[g] + \mathsf{cAdv}[h]$. □

4 Application I: Merkle-Damgård and HAIFA

We demonstrate the usefulness of our framework. Here, we do so by (re)proving the collapsing property Merkle-Damgård, and by showing that the proof trivially translates to the HAIFA variation [3]. In the subsequent section we analyze the Sponge construction [1]. Our proofs argue entirely by means of decomposing the iteration function under consideration into a few composition operations.

Here and in the remainder, for $b \in \{0, 1, \perp\}$ and positive integer $i \in \mathbb{N}$, we write $b^i \in \{0, 1, \perp\}^i$ for the i-fold concatenation $(b, \dots, b)$ of b with itself.

4.1 The Construction

Let $f : \{0,1\}^c \times \{0,1\}^r \to \{0,1\}^c$ be a (total) function, which will act as the round function in the Merkle-Damgård construction. For any positive integer i, we consider the function $IH_i : \big(\{0,1\}^r\big)^i \to \{0,1\}^c$ given recursively by

$$IH_i(x_1, \dots, x_i) := f\big(IH_{i-1}(x_1, \dots, x_{i-1}), x_i\big) \tag{3}$$

with $IH_0() := \mathsf{iv}$, some fixed string in $\{0,1\}^c$ called the *initialization vector*. The Merkle-Damgård hash function is then formally given by[11]

$$MD : \big(\{0,1\}^r\big)^* \to \{0,1\}^c,\ (x_1, \dots, x_i) \mapsto IH_i(x_1, \dots, x_i).$$

For technical reasons, we extend the domain of IH_i above to

$$\mathcal{X}_i := \big\{(x_1, \dots, x_i) \in \big(\{\perp\} \cup \{0,1\}^r\big)^i \,\big|\, x_j = \perp \Rightarrow x_1 = \cdots = x_j = \perp\big\}$$

by setting $IH_i(\perp, \dots, \perp) := \mathsf{iv}$ and keeping the recursive definition (3) for $x_i \neq \perp$. We can now apply IH_L to messages of size $i < L$ blocks by pre-padding it with $\perp$'s: $IH_i(x_1, \dots, x_i) = IH_{i+1}(\perp, x_1, \dots, x_i) = \cdots = IH_L(\perp, \dots, \perp, x_1, \dots, x_i)$, and thus the restriction of MD to messages of block size $0 \le i \le L$ can be expressed as $MD^{\le L}(x_1, \dots, x_i) = IH_L(\perp, \dots, \perp, x_1, \dots, x_i)$.

4.2 The Analysis

Using our framework, we will now prove the following security statement for Merkle-Damgård. The assumption on $\mathfrak{c}(f)$ is simply for normalization, and for f to be *β-iv-preimage-resistant* means, by definition, that the predicate $1_{f(y) \neq \mathsf{iv}}$, which is 1 if y satisfies $f(y) \neq \mathsf{iv}$ and 0 otherwise, is β-almost-certain.

Theorem 1. *If f has complexity $\mathfrak{c}(f) = 1$, is ε-collapsing and β-iv-preimage-resistant, then, for any integer $L \ge 0$, the function $MD^{\le L}$ is γ-collapsing with*

$$\gamma(q) = L \cdot \varepsilon\big(q + \tfrac{1}{2}L(L+1)\big) + \sqrt{2\beta(q + L)}\,.$$

[11] Since the bit size of the input to MD must be an integer multiple of r, the Merkle-Damgård construction usually comes with a *padding* that maps a string of arbitrary size into a sequence of blocks of bit size r. We can safely ignore this since any injective padding preserves the collapsing property by Lemma 5.

For the purpose of the proof, we define for any i the predicate $\pi_i : \mathcal{X}_i \to \{0,1\}$ as

$$\pi_i(x_1,\ldots,x_i) = 1 \iff \forall j \in \{1,\ldots,i\} : x_j = \bot \,\vee\, \mathit{IH}_j(x_1,\ldots,x_j) \neq \mathsf{iv},$$

i.e., the bit is set unless the input is a non-trivial iv-preimage of some IH_j. In particular, if $\pi_i(x_1,\ldots,x_i) = 0$ then it must be that $\mathit{IH}_j(x_1,\ldots,x_j) = \mathsf{iv}$ for some j with $x_j \neq \bot$, and thus $y := \big(\mathit{IH}_{j-1}(x_1,\ldots,x_{j-1}), x_j\big)$ satisfies $f(y) = \mathit{IH}_j(x_1,\ldots,x_j) = \mathsf{iv}$ by (3). So, by Lemma 3, the following holds.

Lemma 8. *If f is β-iv-preimage-resistant then π_i is $\beta(q+\mathfrak{c}_{\mathit{IH}_{i-1}})$-almost-certain.*

Recall that $\mathit{IH}_i|_{\pi_i}$ is the partial function that is defined only for the inputs which satisfy π_i. The heart of the proof of Theorem 2 is the following recursive statement, which ensures that if $\mathit{IH}_{i-1}|_{\pi_{i-1}}$ is collapsing then so is $\mathit{IH}_i|_{\pi_i}$. By repeated application, we then get that $\mathit{IH}_L|_{\pi_L}$ is collapsing, and since π_L is almost-certain, IH_L is collapsing as well (by Lemma 2).

