Quantum-secure message authentication via blind-unforgeability

Gorjan Alagic[1], Christian Majenz[2], Alexander Russell[3], and Fang Song[4]

[1]QuICS, University of Maryland, and NIST, Gaithersburg, Maryland
[2]QuSoft and Institute for Logic, Language and Computation, University of Amsterdam
[3]Department of Computer Science and Engineering, University of Connecticut
[4]Computer Science Department, Texas A&M University

November 25, 2018

Abstract

Formulating and designing unforgeable authentication of classical messages in the presence of quantum adversaries has been a challenge, as the familiar classical notions of unforgeability do not directly translate into meaningful notions in the quantum setting. A particular difficulty is how to fairly capture the notion of "predicting an unqueried value" when the adversary can query in quantum superposition. In this work, we uncover serious shortcomings in existing approaches, and propose a new definition. We then support its viability by a number of constructions and characterizations.

Specifically, we demonstrate a function which is secure according to the existing definition by Boneh and Zhandry, but is clearly vulnerable to a quantum forgery attack, whereby a query supported only on inputs that start with 0 divulges the value of the function on an input that starts with 1. We then propose a new definition, which we call "blind-unforgeability" (or BU.) This notion matches "intuitive unpredictability" in all examples studied thus far. It defines a function to be predictable if there exists an adversary which can use "partially blinded" oracle access to predict values in the blinded region. Our definition (BU) coincides with standard unpredictability (EUF-CMA) in the classical-query setting. We show that quantum-secure pseudorandom functions are BU-secure MACs. In addition, we show that BU satisfies a composition property (Hash-and-MAC) using "Bernoulli-preserving" hash functions, a new notion which may be of independent interest. Finally, we show that BU is amenable to security reductions by giving a precise bound on the extent to which quantum algorithms can deviate from their usual behavior due to the blinding in the BU security experiment.

1 Introduction

1.1 Background.

Large-scale quantum computers will break all widely-deployed public-key cryptography, and may even threaten certain post-quantum candidates [19, 7, 8, 9, 4]. Basic symmetric-key constructions like Feistel ciphers and CBC-MACs also become vulnerable in a quantum attack model [14, 15, 13, 18], where the adversary is presumed to have quantum query access to some part of the cryptosystem. For example, the adversary may gain access to the unitary operator $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f_k(x)\rangle$ where f_k is the encryption or decryption function with the key k. While it is unclear if this model is directly relevant to physical implementations of symmetric-key cryptography, it appears necessary in a number of generic settings, such as public-key encryption and hashing with public hash functions. It could also be relevant when private-key primitives are composed in larger protocols, e.g., by exposing circuits via obfuscation [17]. Setting down appropriate security definitions in this quantum attack model is the subject of several threads of recent research [6, 10].

In this article, we study authentication of classical information in this quantum-secure model. Here, the adversary is granted quantum query access to the signing algorithm of a message authentication code (MAC) or a digital signature scheme, and is tasked with producing valid forgeries. In the purely classical setting, we insist that the forgeries are fresh, i.e., distinct from previous queries to the oracle, so that the security definition does not become vacuous. When the function may be queried in superposition, however, it's unclear how to meaningfully reflect this constraint that a forgery was "unqueried" without ruling out natural, intuitive attacks. For example, consider a uniform superposition query. Simply measuring the output state to get a forgery—a feasible attack against any function—should not be considered a break. On the other hand, an adversary who uses the same query to discover some structural property (e.g., a superpolynomial-size period in the MAC) should be considered successful. Examples like these indicate the difficulty of the problem. How do we correctly "price" the queries? How do we decide if a forgery is fresh? Furthermore, how do we do this in a manner that is consistent with these examples, and many others? This problem has a natural interpretation that goes well beyond cryptography: *What does it mean for a classical function to appear unpredictable to a quantum oracle algorithm?*[1]

Previous approaches. The first approach to this problem was suggested by Boneh and Zhandry [5]. They define a MAC to be unforgeable, if no adversary can use q queries to the MAC to produce $q+1$ valid input-output pairs except with negligible probability. We will refer to this notion as "BZ security" (and k-BZ for the case where the adversary is permitted a maximum of k queries). Boneh and Zhandry prove a number of results about this notion, including that it can be realized by a quantum-secure pseudorandom function (qPRF).

In an approach by Garg, Yuen and Zhandry [11], a MAC is considered *one-time* secure if only a trivial "query, measure in computational basis, output result" attack is allowed; we call this notion GYZ. Unfortunately, it is not clear how to extend GYZ beyond the single-query case. Zhandry recently gave a separation example between BZ and GYZ by means of indistinguishability obfuscation [25].

It is interesting to note that similar problems are present in defining unforgeability for *authentication of quantum data.* A convincing solution was recently found [2]. This approach relies on the fact that, for quantum messages, *authentication implies secrecy*; this enables "tricking" the adversary by replacing their queries with "trap" plaintexts to detect replays. As a result, the approach of [2] is inapplicable to the setting of classical messages, where unforgeability and secrecy are orthogonal. Indeed, in situations where unforgeability is required but secrecy is not, adversaries would easily recognize spoofed oracles.

Unresolved issues. BZ security, the only candidate definition of quantum-secure unforgeability in the setting of more than one query, appears to be insufficient for several reasons. First, as observed in [11], it is a-priori unclear if BZ security rules out adversaries who forge a message in region A after querying the signing oracle on a disjoint message region B. Second, BZ may not capture the unique features of quantum information, such as the destructiveness of measurement. Quantum algorithms must sometimes "consume" (i.e., fully measure) a state to extract some useful information, such as a symmetry in the oracle. It's plausible that, for some MACs, there is an adversary who makes one or more quantum queries but then must consume the post-query states completely in order to make a single convincing forgery.

Despite these philosophical criticisms, prior to this work no BZ-secure schemes have been shown to be manifestly insecure. It is thus essential to gain a concrete understanding of these potential issues, and thereby place the security of MACs and other primitives against quantum attacks on firmer foundations.

1.2 Summary of results

1.2.1 The problem with BZ.

Our first result is a construction of a MAC which is forgeable (in a strong intuitive sense) and yet is classified by BZ as secure.

[1] The related notion of "appearing random" has a satisfying definition, which can be fulfilled efficiently [24].

Construction 1. *Given a triple $k = (p, f, g)$ where $p \in \{0,1\}^n$ and $f, g : \{0,1\}^n \to \{0,1\}^n$, define $M_k : \{0,1\}^{n+1} \to \{0,1\}^{2n}$ by*

$$M_k(x) = \begin{cases} 0^{2n} & x = 0\|p \\ 0^n \| f(x') & x = 0\|x',\ x' \neq p \\ g(x' \bmod p)\|f(x') & x = 1\|x'\,. \end{cases}$$

Consider an adversary that queries only on messages starting with 1, as follows:

$$\sum_{x,y} |1,x\rangle_X |0^n\rangle_{Y_1} |y\rangle_{Y_2} \longmapsto \sum_{x,y} |1,x\rangle_X |g_p(x)\rangle_{Y_1} |y \oplus f(x)\rangle_{Y_2}\,. \tag{1}$$

Since $\sum_y |y \oplus f(x)\rangle_{Y_2} = \sum_y |y\rangle_{Y_2}$, discarding the first qubit and Y_2 yields $\sum_x |x\rangle|g_p(x)\rangle$. One can then recover p via period-finding and output $(0\|p, 0^{2n})$. We emphasize that the forgery was queried with *zero* amplitude. One can interpret this attack as, e.g., querying only on messages starting with "From: Alice" and then forging a message starting with "From: Bob". Despite this, we can show that M is BZ-secure.

Theorem 1. *The family M_k (for uniformly random $k = (p, f, g)$) is BZ-secure.*

The BZ security of M relies on a dilemma the adversary faces at each query: either learn an output of f, or obtain a superposition of $(x, g(x))$-pairs for Fourier sampling. Our proof shows that, once the adversary commits to one of these two choices, the other option is irrevocably lost. Our result can thus be understood as a refinement of an observation of Aaronson: quantumly learning a property sometimes requires *uncomputing* some information [1]. Note that, while Aaronson could rely on standard (asymptotic) query complexity techniques, our problem is quite fragile: BZ security describes a task which should be hard with q queries, but is completely trivial given $q + 1$ queries. Our proof is inspired by a new quantum random oracle technique of Zhandry [26].

1.2.2 A new definition: Blind-unforgeability.

We then develop a new definition of unpredictability. Given the context of quantum-secure MACs and digital signatures, we call our notion "blind-unforgeability" (or BU). In this approach, we examine the behavior of adversaries in the following experiment. The adversary is granted quantum oracle access to the MAC, "blinded" at a random region B. Specifically, we set B to be a random ϵ-fraction of the message space, and declare that the oracle function will output $\perp$ on all of B.

$$B_\epsilon \mathsf{Mac}_k(x) := \begin{cases} \perp & \text{if } x \in B_\epsilon, \\ \mathsf{Mac}_k(x) & \text{otherwise.} \end{cases}$$

Given a MAC $(\mathsf{Mac}, \mathsf{Ver})$, an adversary $\mathcal{A}$, and adversary-selected parameter ϵ, the "blind forgery experiment" is:

1. Generate key k and random blinding B_ϵ;
2. Produce candidate forgery $(m, t) \leftarrow \mathcal{A}^{B_\epsilon \mathsf{Mac}_k}(1^n)$.
3. Output win if $\mathsf{Ver}_k(m, t) = \mathsf{acc}$ and $m \in B_\epsilon$; otherwise output rej.

Definition 1. *A MAC is blind-unforgeable* (BU) *if for every adversary $(\mathcal{A}, \epsilon)$, the probability of winning the blind forgery experiment is negligible.*

In this work, BU will typically refer to the case where $\mathcal{A}$ is an efficient quantum algorithm (QPT) and the oracle is quantum, i.e., $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus B_\epsilon \mathsf{Mac}_k(x)\rangle$. We will also consider q-BU, the information-theoretic variant where the total number of queries is a-priori fixed to q. We remark that the above definition is also easy to adapt to other settings, e.g., classical security against PPT adversaries, quantum or classical security for digital signatures, etc.

1.2.3 Results about blind-unforgeability.

Next, we collect a series of results which build up confidence in BU as a viable definition of unforgeability. These results allow us to conclude that BU classifies a wide range of examples (in fact, all examples we have examined) as either forgeable or unforgeable in a way that agrees with our intuition about the meaning of unpredictability. First, we show that BU correctly classifies unforgeability in the classical-query setting.

Proposition 1. *In the setting of classical queries,* BU $\Leftrightarrow$ EUF-CMA.

Next, we give a general simulation theorem which tightly controls the deviation in the adversary's behavior when subjected to the BU experiment.

Theorem 2. *Let $\mathcal{A}$ be a quantum query algorithm making at most T queries. Let $F : X \to Y$ be a function, B_ϵ a random ϵ-blinding subset of X, and P any function with support B_ϵ. Then*

$$\mathop{\mathbb{E}}_{B_\epsilon} \left\| \mathcal{A}^F(1^n) - \mathcal{A}^{F \oplus P}(1^n) \right\|_1 \leq 2T\sqrt{\epsilon}\,.$$

The above fact can be viewed as evidence that adversaries that produce "good forgeries" (in any reasonably intuitive sense) will not be disturbed too much by blinding, and will thus in fact also win the BU experiment. We can formulate and prove this intuition explicitly for a wide class of adversaries, as follows. Given an oracle algorithm $\mathcal{A}$, we let $\mathsf{supp}(\mathcal{A})$ denote the union of the supports of all the queries of $\mathcal{A}$, taken over all choices of oracle function.

Theorem 3 (informal)**.** *Let $\mathcal{A}$ be QPT and $\mathsf{supp}(\mathcal{A}) \cap R = \emptyset$ for some $R \neq \emptyset$. Let* Mac *be a MAC, and suppose $\mathcal{A}^{\mathsf{Mac}_k}(1^n)$ outputs a valid pair $(m, \mathsf{Mac}_k(m))$ with $m \in R$ with noticeable probability. Then* Mac *is not* BU *secure.*

A straightforward application of Theorem 3 shows that Construction 1 is BU-insecure. In particular, we have the following.

Corollary 1. *There exists a* BZ*-secure MAC which is* BU*-insecure.*

1.2.4 Blind-unforgeable MACs.

Next, we show that several natural constructions satisfy BU. We first show that a random function is blind-unforgeable.

Theorem 4. *Let $R : X \to Y$ be a random function such that $1/|Y|$ is negligible. Then R is a blind-unforgeable MAC.*

By means of results of Zhandry [24] and Boneh and Zhandry [5], this leads to efficient BU-secure constructions.

Corollary 2. *Quantum-secure pseudorandom functions* (qPRF) *are* BU*-secure MACs, and (4q+1)-wise independent functions are q-*BU*-secure MACs.*

We can then invoke a recent result about the quantum-security of domain-extension schemes such as NMAC and HMAC [20], and obtain variable-length BU-secure MACs from any qPRF.

Hash-and-MAC. Consider the following natural variation on the blind-forgery experiment. To blind $F : X \to Y$, we first select a hash function $h : X \to Z$ and a blinding set $B_\epsilon \subseteq Z$; we then declare that F will be blinded on $x \in X$ whenever $h(x) \in B_\epsilon$. We refer to this as "hash-blinding." We say that a hash function h is a *Bernoulli-preserving hash* if, for every oracle function F, no QPT can distinguish between an oracle that has been hash-blinded with h, and an oracle that has been blinded in the usual sense.

Recall that the notion of collapsing hash [22] is a quantum analogue of classical collision-resistance, which plays an important role in the construction of post-quantum digital signatures.

Theorem 5. *Let $h : X \to Y$ be a hash function. If h is a Bernoulli-preserving hash, then it is also collapsing. Moreover, against adversaries with classical oracle access, h is a Bernoulli-preserving hash if and only if it is collision-resistant.*

We apply this new notion to show security of the Hash-and-MAC construction $\Pi^h = (\mathsf{Mac}^h, \mathsf{Ver}^h)$ with $\mathsf{Mac}^h_k(m) := \mathsf{Mac}_k(h(m))$.

Theorem 6. *Let $\Pi = (\mathsf{Mac}_k, \mathsf{Ver}_k)$ be a BU-secure MAC with $\mathsf{Mac}_k : X \to Y$, and let $h : Z \to X$ a Bernoulli-preserving hash. Then Π^h is a BU-secure MAC.*

Finally, we show that the Bernoulli-preserving property can be satisfied by pseudorandom constructions, as well as a (public-key) hash based on *lossy functions* from the Learning with Errors (LWE) assumption [16, 21].

2 Preliminaries

Basic notation, conventions. Given a finite set X, the notation $x \in_R X$ will mean that x is a uniformly random element of X. Given a subset B of a set X, let $\chi_B : X \to \{0,1\}$ denote the characteristic function of B, i.e., $\chi_B(x) = 1$ if $x \in B$ and $\chi_B(x) = 0$ else. When we say that a classical function F is efficiently computable, we mean that there exists a uniform family of deterministic classical circuits which computes F.

We will consider three classes of algorithms: (i.) unrestricted algorithms, modeling computationally unbounded adversaries, (ii.) probabilistic poly-time algorithms (PPTs), modeling classical adversaries, and (iii.) quantum poly-time algorithms (QPTs), modeling quantum adversaries. We assume that the latter two are given as polynomial-time uniform families of circuits. For PPTs, these are probabilistic circuits. For QPTs, they are quantum circuits, which may contain both unitary gates and measurements. We will often assume (without loss of generality) that the measurements are postponed to the end of the circuit, and that they take place in the computational basis.

Given an algorithm $\mathcal{A}$, we let $\mathcal{A}(x)$ denote the (in general, mixed) state output by $\mathcal{A}$ on input x. In particular, if $\mathcal{A}$ has classical output, then $\mathcal{A}(x)$ denotes a probability distribution. Unless otherwise stated, the probability is taken over all random coins and measurements of $\mathcal{A}$, and any randomness used to select the input x. If $\mathcal{A}$ is an oracle algorithm and F a classical function, then $\mathcal{A}^F(x)$ is the mixed state output by $\mathcal{A}$ equipped with oracle F and input x; the probability is now also taken over any randomness used to generate F.

We will distinguish between two ways of presenting a function $F : \{0,1\}^n \to \{0,1\}^m$ as an oracle. First, the usual "classical oracle access" simply means that each oracle call grants one classical invocation $x \mapsto F(x)$. This will always be the oracle model for PPTs. Second, "quantum oracle access" will mean that each oracle call grants an invocation of the $(n+m)$-qubit unitary gate $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus F(x)\rangle$. For us, this will always be the oracle model for QPTs. Note that both QPTs and unrestricted algorithms could in principle receive either oracle type.

We will need the following lemma. We use the formulation from [6, Lemma 2.1], which is a special case of a more general "pinching lemma" of Hayashi [12].

Lemma 1. *Let $\mathcal{A}$ be a quantum algorithm and $x \in \{0,1\}^*$. Let $\mathcal{A}_0$ be another quantum algorithm obtained from $\mathcal{A}$ by pausing $\mathcal{A}$ at an arbitrary stage of execution, performing a measurement that obtains one of k outcomes, and then resuming $\mathcal{A}$. Then $\Pr[\mathcal{A}_0(1^n) = x] \geq \Pr[\mathcal{A}(1^n) = x]/k$.*

We denote the trace distance between states ρ and σ by $\delta(\rho, \sigma)$. Recall that this is simply half the trace norm of the difference, i.e., $\delta(\rho, \sigma) = \frac{1}{2}\|\rho - \sigma\|_1$. When ρ and σ are classical probability distributions, the trace distance is equal to the total variation distance.

Quantum-secure pseudorandomness. A quantum-secure pseudorandom function (qPRF) is a family of classical, deterministic, efficiently-computable functions which appear random to QPT adversaries with quantum oracle access.

