On the (In)Security of the BUFF Transform

Abstract. The BUFF transform is a generic transformation for digital signature schemes, with the purpose of obtaining additional security properties beyond standard unforgeability, e.g., *exclusive ownership* and *non-resignability*. In the call for additional post-quantum signatures, these were explicitly mentioned by the NIST as "*additional desirable security properties*", and some of the submissions indeed refer to the BUFF transform with the purpose of achieving them, while some other submissions follow the design of the BUFF transform without mentioning it explicitly.

In this work, we show the following negative results regarding the non-resignability property in general, and the BUFF transform in particular. In the plain model, we observe by means of a simple attack that any signature scheme for which the message has a high entropy given the signature does not satisfy the non-resignability property (while non-resignability is trivially not satisfied if the message can be efficiently computed from its signature). Given that the BUFF transform has high entropy in the message given the signature, it follows that the BUFF transform does *not* achieve non-resignability whenever the random oracle is instantiated with a hash function, no matter what hash function.

When considering the random oracle model (ROM), the matter becomes slightly more delicate since prior works did not rigorously define the non-resignability property in the ROM. For the natural extension of the definition to the ROM, we observe that our impossibility result still holds, despite there having been positive claims about the non-resignability of the BUFF transform in the ROM. Indeed, prior claims of the non-resignability of the BUFF transform rely on faulty argumentation.

On the positive side, we prove that a *salted* version of the BUFF transform satisfies a slightly *weaker* variant of non-resignability in the ROM, covering both classical and quantum attacks, *if* the entropy requirement in the (weakened) definition of non-resignability is statistical; for the computational variant, we show yet another negative result.

1 Introduction

1.1 Non-Resignability and the BUFF Transform

Since their introduction in the seminal work by Diffie and Hellman [DH76] as a concept, and by Rivest, Shamir, and Adleman [RSA78] with the first proposed instantiation, *digital signature schemes* play an indispensable role in modern cryptography, both in the theory of cryptography and in practical applications.

The gold standard security property for digital signatures is *unforgeability* (under chosen message attacks), demanding that it is hard to produce a valid signature without knowledge of the secret key. However, in certain situations,

additional security properties may be desirable, like *exclusive ownership* [PS05], *message-bound signatures*, and *non-resignability* [JCCS19]. The NIST explicitly mentioned them as "*additional desirable security properties*" in their call for additional post-quantum signatures [NIST22]. As discussed in [CDF+21], there are real-life attacks in certain applications that exploit the lack of these additional security properties.

Non-resignability for example requires, informally, that it is hard for an attacker to maul a signature σ for a message m into a signature $\overline{\sigma}$ under his own public-key, when he only gets to see the signature σ of m (under someone else's public key) and some auxiliary information on m, but not the message m itself. The relevance of non-resignability has been shown in [JCCS19], where the authors identified an attack against the "Dynamically Recreatable Key" protocol [KBJ+14] that indeed applies in case the deployed signature scheme does not satisfy non-resignability, uncovering a flaw in the protocol's original security analysis [ZBPB17].

On top of discussing these additional security properties and their relevance in applications, Cremers, Düzlü, Fiedler, Fischlin, and Janson [CDF+21] offer a generic transformation, the *BUFF transform* (which stands for **B**eyond **U**n**F**orgeability **F**eatures), that turns any signature scheme into a new signature scheme that is argued to then satisfy these additional security properties either in the random oracle model (ROM) or in the plain model under some non-standard assumptions on the hash function.[1]

Motivated by the reference in the NIST call and the little overhead caused by the BUFF transform, several of the submissions to the NIST call for additional post-quantum signatures have the BUFF transform built in, or mention the possibility of applying the BUFF transform to the proposed scheme.[2]

There have also been some claims about (some of) these additional security properties being achieved by the three signature schemes that were selected by the NIST in 2022 to be standardized. Indeed, Cremers *et al.* [CDF+21] argue that Dilithium [LDK+20] uses the BUFF transform *implicitly*, allowing them to apply their main result regarding the BUFF transform. Falcon [PFH+20] does not achieve the beyond unforgeability features; however, the Falcon team announced that they will deploy the BUFF transform to achieve them [FHK+22]. Finally, Cremers *et al.* expect that Sphincs$^+$ [HBD+20] also achieves non-resignability, though only using some informal arguments.