Proposition 1. *For any positive integer i:*

$$\mathsf{cAdv}\big[\mathit{IH}_i|_{\pi_i}\big](q) \;\leq\; \mathsf{cAdv}\big[\mathit{IH}_{i-1}|_{\pi_{i-1}}\big]\big(q + \mathfrak{c}_{\mathit{IH}_{i-1}}\big) + \varepsilon\big(q + \mathfrak{c}_{\mathit{IH}_{i-1}}\big)\,.$$

Proof. We let $\dot{\mathit{IH}}_i$ and $\dot{\pi}_i$ be the respective restrictions of IH_i and π_i to the domain $\dot{\mathcal{X}}_i := \mathcal{X}_i \setminus \{\bot^i\}$. Then, we see that $\mathit{IH}_i|_{\pi_i}$ is the disjoint union of the trivial function $\{\bot^i\} \to \{\mathsf{iv}\}$ and $\dot{\mathit{IH}}_i|_{\dot{\pi}_i}$; the crucial observation here is that the image of $\dot{\mathit{IH}}_i|_{\dot{\pi}_i}$ is disjoint with $\{\mathsf{iv}\}$. Therefore, by Lemma 7,

$$\mathsf{cAdv}\big[\mathit{IH}_i|_{\pi_i}\big](q) \leq \mathsf{cAdv}\big[\dot{\mathit{IH}}_i|_{\dot{\pi}_i}\big](q) \leq \mathsf{cAdv}\big[\dot{\mathit{IH}}_i|_{\pi_{i-1}}\big](q)\,,$$

where the latter inequality is by Lemma 1, given that $\dot{\pi}_i$ implies π_{i-1}.[12] Furthermore, since

$$\dot{\mathit{IH}}_i(x_1,\ldots,x_i) = f\big(\mathit{IH}_{i-1}(x_1,\ldots,x_{i-1}), x_i\big)$$

on its domain $\dot{\mathcal{X}}_i$, i.e., it is the nested composition of f with the concurrent composition of IH_{i-1} and the identity function $x_i \mapsto x_i$, Lemma 4 and 5 imply

$$\mathsf{cAdv}\big[\dot{\mathit{IH}}_i|_{\pi_{i-1}}\big](q) \;\leq\; \mathsf{cAdv}\big[\mathit{IH}_{i-1}|_{\pi_{i-1}}\big]\big(q + \mathfrak{c}_{\mathit{IH}_{i-1}}\big) + \mathsf{cAdv}\big[f\big]\big(q + \mathfrak{c}_{\mathit{IH}_{i-1}}\big)\,,$$

which completes the proof. □

Proof (of Theorem 1). $\mathit{IH}_0|_{\pi_0} = \mathit{IH}_0$ is trivially 0-collapsing. For convenience, we let n_i be the sum of integers $n_i := 1 + 2 + \cdots i = \frac{1}{2}i(i+1)$. Assuming by induction that $\mathsf{cAdv}[\mathit{IH}_i|_{\pi_i}](q) \leq i \cdot \varepsilon(q + n_{i-1})$, we get from Proposition 1 that

$$\mathsf{cAdv}\big[\mathit{IH}^1_{i+1}|_{\pi_{i+1}}\big](q) \leq \varepsilon(q+i) + i \cdot \varepsilon(q + n_{i-1} + i) \leq (i+1)\cdot\varepsilon(q + n_i)\,,$$

using that $\mathfrak{c}_{\mathit{IH}_i} = i \cdot \mathfrak{c}_f = i$ and $n_{i-1} + i = n_i$. Hence, the induction assumption holds for all i, and

$$\begin{aligned}\mathsf{cAdv}\big[\mathit{IH}_L\big](q) &\leq \mathsf{cAdv}\big[\mathit{IH}_L|_{\pi_L}\big](q+L) + \sqrt{2\beta(q+L)} \qquad \text{(Lemma 2 \& 8)}\\ &\leq L\cdot\varepsilon\big(q + L + n_{L-1}\big) + \sqrt{2\beta(q+L)}\,.\end{aligned}$$

□

[12] Here, we understand π_{i-1} as $\pi_{i-1} : \dot{\mathcal{X}}_i \to \{0,1\}$, $(x_1,\ldots,x_i) \mapsto \pi_{i-1}(x_1,\ldots,x_{i-1})$.

4.3 Instantiation with a Random Oracle

If f is a *random oracle*, which formally means that we consider the oracle $\mathcal{O}$ that is a uniformly random function $\{0,1\}^c \times \{0,1\}^r \to \{0,1\}^c$ and f is the trivial oracle function that outputs whatever $\mathcal{O}$ outputs on the given input, then, as shown by Unruh in [5], f is $O\big(\sqrt{q^3/2^c}\big)$-collapsing.[13] Furthermore, by the results on the hardness of quantum search from [4, Theorem 1], applied to the oracle function $F : \{0,1\}^c \times \{0,1\}^r \to \{0,1\}$ given by $F(y) = 1$ if and only if $f(y) = \mathsf{iv}$, we immediately get that f is $8(q{+}1)^2/2^c$-iv-preimage-resistant. As such, we obtain that for messages of block-size at most L, the Merkle-Damgård hash function $MD^{\leq L}$ is ε-collapsing with

$$\varepsilon(q) = O\Big(L\sqrt{(q + L^2)^3/2^c}\Big) .$$

As far as we understand, the results of [6] imply a collapsing advantage of $O\big(L\sqrt{(q + L)^3/2^c}\big)$, which is slightly better because of the L^2 that we have in our bound, but this is insignificant in typical settings where $q \gg L$.

