Making Existential-Unforgeable Signatures Strongly Unforgeable in the Quantum Random-Oracle Model

Edward Eaton[1] and Fang Song[1,2]

1 **Department of Combinatorics & Optimization, University of Waterloo**
{eeaton,fang.song}@uwaterloo.ca
2 **Institute for Quantum Computing, University of Waterloo**

Abstract

Strongly unforgeable signature schemes provide a more stringent security guarantee than the standard existential unforgeability. It requires that not only forging a signature on a new message is hard, it is infeasible as well to produce a new signature on a message for which the adversary has seen valid signatures before. Strongly unforgeable signatures are useful both in practice and as a building block in many cryptographic constructions.

This work investigates a generic transformation that compiles any existential-unforgeable scheme into a strongly unforgeable one, which was proposed by Teranishi et al. [30] and was proven in the classical random-oracle model. Our main contribution is showing that the transformation also works against *quantum* adversaries in the *quantum* random-oracle model. We develop proof techniques such as adaptively programming a quantum random-oracle in a new setting, which could be of independent interest. Applying the transformation to an existential-unforgeable signature scheme due to Cash et al. [10], which can be shown to be quantum-secure assuming certain lattice problems are hard for quantum computers, we get an efficient quantum-secure strongly unforgeable signature scheme in the quantum random-oracle model.

1998 ACM Subject Classification E.3 Public key cryptosystems

Keywords and phrases digital signatures, strongly unforgeable, quantum random-oracle, lattices

Digital Object Identifier 10.4230/LIPIcs.TQC.2015.p

1 Introduction

Digital signature is a fundamental primitive in modern cryptography and has numerous applications. In a signature scheme, a signer uses his/her secret key to generate a signature on a message. Anyone who knows the corresponding public key can verify the integrity of the message and that it comes from the genuine signer. A standard security notion for digital signatures is called *existential-unforgeable* under *adaptive chosen-message-attacks* (eu-acma in short). Basically it means that an adversary, without knowing the secret key of a user, cannot forge a valid signature on a *new* message. This should hold even if the adversary has seen a few signatures generated by the honest user on messages adaptively chosen by the adversary. Another important security notion, stronger than eu-acma, is called *strongly* existential-unforgeable (su-acma). Here, in addition to eu-acma, it should be infeasible to forge a *new* signature on a previously signed message. Aside from applications in some practical scenarios [26], su-acma signatures turn out to be a very powerful tool in other cryptographic constructions. For instance they are used in transforming encryption schemes

10th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2015).
Editors: Salman Beigi and Robert Koenig; pp. 1–16

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

that are secure under chosen-plain-text attacks into secure schemes under *chosen-ciphertext-attacks* [13, 6]; and in constructing identity-based blind signatures [15] and group signature schemes [2, 5].

Strongly unforgeable signature schemes can be obtained from existential-unforgeable ones via generic transformations [29, 18, 30]. The transformation in [30] (referred to as TOO hereafter) is particularly interesting because it only needs a mild computational assumption and the overhead it causes to the efficiency is small. This work studies this transformation in the quantum setting, where adversaries have the power of processing quantum information. We want to ask: *does* TOO *transformation still hold in the presence of quantum adversaries, and furthermore can we obtain quantum-secure* su-acma *signatures systematically?*

There is no quick answer to this question. In general a classically secure cryptographic construction can completely fall apart against quantum adversaries for at least two reasons. First of all, quantum computers can solve some problems efficiently which are otherwise believed hard classically. This breaks the computational assumption in many constructions. For example, many existing eu-acma signature schemes, the starting point of the transformation, are based on factoring or discrete logarithm. The TOO transformation itself also uses the discrete logarithm problem. They are immediately broken by Shor's quantum algorithms [27]. Naturally we may want to switch to *quantum-safe* assumptions. For example, we assume certain lattice problems are hard even against quantum algorithms and then construct crypto-systems based on them [25, 4]. However, this does not fix everything immediately due to another reason, which is more subtle. Security of a construction is established by a security reduction, which is a proof by contradiction showing that if a scheme is not secure, then one can break a computational assumption. Unfortunately, as pointed out by a line of works (e.g., [35, 16, 31, 28]), classical security reductions may not hold in the presence of quantum adversaries due to technical difficulties such as *quantum rewinding*.

There is an additional complication, which turns out to be the main difficulty towards making the TOO transformation go through in the quantum setting. Classically, TOO is proven in the random-oracle model (RO), where a hash function is treated as a truly random function and all users evaluate the hash function by querying the random function. However once an adversary becomes quantum, we should naturally allow the queries to be in quantum superposition. This is formalized as the quantum random-oracle model (QRO) [7]. The bad news is that many classical tricks in RO become difficult to apply in QRO, if not entirely impossible. For starters, classically it is trivial to answer random-oracle queries on-the-fly by generating fresh random value for new queries while maintaining a table to keep consistency. It is not obvious that some similar trick can handle quantum superposition queries. There have been a host of works in recent years developing proof techniques in QRO [36, 33, 32], but many classical techniques are still missing their counterparts in QRO.

Our Contributions. Our main result is showing that the TOO transformation still works against quantum adversaries in the quantum random-oracle model under reasonable computational assumptions. Specifically, we first make a simple observation that classically the TOO transformation actually holds using any (generic) chameleon hash function, rather than the specific instantiation by the discrete log problem. As our central contribution, we prove that once the chameleon hash function and the eu-acma signature scheme are both quantum-safe, then TOO transformation will produce a quantum-safe su-acma signature scheme in the quantum random-oracle model. In our proof, we develop a technique that allows for adaptively programming a quantum random-oracle in a new setting. We hope this technical can find applications and extensions elsewhere.

Once we have the transformation ready, we demonstrate instantiations of the building

blocks to obtain concrete quantum-safe su-acma schemes. Using tools from [28], it is easy to verify that the bonsai-tree signature scheme by Cash et al. [10] is eu-acma against quantum adversaries assuming some lattice problem is quantum-safe[1]. In [10], a chameleon hash function was also proposed based on the same computational assumptions, which is easy to check that it is quantum-safe as well. Putting these pieces together, we can get a quantum-safe su-acma scheme.

Overview of Our Proof Techniques in QRO. As we mentioned earlier, many proof techniques in classical RO do not immediately go through in the QRO model. Roughly speaking, the classical proof for the TOO transformation relies on two features in the classical RO model: the history of queries that an adversary makes to the RO can the recorded, and at various steps one can assign a fresh random value on an input, since the response at an input needs not to be determined before being queried. Both become difficult in the quantum setting. Copying quantum superposition queries which are unknown quantum states is generally impossible, and apparently a single quantum query of the form $\sum |x, y\rangle \mapsto \sum |x, \mathcal{O}(x) \oplus y\rangle$ would "see" the function values at all inputs. It is hence unclear how to change $\mathcal{O}(x)$ later without being caught.

The first issue turns out to be non-essential. The purpose of keeping the RO queries is to make sure some special input x^* has not been queried by the adversary. Otherwise x^* can be used to break some assumption. In the quantum setting, we can just pick one of the queries at random and measure it. If the overall amplitude that adversary intends to query at x^* is high, the probability we recover x^* is only reduced by essentially a poly-factor (the number of the adversary's RO queries).

