Collapse-binding quantum commitments without random oracles

Dominique Unruh

University of Tartu

Abstract. We construct collapse-binding commitments in the standard model. Collapse-binding commitments were introduced in (Unruh, Eurocrypt 2016) to model the computational-binding property of commitments against quantum adversaries, but only constructions in the random oracle model were known.

Furthermore, we show that collapse-binding commitments imply selected other security definitions for quantum commitments, answering an open question from (Unruh, Eurocrypt 2016).

Keywords: Quantum cryptography, commitments, hash functions.

1 Introduction

Commitment schemes are one of the most fundamental primitives in cryptography. A commitment scheme is a two-party protocol consisting of two phases, the commit and the open phase. The goal of the commitment is to allow the sender to transmit information related to a message m during the commit phase in such a way that the recipient learns nothing about the message (hiding property). But at the same time, the sender cannot change his mind later about the message (binding property). Later, in the open phase, the sender reveals the message m and proves that this was indeed the message that he had in mind earlier (by sending some "opening information" u). Unfortunately, it was shown by [11] that the binding and hiding property of a commitment cannot both hold with statistical (i.e., information-theoretical) security even when using quantum communication. Thus, one typically requires one of them to hold only against computationally-limited adversaries. Since the privacy of data should usually extend far beyond the end of a protocol run, and since we cannot tell which technological advances may happen in that time, we may want the hiding property to hold statistically, and thus are interested in *computationally binding* commitments. Unfortunately, computationally binding commitments turn out to be a subtle issue in the quantum setting. As shown in [1], if we use the natural analogue to the classical definition of computationally binding commitments (called "classical-style binding"),[1] we get a definition that is basically meaningless

[1] This definition, called classical-style style binding in [16], roughly states, that it is computationally hard to find a commitment c, two messages $m \neq m'$ and corresponding valid opening informations u, u'.

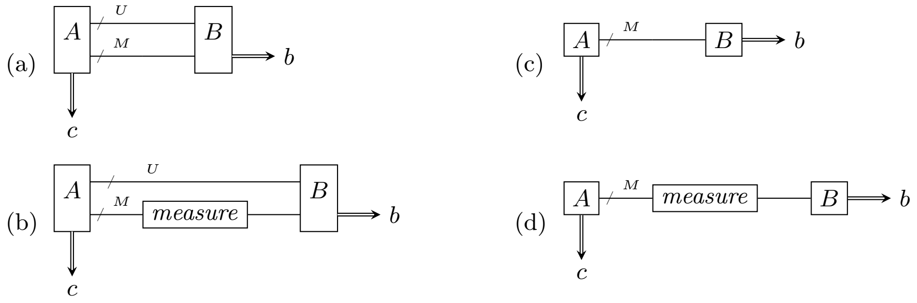


Fig. 1: For collapse-binding commitments, (a) and (b) should be indistinguishable, i.e., $\Pr[b = 1]$ negligibly close in both cases. For collapsing hash functions, (c) and (d) should be indistinguishable.

(the adversary can open the commitment to whatever message he wishes). [16] suggested a new definition, “collapse binding” commitments, that better captures the idea of computationally binding commitments against quantum adversaries. This definition was shown to perform well in security proofs that use rewinding.[2] (They studied classical non-interactive commitments, i.e., all exchanged messages are classical, but the adversary is quantum.)

We describe basic idea of “collapse-binding” commitments: When committing to a message m using a commitment c, it should be impossible for a quantum adversary to produce a superposition of different messages m that he can open to. Unfortunately, this requirement is too strong to achieve (at least for an statistically hiding commitment).[3] Instead, we require something slightly weaker: Any superposition of different messages m that the adversary can open to should *look like* it is a superposition of only a single message m. Formally, if the adversary produces a classical commitment c, and a superposition of openings m, u in registers M, U, the adversary should not be able to distinguish whether M is measured in the computational basis or not measured. That is, for all quantum-polynomial-time A, B, the circuits (a) and (b) in Figure 1 are indistinguishable (assuming A only outputs superpositions that contain only valid openings).

[16] showed that collapse-binding commitments avoid various problems of other definitions of computationally binding commitments in the quantum setting. In particular, they compose in parallel and are well suited for proofs that involve rewinding (e.g., when constructing zero-knowledge arguments of knowledge).

[2] We do not claim that they will work in every rewinding-based proof, but [16] showed their usefulness in the construction of arguments of knowledge. The proof of their construction did involve the quantum rewinding techniques from [17] and [14].

[3] The adversary can initialize a register M with the superposition of all messages, run the commit algorithm in superposition, and measure the resulting commitment c. Then M will still be in superposition between many different messages m which the adversary can open c to.

[16] further showed that in the quantum random oracle model, collapse-binding, statistically hiding commitments can be constructed. However, they left open two big questions:

- Can collapse-binding commitments be constructed in the standard model? That is, without the use of random oracles?
- One standard minimum requirement for commitments (called "sum-binding" in [16]) is that for quantum-polynomial-time A, $p_0 + p_1 \leq 1 + \textit{negligible}$ where p_b is the probability that A opens a commitment to b when he learns b only *after* the commit phase. Surprisingly, [16] left it open whether the collapse-binding property implies the sum-binding property.

First contribution: collapse-binding commitments in the standard model. We show that collapse-binding commitments exist in the standard model. More precisely, we construct a non-interactive, classical commitment in the public parameter model (i.e., we assume that some parameters are globally fixed), for arbitrarily long messages (the length of the public parameters and the commitment itself do not grow with the message length), statistically hiding, and collapse-binding. The security assumption is the existence of lossy trapdoor functions [13] with lossiness rate $> \frac{1}{2}$, or alternatively that SIVP and GapSVP are hard for quantum algorithms to approximate within $\tilde{O}(d^c)$ factors for some constant $c > 5$.

The basic idea of our construction is the following: In [16], it was shown that statistically hiding, collapse-binding commitments can be constructed from "collapsing" hash functions (using a classical construction from [6,9]). A function H is collapsing if an adversary that outputs h and a superposition M of H-preimages of h cannot distinguish whether M is measured or not. That is, the circuits (c) and (d) in Figure 1 should be indistinguishable. So all we need to construct is a collapsing hash function in the standard model.

To do so, we use a lossy trapdoor function (we do not actually need the trapdoor part, though). A lossy function $F_s : A \to B$ is parametrized by a public parameter s. There are two kinds of parameters, which are assumed to be indistinguishable: We call s lossy if $|\operatorname{im} F| \ll |A|$, that is, if its image is very sparse. We call s injective if F_s is injective.

If s is injective, then it is easy to see that F_s is collapsing: There can be only one preimage of F_s on register M, so measuring M will not disturb M. But since lossy and injective s are indistinguishable, it follows that F_s is also collapsing for lossy s. Note, however, that F_s is not yet useful on its own, because its range B is much bigger than A, while we want a compressing hash functions (output smaller than input).

However, for lossy s, $|\operatorname{im} F_s| \ll |A|$. Let $h_r : B \to C$ be a universal hash function, indexed by r, with $|\operatorname{im} F_s| \ll |C| \ll |A|$. We can show that with overwhelming probability, h_r is injective on $\operatorname{im} F_s$, for suitable choice of C. Hence h_r is collapsing (on $\operatorname{im} F_s$). The composition of two collapsing functions is collapsing, thus $H_{(r,s)} := h_r \circ F_s$ is collapsing for lossy s. (Note that $\operatorname{im} F_s$ is not an efficiently decidable set. Fortunately, we can construct all our reductions such that we never need to decide that set.)

Thus far, we have found a collapsing $H_{(r,s)} : A \to C$ that is compressing. But we need something stronger, namely a collapsing hash function $\{0,1\}^* \to C$, i.e., applicable to arbitrary long inputs. A well-known construction (in the classical setting) is the Merkle-Damgård construction, that transforms a compressing collision-resistant function H into a collision-resistant one with domain $\{0,1\}^*$. We prove that the Merkle-Damgård construction also preserves the collapsing property. (This proof is done by a sequence of games that each measure more and more about the hashed message m, each time with a negligible probability of being noticed due to the collapsing property of H_k.) Applying this result to $H_{(r,s)}$, we get a collapsing hash function $\mathrm{MD}_{(r,s)} : \{0,1\}^* \to C$. And from this, we get collapse-binding commitments.

We present our results with concrete security bounds, and our reductions have only constant factors in the runtime, and the security level only has an $O(\mathit{message\ length})$ factor.

We stress that the security proof for the Merkle-Damgård construction has an additional benefit: It shows that existing hash function like SHA-2 [12] are collapsing, assuming that the compression function is collapsing (which in turn is suggested by the random oracle results in [16]). Since we claim that collapsing is a desirable and natural analogue to collision-resistance in the post-quantum setting, this gives evidence for the post-quantum security of SHA-2.

Second contribution: Collapse-binding implies sum-binding. In the classical setting, it relatively straightforward to show that a computationally binding *bit* commitment satisfies the (classical) sum-binding condition. Namely, assume that the adversary breaks sum-binding, i.e., $p_0 + p_1 \geq 1 + \textit{non-negligible}$. Then one runs the adversary, lets him open the commitment as $m = 0$ (which succeeds with probability p_0), then rewinds the adversary, and lets him open the same commitment as $m = 1$ (which succeeds with probability p_1). So the probability that both runs succeed is at least $p_0 + p_1 - 1 \geq \textit{non-negligible}$, which is a contradiction to the computational binding property.

Since collapse-binding commitments work well with rewinding, one would assume that a similar proof works using the quantum rewinding technique from [14]. Unfortunately, existing quantum rewinding techniques do not seem to work.

To show that a collapse-binding commitment is sum-binding, another proof technique is needed. The basic idea is, instead of simulating two executions of the adversary (opening $m = 0$ and opening $m = 1$) after each other, we perform the two executions in superposition, controlled by a register M, initially in state $|+\rangle$. This entangles M with the execution of the adversary and thus disturbs M. It turns out that the disturbance of M is greater if we measure which bit the adversary opens than if we do not. This allows us to distinguish between measuring and not measuring, breaking the collapse-binding property.

The same proof technique can be used to show that a collapse-binding *string* commitment satisfies the generalization of sum-binding presented in [3]. (In this case we have to use a superposition of a polynomial-number of adversary executions.)

Possibly the technique of "rewinding in superposition" used here might be a special case of a more general new quantum rewinding technique (other than [17,14]), we leave this as an open question.

On the necessity of public parameters. Our commitment scheme assumes the existence of public parameters. This raises the question whether these are necessary. We argue that it would be unlikely to be able to construct non-interactive, statistically hiding, computationally binding commitments without public parameters (not even only classically secure ones) from standard assumptions other than collision-resistant or collapsing hash functions. Namely, such a commitment can always be broken by a *non-uniform* adversary. (Because the adversary could have a commitment and two valid openings hardcoded.) Could there be a such a commitment secure only against uniform adversaries, based on some assumption X? That is, a uniform adversary breaking the commitment could be transformed into an adversary against assumption X. All cryptographic proof techniques that we are aware of would then also transform a *non-uniform* adversary breaking the commitment into a *non-uniform* adversary breaking X. Since a non-uniform adversary breaking the commitment always exists, it follows that X must be an assumption that cannot be secure against non-uniform adversaries. The only such assumptions that we are aware of are (unkeyed) collision-resistant and collapsing hash functions.[4] Thus it is unlikely that there are non-interactive, statistically hiding, computationally binding commitments without public parameters based on standard assumptions different from those two. (We are aware that the above constitutes no proof, but we consider it a strong argument.) We know how to construct such commitments from collapsing hash functions [16]. We leave it as an open problem whether such commitments can be constructed from collision-resistant hash functions.

Of course, it might be possible to have *interactive* statistically-hiding collapse-binding commitments. In fact, our construction can be easily transformed into a two-round scheme by letting the recipient choose the public parameters. This does not affect the collapsing property (because for that property we assume the recipient to be trusted), nor the statistical hiding property (because the proof of hiding did not make any assumptions about the distribution of the public parameters).

Related work. Security definitions for quantum commitments were studied in a number of works: What we call the "sum-binding" definition occurred implicitly and explicitly in different variants in [2,11,7,4]. Of these, [11] showed the impossibility of statistically satisfying that definition (thus breaking [2]). [7] gave a construction of a statistically hiding commitment based on quantum one-way permutations (their commitment sends quantum messages). [4] gives statistically secure commitments in the multi-prover setting. [3] generalizes the sum-binding definition for string commitments, arriving at a computational-

[4] By unkeyed hash function, we mean a function that depends only on the security parameter. Such a function might be collision-resistant against uniform adversaries, but not against non-uniform ones.

binding definition we call CDMS-binding. (Both sum-binding and CDMS-binding are implied by collapse-binding as we show in this paper.) [5] gives another definition of computational-binding (called Q-binding in [16]; see there for a discussion of the differences to collapse-binding commitments). They also show how to construct Q-binding commitments from sigma-protocols. (Both their assumptions and their security definition seem incomparable to ours; finding out how their definition relates to ours is an interesting open problem.) [18] gives a statistical binding definition of commitments sending quantum messages and shows that statistically binding, computationally hiding commitments (sending quantum messages) can be constructed from pseudorandom permutations (and thus from quantum one-way functions, if the results from [10] hold in the quantum setting, as is claimed, e.g., in [19]). [16] gave the collapse-binding definition that we achieve in this paper; they showed how to construct statistically hiding, collapse-binding commitments in the random oracle model. [1] showed that classical-style binding does not exclude that the adversary can open the commitment to any value he chooses. [16] generalized this by showing that this even holds for certain natural constructions based on collision-resistant hash functions.

