6.845 PROJECT: CLASSICAL CRYPTOGRAPHY, QUANTUM QUERIES

EMILY STARK

1. Introduction

Post-quantum cryptography is the study of cryptographic primitives that remain secure against quantum adversaries. In this project, I survey several recent papers in this field that study the security of classical constructions against adversaries that make quantum oracle queries. One set of results studies the *quantum random oracle* model, where a quantum adversary can query a random oracle on a superposition of inputs. Another set of results shows how to prove security for cryptographic primitives when the adversary is allowed to query the constructed primitive on a superposition of inputs; specifically, I review recent proofs for quantum-secure pseudorandom functions and message authentication codes.

The rest of this project report is organized as follows. First I briefly present cryptographic definitions that I use later. In this project, I survey four recent papers [4, 9, 8, 5], but instead of presenting each paper individually, I instead group several interesting results from the papers under the broad categories of quantum random oracles (Section 3) and quantum queries to cryptographic primitives (Section 4). Each of these sections also contains separation results showing that quantum queries can break existing algorithms.

2. Background

In this section, I briefly explain the notation and cryptography definitions that will be used in the rest of this paper. I assume the reader is familiar with far more quantum computation and quantum complexity theory than I am, so this section includes only some informal definitions of the cryptographic primitives that are used later. Formal definitions for most terms can be found in [2].

Definition A *near-collision resistant hash function* is a pair $(\mathsf{Gen}, \mathsf{H})$ such that for any efficient algorithm A, for $k \leftarrow \mathsf{Gen}$, for constant l between 1 and n, A cannot produce a $l-$*near collision* with non-negligible probability. A $l-$*near collision* is a pair x, y where $x \neq y$ such that the first l bits of $H(k, x)$ and $H(k, y)$ are the same.

A quantum algorithm can find $l-$near collisions in time $2^{l/3}$ using Grover's algorithm (where the function values are truncated to l bits). A classical algorithm is bounded by the birthday attack.

Definition An *identification scheme* is a protocol $(\mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$. An identification scheme is usually used to prove knowledge of a secret key corresponding to some public key, but for this project it can be thought of as any interactive protocol with completeness and soundness requirements.

Definition A *pseudorandom function family* (PRF) is a family of functions from $\mathcal{K} \times \mathcal{X}$ to $\mathcal{Y}$. Classical, a PRF is secure if an efficient algorithm given oracle access to

either $\mathsf{PRF}(k, \cdot)$ for a random k, or to a random function, cannot determine which with non-negligible probability.

In the rest of this paper, I usually use "standard secure" or "classically secure" PRF to refer a PRF that is secure against a quantum adversary making classical oracle queries, whereas "quantum PRF" refers to an adversary making quantum oracle queries.

Definition A *claw-free permutation pair* is a pair $(\mathcal{F}_1, \mathcal{F}_2)$, where $\mathcal{F}_i = (G, f_i, f_i^{-1})$. G is a key generation algorithm, and each f_i is a trapdoor permutation. No efficient algorithm can find x, y such that $f_1(x) = f_2(y)$ (a claw).

Definition An *identity based encryption scheme* is a cryptosystem with a set of identities, a master public/secret key pair, an extraction algorithm that generates a secret key given an identity and the master secret key, an encryption algorithm that encrypts messages to identities, and a decryption algorithm that decrypts a message given the secret key for the identity that it was encrypted to.

Definition A *pre-image sampleable function* (PSF) is a tuple $(\mathsf{Gen}, \mathsf{Sample}, f, f^-1)$. Let D be a distribution on the domain of f. Then Gen outputs pk, sk, Sample samples x from D such that $f_{pk}(x)$ is uniform, and $f_{sk}^{-1}(y)$ samples x from D conditioned on $f_{pk}(x) = y$. The PSF is secure if no efficient algorithm can produce x given $f_{pk}(x)$.

3. The quantum random oracle model

In the quantum random oracle model, studied by Boneh et al. [4], all algorithms are given access to a random function H, but the adversary is allowed to query H on quantum states. Previous work considered quantum adversaries who could only make classical random oracle queries, but in practice, the random oracle would be instantiated with a real function with a succinct description, which the adversary could easily implement on his own and evaluate on quantum states.

In this section, I explain a sketch of a separation result for this setting, a general proof technique for signatures in the quantum random oracle model, and a quantum random oracle security proof for an identity-based encryption scheme based on lattices.

3.1. Separation result.

Boneh et al. [4] exhibit a protocol that is secure against classical adversaries, secure against quantum adversaries making classical random oracle queries, and *insecure* against quantum adversaries making quantum oracle queries. Intuitively, the construction is to start with a quantum-secure identification scheme, and prepend it with a collision-finding stage that tests the prover's ability to find collisions in limited time. Then, the verifier accepts if the prover found enough collisions in time, or if the identification scheme accepted, so that a quantum adversary can cause the verifier to accept regardless of the outcome of the original protocol.

Construction 3.1 (Quantum-insecure IS)**.** *Let $\mathsf{IS} = (\mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ be an identification scheme that is secure against quantum adversaries. Let $H = (\mathsf{H.KeyGen}, \mathsf{H.Eval})$ be a family of near collision resistant hash functions. Let $(\mathsf{KeyGen'}, \mathsf{P'}, \mathsf{V'})$ be the identification scheme defined as follows:*

- $\mathsf{KeyGen'}(1^n)$*: Ouptut* $\mathsf{KeyGen}(1^n)$.
- $(\mathsf{P'}(pk, sk, l), \mathsf{V'}(pk, l))$*:*

(1) *V′ initializes a collision counter to* 0.
(2) *For* $i = 1, ..., r$, *V′ computes* $k_i \leftarrow \mathsf{H.KeyGen}(1^n)$ *and sends* k_i *to P′.*
(3) *While V′ computes* $\mathsf{H.Eval}(k_i, c)$ *for* $c = 1, 2, ..., \sqrt[3]{2^l}$, *P′ searches for a* l*-near collision in* $\mathsf{H.Eval}(k_i, \cdot)$.
(4) *If V′ receives a valid collision from P′ before* $c = \sqrt[3]{2^l}$, *then V′ increments the collision counter.*
(5) *V′ and P′ run V and P as subroutines.*
(6) *V′ accepts if the collision counter is at least* $r/4$ *or if V accepted, and rejects otherwise.*

Theorem 3.2. $\mathsf{IS'} = (\mathsf{KeyGen'}, \mathsf{P'}, \mathsf{V'})$ *is complete and sound against classical provers and quantum provers with classical query access to* H, *if* IS *is complete and sound against classical and quantum provers.* IS′ *is complete, but not sound against quantum provers with quantum query access to* H.