We then come up with a technique for adaptively programming a QRO in a new setting. Namely we want to change the function value at various inputs that the adversary has partial control (e.g., the prefix of these inputs are chosen by the adversary). Intuitively this is possible when these inputs still have sufficient uncertainty to the adversary. There exist techniques previously when these input strings are *information-theoretically* undetermined, possessing a high min-entropy for example [32, 34]. In contrast, in our case these inputs are *computationally* difficult to decide by the adversary. Namely, these inputs remain uncertain to the adversary unless some computational assumption is broken. We show that this is already sufficient freedom for programming the answers on these inputs. Being a little more specific, we show that the computational assumption implies *indistinguishability* of two functions which a distinguisher can have quantum access to: one is the all-zero function, and the other marks a set of strings that could be used to break the computational assumption. This may be interpreted as a computational analogue of the Grover search lower bound in quantum query complexity. This enables us to program a quantum random-oracle adaptively. Basically, the random-oracle embeds one of the preceding functions, and programming the random-oracle roughly amounts to switching between the two functions. Since the two functions are indistinguishable, any efficient quantum algorithm querying the random-oracle cannot notice whether we have re-programmed the quantum random-oracle. From a technical point of view, these claims may not sound very surprising. Nonetheless, we view them as an interesting conceptual shift, which is similar in spirit to [11] where the authors showed that *computational* constraints can force measurement on a quantum state and cause collapse to particular subspaces. Our techniques also complements existing ones that are of information-theoretical flavor.

[1] Actually, we observe a tighter security reduction so that a slightly weaker assumption on the lattice problem is sufficient.

Related Works. Boneh and Zhandry [8] considered a stronger type of quantum attacks on signature schemes where an adversary can query a signing oracle in superposition. They proposed general transformations which amplify schemes that are secure against ordinary quantum adversaries (i.e., those who only issue classical signing query as we consider in this work), to achieve security under attacks with superposition signing queries. In contrast, the transformation in our work only considers ordinary quantum adversaries, but tries to amplify in terms of the type of forgeries that an adversary can produce. Lyubashevsky [21, 22] applied the Fiat-Shamir paradigm to construct lattice-based su-acma signatures in the random-oracle model from identification schemes. However whether these schemes are quantum-secure is unclear, because proving quantum security of the identification schemes faces the difficulty of quantum rewinding. More importantly, there is negative evidence that Fiat-Shamir paradigm may not hold in general in the QRO model [12, 1]. Dagdelen et al. [12] showed that a variant of Fiat-Shamir works in the QRO model, but only for a very special form of identification schemes. In a recent work by Unruh [34], a general transformation is proposed, which can produce (quantum-safe) strongly-unforgeable signatures in the QRO model from general Σ-protocols. However the overhead is much larger than the Fiat-Shamir transformation, and the resulting signature schemes are less efficient than what can be obtained from our work. We remark that there is a generic Merkle-tree approach that produces su-acma schemes out of su-acma one-time signature schemes, which should still hold against quantum adversaries. Therefore in principle, lattice-based one-time signatures, as in [23], would suffice for full-fledged quantum-safe su-acma schemes. However the resulting scheme is usually far less efficient and costly to manage (because it is typically stateful).

2 Preliminary

We review necessary definitions and cryptographic tools in this section.

▶ **Definition 1** (Signature Scheme)**.** A **signature scheme** is composed of a triplet of probabilistic polynomial-time algorithms (G, S, V), satisfying the following:

- G is the key generation algorithm. On running, it produces a pair, (pk, sk). pk is the public key, or verification key, while sk is the secret key, or signing key.
- S is the signing algorithm. Upon input of a message M from a message space $\mathcal{M}$, as well as a secret key sk, it produces a signature σ on that message.
- V is the verification algorithm. It takes in a message M, a signature σ, and a public key pk, and will output either 'accept' or 'reject'.

Signature schemes must satisfy the **correctness requirement**, which is that for any (pk, sk) generated by G, and any $M \in \mathcal{M}$, if $\sigma \leftarrow S(M, sk)$ then $V(M, \sigma, pk) =$ 'accept'.

A standard security notion for signature schemes is **existential unforgeability under adaptive chosen message attack** (eu-acma).

▶ **Definition 2** (Existential Unforgeability under Adaptive Chosen Message Attack)**.** Consider the following game between a challenger $\mathcal{C}$ and a forger $\mathcal{A}$:

- $\mathcal{C}$ runs G, and send the resulting pk to $\mathcal{A}$.
- $\mathcal{A}$ sends up to q messages $M_1, M_2, ..., M_q$ to $\mathcal{C}$, one at a time. For each message $\mathcal{C}$ receives, she sends back $\sigma_i = S(M_i, sk)$ to $\mathcal{A}$.
- $\mathcal{A}$ finally outputs a pair (M^*, σ^*) to $\mathcal{C}$. We call this a valid forgery if $M^* \neq M_i \forall i \in \{1, ..., q\}$ and $V(M^*, \sigma^*, pk) =$ 'accept'.

If, for polynomially bounded q, it is computationally infeasible for $\mathcal{A}$ to come up with a valid forgery, the scheme is said to be existentially unforgeable under adaptive chosen message attack.

▶ **Definition 3** (Strong Unforgeability under Adaptive Chosen Message Attack)**. Strong unforgeability under Adaptive Chosen Message attack**, or **su-acma**, is defined in the same way as eu-acma, except that the pair (M^*, σ^*) that $\mathcal{A}$ eventually submits must only require that $(M^*, \sigma^*) \neq (M_i, \sigma_i)$ for all i, instead of the requirement that $M^* \neq M_i$. This change means that the forgery $\mathcal{A}$ submits may either be a new message, or may be a message that $\mathcal{C}$ has already signed, but with a new signature.

Note that by allowing $\mathcal{A}$ to submit more kinds of forgeries, if it is still computationally infeasible for $\mathcal{A}$ to succeed, then we know that this type of forgery also cannot be created, making the scheme in a sense stronger.

Chameleon hash functions. Chameleon hash functions were introduced by Krawczyk and Rabin [19]. We need a slight generalization proposed in [10]. A family $\mathcal{H}$ of chameleon hash function is a collection of functions h that takes in a message m from a message space $\mathcal{M}$ and some randomness r from a randomness space $\mathcal{R}$, and outputs to a range $\mathcal{Y}$, ie, $h : \mathcal{M} \times \mathcal{R} \to \mathcal{Y}$. The randomness space is associated with some efficiently sampleable distribution. There are three properties we need for a family of chameleon hash functions:

- (Chameleon property) We require an algorithm HG that samples a hash function $h \in \mathcal{H}$ together with trapdoor information td satisfying that for any $m \in \mathcal{M}$ and $y \in \mathcal{Y}$, it is possible to efficiently sample $r \leftarrow h_{td}^{-1}(m, y)$ under the distribution associated with $\mathcal{R}$ such that $h(m, r) = y$.
- (Uniformity) For $h \leftarrow \mathcal{H}$ and $r \leftarrow \mathcal{R}$, $(h, h(m, r))$ is uniform over $(\mathcal{H}, \mathcal{Y})$ up to negligible statistical distance.
- (Collision resistance) For a hash function $h \leftarrow \mathcal{H}$, it is computationally infeasible for an adversary to find $(m, r), (m', r')$, with $(m, r) \neq (m', r')$ such that $h(m, r) = h(m', r')$.