4.4 HAIFA

Along the very same lines as for the original Merkle-Damgård construction, we can easily show that also HAIFA, a variant proposed by Biham and Dunkelmann [3], is collapsing, under the same assumptions. HAIFA works similarly to Merkle-Damgård except that

$$IH_i(salt, x_1, \dots, x_i) := f\big(salt, IH_{i-1}(salt, x_1, \dots, x_{i-1}), x_i, i\big)$$

i.e., the round function takes as additional inputs the round number i and some salt (that is the same for every round).[14] Proposition 1 immediately extends to HAIFA; the only thing that changes in the proof is that f becomes $f_i = f(\cdot,\cdot,\cdot,i)$, which is collapsing if f is, and we also have to use Corollary 2 to argue that the parallel composition of $(salt, x_1, \dots, x_i) \mapsto salt$ with the concurrent composition of IH_{i-1} and $x_i \mapsto x_i$ stays collapsing. The collapsing property of HAIFA then follows easily by inductively applying this variation of Proposition 1 as in the proof of Theorem 1.

4.5 Merkle-Damgård Without iv-Preimage-Resistance

We can also recover Unruh's original result on MD, which does not require f to be iv-preimage-resistant but instead restricts the set of inputs to be *suffix-free*.

[13] Even though our definition of the collapsing property differs slightly from the definition in [5], these differences disappear in such asymptotic statements, as discussed in Sect. 3. See also Appendix C.

[14] For the purpose of collisions and the collapsing property, we can think of the salt simply as part of the input: we do not want collisions even for different choices of the salt.

For that, given a fixed integer $L > 0$ and arbitrary $0 \le i \le L$, consider the map IH_i^* given by

$$IH_i^* : (x_1, \dots, x_L) \mapsto \big(IH_i(x_1, \dots, x_i), x_{i+1}, \dots, x_L\big) ,$$

defined on the considered suffix-free inputs of size at most L blocks, left-padded with $\perp$'s, and we argue the following variant of Proposition 1: if $IH_i^*|_{x_i \neq \perp}$ is collapsing then $IH_{i+1}^*|_{x_{i+1} \neq \perp}$ is collapsing too (for $i < L$). This variant of Proposition 1 follows from the observation that the latter is obtained as the nested composition $IH_{i+1}^*|_{x_{i+1} \neq \perp} = (f \| id) \circ IH_i^*|_{x_{i+1} \neq \perp}$ of $IH_i^*|_{x_{i+1} \neq \perp}$ with the concurrent composition of f and the identity id acting on $x_{i+2}, \dots, x_L$. Furthermore,

$$IH_i^*|_{x_{i+1} \neq \perp}(x_1,\dots,x_L) = \begin{cases} IH_i^*|_{x_i \neq \perp}(x_1,\dots,x_L) = \big(IH_i(x_1,\dots,x_i), x_{i+1},\dots,x_L\big) & \text{if } x_i \neq \perp \\ \big(\mathit{iv}, x_{i+1}, x_{i+2},\dots,x_L\big) & \text{if } x_i = \perp \end{cases}$$

and therefore $IH_i^*|_{x_{i+1} \neq \perp}$ is the disjoint union of $IH_i^*|_{x_i \neq \perp}$ and the function $(\perp^i, x_{i+1}, \dots, x_L) \mapsto (\mathit{iv}, x_{i+1}, x_{i+2}, \dots, x_L)$. Here, we are using the suffix-freeness of the considered inputs $x_1, \dots, x_L$; this ensures that not only the domains but also the images of the two functions are disjoint: if $(\perp^i, x_{i+1}, \dots, x_L)$ is "allowed" then $(x_1, \dots, x_L)$ is not unless x_1 up to x_i are all $\perp$. The above variant of Proposition 1 then follows from the preservation of the collapsing property under the different compositions, and then, by inductively applying this variant of Proposition 1, we obtain that $IH_L^*|_{x_L \neq \perp}$ is collapsing, and thus $MD^{\le L}$ is, given that the input is from a suffix-free set.

5 Application II: The Sponge

Here, we apply our framework to the Sponge construction [1]. As one can see, we follow the exact same blueprint as in Sect. 4.

5.1 The Construction

Let $f = (f^0, f^1) : \{0,1\}^r \times \{0,1\}^c \to \{0,1\}^r \times \{0,1\}^c$ be a (total) function, which will act as the round function in the Sponge construction. For any positive integer i, consider the function

$$S_i = (S_i^0, S_i^1) : \big(\{0,1\}^r\big)^i \to \{0,1\}^r \times \{0,1\}^c$$

given recursively by

$$S_i(x_1, \dots, x_i) := f\big(S_{i-1}^0(x_1, \dots, x_{i-1}) \oplus x_i, S_{i-1}^1(x_1, \dots, x_{i-1})\big) \tag{4}$$

with $S_0() := 0$. The sponge function (with s rounds of "squeezing") is then formally given by[15]

$$\begin{aligned} \mathit{Sponge}[s] : \big(\{0,1\}^r\big)^* &\to \big(\{0,1\}^r\big)^s \\ (x_1,\dots,x_i) &\mapsto \big(S_i^0(x_1,\dots,x_i), S_{i+1}^0(x_1,\dots,x_i,0^r),\dots, S_{i+s-1}^0(x_1,\dots,x_i,0^r,\dots,0^r)\big) . \end{aligned}$$

[15] Like for Merkle-Damgård, we can safely ignore the padding here.