Organization. In Section 2, we give some mathematical preliminaries and cryptographic definitions. In Section 3, we recall the notions of collapse-binding commitments and collapsing hash functions, with suitable extensions to model public parameters and to allow for more refined concrete security statements. We also state some known or elementary facts about collapse-binding commitments and collapsing hash functions there. In Section 4 we show that the Merkle-Damgård construction allows us to get collapsing hash functions with unbounded input length from collapsing compression functions. In Section 5 we show how to construct collapsing hash functions from lossy functions (or from lattice assumptions). Combined with existing results this gives us statistically hiding, collapse-binding commitments for unbounded messages, interactive and non-interactive. In Section 6 we show that collapse-binding implies the existing definitions of sum-binding and CDMS-binding. In the full version [15] we give proofs for getting concrete security bounds. Those proofs use the same techniques as the proofs in this paper, but are somewhat less readable due to additional calculations and indices.

2 Preliminaries

Given a function $f : X \to Y$, let $\operatorname{im} f = f(X)$ denote the image of f.

Given a distribution $\mathcal{D}$ on a countable set X, let $\operatorname{supp}\mathcal{D}$ denote the support of $\mathcal{D}$, i.e., the set of all values that have non-zero probability. The statistical distance between two distributions or random variables X, Y with countable range is defined as $\frac{1}{2}\sum_a \bigl|\Pr[X = a] - \Pr[Y = a]\bigr|$.

Let λ denote the empty word.

We assume that all algorithms and parameters depend on an integer $\eta > 0$, the security parameter (unless a parameter is explicitly called "constant"). We will keep this dependence implicit (i.e., we write $A(x)$ instead of $A(\eta, x)$ for an algorithm A, and ℓ instead of $\ell(\eta)$ for an integer parameter ℓ). When calling an adversary (quantum-)polynomial-time, we mean that the runtime is polynomial in η.

We do not specify whether our adversaries are uniform or non-uniform. (I.e., whether the adversary's code may depend in an noncomputable way on the security parameter.) All our results hold both in the uniform and in the non-uniform case.

Definition 1 (Universal hash function). *A universal hash function is a function family $h_r : X \to Y$ (with $r \in R$) such that for any $x, x' \in X$ with $x \neq x'$, we have* $\Pr[h_r(x) = h_r(x') : r \xleftarrow{\$} R] = 1/|Y|$.

We define lossy functions, which are like lossy trapdoor functions [13], except that we do not require the existence of a trapdoor.

Definition 2 (Lossy functions). *A collection of (ℓ, k)-*lossy functions *consists of a PPT algorithm S_F and polynomial-time computable deterministic function F_s on $\{0,1\}^\ell$ and a message space M_k such that:*

- *Existence of injective keys: There is a distribution $\mathcal{D}_{inj}$ such that for any $s \in \operatorname{supp} \mathcal{D}_{inj}$ we have that F_s is injective. (We call such a key s injective.)*
- *Existence of lossy keys: There is a distribution $\mathcal{D}_{lossy}$ such that for any $s \in \operatorname{supp} \mathcal{D}_{lossy}$ we have that $|\operatorname{im} F_s| \leq 2^{\ell-k}$. (We call such a key s lossy.)*
- *Hard to distinguish injective from lossy: For any quantum-polynomial-time adversary A, the advantage $\bigl|\Pr[A(s) = 1 : s \leftarrow \mathcal{D}_{inj}] - \Pr[A(s) = 1 : s \leftarrow \mathcal{D}_{lossy}]\bigr|$ is negligible.*
- *Hard to distinguish lossy from S: For any quantum-polynomial-time adversary A, the advantage $\bigl|\Pr[A(s) = 1 : s \leftarrow \mathcal{D}_{lossy}] - \Pr[A(s) = 1 : s \leftarrow S_F]\bigr|$ is negligible.*

The parameter k is called the lossiness *of F_s.*

This is a weakening of the definition of lossy trapdoor functions from [13]. Our definition does not require the existence of trapdoors, and also does not require that lossy or injective keys can be efficiently sampleable. (We only require that keys that are indistinguishable from both lossy and injective keys can be sampled efficiently using S_F.)

If $k/\ell \geq K$ for some constant K, and $\ell \in \omega(\log \eta)$, we say that the lossy function has *lossiness rate K*.

Any "almost-always lossy trapdoor function" $(S_{\mathrm{ldtf}}, F_{\mathrm{ldtf}}, F_{\mathrm{ldtf}}^{-1})$ in the sense of [13] is a lossy function in the sense of Definition 2.[5]

[13] shows that for any constant $K < 1$, there is an almost-always lossy trapdoor function with lossiness rate K based on the LWE assumption for

[5] To see that, let $\mathcal{D}_{inj}$ be the distribution of the first output (i.e., discarding the trapdoor) of the injective key sampler $S_{\mathrm{ldtf}}(\eta, 1)$ conditioned on outputting an injective key. Let $\mathcal{D}_{lossy}$ be the distribution of the first output of the lossy key sampler $S_{\mathrm{ldtf}}(\eta, 0)$ conditioned on outputting a lossy key. Let S_F return the first output of $S_{\mathrm{ldtf}}(\eta, 0)$ (or $S_{\mathrm{ldtf}}(\eta, 1)$). Let $F_k(x) := F_{\mathrm{ldtf}}(k, x)$. For those choices, it is easy to see that (S_F, F_k) satisfies Definition 2.

suitable parameters. [13] further shows that almost-always (ℓ, k)-lossy trapdoor functions with lossiness rate K exist if SIVP and GapSVP are hard for quantum algorithms to approximate within $\tilde{O}(d^c)$ factors, where $c = 2 + \frac{3}{2(1-K)} + \delta$ for any desired $\delta > 0$. The same thus holds for lossy functions in our sense. Furthermore, the construction from [13] has keys that are indistinguishable from uniformly random, hence we can choose S_F to simply return $s \xleftarrow{\$} \{0,1\}^{\ell_s}$ for suitable ℓ_s.[6]

3 Collapse-binding commitments and collapsing hash functions

We reproduce the relevant results from [16] here. Note we have extended the definitions in two ways: We include a public parameter $k \leftarrow \mathbf{P}$. And we give additional equivalent definitions for a more refined treatment of the concrete security of commitments.

Commitments. A *commitment scheme* consists of three algorithms $(\mathbf{P}, \mathsf{com}, \mathsf{verify})$. $k \leftarrow \mathbf{P}$ chooses the public parameter. $(c, u) \leftarrow \mathsf{com}(k, m)$ produces a commitment c for a message m, and also returns opening information u to be revealed later. $ok \leftarrow \mathsf{verify}(k, c, m, u)$ checks whether the opening information u is correct for a given commitment c and message m (if so, $ok = 1$, else $ok = 0$).

Definition 3 (Collapse-binding). *For algorithms (A, B), consider the following games:*

$$\begin{array}{lllll} \mathsf{Game}_1: & k \leftarrow \mathbf{P}, & (S,M,U,c) \leftarrow A(k), & m \leftarrow \mathcal{M}(M), & b \leftarrow B(S,M,U) \\ \mathsf{Game}_2: & k \leftarrow \mathbf{P}, & (S,M,U,c) \leftarrow A(k), & & b \leftarrow B(S,M,U) \end{array}$$

Here S, M, U are quantum registers. $\mathcal{M}(M)$ is a measurement of M in the computational basis.

We call an adversary (A, B) c.b.-valid for verify *iff for all k,* $\Pr[\mathsf{verify}(k, c, m, u) = 1] = 1$ *when we run $(S, M, U, c) \leftarrow A(k)$ and measure M in the computational basis as m, and U in the computational basis as u.*

[6] This is not explicitly mentioned in [13], but can be seen as follows: [13] constructs a matrix encryption scheme whose ciphertexts are pairs of matrices $(\mathbf{A}, \mathbf{C}')$ over $\mathbb{Z}_q$ for a suitable prime q. We can see those ciphertexts as a tuple $s' \in \mathbb{Z}_q^n$ for some n. The proof of Lemma 6.2 in [13, full version] shows that the matrix encryption scheme produces ciphertexts that are indistinguishable from uniformly random $s' \xleftarrow{\$} \mathbb{Z}_q^n$.

The lattice-based lossy trapdoor function from [13] uses a ciphertext of that lossy encryption scheme as its key. Thus a key is indistinguishable from $s' \xleftarrow{\$} \mathbb{Z}_q^n$. Hence we can choose S_F to simply return a uniformly random $s' \xleftarrow{\$} \mathbb{Z}_q^n$.

To get an S_F that returns $s \xleftarrow{\$} \{0,1\}^{\ell_s}$ instead, we let S_F choose $s \in \{0, \dots, 2^\ell - 1\}^n$ and set $s'_i := s_i \bmod q$. For sufficiently large ℓ, this changes the distribution of s' only by a negligible amount. Then s can be encoded as an ℓ_s-bitstring with $\ell_s := n\ell$. (Since this way of sampling s'_i is "oblivious", i.e., given s'_i we can efficiently find randomness s_i that leads to that s'_i, the security of the lossy function is not affected by outputting s_i as the key instead of s'_i.)

A commitment scheme is collapse-binding *iff for any quantum-polynomial-time adversary (A,B) that is c.b.-valid for* verify, $\bigl|\Pr[b=1:\mathsf{Game}_1]-\Pr[b=1:\mathsf{Game}_2]\bigr|$ *is negligible.*

The only difference to the definition from [16] is that we have introduced a public parameter k chosen by $\mathbf{P}$. The proofs in [16] are not affected by this change.

For stating concrete security results (i.e., with more specific claims about the runtimes and advantages of adversaries than "polynomial-time" and "negligible"), we could simply call $\bigl|\Pr[b=1:\mathsf{Game}_1]-\Pr[b=1:\mathsf{Game}_2]\bigr|$ the advantage of the adversary (A,B). However, we find that we get stronger results if we directly specify the advantage of an adversary that attacks t commitments simultaneously.[7] This leads to the following definition of advantage. (A reader only interested in asymptotic results may ignore this definition. The main body of this paper will provide statements and proofs with respect to the simpler asymptotic definitions. Concrete security proofs are given in the full version [15].)

Definition 4 (Collapse-binding – concrete security). *For algorithms (A,B), consider the following games:*

$$\begin{aligned}
\mathsf{Game}_1:\ & k\leftarrow\mathbf{P},\ (S,M_1,\dots,M_t,U_1,\dots,U_t,c_1,\dots,c_t)\leftarrow A(k),\\
& m_1\leftarrow\mathcal{M}(M_1),\ \dots,\ m_t\leftarrow\mathcal{M}(M_t),\\
& b\leftarrow B(S,M_1,\dots,M_t,U_1,\dots,U_t)\\
\mathsf{Game}_2:\ & k\leftarrow\mathbf{P},\ (S,M_1,\dots,M_t,U_1,\dots,U_t,c_1,\dots,c_t)\leftarrow A(k),\\
& b\leftarrow B(S,M_1,\dots,M_t,U_1,\dots,U_t)
\end{aligned}$$

Here $S,M_1,\dots,M_t,U_1,\dots,U_t$ are quantum registers. $\mathcal{M}(M_i)$ is a measurement of M_i in the computational basis.

We call an adversary (A,B) t-c.b.-valid for verify *iff for all k, $\Pr[\forall i.\ \mathsf{verify}(k,c_i,m_i,u_i)=1]=1$ when we run $(S,M_1,\dots,M_t,U_1,\dots,U_t,c_1,\dots,c_t)\leftarrow A(k)$ and measure all M_i in the computational basis as m_i, and all U_i in the computational basis as u_i.*

For any adversary (A,B), we call $\bigl|\Pr[b=1:\mathsf{Game}_1]-\Pr[b=1:\mathsf{Game}_2]\bigr|$ the collapse-binding-advantage *of (A,B) against $(\mathbf{P},\mathsf{com},\mathsf{verify})$.*

Lemma 5. *A commitment scheme $(\mathbf{P},\mathsf{com},\mathsf{verify})$ is collapse-binding iff for any polynomially-bounded t, and any quantum-polynomial-time adversary (A,B) that is t-c.b.-valid for* verify, *the collapse-binding-advantage of (A,B) against $(\mathbf{P},\mathsf{com},\mathsf{verify})$ is negligible.*

[7] We could simply analyze all schemes for adversaries that attack a single commitment at a time, and then invoke the parallel composition theorem from [16] to get the advantage when attacking t commitments. That theorem will then introduce a factor t in the advantage. ([16] states the theorem without concrete security bounds, but they are easily extracted from the proof.) In contrast, a direct analysis for t commitments may give better bounds, since the advantages we get in this paper do not depend on t.