[1] The original publication [CDF+21] has been revised in reaction to this work; our discussion here is with respect to the original version of [CDF+21]; we discuss the revision [CDF+23] explicitly in Sect. 1.3.

[2] The following submissions explicitly refer to the BUFF transform: Squirrels [ENST23], Racoon [dPEK+23], HAWK [BBD+23], PROV [GCF+23], Vox [PCF+23], and eMLE [LZ23].

1.2 Our Results.

In this work, we give both negative and positive results on the non-resignability property in general, and the BUFF transform in particular, as discussed below in more detail.

Negative Results. First, we consider the plain model, and we observe that for any signature scheme with the property that there is sufficient (computational) entropy in the message when given its signature (and the public key), there is a simple attack that entirely breaks non-resignability of the signature scheme.[3]

Given that, by design, the BUFF transform satisfies this entropy requirement, it follows directly that the BUFF transform does *not* satisfy non-resignability in the plain model, regardless of the hash function used (and regardless of the hash function being fixed or chosen from a family of possible hash functions). We stress that not only is there no proof for the non-resignability of the BUFF transform in the plain model, but our aforementioned attack easily breaks it.

Moving to the random oracle model (ROM), somewhat surprising in the light of the positive results claimed in [CDF+21] on the BUFF transform in the ROM, we show that, as a matter of fact, also in the ROM the BUFF transform does not (necessarily) satisfy non-resignability. The matter is slightly more subtle here since prior works did not rigorously define the non-resignability property in the ROM. What we show is that for the *natural extension* of the non-resignability property to the ROM, our negative results from the plain model carry over, and thus, in particular, that the BUFF transform does not achieve (this natural notion of) non-resignability in the ROM.

Given the positive claims in prior work, we discuss what is wrong with the reasoning in [CDF+21], where the BUFF transform is claimed to satisfy non-resignability. Namely, the issue lies in the Φ-non-malleability property of the random oracle, incorrectly claimed in [BFS11] and used in [CDF+21]. More precisely, we show that Φ-non-malleability as stated in these works is unachievable.

We note that our generic attack on the non-resignability property is embarrassingly simple in retrospect. It exploits that there is no restriction on the attacker's auxiliary information on the signed message m, subject to that it does not reveal m; this pretty much allows to embed the mauled signature $\overline{\sigma}$ into the auxiliary information, making the attacker's job of finding $\overline{\sigma}$ trivial. This attack has no (direct) real-world impact, since the auxiliary information is typically not adversarially chosen, but determined by the application. Instead, the point of our attack is to show that the formal definition put forward in [CDF+21] is too strong, and that prior positive results on achieving non-resignability are incorrect. Thus, we need to go back to the drawing board: both the formal definition as well as achievability results need to be revised. This is what we do, to a certain degree, in the main part of the paper, as discussed below.

[3] On the other hand, if the message can be efficiently computed from its signature, non-resignability is also not satisfied, as already pointed out in [CDF+21].

Positive (and More Negative) Results. Facing the above strong negative result, we introduce a weaker variant of the original definition of the non-resignability property, which is still meaningful from an application perspective yet avoids the above generic attack, by requesting the auxiliary information to be computed without access to the random oracle.[4] This definition is thus still meaningful whenever in the considered application the computation of the auxiliary information does not depend on the random oracle that is used in the signing process for the considered signature scheme (which can typically be enforced via domain separation).

A natural question then is whether the BUFF transform satisfies this weakened variant of the non-resignability property. Interestingly, this remains a non-trivial problem; as a matter of fact, depending on the precise formulation of the entropy requirement, which captures that the signed message m should remain hidden to the attacker, we show yet another negative result (see below).

On the positive side, we show that the above weakened variant of the non-resignability property is satisfied in the ROM by a *salted version* of the BUFF transform, *if* the entropy requirement on the message m is *statistical* (rather than computational). The salted version of the BUFF transform includes a random salt in the hash and appends the salt to the signature.