Definition 2. *An efficiently computable function family $f : K \times X \to Y$ is a quantum-secure pseudorandom function* (qPRF) *if, for all QPTs $\mathcal{D}$,*

$$\left| \Pr_{k \in_R K}\left[\mathcal{D}^{f_k}(1^n) = 1\right] - \Pr_{g \in_R \mathcal{F}_X^Y}\left[\mathcal{D}^{g}(1^n) = 1\right] \right| \leq \mathrm{negl}(n)\,.$$

Here $\mathcal{F}_X^Y$ denotes the set of all functions from X to Y. The standard "GGM+GL" construction of a PRF yields a qPRF when instantiated with a quantum-secure one-way function [24]. One can also construct a qPRF directly from LWE [24]. If we have an a-priori bound on the number of allowed queries, then a computational assumption is not needed.

Theorem 7 (Lemma 6.4 in [5])**.** *Let $q, c \geq 0$ be integers, and $f : K \times X \to Y$ a $(2q + c)$-wise independent family of functions. Let $\mathcal{D}$ be an algorithm making no more than q quantum oracle queries and c classical oracle queries. Then*

$$\Pr_{k \in_R K}\left[\mathcal{D}^{f_k}(1^n) = 1\right] = \Pr_{g \in_R \mathcal{F}_X^Y}\left[\mathcal{D}^{g}(1^n) = 1\right].$$

BZ-unforgeability. Boneh and Zhandry define unforgeability (against quantum queries) for classical MACs as follows [5]. They also show that random functions satisfy this notion.

Definition 3. *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Mac}, \mathsf{Ver})$ be a MAC with message set X. Consider the following experiment with an algorithm $\mathcal{A}$:*

1. Generate key: $k \leftarrow \mathsf{KeyGen}(1^n)$.
2. Generate forgeries: *$\mathcal{A}$ receives quantum oracle for Mac_k, makes q queries, and outputs a string s;*
3. Outcome: *output* win *if s contains $q + 1$ distinct input-output pairs of Mac_k, and* fail *otherwise.*

We say that Π is BZ*-secure if no adversary can succeed at the above experiment with better than negligible probability.*

The Fourier Oracle. Our separation proof will make use of a new technique of Zhandry [26] for working with random oracles. We now briefly describe this idea.

A random function f from n bits to m bits can be viewed as the outcome of a quantum measurement. More precisely, let $\mathcal{H}_F = \bigotimes_{x \in \{0,1\}^n} \mathcal{H}_{F_x}$, where $\mathcal{H}_{F_x} \cong \mathbb{C}^{2^m}$. Then set $f(x) \leftarrow \mathcal{M}_{F_x}(\eta_F)$ with $\eta_F = |\phi_0\rangle\langle\phi_0|^{\otimes 2^n}$, $|\phi_0\rangle = 2^{-\frac{m}{2}} \sum_{y \in \{0,1\}^m} |y\rangle$, and where $\mathcal{M}_{F_x}$ denotes the measurement of the register F_x in the computational basis. This measurement commutes with any $\mathrm{CNOT}_{A:B}$ gate with control qubit A in F_x and target qubit B outside F_x. It follows that, for any quantum algorithm making queries to a random oracle, the output distribution is identical if the algorithm is instead run with the following oracle:

1. Setup: prepare the state η_F.
2. Upon a query with query registers X and Y, controlled on X being in state $|x\rangle$, apply $(\mathrm{CNOT}^{\otimes m})_{F_x:Y}$.
3. After the algorithm has finished, measure F to determine the success of the computation.

We denote the oracle unitary defined in step 2 above by U^{O}_{XYF}. Having defined this oracle representation, we are free to apply any unitary U_H to the oracle state, so long as we then also apply the conjugated query unitary $U_H(\mathrm{CNOT}^{\otimes m})_{F_x:Y}U_H^\dagger$ in place of U^{O}_{XYF}. We choose $U_H = H^{\otimes m 2^n}$, which means that the oracle register starts in the all-zero state now. Applying Hadamard to both qubits reverses the direction of CNOT, i.e.

$$H_A \otimes H_B \mathrm{CNOT}_{A:B} H_A \otimes H_B = \mathrm{CNOT}_{B:A},$$

so the adversary-oracle-state after a first query with query state $|x\rangle_X|\phi_y\rangle_Y$ is

$$|x\rangle_X|\phi_y\rangle_Y|0^m\rangle^{\otimes 2^n} \longmapsto |x\rangle_X|\phi_y\rangle_Y|0^m\rangle^{\otimes(\mathsf{lex}(x)-1)}|y\rangle_{F_x}|0^m\rangle^{\otimes(2^n-\mathsf{lex}(x))}, \tag{2}$$

where $\mathsf{lex}(x)$ denotes the position of x in the lexicographic ordering of $\{0,1\}^n$, and we defined the Fourier basis state $|\phi_y\rangle = H^{\otimes m}|y\rangle$. In the rest of this section, we freely change the order in which tensor products are written, and keep track of the tensor factors through the use of subscripts. This adjusted representation is called the *Fourier oracle* (FO), and we denote its oracle unitary by

$$U^{\mathrm{FO}}_{XYF} = \left(H^{\otimes m 2^n}\right)_F U^{\mathrm{O}}_{XYF} \left(H^{\otimes m 2^n}\right)_F .$$

An essential fact about the FO is that each query can only change the number of non-zero entries in the FO's register by at most one. To formalize this idea, we define the "number operator"

$$N_F = \sum_{x\in\{0,1\}^n} (\mathbb{1} - |0\rangle\langle 0|)_{F_x} \otimes \mathbb{1}^{\otimes(2^n-1)}. \tag{3}$$

The number operator can also be written in its spectral decomposition,

$$N_F = \sum_{l=0}^{2^n} l P_l \qquad \text{where} \qquad P_l = \sum_{r\in S_l} |r\rangle\langle r| ,$$
$$S_l = \left\{ r \in (\{0,1\}^m)^{2^n} \,\middle|\, |\{x\in\{0,1\}^n | r_x \neq 0\}| = l \right\}.$$

Note that the initial joint state of a quantum query algorithm and the oracle (in the FO-oracle picture described above) is in the image of P_0. The following fact is essential for working with the Fourier Oracle; the proof is given in Appendix A.

Lemma 2. *The number operator satisfies*

$$\left\| \left[N_F, U^{FO}_{XYF}\right] \right\|_\infty = 1.$$

In particular, the joint state of a quantum query algorithm and the oracle after the q-th query is in the kernel of P_l for all $l > q$.

3 The problem with BZ-unforgeability

3.1 Intuition, and some obstacles

We begin by motivating our search for a new definition of unforgeability for quantum-secure authentication. We point out a significant security concern not addressed by the existing definition (BZ security) [5]. Before getting to the specifics, we briefly discuss some intuition behind the problem with BZ security, as well as some obstacles to making this intuition concrete.

One intuitive concern about BZ is that it might rule out adversaries who have to measure and thereby "fully destroy" one or more post-query states before they can produce an interesting forgery. At first, constructing such an example does not seem difficult. For instance, let us look at one-time BZ, and construct a MAC from a qPRF f by sampling a key k for f and a superpolynomially-large prime p, and setting

$$\mathsf{Mac}_{k,p}(m) = \begin{cases} 0^n & \text{if } m = p \\ f_k(m \bmod p) & \text{otherwise.} \end{cases} \tag{4}$$

This MAC is forgeable: a quantum adversary can use a single query to perform period-finding on the MAC, and then forge at 0^n. Intuitively, it seems plausible that the MAC might be 1-BZ secure, since period-finding uses a full measurement, and the outputs of the MAC are random everywhere else. As it turns out, this is incorrect, and for a somewhat subtle reason: identifying the hidden symmetry does not fully consume the post-query state. On the contrary, the fact that the period-finding measurement succeeds with high

probability implies that the measured post-query state is not too different from the unmeasured post-query state. In particular, one can still extract an input-output pair of f_k from the measured post-query state.

More generally, let $\mathcal{A}^f$ be an algorithm that makes a uniform-superposition query to some function f and then outputs a property $p(f)$ with non-negligible probability δ. Let $|\psi\rangle = \sum_x |x\rangle|f(x)\rangle$ denote the post-query state, and consider applying to $|\psi\rangle$ the POVM $\{E_p\}_p$ which identifies the property (but measures nothing else.) By assumption, there exists a particular POVM element (i.e., $E_{p(f)}$) that is observed with probability δ. This implies that $|\psi\rangle\langle\psi|$ has roughly $\sqrt{\delta}$ overlap with the post-measurement state $\rho := E_{p(f)}|\psi\rangle\langle\psi|E_{p(f)}^\dagger$. This means that, *even after extracting* $p(f)$, measuring ρ in the computational basis will result in a random input-output pair of f with probability $\delta = 1/\mathsf{poly}(n)$.

One can also try an idea similar to (4) but with Simon's problem rather than period-finding, with the aim of requiring the adversary to consume $O(n)$ queries in order to produce a single good forgery (e.g., whose tag is the nontrivial element k of the hidden subgroup.) This again fails by similar reasoning: we can make all the queries in parallel, postpone the measurement which identifies k until the end of the algorithm, and then observe that each post-query state is not disturbed too much by this measurement. This allows us to extract input-output pairs from every query, with non-negligible success probability overall.

These rather general features of quantum algorithms make it difficult to instantiate the above intuition about the problems with BZ security with a concrete scheme. We formalize these somewhat surprising observations in Lemma 8 in Appendix B.4.

3.2 A counterexample to BZ

In order to construct an explicit function which exemplifies the issues with BZ, we will make use of both the intuition described above, and the well-known (but only partially understood) necessity of *uncomputing* certain registers when attempting to extract some data from an oracle [1]. Consider the following MAC construction.

Construction 2. *Select a uniformly random string* $p \in_R \{0,1\}^n$ *and two random functions* $f, g : \{0,1\}^n \to \{0,1\}^n$*, and define a MAC for* $n+1$ *bit messages by*

$$\mathsf{Mac}_k(x) = \begin{cases} 0^{2n} & x = 0\|p \\ 0^n\|f(x') & x = 0\|x',\ x' \neq p \\ g(x' \bmod p)\|f(x') & x = 1\|x' \end{cases} \tag{5}$$

with $k = (p, f, g)$.

Consider an adversary that queries as follows

$$\sum_{x,y} |1,x\rangle_X |0^n\rangle_{Y_1} |y\rangle_{Y_2} \longmapsto \sum_{x,y} |1,x\rangle_X |g_p(x)\rangle_{Y_1} |y \oplus f(x)\rangle_{Y_2}\,, \tag{6}$$

and then discards the first qubit and the Y_2 register; this yields $\sum_x |x\rangle|g_p(x)\rangle$. The adversary can extract p via period-finding from polynomially-many such states, and then output $(0\|p, 0^{2n})$. This attack only queries the MAC on messages starting with 1 (e.g., "from Alice"), and then forges at a message which starts with 0 (e.g., "from Bob.") We emphasize that the forgery was never queried, not even with negligible amplitude. It is thus intuitively clear that this MAC does not provide secure authentication. And yet, despite this obvious and intuitive vulnerability, this MAC is in fact BZ-secure.

Theorem 8. *The MAC from Construction 2 is* BZ*-secure.*

Proof. Let $\mathcal{A}$ be an adversary that makes q quantum queries and outputs $q+1$ distinct candidate forgeries (where q is selected by $\mathcal{A}$ at runtime.) We let this adversary interact with a mixed oracle, where g and p are treated as random variables, and f is represented as a Fourier Oracle as in Section 2. We denote the relevant quantum registers as follows. First, the quantum oracle for Mac_k is a unitary operator on three

registers: (i.) the $(n+1)$-qubit input register X, (ii.) the n-qubit output register Y_1 into which $g_p : x \mapsto g(x \bmod p)$ is computed, and (iii.) the n-qubit output register Y_2 which interacts with the Fourier Oracle. We set $Y = Y_1Y_2$. The Fourier Oracle is an $(n \cdot 2^n)$-qubit register denoted by F, with the subregister corresponding to input $x \in \{0,1\}^n$ denoted by F_x. Finally, the workspace of $\mathcal{A}$ is a $\mathsf{poly}(n)$-qubit register denoted by E.

Let $|\psi\rangle_{XYEF}$ denote the final state of $\mathcal{A}$ and the Fourier Oracle, after the $q+1$ candidate forgeries have been measured, but prior to any other measurements. Recall that each "number projector" P_l from Section 2 projects F to the subspace spanned by basis states with exactly l non-zero entries. We apply to $|\psi\rangle$ the two-outcome measurement defined by $P_{<q} = \sum_{l=0}^{q-1} P_l$ and its complementary projector $P_{\geq q} = \mathbb{1} - P_{<q}$, effectively measuring whether F contains fewer than q non-zero entries (i.e., registers F_x containing a state other than 0^n); note that it cannot contain more than q by Lemma 2. By Lemma 1, applying this measurement decreases the success probability of $\mathcal{A}$ at any particular task by a factor $1/2$. We handle the two possible outcomes ($< q$ and q) separately.

Case $< q$: Let $|\psi^{<q}\rangle_{XYEF} := P_{<q}|\psi\rangle_{XYEF}$ be the post-measurement state. Note that $P_l|\psi^{<q}\rangle = 0$ for all $l \geq q$, i.e., each basis component of $|\psi^{<q}\rangle$ has fewer than q non-zero entries in F. On the other hand, the output of $\mathcal{A}$ contains at least q candidate input-output pairs (x_i, y_i) of f (since $(0\|p, 0^{2n})$ is the only input-output pair of Mac_k that does not also contain an input-output pair of f.) We apply the q-outcome measurement to F which asks: "among the registers $\{F_{x_i}\}_{i=1}^q$, which is the first one to contain 0^n?" This measurement is defined by projectors

$$\Pi_j := \bigotimes_{i=1}^{j} \left(\mathbb{1} - |0^n\rangle\langle 0^n|\right)_{F_{x_i}} \otimes |0^n\rangle\langle 0^n|_{F_{x_j}} .$$

Adding this measurement to $\mathcal{A}$ ensures that F_{x_j} is in the state 0^n for some j, at the cost of multiplying $\mathcal{A}$'s success probability by $1/q$ (by Lemma 1.) Recalling that, in the Fourier Oracle picture, $f(x_j)$ is the result of QFT-ing and then fully measuring F_{x_j}, we see that $f(x_j)$ is now uniformly random and independent of y_j. The original $\mathcal{A}$ (i.e., without the measurement $\{\Pi_j\}_j$) thus succeeded with probability at most $q \cdot 2^{-n}$. [2]

Case q: We will denote the post-measurement state in this case by $|\psi^q_{g_p}\rangle := P_q|\psi\rangle$, emphasizing that the state was produced by interacting with the oracle g_p. By the BZ-security of f (Theorem 19) it suffices to show that the correct period p is output by $\mathcal{A}$ (by measuring, say, some designated subregister of E of the state $|\psi^q_{g_p}\rangle$) with at most negligible probability. Since testing success at outputting p does not involve the register F, we are free to apply any quantum channel to the F register of $|\psi^q_{g_p}\rangle$. We choose to measure which q subregisters of F are in a non-zero state. This PVM is defined by projectors

$$P_K = \bigotimes_{x \in K} \left(\mathbb{1} - |0^n\rangle\langle 0^n|\right)_{F_x} \otimes \bigotimes_{x \notin K} |0^n\rangle\langle 0^n|_{F_x} \qquad \text{and} \qquad P_{\text{rest}} = \mathbb{1} - \sum_K P_K \,, \tag{7}$$

where $K \subset \{0,1\}^n$ with $|K| = q$. Note that $P_{\text{rest}} = \mathbb{1} - P_q$, so the outcome "rest" never occurs for $|\psi^q_{g_p}\rangle$. In the following we denote by $\mathbf{K}$ the random variable obtained from this measurement. We also set some other random variables in boldface to better distinguish them from particular values they can take.

Now consider the preparation of the state $|\psi^q\rangle$ (by $\mathcal{A}$ and the Fourier Oracle) with an arbitrary choice of oracle function $h : \{0,1\}^n \to \{0,1\}^n$ in place of g_p. We will denote this state by $|\psi^q_h\rangle$. We now show that, conditioned on a particular measurement outcome K, we can arbitrarily relabel the values of h outside K, without affecting the output state of the algorithm.

Lemma 3. *Let $K \subset \{0,1\}^n$ with $|K| = q$ and $h, h' : \{0,1\}^n \to \{0,1\}^n$ a pair of functions satisfying $h(x) = h'(x)$ for all $x \in K$. Then $P_K|\psi^q_h\rangle = P_K|\psi^q_{h'}\rangle$.*

Proof. Let $W^{(j)}_{XYEF} := V^{(j)}_{XYE} U^{(h)}_{XY_1} U^{\text{FO}}_{XY_2F}$, where $V^{(j)}$ is $\mathcal{A}$'s j-th internal unitary, $U^{(h)}$ is the standard oracle unitary for h, and U^{FO} is the Fourier Oracle unitary as described in Section 2. The intermediate states are

$$|\varphi_{h,k}\rangle_{XYEF} := W^{(k)} \cdots W^{(1)} V^{(0)} |0\rangle_{XYEF} \,, \tag{8}$$

[2] This argument amounts to an alternative proof that random functions are BZ-secure.

and the final state is $|\psi_h\rangle := |\varphi_{h,q}\rangle$. By Lemma 2, $P_l|\varphi_{k,h}\rangle = 0$ for all $l > k$, so

$$|\psi_h^q\rangle = P_q|\psi_h\rangle = P_q W^{(q)} \cdots W^{(k+1)}|\varphi_{k,h}\rangle = \sum_{l=0}^{k} P_q W^{(q)} \cdots W^{(k+1)} P_l |\varphi_{k,h}\rangle\,.$$

For the l term in the sum above, the unitary applies $q-k$ queries to $P_l|\varphi_{k,h}\rangle$; by Lemma 2 this term is thus zero unless $l = k$. We can therefore insert a P_k after the k-th query for free when projecting with P_q in the end. Explicitly,

$$|\psi_h^q\rangle = P_q W^{(q)} P_{q-1} W^{(q-1)} P_{q-2} \cdots P_1 W^{(1)} V^{(0)} |0\rangle_{XYEF}\,. \tag{9}$$

We first show that we can apply

$$\tilde{P}_K := \bigotimes_{x\in K} \mathbb{1}_{F_x} \otimes \bigotimes_{x\in K^c} |0^n\rangle\langle 0^n|_{F_x}$$

after every query of $\mathcal{A}$.