This follows from the parallel composition theorem from [16].

In [16], two different definitions of collapse-binding were given. The second definition does not require an adversary to be valid (i.e., to output only valid openings) but instead measures whether the adversary's openings are valid. We restate the equivalence here in the public parameter setting, the proof is essentially unchanged.

Lemma 6 (Collapse-binding, alternative characterization). *For a commitment scheme* $(\mathbf{P}, \mathsf{com}, \mathsf{verify})$*, and for algorithms* (A, B)*, consider the following games:*

$$\begin{aligned}
&\mathsf{Game}_1: \ k \leftarrow \mathbf{P}, (S, M, U, c) \leftarrow A(k), ok \leftarrow V_c(M, U), x \leftarrow \mathcal{M}_{ok}(M), b \leftarrow B(S, M, U)\\
&\mathsf{Game}_2: \ k \leftarrow \mathbf{P}, (S, M, U, c) \leftarrow A(k), ok \leftarrow V_c(M, U), \qquad\qquad\qquad\quad b \leftarrow B(S, M, U)
\end{aligned}$$

Here V_c *is a measurement whether* M, U *contains a valid opening. Formally* V_c *is defined through the projector* $\sum_{\substack{m,u\\ \mathsf{verify}(k,c,m,u)=1}} |m\rangle\langle m| \otimes |u\rangle\langle u|$*.* $\mathcal{M}_{ok}$ *is a measurement of* M *in the computational basis if* $ok = 1$*, and does nothing if* $ok = 0$ *(i.e., it sets* $m := \bot$ *and does not touch the register* M*).*

$(\mathbf{P}, \mathsf{com}, \mathsf{verify})$ *is collapse-binding iff for all polynomial-time adversaries* (A, B)*,* $\bigl|\Pr[b = 1 : \mathsf{Game}_1] - \Pr[b = 1 : \mathsf{Game}_2]\bigr|$ *is negligible.*

Hash functions. A *hash function* is a pair $(\mathbf{P}, H_k)$ of a parameter sampler $\mathbf{P}$ and a function $H_k : X \to Y$ for some range X and domain Y. H_k is parametric in the public parameter $k \leftarrow \mathbf{P}$. (Typically, Y consists of fixed length bitstrings, and X consists of fixed length bitstrings or $\{0, 1\}^*$.)

Definition 7 (Collapsing). *For algorithms* A*,* B*, consider the following games:*

$$\begin{aligned}
&\mathsf{Game}_1: \quad k \leftarrow \mathbf{P}, \ (S, M, h) \leftarrow A(k), \ m \leftarrow \mathcal{M}(M), \ b \leftarrow B(S, M)\\
&\mathsf{Game}_2: \quad k \leftarrow \mathbf{P}, \ (S, M, h) \leftarrow A(k), \qquad\qquad\qquad b \leftarrow B(S, M)
\end{aligned}$$

Here S, M *are quantum registers.* $\mathcal{M}(M)$ *is a measurement of* M *in the computational basis.*

For a family of sets $\mathbf{M}_k$*, we call an adversary* (A, B) valid on $\mathbf{M}_k$ for H_k *iff for all* k*,* $\Pr[H_k(m) = c \ \wedge \ m \in \mathbf{M}_k] = 1$ *when we run* $(S, M, h) \leftarrow A(k)$ *and measure* M *in the computational basis as* m*. If we omit "on* $\mathbf{M}_k$*", we assume* $\mathbf{M}_k$ *to be the domain of* H_k*.*

A function H *is* collapsing *(on* $\mathbf{M}_k$*) iff for any quantum-polynomial-time adversary* (A, B) *that is valid for* H_k *(on* $\mathbf{M}_k$*),* $\bigl|\Pr[b = 1 : \mathsf{Game}_1] - \Pr[b = 1 : \mathsf{Game}_2]\bigr|$ *is negligible.*

In contrast to [16] we have added the public parameter k. Furthermore, we have extended the definition to allow to specify the set $\mathbf{M}_k$ of messages the adversary is allowed to use. This extra expressiveness will be needed for stating some intermediate results.

Analogously to case of commitments, we give a definition of advantage for a t-session adversary to get more precise results.

Definition 8 (Collapsing – concrete security). *For algorithms A, B, and an integer t, consider the following games:*

$$\begin{aligned}
\mathsf{Game}_1: \quad & k \leftarrow \mathbf{P},\ (S, M_1, \dots, M_t, h_1, \dots, h_t) \leftarrow A(k), \\
& m_1 \leftarrow \mathcal{M}(M_1),\ \dots,\ m_t \leftarrow \mathcal{M}(M_t), \\
& b \leftarrow B(S, M_1, \dots, M_t) \\
\mathsf{Game}_2: \quad & k \leftarrow \mathbf{P},\ (S, M_1, \dots, M_t, h_1, \dots, h_t) \leftarrow A(k), \\
& b \leftarrow B(S, M_1, \dots, M_t)
\end{aligned}$$

Here $S, M_1, \dots, M_t$ are quantum registers. $\mathcal{M}(M_i)$ is a measurement of M_i in the computational basis.

For a family of sets $\mathbf{M}_k$, we call an adversary (A, B) t-valid on $\mathbf{M}_k$ for H_k *iff for all k, $\Pr[\forall i.\ H_k(m_i) = c_i \ \wedge \ m_i \in \mathbf{M}_k] = 1$ when we run $(S, M_1, \dots, M_t, h_1, \dots, h_t) \leftarrow A(k)$ and measure all M_i in the computational basis as m_i. If we omit "on $\mathbf{M}_k$", we assume $\mathbf{M}_k$ to be the domain of H_k.*

We call $adv := \bigl|\Pr[b = 1 : \mathsf{Game}_1] - \Pr[b = 1 : \mathsf{Game}_2]\bigr|$ the collapsing-advantage *of (A, B) against $(\mathbf{P}, H_k)$.*

Lemma 9. *A hash function $(\mathbf{P}, H_k)$ is collapsing (on $\mathbf{M}_k$) iff for any polynomially-bounded t, and any quantum-polynomial-time adversary (A, B) that is t-valid for H_k (on $\mathbf{M}_k$), the collapsing-advantage of (A, B) against $(\mathbf{P}, H_k)$ is negligible.*

This follows from the parallel composition theorem for hash functions from [16].

Constructions of commitments. In [16] it was shown that the statistically hiding commitment from Halevi and Micali [9] (which is almost identical to the independently and earlier discovered commitment by Damgård, Pedersen, and Pfitzmann [6]) is collapse-binding, assuming a collapsing hash function. We restate their results with respect to public parameters, the proofs are essentially unchanged.

Definition 10 (Unbounded Halevi-Micali commitment [9]). *Let $(\mathbf{P}, H_k)$ with $H_k : \{0,1\}^* \to \{0,1\}^\ell$ be a hash function. Let $L := 6\ell + 4$. Let $h_r : \{0,1\}^L \to \{0,1\}^\ell$ with $r \in \{0,1\}^{\ell_r}$ be an universal hash function.*

We define the unbounded Halevi-Micali commitment $(\mathbf{P}, \mathsf{com}_{HMu}, \mathsf{verify}_{HMu})$ *as:*

- $\mathbf{P}$ *is the same parameter sampler as in $(\mathbf{P}, H_k)$.*
- $\mathsf{com}_{HMu}(k, m)$*: Pick $r \in \{0,1\}^{\ell_r}$ and $u \in \{0,1\}^L$ uniformly at random, conditioned on $h_r(u) = H_k(m)$.*[8] *Compute $h := H_k(u)$. Let $c := (h, r)$. Return commitment c and opening information u.*

[8] In general, this can be computationally hard. However, should h_r be a universal hash function where this is hard, one can replace h_r by $h'_{(r,t)}$ defined as $h'_{(r,t)}(x) := t \oplus h_r(x)$. This function is still a universal hash function, and sampling r, t, u is easy.

- $\mathsf{verify}_{HMu}(k, c, m, u)$ *with* $c = (h, r)$*: Check whether* $h_r(u) = H_k(m)$ *and* $h = H_k(u)$*. If so, return* 1*.*

We define the statistical hiding property in the public parameter model. We use an adaptive definition where the committed message may depend on the public parameter.

Definition 11 (Statistically hiding). *Fix a commitment* $(\mathbf{P}, \mathsf{com}, \mathsf{verify})$ *and an adversary* (A, B)*. Let*

$$p_b := \Pr[b' = 1 : k \leftarrow \mathbf{P},\ (S, m_0, m_1) \leftarrow A(k),\ (c, u) \leftarrow \mathsf{com}(k, m_b),\ b' \leftarrow B(S, c)].$$

We call $|p_0 - p_1|$ *the* hiding-advantage *of* (A, B)*. We call* $(\mathbf{P}, \mathsf{com}, \mathsf{verify})$ statistically hiding *iff for any (possibly unbounded)* (A, B)*, the hiding-advantage is negligible.*

Theorem 12 (Security of the unbounded Halevi-Micali commitment). $(\mathbf{P}, \mathsf{com}_{HMu}, \mathsf{verify}_{HMu})$ *is statistically hiding and collapse-binding.*

Miscellaneous facts. These simple facts will be useful throughout the paper.

Lemma 13. *Let* $\mathbf{M}_k$ *be a family of sets. Assume that* $\Pr[H_k \text{ is not injective on } \mathbf{M}_k : k \overset{\$}{\leftarrow} \mathbf{P}]$ *is negligible. Then* $(\mathbf{P}, H_k)$ *is collapsing on* $\mathbf{M}_k$*.*

Lemma 14. *Fix hash functions* $(\mathbf{P}, f_k)$ *and* $(\mathbf{P}, g_k)$ *with the same* $\mathbf{P}$ *and with polynomial-time computable* f_k*. If* $(\mathbf{P}, f_k)$ *is collapsing and* $(\mathbf{P}, g_k)$ *is collapsing on* $\operatorname{im} f_k$*, then* $(\mathbf{P}, g_k \circ f_k)$ *is collapsing.*

Lemma 15. *If* $\mathbf{P}_1$ *and* $\mathbf{P}_2$ *are computationally indistinguishable, and* $(\mathbf{P}_1, H_k)$ *is collapsing, then* $(\mathbf{P}_2, H_k)$ *is collapsing.*

4 Security of Merkle-Damgård hashes

For this section, fix a hash function $(\mathbf{P}, H_k)$ with $H_k : \{0,1\}^{\ell_{in}} \to \{0,1\}^{\ell_{out}}$ and $\ell_{in} > \ell_{out}$. Let $\ell_{block} := \ell_{in} - \ell_{out}$. Fix some bitstring $iv \in \{0,1\}^{\ell_{out}}$ (may depend on the security parameter). Fix a message space $\mathbf{M}$ with $|\mathbf{M}| \geq 2$ (e.g., $\mathbf{M} = \{0,1\}^*$). Fix a function $pad : \mathbf{M} \to (\{0,1\}^{\ell_{block}})^*$.

Definition 16 (Iterated hash). *We define the* iterated hash $\mathrm{IH}_k : (\{0,1\}^{\ell_{block}})^* \to \{0,1\}^{\ell_{out}}$ *as* $\mathrm{IH}_k(\lambda) := iv$ *for the empty word* λ *and* $\mathrm{IH}_k(\mathbf{m}\|m') := H_k(\mathrm{IH}_k(\mathbf{m})\|m')$ *for* $\mathbf{m} \in (\{0,1\}^{\ell_{block}})^*$ *and* $m' \in \{0,1\}^{\ell_{block}}$*.*

Definition 17 (Merkle-Damgård). *We call* pad *a* Merkle-Damgård padding *iff* pad *is injective and for any* $x, y \in \mathbf{M}$ *with* $x \neq y$*, we have that* $pad(x)$ *is not a suffix of* $pad(y)$ *(in other words,* $pad(\mathbf{M})$ *is a suffix code).*[9]

We define the Merkle-Damgård construction $\mathrm{MD}_k : \mathbf{M} \to \{0,1\}^{\ell_{out}}$ *by* $\mathrm{MD}_k := \mathrm{IH}_k \circ pad$*.*

[9] Commonly, stronger conditions are placed on pad, see, e.g., [8, Def. 8.7]. However, "suffix-code" and "injective" turns out to be sufficient. For example, the padding using in SHA-256 [12] is a Merkle-Damgård padding for $\mathbf{M} = \{0,1\}^{\leq 2^{64}-1}$ according to our definition.

Note that IH_k and MD_k depend on the choice of H_k, iv, and pad, but we leave this dependence implicit for brevity.