For technical reasons, we extend the domain of S_i above to

$$\mathcal{X}_i := \big\{(x_1,\ldots,x_i) \in \big(\{\perp\} \cup \{0,1\}^r\big)^i \,\big|\, x_j = \perp \Rightarrow x_1 = \cdots = x_j = \perp\big\}$$

i.e., to strings that may have $\perp$-prefixes. We do so by setting

$$S_i(\perp,\ldots,\perp) := 0^{r+c}$$

and keeping the recursive definition (4) for $x_i \neq \perp$. This extension allows us to apply S_L to messages $(x_1,\ldots,x_i) \in (\{0,1\}^r)^i$ of size $i < L$ blocks by pre-padding it with $\perp$'s: $S_i(x_1,\ldots,x_i) = S_{i+1}(\perp,x_1,\ldots,x_i) = \cdots = S_L(\perp,\ldots,\perp,x_1,\ldots,x_i)$, and thus the restriction of *Sponge*$[s]$ to messages of block size $1 \leq i \leq L$ can be expressed as:

$$\mathit{Sponge}[s]^{\leq L}(x_1,\ldots,x_i) = \big(S^0_L(\perp^{L-i},x_1,\ldots,x_i), S^0_{L+1}(\perp^{L-i},x_1,\ldots,x_i,0^r),\ldots\big) \quad (5)$$

where we note that we insist here on $i \geq 1$, i.e., the message is non-empty.

5.2 The Analysis

Here, we prove the following. Also here, the assumption on $\mathfrak{c}(f)$ is simply for normalization, and for f^1 to be *β-zero-preimage-resistant* means, by definition, that the predicate $1_{f^1(y)\neq 0^c}$ is β-almost-certain.

Theorem 2. *If f has complexity 1, and f^0 and f^1 are ε^0- and ε^1-collapsing, and f^1 is β-zero-preimage-resistant, then, for any integer $L \geq 0$, the Sponge function Sponge$[s]^{\leq L}$ is γ-collapsing with*

$$\gamma(q) \;\leq\; \varepsilon^0(q+2L-1) + (L-1)\cdot\varepsilon^1\big(q+\tfrac{1}{2}L(L+1)\big) + \sqrt{2\beta(q+L)}\,.$$

For the purpose of the proof, we define for any i the predicate $\pi_i : \mathcal{X}_i \to \{0,1\}$ as

$$\pi_i(x_1,\ldots,x_i) = 1 \iff \forall\, j \in \{1,\ldots,i\} : x_j = \perp \,\vee\, S^1_j(x_1,\ldots,x_j) \neq 0^c\,,$$

i.e., the bit is set unless the input is a non-trivial zero-preimage of some S^1_j. In particular, if $\pi_i(x_1,\ldots,x_i) = 0$ then $S^1_j(x_1,\ldots,x_j) = 0^c$ for some j with $x_j \neq \perp$, and thus $y := \big(S^0_{j-1}(x_1,\ldots,x_{j-1}) \oplus x_j, S^1_{j-1}(x_1,\ldots,x_{j-1})\big)$ satisfies $f^1(y) = S^1_j(x_1,\ldots,x_j) = 0^c$ by (4). Thus, by Lemma 3, the following holds.

Lemma 9. *If f^1 is β-zero-preimage-resistant then π_i is $\beta(q+\mathfrak{c}_{S_{i-1}})$-almost-certain, and the same holds for $\dot\pi_i$, defined as below.*

For any i, let $\dot S^b_i$ and $\dot\pi_i$ be the respective restrictions of S^b_i and π_i to the domain $\dot{\mathcal{X}}_i := \mathcal{X}_i \setminus \{\perp^i\}$. The heart of the proof of Theorem 2 is the following recursive statement, which ensures that if $S^1_{i-1}|_{\pi_{i-1}}$ is collapsing then so are $\dot S^0_i|_{\dot\pi_i}$ and $S^1_i|_{\pi_i}$. By repeated application, we then get that $\dot S^0_L|_{\dot\pi_L}$ is collapsing, and since $\dot\pi_L$ is almost-certain, $\dot S^0_L$ is collapsing as well (by Lemma 2).

Proposition 2. *For any positive integer i:*

$$\mathsf{cAdv}\big[\dot{S}^0_i|_{\dot{\pi}_i}\big](q), \mathsf{cAdv}\big[S^1_i|_{\pi_i}\big](q) \;\leq\; \mathsf{cAdv}\big[S^1_{i-1}|_{\pi_{i-1}}\big]\big(q+\mathfrak{c}_{S_{i-1}}\big) + \varepsilon^b\big(q+\mathfrak{c}_{S_{i-1}}\big)\,.$$