We are interested in the state $P_K|\psi\rangle_{XYEF} = P_K P_q|\psi\rangle_{XYEF}$. We can make a similar argument as above to show that we can project with $\tilde{P}_K$ after every query as well. As the FO-unitary is the only one that acts on F, and because $\tilde{P}_K|0\rangle^{\otimes n2^n} = |0\rangle^{\otimes n2^n}$, we can even apply the projector $\tilde{P}_K$ before and after each query. We write $N = N_K + N_{K^c}$, where

$$N_K = \sum_{x\in K} (\mathbb{1} - |0\rangle\langle 0|)_{F_x} \otimes \mathbb{1}^{\otimes(2^n-1)}, \tag{10}$$

i.e. N_K and N_{K^c} measure the number of non-zero entries inside and outside K, respectively. Lemma 2 applies to N_K and N_{K^c} separately, and $P_K N_K|\psi\rangle_{XYEF} = N_K P_K|\psi\rangle_{XYEF} = qP_K|\psi\rangle_{XYEF}$. Therefore we have, defining

$$U_{>k} = V^{(q)}_{XYE} U^{(h)}_{XY_1} U^{\mathrm{FO}}_{XY_2F} V^{(q-1)}_{XYE} U^{(h)}_{XY_1} U^{\mathrm{FO}}_{XY_2F} \ldots V^{(k+1)}_{XYE} U^{(h)}_{XY_1} U^{\mathrm{FO}}_{XY_2F} V^{(k)}_{XYE} \tag{11}$$

and using the same argument as above, that

$$P_K U_{>k} N|\psi^k\rangle = P_K U_{>k} N_K|\psi^k\rangle = kP_K U_{>k}|\psi^k\rangle, \tag{12}$$

and hence

$$P_K U_{>k} N_{K^c}|\psi^k\rangle = P_K U_{>k} N|\psi^k\rangle - P_K U_{>k} N_K|\psi^k\rangle = 0, \tag{13}$$

implying $N_{K^c}|\psi^k\rangle = 0$. But the projector onto the zero-eigenspace of N_{K^c} is $\tilde{P}_K$, so $\tilde{P}_K|\psi^k\rangle = |\psi^k\rangle$.

With an even simpler argument we can insert a projector $P^{\neq 0}_{Y_2} = (\mathbb{1} - |0\rangle\langle 0|)_{Y_2}$ before every query. This is because $U^{\mathrm{FO}}|0\rangle_{Y_2}|\gamma\rangle_{XF} = |0\rangle_{Y_2}|\gamma\rangle_{XF}$, and therefore the number operator eignenvalue does not increase.

To show that $U^{(h)}\tilde{P}_K U^{\mathrm{FO}}\left(P^{\neq 0}_{Y_2} \otimes (\tilde{P}_K)_F\right)$ is independent of the values outside K, we observe that for all $x \notin K$, $y \in \{0,1\}^n \setminus \{0^n\}$ and for all states $|\gamma\rangle_{Y_1EF}$, we have

$$\begin{aligned}
&U^{(g,p)}\tilde{P}_K U^{\mathrm{FO}}\left(P^{\neq 0}_{Y_2} \otimes (\tilde{P}_K)_F\right)|x\rangle_X \otimes |\phi_y\rangle_{Y_2} \otimes |\gamma\rangle_{Y_1EF} \\
&= U^{(g,p)}|x\rangle_X \otimes \left(\tilde{P}_K \left(H^{\otimes n}\right)_{Y_2} \mathrm{CNOT}_{Y_2:F_x}|y\rangle_{Y_2} \otimes \tilde{P}_K|\gamma\rangle_{Y_1EF}\right) \\
&= U^{(g,p)}|x\rangle_X \otimes \left(\tilde{P}_K|0\rangle\langle 0|_{F_x} \left(H^{\otimes n}\right)_{Y_2} \mathrm{CNOT}_{Y_2:F_x}|0\rangle\langle 0|_{F_x}|y\rangle_{Y_2} \otimes \tilde{P}_K|\gamma\rangle_{Y_1EF}\right) \\
&= U^{(g,p)}|x\rangle_X \otimes \left(\tilde{P}_K|0\rangle\langle 0|_{F_x}|y\rangle\langle 0|_{F_x} \otimes |\phi_y\rangle_{Y_2} \otimes \tilde{P}_K|\gamma\rangle_{Y_1EF}\right) \\
&= 0,
\end{aligned} \tag{14}$$

where we have used that for all $x \notin K$ it holds that $|0\rangle\langle 0|_{F_x}\tilde{P}_K = \tilde{P}_K$. This implies that our artificial oracle $U^{(g,p)}\tilde{P}_K U^{\mathrm{FO}}\left(P^{\neq 0}_{Y_2} \otimes (\tilde{P}_K)_F\right)$ (together with a renormalization) only gives $\mathcal{A}$ access to $g(x \mod p)$ for inputs $x \in K$.

This concludes the proof of Lemma 3. □

We now continue with the "**case** q" proof of Theorem 8. We bound $\mathcal{A}$'s success probability separately for each outcome K. Indeed, it suffices to show that for all $K \subset \{0,1\}^n$, $|K| = q$ the probability that the output contains a pair $(0\|p, 0^{2n})$ is negligible if $\mathcal{A}$ continues with

$$|\psi^{q,K}\rangle := \frac{P_K|\psi^q\rangle}{\|P_K|\psi^q\rangle\|_2} \tag{15}$$

in place of $|\psi\rangle$.

We show that the periodic oracle can be replaced by a non-periodic one, except with negligible probability. More precisely, if p' is $\mathcal{A}$'s output, there exists an event E such that $\Pr[E] = 1 - \mathsf{negl}(n)$ and $\Pr[p' = p_0|E, p = p_0] = \Pr[p' = p_0|E, p = 0]$ for all $p_0 \in \{0,1\}^n$. In the following, let us denote the oracle for the MAC of Construction 2 with functions f and g and period p by $\mathcal{O}_{f,g_p}$. We define

$$\mathcal{P}_K^{\text{bad}} = \left\{ p \in \{0,1\}^n \middle| \exists x, x' \in K : p | x - x' \right\}. \tag{16}$$

For $K \subset \{0,1\}^n$ and $p \in \{0,1\}^n$, if $p \notin \mathcal{P}_K^{\text{bad}}$, let $T_{K,p} \subset \{0,1\}^n$ be a transversal for p (i.e., a maximal set such that for $x, y \in T_{K,p}$ it holds that $x \neq y \mod p$) such that $T_{K,p} \cap K = K$. Using this transversal, we can define for each K a random periodic function $g_p^{(K)}$ that is identically distributed with g_p, as follows.

- If $p \in \mathcal{P}_K^{\text{bad}}$, we set $g_p^{(K)}(x) = g(x \mod p)$.
- If $p \notin \mathcal{P}_K^{\text{bad}}$, we set $g_p^{(K)}(x) = g(y)$ for $y \in T_{K,p}$ such that $x = y \mod p$.

For a unitary algorithm $\tilde{\mathcal{A}}$ that makes ℓ queries to an oracle $\mathcal{O}_{f,g_p}$, we define the following procedures:

Procedure 0

1. Sample f, g and p.
2. Run $\tilde{\mathcal{A}}$ with oracle $\mathcal{O}_{f,g_p}$ resulting in a final adversary-oracle state $|\hat{\psi}\rangle$. Apply the measurement $\{P_{\geq \ell}, P_{<\ell}\}$ to F. If outcome is $< \ell$, output "fail."
3. Measure K. If $p \in \mathcal{P}_K^{\text{bad}}$, output "bad." Otherwise, let $|\psi\rangle$ be the post-measurement state of adversary and oracle, i.e. $|\psi\rangle = P_K P_{\geq \ell}|\hat{\psi}\rangle = P_K|\hat{\psi}\rangle$.
4. Output $(K, p, |\psi\rangle)$.

Procedure 0_K

Same as Procedure 0, except with oracle $\mathcal{O}_{f,g_p^{(K)}}$ instead of $\mathcal{O}_{f,g_p}$.

Procedure 1

1. Sample f and g.
2. Run $\mathcal{A}$ with an oracle $\mathcal{O}_{f,g_0}$ resulting in a final adversary-oracle state $|\hat{\psi}\rangle$. Apply the measurement $\{P_{\geq \ell}, P_{<\ell}\}$ to F. If outcome is $< \ell$, output "fail."
3. Measure K and sample p. If $p \in \mathcal{P}_K^{\text{bad}}$, output "bad." Otherwise, let $|\psi\rangle$ be the post-measurement state of adversary and oracle, i.e. $|\psi\rangle = P_K P_{\geq \ell}|\hat{\psi}\rangle = P_K|\hat{\psi}\rangle$.
4. Output $(K, p, |\psi\rangle)$.

We first observe that for all K, the outputs of procedures 0 and 0_K are identically distributed because g_p and $g_{p,K}$ are. Note that for any fixed K, $P_K P_q = P_K$; this, together with Lemma 3, implies that

$$\Pr\left[(K, p, |\psi\rangle) \leftarrow \text{Procedure } 0_K\right] = \Pr\left[(K, p, |\psi\rangle) \leftarrow \text{Procedure 1}\right] \tag{17}$$

It follows that, still for a fixed K,

$$\Pr\left[(K,p,|\psi\rangle) \leftarrow \text{Procedure 0}\right] = \Pr\left[(K,p,|\psi\rangle) \leftarrow \text{Procedure 1}\right]. \tag{18}$$

This implies also that in any of the three procedures, conditioned on the event that the output is neither "fail" nor "bad" and on a fixed first output K, p is uniformly distributed on $\{0,1\}^n \setminus \mathcal{P}_{\text{bad}}$. In other words,

$$\Pr\left[\mathbf{p} = p \,|\, \mathbf{K} = K \wedge \mathbf{p} \notin \mathcal{P}_K^{\text{bad}}\right] = \begin{cases} \left(2^n - |\mathcal{P}_K^{\text{bad}}|\right)^{-1} & p \notin \mathcal{P}_K^{\text{bad}} \\ 0 & \text{else.} \end{cases} \tag{19}$$

Let us denote the event that a procedure outputs a triple $(K,p,|\psi\rangle)$ by "good."

In what follows, we fix a particular period p, an outcome of the period-sampling step (step 1 in Procedures 0 and 0_K and step 3 in Procedure 1). Given a number ℓ of queries we identify three subspaces of $\mathcal{H}_F$ corresponding to the three outcomes "good," "bad" and "fail" of the procedures above:

$$S_{\text{fail}}^\ell = \text{range}(P_{<\ell}) \tag{20}$$

$$S_{\text{bad}}^\ell = \text{span}\left\{\text{range}\left(P_K\right) \middle| K \subset \{0,1\}^n,\ |K| = \ell,\ \exists x,y \in K : p|x-y\right\}, \text{ and} \tag{21}$$

$$S_{\text{good}}^\ell = \left(S_{\text{fail}}^\ell\right)^\perp \cap \left(S_{\text{bad}}^\ell\right)^\perp. \tag{22}$$

We emphasize that the decomposition defined by these subspaces depends on the aforementioned period p. We let P_i^ℓ for $i \in \{\text{good}, \text{bad}, \text{fail}\}$ denote the projectors onto these subsets.

By the above reasoning we know that for any algorithm that makes ℓ queries to an oracle $\mathcal{O}$ and has final state $|\psi_{\mathcal{O}}^\ell\rangle_{AF}$, it holds that $P_{\text{good}}^\ell |\psi_{\mathcal{O}_{f,g_p}}^\ell\rangle_{AF} = P_{\text{good}}^\ell |\psi_{\mathcal{O}_{f,g_0}}^\ell\rangle_{AF}$. It is easy to see that when another query is made, i.e. the $\ell+1$st query of some algorithm, some transitions from S_i^ℓ to $S_j^{\ell+1,p}$ are impossible. We only need one impossibility, namely that according to Lemma 2, $P_i^{\ell+1} U^{\text{FO}} P_{\text{fail}}^\ell = 0$ for all $i \neq$ fail. In words, once an adversary has fallen behind his q-query plan of making one non-trivial query to f in every query, he can never catch up. Also note that for $\ell = 0$, $S_{\text{fail}}^\ell = S_{\text{bad}}^\ell = 0$. It is now easy to show by induction that for a q-query adversary $\mathcal{A}$ with final adversary-oracle state $|\phi\rangle$ it holds that

$$\left\|P_{\text{bad}}^q |\phi\rangle\right\|_2 \leq \sum_{\ell=1}^q \left\| P_{\text{bad}}^\ell U^{\text{FO}} P_{\text{good}}^{\ell-1} |\phi_\ell\rangle \right\|_2, \tag{23}$$

where $|\phi_\ell\rangle$ is the adversary oracle state before the ℓth query. The induction step is proven as follows. Assume the above formula is true for q. Then we have for a $(q+1)$-query adversary $\mathcal{A}$ with final adversary-oracle state $|\phi\rangle$

$$\begin{aligned}
\left\|P_{\text{bad}}^{q+1} |\phi\rangle\right\|_2 &= \left\|P_{\text{bad}}^{q+1} |\psi_{q+1}\rangle\right\|_2 && (24)\\
&\leq \left\|P_{\text{bad}}^{q+1} U^{\text{FO}} P_{\text{good}}^q |\phi_{q+1}\rangle\right\|_2 + \left\|P_{\text{bad}}^{q+1} U^{\text{FO}} P_{\text{bad}}^q |\phi_{q+1}\rangle\right\|_2 && (25)\\
&\quad + \left\|P_{\text{bad}}^{q+1} U^{\text{FO}} P_{\text{fail}}^q |\phi_{q+1}\rangle\right\|_2 && (26)\\
&= \left\|P_{\text{bad}}^{q+1} U^{\text{FO}} P_{\text{good}}^q |\phi_{q+1}\rangle\right\|_2 + \left\|P_{\text{bad}}^{q+1} U^{\text{FO}} P_{\text{bad}}^q |\phi_{q+1}\rangle\right\|_2 && (27)\\
&\leq \left\|P_{\text{bad}}^{q+1} U^{\text{FO}} P_{\text{good}}^q |\phi_{q+1}\rangle\right\|_2 + \left\|U^{\text{FO}} P_{\text{bad}}^q |\phi_{q+1}\rangle\right\|_2 && (28)\\
&= \left\|P_{\text{bad}}^{q+1} U^{\text{FO}} P_{\text{good}}^q |\phi_{q+1}\rangle\right\|_2 + \left\|P_{\text{bad}}^q |\psi_q\rangle\right\|_2. && (29)
\end{aligned}$$

Here we have used the unitary invariance of the Euclidean together with the observation that the state $|\phi\rangle$ is obtained from the state $|\psi_{q+1}\rangle$ right after the $(q+1)$-st query of $\mathcal{A}$ by a unitary acting on the adversary's space only and which therefore commutes with P_{bad}^q in the first, the triangle inequality in the second line,

the observation that $P_i^{\ell+1} U^{\mathrm{FO}} P_{\mathrm{fail}}^{\ell} = 0$ in the third line, and the fact that $\|P\|_\infty \leq 1$ for any projector P in the fourth line. In the fifth line we use the same argument as in the first line, just for $|\phi_{q+1}\rangle$ and $|\psi_q\rangle$. This proves Equation (23).