Proof sketch Completeness for all types of provers follows from the completeness of the original protocol: any classical or quantum prover can run P honestly to cause V′ to accept.

Soundness against classical provers and quantum provers with classical queries follows from known collision-finding bounds due to the birthday attack. In time $\sqrt[3]{2^l}$, a prover P^* cannot make $\sqrt{2^l}$ oracle queries and therefore has probability less than $\frac{1}{2}$ of finding a $l-$near collision in time. Since each collision-finding round uses a new key for H, one can apply the Chernoff bound to show that, for r rounds where r is polynomial in the security parameter n, P^* has only negligible probability of finding $r/4$ collisions. Thus, in the case of classical provers and quantum provers with classical query access, the soundness of IS′ reduces to the soundness of IS.

Now one must show that a quantum P^* with quantum query access to H can cause V^* to accept even when the sub-protocol (V, P) rejects. P^* applies known quantum collision finding algorithms such as [1] and finds a $l-$near collision with probability greater than $\frac{1}{2}$. The Chernoff bound then says that P^*'s probability of finding fewer than $r/4$ collisions is negligible. Therefore a quantum prover with quantum query access can cause V′ to accept with overwhelming probability. □

Another way of viewing this result is that, while the protocol IS′ is secure against classical and quantum adversaries when H is modeled as a random oracle, *any* instantiation of H makes the protocol insecure against quantum provers, since a quantum prover can implement H as a quantum circuit.

3.2. **History free reductions.** In this section, I describe a general technique, history free reductions, that Boneh et al. give for proving that signature schemes are secure in the quantum random oracle model. This technique says that if the classical proof of a signature scheme has certain characteristics, and the proof reduces the security of the scheme to a problem that is difficult for quantum computers, then the scheme is also secure in the quantum random oracle model. Informally, the reduction used in the original proof must answer an adversary's oracle queries independently of previous queries. This property allows the classical adversary in the reduction to be replaced with a quantum adversary making quantum queries, since answers to each basis state in the quantum query can be computed independently. Loosely speaking, it is okay in classical settings for the answers to queries to depend on previous queries because there are only a polynomial number of them.

In quantum settings, there could be in some sense an exponential number of previous queries, and so it is not clear that such a reduction can be translated into an efficient algorithm, even quantum.

History free reduction Let $\mathsf{S} = (\mathsf{G}, \mathsf{Sign}^O, \mathsf{Verify}^O)$ be a signature scheme in the random oracle model. Consider a proof of security for S that uses a classical adversary A for S to construct a classical algorithm B that solves some hard problem. The construction of algorithm B is a *history free reduction* if B contains classical algorithms $START^P, RAND^{O_c}, SIGN^{O_c,P}$, and $FINISH^{O_c,P}$, where P is an oracle provided by the challenger for the hard problem, and O_c is a classical random oracle, with the following properties:

(1) On an instance x of the hard problem, B runs $START^P(x)$, which outputs a public key pk for S and possibly some private state z to be used by B.
(2) If A queries the random oracle at r, B responds with $RAND^{O_c}(r, z)$.
(3) If A requests a signature on m, B responds with $SIGN^{O_c,P}(m, z)$.
(4) When A outputs an attempted forgery (m, σ), B outputs $FINISH^{O_c,P}(m, \sigma, z)$.
(5) There is an efficient algorithm to compute an instance x of the hard problem, given a public key pk for S. If pk is generated by G, then this algorithm must produce an instance of the hard problem distributed according to the distribution that the challenger for the hard problem uses.
(6) Let O_q be the quantum oracle that transforms $|x, y\rangle$ to $|x, y \oplus RAND^{O_c}(r, z)\rangle$. O_q must be computationally indistinguishable from a random oracle against quantum distinguishers.
(7) $SIGN^{O_c,P}$ can abort, causing B to abort, but it must be the case that, with non-negligble probability, all runs of $SIGN^{O_c,P}$ do not abort. If $SIGN^{O_c,P}$ does not abort, then the outputted signature must be a valid signature relative to the oracle $RAND^{O_c}(\cdot, z)$, and the distribution of the outputted signatures must be negligibly close to Sign.
(8) If A outputs a valid signature forgery for pk with oracle $RAND^{O_c}(\cdot, z)$, then $FINISH^{O_c,P}$ is an output that causes the challenger for the hard problem to output 1 with non-negligible probability.

Theorem 3.3. *If a signature scheme $\mathsf{S} = (\mathsf{G}, \mathsf{Sign}^O, \mathsf{Verify}^O)$ has a history free reduction for a problem that is hard for quantum computers, then S is secure in the quantum random oracle model, assuming the existence of quantum pseudorandom functions.*

Proof sketch Let A be a classical PPT adversary for S. Let B be an adversary for the hard problem that uses A. Let B be part of a history free reduction, so that B contains the algorithms $START, RAND, SIGN$, and $FINISH$.

Now let A_Q be a quantum adversary who outputs a valid forgery for S with non-negligible probability ϵ. The proof proceeds by making a series of modifications to A_Q's challenger, without affecting A_Q's success probability, culminating in a challenger who also acts as an adversary for the hard problem.

We begin replacing parts of A_Q's challenger with components from the history free reduction. The properties of the history free reduction ensure that these replacements only negligibly affect A_Q's output distribution. The first modification is that the challenger computes an instance x of the hard problem from the generated public key (which is efficiently computable by Property 5 of the history free reduction), and then runs $START^P(x)$ to obtain the private state. When A_Q

queries the random oracle, the challenger answers with the oracle that maps $|x, y\rangle$ to $|x, y \oplus RAND^{O_Q}(x, z)\rangle$, where O_Q is a quantum random oracle. By Property 6 of the history free reduction, this change does not affect A_Q's output distribution non-negligibly, since this oracle is indistinguishable from a truly random oracle.