Lemma 18 (Security of iterated hash). *Let $\tilde{\mathbf{M}} \subseteq (\{0,1\}^{\ell_{block}})^*$ be a suffix code with $|\tilde{\mathbf{M}}| \geq 2$. If $(\mathbf{P}, H_k)$ is a polynomial-time computable collapsing hash function, then $(\mathbf{P}, \mathrm{IH}_k)$ is collapsing on $\tilde{\mathbf{M}}$.*

We sketch the idea of the proof: What we have to show is that, if the adversary classically outputs $\mathrm{IH}_k(\mathbf{m})$, we can measure $\mathbf{m}$ on register M without the adversary noticing. We show this by successively measuring more and more information about the message $\mathbf{m}$ on M, each time noting that the additional measurement is not noticed by the adversary. First, measuring $\mathrm{IH}_k(\mathbf{m})$ does not disturb M because $\mathrm{IH}_k(\mathbf{m})$ is already known. Note that $\mathrm{IH}_k(\mathbf{m}) = H_k(\mathrm{IH}_k(\mathbf{m}')\|m)$ for $\mathbf{m} =: \mathbf{m}'\|m$. Thus, we have measured the image of $\mathrm{IH}_k(\mathbf{m}')\|m$ under H_k. Since H_k is collapsing, we know that, once we have measured the hash of a value, we can also measure that value itself without being noticed. Thus we can measure $\mathrm{IH}_k(\mathbf{m}')\|m$ (this value will be called $\mathrm{step}_0(\mathbf{m})$ in the full proof). Now we use the same argument again: $\mathrm{IH}_k(\mathbf{m}') = H_k(\mathrm{IH}_k(\mathbf{m}'')\|m')$ for $\mathbf{m}' =: \mathbf{m}''\|m'$. Since we know classically $\mathrm{IH}_k(\mathbf{m}')$, we can measure $\mathrm{IH}_k(\mathbf{m}'')\|m'$ (this value will be called $\mathrm{step}_1(\mathbf{m})$). Now we already have measured the two last blocks $m'\|m$ of $\mathbf{m}$ without being noticed. We can continue this way, until we have all of $\mathbf{m}$. Since in each step, the adversary did not notice the measurement, he will not notice if we measure all of $\mathbf{m}$.

There is one hidden problem in the above argument: We claimed that given $\mathrm{IH}_k(\mathbf{m}')$, we have that $\mathrm{IH}_k(\mathbf{m}') = H_k(\mathrm{IH}_k(\mathbf{m}'')\|m')$. This is only correct if $\mathbf{m}'$ is not empty! So, the above measurement procedure will implicitly measure whether $\mathbf{m}'$ is empty (and similarly for the values $\mathbf{m}''$ etc. that are measured afterwards). Such a measurement might disturb the state. Here the assumption comes in that $\tilde{\mathbf{M}}$ is a suffix code. Namely, since we know m such that $\mathbf{m} = \mathbf{m}'\|m$, we can tell whether $m \in \tilde{\mathbf{M}}$ (then $\mathbf{m}'$ must be empty) or $m \notin \tilde{\mathbf{M}}$ (then $\mathbf{m}'$ cannot be empty). Thus we already know whether $\mathbf{m}'$ is empty, and measuring this information will not disturb the state. Similarly, we deduce from $m'\|m$ whether $\mathbf{m}''$ is empty, etc.

We now give the formal proof:

Proof of Lemma 18. Assume a polynomial-time adversary (A, B) that is valid for IH_k on $\mathbf{M}$. Let Game_1 and Game_2 be the games from Definition 7 for adversary (A, B). Let

$$\varepsilon := \bigl|\Pr[b = 1 : \mathsf{Game}_1] - \Pr[b = 1 : \mathsf{Game}_2]\bigr|. \tag{1}$$

We will need to show that ε is negligible.

We have $\lambda \notin \tilde{\mathbf{M}}$. ($\lambda$ denotes the empty word.) Otherwise, we would have $\tilde{\mathbf{M}} = \{\lambda\}$ since $\tilde{\mathbf{M}}$ is a suffix code, which contradicts $|\tilde{\mathbf{M}}| \geq 2$.

For a multi-block message $\mathbf{m} \in (\{0,1\}^{\ell_{block}})^*$, let $|\mathbf{m}|$ denote the number of ℓ_{block}-bit blocks in $\mathbf{m}$. (I.e., $|\mathbf{m}|$ is the bitlength of $\mathbf{m}$ divided by ℓ_{block}.) Let $\mathbf{m}_i$ denote the i-th block of $\mathbf{m}$, and let $\mathbf{m}_{-i}$ denote the i-th block from the end (i.e., $\mathbf{m}_{-i} = \mathbf{m}_{|m|-i+1}$). Let $\mathbf{m}_{\geq -i}$ denote all the blocks in $\mathbf{m}$ starting from $\mathbf{m}_{-i}$ (i.e., $\mathbf{m}_{\geq -i}$ consists of the last i blocks of $\mathbf{m}$). Let $\mathbf{m}_{<-i}$ denote the blocks before $\mathbf{m}_{-i}$. (I.e., $\mathbf{m} = \mathbf{m}_{<-i} \,\|\, \mathbf{m}_{\geq -i}$ for $i \leq |\mathbf{m}|$.)

Let B be a polynomial upper bound on the number of blocks in the message $\mathbf{m}$ output by A on register M.

For a function f, let $\mathcal{M}_f(M)$ denote a measurement that, given a register M that contains values $|\mathbf{m}\rangle$ in superposition, measures $f(\mathbf{m})$, but without measuring more information than that. Formally, $\mathcal{M}_f$ is a projective measurement consisting of projectors P_y $(y \in \operatorname{im} f)$ with $P_y = \sum_{\mathbf{m}: f(\mathbf{m})=y} |\mathbf{m}\rangle\langle\mathbf{m}|$.

For $\mathbf{m} \in \tilde{\mathbf{M}}$, we define

$$\operatorname{partial}_i(\mathbf{m}) := \begin{cases} (\bot, \mathbf{m}) & (\text{if } |\mathbf{m}| \leq i) \\ (\mathrm{IH}_k(\mathbf{m}_{<-i}), \mathbf{m}_{\geq -i}) & (\text{if } |\mathbf{m}| > i) \end{cases}$$

(The function $\operatorname{partial}_i$ also depends on k, but we leave that dependence implicit.) Intuitively, $\operatorname{partial}_i(\mathbf{m})$ represents a partial evaluation of $\mathrm{IH}_k(\mathbf{m})$, with the last i blocks not yet processed.

Note that $\operatorname{partial}_i(\mathbf{m})$ always contains enough information to compute $\mathrm{IH}_k(\mathbf{m})$. And the larger i is, the more about $\mathbf{m}$ is revealed. In fact, learning $\operatorname{partial}_0(\mathbf{m})$ is equivalent to learning $\mathrm{IH}_k(\mathbf{m})$, and learning $\operatorname{partial}_B(\mathbf{m})$ is equivalent to learning m as the following easy to verify facts show:

Fact 1 $\operatorname{partial}_0(\mathbf{m}) = (\mathrm{IH}_k(\mathbf{m}), \lambda)$ *for all* $\mathbf{m} \in \tilde{\mathbf{M}}$.

Fact 2 $\operatorname{partial}_B(\mathbf{m}) = (\bot, \mathbf{m})$ *for all* $\mathbf{m} \in \tilde{\mathbf{M}}$ *with* $|\mathbf{m}| \leq B$.

We will need one additional auxiliary function step_i, defined by $\operatorname{step}_i(\mathbf{m}) := \mathrm{IH}_k(\mathbf{m}_{<-(i+1)}) \| \mathbf{m}_{-(i+1)}$ for $|\mathbf{m}| \geq i+1$. (And $\operatorname{step}_i(\mathbf{m}) := \bot$ if $|\mathbf{m}| \leq i$.) Intuitively, $\operatorname{step}_i(\mathbf{m})$ is the input to last call of H_k when computing $\operatorname{partial}_i(\mathbf{m})$. The following facts are again easy to verify using the definition of $\operatorname{partial}_i$, step_i, and IH_k:

Fact 3 *If* $\operatorname{partial}_i(\mathbf{m}) = (h, s)$ *and* $h \neq \bot$, *then* $H_k(\operatorname{step}_i(\mathbf{m})) = h$.

Fact 4 *From* $(\operatorname{partial}_i(\mathbf{m}), \operatorname{step}_i(\mathbf{m}))$ *one can compute* $\operatorname{partial}_{i+1}(\mathbf{m})$ *and vice versa. Formally: there are functions* f, g *such that for all* $m \in \tilde{\mathbf{M}}$, $f(\operatorname{partial}_i(\mathbf{m}), \operatorname{step}_i(\mathbf{m})) = \operatorname{partial}_{i+1}(\mathbf{m})$ *and* $g(\operatorname{partial}_{i+1}(\mathbf{m})) = (\operatorname{partial}_i(\mathbf{m}), \operatorname{step}_i(\mathbf{m}))$.

In a sense, $\operatorname{partial}_i(\mathbf{m})$ interpolates between the knowledge of only $\mathrm{IH}_k(\mathbf{m})$ (case $i = 0$), and full knowledge of $\mathbf{m}$ (case $i = B$). (Cf. Facts 1, 2.) We make this more formal by defining the following hybrid game for $i = 0, \dots, B$:

$$\begin{aligned} \mathsf{Game}_i^{hyb}: \quad & k \leftarrow \mathbf{P},\ (S, M, h) \leftarrow A(k), \\ & (h', s) \leftarrow \mathcal{M}_{\operatorname{partial}_i}(M), \\ & b \leftarrow B(S, M). \end{aligned}$$

(Here $\mathcal{M}_{\operatorname{partial}_i}$ is $\mathcal{M}_f$ as defined above with $f := \operatorname{partial}_i$.)

Consider Game_0^{hyb}. By assumption, (A, B) is valid for IH_k on $\tilde{\mathbf{M}}$, so we have that the register M contains superpositions of states $|\mathbf{m}\rangle$ with $\mathrm{IH}_k(\mathbf{m}) = h_j$

and $\mathbf{m} \in \tilde{\mathbf{M}}$. By Fact 1, this implies that the measurement $\mathcal{M}_{\text{partial}_0}(M)$ will always yield the outcome $(h', s) = (h, \lambda)$. Hence the measurement $\mathcal{M}_{\text{partial}_0}(M)$ has a deterministic outcome. Thus, the probability of $b = 1$ in Game_0^{hyb} does not change if we omit the measurements $y \leftarrow \mathcal{M}_{\text{partial}_i}(M)$. Thus

$$\Pr[b = 1 : \mathsf{Game}_0^{hyb}] = \Pr[b = 1 : \mathsf{Game}_2]. \tag{2}$$

Consider Game_B^{hyb}. By assumption, A outputs only states on M which are superpositions of $|\mathbf{m}\rangle$ with $\mathbf{m} \in \tilde{\mathbf{M}}$ and $|\mathbf{m}| \le B$. Thus, by Fact 2, $(h', s) \leftarrow \mathcal{M}_{\text{partial}_B}(M)$ is a complete measurement in the computational basis. Hence

$$\Pr[b = 1 : \mathsf{Game}_B^{hyb}] = \Pr[b = 1 : \mathsf{Game}_1]. \tag{3}$$

From (1,2,3), we get

$$\bigl|\Pr[b = 1 : \mathsf{Game}_0^{hyb}] - \Pr[b = 1 : \mathsf{Game}_B^{hyb}]\bigr| = \varepsilon. \tag{4}$$

For $i = 0, \dots, B$ we now define an adversary (A_i^*, B^*) against H_k.

```
Algorithm $A_i^*(k)$ runs:
 – $(S^*, M^*, h^*) \leftarrow A(k)$.
 – $(h', s) \leftarrow \mathcal{M}_{\text{partial}_i}(M^*)$.
 – Initialize $M$ with $|0^{\ell_{in}}\rangle$.
 – If $h' \neq \bot$:
     • Apply $U_{\text{step}_i}$ to $M^*, M$.
     • $h := h'$.
 – If $h' = \bot$:
     • Let $h := H_k(0^{\ell_{in}})$.
 – Let $S := S^*, M^*, h', i$. (That is, all those registers and classical values are
   combined into a single register $S$.)
 – Return $(S, M, h)$.
```

Here U_{step_i} refers to the unitary transformation $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus \text{step}_i(x)\rangle$. See the left dashed box in Figure 2 for a circuit-representation of A_i^*.

```
Algorithm $B^*(S, M)$ runs:
 – Let $S^*, M^*, h', i := S$.
 – If $h' \neq \bot$: apply $U_{\text{step}_i}$ to $M^*, M$.
 – Run $b \leftarrow B(S^*, M^*)$.
 – Return $b$.
```

See the left dashed box in Figure 2 for a circuit-representation of B^*.

Claim. (A_i^*, B^*) is valid.