It remains to bound $\left\| P_{\mathrm{bad}}^{\ell} U^{\mathrm{FO}} P_{\mathrm{good}}^{\ell-1} |\phi_\ell\rangle \right\|_2$. To this end, suppose that we measure the X-register of $|\phi_\ell\rangle$ in the computational basis with outcome $\mathbf{X}_\ell$, as well as $\mathbf{K}^{(\ell-1)}$ the set of nonzero registers in F. According to Equations (18) and (19), we have that $\mathbf{X}_\ell$ and $\mathbf{p}$ are independent and $\mathbf{p}$ is uniformly distributed on $\{0,1\}^n \setminus \mathcal{P}_K^{\mathrm{bad}}$ conditioned on $\mathbf{p} \notin \mathcal{P}_K^{\mathrm{bad}}$ and $\mathbf{K} = K$ for a fixed $(\ell-1)$-element set K. It follows that

$$\Pr\left[\mathbf{p} \in \mathcal{P}^{\mathrm{bad}}_{\mathbf{K}\cup\{\mathbf{X}_\ell\}} \middle| \mathbf{K} = K \wedge \mathbf{p} \notin P_K^{\mathrm{bad}}\right] \tag{30}$$

$$= \Pr\left[\exists y \in K : \mathbf{p} | (\mathbf{X}_\ell - y) \middle| \mathbf{K} = K \wedge \mathbf{p} \notin P_K^{\mathrm{bad}}\right] \tag{31}$$

$$\leq \frac{(\ell-1) 2^{\frac{c' n}{\log n}}}{2^n - \frac{(\ell-1)(\ell-2)}{2} 2^{\frac{c' n}{\log n}}} \leq (\ell-1) 2^{-n\left(1 - \frac{c}{\log n}\right)} \tag{32}$$

Here the last inequality holds for some $0 < c < c'$ and large enough n, and we have used in the third line that there exists a constant $c' > 0$ such that the number of divisors of an integer M is bounded by $2^{c \frac{\log M}{\log \log M}}$ which also implies

$$\left|\mathcal{P}_K^{\mathrm{bad}}\right| \leq \frac{(\ell-1)(\ell-2)}{2} 2^{c \frac{n}{\log n}} \tag{33}$$

for all $K \subset \{0,1\}^n$, $|K| = \ell$. We would now like to relate the above probability to

$$\mathbb{E}\left[\left\| P_{\mathrm{bad}}^{\ell} U^{\mathrm{FO}} P_{\mathrm{good}}^{\ell-1} |\phi_\ell\rangle \right\|_2^2\right].$$

To this end we analyze how the operator $P_{\mathrm{bad}}^{\ell} U^{\mathrm{FO}} P_{\mathrm{good}}^{\ell-1}$ behaves on states of the form $|x\rangle_X \otimes |\phi_y\rangle \otimes |\zeta\rangle_{EF}$ such that $(P_K)_F |\zeta\rangle_{EF} = |\zeta\rangle_{EF}$ for some fixed $K \not\ni x$ and $p \in \{0,1\}^n$ such that $p \notin \mathcal{P}_K^{\mathrm{bad}}$. We calculate

$$U^{\mathrm{FO}} P_{\mathrm{good}}^{\ell-1} |x\rangle_X \otimes |\phi_y\rangle \otimes |\zeta\rangle_{EF} \tag{34}$$

$$= U^{\mathrm{FO}} |x\rangle_X \otimes |\phi_y\rangle \otimes |\zeta\rangle_{EF_K} \otimes |0^{n(2^n - \ell + 1)}\rangle_{F_{K^c}} \tag{35}$$

$$= \left(H^{\otimes n}\right)_Y \mathrm{CNOT}_{Y:F_x} |x\rangle_X \otimes |y\rangle \otimes |\zeta\rangle_{EF_K} \otimes |0^{n(2^n - \ell + 1)}\rangle_{F_{K^c}} \tag{36}$$

$$= \left(H^{\otimes n}\right)_Y |x\rangle_X \otimes |y\rangle \otimes |\zeta\rangle_{EF_K} \otimes |y\rangle_{F_x} \otimes |0^{n(2^n - \ell)}\rangle_{F_{(K\cup\{x\})^c}} \tag{37}$$

$$= |x\rangle_X \otimes |\phi_y\rangle \otimes |\zeta\rangle_{EF_K} \otimes |y\rangle_{F_x} \otimes |0^{n(2^n - \ell)}\rangle_{F_{(K\cup\{x\})^c}}. \tag{38}$$

In the first equation we have use the assumptions that $(P_K)_F |\zeta\rangle_{EF} = |\zeta\rangle_{EF}$ and $p \notin \mathcal{P}_K^{\mathrm{bad}}$; the rest of the calculation is analogous to Equation (14). This implies that

$$P_{K\cup\{x\}} U^{\mathrm{FO}} P_{\mathrm{good}}^{\ell-1} |x\rangle_X \otimes |\phi_y\rangle \otimes |\zeta\rangle_{EF} = U^{\mathrm{FO}} P_{\mathrm{good}}^{\ell-1} |x\rangle_X \otimes |\phi_y\rangle \otimes |\zeta\rangle_{EF} \tag{39}$$

and therefore

$$P_{\mathrm{bad}}^{\ell} U^{\mathrm{FO}} P_{\mathrm{good}}^{\ell-1} |x\rangle_X \otimes |\phi_y\rangle \otimes |\zeta\rangle_{EF} \tag{40}$$

$$= \begin{cases} U^{\mathrm{FO}} P_{\mathrm{good}}^{\ell-1} |x\rangle_X \otimes |\phi_y\rangle \otimes |\zeta\rangle_{EF} & \text{if } \exists x' \in K : p | (x - x') \\ 0 & \text{otherwise.} \end{cases} \tag{41}$$

We therefore calculate for a fixed p,

$$\begin{aligned}
&\left\| P^{\ell}_{\mathrm{bad}} U^{\mathrm{FO}} P^{\ell-1}_{\mathrm{good}} |\phi_\ell\rangle \right\|_2^2 \\
&= \left\| \sum_{\substack{K\subset\{0,1\}^n \\ |K|=\ell-1 \\ p\notin\mathcal{P}^{\mathrm{bad}}_K}} \sum_{x\in\{0,1\}^n} P^{\ell}_{\mathrm{bad}} U^{\mathrm{FO}} \left(|x\rangle\langle x|_X \otimes P_K\right) |\phi_\ell\rangle \right\|_2^2 \\
&= \left\| U^{\mathrm{FO}} \sum_{\substack{K\subset\{0,1\}^n \\ |K|=\ell-1 \\ p\notin\mathcal{P}^{\mathrm{bad}}_K}} \sum_{\substack{x\in\{0,1\}^n\setminus K \\ \exists x'\in K: p|(x-x')}} \left(|x\rangle\langle x|_X \otimes P_K\right) |\phi_\ell\rangle \right\|_2^2 \\
&= \sum_{\substack{K\subset\{0,1\}^n \\ |K|=\ell-1 \\ p\notin\mathcal{P}^{\mathrm{bad}}_K}} \sum_{\substack{x\in\{0,1\}^n\setminus K \\ \exists x'\in K: p|(x-x')}} \left\| \left(|x\rangle\langle x|_X \otimes P_K\right) |\phi_\ell\rangle \right\|_2^2 \\
&= \Pr\left[p\notin\mathcal{P}^{\mathrm{bad}}_{\mathbf{K}} \wedge p\in\mathcal{P}^{\mathrm{bad}}_{\mathbf{K}\cup\{\mathbf{X}_\ell\}} \middle| \mathbf{p}=p \right] .
\end{aligned}$$

Using Equation (32) we can bound

$$\begin{aligned}
&\mathbb{E}_{p\leftarrow\{0,1\}^n}\left[\Pr\left[p\notin\mathcal{P}^{\mathrm{bad}}_{\mathbf{K}} \wedge p\in\mathcal{P}^{\mathrm{bad}}_{\mathbf{K}\cup\{\mathbf{X}_\ell\}} \right]\right] \\
&= \Pr\left[\mathbf{p}\notin\mathcal{P}^{\mathrm{bad}}_{\mathbf{K}} \wedge \mathbf{p}\in\mathcal{P}^{\mathrm{bad}}_{\mathbf{K}\cup\{\mathbf{X}_\ell\}} \right] \\
&= \sum_{K\subset\{0,1\}^n} \Pr\left[\mathbf{p}\notin\mathcal{P}^{\mathrm{bad}}_{K} \wedge \mathbf{p}\in\mathcal{P}^{\mathrm{bad}}_{K\cup\{\mathbf{X}_\ell\}} \middle| \mathbf{K}=K_0 \right] \Pr[\mathbf{K}=K_0] \\
&= \sum_{K\subset\{0,1\}^n} \Pr\left[\mathbf{p}\in\mathcal{P}^{\mathrm{bad}}_{K\cup\{\mathbf{X}_\ell\}} \middle| \mathbf{K}=K_0 \wedge \mathbf{p}\notin\mathcal{P}^{\mathrm{bad}}_{\mathbf{K}} \right] \Pr\left[\mathbf{p}\notin\mathcal{P}^{\mathrm{bad}}_{K} \wedge \mathbf{K}=K \right] \\
&\le \Pr\left[\mathbf{p}\notin\mathcal{P}^{\mathrm{bad}}_{\mathbf{K}} \right] (\ell-1) 2^{-n\left(1-\frac{c}{\log n}\right)} \\
&\le (\ell-1) 2^{-n\left(1-\frac{c}{\log n}\right)} .
\end{aligned}$$

Here we have used Equation (32) in the first inequality. The probability in the first line is taken over a run of the adversary with a fixed period and random g and f, and in the other lines the period is picked uniformly at random from $\{0,1\}^n$ as for a properly generated key in Construction 2. The last two equations together imply

$$\mathbb{E}\left[\left\| P^{\ell}_{\mathrm{bad}} U^{\mathrm{FO}} P^{\ell-1}_{\mathrm{good}} |\phi_\ell\rangle \right\|_2^2 \right] \le (\ell-1) 2^{-n\left(1-\frac{c}{\log n}\right)}. \tag{42}$$

Plugging this into Equation (23) yields

$$\begin{aligned}
\Pr\left[\mathbf{p} \in \mathcal{P}_{\mathbf{K}}^{\mathrm{bad}}\right] &= \mathbb{E}\left[\|P_{\mathrm{bad}}^{q}|\phi\rangle\|_2^2\right] \\
&\leq \mathbb{E}\left[\left(\sum_{i=1}^{q}\left\|P_{\mathrm{bad}}^{\ell}U^{\mathrm{FO}}P_{\mathrm{good}}^{\ell-1}|\phi_\ell\rangle\right\|_2\right)^2\right] \\
&\leq q\sum_{i=1}^{q}\mathbb{E}\left[\left\|P_{\mathrm{bad}}^{\ell}U^{\mathrm{FO}}P_{\mathrm{good}}^{\ell-1}|\phi_\ell\rangle\right\|_2^2\right] \\
&\leq \left(\sum_{\ell=1}^{q}\sqrt{(\ell-1)2^{-n\left(1-\frac{c}{\log n}\right)}}\right)^2 \\
&\leq \frac{q^2(q-1)}{2}2^{-n\left(1-\frac{c}{\log n}\right)} \qquad (43)
\end{aligned}$$

using the Cauchy-Schwartz inequality in the second line. This finally implies that the adversary's guess p' is equal to p and the measurement $< q$ vs. $\geq q$ returns $\geq q$ with probability at most

$$\begin{aligned}
&\Pr[\mathbf{p} = \mathbf{p}' \wedge \text{"} \geq q\text{"}] && (44)\\
\leq &\Pr\left[\mathbf{p} \in \mathcal{P}_{\mathbf{K}}^{\mathrm{bad}} \wedge \text{"} \geq q\text{"}\right] + \Pr\left[\mathbf{p} \notin \mathcal{P}_{\mathbf{K}}^{\mathrm{bad}} \wedge \mathbf{p} = \mathbf{p}' \wedge \text{"} \geq q\text{"}\right] && (45)\\
\leq &\Pr\left[\mathbf{p} \in \mathcal{P}_{\mathbf{K}}^{\mathrm{bad}} \wedge \text{"} \geq q\text{"}\right] + \Pr\left[\mathbf{p} = \mathbf{p}' \middle| \mathbf{p} \notin \mathcal{P}_{\mathbf{K}}^{\mathrm{bad}} \wedge \text{"} \geq q\text{"}\right] && (46)\\
\leq &\frac{q^2(q-1)}{2}2^{-n\left(1-\frac{c}{\log n}\right)} + \left(2^n - \frac{(\ell-1)(\ell-2)}{2}2^{\frac{c'n}{\log n}}\right)^{-1} && (47)\\
\leq &\mathsf{negl}(n)\,. && (48)
\end{aligned}$$

Here we have used Equation (43) and the uniformity of $\mathbf{p}$ conditioned on $\mathbf{p} \notin \mathcal{P}_{\mathbf{K}}^{\mathrm{bad}}$ and $\mathbf{K} = K$ in the last line. □

Remark. As we will later show, this BZ-secure MAC is not secure in our proposed notion of blind-unforgeability. It's not hard to see that it is also not GYZ-secure. Indeed, observe that the forging adversary described above queries on messages starting with 0 only, and then forges successfully on a message starting with 1. If the scheme was GYZ secure, then in the accepting case, the portion of this adversary between the query and the final output would have a simulator which leaves the computational basis invariant. Such a simulator cannot change the first bit of the message from 0 to 1, a contradiction.

4 The new notion: Blind-Unforgeability

4.1 Formal definition

For simplicity, our discussion will concentrate on the case of MACs with canonical verification. The case of digital signatures with deterministic signing algorithm is a simple adaptation. We will also later show how to extend our approach to the case of MACs and signatures with non-canonical verification. We begin by defining a "blinding" operation. Let $f : X \to Y$ and $B \subseteq X$. We let

$$Bf(x) = \begin{cases} \perp & \text{if } x \in B, \\ f(x) & \text{otherwise.} \end{cases}$$

We say that f has been "blinded" by B. In this context, we will be particularly interested in the setting where elements of X are placed in B independently at random with a particular probability ϵ; we let B_ϵ denote this random variable. (It will be easy to infer X from context, so we do not reflect it in the notation.)

Next, we define a security game in which an adversary is tasked with using a blinded MAC oracle to produce a valid input-output pair in the blinded set.

Definition 4. *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Mac}, \mathsf{Ver})$ *be a MAC with message set* X*. Let* $\mathcal{A}$ *be an algorithm, and* $\epsilon : \mathbb{N} \to [0,1]$ *an efficiently computable function. The blind forgery experiment* $\mathsf{BlindForge}_{\mathcal{A},\Pi}(n,\epsilon)$ *proceeds as follows:*

1. Generate key: $k \leftarrow \mathsf{KeyGen}(1^n)$.
2. Generate blinding: *select* $B_\epsilon \subseteq X$ *by placing each* m *into* B_ϵ *independently with probability* $\epsilon(n)$.
3. Produce forgery: $(m,t) \leftarrow \mathcal{A}^{B_\epsilon \mathsf{Mac}_k}(1^n)$.
4. Outcome: *output* 1 *(win) if* $\mathsf{Ver}_k(m,t) = \mathsf{acc}$ *and* $m \in B_\epsilon$*; otherwise output* 0 *(lose.)*

We say that a scheme is blind-unforgeable if, for any efficient adversary, the probability of winning the game is negligible. The probability is taken over the choice of key, the choice of blinding set, and any internal randomness of the adversary. We remark that specifying an adversary requires specifying (in a uniform fashion) both the algorithm $\mathcal{A}$ and the blinding function ϵ.

Definition 5. *A MAC* Π *is blind-unforgeable (BU) if for every polynomial-time uniform adversary* $(\mathcal{A},\epsilon)$*,* $\Pr\left[\mathsf{BlindForge}_{\mathcal{A},\Pi}(n,\epsilon(n)) = 1\right] \leq \mathsf{negl}(n)$.

We also define the "q-time" variant of the blinded forgery game, which is identical to Definition 4 except that the adversary is only allowed to make q queries to $B_\epsilon \mathsf{Mac}_k$ in step (3). We call the resulting game $\mathsf{BlindForge}^q_{\mathcal{A},\Pi}(n,\epsilon)$, and give the corresponding definition of q-time security (now against computationally unbounded adversaries.)

Definition 6. *A MAC* Π *is* q*-time blind-unforgeable (*q*-BU) if for every* q*-query adversary* $(\mathcal{A},\epsilon)$*, we have* $\Pr\left[\mathsf{BlindForge}^q_{\mathcal{A},\Pi}(n,\epsilon(n)) = 1\right] \leq \mathsf{negl}(n)$.

The above definitions are agnostic regarding the computational class of the adversary and the type of oracle provided. For example, selecting PPT adversaries and classical oracles in Definition 5 yields a definition of classical unforgeability; we will later show that this is equivalent to standard EUF-CMA. The main focus of our work will be on BU against QPTs with quantum oracle access, and q-BU against unrestricted adversaries with quantum oracle access.

4.1.1 Some technical details.

We now remark on a few details in the usage of BU. First, strictly speaking, the blinding sets in the security games above cannot be generated efficiently. However, a pseudorandom blinding set will suffice. Pseudorandom blinding sets can be generated straightforwardly using an appropriate pseudorandom function, such as a PRF against PPTs or a qPRF against QPTs. A precise description of how to perform this pseudorandom blinding is given in the proof of Corollary 3. Note that simulating the blinding requires computing and uncomputing the random function, so we must make two quantum queries for each quantum query of the adversary. Moreover, verifying whether the forgery is in the blinding set at the end requires one additional classical query. This means that $(4q+1)$-wise independent functions are both necessary and sufficient for generating blinding sets for q-query adversaries (see [5, Lemma 6.4].) In any case, an adversary which behaves differently in the random-blinding game versus the pseudorandom-blinding game immediately yields a distinguisher against the corresponding pseudorandom function.

The blinding symbol. There is some flexibility in how one defines the blinding symbol $\perp$. In situations where the particular instantiation of the blinding symbol might matter, we will adopt the convention that the blinded version Bf of $f : \{0,1\}^n \to \{0,1\}^\ell$ is defined by setting $Bf : \{0,1\}^n \to \{0,1\}^{\ell+1}$, where $Bf(m) = 0^\ell||1$ if $m \in B$ and $Bf(m) = f(m)||0$ otherwise. One advantage of this convention (i.e., that $\perp = 0^\ell||1$) is that we can compute on and/or measure the blinded bit (i.e., the $(\ell+1)$-st bit) without affecting the output register of the function. This will also turn out to be convenient for uncomputation.

Non-canonical verification. Some care is needed when using the above definitions for MACs and digital signatures with non-canonical verification. Consider the following MAC. Let $F : \{0,1\}^n \to \{0,1\}^n$ be a random function, and define $\mathsf{Mac}(m) = F(m)||0$ and $\mathsf{Ver}(m, t||b) = \delta_{t,F(m)}$. Forging is trivial: we query on 0^n and flip the last bit of the tag, producing the valid and fresh pair $(0^n, F(0^n)||1)$. And yet, this MAC is BU secure: producing either $F(x)||0$ or $F(x)||1$ for a blinded x would imply an efficient algorithm that predicts values of F. (We will later show that random functions are BU-secure.)

This issue is addressed with a simple and natural adjustment: we blind (message, tag) pairs rather than just messages. We briefly describe this for the case of MACs. Let $\Pi = (\mathsf{KeyGen}, \mathsf{Mac}, \mathsf{Ver})$ be a MAC with message set M, randomness set R and tag set T, so that $\mathsf{Mac}_k : M \times R \to T$ and $\mathsf{Ver}_k : M \times T \to \{\mathsf{acc}, \mathsf{rej}\}$ for every $k \leftarrow \mathsf{KeyGen}$. Given a parameter ϵ and an adversary $\mathcal{A}$, the blind forgery game proceeds as follows:

1. Generate key: $k \leftarrow \mathsf{KeyGen}$; generate blinding: select $B_\epsilon \subseteq M \times T$ by placing pairs (m, t) in B_ϵ independently with probability ϵ;
2. Produce forgery: produce (m, t) by executing $\mathcal{A}(1^n)$ with quantum oracle access to the function

$$B_\epsilon \mathsf{Mac}_{k;r}(m) := \begin{cases} \perp & \text{if } (m, \mathsf{Mac}_k(m; r)) \in B_\epsilon, \\ \mathsf{Mac}_k(m; r) & \text{otherwise.} \end{cases}$$

   where r is sampled uniformly for each oracle call.
3. Outcome: output 1 if $\mathsf{Ver}_k(m, t) = \mathsf{acc} \wedge (m, t) \in B_\epsilon$; otherwise output 0.