Proof. We note that $S^1_i|_{\pi_i}$ is the disjoint union of the trivial function $\{\perp^i\} \to \{0^c\}$ and $\dot{S}^1_i|_{\dot{\pi}_i}$; the crucial observation here is that the image of $\dot{S}^1_i$ does not contain 0^c. Therefore, by Lemma 7,

$$\mathsf{cAdv}\big[S^1_i|_{\pi_i}\big](q) \leq \mathsf{cAdv}\big[\dot{S}^1_i|_{\dot{\pi}_i}\big](q) \leq \mathsf{cAdv}\big[\dot{S}^1_i|_{\pi_{i-1}}\big](q)\,.$$

where the latter inequality is by Lemma 1, given that $\dot{\pi}_i$ implies π_{i-1}.[16] Furthermore, since

$$\dot{S}^1_i(x_1,\ldots,x_i) = f^1\big(S^0_{i-1}(x_1,\ldots,x_{i-1}) \oplus x_i, S^1_{i-1}(x_1,\ldots,x_{i-1})\big)$$

on its domain $\dot{\mathcal{X}}_i$, i.e., it is a nested composition of f^1 with a function that is obtained as a composition as considered in Lemma 6, Lemmas 5 and 6 imply that

$$\mathsf{cAdv}\big[\dot{S}^1_i|_{\pi_{i-1}}\big](q) \;\leq\; \mathsf{cAdv}\big[S^1_{i-1}|_{\pi_{i-1}}\big]\big(q+\mathfrak{c}_{S_{i-1}}\big) + \mathsf{cAdv}\big[f^b\big]\big(q+\mathfrak{c}_{S_{i-1}}\big)\,,$$

which was to be proven. The reasoning for $\dot{S}^0_i|_{\dot{\pi}_i}$ is exactly as for $\dot{S}^1_i|_{\dot{\pi}_i}$ above. □

Proof (of Theorem 2). $S^1_0|_{\pi_0} = S^1_0$ is trivially 0-collapsing. For convenience, we let n_i be the sum of integers $n_i := 1+2+\cdots i = \frac{1}{2}i(i+1)$. Assuming by induction that $\mathsf{cAdv}[S^1_i|_{\pi_i}](q) \leq i\cdot\varepsilon^1(q+n_{i-1})$, we get from Proposition 2 that

$$\mathsf{cAdv}\big[S^1_{i+1}|_{\pi_{i+1}}\big](q) \leq \varepsilon^1(q+i) + i\cdot\varepsilon^1(q+n_{i-1}+i) \leq (i+1)\cdot\varepsilon^1(q+n_i)\,,$$

using that $\mathfrak{c}_{S_i} = i\cdot\mathfrak{c}_f = i$ and $n_{i-1}+i = n_i$. Hence, the induction assumption holds for all i, and

$$\begin{aligned}\mathsf{cAdv}\big[\dot{S}^0_L\big](q) &\leq \mathsf{cAdv}\big[\dot{S}^0_L|_{\dot{\pi}_L}\big](q+L) + \sqrt{\beta(q+L)}\\ &\leq \varepsilon^0(q+2L-1) + \mathsf{cAdv}\big[S^1_{L-1}|_{\pi_{L-1}}\big](q+2L-1) + \sqrt{2\beta(q+L)}\\ &\leq \varepsilon^0(q+2L-1) + (L-1)\cdot\varepsilon^1\big(q+n_L\big) + \sqrt{2\beta(q+L)}\,.\end{aligned}$$

where the first inequality is by Lemmas 2 and 9, and the second by Proposition 2. The claim on *Sponge*$[s]^{\leq L}$ follows now from (5) and Corollary 2. □

5.3 Instantiation with a Random Oracle

If $f = (f^0, f^1)$ is a *random oracle*, then it follows easily from the work of Unruh in [5] on the collapsing property of the random oracle that f^0 and f^1 are respectively $O\big(\sqrt{q^3/2^r}\big)$- and $O\big(\sqrt{q^3/2^c}\big)$-collapsing. Furthermore, as pointed out in [2], by the results on the hardness of quantum search from [4, Theorem 1] to the oracle function $F : \{0,1\}^r \times \{0,1\}^c \to \{0,1\}$ given by $F(y) = 1$ if and

[16] Here, we understand π_{i-1} as $\pi_{i-1} : \dot{\mathcal{X}}_i \to \{0,1\}$, $(x_1,\ldots,x_i) \mapsto \pi_{i-1}(x_1,\ldots,x_{i-1})$.

only if $f^1(y) = 0^c$, we immediately get that the function f^1 is $8(q+1)^2/2^c$-zero-preimage-resistant. Therefore, we get that for messages of block-size at most L, the sponge function $\mathit{Sponge}[s]^{\leq L}$, with the round function modeled by a random oracle, is ε-collapsing with

$$\varepsilon(q) = O\Big(\sqrt{(q+L)^3/2^r} + L\sqrt{(q+L^2)^3/2^c}\Big).$$

This matches with single-execution-variant (i.e. $t = 1$) of Theorem 33 of [2], except for the square in the L^2 term. When considering a t-fold parallel composition $\mathit{Sponge}[s]^{\leq L}\| \cdots \|\mathit{Sponge}[s]^{\leq L}$, it follows immediately from Lemma 6 that the collapsing parameter grows linearly with t, i.e., as

$$O\Big(t\sqrt{(q+L)^3/2^r} + tL\sqrt{(q+L^2)^3/2^c}\Big),$$

which is comparable to Theorem 33 of [2] with a general t, which states a collapsing advantage of

$$O\Big(t\sqrt{(q+tL)^3/2^r} + tL\sqrt{(q+tL)^3/2^c}\Big).$$

6 Conclusion

We consider the quantum *collapsing* property of classical hash functions, which replaces the notion of *collision resistance* in the presence of quantum attacks, and we propose a formalism and a framework that enables to argue about the collapsing property of hash domain extension constructions simply by means of decomposing the iteration function under consideration into elementary composition operations. In particular, our framework allows us to argue by purely classical means that hash functions are secure against quantum attacks.