Next, the challenger's key generation is replaced by an instance x received from a challenger for the hard problem, so that A_Q receives the public key outputted by $START^P(x)$. This does not affect A_Q's behavior because the distribution of x instances when computed from public keys is negligibly close to the distribution produced by the challenger for the hard problem. Next, the challenger is modified to answer A_Q's signature queries with $SIGN^{O_Q,P}$ where O_Q is a quantum random oracle. By Property 7, if none of the $SIGN$ runs abort, then using $SIGN$ does not affect A_Q's output distribution, and all the $SIGN$ runs complete with non-negligible probability. Therefore A_Q still outputs a valid forgery with non-negligible probability in this scenario.

Finally, replace the quantum random oracle with a quantum PRF. This challenger first generates a key for the PRF, and then answers random oracle queries with it. Since the PRF is indistinguishable from a random function, even against quantum adversaries making quantum queries, this modification does not affect A_Q's behavior.

Now we have a challenger who can act as an adversary for the hard problem. Let this algorithm be B_Q. Given an instance x, B_Q generates a public key and runs A_Q, answering all queries as in as described. If B_Q does not abort (which happens with non-negligible proability), then when A_Q outputs a forgery (m, σ), B_Q runs $FINISH$ and outputs the result, thereby breaking the hard problem with non-negligible probability by Property 8 of the history free reduction. □

3.2.1. *An example.* Of course, after all this, the history free reduction technique is only useful if there are classical signature schemes that have such reductions! Boneh et al. give several examples of such schemes, and here I explain one of their examples, a signature scheme built from a pair of claw-free permutations. The proof of this scheme follows a proof given by Coron [6].

Construction 3.4 (Signature scheme from claw-free permutations)**.** *Let $(\mathcal{F}_1, \mathcal{F}_2)$ be a pair of claw-free permutations, where $\mathcal{F}_i = (G, f_i, f_i^{-1})$. The signature scheme $\mathsf{S} = (\mathsf{G}, \mathsf{Sign}^O, \mathsf{Verify}^O)$ is as folows.*

- $\mathsf{G}(1^n)$ *runs G and outputs the result.*
- $\mathsf{Sign}^O(sk, m) = f_1^{-1}(sk, O(m))$
- $\mathsf{Verify}^O(pk, m, \sigma) = 1$ *if and only if* $f_1(pk, \sigma) = O(m)$.

A natural proof of security for this scheme turns out to be a history free reduction. The intuition is as follows. Randomly choose a small but non-negligible fraction of the message space and designate those messages as "unlucky." When the adversary wants to evaluate the random oracle on an unlucky point, we answer by evaluating f_2 at a random point, and on all other queries we answer by evaluating f_1 at a random point. (Note that computing this answer does not rely on any previous queries, which is crucial for the reduction to be history free.) If the adversary tries to sign an unlucky message, we abort, but otherwise we answer with a valid signature relative to the random oracle that is being simulated for the adversary. The adversary cannot tell lucky messages from unlucky, so with some non-negligible

probability, the adversary outputs a valid forgery for an unlucky message with respect to the simulated random oracle, thereby revealing a claw.

Below I explain this result more formally and argue that the reduction is history free.

Theorem 3.5. *S has a history free reduction to the problem of finding a claw for $(\mathcal{F}_1, \mathcal{F}_2)$.*

Proof sketch Let A be an efficient classical adversary that breaks S with non-negligible probability ϵ. An algorithm B that uses A to find a claw for $(\mathcal{F}_1, \mathcal{F}_2)$ uses the following four subprocedures. The subprocedures can query a classical random oracle O that outputs pairs (a, b), where a is in the domain of $\mathcal{F}_i$ and b is in $\{1, ..., p\}$, where p is the maximum number of signing queries that A makes.

- $START(pk)$ outputs (pk, pk).
- $RAND^O(r, pk)$ queries for $(a, b) \leftarrow O(r)$, and it returns $f_2(pk, a)$ if $b = 1$ and $f_1(pk, a)$ otherwise.
- $SIGN^O(m, pk)$ queries for $(a, b) \leftarrow O(m)$ and returns a if $b \neq 1$. If $b = 1$, then $SIGN$ aborts.
- $FINISH^O(m, \sigma, pk)$ queries for $(a, b) \leftarrow O(m)$ and outputs (σ, a) as a candidate claw.

Let $(a, b) = O(m)$. First, note that on a particular signing query, B only aborts with probability $1/p$, where p is the number of signing queries, so overall B does not abort with non-negligible probability. If B does not abort, and if (m, σ) is a valid forgery with respect to the oracle $RAND^O$, then (σ, a) is a claw with non-negligible probability. This is because, if $b = 1$, then $f_1(pk, \sigma) = RAND^O(m) = f_2(pk, a)$, so the pair forms a claw. What, then, is the probability that $b = 1$? The only way that A might have "chosen" the distribution of b would have been by querying the random oracle at m, since if A had queried for a signature on m the forgery would not be considered valid. But for a randomly chosen r, A cannot distinguish between $f_1(pk, r)$ and $f_2(pk, r)$ because f_1 and f_2 have the same domain and range. Therefore, the probability that $b = 1$ is always $1/p$, which is non-negligible. Overall, since B does not abort with non-negligible probability, A outputs a valid forgery with non-negligible probability, and $b = 1$ with non-negligible probability, B breaks the claw-freeness of $(\mathcal{F}_1, \mathcal{F}_2)$.

Now it remains to show that B constitutes a history free reduction, by checking that the subalgorithms satisfy the properties from the definition. First, a public key for the signature scheme is chosen exactly according to the key generation for the pair of claw-free permutations, so it is trivial to obtain a properly distributed instance of the claw-free problem. $RAND$ always evaluates a permutation at a random point, so it returns uniformly independent random values; this satisfies Property 6 from the history free definition. To satisfy Property 7, note that if $SIGN$ does not abort, then it outputs a preimage of $RAND^O(m)$, so it is a valid signature with respect to the $RAND^O$ oracle. Further, $SIGN$ only aborts with probability $1/p$, so there is a non-negligible probability that none of the $SIGN$ runs aborts. Finally, as argued above, if B does not abort and A outputs a valid forgery, then $FINISH$ outputs a claw for $(\mathcal{F}_1, \mathcal{F}_2)$. Thus, this reduction is history free. □

Since S has a history free reduction to the problem of breaking a pair of claw-free permutations, S is secure in the quantum random oracle model, assuming the existence of quantum-secure claw free permutations.