Quantum Random-Oracle Model. The random oracle model is a technique used in cryptographic proofs. In it, Hash functions are replaced with random oracles. An adversary is given access to query this random oracle by providing an input, x, and is returned the response, $\mathcal{O}(x)$. These random oracles exist to replace hash functions in our proof. When we examine the proof in the context of quantum computers, Boneh et al. [7] have pointed out that since superposition queries to hash functions are possible, to truly capture this in a model allowing quantum computers, we must allow superposition queries to the random oracle. So we will allow superpositions of queries to our random oracle, $\sum a_x |x, y\rangle$, which will be responded to with a superposition of answers, $\sum a_x |x, y \oplus \mathcal{O}(x)\rangle$.

A cryptographic scheme is said to be *quantum-safe* (or quantum-secure) if the security conditions still hold once the adversaries become efficient quantum computers. We do not go into more precise definitions. See for example [16] for details.

3 Getting SU from EU in QRO

In this section we prove our main theorem.

▶ **Theorem 4.** *There exists a generic conversion that takes an quantum-safe* eu-acma *signature scheme $\Sigma = (G, S, V)$ and a family of quantum-safe collision-resistant chameleon hash functions $\mathcal{H}$ and produces a quantum-safe* su-acma *signature scheme $\Sigma' = (G', S', V')$ in the quantum random-oracle model.*

3.1 The Transformation

We first recall the TOO transformation [30] with a minor change. We use a generic chameleon hash function instead of an instantiation from the discrete log problem.

- G'. On input a security parameter 1^n, do the following:
  - Run G, obtaining (pk, sk).
  - Run HG obtaining a chameleon hash function h with trapdoor td.
  - Set $pk' = (pk, h)$ and $sk' = (sk, td)$.
- S'. On input of message M, do the following:
  - Sample a random C from the range of h.
  - Sign C using the signing algorithm S, obtaining $\sigma = S(C, sk)$
  - Compute $m = \mathcal{O}(M\|\sigma)$, where $\mathcal{O}$ is a hash function (to be replaced with a random oracle in the proof).
  - Using the trapdoor information td, find an r such that $h(m, r) = C$.
  - Output $\sigma' = (\sigma, r)$.
- V'. On input of a message M and a signature $\sigma' = (\sigma, r)$, do the following:
  - Compute $m = \mathcal{O}(M\|\sigma)$ and $C = h(m, r)$.
  - Output 'Accept' if and only if $V(C, \sigma, pk)$ = 'Accept' (otherwise, output 'Reject').

The correctness of the algorithm can be seen easily. If σ' was a signature generated on M using S', then C will be the same C generated during the running of S', and is precisely what σ is a signature for.

3.2 Main Technical Lemma: Adaptively Programming a Quantum RO

To prove the main theorem, we demonstrate a new scenario where we can adaptively program a quantum random-oracle. This extends existing works (e.g [32, 33, 34]) from information-theoretical setting to a computational setting, and we believe it is potentially useful elsewhere. We will formalize a probabilistic game which we call *witness-search*. It potentially captures the essence of numerous security definitions for cryptographic schemes (e.g. signatures). Then we show that the (computational) hardness of witness-search allows for adaptively programming a quantum random-oracle.

Let Samp be an instance-sampling algorithm. On input 1^n, Samp generates public information pk, description of a predicate P, and a witness w satisfying $P(pk, w) = 1$. Define a witness-search game WS as below.

Witness-Search Game WS

1. Challenger $\mathcal{C}$ generates $(pk, w, P) \leftarrow \mathsf{Samp}(1^n)$. Ignore w. Let $W_{pk} := \{w : P(pk, w) = 1\}$ be the collection of valid witnesses.
2. $\mathcal{A}$ receives pk and produces a string $\hat{w}$ as output.
3. We say $\mathcal{A}$ wins the game if $\hat{w} \in W_{pk}$.

We say WS(Samp) is hard, if for any poly-time $\mathcal{A}$, $\Pr[\mathcal{A} \text{ wins}] \leq \mathsf{negl}(n)$. For instance, Samp could be the KeyGen algorithm of a signature scheme. pk consists of the public key and description of the signature scheme. Predicate P is the verification algorithm and a witness consists of a valid message-signature pair. Security of the signature scheme implies hardness of WS(Samp).

▶ **Lemma 5** (Hardness of Witness-Search to Programming QRO). *Let two experiments E and E' be as below. If* WS *is hard, then* $\mathrm{ADV} := |\Pr_E[b = 1] - \Pr_{E'}[b = 1]| \leq \mathsf{negl}(n)$.

Note that E' differs from E only in that we reprogram the random oracle at some point in E'. We defer the proof of this lemma to Appendix **??**.

Experiment E

1. Generate $(pk, w, P) \leftarrow \mathsf{Samp}(1^n)$.
2. $\mathcal{O} \leftarrow \mathcal{F}$ is drawn uniformly at random from the collection of all functions $\mathcal{F}$.
3. $\mathcal{A}_1$ receives pk as input and makes at most q_1 queries to $\mathcal{O}$. $\mathcal{A}_1$ produces a classical string x.
4. Set $z := \mathcal{O}(x\|w)$.
5. $\mathcal{A}_2$ gets (x, w, z) and may access the final state of $\mathcal{A}_1$. $\mathcal{A}_2$ makes at most q_2 queries to $\mathcal{O}$. It outputs $b \in \{0, 1\}$ at the end.

Experiment E'

1. Generate $(pk, w, P) \leftarrow \mathsf{Samp}(1^n)$.
2. $\mathcal{O} \leftarrow \mathcal{F}$ is drawn uniformly at random from the collection of all functions $\mathcal{F}$.
3. $\mathcal{A}_1$ makes at most q_1 queries to $\mathcal{O}$. It produces a classical string x.
4. Pick a random $z \in_R \mathsf{Range}(\mathcal{O})$. Reprogram $\mathcal{O}$ to $\mathcal{O}'$: $\mathcal{O}'(y) = \mathcal{O}(y)$ except that $\mathcal{O}'(x\|w) = z$.
5. $\mathcal{A}_2$ gets (x, w, z) and may access the final state of $\mathcal{A}_1$. $\mathcal{A}_2$ makes at most q_2 queries to $\mathcal{O}'$. It outputs $b \in \{0, 1\}$ at the end.

To prove Lemma 5, we need another lemma below to pave the road. Roughly we want to argue that if witness-search is hard, then given an oracle which is either the all-zero function or a function that marks the witness set W_{pk}, no efficient algorithms can distinguish them. This may be intuitively interpreted as a computational analogue of Grover search lower bound. Its proof can be found in Appendix B.

▶ **Lemma 6.** *Let f be the all-zero function, and f_S be the characteristic function of a set S. Namely $f_S(x) = 1$ iff. $x \in S$. Define two experiments G and G' as below. If* $\mathsf{WS}(\mathsf{Samp})$ *is hard, then for any efficient $\mathcal{A}$ making $q \leq poly(n)$ queries, $|\Pr_G[b = 1] - \Pr_{G'}[b = 1]| \leq \mathsf{negl}(n)$.*

Experiment G

1. Generate $(pk, w, P) \leftarrow \mathsf{Samp}(1^n)$.
2. $\mathcal{A}$ is given pk and (quantum) access to f. $\mathcal{A}$ makes at most q queries to f and afterwards w is given to $\mathcal{A}$. It outputs $b \in \{0, 1\}$ and aborts.