We show this claim: After the measurement $(h', s) \leftarrow \mathcal{M}_{\text{partial}_i}(M^*)$, we have that M^* contains a superposition of $|\mathbf{m}\rangle$ with $\text{partial}_i(\mathbf{m}) = (h', s)$. If $h' = \bot$, then A_i^* initializes M with $|0^{\ell_{in}}\rangle$ and sets $h := H_k(0^{\ell_{in}})$. Thus in this case, M trivially contains a superposition of $|m\rangle$ with $H_k(m) = h$. If $h' \neq \bot$, then by Fact 3, M^* contains a superposition of $|\mathbf{m}\rangle$ with $H_k(\text{step}_i(\mathbf{m})) = h' = h$. Then

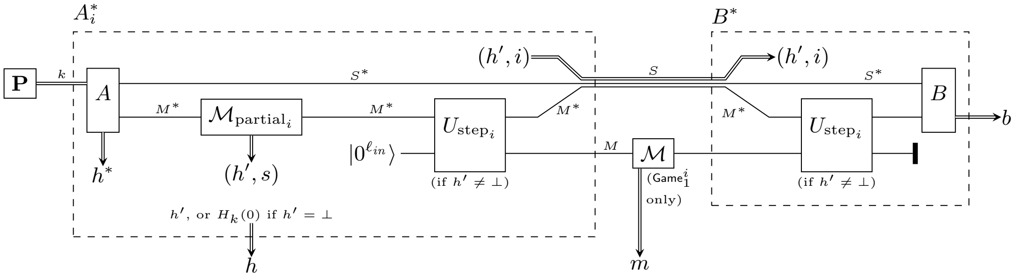


Fig. 2: The adversary (A_i^*, B^*) in games Game_1^i and Game_2^i. Depicted is Game_1^i. Game_2^i is derived by omitting the measurement $\mathcal{M}$ in the middle.

A^* initializes M with $|0^{\ell_{in}}\rangle$ and applies U_{step_i} to M^*, M. Thus after that, M is in a superposition of $|m\rangle$ with $H_k(m) = h_j$. Concluding, in both cases M is in a superposition of $|m\rangle$ with $H_k(m) = h$, thus (A_i^*, B^*) is valid and the claim follows.

Let Game_1^i denote Game_1 from Definition 7, but with adversary (A_i^*, B^*) and hash function $(\mathbf{P}, H_k)$. Analogously Game_2^i. Figure 2 depicts both games.

Claim. $\Pr[b = 1 : \mathsf{Game}_2^i] = \Pr[b = 1 : \mathsf{Game}_i^{hyb}]$.

We show this claim: In Game_2^i, no measurement occurs between the invocation of U_{step_i} by A_i^* and the invocation of U_{step_i} by B^*. (Cf. Figure 2.) Since U_{step_i} is an involution, those two invocations cancel out. Thus only the invocations of $\mathbf{P}$, A, $\mathcal{M}_{\text{partial}_i}$, and B remain. This is exactly Game_i^{hyb}. This shows the claim.

Claim. $\Pr[b = 1 : \mathsf{Game}_1^i] = \Pr[b = 1 : \mathsf{Game}_{i+1}^{hyb}]$.

We show the claim: Note that in Game_1^i, after the measurement $\mathcal{M}_{\text{partial}_i}$, on the registers M^*, M, we have the following sequence of operations if $h' \neq \bot$:

M is initialized with $|0^{\ell_{in}}\rangle$. U_{step_i} is applied to M^*, M. M is measured in the computational basis (outcome m). U_{step_i} is applied to M^*, M. M is discarded.

This is equivalent to just executing $m \leftarrow \mathcal{M}_{\text{step}_i}(M^*)$.

Furthermore, if $h = \bot$, then the sequence of operations is simply: Initialize M with $|0^{\ell_{in}}\rangle$. Measure M. Discard M. This is equivalent to doing nothing. And doing nothing is equivalent to $m \leftarrow \mathcal{M}_{\text{step}_i}(M^*)$ in case $h' = \bot$. (Because in that case, M^* is in a superposition of $|\mathbf{m}\rangle$ with $|\mathbf{m}| \leq i$, and thus $\text{step}_i(\mathbf{m}) = \bot$, and hence the outcome of $\mathcal{M}_{\text{step}_i}$ is deterministic.)

Thus Game_1^i is equivalent to the following Game_{1*}^i (in the sense that $\Pr[b = 1]$ is the same in both games):

$$\begin{aligned}\mathsf{Game}_{1*}^i : \quad & k \leftarrow \mathbf{P},\ (S^*, M^*, h^*) \leftarrow A(k), \\ & (h', s) \leftarrow \mathcal{M}_{\text{partial}_i}(M^*),\ m \leftarrow \mathcal{M}_{\text{step}_i}(M^*), \\ & b \leftarrow B(S^*, M^*).\end{aligned}$$

By Fact 4, measurements $\mathcal{M}_{\mathrm{partial}_i}(M^*)$ and $\mathcal{M}_{\mathrm{step}_i}(M^*)$ have the same effect on M^* as $\mathcal{M}_{\mathrm{partial}_{i+1}}(M^*)$. (The measurement outcome may be different, but we do not use the measurement outcome in our games.) Thus Game^i_{1*} is equivalent to Game^i_{1**} (in the sense that $\Pr[b=1]$ is the same in both games):

$$\begin{aligned}\mathsf{Game}^i_{1**}:\quad & k \leftarrow \mathbf{P},\ (S^*, M^*, h^*) \leftarrow A(k),\\ & (h', s) \leftarrow \mathcal{M}_{\mathrm{partial}_{i+1}}(M^*),\\ & b \leftarrow B(S^*, M^*).\end{aligned}$$

But Game^i_{1**} is the same as $\mathsf{Game}^{hyb}_{i+1}$, except that S, M, h are renamed to S^*, M^*, h^*. Hence $\Pr[b=1]$ is the same in Game^i_1 and $\mathsf{Game}^{hyb}_{i+1}$, the claim follows.

Let A^* pick $i \xleftarrow{\$} \{0, \dots, B-1\}$ and then run A^*_i. From Figure 21, it follows that (A^*, B^*) is valid, too. Let Game^*_1 denote Game_1 from Definition 7, but with adversary (A^*, B^*) and hash function $(\mathbf{P}, H_k)$. Analogously Game^*_2.

Since $(\mathbf{P}, H_k)$ is collapsing by assumption, and (A^*, B^*) is valid and polynomial-time, we have that $\varepsilon^* := \bigl|\Pr[b=1 : \mathsf{Game}^*_1] - \Pr[b=1 : \mathsf{Game}^*_2]\bigr|$ is negligible.

Then we have:

$$\begin{aligned}\varepsilon^* &= \bigl|\Pr[b=1 : \mathsf{Game}^*_1] - \Pr[b=1 : \mathsf{Game}^*_2]\bigr|\\ &= \frac{1}{B}\Bigl|\sum_{i=0}^{B-1}\Pr[b=1 : \mathsf{Game}^i_1] - \sum_{i=0}^{B-1}\Pr[b=1 : \mathsf{Game}^i_2]\Bigr|\\ &\overset{(*)}{=} \frac{1}{B}\Bigl|\sum_{i=0}^{B-1}\Pr[b=1 : \mathsf{Game}^{hyb}_{i+1}] - \sum_{i=0}^{B-1}\Pr[b=1 : \mathsf{Game}^{hyb}_i]\Bigr|\\ &= \frac{1}{B}\Bigl|\Pr[b=1 : \mathsf{Game}^{hyb}_B] - \Pr[b=1 : \mathsf{Game}^{hyb}_0]\Bigr| \overset{(4)}{=} \frac{\varepsilon}{B}.\end{aligned}$$

Here $(*)$ follows from Claims 21 and 21.

Since ε^* is negligible, $\varepsilon = B\varepsilon^*$ is negligible. □

Theorem 19 (Security of Merkle-Damgård). *Assume that pad is a polynomial-time computable Merkle-Damgård padding. If* $(\mathbf{P}, H_k)$ *is a polynomial-time computable collapsing hash function,* $(\mathbf{P}, \mathrm{MD}_k)$ *is collapsing.*

A concrete security statement is given in Theorem 20.

Proof. Since *pad* is a Merkle-Damgård padding, we have that *pad* is injective and $\operatorname{im} pad$ is a suffix code. Since the domain of *pad* is $\mathbf{M}$, and $|\mathbf{M}| \geq 2$ by assumption, $|\operatorname{im} pad| \geq 2$. Thus by Lemma 18, $(\mathbf{P}, \mathrm{IH}_k)$ is collapsing on $\operatorname{im} pad$.

Since *pad* is injective, $(\mathbf{P}, pad)$ is collapsing by Lemma 13.

Since $\mathrm{MD}_k = \mathrm{IH}_k \circ pad$, by Lemma 14, $(\mathbf{P}, \mathrm{MD}_k)$ is collapsing. □

Concluding, we also state Theorem 19 in its concrete security variant. Let τ_H denote an upper bound on the time needed for evaluating H_k. Let $\tau_{pad}(\ell)$ denote an upper bound on the time for computing $pad(\mathbf{m})$ for $|\mathbf{m}| \leq \ell$. Let $\ell_{pad}(\ell)$ denote an upper bound on $|pad(\mathbf{m})|$ for $|\mathbf{m}| \leq \ell$. ($|\cdot|$ refers to the length in bits.)

Theorem 20 (Concrete security of Merkle-Damgård). *Assume that pad is a Merkle-Damgård padding.*

Let (A, B) be a τ-time adversary, t-valid for MD_k on $\mathbf{M}$, with collapsing-advantage ε against $(\mathbf{P}, \mathrm{MD}_k)$.

Then there is a $(\tau + O(t\tau_{pad}(\ell_A) + t\ell_{pad}(\ell_A)\tau_H/\ell_{block}))$-time adversary (A^*, B^*), t-valid for H_k, with collapsing-advantage $\geq \varepsilon\ell_{block}/\ell_{pad}(\ell_A)$ against $(\mathbf{P}, H_k)$.

5 Collapsing hashes in the standard model

In the following, let (S_F, F_s) be am (ℓ_{in}, k)-lossy function with $F_s : \{0,1\}^{\ell_{in}} \to \{0,1\}^{\ell_{mid}}$. Let $h_r : \{0,1\}^{\ell_{mid}} \to \{0,1\}^{\ell_{out}}$ be a universal hash function (with key $r \in \{0,1\}^{\ell_{seed}}$). Let $\mathcal{D}_{inj}$ and $\mathcal{D}_{lossy}$ be as in Definition 2.

We will often write $F_{(r,s)}$ and $h_{(r,s)}$ for F_s and h_r to unify notation (one of the parameters will be silently ignored in this case).

Construction 1 (Collapsing compression function) *We define the parameter sampler $\mathbf{P}_{inj}$ to return (r, s) with $r \xleftarrow{\$} \{0,1\}^{\ell_{seed}}$, $s \leftarrow \mathcal{D}_{inj}$. We define the parameter sampler $\mathbf{P}_{lossy}$ to return (r, s) with $r \xleftarrow{\$} \{0,1\}^{\ell_{seed}}$, $s \leftarrow \mathcal{D}_{lossy}$. We define the parameter sampler $\mathbf{P}_H$ to return (r, s) with $r \xleftarrow{\$} \{0,1\}^{\ell_{seed}}$, $s \leftarrow S_F$.*

We define the hash function $H_{(r,s)} : \{0,1\}^{\ell_{in}} \to \{0,1\}^{\ell_{out}}$ by $H_{(r,s)} := h_{(r,s)} \circ F_{(r,s)}$.

Note that we are mainly interested in the case where $\ell_{out} < \ell_{in}$. Otherwise, $H_{(r,s)}$ could simply be chosen to be an injective function which is always collapsing (Lemma 13).

Furthermore, note that $\mathbf{P}_{inj}$ and $\mathbf{P}_{lossy}$ are not necessarily polynomial-time. The final construction will use $\mathbf{P}_H$, but we need $\mathbf{P}_{inj}$ and $\mathbf{P}_{lossy}$ to state intermediate results.

Lemma 21. *If (S_F, F_s) is a lossy function, then $(\mathbf{P}_{lossy}, F_{(r,s)})$ is collapsing.*

Proof. For $(r, s) \leftarrow \mathbf{P}_{inj}$, $F_{(r,s)}$ is always injective. Hence by Lemma 13, $(\mathbf{P}_{inj}, F_{(r,s)})$ is collapsing.

Since (S_F, F_s) is a lossy function, we have that $\mathcal{D}_{inj}$ and $\mathcal{D}_{lossy}$ are computationally indistinguishable. Hence $\mathbf{P}_{inj}$ and $\mathbf{P}_{lossy}$ are computationally indistinguishable.

Thus by Lemma 15, $(\mathbf{P}_{lossy}, F_{(r,s)})$ is collapsing. □

Lemma 22. *If (S_F, F_s) is a lossy function with lossiness rate K, and if $\ell_{out}/\ell_{in} \geq c > 2 - 2K$ for some constant c, $(\mathbf{P}_{lossy}, h_{(r,s)})$ is collapsing on $\operatorname{im} F_{(r,s)}$.*

Proof. We first compute the probability that $h_{(r,s)}$ is not injective on $\operatorname{im} F_{(r,s)}$.

$$\begin{aligned}
&\Pr[h_{(r,s)} \text{ is not injective on } \operatorname{im} F_{(r,s)} : (r,s) \leftarrow \mathbf{P}_{lossy}] \\
&\overset{(*)}{=} \sum_s \Pr[\mathcal{D}_{lossy} = s] \Pr[h_{(r,s)} \text{ is not injective on } \operatorname{im} F_{(r,s)} : r \overset{\$}{\leftarrow} \{0,1\}^{\ell_{seed}}] \\
&\leq \sum_s \Pr[\mathcal{D}_{lossy} = s] \sum_{\substack{x,y \in \operatorname{im} F_s \\ x \neq y}} \Pr[h_{(r,s)}(x) = h_{(r,s)}(y) : r \overset{\$}{\leftarrow} \{0,1\}^{\ell_{seed}}] \\
&\overset{(**)}{\leq} \sum_s \Pr[\mathcal{D}_{lossy} = s] \sum_{\substack{x,y \in \operatorname{im} F_s \\ x \neq y}} \frac{1}{2^{\ell_{out}}} \overset{(***)}{\leq} \sum_s \Pr[\mathcal{D}_{lossy} = s] \frac{(2^{\ell_{in}-k})^2}{2^{\ell_{out}}} \\
&= 2^{2\ell_{in}-2k-\ell_{out}} =: \varepsilon. \qquad (5)
\end{aligned}$$

Here $(*)$ uses the fact that $(r,s) \leftarrow \mathbf{P}_{lossy}$ is the same as $r \overset{\$}{\leftarrow} \{0,1\}^{\ell_{seed}}$, $s \leftarrow \mathcal{D}_{lossy}$. And $(**)$ is by definition of universal hash functions. And $(***)$ follows from the fact that for any s in the support of $\mathcal{D}_{lossy}$, $\operatorname{im} F_s = \operatorname{im} F_{(r,s)}$ has size at most $2^{\ell_{in}-k}$ (recall that k is the lossiness of F_s).