3.3. **IBE in the quantum random oracle model.** History free reductions give a convenient method of proof for signature schemes in the quantum random oracle model. In contrast, no such general technique has been discovered yet for encryption schemes. Boneh et al. mention without much elaboration that it is "considerably more complicated" to define a history free reduction for encryption, and then they go on to directly prove quantum CCA security for a public key encryption scheme. One reason for the complication is that CCA security proofs for public key schemes often involve not one but two hard problems (such as breaking a trapdoor function or a symmetric key encryption scheme that is used in the construction), so the proofs do not fall into clean reduction "templates" as many signature proofs do.

A later result for identity based encryption by Zhandry [9] illuminates another reason why it is difficult to define history free reductions for encryption schemes: reductions for encryption schemes often simulate random oracles for which it is not obvious that the oracle simulated for the adversary is quantum computationally indistinguishable from a truly random oracle. In fact, in Zhandry's result, the simulated oracle *is* distinguishable from random, but one can bound the amount to which it influences the adversary's outputs. In this section, I outline the proof for an identity based encryption scheme given by Zhandry, choosing to focus on this proof rather than the Boneh et al. proof of the public key scheme, because the former involves a more general insight about quantum algorithms.

Zhandry proves the security of the following construction based on lattices, which was first proposed by Gentry et al. [7]

Construction 3.6 (Quantum secure IBE)**.** *Let* $\mathsf{F} = (\mathsf{F.Gen}, \mathsf{F.Sample}, f, f^{-1})$ *be a pre-image sampleable function. Let* $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *be a CPA secure encryption scheme against quantum adversaries, such that* $\mathsf{E.Gen}(1^n)$ *generates* $(msk, mpk) \leftarrow \mathsf{F.Gen}(1^n)$*, samples* $sk \leftarrow \mathsf{F.Sample}(1^n)$*, computes* $pk = f_{mpk}(sk)$*, and outputs* $(sk, (pk, mpk))$*. (The desired property is that the trapdoor* f^{-1}_{msk} *allows computation of secret keys from public keys, which will be used to extract secret keys for identities.) Let* H *be a random oracle mapping identities to public keys for* E*.*

The IBE scheme is $\mathsf{IBE} = (\mathsf{IBE.Gen}, \mathsf{IBE.Extract}^H, \mathsf{IBE.Enc}^H, \mathsf{IBE.Dec})$*, defined as follows:*

- $\mathsf{IBE.Gen}(1^n)$ *runs* $\mathsf{F.Gen}(1^n)$ *and outputs the result.*
- $\mathsf{IBE.Extract}^H_{msk}(id) = f^{-1}_{msk}(H(id))$
- $\mathsf{IBE.Enc}^H(id, m) = \mathsf{E.Enc}(H(id), m)$
- $\mathsf{IBE.Dec}(sk_{id}, c) = \mathsf{E.Dec}(sk_{id}, c)$

The idea of the proof to make a quantum analogue of the classical proof strategy. The classical proof takes the challenge and "hides" it in the answer to a single of the adversary's random oracle queries, say the query for id. Since the adversary has no idea which query the challenge is hidden in, there is a non-negligible probability that the adversary will not perform an extract query for id, but will choose to attack id, in which case the adversary has solved some hard problem for us. In the quantum setting, of course hiding the challenge in a single query will not give a non-negligible probability that the adversary uses that query to break the cryptosystem. Instead, the quantum analogue is to hide the challenge in a

fraction of the oracle queries: the resulting oracle distribution will indeed give a non-negligible probability that the adversary breaking the cryptosystem solves the challenge for us, but without affecting the adversary's probability of success too much. Zhandry's proof of security in the quantum random oracle model roughly follows 3 steps.

(1) Let Q be a quantum algorithm with access to a quantum oracle H drawn from some distribution D, such that Q makes q queries to H. Zhandry shows that Q's probability of outputting a particular value can be expressed as a linear combination of the probabilities that H outputs a particular set of outputs on a particular set of $2q$ inputs.
(2) Let SC_λ be the oracle distribution where, for H drawn from SC_λ, there is a random "distinguished" value y such that $H(x) = y$ with probability λ, and with probability $1 - \lambda$, $H(x)$ is chosen uniformly at random. Zhandry uses the previous result to show that if the random oracle queries of a successful IBE adversary are answered with an oracle drawn from SC_λ, then the adversary's probability of breaking the scheme remains non-negligible.
(3) Finally, the security of IBE is reduced to the security of E by using a challenge public key as the "distinguished" value y in the oracle H. Zhandry shows that, with non-negligible probability, the adversary chooses to attack some id^* such that $H(id^*) = pk$, and yet never queries for any other id such that $H(id) = pk$. This allows the construction of an adversary for E.

In this paper, I give a formal statement and proof sketch for Step 1, but only briefly explain the other two steps. (Step 1 is used in a later section to help prove the security of a quantum PRF.) To prove Step 2, Zhandry uses Step 1 to rewrite the adversary's probability of winning in terms of the random oracle distribution SC_λ. Some rearranging of terms, basic calculus, and the Markov brother's inequality are then used to show that the adversary wins with advantage $\epsilon\lambda - \frac{l(q)}{4}\lambda^2$, where l is a polynomial in the number of adversary queries and ϵ is the adversary's advantage with a truly random oracle. Finally, with these two steps completed, a straightforward reduction (which resembles the classical proof of security) shows that an adversary with non-negligible advantage for IBE also has non-negligible advantage for breaking E.

Below I provide more detail on Step 1 of the proof.

Theorem 3.7. *Let A be a quantum algorithm make q quantum queries to an oracle $H : \mathcal{X} \to \mathcal{Y}$ drawn from some distribution D. For every possible output z of A, $Pr_{H \leftarrow D}[A^H() = z]$ is a linear combination of elements from the set*

$$\{Pr_{H \leftarrow D}[H(x_i) = r_i \forall i \in 1, ..., 2q] : x_1, ..., x_{2q} \in \mathcal{X}, y_1, ..., y_q \in \mathcal{Y}\}.$$

Proof sketch The actual proof has some frightening symbol-pushing, so I attempt to give a higher-level sketch. Consider the density matrix of the algorithm A after making q oracle queries. This density matrix can be written as a sum, over all possible oracles H, of the state of the algorithm in the case that its oracle queries were answered by H. This sum is weighted by the probability that H was drawn as A's oracle.