Our proof follows a similar blueprint as the (faulty) proof in [CDF+21]. Indeed, we first show that the reduction from [CDF+21], which reduces non-resignability of the BUFF transform to Φ-non-malleability of the random oracle (where the latter, however, is not satisfied), carries over to the *salted* BUFF transform when considering the *weaker* variant of non-resignability and a correspondingly *weaker* and *salted* variant of Φ-non-malleability, and considering the entropy requirement to be *statistical*. Then, the main technical challenge, and thus the main technical contribution of this work, lies in proving that the random oracle satisfies the considered weaker and salted variant of Φ-non-malleability.

The above positive result is proven in the classical as well as in the quantum ROM (with different respective reduction losses), covering thus both classical and quantum attacks. We note that despite the innocent-looking nature of the core problem, proving the considered Φ-non-malleability variant for the random oracle turns out to be highly non-trivial, even just in the classical case.

Yet again on the negative side, by means of a counterexample in the form of a contrived signature scheme, we show that the above result on the salted version of the BUFF transform does not carry over in case the entropy requirement on the message m is *computational* (by means of the HILL entropy), as originally considered in [CDF+21]. This in particular applies to the original (unsalted) BUFF transform.

Thus, despite our weakened version of the non-resignability property, showing positive results remains challenging. In particular, whether the original BUFF transform satisfies our weaker non-resignability notion in case the entropy requirement is statistical, remains an unsettling open question.

[4] A more radical solution is to disallow any auxiliary information altogether, which in essence is done in [CDF+23]; see later for a more elaborate discussion of [CDF+23].

Conclusion. Altogether, our work shows that the non-resignability property for digital signature schemes, introduced in [JCCS19], later formalized in [CDF+21], and mentioned by NIST as an desirable property in their call for additional post-quantum signatures, is a very delicate security notion, and whether it is achieved (by one or another construction)—or even achievable at all—depends on subtle choices in the formal definition. Furthermore, our work shows that we actually have only very limited positive results so far; in particular, there is currently no positive (meaningful) result on the non-resignability property of the original BUFF transform.

1.3 Related Work

We already mentioned [CDF+21] and [JCCS19], upon which our work builds up. In reaction to our negative results, the authors of [CDF+21] have updated their work; we briefly discuss this update [CDF+23] here.

In order to avoid our negative results (cf. Theorem 5), which exploit that the auxiliary information can be misused to embed a mauled signature, the authors modified the definition of non-resignability to require the auxiliary information to be *computationally independent* of the message (see [CDF+23, Fig. 4] for the non-resignability game and [CDF+23, Definition 4.3] for the actual definition). This is equivalent to not allowing any auxiliary information at all, and thus weaker than the variant of non-resignability we consider. Indeed, in the security reduction it is argued that, due to the computational independence, one can drop the auxiliary information altogether, and so reduce the non-resignability of the original BUFF transform to a variant of Φ-non-malleability with no auxiliary information (see [CDF+23, Fig. 1, right]).

Interestingly though, the authors have not adjusted their reasoning for their claim on the random oracle satisfying (the now weaker notion of) Φ-non-malleability, which is the place where the original flaw was hiding. They still argue via the very same informal reasoning as in the original version (see the quote in our Section 3.3). Although it is tempting to believe that this argumentation is sufficient, it is actually not.

Indeed, by means of a simple counter example we show in Appendix A that *no* hash function (or hash function familiy), including the random oracle, satisfies the considered (weaker) notion of Φ-non-malleability. Thus, their claim on the BUFF transform satisfying the version of non-resignability considered in [CDF+23], under the assumption that the hash function satisfies the considered Φ-non-malleability notion, is vacuous.

1.4 Overview

Given the various dimensions involved, Table 1 provides an overview of the possibility and impossibility results that follow from our work, and of what is still open. In this overview, we distinguish between the original non-resignability security game NR^{H} (Fig. 5) as proposed in [CDF+21] (but considered in the random oracle model now), our weaker notion $\mathsf{NR}^{\mathrm{H},\perp}$ (Fig. 10, left) which requires the

auxiliary information to be computed without access to the random oracle H, and the further weakened notion $\mathsf{NR}^{\mathrm{H},0}$ where no auxiliary information at all is allowed (as considered in [CDF+23], in essence). In the other directions, we distinguish between the original and the salted BUFF transformation (respectively denoted by BUFF and $-BUFF), and between the computational and the statistical entropy requirement.