Experiment G'

1. Generate $(pk, w, P) \leftarrow \mathsf{Samp}(1^n)$. Let $f_{pk} := f_{W_{pk}}$, where $W_{pk} = \{w : P(w) = 1\}$. (i.e., $f_{pk}(x) = 1$ iff. $x \in W_{pk}$)
2. $\mathcal{A}$ is given pk and (quantum) access to f_{pk}. $\mathcal{A}$ makes at most q queries to f_{pk} and afterwards w is given to $\mathcal{A}$. It output $b \in \{0, 1\}$ and aborts.

Proof of Lemma 5. We use a hybrid argument to prove the theorem. Define $E_i, i = 1, \ldots, 4$ as follows.

- $E_1 := E$. ($\mathcal{A}_1^{\mathcal{O}}/\mathcal{A}_2^{\mathcal{O}}$ in short.)
- E_2: identical to E_1 except that in step 3, $\mathcal{O}$ is replaced by $\bar{\mathcal{O}}$ where $\bar{\mathcal{O}}(y) = \mathcal{O}(y)$ but $\bar{\mathcal{O}}(y) = 0$ for any $y = \cdot \| w$ where $w \in W_{pk}$. ($\mathcal{A}_1^{\bar{\mathcal{O}}}/\mathcal{A}_2^{\mathcal{O}}$)
- E_3: identical to E_2 except that after step 3, we use $\mathcal{O}'$ as defined in E' instead of $\mathcal{O}$. Observe that E_3 can also be obtained from E' by substitute $\bar{\mathcal{O}}$ for $\mathcal{O}$ in step 3. ($\mathcal{A}_1^{\bar{\mathcal{O}}}/\mathcal{A}_2^{\mathcal{O}'}$)
- $E_4 := E'$. ($\mathcal{A}_1^{\mathcal{O}}/\mathcal{A}_2^{\mathcal{O}'}$)

Define $\mathrm{ADV}_i := \left|\Pr_{E_i}[b=1] - \Pr_{E_{i+1}}[b=1]\right|$. We will show that ADV_1 and ADV_3 are both negligible using Lemma 6. $\mathrm{ADV}_2 = 0$ since in both E_2 and E_3, the function values for W_{pk} are assigned uniformly at random and independent of anything else. Therefore we conclude that $\mathrm{ADV} = |\Pr_E[b=1] - \Pr_{E'}[b=1]| \leq \sum \mathrm{ADV}_i = \mathsf{negl}(n)$.

We are only left to prove that $\mathrm{ADV}_1 \leq \mathsf{negl}(n)$, and $\mathrm{ADV}_3 \leq \mathsf{negl}(n)$ follows by similar argument. Suppose for contradiction that there exist $(\mathcal{A}_1, \mathcal{A}_2)$ such that $\mathrm{ADV}_1 \geq 1/p(n)$ for some polynomial $p(\cdot)$. We show that this will lead to a contradiction to Lemma 6 that $|\Pr_G[b=1] - \Pr_{G'}[b=1]| \leq \mathsf{negl}(n)$, which in turn contradicts the hardness of witness-search. To see this, we construct an algorithm D from $(\mathcal{A}_1, \mathcal{A}_2)$ that runs in G and G' such that $|\Pr_G[b=1:D] - \Pr_{G'}[b=1:D]| \geq 1/p(n)$. Let F be an oracle which ignores the first part of the input and then applies either all-zero function f or f_{pk} (as defined in G') on the second part. Let g be a random function. Define another oracle $H := g \circ F$ that implements the following transformation:

$$
\begin{aligned}
|x, y\rangle &\mapsto |x, y\rangle \otimes |0\rangle \quad \text{append an auxiliary register} \\
&\mapsto |x, y\rangle \otimes |\overline{F(x)}\rangle \quad \text{compute the negation of } F \text{ on aux.} \\
&\mapsto |x, y \oplus \overline{F(x)} \cdot g(x)\rangle \otimes |\overline{F(x)}\rangle \quad \text{controlled-}g \\
&\mapsto |x, y \oplus \overline{F(x)} \cdot g(x)\rangle \quad \text{uncompute negation of } F \text{ and disgard aux.}
\end{aligned}
$$

Observe that if F is induced from f then H is identical to a random function $\mathcal{O}$. Whereas if F comes from f_{pk} then H is identical to $\bar{\mathcal{O}}$ as in E_2. For an algorithm that queries at most q times to H, we can sample h from a family of $2q$-wise independent functions and simulate H efficiently (with access to F) without any noticeable difference.

Construction of D

1. D receives pk and an oracle F (one of the two candidates above).
2. D simulates oracle $H = g \circ F$ as defined above. D then simulates $\mathcal{A}_1$, for each of query from $\mathcal{A}_1$, it is answered by H with (two) oracle calls to F. Let x be the output of $\mathcal{A}_1$.
3. D receives w (from external challenger). It then simulates $\mathcal{A}_2$ on input $(x, w, z := H(x\|w))$ and oracle queries are answered by h.
4. D outputs the output of $\mathcal{A}_2$.

It is easy to see that if F is induced from f, the view of $\mathcal{A}_1$ and $\mathcal{A}_2$ is identical to that of E_1. Likewise if F is induced by f_{pk} then it is the same view as in E_2. Therefore $|\Pr_G[b = 1:D] - \Pr_{G'}[b=1:D]| = |\Pr_{E_1}[b=1:(\mathcal{A}_1,\mathcal{A}_2)] - \Pr_{E_2}[b=1:(\mathcal{A}_1,\mathcal{A}_2)]| \geq 1/p(n)$. This gives a contradiction.

◀

3.3 Proof of Theorem 4

Brief Review of Classical Proof. Classical proof roughly goes as follows: consider a forger $\mathcal{A}$. If (M^*, σ'^*) is the forgery that $\mathcal{A}$ eventually submits, we will let $C^* = h(\mathcal{O}(M^*\|\sigma^*), r^*)$.

Similarly, for a signing query made by the forger M_i, we let $C_i = h(\mathcal{O}(M_i\|\sigma_i), r_i)$.

We then analyze two separate cases. First the instance where $C^* \neq C_i$ for all i. In this case we show that this gives a break to the existential unforgeability of the signature scheme Σ, by way of (C^*, σ^*). Next, we examine the case where $C^* = C_i$ for some i. In this case we show that $(\mathcal{O}(M^*||\sigma^*), r^*)$ and $(\mathcal{O}(M_i||\sigma_i), r_i)$ provide a break to the collision resistance of the chameleon hash function.

For completeness the full classical proof is included in Appendix A. It is adapted from [30] and we use a generic chameleon hash function instead of a concrete instantiation from the discrete logarithm problem. There are also changes which by our opinion make the proof easier to understand.

Proof in the quantum random-oracle model. Let $\mathcal{A}$ be the forger making at most q queries, and let ϵ be the probability that $\mathcal{A}$ succeeds in her forgery. We construct $\mathcal{B}$ that either breaks existential unforgeablity of Σ or can find collisions in $\mathcal{H}$.