Given a particular oracle, the state after q queries can be written as an alternating product of unitaries and quantum queries starting on some initial state. The resulting density matrix is a large weighted sum of matrices, and a particular component of the density matrix is a large weighted sum of products of components of

the unitaries (and their inverses). In examining a single component of the density matrix, each summation runs over all possible basis states. Crucially, one term of the sum only evaluates H at $2q$ points: one term is a product of q unitary components and q unitary inverse components. Each oracle query affects which component of the unitary is used in the next outermost multiplicand. This means that we can reweight the sum; instead of weighting each term by $Pr_{H\leftarrow D}[H]$, we can weight the term by the probability that H outputs any $2q$ values for its inputs in this term, and sum over all possible $2q$ outputs. This gives a sum of products (which don't depend on H), weighted by $Pr_{H\leftarrow D}[H(x_i) = r_i, H(x'_i) = r'_i \forall i \in 1, ..., q]$, as desired.

In more mathematical terms, for an algorithm with access to an oracle drawn from H, a particular component xyz of the density matrix after q queries can be written after some manipulation as

$$\rho_{xyz} =$$

$$\sum_H Pr_D[H] \cdot$$

$$\sum_{x_q y_q z_q} \sum_{x'_q y'_q z'_q} ... \sum_{x_1 y_1 z_1} \sum_{x'_1 y'_1 z'_1} U_{xyzx_q y_q \oplus H(x_q) z_q} U_{x_q y_q z_q x_{q-1} y_{q-1} \oplus H(x_{q-1}) z_{q-1}} ... U_{x_2 y_2 z_2 x_1 y_1 \oplus H(x_1) z_1} \cdot$$

$$\alpha_{x_1 y_1 z_1} \alpha^*_{x'_1 y'_1 z'_1} U^*_{x'_2 y'_2 z'_2 x'_1 y'_1 \oplus H(x'_1) z'_1} ... U^*_{x' y' z' x'_q y'_q \oplus H(x'_q) z'_q}$$

In this form, we can move $Pr_D[H]$ to the inside of all the sums, and now see that the terms of the sum only depend on H's output values on $x_1, ..., x_q$ and $x'_1, ..., x'_q$. The sum can then be rewritten to run over all possible settings of these output values, and weight by $Pr_D[H(x_i) = r_i, H(x'_i) = r'_i \forall i \in \{1, ..., q\}]$ instead of $Pr[H]$, which uncovers the desired linear combination. □

4. Primitives that resist quantum queries

Quantum random oracles are not the only types of oracles that have presented problems for post-quantum cryptography. Given a classical construction based on a problem that is believed to be hard for quantum computers, we can ask whether it is secure when the adversary is allowed to evaluate the construction on quantum states. In this section, I describe two primitives that can be shown to hold up to such queries: a quantum PRF and a quantum MAC, though I focus on the former.

4.1. **Quantum-secure PRFs.**

In this section, I survey a recent paper by Zhandry that shows how to construct functions that are indistinguishable from random functions, even when the adversary can make quantum queries [8]. The main result in this paper is the result that an algorithm given oracle access to one of two distributions cannot distinguish the two with better advantage than an adversary given just a single sample from one of the two.

4.1.1. *Separation result.* Zhandry first motivates the study of quantum PRFs by showing that there exist PRFs which are secure against classical adversaries but not against quantum adversaries. He does this by constructing such a PRF out of a classically secure PRF, where the construction has a large random period. The period is detectable by a quantum adversary using known period-finding algorithms, but not by a classical adversary.

Theorem 4.1. *Let $\mathsf{PRF}: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a classical secure pseudorandom function, where $\mathcal{X}$ is the set of integers from 1 to N and $|\mathcal{Y}| \geq N^2$. Let $\mathcal{A}$ be the set of integers between $N/2$ and N, and let $\mathcal{X}$' be integers up to the smallest power of 2 greater than $4N^2$. Let $\mathsf{PRF}' : \mathcal{K} \times \mathcal{A} \times \mathcal{X} \to \mathcal{Y}$ be the function such that $\mathsf{PRF}'(k, a, x) = \mathsf{PRF}(k, x \bmod a)$. Then one of PRF or PRF' is secure against classical adversaries but not against quantum adversaries.*

Proof sketch The idea of the proof is to first show that PRF' is secure against classical adversaries if PRF is. To see this, first use a random function O in place of PRF to answer the queries of a PRF' adversary A, which does not affect the adversary's behavior except negligibly. Then a probability analysis shows that no classical adversary can find two points x and x' such that $x \equiv x' \mod a$ except with negligible probability. This means that A cannot distinguish $O(\cdot \mod A)$ from $O(\cdot)$ with polynomially many queries, which in turn implies that A cannot distinguish $\mathsf{PRF}(k, \cdot \mod a) = \mathsf{PRF}'(k, a, \cdot)$ from $O(\cdot)$.

Next, Zhandry shows that the quantum security of PRF implies the quantum insecurity of PRF'. (If PRF is not quantum secure, then the proof is complete.) In brief, we must establish that PRF' is periodic with period a, and then a known quantum period finding algorithm (for example, Boneh and Lipton [3]) can be applied to find the period and thereby distinguish PRF' from random. If PRF' were constructed from a truly random function instead of PRF, then by the birthday bound, the random function has no collisions with probability at least $1/2$. Therefore, with constant probability, $O(\cdot \mod a)$ is periodic with probability a, whereas $O(\cdot)$ is periodic with very low probability. This allows a quantum adversary to distinguish $O(\cdot \mod a)$ from $O(\cdot)$. Replacing $O(\cdot \mod a)$ with $\mathsf{PRF}(\cdot)$ does not affect the ability of an adversary to distinguish from random, since PRF is quantum secure by assumption. □

4.1.2. *Oracle indistinguishability.* In this section, I sketch Zhandry's proof of the equivalence of indistinguishability and oracle indistinguishability for quantum algorithms.[1] First, what is meant by "oracle indistinguishability?" Loosely speaking, it means that an efficient quantum algorithm making quantum queries to an oracle function "representing" one of two distributions cannot determine which distribution is being used. Each value of the oracle function is an independent sample from the distribution, and the algorithm is allowed to query a superposition to receive a superposition of samples.

Let D_1, D_2 be distributions over a set $\mathcal{Y}$. Recall that D_1 and D_2 are computationally indistinguishable if, for any efficient quantum algorithm Q, there exists a negligible function ϵ such that

$$|Pr_{y \leftarrow D_1}[Q(y) = 1] - Pr_{y \leftarrow D_2}[Q(y) = 1]| < \epsilon.$$

Now let $O_{\mathcal{X}, D_i}$ be the distribution over functions from $\mathcal{X}$ to $\mathcal{Y}$, where for $H \leftarrow O_{\mathcal{X}, D_i}$, $H(x)$ is chosen independently according to D_i.