We demonstrate this proof methodology on several examples. For Merkle-Damgård and the Sponge construction, we recover what has already been proven in [2,6], up to insignificant differences, whereas our result for HAIFA is, strictly speaking, new. It is well possible that the respective proof provided in [6] extends to HAIFA as well; however, this is cumbersome to verify (we challenge the reader to do so). With our approach, on the other hand, it is *trivial* to see that our proof for Merkle-Damgård extends to this variation: the only thing that needs to be verified is that the modified iteration function still decomposes into composition operations that are covered by our framework.

We think it is fair to say that, compared to previous work which proves that some hash domain extension constructions *are* collapsing, our approach gives much more insight into *why* they are collapsing. Furthermore, our framework should be a helpful tool when designing new hash functions that are meant to withstand quantum attacks.

Last but not least, from a conceptual perspective, we find it particularly interesting to see that our simplified proofs are the result of departing from the common methodology of proving a conditional security statement by means of an algorithmic reduction. Instead of assuming an attack against the construction and then building an attack against the underlying component, we argue *directly*—and in some sense "algebraically"—that if the underlying component is secure then so is the construction.

A Randomized Functions and CPTN Maps

In this work, we will consider two variants of the notion of a *randomized function*, and our techniques will apply to both. Formally, a randomized function is a function $f : \mathcal{R} \times \mathcal{X} \to \mathcal{Y}$ for a fixed choice of the (finite) set $\mathcal{R}$, and it is understood that $r \in \mathcal{R}$ is chosen uniformly at random once-and-for-all. Informally, we think of such a randomized function as a function $f : \mathcal{X} \to \mathcal{Y}$ that produces its output $f(x) = f(r, x)$ for any input x dependent on some "global randomness" r, which is the same for all inputs and all randomized functions considered at a time.

Informally, the two variants we consider in this work differ in the way the randomness r is accessed by the function. In one case, r is *explicitly* given as input to the function f (or to the algorithm that computes f, if you prefer); one then typically speaks of *keyed* (or *seeded*) functions. In the other case, r is not explicitly given to f but instead, f makes *oracle queries* to a designated randomized function $\mathcal{O}$, called the *oracle*, which computes every reply dependent on r. This latter case is typically referred to as an *oracle function.*

We point out that from a mathematical perspective, there is no distinction yet between a keyed and an oracle function, in that both are merely functions that additionally act on some global randomness r. The way the two variants differ formally is by the way we capture *complexity*: for keyed functions we consider the *computational complexity* whereas for oracle functions we consider the *query complexity.*[17] We address this in more detail in the subsequent section.

In line with the above, we can also consider the notion of a *randomized CPTN map* $\mathbb{T}$, which is a CPTN map whose action on a quantum state depends on the global randomness r, and we can distinguish between *keyed* CPTN maps that have *direct* access to r, and *oracle* CPTN maps that have *quantum oracle access* to a designated randomized function $\mathcal{O} : \mathcal{R} \times \mathcal{U} \to \mathcal{V}$. Here, "quantum oracle access" means that $\mathbb{T}$ can query $\mathcal{O}$ in superposition, i.e., it may ask to have the unitary $|u\rangle|v\rangle \mapsto |u\rangle|v + \mathcal{O}(r, u)\rangle$ applied to any state (of appropriate dimension). Again, the formal distinction between the two variants is in terms of the complexity measure.

We point out that by considering randomized CPTN maps $\mathbb{T}$ (of either flavor) that act on the empty system with trivial one-dimensional state space, we may also speak of *randomized states* (of either flavor) as the states ρ produced as $\rho = \mathbb{T}(1)$ for such a randomized CPTN map. We take it as understood here that the description of such a randomized state includes the dependency on the randomness r.

[17] For the latter, one could actually consider both simultaneously.

B Complexity

We introduce here the abstract notion of *complexity* that we consider in this work and discuss below the two main instantiations that are relevant for us. In the context of randomized functions $f : \mathcal{X} \to \mathcal{Y}$, we consider a map that assigns to any such function a non-negative integer $\mathfrak{c}(f)$, called the *complexity* of f, which is meant to express how hard it is to "compute" f. We assume that our abstract notion satisfies natural properties, like that the identity function on any set has zero complexity, and that it behaves well under composition, so that

$$\mathfrak{c}(g \circ f) \leq \mathfrak{c}(f) + \mathfrak{c}(g) \qquad \text{and} \qquad \mathfrak{c}(f \| g) \leq \mathfrak{c}(f) + \mathfrak{c}(g)$$

for any f and g with appropriate domain/range, where $g \circ f : x \mapsto g\bigl(f(x)\bigr)$ and $f \| g : (x, w) \mapsto \bigl(f(x), g(w)\bigr)$. For simplicity, we additionally assume that certain "simple" functions have zero complexity. These are: constants, copying, deleting, swapping, checking equality, as well as bit-wise XOR. Also, to avoid certain technical complications, once $\mathfrak{c}$ is fixed we only consider randomized functions $f : \mathcal{X} \to \mathcal{Y}$ for which $\mathcal{X}$ can be recognized with zero complexity.[18] For lighter notation, we may also write $\mathfrak{c}_f$ instead of $\mathfrak{c}(f)$.