- **Case 1**: We define this case as occurring when $C^* \neq C_i$ for all i.
  Firstly, $\mathcal{B}$ will be acting as a quantum random oracle for $\mathcal{C}$. To do this, $\mathcal{B}$ simply chooses a $2q$-wise independent hash function, $\mathcal{O}$, and for any query $\mathcal{A}$ makes, $\Sigma\alpha_{x,z}|x,z\rangle$, $\mathcal{B}$ responds with $\Sigma\alpha_{x,z}|x,\mathcal{O}(x)\oplus z\rangle$.

Construction of Existential Forger $\mathcal{B}$

1. $\mathcal{B}$ receives a public key pk from the challenger $\mathcal{C}$
2. $\mathcal{B}$ simulates a variant of the strongly-unforgeable game with $\mathcal{A}$:
   i) $\mathcal{B}$ generates $(h, td) \leftarrow HG(1^n)$. Initiate $\mathcal{A}$ with $pk' = (pk, h)$
   ii) $\mathcal{B}$ simulates a random-oracle using a $2q$-wise independent hash function.
   iii) On the ith signing query M_i from $\mathcal{A}$, $\mathcal{B}$ chooses a random C_i. It then signs C_i by submitting it to $\mathcal{C}$, obtaining σ_i. It computes $m_i = \mathcal{O}(M_i||\sigma_i)$, and using the trapdoor information td, finds an r_i such that $h(m_i, r_i) = C_i$. It sends $\sigma_i' = (\sigma_i, r_i)$ to $\mathcal{A}$.
3. Let $(M^*, (\sigma^*, r^*))$ be the final forgery produced by $\mathcal{A}$. Output (C^*, σ^*) as the forgery.

  From $\mathcal{A}$'s point of view, a $2q$-wise independent function is identical to a random function [36]. Noting that $C^* \neq C_i$ for all i, and the C_i's are precisely what was submitted to $\mathcal{C}$ for signing queries, and finally, seeing as this is a valid forgery, so $V(C^*, \sigma^*) =' accept'$, we can see that $\mathcal{B}$ submits $(C^*, \sigma*)$ as a valid new forgery, breaking the existential unforgeability of Σ and winning his game with $\mathcal{C}$. Thus in this case whenever $\mathcal{A}$ succeeds, so does $\mathcal{B}$, and so the probability $\mathcal{B}$ succeeds given we are in this case is ϵ.
- **Case 2**: This case is defined as occurring when $C^* = C_i$ for some i. In this case we will show a reduction to break the collision resistance of the chameleon hash function.
  It is easy to see that $\mathcal{B}$ finds a valid collision as long as $\mathcal{A}$ produces a valid forgery, with overwhelming probability. This is because if $C^* = C_i$, then $h(\mathcal{O}(M^*||\sigma^*), r^*) = h(\mathcal{O}(M_i||\sigma_i), r_i)$. We simply need to ensure that this is not a trivial collision. Note that since this must be a new forgery, $(M^*, \sigma^*, r^*) \neq (M_i, \sigma_i, r_i)$. If $r^* \neq r_i$, we are done. Otherwise, we can see that $M^*||\sigma^* \neq M_i||\sigma_i$, and thus since the values for $\mathcal{O}(M_i||\sigma_i)$ were chosen uniformly at random, $\mathcal{O}(M^*||\sigma^*) \neq \mathcal{O}(M_i||\sigma_i)$ with overwhelming probability.
  Therefore if we let EVT be the event that $\mathcal{A}$ produces a valid forgery, we only need to show that EVT occurs with probability $\Omega(\varepsilon)$ in the construction of $\mathcal{B}$. We prove it by a hybrid argument which transforms the standard strongly unforgeable game into the

Construction of Collision-Finding Adversary $\mathcal{B}$

1. $\mathcal{B}$ receives h from the challenger, which is sampled from the Chameleon hash function family.
2. $\mathcal{B}$, playing the role of a challenger, simulates a variant of the strongly-unforgeable game with $\mathcal{A}$:
   i) $\mathcal{B}$ generates $(pk, sk) \leftarrow G(1^n)$. Initialize $\mathcal{A}$ with $pk' = (pk, h)$. For $i = \{1, \ldots, q\}$, $\mathcal{B}$ generates m_i uniformly at random and $r_i \leftarrow \mathcal{R}$ (according to the specification of h). $\mathcal{B}$ computes $C_i := h(m_i, r_i)$ and $\sigma_i := S(sk, C_i)$.
   ii) $\mathcal{B}$ simulates a random-oracle in the usual way (i.e. t-wise independent hash function).
   iii) On the ith signing query M_i from $\mathcal{A}$, $\mathcal{B}$ reprograms the random-oracle: $\mathcal{O}(M_i \| \sigma_i) \leftarrow m_i$ and returns (σ_i, r_i) to $\mathcal{A}$.
3. Let $(M^*, (\sigma^*, r^*))$ be the final forgery produced by $\mathcal{A}$. We know $C^* = C_i$ for some i. Output $(\mathcal{O}(M^* || \sigma^*), r^*), (\mathcal{O}(M_i || \sigma_i), r_i)$ as the collision.

variant as in the construction of $\mathcal{B}$. We will show that the probablity of EVT is esstially preserved in the hybrid argument.

Let Hyd_0 the standard strongly-unforgeable game with $\mathcal{A}$. By hypothesis $\Pr[\mathsf{EVT} : \mathsf{Hyd}_0] \geq \varepsilon$. Consider the first hybrid Hyd_1 that makes only one change to Hyd_0: when the challenger answers a signing query, instead of querying the random-oracle $\mathcal{O}$ to obtain $m_i := \mathcal{O}(M_i \| \sigma_i)$, it samples a random m_i and programs the random oracle so that $\mathcal{O}(M_i \| \sigma) = m_i$. Note that in particular the challenger still uses the trapdoor to find $r_i \leftarrow h^{-1}(C_i, m_i)$. By Lemma 5, we claim that[2] $\Pr[\mathsf{EVT} : \mathsf{Hyd}_0] - \Pr[\mathsf{EVT} : \mathsf{Hyd}_1]| \leq \mathsf{negl}(n)$. Specifically we instantiate Samp as follows. pk will consists of a public key for Σ, hash function h, and random messages C_i. P will be the verification algorithm of Σ. $w := \sigma_i = S(sk, C_i)$ is the signature generated by $\mathcal{B}$ in 2.i), and W_{pk} consists of all strings that form a valid signature of C_i under Σ. $\mathsf{WS}(\mathsf{Samp})$ is hard because Σ is existential-unforgeable.

Hyd_2 is obtained by a small change in Hyd_1. Instead of sampling a random C_i, it is obtained by computing $h(m_i, r_i)$ from random (m_i, r_i). This change only causes (statistically) a negligible error. This is because if $h \leftarrow \mathcal{H}$ and $r_i \leftarrow \mathcal{R}$ then $C_i := h(m_i, r_i)$ will be uniformly random by the uniformity property of $\mathcal{H}$. In addition the chameleon property of $\mathcal{H}$ tells us that $r_i \leftarrow h_{td}^{-1}(C_i, m_i)$ is distributed statistically close to sampling $r_i \leftarrow \mathcal{R}$. Therefore the order of generating C_i and r_i does not matter.