Definition D_1 and D_2 are computationally *oracle indistinguishable* if, for any efficient quantum algorithm Q with a quantum oracle, there exists a negligible function ϵ such that

[1] Zhandry proves results for both computational and statistical indistinguishability, but in this report I focus on the computational setting for simplicity.

$$|Pr_{H\leftarrow O_{\mathcal{X},D_1}}[Q^{|H\rangle}() = 1] - Pr_{H\leftarrow O_{\mathcal{X},D_2}}[Q^{|H\rangle}() = 1]| < \epsilon.$$

At first glance, oracle indistinguishability seems stronger than regular indistinguishability, because the quantum algorithm can query the oracle on quantum states to receive exponentially many samples at once. However, Zhandry shows that in fact the two definitions are equivalent: quantum queries to a distribution give an algorithm no more distingushing power than a single sample!

Theorem 4.2. *D_1 and D_2 are computationally indistinguishable if and only if they are computationlly oracle indistinguishable.*

From a bird's-eye view, the proof of this theorem uses a polynomial number of samples to simulate the quantum-accessible oracle, by obtaining a polynomial-sized pool of samples and then implementing the oracle that randomly assigns each input to one of the samples in the pool. This simulated oracle distribution is close enough to the actual oracle distribution that the distinguishing algorithm still succeeds with non-negligible probability. A standard hybrid argument then reduces the number of samples needed to one.

Zooming in a bit, we must argue that this simulated oracle distribution is close enough to the real oracle distribution. From Theorem 3.7, we know that, for any $q-$query distinguishing algorithm, we can express its output behavior as a linear combination of the oracle distribution's behavior on all $2q$ possible input/output pairs. Suppose that we can write each of the terms in this linear combination as a polynomial with certain characteristics (Zhandry shows that we can). Then the algorithm's overall distinguishing power can also be expressed as a polynomial with specific behavior, and Zhandry proves a bound on such polynomials. This bound limits the algorithm's distinguishing power.

Now, I sketch these steps in more detail, focusing on how to characterize the terms in the linear combination as polynomials with the desired behavior, and how to bound an algorithm's distinguishing power using Zhandry's polynomial bound.

First, I state Zhandry's result for polynomials:

Theorem 4.3. *Let $p(\lambda)$ be a polynomial of degree d such that the first $\Delta - 1$ derivatives are 0 at 0, $p(0)$ is between 0 and 1, and $0 \leq p(1/r) \leq 1$ for all positive integers r. Then for all positive integers r,*

$$|p(1/r) - p(0)| < 2^{2-\Delta}\zeta(2\Delta)(1/r)^{\Delta}d^{3\Delta}$$

where ζ is the Riemann zeta function.

In this theorem, r is the parameter that will be used in the simulated oracle distribution, and the polynomial p corresponds to any quantum algorithm's power to distinguish between the simulated oracle distribution and the actual oracle distribution. The point of the theorem is to have a bound on the algorithm's distinguishing power for any choice of r. In particular, just by showing that the polynomial in question has has degree d and taking $\Delta = 1$, the distinguishing power of any algorithm can be bounded by $2\zeta(2)(1/r)d^3 = \pi^2 d^3/3r$.

The proof of this theorem can be found in Appendix B of[8]. At a very high level, the proof defines a related polynomial and uses Lagrange interpolation to interpolate this related polynomial. The difference between the two points $p(\lambda)$ and $p(0)$ is then written in terms of the interpolated polynomial. The assumptions that $p(x)$ is between 0 and 1 for $x = 0$ or x an integer can be used to upper bound

the difference, simplifying the sum that must be bounded. Each term in the sum is expanded into several terms, each of which is bounded individually using some algebraic manipulation.

With Theorem 4.3 in hand, I describe how it can be used to prove the equivalence of distinguishability and oracle indistinguishability. The plan is to describe a family of distributions and bound the ability of an algorithm to distinguish two distributions in this family, and then to show that the simulated oracle distribution is such a family.

Theorem 4.4. *Let D_r be a family of distributions on functions from $\mathcal{X}$ to $\mathcal{Y}$, for $r \in \mathbb{Z}^+ \cup \{\infty\}$. For any $2q$ pairs $(x_i, r_i) \in \mathcal{X} \times \mathcal{Y}$, suppose $p(\lambda) = Pr_{H \leftarrow D_{1/\lambda}}[H(x_i) = r_i \forall i]$ is represented by a polynomial of degree d in λ. Then for any quantum algorithm A making q queries, the output distributions of A under D_r and D_∞ are $\pi^2 d^3/3r-$close.*

Proof sketch The proof of this fact is fairly straightforward once Theorems 3.7 and 4.3 are established. The basic observation is that a distinguisher's advantage can be written as a difference of points on a polynomial meeting the conditions of Theorem 4.3.

From Theorem 3.7, for any output z, $Pr_{H \leftarrow D_{1/\lambda}}[A^H() = z]$ is a linear combination of polynomials of degree d, so it is itself a polynomial of maximum degree d in λ.

Suppose for simplicity that A outputs only 1 or 0, and that A's advantage in distinguishing $D_{1/\lambda}$ from D_∞ is $\epsilon(\lambda)$. Let z_λ be the more likely output under $D_{1/\lambda}$. This means that

$$Pr_{H \leftarrow D_{1/\lambda}}[A^H() = z_\lambda] - Pr_{H \leftarrow D_\infty}[A^H() = z_\lambda] = \epsilon(\lambda).$$

Now let $p_\lambda(\lambda') = Pr_{H \leftarrow D_{1/\lambda'}}[A^H() = z_\lambda]$. As mentioned above, p_λ is a degree d polynomial in λ'. At positive integer values of $1/\lambda'$ (as well as $\lambda' = 0$), the polynomial is equal to a probability so it is between 0 and 1. Therefore, we can apply Theorem 4.3 to p_λ to obtain that $p_\lambda(\lambda') - p_\lambda(0)$ is within the desired bound. Finally, observe that for any λ,

$$p_\lambda(\lambda) - p_\lambda(0) = Pr_{H \leftarrow D_{1/\lambda}}[A^H() = z_\lambda] - Pr_{H \leftarrow D_\infty}[A^H() = z_\lambda] = \epsilon(\lambda)$$

which completes the theorem. □

At the beginning of this subsection, I described the general form of the proof, in which we run a quantum distinguisher and answer its oracle queries with a simulated oracle distribution, such that answering the queries with the simulated oracle only requires a polynomial number of samples from the distribution. I now explain why this simulated oracle distribution is close enough to the actual oracle distribution, by showing that the two distributions are drawn from a family conforming to the conditions of Theorem 4.4.