Since (S_F, F_s) has lossiness rate K, we have $k \geq K\ell_{in}$ by definition, and ℓ_{in} is superlogarithmic. Remember that $\ell_{out}/\ell_{in} \geq c$. Then

$$\varepsilon = 2^{2\ell_{in}-2k-\ell_{out}} \leq 2^{2\ell_{in}-2K\ell_{in}-c\ell_{in}} = 2^{(2-2K)\ell_{in}-c\ell_{in}} = 2^{-d\ell_{in}} \qquad \text{for } d := c - (2-2K).$$

Since by assumption, c and K are constants and $c > 2 - 2K$, we have that $d > 0$ is a constant. Since ℓ_{in} is superlogarithmic, this implies that $\varepsilon \leq 2^{-d\ell_{in}}$ is negligible.

From (5) and Lemma 13, we then have that $(\mathbf{P}_{lossy}, h_{(r,s)})$ is collapsing on $\operatorname{im} F_{(r,s)}$. □

Theorem 23. *If (S_F, F_s) is a polynomial-time computable lossy function with lossiness rate K, and if $\ell_{out}/\ell_{in} \geq c > 2 - 2K$ for some constant c, then $(\mathbf{P}_H, H_{(r,s)})$ is collapsing.*

Proof. By Lemma 21, $(\mathbf{P}_{lossy}, F_{(r,s)})$ is collapsing. By Lemma 22, $(\mathbf{P}_{lossy}, h_{(r,s)})$ is collapsing on $\operatorname{im} F_{(r,s)}$. By Construction 1, $H_{(r,s)} = h_{(r,s)} \circ F_{(r,s)}$. Thus, by Lemma 14, $(\mathbf{P}_{lossy}, H_{(r,s)})$ is collapsing.

Since (S_F, F_s) is a lossy function, $\mathcal{D}_{lossy}$ and S_F are computationally indistinguishable. Hence $\mathbf{P}_{lossy}$ and $\mathbf{P}_H$ are computationally indistinguishable. Hence by Lemma 15, $(\mathbf{P}_H, H_{(r,s)})$ is collapsing. □

Theorem 24. *Assume $\ell_{in} > \ell_{out}$. Let $\mathrm{MD}_{(r,s)}$ be the Merkle-Damgård construction applied to $H_{(r,s)}$ (using a Merkle-Damgård padding pad).*

If (S_F, F_s) is a polynomial-time computable lossy function with lossiness rate K, and h_r is polynomial-time computable, and if $\ell_{out}/\ell_{in} \geq c > 2 - 2K$ for some constant c, then $(\mathbf{P}_H, \mathrm{MD}_{(r,s)})$ is collapsing.

Proof. By Theorem 23, $(\mathbf{P}_H, H_{(r,s)})$ is collapsing. Then by Theorem 19, $(\mathbf{P}_H, \mathrm{MD}_{(r,s)})$ is collapsing. $\square$

Theorem 25. *Assume $\ell_{in} > \ell_{out}$. Let $\mathrm{MD}_{(r,s)}$ be the Merkle-Damgård construction applied to $H_{(r,s)}$. Let $(\mathsf{com}_{HMu}, \mathsf{verify}_{HMu})$ denote the unbounded Halevi-Micali commitment using $\mathrm{MD}_{(r,s)}$.*

If (S_F, F_s) is a polynomial-time computable lossy function with lossiness rate K, and h_r is polynomial-time computable, and if $\ell_{out}/\ell_{in} \geq c > 2 - 2K$ for some constant c, then $(\mathbf{P}_H, \mathsf{com}_{HMu}, \mathsf{verify}_{HMu})$ is statistically hiding and collapse-binding.

Proof. By Theorem 24, $(\mathbf{P}_H, \mathrm{MD}_{(r,s)})$ is collapsing. Then by Theorem 12, $(\mathbf{P}_H, \mathsf{com}_{HMu}, \mathsf{verify}_{HMu})$ is statistically hiding and collapse-binding. $\square$

Note that if $K > \frac{1}{2}$, we have $2 - 2K < 1$. Then h_r, c can always be chosen to satisfy the conditions of Theorem 24 and Theorem 25 (namely $\ell_{out}/\ell_{in} \geq c > 2 - 2K$ and $\ell_{in} > \ell_{out}$).

For completeness, we now give the concrete security variant of Theorem 25 here. Let τ_F denote the time needed for evaluating $F_{(r,s)}$. Let τ_h denote the time needed for evaluating $h_{(r,s)}$. Let τ'_h denotes an upper bound on the time needed for computing the universal hash function from Definition 10. For a given adversary (A, B), let ℓ_A be a upper bound on the length of each message output by A on the registers M_i (cf. Definition 8).

Theorem 26. *Assume $\ell_{in} > \ell_{out}$. Let $\mathrm{MD}_{(r,s)}$ be the Merkle-Damgård construction applied to $H_{(r,s)}$. Let $(\mathsf{com}_{HMu}, \mathsf{verify}_{HMu})$ denote the unbounded Halevi-Micali commitment using $\mathrm{MD}_{(r,s)}$.*

Then any adversary against $(\mathbf{P}_H, \mathsf{com}_{HMu}, \mathsf{verify}_{HMu})$ has hiding-advantage $\leq 2^{-\ell_{out}-1}$.

Let (A, B) be a τ-time adversary t-c.b.-valid for verify *with collapsing-advantage ε against $(\mathbf{P}_H, \mathsf{com}_{HMu}, \mathsf{verify}_{HMu})$.*

Then there are $(\tau + O(t\tau_{pad}(\ell_A) + t\ell_{pad}(\ell_A)(\tau_F + \tau_h)/(\ell_{in} - \ell_{out}) + \ell_{seed} + t\tau'_h))$-time adversaries $C_1, \dots, C_6$, such that C_1, C_2, C_3 distinguish S_F and $\mathcal{D}_{lossy}$ with some advantages $\varepsilon_1, \varepsilon_2, \varepsilon_3$, and C_4, C_5, C_6 distinguish $\mathcal{D}_{inj}$ and $\mathcal{D}_{lossy}$ with some advantages $\varepsilon_4, \varepsilon_5, \varepsilon_6$, and $\varepsilon \leq (2^{2\ell_{in}-2k-\ell_{out}} + 2\sum_{i=1}^{6} \varepsilon_i) \cdot \frac{\ell_{pad}(\ell_A)}{(\ell_{in}-\ell_{out})}$.

By using existing constructions of lossy functions, we further get:

Theorem 27. *If SIVP and GapSVP are hard for quantum algorithms to approximate within $\tilde{O}(d^c)$ factors for some $c > 5$, then there is a collapsing hash function with domain $\{0,1\}^*$ and codomain $\{0,1\}^{\ell_{out}}$ for some ℓ_{out}, as well as a non-interactive, statistically hiding, collapse-binding commitment schemes with message space $\{0,1\}^*$.*

Furthermore, the hash function and the commitment scheme can be chosen such that their parameter sampler $\mathbf{P}$ returns a uniformly random bitstring.

Proof. [13] shows that almost-always lossy trapdoor functions with lossiness rate $K < 1$ exist if SIVP and GapSVP are hard for quantum algorithms to approximate within $\tilde{O}(d^c)$ factors, where $c = 2 + \frac{3}{2(1-K)} + \delta$ for any desired $\delta > 0$. Almost-always lossy trapdoor functions are in particular lossy functions. If $c > 5$, we can chose some constant $K > \frac{1}{2}$ such that $c = 2 + \frac{3}{2(1-K)} + \delta$ for some $\delta > 0$. Thus there is a lossy function with constant lossiness rate $K > \frac{1}{2}$. Hence by Theorem 24 and Theorem 25 there are a collapsing hash function $(\mathbf{P}_H, H_{(r,s)})$ and a non-interactive collapse-binding statistically hiding commitment $(\mathbf{P}_H, \mathsf{com}_{HMu}, \mathsf{verify}_{HMu})$.

$\mathbf{P}_H$ returns (s, r) with $s \leftarrow S_F$ and $r \xleftarrow{\$} \{0,1\}^{\ell_{seed}}$. Furthermore, as discussed after Definition 2, the lossy function (S_F, F_s) can be chosen such that S_F returns uniformly random keys s. In that case $\mathbf{P}_H$ returns a uniformly random bitstring. □

Interactive commitments without public parameters. The above text analyzed non-interactive commitments using public parameters. We refer to the introduction for the reason why it is unlikely that we can get rid of the public parameters in the non-interactive setting. However, in the interactive setting, we get the following result:

Theorem 28. *If lossy function with lossiness rate $K > \frac{1}{2}$ exist, or if SIVP and GapSVP are hard for quantum algorithms to approximate within $\tilde{O}(d^c)$ factors for some $c > 5$, then there is a collapse-binding*[10] *statistically-hiding commitment scheme with two-round commit phase and non-interactive verification, without public parameters.*

Proof. Let $(\mathbf{P}_H, \mathsf{com}_{HMu}, \mathsf{verify}_{HMu})$ be the commitment scheme analyzed above.

We construct an interactive commitment scheme as follows: To commit to a message m, the recipient runs $k \leftarrow \mathbf{P}_H$ and sends k to the committer. Then the committer computes $(c, u) \leftarrow \mathsf{com}_{HMu}(k, m)$ and sends c. To open to m, the committer sends u, and the verifier checks whether $\mathsf{verify}_{HMu}(k, c, m, u) = 1$.

It is easy to see that if $(\mathbf{P}_H, \mathsf{com}_{HMu}, \mathsf{verify}_{HMu})$ is collapse-binding, so is the resulting interactive scheme. (In the collapse-binding game, the verifier is honest. Hence it is equivalent whether the verifier or $\mathbf{P}_H$ picks k.)

In general, having the verifier pick k may break the hiding property of the commitment. However, the proof of the hiding property of $(\mathbf{P}_H, \mathsf{com}_{HMu}, \mathsf{verify}_{HMu})$ (in the full version) reveals that that commitment is statistically hiding for any choice of k. Thus the interactive commitment is statistically hiding. □

6 Collapse-binding implies sum-binding

For the remainder of this section, let $(\mathbf{P}, \mathsf{com}, \mathsf{verify})$ be a commitment scheme with message space $\{0, 1\}$. (I.e., a bit commitment.)

[10] We refer to [16] for the definition of "collapse-binding" for interactive commitments.

A very simple and natural definition of the binding property for bit commitment schemes is the following one (it occurred implicitly and explicitly in different variants in [2,11,7,3,4]): If an adversary produces a commitment c, and is told only afterwards which bit m he should open it to, then $p_0 + p_1 \leq 1 + \textit{negligible}$. Here p_0 is the probability that he successfully opens the commitment to $m = 0$, and p_1 analogously. This definition is motivated by the fact that a perfectly binding commitment trivially satisfies $p_0 + p_1 \leq 1 + \textit{negligible}$.

Definition 29 (Sum-binding). *For any adversary (C_0, C_1) and $m \in \{0,1\}$, let*

$$p_m(C_0, C_1) := \Pr[\mathsf{verify}(k, c, m, u) = 1 : k \leftarrow \mathbf{P}, (S, c) \leftarrow C_0(k),\ u \leftarrow C_1(S, m)].$$

Here S is a quantum register, and c a classical value. We call $adv := p_0 + p_1 - 1$ the sum-binding-advantage *of (C_0, C_1). (With $adv := 0$ if the difference is negative.)*

A commitment is sum-binding *iff for any quantum-polynomial-time (C_0, C_1), adv is negligible.*

Unfortunately, this definition seems too weak to be useful (see [16] for more discussion), but certainly it seems that the sum-binding property is a minimal requirement for a bit commitment scheme. Yet, it was so far not known whether collapse-binding bit commitments are sum-binding. In this section, we will show that collapse-binding bit commitments are sum-binding, thus giving additional evidence that collapse-binding is a sensible definition.

Proof attempt using rewinding. Before we prove our result, we first explain why existing approaches (i.e., rewinding) do not give the required result.