Thus we see that $\mathcal{B}$ is able to break the collision-resistance property of the Chameleon hash function.

In sum, we have shown that if there is an adversary $\mathcal{A}$ breaking Σ', then there is an adversary who manages to break either the collision resistance of the chameleon hash function $\mathcal{H}$, or the existential unforgeability of the original signature scheme Σ with probablity $\Omega(\varepsilon)$. This contradicts the security of Σ and $\mathcal{H}$ if $\varepsilon \geq 1/poly(n)$. Thus we conclude that Theorem 4 holds.

[2] More precisely, we need to introduce sub-hybrids and each sub-hybrid makes such a change for just one signing query.

4 Discussion

Obtaining a quantum-safe su-acma **signature scheme**. In [10], the authors presented a scheme for generating chameleon hash functions, based off the short integer solution problem for lattices. They also demonstrate a reduction showing an efficient algorithm to break the collision resistance of the hash function implies an efficient algorithm to break the short integer solution problem for lattices. Using results from [28] this reduction can be shown to carry through to the quantum setting. As this problem is currently believed to be hard even for quantum computers, these chameleon hash functions' collision resistance remains even when faced with a quantum adversary. This chameleon hash function scheme can therefore be used in the transformation in this paper to get a quantum-secure transformation. This transformation, used with any quantum-safe eu-acma signature scheme will give a quantum-safe su-acma scheme in the quantum random-oracle model.

When implementing the scheme with the chameleon hash function from [10] we can see what the overhead would be in an actual realization. Let $n \geq 1, q \geq 2$, and $m = O(n \log q)$. Let k be the output length of the hash function. Then the public key, pk' will now carry with it a $\mathbb{Z}_q^{n \times m}$ matrix, so $|pk'| = |pk| + n(k+m)$. The secret key now includes a specialized lattice basis, which can be written as an $m \times m$ matrix over $\mathbb{Z}_q$, giving us $|sk'| = |sk| + m^2$. Finally, the signature overhead is the inclusion of a vector in $\mathbb{Z}_q^m$, so $|\sigma'| = |\sigma| + m$.

A signature scheme based off the Short Integer Solution problem for lattices is also presented in [10]. Examining the proof presented there with tools from [28], we can see that this signature scheme is quantum-safe eu-acma. Applying this transformation to this scheme, we obtain a quantum-safe su-acma signature scheme. In fact, we can show that the reduction shown in [10] is not as tight as it could be, and for a message of length k and at most Q queries, we can show that for adversary $\mathcal{F}$ and reduction $\mathcal{S}$, we have that $\mathrm{ADV}_{SIS}(\mathcal{S}^{\mathcal{F}}) \geq \mathrm{ADV}(\mathcal{F})^{\text{eu-acma}}_{SIG}/(Q(k - \log Q))$. This is a small improvement over the result of the paper, showing that $\mathrm{ADV}_{SIS}(\mathcal{S}^{\mathcal{F}}) \geq \mathrm{ADV}(\mathcal{F})^{\text{eu-acma}}_{SIG}/(Q(k-1)+1)$

Future directions. Our work has studied a very specific transformation that gives a systematic way of getting quantum-safe su-acma signatures. There are a few more transformations in the plain model (i.e. without a random-oracle) [29, 20, 18, 17]. We conjecture that they also hold against quantum adversaries. If this is the case, it will be meaningful to evaluate all these transformations and figure out which one is preferable under specific applications. On the other hand, we chose the Bonsai-tree signature scheme [10] to instantiate the TOO transformation. There are many recent improvements on lattice-based signatures in terms of key size and computational efficiency [9, 24, 14], which are shown to be eu-acma classically. If they can be shown to be quantum-safe, they we can get more efficient quantum-safe su-acma schemes in the quantum random-oracle model.

Acknowledgements

The authors are grateful to Andrew Childs for helpful discussions. EE was supported by NSERC on an undergraduate research award at the Institute for Quantum Computing, University of Waterloo. FS acknowledges support from NSERC, CryptoWorks21, ORF and US ARO.

References

1 Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *Foundations of Computer Science*

(FOCS), 2014 IEEE 55th Annual Symposium on, pages 474–483. IEEE, 2014.

2 Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Advances in Cryptology–CRYPTO 2000*, pages 255–270. Springer, 2000.

3 Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.

4 Daniel J Bernstein, Johannes Buchmann, and Erik Dahmen. *Post-quantum cryptography*. Springer, 2009.

5 Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Advances in Cryptology–CRYPTO 2004*, pages 41–55. Springer, 2004.

6 Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2006.

7 Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *Advances in Cryptology–ASIACRYPT 2011*, pages 41–69. Springer, 2011.

8 Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In *Proceedings of CRYPTO 2013*, 2013.

9 Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *Public Key Cryptography–PKC 2010*, pages 499–517. Springer, 2010.

10 David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of cryptology*, 25(4):601–639, 2012.

11 Claude Crépeau, Paul Dumais, Dominic Mayers, and Louis Salvail. Computational collapse of quantum state with application to oblivious transfer. In *Theory of Cryptography (TCC)*, pages 374–393. Springer, 2004.

12 Özgür Dagdelen, Marc Fischlin, and Tommaso Gagliardoni. The fiat–shamir transformation in a quantum world. In *Advances in Cryptology-ASIACRYPT 2013*, pages 62–81. Springer, 2013.

13 Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM review*, 45(4):727–784, 2003.

14 Léo Ducas and Daniele Micciancio. Improved short lattice signatures in the standard model. In *Advances in Cryptology–CRYPTO 2014*, pages 335–352. Springer, 2014.

15 David Galindo, Javier Herranz, and Eike Kiltz. On the generic construction of identity-based signatures with additional properties. In *Advances in Cryptology–ASIACRYPT 2006*, pages 178–193. Springer, 2006.

16 Sean Hallgren, Adam Smith, and Fang Song. Classical cryptographic protocols in a quantum world. In *Advances in Cryptology–CRYPTO 2011*, pages 411–428. Springer, 2011.

17 Qiong Huang, Duncan S Wong, Jin Li, and Yi-Ming Zhao. Generic transformation from weakly to strongly unforgeable signatures. *Journal of Computer Science and Technology*, 23(2):240–252, 2008.

18 Qiong Huang, Duncan S Wong, and Yiming Zhao. Generic transformation to strongly unforgeable signatures. In *Applied Cryptography and Network Security*, pages 1–17. Springer, 2007.

19 Hugo Krawczyk and Tal Rabin. Chameleon hashing and signatures. In *Proc. of NDSS*, pages 143–154, 2000.

20 Jin Li, Kwangjo Kim, Fangguo Zhang, and Duncan S Wong. Generic security-amplifying methods of ordinary digital signatures. In *Applied Cryptography and Network Security*, pages 224–241. Springer, 2008.

21 Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *Advances in Cryptology–ASIACRYPT 2009*, pages 598–616. Springer, 2009.

22 Vadim Lyubashevsky. Lattice signatures without trapdoors. In *Advances in Cryptology–EUROCRYPT 2012*, pages 738–755. Springer, 2012.

23 Vadim Lyubashevsky and Daniele Micciancio. Asymptotically efficient lattice-based digital signatures. In *Theory of Cryptography*, pages 37–54. Springer, 2008.

24 Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Advances in Cryptology–EUROCRYPT 2012*, pages 700–718. Springer, 2012.