Theorem 4.5. *Let D be a distribution over the set $\mathcal{Y}$. For positive integers r, let SR_r^D be the distribution of oracles from $\mathcal{X}$ to $\mathcal{Y}$ generated by first picking a set W of r samples from D, and then mapping each input to a uniformly random element of W. Then for all pairs (x_i, r_i) for $i = 1, ..., 2q$, $Pr_{H \leftarrow SR_r^D}[H(x_i) = r_i \forall i]$ is a polynomial in $1/r$ of degree d.*

Proof sketch To prove this theorem, it is helpful to think of SR_r^D as a composition of two distributions: O_1 that uniformly maps elements of $\mathcal{X}$ to $\{1, ..., r\}$, and O_2

that maps $\{1, ..., r\}$ to elements of $\mathcal{Y}$ according to D. If we consider the former O_1 distribution and restrict it to the x_i values, then each O_1 oracle can be uniquely associated with a partition of $\{1, ..., 2q\}$ and a mapping from each part to an element of $\{1, ..., r\}$. (All elements in the same part have the same oracle value.) The number of such mappings is easy to count and is a polynomial in r of degree at most $2q$. The distribution on O_1 oracles is uniform, and there are r^{2q} such oracles when restricted to the x_i values, so the polynomial from the number of mappings is divided by r^{2q}. Finally, the remaining terms don't depend on r, so the result is a polynomial in $1/r$ of degree at most $2q$. □

By applying Theorem 4.4 to the SR family, we can immediately get that no q query quantum algorithm can distinguish SR_r^D from SR_∞^D with probability greater than $\pi^2(2q)^3/3r$. Note that SR_∞^D is the distribution where each output value is chosen independently according to D.

Finally, we have all the tools needed to prove Theorem 4.2. For simplicity, here I explain a simplified proof that oracle indistinguishability is equivalent to computational indistinguishability given a polynomial number of samples; from there, a standard hybrid argument proves that a single sample is enough. As stated at the beginning of this subsection, the broad idea is to simulate the real oracle distribution using one of the SR distributions, requiring only a polynomial number of samples.

A partial proof sketch for Theorem 4.2 Let D_1 and D_2 be two distributions, and let A be an efficient quantum algorithm such that

$$|Pr_{H \leftarrow D_1}[A^{|H\rangle}() = 1] - Pr_{H \leftarrow D_2}[A^{|H\rangle}() = 1]| = \epsilon$$

for non-negligible ϵ. Let $l(q) = \pi^2(2q)^3/3$, and choose $r = 4l(q)/\epsilon$. (Note that r is polynomial in the security parameter.) As stated above, drawing H from $SR_r^{D_i}$ instead of D_i cannot change A's output distribution by more than $l(q)/r = \epsilon/4$, so

$$|Pr_{H \leftarrow SR_r^{D_1}}[A^{|H\rangle}() = 1] - Pr_{H \leftarrow SR_r^{D_2}}[A^{|H\rangle}() = 1]| \geq \epsilon/2$$

An algorithm B to distinguish D_1 from D_2 is as follows. B first queries for r samples from the challenge distribution; let these samples be $y_1, ..., y_r$. B runs A, and when A queries for the oracle value at x, B runs a randomly selected y_i. B outputs the output of A.

When B runs A, A's oracle distribution is SR_r^D, where D is B's challenge distribution. Since A distinguishes $SR_r^{D_1}$ from $SR_r^{D_2}$ with probability at least $\epsilon/2$, and B correctly distinguishes D_1 from D_2 whenever A is correct, B is an efficient algorithm that distinguishes D_1 from D_2 given a polynomial number of samples. □

4.1.3. *Security proof for GGM.* Recall that the GGM construction builds a pseudorandom function out of a length-doubling pseudorandom generator G. If $G(x) = (G_0(x), G_1(x))$ for any input x whose bits are $x_1, ..., x_n$, then the GGM construction is $\mathsf{PRF}_k(x) = G_{x_1}(G_{x_2}(...G_{x_n}(k)))$. A simple way to visualize this construction is a tree, where the left child of a node y is $G_0(y)$ and the right child is $G_1(y)$, and the root is k. Then the output of the PRF on x corresponds to taking the path down the tree dictated by the bits of x, and returning the value on the leaf.

The classical security proof for the GGM construction uses two hybrids. First, define H_i to be the construction where each node in the top i levels of the tree is a random value instead of G. Now, let H_j be a hybrid such that an algorithm A can distinguish H_j from H_{j-1} with non-negligible probability. Given a polynomial

number of samples from the challenge distribution, we can run A, and answer all of A's queries with random functions in the first $j-1$ levels, samples from the distribution in the j^th level, and G on the rest. (Since A only makes polynomially many oracle queries, we can answer these queries with polynomially many samples from the challenge distribution.) If the challenge distribution is uniform, then A sees H_{j-1}, and if the challenge is the pseudorandom generator, then A sees H_j, so A successfully distinguishes the two cases. Finally, a standard hybrid argument that if there exists a distinguisher for the pseudorandom generator that uses polynomially many samples, then there exists a distinguisher that uses only one.

In the quantum setting, Zhandry shows a slightly modified proof for the GGM construction, using the previous results showing that oracle access to either a pseudorandom generator or a uniform distribution gives an adversary no more distinguishing power than a single sample.

Proof sketch The first step of the classical proof translates directly to the quantum setting: in hybrid i, the adversary is given a quantum-accessible oracle for H_i as defined above. H_0 corresponds to a quantum-accessible oracle to PRF and H_n to a random oracle. Let H_j be some hybrid such that an algorithm $A^{|O\rangle}$ can distinguish $O = H_j$ from $O = H_{j-1}$ with non-negligible probability. Then consider the algorithm B that runs A and give it the quantum-accessible oracle O such that $O(x)$ uses random values at the first $j-1$ levels of the tree, samples from the challenge distribution oracle at the j^{th} level, and G for the rest of the path. One query on a quantum state to O results in B making one query on a quantum state to the oracle representing the challenge distribution. As in the classical case, A succeeds with non-negligible probability, allowing B to distinguish an oracle where each output is drawn uniformly at random from an oracle where each output is $G(x)$ for uniformly random x. Finally, because oracle indistinguishability is equivalent to indistinguishability, we have shown that $G(x)$ for random x is distinguishable from uniform, a contradiction. □

4.2. **Quantum-secure MACs.** In this subsection, I give a whirlwind tour of another primitive that has been shown to be secure against adversaries who can query it on quantum states: quantum-secure message authentication codes, studied by Boneh and Zhandry [5]. I explain the security model and the main results, but will only briefly mention the proof techniques used.