First, the classical case as a warm up. Assume a classical adversary with $p_0 + p_1 = 1 + \varepsilon$ for non-negligible ε. We then break the classical computational-binding property as follows: Run the adversary to get c. Then ask him to provide an opening u for $m = 0$. Then rewind him to the state where he produced c. Then ask him to provide an opening u' for $m = 1$. The probability that u is valid is p_0, the probability that u' is valid is p_1. From the union bound, we get that the probability that both are valid is at least $p_0 + p_1 - 1 = \varepsilon$.[11] But that means that the adversary has non-negligible probability ε of finding c, m, m', u, u' with $m \neq m'$ and u, u' being valid openings for m, m'. This contradicts the classical-style binding property.

Now what happens if we try to use rewinding in the quantum case to show that collapse-binding implies sum-binding? If we use the rewinding technique from [14], the basic idea is the following:

Run the adversary to get a commitment c (i.e., $(S, c) \leftarrow C_0(k)$). Run the adversary to get an opening u for $m = 0$ (i.e., run $u \leftarrow C_1(S, 0)$). Here we assume w.l.o.g. that C_1 is unitary. Measure u. Run the inverse of the unitary $C_1(S, 0)$. Run the adversary to get an opening u' for $m = 1$ (i.e., run $u \leftarrow C_1(S, 1)$).

[11] Namely, $\Pr[u \text{ invalid}] = 1 - p_0$, $\Pr[u' \text{ invalid}] = 1 - p_1$. Hence $\Pr[u \text{ invalid or } u' \text{ invalid}] \leq (1 - p_0) + (1 - p_1)$. Thus $\Pr[u, u' \text{ valid}] \geq 1 - \big((1 - p_0) + (1 - p_1)\big) = p_0 + p_1 - 1$.

To get a contradiction, we need to show that with non-negligible probability u and u' are both valid openings. While u will be valid with probability p_0, there is nothing we can say about u'. This is because measuring u will disturb the state of the adversary so that $C_1(S,1)$ may return nonsensical outputs. [14] shows that *if there is only one valid* u, then rewinding works. But there is nothing that guarantees that there is only one valid u.[12] At this point the rewinding-based proof fails.

Collapse-binding implies sum-binding. We now formally state and prove the main result of this section with a technique different from rewinding. (But possibly this is a new rewinding technique under the hood.)

Theorem 30. *If* $(\mathbf{P}, \mathsf{com}, \mathsf{verify})$ *is collapse-binding, then* $(\mathbf{P}, \mathsf{com}, \mathsf{verify})$ *is sum-binding.*

An interesting open question is whether the converse holds. If so, this would immediate give strong results for the parallel composition of sum-binding commitments and their use in rewinding proofs (because all the properties of collapse-binding commitments would carry over).

We give a proof sketch first: As we have seen, running two executions of the adversary sequentially (first opening to $m = 0$, then opening to $m = 1$) via rewinding is problematic because the second execution may not be successful any more. Instead, we will run both executions at the same time in superposition:

Assume an adversary against sum-binding with non-negligible advantage ε. We initialize a qubit M with $|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. Then we let the adversary commit $((S, c) \leftarrow C_0(k))$, and then we run $C_1(S,0)$ or $C_1(S,1)$ in superposition, controlled by the register M. This may entangle M with the rest of the system. And we get openings for $m = 0$ and $m = 1$ in superposition on a register U. Now if we measure whether U contains a valid opening for the message on register M, the answer will be yes with probability $\delta := \frac{p_0+p_1}{2} = \frac{1+\varepsilon}{2}$ where p_0, p_1 are as in Definition 29 (call this measurement V_c). Now, we either measure the register M in the computational basis or we do not. And finally we apply the inverse of $C_1(S,0)$ or $C_1(S,1)$ in superposition. And finally we measure whether M is still in the state $|+\rangle$ (call this measurement $\mathcal{M}_+$).

We distinguish two cases: If we measure M in the computational basis, then $M = |0\rangle$ or $M = |1\rangle$ afterwards. So the measurement $\mathcal{M}_+$ succeeds with probability $\frac{1}{2}$. Hence the probability that both V_c and $\mathcal{M}_+$ succeed is $\frac{\delta}{2}$.

If we do not measure M in the computational basis, then we have the following situation. The invocation $C_1(S,0)$ or $C_1(S,1)$ in superposition, together with the measurement V_c, together with the uncomputation of $C_1(S,0)$ or $C_1(S,1)$ can be seen as a single binary measurement R_c. Now if we have a measurement that

[12] Collapse-binding commitments are rewinding-friendly, but this refers only to the case where we wish to measure the opened message m. Roughly, collapse-binding implies that measuring m does disturb the state more than measuring whether the commitment was opened correctly or not, and in that case, the rewinding technique from [14] applies. The [16] for example proofs using this technique.

succeeds with high probability, it cannot change the state much. Thus, the higher the success probability δ of R_c, the more likely it is that M is still in state $|+\rangle$ and $\mathcal{M}_+$ succeeds. An exact computation reveals: the probability that both R_c (a.k.a. V_c) and $\mathcal{M}_+$ succeed is δ^2.

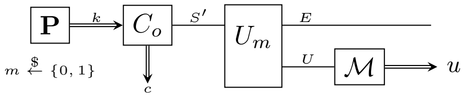
Thus the measurement $\mathcal{M}_+$ distinguishes between measuring and not measuring M with non-negligible probability $\frac{\delta}{2} - \delta^2 \geq \frac{\varepsilon}{4}$. This contradicts the collapse-binding property, the theorem follows.

We now give the full proof:

Proof of Theorem 30. Let (C_0, C_1) be an adversary in the sense of Definition 29 (against sum-binding). Let $p_0 := p_0(C_0, C_1)$ and $p_1 := p_1(C_0, C_1)$. We have to show that the advantage $\varepsilon := p_0 + p_1 - 1$ is upper bounded by a negligible function.

Without loss of generality, we can assume that C_1 is unitary. More precisely, $C_1(S, m)$ applies a unitary circuit U_m to S, resulting in two output registers U and E. Then he measures U in the computational basis and returns the outcomes u.

With that notation, we can express the game from Definition 29 as the following circuit (renaming the register S to S' to avoid name clashes later):



(6)

(Here and in the following, $\mathcal{M}$ denotes a measurement in the computational basis.) In that circuit, $\Pr[\mathsf{verify}(k, c, m, u) = 1] = \delta := \frac{1}{2}(1 + \varepsilon)$.

Let M denote a one-qubit quantum register, and define $U_M : |m\rangle_M \otimes |\Psi\rangle_{S'} \mapsto |m\rangle_M \otimes U_m|\Psi\rangle_{S'}$. That is, U_M is a unitary with two input registers M, S', and three output registers M, U, E which is realized by applying U_0 or U_1 to S', depending on whether M is $|0\rangle$ or $|1\rangle$.

Let $\mathcal{M}_+$ be the binary measurement that checks whether register M is in state $|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. Formally, $\mathcal{M}_+$ is defined by the projector $P_+ := |+\rangle\langle+|$ on M.

Recall that V_c from Lemma 6 is the measurement defined by the projector $P_c := \sum_{\substack{m,u \\ \mathsf{verify}(k,c,m,u)=1}} |m\rangle\langle m| \otimes |u\rangle\langle u|$.

We define an adversary (A, B) against the collapse-binding property of com (using the alternative definition from Lemma 6). Algorithm $A(k)$ performs the following steps (see also Figure 3):

- Run $(S', c) \leftarrow C_0(k)$.
- Initialize a register M with $|+\rangle$.
- $(M, U, E) \leftarrow U_M(M, S')$. That is, apply U_M to M, S'.
- $S := E$. (That is, we rename the register E.)
- Return (S, M, U, c).

Algorithm $B(S, M, U)$ performs the following steps (see also Figure 3):

- $E := S$.

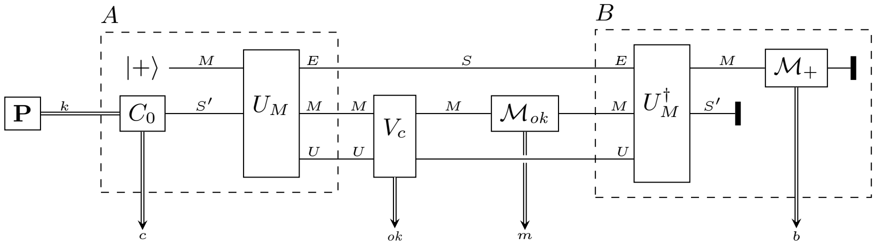
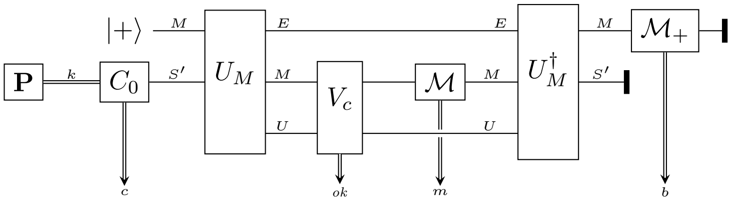


Fig. 3: Circuit describing Game_1. Game_2 can be derived by omitting $\mathcal{M}_{ok}$. The adversary algorithms A and B are depicted in the dashed boxes. (To avoid wires crossing gates, the outgoing wires of U_M are ordered E, M, U, not M, U, E as in the text.)

- $(M, S') \leftarrow U_M^\dagger(M, U, E)$.
- $b \leftarrow \mathcal{M}_+(Y)$.
- Return b.

Let $\mathsf{Game}_1, \mathsf{Game}_2$ refer to the games from Lemma 6 with adversary (A, B). Figure 3 depicts those games as a quantum circuit.

We consider Game_1 first. We are interested in computing the probability $p := \Pr[b = 1 \wedge ok = 1]$ in this game. Observe that replacing $\mathcal{M}_{ok}$ by $\mathcal{M}$ (the latter being the measurement in the computational basis, applied even when $ok = 0$) does not change p. (Because $\mathcal{M}_{ok}$ and $\mathcal{M}$ behave differently only when $ok = 0$.) Thus, replacing $\mathcal{M}_{ok}$ on M by $\mathcal{M}$ does not change p. Thus, we get the following circuit:



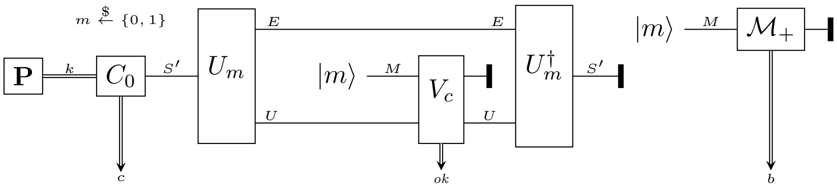
(7)

and have

$$\Pr[b = 1 \wedge ok = 1 : \text{Circuit (7)}] = \Pr[b = 1 \wedge ok = 1 : \mathsf{Game}_1]. \tag{8}$$

Note that $\mathcal{M}$ on M commutes with V_c and U_M. So we can move $\mathcal{M}$ to the beginning (right after initializing M with $|+\rangle$). But measuring $|+\rangle$ in the computational basis yields a uniformly distributed bit m. And furthermore, if M contains $|m\rangle$, then U_M degenerates to U_m on register S', and M stays in state

$|m\rangle$ until the measurement $\mathcal{M}_+$. Thus we can simplify (7) as follows:



(9)

We thus have

$$\Pr[b = 1 \wedge ok = 1 : \text{Circuit (7)}] = \Pr[b = 1 \wedge ok = 1 : \text{Circuit (9)}]. \qquad (10)$$

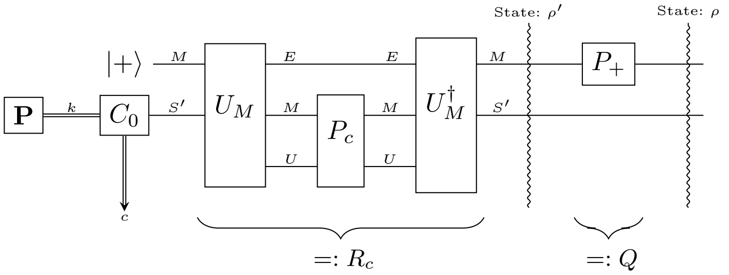
It is easy to see that

$$\Pr[ok = 1 : \text{Circuit (9)}] = \Pr[\mathsf{verify}(k, c, m, u) = 1 : \text{Circuit (6)}] = \delta.$$

Furthermore, in (9), b is independent of ok, and we have $\Pr[b = 1] = \frac{1}{2}$ by definition of $\mathcal{M}_+$. Thus

$$\Pr[b = 1 \wedge ok = 1 : \mathsf{Game}_1] \overset{(8),(10)}{=} \Pr[b = 1 \wedge ok = 1 : \text{Circuit (9)}] = \frac{\delta}{2}. \qquad (11)$$

We now consider Game_2. This game is depicted in Figure 3 (when omitting the measurement $\mathcal{M}_{ok}$). We are interested in computing the probability $q := \Pr[b = 1 \wedge ok = 1]$ in this game. Recall that P_+, P_c are the projectors describing the measurements $\mathcal{M}_+$, V_c. Thus, $q = \operatorname{tr} \rho$ where ρ is the final state of the following circuit:



(12)

We abbreviate the product of the operators U_M, P_c, $U_M^\dagger$ with R_c. Note that R_c is a projector since P_c is a projector and U_M is unitary. Let $Q := P_+ \otimes id_{S'}$. Furthermore, let ρ_c be the state output by C_0 on S' conditioned on classical output c (and let p_c be the probability of that output). We can write ρ_c as $\rho_c = \sum_i p_{ci} |\Psi_{ci}\rangle\langle\Psi_{ci}|$ for some normalized quantum states $|\Psi_{ci}\rangle$ and some probabilities p_{ci} with $\sum_i p_{ci} = 1$. Let $|\Psi'_{ci}\rangle := |+\rangle \otimes |\Psi_{ci}\rangle$. Let $|\Phi_{ci}\rangle := QR_c|\Psi'_{ci}\rangle$. With that notation, we have $\rho = \sum_{c,i} p_c p_{ci} |\Phi_{ci}\rangle\langle\Phi_{ci}|$ and $\sum_{c,i} p_c p_{ci} = 1$. Hence $q = \operatorname{tr} \rho = \sum_{c,i} p_c p_{ci} \big\| |\Phi_{ci}\rangle \big\|^2$.