We also consider a notion of complexity for randomized CPTN maps, which, as above, assigns a non-negative integer $\mathfrak{c}(\mathbb{T})$ to any randomized CPTN map $\mathbb{T}$. Similarly to above, we assume that the identity $\mathbb{I}$ has zero complexity, that

$$\mathfrak{c}(\mathbb{S} \circ \mathbb{T}) \leq \mathfrak{c}(\mathbb{T}) + \mathfrak{c}(\mathbb{S}) \qquad \text{and} \qquad \mathfrak{c}(\mathbb{T} \otimes \mathbb{S}) \leq \mathfrak{c}(\mathbb{T}) + \mathfrak{c}(\mathbb{S}) \,,$$

and that certain "simple" maps have zero complexity, namely: the preparation of states in the computational basis, measurements (with or without post-selection) in the computational basis, partial traces, and swapping registers. On top, we assume the complexity notion for CPTN maps to be consistent with that of functions, in that we require that

$$\mathfrak{c}\bigl(\mathbb{E}[f]\bigr), \mathfrak{c}\bigl(\mathbb{E}^{inv}[f]\bigr), \mathfrak{c}\bigl(\mathbb{M}[f]\bigr), \mathfrak{c}\bigl(\mathbb{M}[f = y]\bigr) \leq \mathfrak{c}(f)$$

where the latter two actually follow from the first, given that partial traces and measurements in the computational basis are "for free".

Given such a notion of complexity (for randomized CPTN maps), we can define the complexity of a randomized state ρ as $\mathfrak{c}(\rho) := \mathfrak{c}(\mathbb{T})$ where $\mathbb{T}$ is the randomized CPTN map with minimal complexity that produces ρ as $\rho = \mathbb{T}(1)$. It obviously holds that $\mathfrak{c}\bigl(\mathbb{T}(\rho)\bigr) \leq \mathfrak{c}(\rho) + \mathfrak{c}(\mathbb{T})$ for any randomized CPTN map. A last requirement we pose onto our abstract complexity measure $\mathfrak{c}$ is that $\mathfrak{c}\bigl(\rho + \bigl(1 - \mathrm{tr}(\rho)\bigr)\sigma\bigr) \leq \mathfrak{c}(\rho) + \mathfrak{c}(\sigma)$ for all randomized states.[19]

[18] Meaning that for any $\mathcal{X}' \supset \mathcal{X}$, the function $\mathcal{X}' \to \{0,1\}$ that maps $x \in \mathcal{X}$ to 1 and $x \in \mathcal{X}' \setminus \mathcal{X}$ to 0 has zero complexity. This is trivially satisfied for *any* function f in case of $\mathfrak{c}^{\mathsf{query}}$ (given in Example 2).

[19] This is in line with our interpretation of $\mathrm{tr}(\rho) < 1$ as capturing an "abort" of the preparation process.

Example 1. An important class of examples for such a complexity measure arises by considering keyed functions and keyed CPTN maps and writing them as *circuits* that get the global randomness as additional input. The complexity can then be specified to be the minimal number of gates (of certain types) of any such circuit representation. For instance, writing any randomized function f as a binary circuit with AND and XOR gates, one may define $\mathfrak{c}^{\mathsf{comp}}(f)$ to be the minimal necessary number of AND gates.[20] Similarly for randomized CPTN maps, where one could for instance count the number of gates (or the number of non-Clifford gates) with respect to a fixed universal set of gates. These kinds of notions of complexity are referred to as *computational complexity.*

Example 2. Another example that is relevant for us is the *query complexity* $\mathfrak{c}^{\mathsf{query}}$ for oracle functions and oracle CPTN maps, which counts the number of (quantum) oracle queries that the function or CPTN map makes to the oracle $\mathcal{O}$.

We note that this abstract treatment of complexity allows us to *explicitly* cover computational complexity and query complexity in one go, using one language. Also, all results expressed using this language do *explicitly* not depend on the technical details of any model of computation. Related to the latter, this approach allows us to reason about *functions* (and *CPTN* maps etc.), which are unambiguously defined objects that do not depend on any model of computation.

C On the Definitions of the Collapsing Property in [5,6]

As already mentioned in Sect. 3, the definition of the collapsing property introduced by Unruh comes in a few different variations in [5,6], which we want to briefly recall here, and we discuss how they compare to our definition. Some differences (like allowing non-normalized states), which have some minor quantitative impact, have already been discussed in Sect. 3; here, we focus on some technical differences that are orthogonal to those.

The definition originally proposed in [5, Definition 20] (respectively Definition 23 in the full version) is of asymptotic nature and for a deterministic function h (that depends on a security parameter κ): it requires that every (uniform) quantum-polynomial time adversary has a negligible collapsing advantage. Note that it makes sense to consider a *deterministic* hash function h in this case since a *uniform* model of computation is considered, i.e., the adversary that has a collision hard-wired into its code (for any κ) is not allowed. In the formal statement on the collapsing property of a random oracle [5, Theorem 31], which bounds the advantage by $O(q^3/\text{size of the range})$, an obvious variation (in terms of oracle algorithms, and with the randomness also over the oracle's random choices) of this definition is then (implicitly) considered. It also remains implicit that the bound (and in particular the hidden constant) is independent of the running time of the adversary; indeed, the proof considers "q-query adversaries", which have bounded query complexity but (possibly) unbounded running time. As such, the

[20] Remember that we want XOR's to be "for free".

bound $O(q^3/\text{size of the range})$ carries over to our non-asymptotic definition of the collapsing property (with the complexity measure being the query complexity), with the understanding that the bound includes a hidden constant and only applies for a large enough range of h (compared to q).