4.2.1. *Security model.* In the classical setting, a MAC is secure if, for any PPT adversary A, where A is given MACs on q messages of A's (adaptive) choosing, then A cannot produce $q+1$ valid message/MAC pairs except with negligible probability. In the quantum setting, the adversary is allowed to query the MAC function on a polynomial number of *quantum states*, and still should not be able to product $q+1$ valid message/MAC pairs except with negligible probability.

Boneh and Zhandry almost admit that the motivation for quantum-secure MACs is somewhat tenuous, since the computer performing the MAC queries can always measure before returing the result. It seems unlikely that in the real world (even a real world with rampant quantum computers) an adversary would actually be able to receive a MAC on a quantum state. In contrast, quantum-secure PRFs are well-motivated because they are used to simulate random oracles, in which case each party (including the adversary) has the full code for the function and can evaluate it himself on quantum states. Nevertheless, Boneh and Zhandry argue for

quantum-secure MACs as a conservative security model. Perhaps cryptographic primitives that resist quantum queries can be viewed in the same line of work as leakage-resilient cryptography, where we want to design cryptographic primitives that are secure even in the face of some implementation error (in this case, an implementation that "forgets" to measure before returning the MAC).

4.2.2. *Quantum-secure MACs from quantum-secure PRFs.* In the classical setting, it is almost immediate that a PRF provides a MAC. However, even a quantum-secure PRF cannot be used as a quantum-secure MAC without some further study: querying the PRF on a superposition of messages might reveal information about the MAC of more than one message. To show that a quantum-secure PRF is indeed a quantum-secure MAC, Boneh and Zhandry introduce a general technique for proving lower bounds on quantum algorithms, which they call the rank method.

At a high level, the rank method bounds the dimension of the space spanned by all possible outputs of a quantum algorithm, thereby bounding the ability of the algorithm to produce an output which is outside the space. More formally, suppose a quantum algorithm is given access to some oracle $z \in Z$ drawn according to distribution D, and outputs some value $w \in W$. Let R be a relation mapping $Z \times W$ to $\{True, False\}$. Now consider the matrix where each row is the final state of A before measurement when A is run with the oracle being a particular z, and let this matrix be $M_{A,Z}$. What, then, is the probability that A outputs some w such that $R(z, w)$ is True? Boneh and Zhandry show that this probability is at most the rank of $M_{A,Z}$ times the best probability of success for any algorithm that outputs a fixed w independent of z.

This means that, to bound a quantum algorithm's power to solve some problem, we must bound the rank of the matrix corresponding to this algorithm and bound the best probability of success for any algorithm that ignores the oracle. To help bound the rank, Boneh and Zhandry first define the quantity

$$C_{k,q,n} = \sum_{r=0}^{q} \binom{k}{r} (n-1)^r.$$

For any $q-$query algorithm A with oracle access to a function from $\mathcal{X}$ to $\mathcal{Y}$, where the function is fixed at all but k points, they show that $rank(M_{A,H}) \leq C_{k,q,n}$. They do this by constructing a basis for the space spanned by the rows of $M_{A,H}$, and counting the number of vectors in this basis to be $C_{k,q,n}$. The basis is composed of final states of A when the oracle is a function that is fixed at all but q of the k "fixed" points from above.

Next, Boneh and Zhandry apply this rank method to the problem of producing $q+1$ input/output pairs of a random function given q quantum queries to it. They write a $q-$query algorithm's success probability in terms of the dimension of the space spanned by vectors representing the outputs of A when run on different oracles, allowing the application of the previous results to bound the algorithm's success probability to $\frac{1}{n^{q+1}} C_{q+1,q,n}$, where n is the size of the oracle's range. For exponentially large n and a polynomial number of queries, this bound becomes a negligible probability.

Finally, Boneh and Zhandry observe that the random function in the previous result can be replaced with a quantum-secure PRF without affecting any algorithm's behavior non-negligibly, so a quantum-secure PRF is a quantum-secure MAC.

5. Conclusion

Quantum queries can present difficult problems for post-quantum cryptography, but a growing collection of tools suggests that many constructions can be proven secure against quantum query adversaries, albeit with a little more work than in the classical case. There are many open problems in this area. Among them include proving security for a quantum pseudorandom permutation, finding quantum-secure claw-free permutations with which to instantiate the signature scheme from Section 3.2.1, and studying security models where adversaries are allowed to make more quantum queries, such as quantum queries to signature oracles or chosen ciphertext oracles.

References

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

[2] Mihir Bellare and Philip Rogaway. Lecture notes on cryptography. `http://cseweb.ucsd.edu/~mihir/papers/gb.pdf`, 2008.

[3] Dan Boneh and Richard J. Lipton. Quantum cryptanalysis of hidden linear functions (extended abstract). In *Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '95, pages 424–437, London, UK, UK, 1995. Springer-Verlag.

[4] Dan Boneh, Ozgu r Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *Proceedings of Asiacrypt*, 2011. Full version available at the Cryptology ePrint Archives: `http://eprint.iacr.org/2010/428/`.

[5] Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes, 2012. Full version available at the Electronic Colloquium on Computational Complexity: `http://eccc.hpi-web.de/report/2012/136`.

[6] Jean-Sébastien Coron. On the exact security of full domain hash. In *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '00, pages 229–235, London, UK, UK, 2000. Springer-Verlag.

[7] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 197–206, New York, NY, USA, 2008. ACM.

[8] Mark Zhandry. How to construct quantum random functions. In *Proceedings of FOCS*, 2012. Full version available at the Cryptology ePrint Archives: `http://eprint.iacr.org/2012/182/`.

[9] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In *Proceedings of Crypto*, 2012. Full version available at the Cryptology ePrint Archives: `http://eprint.iacr.org/2012/076/`.