Security is then defined as before: Π is secure if for all adversaries $\mathcal{A}$ (and their declared ϵ), the success probability at winning the above game is negligible. Note that, for the case of canonical MACs, this definition coincides with Definition 5.

5 Intuitive security and the meaning of BU

In this section, we gather a number of results which build confidence in BU as a correct definition of unforgeability in our setting. We begin by showing that a wide range of "intuitively forgeable" MACs (indeed, all such examples we have examined) are correctly characterized by BU as insecure.

5.1 Intuitively forgeable schemes

As indicated earlier, BU security rules out any MACs where an attacker can query a subset of the message space and forge outside that region. To make this claim precise, we first define the *query support* $\mathsf{supp}(\mathcal{A})$ of an oracle algorithm $\mathcal{A}$. Let $\mathcal{A}$ be a quantum query algorithm with oracle access to the quantum oracle $\mathcal{O}$ for a classical function from n to m bits. Without loss of generality, $\mathcal{A}$ proceeds by applying a sequence of unitaries $\mathcal{O}U_q\mathcal{O}U_{q-1}\cdots\mathcal{O}U_1$ to the initial state $|0\rangle_{XYZ}$, followed by a POVM $\mathcal{E}$. Here, X and Y are the input and output registers of the function and Z is the algorithm's workspace. Let $|\psi_i\rangle$ be the intermediate state of $\mathcal{A}$ after the application of U_i. Then $\mathsf{supp}(\mathcal{A})$ is defined to be the set of input strings x such that there exists a function $f : \{0,1\}^n \to \{0,1\}^m$ such that $\langle x|_X|\psi_i\rangle \neq 0$ for at least one $i \in \{1, ..., q\}$ when $\mathcal{O} = \mathcal{O}_f$.

Theorem 9. *Let $\mathcal{A}$ be a QPT such that $\mathsf{supp}(\mathcal{A}) \cap R = \emptyset$ for some $R \neq \emptyset$. Let Mac be a MAC, and suppose $\mathcal{A}^{\mathsf{Mac}_k}(1^n)$ outputs a valid pair $(m, \mathsf{Mac}_k(m))$ with $m \in R$ with non-negligible probability. Then Mac is not BU-secure.*

To prove Theorem 9, we will need a fact which controls the change in the output state of an algorithm resulting from applying a blinding to its oracle. Given an oracle algorithm $\mathcal{A}$ and two oracles F and G, the trace distance between the output of $\mathcal{A}$ with oracle F and with oracle G is denoted by $\delta(\mathcal{A}^F(1^n), \mathcal{A}^G(1^n))$. Given two functions $F, P : \{0,1\}^n \to \{0,1\}^m$, we define the function $F \oplus P$ by $(F \oplus P)(x) = F(x) \oplus P(x)$.

Theorem 10. *Let $\mathcal{A}$ be a quantum query algorithm making at most T queries, and $F : \{0,1\}^n \to \{0,1\}^m$ a function. Let $B \subseteq \{0,1\}^n$ be a subset chosen by independently including each element of $\{0,1\}^n$ with probability ϵ, and $P : \{0,1\}^n \to \{0,1\}^m$ be any function with support B. Then*

$$\mathop{\mathbb{E}}_{B}\left[\delta\left(\mathcal{A}^{F}(1^n), \mathcal{A}^{F\oplus P}(1^n)\right)\right] \leq 2T\sqrt{\epsilon}\,.$$

The proof is a relatively straightforward adaptation of a hybrid argument in the spirit of the lower bound for Grover search [3]. We provide the complete proof in Appendix B.1. We are now ready to prove Theorem 9.

Proof of Theorem 9. Let $\mathcal{A}$ be a quantum algorithm with $\mathsf{supp}(\mathcal{A})$ for any oracle. By our hypothesis,

$$\tilde{p} := \Pr_{k,(m,t)\leftarrow\mathcal{A}^{\mathsf{Mac}_k}(1^n)}\left[\mathsf{Mac}_k(m) = t \wedge m \notin \mathsf{supp}(\mathcal{A})\right] \geq n^{-c}\,,$$

for some $c > 0$ and sufficiently large n. Since $\mathsf{supp}(A)$ is a fixed set, we can think of sampling a random B_ε as picking $B_0 := B_\varepsilon \cap \mathsf{supp}(A)$ and $B_1 := B_\varepsilon \cap \overline{\mathsf{supp}(A)}$ independently. Let "blind" denote the random experiment of $\mathcal{A}$ running on Mac_k blinded by a random B_ε: $k, B_\varepsilon, (m,t) \leftarrow \mathcal{A}^{B_\varepsilon\mathsf{Mac}_k}(1^n)$, which is equivalent to $k, B_0, B_1, (m,t) \leftarrow \mathcal{A}^{B_0\mathsf{Mac}_k}(1^n)$. The probability that $\mathcal{A}$ wins the BU game is

$$\begin{aligned}
p &:= \Pr_{\mathsf{blind}}[\mathsf{Mac}_k(m) = t \wedge m \in B_\varepsilon] \\
&\geq \Pr_{\mathsf{blind}}[\mathsf{Mac}_k(m) = t \wedge m \in B_1] \\
&\geq \Pr_{\mathsf{blind}}[\mathsf{Mac}_k(m) = t \wedge m \in B_1 \mid m \notin \mathsf{supp}(A)] \cdot \Pr_{\mathsf{blind}}[m \notin \mathsf{supp}(A)] \\
&= \Pr_{\substack{k,B_0 \\ (m,t)\leftarrow\mathcal{A}^{B_0\mathsf{Mac}_k}}}[\mathsf{Mac}_k(m) = t \wedge m \notin \mathsf{supp}(A)] \cdot \Pr_{\substack{k,B_1 \\ (m,t)\leftarrow\mathcal{A}^{B_0\mathsf{Mac}_k}}}[m \in B_1 | m \notin \mathsf{supp}(A)] \\
&\geq \left(\tilde{p} - 2T\sqrt{\varepsilon}\right)\varepsilon \\
&\geq \frac{\tilde{p}^3}{27T^2}\,.
\end{aligned}$$

Here the second-to-last step follows from Theorem 10; in the last step, we chose $\varepsilon = (\tilde{p}/3T)^2$. We conclude that $\mathcal{A}$ breaks the BU security of the MAC. □

Now recall the adversary against the BZ-secure (but intuitively insecure) Construction 2, as described in Section 3.2. This yields the following.

Theorem 11. *The MAC from Construction 2 is BU-insecure.*

5.2 Relationship to other definitions

In the purely classical setting, our notion is equivalent to EUF-CMA. In the strong unforgeability case, this means BU with blinding on message-tag pairs, as described in Section 4.1.1.

Proposition 2. *A MAC is EUF-CMA if and only if it is blind-unforgeable against classical adversaries.*

Proof. Set $F_k = \mathsf{Mac}_k$. Consider an adversary $\mathcal{A}$ which violates EUF-CMA. Such an adversary, given 1^n and oracle access to F_k (for $k \in_R \{0,1\}^n$), produces a fresh forgery (m,t) with non-negligible probability $s(n)$ This same adversary (when coupled with an appropriate ϵ) breaks the MAC under the BU definition. Specifically, let $p(n)$ be the running time of $\mathcal{A}$, in which case $\mathcal{A}$ clearly makes no more than $p(n)$ queries, and define $\epsilon(n) = 1/p(n)$. Consider now a particular $k \in \{0,1\}^n$ and a particular sequence r of random coins for $\mathcal{A}^{F_k}(1^n)$. If this run of $\mathcal{A}$ results in a forgery (m,t), observe that with probability at least $(1-\epsilon)^{p(n)} \approx e^{-1}$ in the choice of B_ϵ, we have $F_k(q) = B_\epsilon F_k(q)$ for every query q made by $\mathcal{A}$. On the other hand, $B_\epsilon(m) = \perp$ with (independent) probability ϵ. It follows that the winning probability of $\mathcal{A}$ in the blind forgery experiment is at least $\epsilon s(n)/e = \Omega(s(n)/p(n))$.

On the the other hand, suppose that $(\mathcal{A}, \epsilon)$ is an adversary that wins blind-unforgeability with inverse-polynomial probability $r(n)$. Consider now the EUF-CMA adversary $\mathcal{A}'^{F_k}(1^n)$ which simulates the adversary $\mathcal{A}^{(\cdot)}(1^n)$ by answering oracle queries according to a locally-simulated version of $B_\epsilon F_k$; specifically, the adversary $\mathcal{A}'$ proceeds by drawing a subset $B_{\epsilon(n)} \subseteq \{0,1\}^*$ (pseudorandomly) and answering queries made by $\mathcal{A}$ according to $B_\epsilon F$. Note that, when $x \in B_\epsilon$, this query is answered without an oracle call to $F(x)$. In addition, $\mathcal{A}'$ can construct the set B_ϵ "on the fly," by determining, when a particular query q is made by $\mathcal{A}$, whether $q \in B_\epsilon$ and "remembering" this information in case the query is asked again ("lazy sampling"). With probability $r(n)$, $\mathcal{A}$ produces a forgery on a point which was not queried by $\mathcal{A}'$, as desired. It follows that $\mathcal{A}'$ produces a (conventional) forgery with non-negligible probability when given F_k for $k \in_R \{0,1\}^n$. □

As we have shown above, there are examples which are BZ-secure but BU-insecure (and intuitively broken.) An interesting question is whether BU-security implies BZ security. While we do not fully settle this question, we give some indication that this may be the case. Specifically, we show (in Appendix B.2) that any adversary that makes k quantum queries and outputs ck^2 forgeries (for some constant c) with high probability, can also be used to break BU.

6 Blind-forgery secure schemes

We now show that a number of natural MAC constructions satisfy blind-unforgeability.

6.1 Random schemes

Theorem 12. *Let $R : X \to Y$ be a uniformly random function such that $1/|Y|$ is negligible in n. Then R is a blind-forgery secure MAC.*

Proof. For simplicity, we assume that the function is length-preserving; the proof generalizes easily. Let $\mathcal{A}$ be an efficient quantum adversary. The oracle $B_\epsilon R$ supplied to $\mathcal{A}$ during the blind-forgery game is determined entirely by B_ϵ and the restriction of R to the complement of B_ϵ. On the other hand, the forgery event

$$\mathcal{A}^{B_\epsilon F_k}(1^n) = (m,t) \wedge |m| \geq n \wedge F_k(m) = t \wedge B_\epsilon F_k(m) = \perp$$

depends additionally on values of R at points in B_ϵ. To reflect this decomposition, given R and B_ϵ define $R_\epsilon : B_\epsilon \to Y$ to be the restriction of R to the set B_ϵ and note that—conditioned on $B_\epsilon R$ and B_ϵ—the random variable R_ϵ is drawn uniformly from the space of all (length-preserving) functions from B_ϵ into Y. Note, also, that for every n the purported forgery $(m,t) \leftarrow \mathcal{A}^{B_\epsilon R}(1^n)$ is a (classical) random variable depending only on $B_\epsilon R$. In particular, conditioned on B_ϵ, (m,t) is independent of R_ϵ. It follows that, conditioned on $m \in B_\epsilon$, that $t = R_\epsilon(m)$ with probability no more than $1/2^n$ and hence $\phi(n,\epsilon) \leq 2^{-n}$, as desired. □

Next, we show that a qPRF is a blind-unforgeable MAC.

Corollary 3. *Let m and t be $\mathsf{poly}(n)$, and $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^t$ a qPRF. Then F is a blind-forgery-secure fixed-length MAC (with length $m(n)$).*

Proof. For a contradiction, let $\mathcal{A}$ be a QPT which wins the blind forgery game for a certain blinding factor $\varepsilon(n)$, with running time $q(n)$ and success probability $\delta(n)$. We will use $\mathcal{A}$ to build a quantum oracle distinguisher $\mathcal{D}$ between the qPRF F and the perfectly random function family $\mathcal{F}_m^t$ with the same domain and range.

First, let $k = q(n)$ and let $\mathcal{H}$ be a family of $(4k+1)$-wise independent functions with domain $\{0,1\}^m$ and range $\{0,1,\ldots,1/\varepsilon(n)\}$. The distinguisher $\mathcal{D}$ first samples $h \in_R \mathcal{H}$. Set $B_h := h^{-1}(0)$. Given its oracle $\mathcal{O}_f$, $\mathcal{D}$ can implement the function $B_h f$ (quantumly) as follows:

$$\begin{aligned}|x\rangle|y\rangle &\mapsto |x\rangle|y\rangle|H_x\rangle|\delta_{h(x),0}\rangle \mapsto |x\rangle|y\rangle|H_x\rangle|\delta_{h(x),0}\rangle|f(x)\rangle \\ &\mapsto |x\rangle|y \oplus f(x)\cdot(1-\delta_{h(x),0})\rangle|H_x\rangle|\delta_{h(x),0}\rangle|f(x)\rangle \\ &\mapsto |x\rangle|y \oplus f(x)\cdot(1-\delta_{h(x),0})\rangle\,.\end{aligned}$$

Here we used the CCNOT (Toffoli) gate from step 2 to 3 (with one control bit reversed), and uncomputed both h and f in the last step. After sampling h, the distinguisher $\mathcal{D}$ will execute $\mathcal{A}$ with the oracle $B_h f$. If $\mathcal{A}$ successfully forges a tag for a message in B_h, $\mathcal{A}'$ outputs "pseudorandom"; otherwise "random."

Note that the function $B_h f$ is perfectly ϵ-blinded if h is a perfectly random function. Note also that the entire security experiment with $\mathcal{A}$ (including the final check to determine if the output forgery is blind) makes at most $2k$ quantum queries and 1 classical query to h, and is thus (by Theorem 7) identically distributed to the perfect-blinding case.

Finally, by Theorem 12, the probability that $\mathcal{D}$ outputs "pseudorandom" when $f \in_R \mathcal{F}_m^t$ is negligible. By our initial assumption about $\mathcal{A}$, the probability that $\mathcal{D}$ outputs "pseudorandom" becomes $\delta(n)$ when $f \in_R F$. It follows that $\mathcal{D}$ distinguishes F from perfectly random. □

Next, we give a information-theoretically secure q-time MACs (Definition 6.)

Theorem 13. *Let $\mathcal{H}$ be a $(4q+1)$-wise independent function family with range Y, such that $1/|Y|$ is a negligible function. Then $\mathcal{H}$ is a q-time BU-secure MAC.*

Proof. Let $(\mathcal{A}, \epsilon)$ be an adversary for the q-time game $\mathsf{BlindForge}^q_{\mathcal{A},h}(n, \epsilon(n))$, where h is drawn from $\mathcal{H}$. We will use $\mathcal{A}$ to construct a distinguisher $\mathcal{D}$ between $\mathcal{H}$ and a random oracle. Given access to an oracle $\mathcal{O}$, $\mathcal{D}$ first runs $\mathcal{A}$ with the blinded oracle $B\mathcal{O}$, where the blinding operation is performed as in the proof of Corollary 3 (i.e., via a $(4q+1)$-wise independent function with domain size $1/\epsilon(n)$.) When $\mathcal{A}$ is completed, it outputs (m, σ). Next, $\mathcal{D}$ queries $\mathcal{O}$ on the message m and outputs 1 if and only if $\mathcal{O}(m) = \sigma$ and $m \in B$. Let $\gamma_{\mathcal{O}}$ be the probability of the output being 1.

We consider two cases: (i.) $\mathcal{O}$ is drawn as a random oracle R, and (ii.) $\mathcal{O}$ is drawn from the family $\mathcal{H}$. By Theorem 7, since $\mathcal{D}$ makes only $2q$ quantum queries and one classical query to $\mathcal{O}$, its output is identical in the two cases. Observe that γ_R (respectively, $\gamma_{\mathcal{H}}$) is exactly the success probability of $\mathcal{A}$ in the blind-forgery game with random oracle R (respectively, $\mathcal{H}$). We know from Theorem 12 that γ_R is negligible; it follows that $\gamma_{\mathcal{H}}$ is as well. □

Several domain-extension schemes, including NMAC (a.k.a. encrypted cascade), HMAC, and AMAC, can transform a fixed-length qPRF to a qPRF that takes variable-length inputs [20] . As a corollary, starting from a qPRF, we also obtain a number of quantum blind-unforgeable variable-length MACs.

6.2 Hash-and-MAC

To authenticate messages of arbitrary length with a fixed-length MAC, it is common practice to first compress a long message by a *collision-resistant* hash functon and then apply the MAC. This is known as Hash-and-MAC. However, when it comes to BU-security (and quantum security in general), collision-resistance may not be sufficient. We therefore propose a new notion which generalizes collision-resistance in the quantum setting, and show that it is sufficient for Hash-and-MAC with BU security.

Recall that, given a subset B of a set X, $\chi_B : X \to \{0,1\}$ denotes the characteristic function of B.