In terms of comparing Unruh's original definition [5, Definition 20] with our Definition 2, though being similar in spirit (up small differences as discussed in Sect. 3), they are technically incomparable: an asymptotic definition as [5, Definition 20] makes no meaningful statement about a fixed instance, while, on the other hand, our non-asymptotic definition is meaningless for a deterministic hash function, for instance. If we consider an asymptotic variant of our Definition 2, which would ask $\mathsf{cAdv}[h_\kappa](q(\kappa))$ to be negligible in the security parameter κ for any polynomially bounded function q, we get a *non-uniform* (and thus stronger) variant of [5, Definition 20]. It is easy to see that all our results directly carry over to such an asymptotic variant.

In [6, Definition 8], Unruh also considers a non-asymptotic "concrete security" variant of his original definition, which considers a *keyed* hash function (using our terminology) and defines the "collapsing advantage" of an arbitrary but fixed adversary in the obvious way, as the advantage of this adversary distinguishing the two games.[21] For a function h being $\varepsilon(q)$-collapsing according to our definition thus immediately implies that the "collapsing advantage" according to [6, Definition 8] is bounded by $\varepsilon(q)$ (up to a small constant factor, as explained in Sect. 3) for any adversary that is bounded by q, and vice versa. In that sense, our Definition 2 and Unruh's "concrete security" variant [6, Definition 8] are *equivalent* (again, up to negligible quantitative differences).

However, formalized as in [6], i.e., not as a security property of h but as a property of an arbitrary but fixed adversary that is attacking h, security statements are bound to be in reductionistic form. Indeed, all the concrete-security statements in [6] are like:

> "Let A be a q-time (or query) adversary with collapsing advantage ε against function h, then there exists a $2q$-time (or query) adversary A' with collapsing advantage ε^2 against h'"

where we consider concrete example functions for the "security loss". On the other hand, our definition allows us to express such a statement simply as:

> "If the function h' is $\varepsilon'(q')$-collapsing then h is $\sqrt{\varepsilon'(2q)}$-collapsing"

or even more compactly as

$$\mathsf{cAdv}[h](q) \leq \sqrt{\mathsf{cAdv}[h](2q)}\,.$$

Again, these statements are *equivalent* (up to the minor quantitative differences discussed in Sect. 3), so it is merely a matter of taste what sort of language one prefers.

[21] As a matter of fact, [6, Definition 8] considers a t-fold parallel repetition of the game, where t is an additional parameter of the definition. We ignore this for the discussion here, recalling that we obtain immediately a similar variant of our definition by means of concurrent composition.

A more crucial difference is that we not only can state but also *prove* these kinds of security claims "in the forward direction", i.e., not via the counter position of assuming an attacker A that breaks the target primitive and turning it into an attacker A' that breaks something else. Indeed, we have "algebraic" proofs that avoid reasoning about algorithms altogether. On the other hand, the proofs in [2,5,6] are typically constructive, in that the adversary A' is *explicitly* constructed from the adversary A. One can then for instance easily check that A' does not rely on any non-uniform auxiliary information, and thus that the reductions are *uniform*. Our proofs do not spell out the reductions; however, if desired, they could still be extracted by backtracking the proofs all the way down to the basic properties of the pseudometric δ_q upon which the proofs rely, and which all have simple—uniform—reduction proofs. This ensures that also our implicitly defined reductions are uniform.

References

1. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the indifferentiability of the sponge construction. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 181–197. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_11
2. Czajkowski, J., Groot Bruinderink, L., Hülsing, A., Schaffner, C., Unruh, D.: Post-quantum security of the sponge construction. In: Lange, T., Steinwandt, R. (eds.) PQCrypto 2018. LNCS, vol. 10786, pp. 185–204. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-79063-3_9
3. Biham, E., Dunkelman, O.: A framework for iterative hash functions – HAIFA. In: Second NIST Cryptographic Hash Workshop (2006). https://eprint.iacr.org/2007/278.pdf
4. Hülsing, A., Rijneveld, J., Song, F.: Mitigating multi-target attacks in hash-based signatures. In: Cheng, C.-M., Chung, K.-M., Persiano, G., Yang, B.-Y. (eds.) PKC 2016. LNCS, vol. 9614, pp. 387–416. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49384-7_15
5. Unruh, D.: Computationally binding quantum commitments. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 497–527. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_18
6. Unruh, D.: Collapse-binding quantum commitments without random oracles. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 166–195. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_6
7. Winter, A.: Coding theorem and strong converse for quantum channels. IEEE Trans. Inf. Theory **45**(7), 2481–2485 (1999). https://arxiv.org/abs/1409.2536
8. Wilde, M.: From Classical to Quantum Shannon Theory, 2nd edn. (2016). https://arxiv.org/abs/1106.1445. Manuscript