Furthermore, if ρ' is the state in circuit (12) after $U_M^\dagger$, then it is easy to see that $\operatorname{tr}\rho' = \delta$ (recall that δ is the success probability in (6)). We then have that $\delta = \operatorname{tr}\rho' = \sum_{c,i} p_c p_{ci} \big\| R_c|\Psi'_{ci}\rangle \big\|^2 = \sum_{c,i} p_c p_{ci} \delta_{cf}$ with $\delta_{cf} := \big\| R_c|\Psi'_{ci}\rangle \big\|^2$.

By definition of Q and $|\Psi'_{ci}\rangle$, we have that $Q|\Psi'_{ci}\rangle = |\Psi'_{ci}\rangle$. Then

$$\delta_{ci} = \langle\Psi'_{ci}|R_c|\Psi'_{ci}\rangle = \langle\Psi'_{ci}|QR_c|\Psi'_{ci}\rangle \leq \big\|QR_c|\Psi'_{ci}\rangle\big\| = \big\||\Phi_{ci}\rangle\big\|.$$

Thus

$$q = \sum_{c,i} p_c p_{ci} \big\||\Phi_{ci}\rangle\big\|^2 \geq \sum_{c,i} p_c p_{ci} \delta_{ci}^2 \overset{(*)}{\geq} \Big(\sum_{c,i} p_c p_{ci} \delta_{ci}\Big)^2 = \delta^2$$

Here $(*)$ uses Jensen's inequality and the fact that $\sum_{c,i} p_c p_{ci} = 1$.

Thus

$$\Pr[b = 1 \wedge ok = 1 : \mathsf{Game}_2] = q \geq \delta^2. \tag{13}$$

Since Game_1 and Game_2 are identical unless $ok = 1$, we have that

$$\Pr[b = 1 \wedge ok \neq 1 : \mathsf{Game}_1] = \Pr[b = 1 \wedge ok \neq 1 : \mathsf{Game}_2]. \tag{14}$$

Thus

$$\begin{aligned}
&\Pr[b = 1 : \mathsf{Game}_2] - \Pr[b = 1 : \mathsf{Game}_1] \\
&= \big(\Pr[b = 1 \wedge ok = 1 : \mathsf{Game}_2] + \Pr[b = 1 \wedge ok \neq 1 : \mathsf{Game}_2]\big) \\
&\quad - \big(\Pr[b = 1 \wedge ok = 1 : \mathsf{Game}_1] + \Pr[b = 1 \wedge ok \neq 1 : \mathsf{Game}_1]\big) \\
&\overset{(14)}{=} \Pr[b = 1 \wedge ok = 1 : \mathsf{Game}_2] - \Pr[b = 1 \wedge ok \neq 1 : \mathsf{Game}_1] \\
&\overset{(13),(11)}{\geq} \delta^2 - \frac{\delta}{2} \geq \frac{\varepsilon}{4}.
\end{aligned}$$

Thus

$$\Big|\Pr[b = 1 : \mathsf{Game}_1] - \Pr[b = 1 : \mathsf{Game}_2]\Big| \geq \frac{\varepsilon}{4}. \tag{15}$$

Since (C_0, C_1) is polynomial-time adversary, (A, B) is polynomial-time. By assumption, $(\mathbf{P}, \mathsf{com}, \mathsf{verify})$ is collapse binding. Thus by Lemma 6, the rhs of (15) is negligible. Hence ε is negligible. Since ε was the advantage of the adversary (C_0, C_1) against the sum-binding property, it follows that $(\mathbf{P}, \mathsf{com}, \mathsf{verify})$ is sum-binding. □

6.1 CDMS-binding

For the remainder of this section, let $(\mathbf{P}, \mathsf{com}, \mathsf{verify})$ be a commitment scheme with message space $\{0,1\}^\ell$.

The sum-binding definition is restricted to bit commitments. In [3], a generalization of sum-binding definition is given. Intuitively, for any function f, if the adversary produces a commitment c, then there should be at most one value y such that the adversary can open c to a message m with $f(m) = y$. Slightly more formally, we require that $\sum_y \tilde{p}_y \leq 1 + \mathit{negligible}$ where $\tilde{p}_y$ is the probability that the adversary (who gets y after producing the commitment c) manages to open c to a message m with $f(m) = y$. Again, this definition is motivated by the fact that perfectly binding commitments satisfy $\sum_y \tilde{p}_y \leq 1$. The definition can be parametrized by specifying the set F of allowed functions f.

Definition 31 (CDMS-binding, following [3]). *Let F be a family of functions $\{0,1\}^\ell \to \{0,1\}^\Lambda$.*

For any adversary (C_0, C_1) and any $y \in \{0,1\}^\Lambda$, let

$$\begin{aligned}\tilde p_y(C_0,C_1) := \Pr[&\mathsf{verify}(k,c,m,u)=1 \land f(m)=y : \\ &k \leftarrow \mathbf{P}, (S,c,f) \leftarrow C_0(k), (m,u) \leftarrow C_1(S,y)].\end{aligned}$$

Here S is a quantum register, and c a classical value, and f a function in F (represented as a Boolean circuit).

We call (C_0, C_1) F-CDMS-valid if it only outputs functions $f \in F$.

We call $adv := \sum_{y\in\{0,1\}^\Lambda} \tilde p_y(C_0,C_1) - 1$ the F-CDMS-advantage *of (C_0, C_1). (With $adv := 0$ if the difference is negative.)*

We call a commitment scheme F-CDMS-binding *iff for all quantum-polynomial-time F-CDMS-valid (C_0, C_1), the F-CDMS-advantage of (C_0, C_1) is negligible.*

We have somewhat modified the definition with respect to [3]: Namely, instead of quantifying over all $f \in F$, we let the adversary choose f. This gives the adversary additional power, because f may depend on the public parameter k, but at the same time it also removes some power (because f needs to be efficiently computed in our definition). For non-uniform adversaries, our definition implies the one from [3].

Note that the sum-binding definition is a special case of the CDMS-binding definition: A bit commitment is sum-binding iff it is F-binding where F contains only the identity.

The following theorem is shown using a similar technique as Theorem 30. The main difference is that we have to use a superposition of all possible values y, instead of the superposition $|+\rangle$ of messages 0 and 1. Furthermore, the fact that the adversary has free choice of m, subject to the condition $f(m) = c$ introduces additional technicalities, but these are solved in the full proof.

Theorem 32. *If $(\mathbf{P}, \mathsf{com}, \mathsf{verify})$ is collapse-binding, then $(\mathbf{P}, \mathsf{com}, \mathsf{verify})$ is F-CDMS-binding for any $F \subseteq \{0,1\}^\ell \to \{0,1\}^\Lambda$ with logarithmically-bounded Λ.*

Note the condition that Λ is logarithmically-bounded. This condition is necessary as the following example shows: Let com be a perfectly binding commitment, except that with probability ε the adversary finds a secret that allows him to open the commitment to any message. This small probability ε does not change the fact that the commitment is collapse-binding (and arguably any reasonable definition of computationally binding should tolerate such a negligible error). However, an adversary that commits to 0, then gets $y \in \{0,1\}^\Lambda$, and then tries to open to an arbitrary m with $f(m) = y$ will succeed with probability $\tilde p_y = \varepsilon$ for all $y \neq f(0)$, and with probability $\tilde p_y = 1$ for $y = f(0)$. Hence $\sum_y \tilde p_y = 1 + (2^\Lambda - 1)\varepsilon$. If Λ is superlogarithmic, then $(2^\Lambda - 1)\varepsilon$ will not necessarily be negligible. This example shows that collapse-binding cannot imply CDMS-binding for superlogarithmic Λ and also indicates that probably CDMS-binding with superlogarithmic Λ is not a reasonable definition of computationally binding. (Note: in [3], only

CDMS-binding with logarithmically-bounded Λ was used and is sufficient for their OT protocol.)

Acknowledgements. We thank Dennis Hofheinz for discussions that led to the commitment protocol from Construction 1. This work was supported by institutional research funding IUT2-1 of the Estonian Ministry of Education and Research and by the Estonian ICT program 2011-2015 (3.2.1201.13-0022).

References

1. Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems (the hardness of quantum rewinding). In *FOCS 2014*, pages 474–483. IEEE, 2014. Preprint on IACR ePrint 2014/296.
2. G. Brassard, C. Crépeau, R. Jozsa, and D. Langlois. A quantum bit commitment scheme provably unbreakable by both parties. In *FOCS '93*, pages 362–371, Los Alamitos, CA, USA, 1993. IEEE.
3. Claude Crépeau, Paul Dumais, Dominic Mayers, and Louis Salvail. Computational collapse of quantum state with application to oblivious transfer. In *TCC 2004*, volume 2951 of *LNCS*, pages 374–393. Springer, 2004.
4. Claude Crépeau, Louis Salvail, Jean-Raymond Simard, and Alain Tapp. Two provers in isolation. In Dong Hong Lee and Xiaoyun Wang, editors, *Asiacrypt 2011*, volume 7072 of *LNCS*, pages 407–430. Springer, 2011.
5. Ivan Damgård, Serge Fehr, and Louis Salvail. Zero-knowledge proofs and string commitments withstanding quantum attacks. In Matt Franklin, editor, *Crypto 2004*, volume 3152 of *LNCS*, pages 254–272. Springer, 2004.
6. Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. *J Cryptology*, 10(3):163–194, 1997.
7. Paul Dumais, Dominic Mayers, and Louis Salvail. Perfectly concealing quantum bit commitment from any quantum one-way permutation. In *Eurocrypt 2000*, volume 1807 of *LNCS*, pages 300–315. Springer, 2000.
8. Shafi Goldwasser and Mihir Bellare. Lecture notes on cryptography. `http://cseweb.ucsd.edu/~mihir/papers/gb.html`, 2008. Summer course on cryptography, MIT, 1996-2008.
9. Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In Neal Koblitz, editor, *Crypto '96*, volume 1109 of *LNCS*, pages 201–215. Springer, 1996.
10. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J Computing*, 28(4):1364–1396, 1999.
11. Dominic Mayers. Unconditionally Secure Quantum Bit Commitment is Impossible. *Physical Review Letters*, 78(17):3414–3417, 1997. Online available at `http://arxiv.org/abs/quant-ph/9605044`.
12. National Institute of Standards and Technology (NIST). Secure hash standard (SHS). FIPS PUBS 180-4, 2015. doi:10.6028/NIST.FIPS.180-4.
13. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *STOC*, pages 187–196, New York, NY, USA, 2008. ACM. Full version at `http://ia.cr/2007/279`.
14. Dominique Unruh. Quantum proofs of knowledge. In *Eurocrypt 2012*, volume 7237 of *LNCS*, pages 135–152. Springer, April 2012. Full version is IACR ePrint 2010/212.

15. Dominique Unruh. Collapse-binding quantum commitments without random oracles. IACR ePrint 2016/508, 2016. Full version of this paper.
16. Dominique Unruh. Computationally binding quantum commitments. In *Eurocrypt 2016*, volume 9666 of *LNCS*, pages 497–527. Springer, 2016. Full version is IACR ePrint 2015/361.
17. John Watrous. Zero-knowledge against quantum attacks. *SIAM J. Comput.*, 39(1):25–58, 2009. Online available at `https://cs.uwaterloo.ca/~watrous/Papers/ZeroKnowledgeAgainstQuantum.pdf`.
18. Jun Yan, Jian Weng, Dongdai Lin, and Yujuan Quan. Quantum bit commitment with application in quantum zero-knowledge proof (extended abstract). In *ISAAC 2015*, volume 9472 of *LNCS*, pages 555–565. Springer, 2015.
19. Mark Zhandry. How to construct quantum random functions. In *FOCS 2013*, pages 679–687, Los Alamitos, CA, USA, 2012. IEEE Computer Society. Online version is IACR ePrint 2012/182.