Definition 7 (Bernoulli-preserving hash)**.** *Let $\mathcal{H} : X \to Y$ be an efficiently computable function family. Define the following distributions on subsets of X:*

1. *$\mathcal{B}_\epsilon$: generate $B_\epsilon \subseteq X$ by placing $x \in B_\epsilon$ independently with probability ϵ. Output B_ϵ.*
2. *$\mathcal{B}_\epsilon^{\mathcal{H}}$: generate $C_\epsilon \subseteq Y$ by placing $y \in C_\epsilon$ independently with probability ϵ. Sample $h \in \mathcal{H}$ and define $B_\epsilon^h := \{x \in X : h(x) \in C_\epsilon\}$. Output B_ϵ^h.*

We say that $\mathcal{H}$ is a Bernoulli-preserving hash if for all adversaries $(\mathcal{A}, \epsilon)$,

$$\left| \Pr_{B \leftarrow \mathcal{B}_\epsilon} [\mathcal{A}^{\chi_B}(1^n) = 1] - \Pr_{B \leftarrow \mathcal{B}_\epsilon^{\mathcal{H}}} [\mathcal{A}^{\chi_B}(1^n) = 1] \right| \leq \mathsf{negl}(n) \ .$$

The motivation for the name is simply that selecting $\mathcal{B}_\epsilon$ can be viewed as a Bernoulli process taking place on the set X, while $\mathcal{B}_\epsilon^h$ can be viewed as the pullback (along h) of a Bernoulli process taking place on Y.

We show that the standard Hash-and-MAC construction will preserve BU security, if we instantiate the hash function with a Bernoulli-preserving hash. Recall that, given a MAC $\Pi = (\mathsf{Mac}_k, \mathsf{Ver}_k)$ with message set X and a function $h : Z \to X$, there is a MAC $\Pi^h := (\mathsf{Mac}_k^h, \mathsf{Ver}_k^h)$ with message set Z defined by $\mathsf{Mac}_k^h = \mathsf{Mac}_k \circ h$ and $\mathsf{Ver}_k^h(m,t) = \mathsf{Ver}_k(h(m),t)$.

Theorem 14 (Hash-and-MAC with Bernoulli-preserving hash)**.** *Let $\Pi = (\mathsf{Mac}_k, \mathsf{Ver}_k)$ be a BU-secure MAC with $\mathsf{Mac}_k : X \to Y$, and let $h : Z \to X$ a Bernoulli-preserving hash. Then Π^h is a BU-secure MAC.*

The proof follows in a straightforward way from the definitions of BU and Bernoulli-preserving hash; the details are in Appendix B.6. In Appendix B, we also provide a number of additional results about Bernoulli-preserving hash functions. These results can be summarized as follows.

Theorem 15.

- *If H is a random oracle or a qPRF, then it is a Bernoulli-preserving hash.*
- *If H is $4q$-wise independent, then it is a Bernoulli-preserving hash against q-query adversaries.*
- *Under the LWE assumption, there is a (public-key) family of Bernoulli-preserving hash functions.*
- *If we only allow classical oracle access, then the Bernoulli-preserving property is equivalent to standard collision-resistance.*
- *Bernoulli-preserving hash functions are collapsing.*

The collapsing property is another quantum generalization of collision-resistance, proposed in [22].

Acknowledgements. CM thanks Ronald de Wolf for helpful discussions on query complexity. CM was supported by a Netherlands Organisation for Scientific Research (NWO) VIDI grant (639.022.519). CM thanks QuICS for its hospitality. GA acknowledges support from NSF grant CCF-1763736. AR acknowledges support from NSF grant CCF-1763773.

References

[1] Scott Aaronson. Quantum lower bound for recursive fourier sampling. *Quantum Information & Computation*, 3(2):165–174, 2003.

[2] Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. Unforgeable quantum encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 489–519, Cham, 2018. Springer International Publishing.

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

[4] Jean-François Biasse and Fang Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 893–902, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics.

[5] Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In *Advances in Cryptology - EUROCRYPT 2013*, pages 592–608. Springer, 2013.

[6] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In *Advances in Cryptology – CRYPTO 2013*, pages 361–379. Springer, 2013.

[7] Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography. Technical report, National Institute of Standards and Technology, 2016.

[8] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In *Advances in Cryptology – EUROCRYPT 2016*, pages 559–585. Springer Berlin Heidelberg, 2016.

[9] Kirsten Eisenträger, Sean Hallgren, Alexei Kitaev, and Fang Song. A quantum algorithm for computing the unit group of an arbitrary degree number field. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 293–302, New York, NY, USA, 2014. ACM.

[10] Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. Semantic security and indistinguishability in the quantum world. In *Advances in Cryptology – CRYPTO 2016*, pages 60–89. Springer, 2016.

[11] Sumegha Garg, Henry Yuen, and Mark Zhandry. New security notions and feasibility results for authentication of quantum data. In *Advances in Cryptology – Crypto 2017*, pages 342–371. Springer, 2017.

[12] Masahito Hayashi. Optimal sequence of quantum measurements in the sense of Stein's lemma in quantum hypothesis testing. *Journal of Physics A: Mathematical and General*, 35(50):10759, 2002.

[13] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In *Advances in Cryptology – CRYPTO 2016*, pages 207–237. Springer, 2016.

[14] Hidenori Kuwakado and Masakatu Morii. Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In *Proceedings of IEEE International Symposium on Information Theory*, pages 2682–2685, June 2010.

[15] Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type Even-Mansour cipher. In *Proceedings of the International Symposium on Information Theory and its Applications*, pages 312–316. IEEE Computer Society, 2012.

[16] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 187–196, New York, NY, USA, 2008. ACM.

[17] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 475–484. ACM, 2014.

[18] Thomas Santoli and Christian Schaffner. Using Simon's algorithm to attack symmetric-key cryptographic primitives. *Quantum Information & Computation*, 17(1&2):65–78, 2017.

[19] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM journal on computing*, 26(5):1484–1509, 1997.

[20] Fang Song and Aaram Yun. Quantum security of NMAC and related constructions - PRF domain extension against quantum attacks. In *Advances in Cryptology - CRYPTO 2017*, pages 283–309. Springer, 2017.

[21] Dominique Unruh. Collapse-binding quantum commitments without random oracles. In *Advances in Cryptology—ASIACRYPT 2016*, volume 10032, pages 166–195. Springer-Verlag New York, Inc., 2016.

[22] Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 497–527, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[23] Salil P Vadhan et al. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

[24] Mark Zhandry. How to construct quantum random functions. In *Proceedings of the 53rd Annual Symposium on Foundations of Computer Science*, FOCS '12, pages 679–687, Washington, DC, USA, 2012. IEEE Computer Society.

[25] Mark Zhandry. Quantum Lightning Never Strikes the Same State Twice. *ArXiv e-prints*, November 2017.

[26] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. Cryptology ePrint Archive, Report 2018/276, 2018. https://eprint.iacr.org/2018/276.

A The Fourier Oracle number operator

Recall the "number operator" N_F, defined in Equation (3) from Section 2.

Lemma 4. *The number operator satisfies* $\left\|\left[N_F, U^{FO}_{XYF}\right]\right\|_\infty = 1$. *In particular, the joint state of a quantum query algorithm and the oracle after the q-th query is in the kernel of* P_l *for all* $l > q$.

Proof. Let $|\psi\rangle_{XYEF}$ be an arbitrary query state, where X and Y are the query input and output registers, E is the algorithm's internal register and F is the FO register. We expand the state in the computational basis of X,

$$|\psi\rangle_{XYEF} = \sum_{x\in\{0,1\}^n} p(x)|x\rangle_X|\psi_x\rangle_{YEF}. \tag{49}$$

Set $\widetilde{\mathrm{CNOT}}_{A:B} = H_A\mathrm{CNOT}_{A:B}H_A$ and observe that

$$U^{\mathrm{FO}}_{XYF}|x\rangle_X|\psi_x\rangle_{YEF} = |x\rangle_X\big(\widetilde{\mathrm{CNOT}}^{\otimes m}\big)_{Y:F_x}|\psi_x\rangle_{YEF}\,.$$

Therefore

$$\begin{aligned}\Big[N_F, U_{XYF}\Big]|x\rangle_X|\psi_x\rangle_{YEF} &= |x\rangle_X\left[N_F, \left(\widetilde{\mathrm{CNOT}}^{\otimes m}\right)_{Y:F_x}\right]|\psi_x\rangle_{YEF}\\ &= |x\rangle_X\left[(\mathbb{1}-|0\rangle\langle 0|)_{F_x}, \big(\widetilde{\mathrm{CNOT}}^{\otimes m}\big)_{Y:F_x}\right]|\psi_x\rangle_{YEF}.\end{aligned}$$

It follows that

$$\begin{aligned}\Big\|\Big[N_F, U_{XYF}\Big]|\psi\rangle_{XYEF}\Big\|_2 &= \sum_{x\in\{0,1\}^n} p(x)\,\|[N_F, U_{XYF}]\,|\psi_x\rangle_{YEF}\|_2\\ &= \sum_{x\in\{0,1\}^n} p(x)\left\|\left[(\mathbb{1}-|0\rangle\langle 0|)_{F_x}, \big(\widetilde{\mathrm{CNOT}}^{\otimes m}\big)_{Y:F_x}\right]|\psi_x\rangle_{YEF}\right\|_2\\ &\le \left\|\left[(\mathbb{1}-|0\rangle\langle 0|)_{F_{0^n}}, \big(\widetilde{\mathrm{CNOT}}^{\otimes m}\big)_{Y:F_{0^n}}\right]\right\|_\infty,\end{aligned} \tag{50}$$

where we have used the definition of the operator norm and the normalization of $|\psi\rangle_{XYEF}$ in the last line. For a unitary U and a projector P, it is easy to see that $\|[U,P]\|_\infty \le 1$, as $[U,P] = PU(\mathbb{1}-P) - (\mathbb{1}-P)UP$ is a sum of two operators that have orthogonal support and singular values smaller or equal to 1. We therefore get $\|[N_F, U_{XYF}]\,|\psi\rangle_{XYEF}\|_2 \le 1$, and as the state $|\psi\rangle$ was arbitrary, this implies $\left\|[N_F, U_{XYF}]\right\|_\infty \le 1$. The example from equation (2) shows that the above is actually an equality. The observation that $P_l\eta_F = 0$ for all $l > 0$ and an induction argument proves the second statement of the lemma. □

B More on Bernoulli-preserving hash

In this section, we prove several results about Bernoulli-preserving hash functions. Recalling Definition 7, we refer to blinding according to $\mathcal{B}_\epsilon$ as "uniform blinding," and blinding according to $\mathcal{B}_\epsilon^{\mathcal{H}}$ as "hash blinding." First, we show that random and pseudorandom functions are Bernoulli-preserving, and that this property is equivalent to collision-resistance against classical queries.

Lemma 5. *Let $H : X \to Y$ be a function such that $1/|Y|$ is negligible. Then*

1. *If H is a random oracle or a* qPRF*, then it is a Bernoulli-preserving hash.*
2. *If H is $4q$-wise independent, then it is a Bernoulli-preserving hash against q-query adversaries.*

Proof. The claim for random oracles is obvious: by statistical collision-resistance, uniform blinding is statistically indistinguishable from hash-blinding. The remaining claims follow from the observation that one can simulate one quantum query to $\chi_{B_\epsilon^h}$ using two quantum queries to h (see, e.g., the proof of Corollary 3.) □

Theorem 16. *A function $h : \{0,1\}^* \to \{0,1\}^n$ is Bernoulli-preserving against classical-query adversaries if and only if it is collision-resistant.*

Proof. First, the Bernoulli-preserving hash property implies collision-resistance: testing whether two colliding inputs are either i) both not blinded or both blinded, or ii) exactly one of them is blinded, yields always outcome i) when dealing with a hash-blinded oracle and a uniformly random outcome for a blinded oracle and $\varepsilon = 1/2$. On the other hand, consider an adversary $\mathcal{A}$ that has inverse polynomial distinguishing advantage between blinding and hash-blinding, and let $x_1, ..., x_q$ be its queries. Assume for a contradiction that with overwhelming probability $h(x_i) \neq h(x_j)$ for all $x_i \neq x_j$. Then with that same overwhelming probability the blinded and hash-blinded oracles are both blinded independently with probability ε on each x_i and are hence statistically indistinguishable, a contradiction. It follows that with non-negligible probability there exist two queries $x_i \neq x_j$ such that $h(x_i) = h(x_j)$, i.e., $\mathcal{A}$ has found a collision. □

Bernoulli-preserving hash from LWE. We have observed that any qPRF is immediately a Bernoulli-preserving hash function. Such a hash can be constructed from various quantum-safe computational assumptions (e.g., LWE). Unfortunately, a qPRF requires a secret key, and typically does not give a short digest, which would result in long tags. (In practice, it is probably more convenient and more reliable to instantiate a qPRF from block ciphers, which may not be ideal for message authentication.)

Here we point out that one can also construct a public Bernoulli-preserving hash function based on the quantum security of LWE. Specifically, we show that the collapsing hash function in [21] is also a Bernoulli-preserving hash. This construction relies on a lossy function family $F : X \to Y$ and a universal hash function $G = \{g_k : Y \to Z\}_{k \in \mathcal{K}}$. A lossy function family admits two types of keys: a lossy key $s \leftarrow \mathcal{D}_{los}$ and an injective key $s \leftarrow \mathcal{D}_{inj}$, which are computationally indistinguishable. $F_s : X \to Y$ under a lossy key s is compressing, i.e., $|im(F_s)| << |Y|$; whereas under an injective key s, F_s is injective. See [21, Definition 2] for a formal definition, and [16] for an explicit construction based on LWE. There exist efficient constructions for universal hash families by various means [23]. With these ingredients in hand, one then constructs a hash function family $H = \{h_{s,k}\}$ by $h_{s,k} := g_k \circ F_s$ with public parameters generated by $s \leftarrow \mathcal{D}_{los}, k \leftarrow \mathcal{K}$. The proof of Bernoulli-preserving for this hash function is similar to Unruh's proof that H is collapsing; see Appendix B.7 for details.

Relationship to collapsing. Finally, we relate the Bernoulli-preserving property to another quantum generalization of classical collision-resistance: the collapsing property. Collapsing hash functions are particularly relevant to post-quantum signatures. We first define the collapsing property (slightly rephrasing Unruh's original definition [22]) as follows. Let $h : X \to Y$ be a hash function, and let $\mathcal{S}_X$ and $\mathcal{S}_{XY}$ be the set of quantum states (i.e., density operators) on registers corresponding to the sets X and $X \times Y$, respectively. We define two channels from $\mathcal{S}_X$ to $\mathcal{S}_{XY}$. First, $\mathcal{O}_h$ receives X, prepares $|0\rangle$ on Y, applies $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus h(x)\rangle$,

and then measures Y fully in the computational basis. Second, $\mathcal{O}'_h$ first applies $\mathcal{O}_h$ and then also measures X fully in the computational basis.

$$\begin{aligned}\mathcal{O}_h: &\quad |x\rangle_X \overset{h}{\longmapsto} |x, h(x)\rangle_{X,Y} \overset{\text{measure } Y}{\longmapsto} (\rho^y_X, y)\,,\\ \mathcal{O}'_h: &\quad |x\rangle_X \overset{h}{\longmapsto} |x, h(x)\rangle_{X,Y} \overset{\text{measure } X\&Y}{\longmapsto} (x, y)\,.\end{aligned}$$

If the input is a pure state on X, then the output is either a superposition over a fiber $h^{-1}(s) \times \{s\}$ of h (for $\mathcal{O}_h$) or a classical pair $(x, h(x))$ (for $\mathcal{O}'_h$) .

Definition 8 (Collapsing)**.** *A hash function h is collapsing if for any single-query QPT $\mathcal{A}$, it holds that* $\left|\Pr[\mathcal{A}^{\mathcal{O}_h}(1^n) = 1] - \Pr[\mathcal{A}^{\mathcal{O}'_h}(1^n) = 1]\right| \leq \mathsf{negl}(n)$.

To prove that Bernoulli-preserving hash implies collapsing, we need a technical fact. Recall that any subset $S \subseteq \{0,1\}^n$ is associated with a two-outcome projective measurement $\{\Pi_S, \mathbb{1} - \Pi_S\}$ on n qubits defined by $\Pi_S = \sum_{x \in S} |x\rangle\langle x|$. We will write Ξ_S for the channel (on n qubits) which applies this measurement.

Lemma 6. *Let $S_1, S_2, \ldots, S_{cn}$ be subsets of $\{0,1\}^n$, each of size 2^{n-1}, chosen independently and uniformly at random. Let Ξ_{S_j} denote the two-outcome measurement defined by S_j, and denote their composition $\tilde{\Xi} := \Xi_{S_{cn}} \circ \Xi_{S_{cn-1}} \circ \cdots \circ \Xi_{S_1}$. Let Ξ denote the full measurement in the computational basis. Then $\Pr\left[\tilde{\Xi} = \Xi\right] \geq 1 - 2^{-\varepsilon n}$, whenever $c \geq 2 + \varepsilon$ with $\varepsilon > 0$,*

A proof is given in Appendix B.3. We remark that, to efficiently implement each Ξ_S with a random subset S, we can sample $h_i : [M] \to [N]$ from a pairwise-independent hash family (sampling an independent h_i for each i), and then define $x \in S$ iff. $h(x) \leq N/2$. For any input state $\sum_{x,z} \alpha_{x,z}|x, z\rangle$, we can compute

$$\sum_{x,z} \alpha_{x,z}|x, z\rangle \mapsto \sum_{x,z} |x, z\rangle|b(x)\rangle, \qquad \text{where } b(x) := h(x) \overset{?}{\leq} N/2\,,$$

and then measure $|b(x)\rangle$. Pairwise independence is sufficient by Theorem 7 because only one quantum query is made.

Theorem 17. *If $h : X \to Y$ is Bernoulli-preserving, then it is collapsing.*

Proof. Let $\mathcal{A}$ be an adversary with inverse-polynomial distinguishing power in the collapsing game. Choose n such that $X = \{0,1\}^n$. We define $k = cn$ hybrid oracles $H_0, H_1, \ldots, H_k$, where hybrid H_j is a channel from $\mathcal{S}_X$ to $\mathcal{S}_{XY}$ which acts as follows: (1.) adjoin $|0\rangle_Y$ and apply the unitary $|x\rangle_X|y\rangle_Y \mapsto |x\rangle_X|y \oplus h(x)\rangle_Y$; (2.) measure the Y register in the computational basis; (3.) repeat j times: (i.) select a uniformly random subset $S \subseteq X$ of size 2^{n-1}; (ii.)apply the two-outcome measurement Ξ_S to the X register; (4.) output registers X and Y.