25 Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009.

26 Markus Rückert. Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles. In *Post-Quantum Cryptography*, pages 182–200. Springer, 2010.

27 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.

28 Fang Song. A note on quantum security for post-quantum cryptography. In *Post-Quantum Cryptography*, pages 246–265. Springer, 2014.

29 Ron Steinfeld, Josef Pieprzyk, and Huaxiong Wang. How to strengthen any weakly unforgeable signature into a strongly unforgeable signature. In *Topics in Cryptology–CT-RSA 2007*, pages 357–371. Springer, 2006.

30 Isamu Teranishi, Takuro Oyama, and Wakaha Ogata. General conversion for obtaining strongly existentially unforgeable signatures. In *Progress in Cryptology-INDOCRYPT 2006*, pages 191–205. Springer, 2006.

31 Dominique Unruh. Quantum proofs of knowledge. In *Eurocrypt 2012*, volume 7237 of *LNCS*, pages 135–152. Springer, April 2012. Preprint on IACR ePrint 2010/212.

32 Dominique Unruh. Quantum position verification in the random oracle model. In *Crypto 2014*, volume 8617 of *LNCS*, pages 1–18. Springer, August 2014. Preprint on IACR ePrint 2014/118.

33 Dominique Unruh. Revocable quantum timed-release encryption. In *Advances in Cryptology–EUROCRYPT 2014*, pages 129–146. Springer, 2014.

34 Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *Advances in Cryptology-EUROCRYPT 2015*, pages 755–784. Springer, 2015.

35 John Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009.

36 Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In *Proceedings of CRYPTO 2012*, 2012.

A Classical Proof

Let $\mathcal{A}$ be the forger, $\mathcal{B}$ the reduction, and $\mathcal{C}$ be the challenger. In each case, $\mathcal{B}$ and $\mathcal{A}$ will be playing a game of strong unforgeability. Let the probability that $\mathcal{A}$ succeeds be ϵ. In Case 1, $\mathcal{C}$ and $\mathcal{B}$ will play a game of existential unforgeability on the signature scheme σ. In case 2, $\mathcal{C}$ and $\mathcal{B}$ will play a game of collision resistance on the chameleon hash function h. We show that if the probability $\mathcal{A}$ succeeds in her forgery is ϵ, then the probability that $\mathcal{B}$ succeeds is $\geq \frac{1}{2}\epsilon - \mathsf{negl}(n)$. At the beginning of the reduction, $\mathcal{B}$ will flip a coin, and guess which case the adversary's forgery will fall under. Clearly, $\mathcal{B}$ will be correct with probability $\frac{1}{2}$.

In our reduction, let the forgery that $\mathcal{A}$ eventually submits be $(M^*, \sigma'^* = (\sigma^*, r^*))$ Let $C^* = h(\mathcal{O}(M^*||\sigma^*), r^*)$. Similarly, for each M_i the forger submits to the signing oracle for signing, there is an associated σ_i' and C_i.

- **Case 1**: $C^* \neq C_i$ for all i. We show that whenever the forger succeeds in creating a valid forgery of this type, the reduction succeeds in breaking the existential unforgeability of the original scheme $\Sigma = (G, S, V)$.
  $\mathcal{C}$ and $\mathcal{B}$ will be playing a game of existential unforgeability, while $\mathcal{B}$ and $\mathcal{A}$ will be playing a game of strong unforgeability. We will show that whenever $\mathcal{A}$ wins her game, $\mathcal{B}$ wins his (so long as the forgery is of the type described above).
  The games will play out as follows:
  Firstly, $\mathcal{B}$ will act as the random oracle for $\mathcal{A}$. In the first case at least (and this will change only slightly case to case), he can do this in the following way. Whenever $\mathcal{A}$ queries the random oracle with a query, $\mathcal{B}$ looks up in a maintained table if that query has been made before. If it has, he responds with the value he responded with before. If it has not, he generates a random number and responds with that.
  Now we discuss how the game of strong unforgeability transpires.
  $\mathcal{C}$ sends $\mathcal{B}$ a public key pk from the Σ scheme. $\mathcal{B}$ will generate a chameleon hash function h, (with corresponding trapdoor td) and send the public key and hash function to $\mathcal{A}$ as $pk' = (pk, h)$.
  $\mathcal{A}$ will start submitting messages M_i to $\mathcal{B}$ for signing. For each query, $\mathcal{B}$ does the following:

  - Choose a random $\tilde{m}_i$ and $\tilde{r}_i$ and compute $C_i = H(\tilde{m}_i, \tilde{r}_i)$
  - Sign C_i by submitting it to $\mathcal{C}$ as a signing query, obtaining σ_i
  - Query $M_i||\sigma_i$ to the random oracle, obtaining $m_i = \mathcal{O}(M_i||\sigma_i)$
  - Using the trapdoor information td, find an r_i such that $h(m_i, r_i) = C_i$.
  - $\sigma_i' = (\sigma_i, r_i)$
  - Send σ_i' to $\mathcal{A}$

  Eventually, $\mathcal{A}$ will submit a valid forgery $M^*, \sigma'^* = (\sigma^*, r^*)$.
  Then, $\mathcal{B}$ takes these, and computes $C^* = h(\mathcal{O}(M^*||\sigma^*), r^*)$.
  Noting that $C^* \neq C_i$ for all i, and the C_i's are precisely what was submitted to $\mathcal{C}$ for signing queries, and finally, seeing as this is a valid forgery, so $V(C^*, \sigma^*) =' \mathit{accept}'$, we can see that $\mathcal{B}$ submits $C^*, \sigma*$ as a valid new forgery, breaking the existential unforgeability of Σ and winning his game with $\mathcal{C}$.
  Thus in this case whenever $\mathcal{A}$ succeeds, so does $\mathcal{B}$, and so the probability $\mathcal{B}$ succeeds given we are in this case is ϵ.
- **Case 2**: This case is defined as occurring when $C^* = C_i$ for some i. In this case we will show a reduction to break the collision resistance of the chameleon hash function.
  To start with, $\mathcal{C}$ sends $\mathcal{B}$ the description of a chameleon hash function h, which $\mathcal{B}$ will find a collision for.
  $\mathcal{B}$ then runs the key generation algorithm of the signature scheme Σ, obtaining (pk, sk). He then sends $pk' = (pk, h)$ to $\mathcal{A}$.
  For each signing query M_i that $\mathcal{A}$ sends to $\mathcal{B}$, $\mathcal{B}$ does the following:

  - Choose a random m_i and r_i and compute $C = h(m_i, r_i)$
  - Sign C_i using the signing algorithm S, obtaining $\sigma = S(C, sk)$
  - Reprogram the random oracle so that $\mathcal{O}(M_i||\sigma_i) = m_i$.
  - $\sigma_i' = (\sigma_i, r_i)$
  - Send σ_i' to $\mathcal{A}$.

Note that we have now permitted $\mathcal{B}$ to reprogram the random oracle for the purposes of this proof. Thus it is necessary to show that $\mathcal{A}$ will still output a valid forgery.

When $\mathcal{A}$ eventually submits her forgery, (M^*, σ^*), we can see that $C^* = C_i$ for some i. This implies that $h(\mathcal{O}(M_i||\sigma_i), r_i) = h(\mathcal{O}(M^*||\sigma^*), r^*)$ for that i. This shows us a collision for the chameleon hash function h, which is what $\mathcal{B}$ is looking for. But we must take care to ensure that it isn't a trivial collision.

Note that $(M_i, \sigma_i, r_i) \neq (M^*, \sigma^*, r^*)$, simply because both the message and signature of the forgery can't be the same as that of one of the M_i's. So at least one of these values is different.

If $r_i \neq r^*$, we are done. Otherwise, it must be the case that $M^*||\sigma^* \neq M_i||\sigma_i$. In this case, since the values for the random oracle are chosen uniformly at random, with overwhelming probability, $\mathcal{O}(M^*||\sigma^*) \neq \mathcal{O}(M_i||\sigma_i)$, giving $\mathcal{B}$ a collision for h.

So in this case, $\mathcal{B}$ will succeed as long as $\mathcal{A}$ does up to a negligible probability by Lemma 7.

So the probability $\mathcal{B}$ succeeds is $\geq \epsilon - \mathsf{negl}(n)$

▶ **Lemma 7.** *For a forger $\mathcal{A}$, let $\mathcal{B}_1$ and $\mathcal{B}_2$ be as below, and have them play a game of strong unforgeability with $\mathcal{A}$. Then*

$$|Pr_{\mathcal{B}_1}(\mathcal{A} \text{ wins}) - Pr_{\mathcal{B}_2}(\mathcal{A} \text{ wins})| \leq \mathsf{negl}(n),$$

as long as the underlying signature scheme is existentially unforgeable.

$\mathcal{B}_1$ is defined to operate exactly as the transformation dictates. $\mathcal{B}_2$ will operate as $\mathcal{B}$ was defined to in Case 2 above.

Proof. Say the difference in probability that $\mathcal{A}$ wins was not negligible. As the distribution of all values is the same, the only difference from $\mathcal{A}$'s perspective was that the value of $\mathcal{O}(M_i||\sigma_i)$ was changed for each i.

But clearly the only way to have the information that they changed is if $\mathcal{A}$ had already queried $\mathcal{O}(M_i||\sigma_i)$. But if $\mathcal{A}$ does this with non-negligible probability, then we could construct a reduction to break the existential forgeability of the signature scheme by playing strong unforgeability with $\mathcal{A}$, and before submitting each C_i to the signing oracle, checking to see if $\mathcal{A}$ had queried $M_i||\sigma_i$ to the random oracle. With non-negligible probability, the reduction finds a σ_i that is a valid forgery. So he submits this along with C_i and has broken the existential unforgeability of the scheme. ◀

Therefore in both cases, as long as $\mathcal{B}$ successfully guesses which case the forgery will fall under, he manages to successfully break either the collision resistance of the chameleon hash function h, or the existential unforgeability of the original signature scheme Σ. Since $\mathcal{B}$ correctly guesses what case he is in half of the time, his probability of success is $\geq \frac{1}{2}\epsilon - \mathsf{negl}(n)$.

B Proof of Lemma 6

Proof. Let $\mathcal{A}$ be an arbitrary algorithm running in G (or G'). Consider another algorithm B that runs in an experiment EXT as follows:

Let $p_B := \Pr_{EXT}[z \in W_{pk}]$ be the probability that the output of E is a valid witness. Let $\epsilon := |\Pr_G[b = 1] - \Pr_{G'}[b = 1]|$. In both experiment G and G', pk is selected at random according to Samp. Let P_{pk} be the probability that pk is outputted. Then

Extraction Experiment EXT

1. Generate $(pk, w, P) \leftarrow \mathsf{Samp}(1^n)$. Ignore w.
2. B receives pk and picks $j \in_R \{1, \ldots, q\}$ at random.
3. B simulates $\mathcal{A}$ on pk and (quantum) access to f. Just before $\mathcal{A}$ making the jth query to f, B measures the register that contains $\mathcal{A}$'s query. Let z be the measurement outcome.

$$\begin{aligned}
\epsilon &= \left| \Pr_G[b=1] - \Pr_{G'}[b=1] \right| \\
&= \left| \sum_{pk} \Pr_G[b=1|pk] \cdot P_{pk} - \sum_{pk} \Pr_{G'}[b=1|pk] \cdot P_{pk} \right| \\
&= \sum_{pk} P_{pk} \left| \Pr_G[b=1|pk] - \Pr_{G'}[b=1|pk] \right| .
\end{aligned}$$

Let $\epsilon_{pk} := |\Pr_G[b=1|pk] - \Pr_{G'}[b=1|pk]|$. Let $|\phi_i\rangle$ be the superposition of $\mathcal{A}^G$ on input pk when the i'th query is made. Then let $q_y(|\phi_i\rangle)$ be the sum of squared magnitudes in $\mathcal{A}$ querying the oracle on the string y.

Let $S = [q] \times W_{pk}$. Let $\delta_{pk} = \sum_{(i,y)\in S} q_y(|\phi_i^{pk}\rangle)$. We employ a theorem by Bennet et al. [3], that states that $\| |\phi_i^{pk}\rangle - |\tilde{\phi}_i^{pk}\rangle \| \leq \sqrt{q \cdot \delta_{pk}}$. (Here $|\tilde{\phi}_i^{pk}\rangle$ is defined in the same way as $|\phi_i^{pk}\rangle$ but with G' rather than G).

The same paper [3] also bounds the probability of being able to distinguish the two states, which corresponds to our probability of distinguishing the two experiments, ϵ_{pk}, telling us that

$$\epsilon_{pk} \leq 4 \cdot \left\| |\phi_i^{pk}\rangle - |\tilde{\phi}_i^{pk}\rangle \right\| \leq 4\sqrt{q \cdot \delta_{pk}} .$$

Now note that P_B^{pk} (that is, the probability that EXT outputs a valid witness given pk is chosen) can be written as

$$\begin{aligned}
P_B^{pk} &= \sum_{i\in[0,q]} \left(\Pr[i \text{ chosen}] \cdot \sum_{(j,y)\in S : j=i} q_y(|\phi_j^{pk}\rangle) \right) \\
&= \frac{1}{q} \sum_{i\in[0,q]} \sum_{(j,y)\in S : j=i} q_y(|\phi_j^{pk}\rangle) \\
&= \frac{1}{q} \sum_{(i,y)\in S} q_y(|\phi_i^{pk}\rangle) = \frac{1}{q}\delta_{pk}
\end{aligned}$$

So we can see that $\epsilon_{pk} \leq 4q\sqrt{P_B^{pk}}$. Then

$$\epsilon = \sum_{pk} P_{pk}\epsilon_{pk} \leq 4q \sum_{pk} P_{pk}\sqrt{P_B^{pk}} \overset{(*)}{\leq} 4q\sqrt{\sum_{pk} P_{pk} P_B^{pk}} = 4q\sqrt{P_B} ,$$

where (*) applies Jensen's inequality. Finally, notice that B can be viewed as an adversary in the witness-search game $\mathsf{WS}(\mathsf{Samp})$. Therefore, we conclude that $p_B \leq \mathsf{negl}(n)$ by the hypothesis that $\mathsf{WS}(\mathsf{Samp})$ is hard and hence $|\Pr_G[b=1] - \Pr_{G'}[b=1]| \leq \mathsf{negl}(n)$. ◀