Clearly, H_0 is identical to the $\mathcal{O}_h$ channel in the collapsing game. By Lemma 6, H_k is indistinguishable from the $\mathcal{O}'_h$. By our initial assumption and the triangle inequality, there exists a j such that

$$\left|\Pr[\mathcal{A}^{H_j}(1^n) = 1] - \Pr[\mathcal{A}^{H_{j+1}}(1^n) = 1]\right| \geq 1/\mathsf{poly}(n)\,. \tag{51}$$

We now build a distinguisher $\mathcal{D}$ against the Bernoulli-preserving property (with $\epsilon = 1/2$) of h. It proceeds as follows: (1.) run $\mathcal{A}(1^n)$ and place its query state in register X; (2.) simulate oracle H_j on XY (use 2-wise independent hash to select sets S); (3.) prepare an extra qubit in the $|0\rangle$ state in register W, and invoke the oracle for χ_B on registers X and W; (4.) measure and discard register W; (5.) return XY to $\mathcal{A}$, and output what it outputs.

We now analyze $\mathcal{D}$. After the first two steps of H_j (compute h, measure output register) the state of $\mathcal{A}$ (running as a subroutine of $\mathcal{D}$) can be expressed as

$$\sum_z \sum_{x \in h^{-1}(s)} \alpha_{xz}|x\rangle_X|s\rangle_Y|z\rangle_Z\,.$$

Here Z is a side information register private to $\mathcal{A}$. Applying the j partial measurements (third step of H_j) then results in a state of the form $\sum_z \sum_{x\in M} \beta_{xz}|x\rangle|s\rangle|z\rangle$, where M is some subset of $h^{-1}(s)$. Applying the oracle for χ_B into an extra register then yields

$$\sum_z \sum_{x\in M} \beta_{xz}|x\rangle|s\rangle|z\rangle|\chi_B(x)\rangle_W\,.$$

Now consider the two cases of the Bernoulli-preserving game.

First, in the "hash-blinded" case, $B = h^{-1}(C)$ for some set $C \subseteq Y$. This implies that $\chi_B(x) = \chi_C(h(x)) = \chi_C(s)$ for all $x \in M$. It follows that W simply contains the classical bit $\chi_C(s)$; computing this bit, measuring it, and discarding it will thus have no effect. The state returned to $\mathcal{A}$ will then be identical to the output of the oracle H_j. Second, in the "uniform blinding" case, B is a random subset of X of size 2^{n-1}, selected uniformly and independently of everything else in the algorithm thus far. Computing the characteristic function of B into an extra qubit and then measuring and discarding that qubit implements the channel Ξ_B, i.e., the measurement $\{\Pi_B, \mathbb{1} - \Pi_B\}$. It follows that the state returned to $\mathcal{A}$ will be identical to the output of oracle H_{j+1}.

By (51), it now follows that $\mathcal{D}$ is a successful distinguisher in the Bernoulli-preserving hash game for h, and that h is thus not a Bernoulli-preserving hash. □

B.1 A simulation theorem

Theorem 18. *Let $\mathcal{A}$ be a quantum query algorithm making at most T queries, and $F : \{0,1\}^n \to \{0,1\}^m$ a function. Let $B \subseteq \{0,1\}^n$ be a subset chosen by independently including each element of $\{0,1\}^n$ with probability ϵ, and $P : \{0,1\}^n \to \{0,1\}^m$ be any function with support B. Then*

$$\mathop{\mathbb{E}}_B\big[\delta\big(\mathcal{A}^F(1^n), \mathcal{A}^{F\oplus P}(1^n)\big)\big] \le 2T\sqrt{\epsilon}\,.$$

Proof. For a function $Q : \{0,1\}^n \to \{0,1\}^m$, we let $\mathcal{O}_Q$ denote the unitary map $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus Q(x)\rangle$. Recall that $\mathcal{A}$ is specified by a fixed initial state $|\phi_0\rangle$ in some finite-dimensional Hilbert space, a sequence of T unitary "computation" operators $C_1, \ldots, C_k$, and a POVM $\{P_i : i \in I\}$. The distribution (on I) resulting from the algorithm applied to the oracle $\mathcal{O}_Q$ is given by applying the POVM to the state

$$|\phi^Q\rangle := C_T\mathcal{O}_Q C_{T-1}\cdots\mathcal{O}_Q C_0|\phi_0\rangle\,.$$

Recall that if the trace distance between two such states satisfies

$$\delta\big(|\phi^{Q_1}\rangle, |\phi^{Q_2}\rangle\big) := \sqrt{1 - |\langle\phi^{Q_1}|\phi^{Q_2}\rangle|^2} \le \epsilon$$

then the distance in total variation between the distributions produced by *any* POVM on these two states is no more than ϵ. In our case, we are interested in controlling $\mathbb{E}_B[\delta(\phi^F, \phi^{P\oplus F})]$. Define $F' = F \oplus P$. In preparation for a standard hybrid argument, define

$$|\phi_k\rangle = \underbrace{C_T\mathcal{O}_{F'}\cdots\mathcal{O}_{F'}}_{(\dagger)} C_k \underbrace{\mathcal{O}_F\ldots\mathcal{O}_F C_0}_{(\ddagger)}|\phi_0\rangle \qquad |\phi_k^F\rangle = C_k\underbrace{\mathcal{O}_F\ldots\mathcal{O}_F C_0}_{(\ddagger)}|\phi_0\rangle\,,$$

so that all oracle invocations in (†) are answered according to $\mathcal{O}_{F'}$ and all those in (‡) are answered according to $\mathcal{O}_F$. Since δ is a metric on pure states, we have

$$\mathbb{E}\,\delta(|\phi^F\rangle, |\phi^{P\oplus F}\rangle) \le \mathbb{E}\sum_{k=1}^T \delta(|\phi_k\rangle, |\phi_{k-1}\rangle) = \sum_{k=1}^T \mathbb{E}\,\delta(|\phi_k\rangle, |\phi_{k-1}\rangle)\,.$$

Note that δ is invariant under (simultaneous) unitary action, and hence for any F, B, and P,

$$\begin{aligned}
&\delta(|\phi_k\rangle, |\phi_{k-1}\rangle)\\
&= \delta(C_T\mathcal{O}_{F'}\cdots\mathcal{O}_{F'}C_k\mathcal{O}_F\ldots\mathcal{O}_F C_0|\phi_0\rangle, C_T\mathcal{O}_{F'}\cdots\mathcal{O}_{F'}C_{k-1}\mathcal{O}_F\ldots\mathcal{O}_F C_0|\phi_0\rangle)\\
&= \delta(\mathcal{O}_F C_{k-1}\ldots\mathcal{O}_F C_0|\phi_0\rangle, \mathcal{O}_{F'}C_{k-1}\ldots\mathcal{O}_F C_0|\phi_0\rangle)\\
&= \delta(\mathcal{O}_F|\phi_{k-1}^F\rangle, \mathcal{O}_{F'}|\phi_{k-1}^F\rangle) = \delta(|\phi_{k-1}^F\rangle, \mathcal{O}_F\mathcal{O}_{F'}|\phi_{k-1}^F\rangle) = \delta(|\phi_{k-1}^F\rangle, \mathcal{O}_P|\phi_{k-1}^F\rangle)\,.
\end{aligned}$$

For pure states $|\psi\rangle$ and $|\psi'\rangle$, $\delta(|\psi\rangle, |\psi'\rangle) \leq \||\psi\rangle - |\psi'\rangle\|$. Note that $|\psi\rangle = \Pi_B|\psi\rangle + (I - \Pi_B)|\psi\rangle$, and O_P operates identically on $(I - \Pi_B)|\psi\rangle$. Therefore

$$\begin{aligned}
\mathop{\mathbb{E}}_{B}[\delta(\phi^F, \phi^{P\oplus F})] &\leq T \max_{|\phi\rangle} \mathbb{E}\, \||\phi\rangle - \mathcal{O}_P|\phi\rangle\| \\
&= T \max_{|\phi\rangle} \mathop{\mathbb{E}}_{B} \|\Pi_B|\phi\rangle - \mathcal{O}_P\Pi_B|\phi\rangle + (1 - \mathcal{O}_P)(I - \Pi_B)|\phi\rangle\| \\
&\leq T \max_{|\phi\rangle} \mathop{\mathbb{E}}_{B}(\|\Pi_B|\phi\rangle\| + \|\mathcal{O}_P\Pi_B|\phi\rangle\|) \\
&= 2T \max_{|\phi\rangle} \mathop{\mathbb{E}}_{B} \|\Pi_B|\phi\rangle\| \\
&\leq 2T \max_{|\phi\rangle} \sqrt{\mathop{\mathbb{E}}_{B} |\langle\phi|\Pi_B|\phi\rangle|} \quad \text{(Jensen's inequality)}\,.
\end{aligned}$$

Let π be a uniformly random element of the symmetric group on $\{0,1\}^n$ and U_π be the unitary operator associated with the permutation π. We have that

$$\mathop{\mathbb{E}}_{B}\Big[|\langle\phi|\Pi_B|\phi\rangle|\Big] = \mathop{\mathbb{E}}_{B}\mathop{\mathbb{E}}_{\pi}\Big[|\langle\phi|U_\pi\Pi_B U_\pi^{-1}|\phi\rangle|\Big] = 2^{-n}\mathop{\mathbb{E}}_{B}[\mathrm{Tr}\,(\Pi_B)] = \epsilon\,.$$

Thus we conclude that $\mathbb{E}_B[\delta(\phi^F, \phi^{P\oplus F})] \leq 2T\sqrt{\varepsilon}$. □

B.2 BU implies quadratic BZ

It's interesting to ask if BU-security implies BZ-security, as the BZ definition certainly captures a natural family of attacks that one would like to rule out. We are unable to settle this question completely, but provide some weaker connection. Specifically, we show that if a function is BU-secure, then it is BZ-secure with a weaker definition of BZ-security that forbids an adversary from producing ck^2 forgeries from k queries with high probability.

For this purpose, consider a function $M : X \to Y$ and a BZ-type adversary $\mathcal{A}$ which, given oracle access to M, makes some k queries and produces ck^2 forgeries (with probability 1); here $c \geq 1$ is a constant we set later in the discussion. We consider the behavior of this adversary $\mathcal{A}^{B_\epsilon M}$ supplied with an oracle $B_\epsilon M$ blinded at a random set B_ϵ. We will show that for an appropriate value of c and ϵ, this adversary produces a family of forgeries which includes at least one blinded forgery with constant probability. Finally selecting one of these forgeries at random produces an adversary that breaks the BU security definition.

Returning to the BZ-adversary $\mathcal{A}$, we say that a particular blinding set B is γ-*evasive* if

$$\Pr_{\mathcal{A}}[\mathcal{A}^M \text{ outputs no elements of } B] \geq \gamma\,.$$

(Note that this event is determined by running $\mathcal{A}$ with the *unblinded* oracle M.) Observing that

$$\Pr_{\mathcal{A}, B_\epsilon}[\mathcal{A}^M \text{ outputs no elements of } B_\epsilon] \leq (1-\epsilon)^{ck^2} \leq e^{-c\epsilon k^2}\,.$$

We note that (by Markov's inequality),

$$\Pr_{B_\epsilon}[B_\epsilon \text{ is } \gamma\text{-evasive}] \leq e^{-c\epsilon k}/\gamma\,.$$

Similarly, we say that a particular blinding set B is γ-divergent if

$$\|D_{\mathcal{A}^M} - D_{\mathcal{A}^{BM}}\|_{\text{t.v.}} \geq \gamma\,,$$

where D_M is the distribution of outputs of $\mathcal{A}^M$ and D_{BM} is the distribution of outputs of $\mathcal{A}^{BM}$ when M is blinded on set B. In light of Theorem 2,

$$\mathop{\mathbb{E}}_{B_\epsilon}[\|D_M - D_{B_\epsilon M}\|_{\text{t.v.}}] \leq 2k\sqrt{\epsilon}$$

and it follows by Markov's inequality that

$$\Pr_{B_\epsilon}[B_\epsilon \text{ is } \gamma\text{-divergent}] = \Pr_{B_\epsilon}[\|D_M - D_{BM}\|_{\text{t.v.}} \geq \gamma] \leq 2k\sqrt{\epsilon}/\gamma\,.$$

Fixing $\gamma \leq 1/2 - \delta$ for $\delta > 0$, note that if B is neither γ-evasive nor γ-divergent then

$$\Pr_{\mathcal{A}}[\mathcal{A}^M \text{ outputs an element of } B] \geq 1 - \gamma\,,$$

(associated with the distribution D_M), and hence

$$\Pr_{\mathcal{A}}[\mathcal{A}^{BM} \text{ outputs an element of } B] \geq 1 - 2\gamma \geq 2\delta\,.$$

Finally, note that the probability that B is $(1/2 - \delta)$-evasive or $(1/2 - \delta)$-divergent is no more than

$$\frac{1}{1/2 - \delta} \underbrace{\left[e^{-c\epsilon k^2} + 2k\sqrt{\epsilon}\right]}_{(\dagger)}\,.$$

Then it is clear that one can choose the constants δ and c, and the blinding probability $\epsilon = \Theta(1/k^2)$, so that this quantity is a constant bounded away from one. (For example, set $\delta = 1/6$. Then, with $\epsilon = 1/(144k^2)$ the second term of $(\dagger)$ above is no more than $1/6$; setting $c = 288$ guarantees the first term is likewise no more than $e^{-2} < 1/6$ and the entire expression is a constant less than one. One can achieve better constants with more care, but the quadratic dependence on ϵ in Theorem 2 dictates the quadratic gap between k and the number of forgeries achieved by this simple method of proof.)

Finally, we create a BU adversary for M by running the BZ adversary, blinded as above with $\epsilon = \Theta(1/k^2)$, and selecting one of the ck^2 output values at random.

B.3 Full measurement via random two-outcome measurements

Here we give the proof of Lemma 6, restated below. We remark that the constant c is likely to be improved, and it's not our intention to optimize it since we only need it in an imaginary hybrid game of a reduction proof.

Lemma 7. *Let $S_1, S_2, \ldots, S_{cn}$ be subsets of $\{0,1\}^n$ each of size 2^{n-1}, chosen independently and uniformly at random. Let Ξ_{S_j} denote the two-outcome measurement defined by S_j, and denote their composition $\tilde{\Xi} := \Xi_{S_{cn}} \circ \Xi_{S_{cn-1}} \circ \cdots \circ \Xi_{S_1}$. Let Ξ denote the full measurement in the computational basis. Then*

$$\Pr\left[\tilde{\Xi} = \Xi\right] \geq 1 - 2^{-\varepsilon n}\,,$$

whenever $c \geq 2 + \varepsilon$ with $\varepsilon > 0$.

Proof. We give a combinatorial proof. Consider an arbitrary mixed state of density matrix $\rho = (\rho_{x,y})_{x,y\in\{0,1\}^n}$, the full measurement Ξ on ρ gives

$$\Xi(\rho) = \sum_{x\in\{0,1\}^n} |x\rangle\langle x|\,\rho\,|x\rangle\langle x| = \sum_{x\in\{0,1\}^n} \rho_{x,x}\,|x\rangle\langle x|\,,$$

Given a set $S \subseteq \{0,1\}^n$, the projective measurement Ξ_S on ρ operates as

$$\begin{aligned}\Xi_S(\rho) &= \sum_{x,y\in S} |x\rangle\langle x|\,\rho\,|y\rangle\langle y| + \sum_{x,y\notin S} |x\rangle\langle x|\,\rho\,|y\rangle\langle y| \\ &= \sum_{x,y\in S} \rho_{x,y}\,|x\rangle\langle y| + \sum_{x,y\notin S} \rho_{x,y}\,|x\rangle\langle y|\,.\end{aligned}$$

Namely, Ξ_S will zero-out the entries $\rho_{x,y}$ in ρ, where $(x \in S, y \notin S)$ or $(x \notin S, y \in S)$. It is easy to verify that the same effect occurs when Ξ and Ξ_S are applied to a subsystem of a bipartite state.

Now, for any $c = 2 + \varepsilon$ with $\varepsilon > 0$, consider sampling $S_1, S_2, \ldots, S_{cn}$ independently at random, each of size 2^{n-1}, and define a few random events:

$$\begin{aligned} &E^i_{x,y} : x \in S_i \wedge y \in S_i, \text{ or } x \notin S_i \wedge y \notin S_i \,; \\ &E_{x,y} : \forall i \in \{1, \ldots, cn\} \text{ s.t. } E^i_{x,y} \,; \\ &\mathsf{BAD} : \exists x, y \in \{0,1\}^n, x \neq y \text{ s.t. } E_{x,y} \,. \end{aligned}$$

Observe that if BAD does not occur, it implies that for any $x \neq y$, the off-diagonal entry $\rho_{x,y}$ is eliminated by one of Ξ_{S_i}, and as a result $\tilde{\Xi} = \Xi_{S_{cn}} \circ \ldots \circ \Xi_{S_1}$ will be identical to Ξ.

Fix a pair (x, y) with $x \neq y$, clearly $\Pr[E^i_{x,y}] = 1/2$. Since each S_i is chosen independently,

$$\Pr[E_{x,y}] = \Pi_i \Pr[E^i_{x,y}] = 1/2^{cn} \,.$$

By the union bound,

$$\Pr[\mathsf{BAD}] \leq \binom{2^n}{2} \cdot \Pr[E_{x,y}] \leq 2^{2n}/2^{cn} = 2^{-\varepsilon n} \,.$$

Therefore we conclude that

$$\Pr[\tilde{\Xi} = \Xi] \geq \Pr[\tilde{\Xi} = \Xi \mid \overline{\mathsf{BAD}}] \cdot \Pr[\overline{\mathsf{BAD}}] \geq 1 - 2^{-\varepsilon n} \,.$$

□

B.4 Non-adaptive quantum queries and "double spending"

The following lemma shows that if there exists a non-adaptive quantum algorithm $\mathcal{A}$ making q queries to a function $f : \{0,1\}^n \to \{0,1\}^m$ that learns a certain property $p(f)$, then with inverse polynomial probability, there exists another non-adaptive q-query algorithm that learns $p(f)$ and q input-output-pairs with inverse polynomial probability. For this to hold, we need to assume that $\mathcal{A}$ makes its queries using a blank output register (i.e., initialized in the $|0\rangle$ state). This is the case, e.g., in period-finding and Simon's algorithm.

In the following, denote the set of n-bit-to-m-bit functions by $\mathcal{F}(n, m)$.

Lemma 8 (Double spending lemma)**.** *Let $F \subseteq \mathcal{F}(n,m)$ be a set of functions, P a set, $p : F \to P$ a function, and D a probability distribution on F. Suppose there exists a quantum query algorithm $\mathcal{A}$ which makes q non-adaptive quantum queries to $\mathcal{O}_f$ with blank output register for $f \leftarrow D$ and outputs $p(f)$ with $1/\mathsf{poly}(n)$ probability. Then there also exists an algorithm $\mathcal{A}'$ which makes q non-adaptive quantum queries to $\mathcal{O}_f$ for $f \leftarrow D$ and outputs both $p(f)$ and q input-output pairs of f with $1/\mathsf{poly}(n)$ probability.*

Proof. Let $\mathcal{X} = \{0,1\}^n$, $\mathcal{Y} = \{0,1\}^m$ and $\mathcal{H}_Z = \mathbb{C}^{\mathcal{Z}}$ for $Z = X, Y$. Set $\mathcal{A}^{\mathcal{O}}(1^n) = \mathcal{E}\left(\mathcal{O}^{\otimes q}|\psi\rangle_{X^q} \otimes |0\rangle_{Y^q}\right)$ where $|\psi\rangle$ is some input state and $\mathcal{E} = \{E_p\}_{p \in P}$ is a POVM on $\mathcal{H}_X^{\otimes q} \otimes \mathcal{H}_Y^{\otimes q}$ with outcomes labelled by the possible properties of f. Let $|\psi_1\rangle = \mathcal{O}^{\otimes q}|\psi\rangle_{X^q} \otimes |0\rangle_{Y^q}$. $\mathcal{A}$ outputs $p(f)$ with inverse polynomial probability, say with probability $p_{\text{succ}} = \langle\psi_1|E_{p(f)}|\psi_1\rangle$. It follows that the post-measurement state conditioned on the outcome $p(f)$,

$$|\psi_2^{p(f)}\rangle = \frac{\sqrt{E_{p(f)}}|\psi_1\rangle}{\sqrt{\langle\psi_1|E_{p(f)}|\psi_1\rangle}},$$

has inverse polynomial overlap with $|\psi_1\rangle$,

$$\begin{aligned} \left\langle \psi_1 \mid \psi_2^{p(f)} \right\rangle &= \frac{\langle\psi_1|\sqrt{E_{p(f)}}|\psi_1\rangle}{\sqrt{\langle\psi_1|E_{p(f)}|\psi_1\rangle}} \\ &\geq \sqrt{\langle\psi_1|E_{p(f)}|\psi_1\rangle} \end{aligned} \tag{52}$$

This implies immediately that measuring $|\psi_2^{p(f)}\rangle$ in the computational basis will yield q input output pairs of f with inverse polynomial success probability. □

We remark that the distribution of input-output pairs is at most $1 - 1/\mathsf{poly}(n)$ far from the distribution one would get by simply measuring immediately after the query of $\mathcal{A}$. This means that, in the case of period-finding and Simon's algorithm (where the queries are uniform), the input-output pairs will be distinct with non-negligible probability.

B.5 Alternative proof that random functions are BZ-secure

Using Lemma 2, we can give a simple proof of the fact that a random function is BZ-secure. Because of its simplicity, and because much of it can be reused to prove a separation between BZ and BU, we provide this proof below.

Theorem 19 ([5]). *An algorithm making q quantum queries to a random oracle $f : \{0,1\}^n \to \{0,1\}^m$ produces $q+1$ input-output pairs of f with probability at most*

$$\frac{2^{\lceil \log(q+1) \rceil}}{2^m}. \tag{53}$$

Note that the probability bound is within a factor of 2 of the one obtained in [5], and matches it for $q+1 = 2^k$, $k \in \mathbb{N}$.

Proof. Let $\mathcal{A}$ be an adversary that, when provided with the quantum random oracle f, outputs $q+1$ candidate input-output pairs. Formally, let $\rho_{(X,Y)^{q+1}F}$ be the joint cq-state of the adversary and the FO, where the classical registers $(X,Y)^{q+1}$ contain $\mathcal{A}$'s output and F is the FO's register. If we wanted to determine the success of $\mathcal{A}$ at this point, we would apply the Fourier transform to F, and then measure F and check if the outcome for F_{x_i} is y_i for each (x_i, y_i) output by $\mathcal{A}$.

Note that $P_l \rho = 0$ for all $l > q$ by Lemma 2, i.e., there are at most q entries of F that are nonzero. This implies that the entry corresponding to at least one of the inputs that $\mathcal{A}$ has output is, in fact, equal to 0^m. However, this is only true in superposition: different branches of the superposition may have different entries in the state $|0^m\rangle$. We will deal with this issue by thinking about a new algorithm $\mathcal{B}$, which will simulate the entire execution of $\mathcal{A}$ (including the oracle) and then perform a small number of additional measurements prior to the success check. The additional measurements will find a pair (x_{i_0}, y_{i_0}) in $(X,Y)^{q+1}$ such that $F_{x_{i_0}}$ is actually in the state $|0^m\rangle$ (in every branch of the superposition.) The probability that $y_{i_0} = f(x_{i_0})$ (in the execution of $\mathcal{B}$) will then be 2^{-m}. We will then apply Lemma 1 to show that the success probability of $\mathcal{A}$ is not much better.

We now describe $\mathcal{B}$ in detail. Initially, $\mathcal{B}$ simulates both $\mathcal{A}$ and the oracle. After $\mathcal{A}$ has finished, but before the success check is performed, $\mathcal{B}$ (which is in the state ρ) applies binary search to the $q+1$ inputs that $\mathcal{A}$ has output. The goal is to find an input x_{i_0} such that $F_{x_{i_0}}$ is in state $|0^m\rangle$. We do this using binary measurements that ask "are any of the registers $F_{x_{i_1}}, ..., F_{x_{i_k}}$ in the state $|0^m\rangle$?" We split up the set $S_0 = \{x_1, ..., x_{q+1}\}$ into two subsets $S_0^{\mathsf{L}} = \{x_1, ..., x_{\lfloor (q+1)/2 \rfloor}\}$ and $S_0^{\mathsf{R}} = \{x_{\lfloor (q+1)/2 \rfloor + 1}, ..., x_{q+1}\}$, and measure whether F_x is in a state different from $|0^n\rangle$ for all $x \in S_0^{\mathsf{L}}$. This is done using the binary measurement given by

$$P_1 = (\mathbb{1} - |0\rangle\langle 0|)_{x_1} \otimes ... \otimes (\mathbb{1} - |0\rangle\langle 0|)_{x_{\lfloor (q+1)/2 \rfloor}} \otimes \mathbb{1}^{\otimes (2^n - \lfloor (q+1)/2 \rfloor)} \tag{54}$$

and it's complementary projector $P_0 = \mathbb{1} - P_1$. If the outcome is no, we set $S_1 = S_0^{\mathsf{L}}$, if it is yes then we set $S_1 = S_0^{\mathsf{R}}$. This makes sure that we continue with a set that contains an input such that the corresponding FO register is in state $|0^n\rangle$. Now we repeat the described steps using S_1 in place of S_0 and continue recursively until we encounter a set S_l with only one element, say w. Continuing with the success check, we now know that F_w is in the state $|0^m\rangle$, which implies that $f(w)$ is uniformly random and independent of $\mathcal{A}$'s output. Indeed, a register that is in a pure state is automatically in product with the rest of the universe, and $f(w)$ is determined by applying $H^{\otimes m}$, which transforms $|0^m\rangle$ into $|\phi_0\rangle$, and measuring, which yields a uniformly random outcome. Therefore $\mathcal{A}$'s success probability is at most 2^{-m}. The total number of binary measurements for the binary search procedure is upper-bounded by $\lceil \log(q+1) \rceil$, so an application of Lemma 1 finishes the proof. □

B.6 Hash-and-MAC with Bernoulli-preserving hash

Recall that, given a MAC $\Pi = (\mathsf{Mac}_k, \mathsf{Ver}_k)$ with message set X and a function $h : Z \to X$, there is a MAC $\Pi^h := (\mathsf{Mac}_k^h, \mathsf{Ver}_k^h)$ with message set Z defined by $\mathsf{Mac}_k^h = \mathsf{Mac}_k \circ h$ and $\mathsf{Ver}_k^h(m,t) = \mathsf{Ver}_k(h(m),t)$. This is the standard, so-called "Hash-and-MAC" construction.

Theorem 20. *Let $\Pi = (\mathsf{Mac}_k, \mathsf{Ver}_k)$ be a BU-secure MAC with $\mathsf{Mac}_k : X \to Y$, and let $h : Z \to X$ a Bernoulli-preserving hash. Then Π^h is a BU-secure MAC.*

Proof. Let $\mathcal{A}$ be an adversary against Π^h. We build an adversary $\mathcal{A}_0$ against Π which (given oracle $f : X \to Y$) runs $\mathcal{A}$ and answers its queries with $f \circ h$, i.e., $|m\rangle|t\rangle \mapsto |m\rangle|t \oplus f(h(m))\rangle$. This can be implemented by first computing h into an extra register, then invoking the oracle, and then uncomputing h. When $\mathcal{A}$ produces its final output (m,t), $\mathcal{A}_0$ outputs $(h(m),t)$ and terminates. We claim that $\left|\Pr[\mathsf{BlindForge}_{\mathcal{A},\Pi^h}(n,\epsilon)=1] - \Pr[\mathsf{BlindForge}_{\mathcal{A}_0,\Pi}(n,\epsilon)=1]\right| \leq \mathsf{negl}(n)$. Since the right-hand-side of the difference above is negligible by BU-security of Π, establishing the claim will finish the proof.

We prove the claim by showing that the difference can be viewed as the success probability of a distinguisher $\mathcal{D}$ against the Bernoulli-preserving property of h. The distinguisher $\mathcal{D}$ receives an oracle for χ_B (where $B \subseteq Z$ is sampled according to either $\mathcal{B}_\epsilon$ or $\mathcal{B}_\epsilon^h$) and proceeds as follows:

1. generate a key k for Π;
2. run $\mathcal{A}$, answering its oracle queries with $|m\rangle|t\rangle \mapsto |m\rangle|t\rangle|\chi_B(m)\rangle|\mathsf{Mac}_k(h(m))\rangle$ $\mapsto |m\rangle|t \oplus \chi_B(m)\cdot\mathsf{Mac}_k(h(m))\rangle|\chi_B(m)\rangle$ where we invoked the oracle in the first step and CCNOT in the second.
3. when $\mathcal{A}$ outputs (m,t), compute $b = \mathsf{Ver}_k^h(m,t) = \mathsf{Ver}_k(h(m),t)$. Query the oracle to compute $b' = \chi_B(m)$, and output $b \wedge b'$.

It now remains to check that (i.) if B was sampled according to $\mathcal{B}_\epsilon$ (i.e., uniform blinding), then $\mathcal{D}$ is simulating the game $\mathsf{BlindForge}_{\mathcal{A},\Pi^h}(n,\epsilon)$, and (ii.) If B was sampled according to $\mathcal{B}_\epsilon^h$ (i.e., hash-blinding), then $\mathcal{D}$ is simulating the game $\mathsf{BlindForge}_{\mathcal{A}_0,\Pi}(n,\epsilon)$. Fact (i.) follows directly from the definition[3] of the $\mathsf{BlindForge}$ game. To see fact (ii.), observe that the $\mathsf{BlindForge}$ game against $\mathcal{A}_0$ samples a uniform blinding set $C_\epsilon \subseteq X$ and executes algorithm $\mathcal{A}$ with oracle

$$m \longmapsto \chi_{C_\epsilon}(h(m)) \cdot \mathsf{Mac}_k(h(m)) = \chi_{B_\epsilon^h}(m) \cdot \mathsf{Mac}_k(h(m))\,,$$

precisely as in the execution of $\mathcal{A}$ by $\mathcal{D}$. □

B.7 Even more on Bernoulli-preserving hash

Recall that blinding a function $f : \{0,1\}^n \to \{0,1\}^t$ on a set $B \subseteq \{0,1\}^n$ results in the blinded function Bf defined by $Bf(x) = \perp = (0^t, 1)$ for $x \in B$ and $Bf(x) = (f(x), 0)$ for $x \notin B$.

Lemma 9. *Let $h : \{0,1\}^n \to \{0,1\}^m$ be a Bernoulli-preserving hash and $f : \{0,1\}^n \to \{0,1\}^t$ an efficiently computable function. Then for all oracle QPTs $(\mathcal{A},\epsilon)$, we have*

$$\left| \Pr_{B\leftarrow\mathcal{O}_\epsilon}\left[\mathcal{A}^{Bf}(1^n)=1\right] - \Pr_{B\leftarrow\mathcal{O}_\epsilon^h}\left[\mathcal{A}^{Bf}(1^n)=1\right] \right| \leq \mathsf{negl}(n)\ .$$

[3]Note that we have again used the convention that the blinding symbol $\perp$ is the string $0\ldots01$; in our case, the final bit corresponds to the register containing $\chi_B(m)$. If one chooses a different convention, it may be necessary to adjust $\mathcal{D}$ to uncompute that register with an extra call to the oracle.

Proof. It suffices to observe that one can simulate the oracle for Bf using two calls to an oracle for χ_B and two executions of f, as follows.

$$\begin{aligned}|x\rangle|y\rangle|b\rangle &\mapsto |x\rangle|y\rangle|b\rangle|\chi_B(x)\rangle|f(x)\rangle \\ &\mapsto |x\rangle|y \oplus \chi_B(x)\cdot f(x)\rangle|b \oplus \chi_B(x)\rangle|\chi_B(x)\rangle|f(x)\rangle \\ &\mapsto |x\rangle|y \oplus \chi_B(x)\cdot f(x)\rangle|b \oplus \chi_B(x)\rangle \\ &= |x\rangle|y \oplus Bf(x)\rangle\end{aligned}$$

In the second step, we applied the CCNOT (Toffoli) gate to the second register, with the fourth and fifth register as the controls and a CNOT to the third register with the fourth register as a control. With this observation, it is straightforward to turn any distinguisher for $B_\epsilon f$ vs. $B^h_\epsilon f$ into one for χ_{B_ϵ} vs. $\chi_{B^h_\epsilon}$. □

Finally, we show the Bernoulli-preserving hash property for the hash from Section B.

Theorem 21. *H is Bernoulli-preserving hash if LWE holds against any efficient quantum distinguisher.*

Proof. We proceed in three steps (with help of Lemma 10 below):

1) Since F_s is injective under an injective key, it is clearly Bernoulli-preserving hash. As a result, $F_s, s \leftarrow \mathcal{D}_{los}$ must be Bernoulli-preserving hash too, because a lossy key is indistinguishable from an injective key by definition.

2) Then g_k is chosen properly so that it is injective when restricted to $im(F_s)$ of lossy key s. Therefore g_k is Bernoulli-preserving hash too.

3) Finally, $H_{k,s}$ is Bernoulli-preserving hash by the composition of Bernoulli-preserving hash functions g_k and F_s.

□

Lemma 10. *Any injective function f is Bernoulli-preserving hash. Given any Bernoulli-preserving hash $f : X \to Y$ and $g : Y \to Z$ that is Bernoulli-preserving hash on $im(f)$, then $h = g \circ f$ is also Bernoulli-preserving hash.*

Proof. The first part follows by observing that a ε-random subset in the codomain corresponds exactly to a ε-random subset in the domain under inverse of the function. Let $\mathcal{O} \approx \mathcal{O}'$ denote that two oracles $\mathcal{O}$ and $\mathcal{O}$ are indistinguishable by any quantum poly-time algorithm. For the second part, we need to show that $\chi_{C:C\leftarrow_\varepsilon X} \approx \chi_{C:C=h^{-1}(C_Z),C_Z\leftarrow_\varepsilon Z}$, where $\leftarrow_\varepsilon$ indicates sampling a random subset of fraction ε. Since f is Bernoulli-preserving hash, we have that

$$\chi_{C:C\leftarrow_\varepsilon X} \approx \chi_{C:C=f^{-1}(C_Y),C_Y\leftarrow_\varepsilon Y} \equiv \chi_{C:C=f^{-1}(C'_Y),C'_Y\leftarrow_\varepsilon im(f)}\,.$$

The second equivalence holds by observing that for any $C_Y \subseteq Y$, $f^{-1}(C_Y) = f^{-1}(C_Y \cap im(f))$. Then because g is Bernoulli-preserving on $im(f)$,

$$\chi_{C'_Y:C'_Y\leftarrow_\varepsilon im(f)} \approx \chi_{C'_Y:C'_Y=g^{-1}(C_Z),C_Z\leftarrow_\varepsilon Z}\,.$$

Therefore, we conclude that

$$\chi_{C:C\leftarrow_\varepsilon X} \approx \chi_{C:C=f^{-1}(g^{-1}(C_Z)),C_z\leftarrow_\varepsilon Z} = \chi_{C:C=h^{-1}(C_Z),C_z\leftarrow_\varepsilon Z}\,.$$

□