Table 1. Overview of possibility and impossibility results resulting from our work. In the table, we write H_∞ and HILL_∞ to differentiate between statistical and computation entropy, respectively. Positive results are indicated with ✓ while negative results are visualized with ✗—for both we refer to the corresponding theorem establishing that result. A **?** is used for an open question.

	NR^{H}		$\mathsf{NR}^{\mathrm{H},\perp}$		$\mathsf{NR}^{\mathrm{H},0}$	
	H_∞	HILL_∞	H_∞	HILL_∞	H_∞	HILL_∞
BUFF	✗ Thm. 7	✗ Thm. 7	?	✗ Thm. 18	?	?
$-BUFF	✗ Thm. 7	✗ Thm. 7	✓ Thm. 11	✗ Thm. 18	✓ Thm. 11	?

2 Preliminary

2.1 (HILL) Entropy

We briefly recall that, for a random variable X, specified by its probability distribution P_X, the *guessing probability* is given by $\mathsf{guess}(X) := \max_x P_X(x)$, and the *min-entropy* by $H_\infty(X) := -\log \mathsf{guess}(X)$. As usual, the log is in base 2.

In a similar spirit, for a pair of random variables (X, Z), specified by their joint distribution P_{XZ}, the *conditional guessing probability* $\mathsf{guess}(X \mid Z)$ is defined as

$$\mathsf{guess}(X \mid Z) := \sum_z P_Z(z)\mathsf{guess}(X \mid Z = z)\,,$$

with the natural understanding that $\mathsf{guess}(X \mid Z = z) = \max_x P_{X|Z}(x \mid z)$, and the *conditional min-entropy* $H_\infty(X \mid Z)$ as

$$H_\infty(X \mid Z) := -\log \mathsf{guess}(X \mid Z)\,.$$

Thus, in other words, $H_\infty(X \mid Z) := -\log \sum_z P_Z(z)2^{-H_\infty(X|Z=z)}$

The HILL entropy is a computational variant of the above min-entropy. First, we recall that for two random variables X and Y, the computational distance

$$\delta_s(X, Y) := \max_C \mid \Pr[C(X) = 1] - \Pr[C(Y) = 1] \mid$$

is the maximum distinguishing advantage over all circuits C of size s.

Definition 1. *For a pair of random variables (X, Z), specified by their joint distribution P_{XZ}, the* conditional HILL entropy *(with parameters δ and s) is defined as*

$$^{\delta,s}\mathsf{HILL}_\infty(X \mid Z) := \max_Y H_\infty(Y \mid Z)\,,$$

where the maximum is over all random variables Y, specified by its joint distribution P_{YZ} with Z, such that $\delta_s\big((X,Z),(Y,Z)\big) \le \delta$.

When switching to asymptotic notation, for a family of pairs of random variables $\{(X_\lambda, Z_\lambda)\}_{\lambda\in\mathbb{N}}$, a bound $\mathsf{HILL}_\infty(X_\lambda \mid Z_\lambda) \ge k(\lambda)$ then naturally means that for every λ there exists Y_λ such that $H_\infty(Y_\lambda \mid Z_\lambda) \ge k(\lambda)$, and for every polynomial $s(\lambda)$ there exists a negligible $\delta(\lambda)$ such that $\delta_{s(\lambda)}\big((X_\lambda, Z_\lambda),(Y_\lambda, Z_\lambda)\big) \le \delta(\lambda)$. In this case, we tend to omit the security parameter λ and simply write (X, Y) and $\mathsf{HILL}_\infty(X \mid Z) \ge k$.

2.2 Digital Signatures

We use the standard definition of a signature scheme. By default, the key-generation, signing, and verification algorithms of a signature scheme S are respectively denoted by $\mathtt{KGen}$, $\mathtt{Sign}$, and $\mathtt{Vfy}$, and the message space by $\mathcal{M}$.

In this work, we take it as understood that a signature scheme is *correct* up to a negligible error, i.e., for any $m \in \mathcal{M}$ we have

$$\Pr_{(sk,pk)\leftarrow\mathtt{KGen}(1^\lambda)}[\mathtt{Vfy}(pk, m, \mathtt{Sign}(sk, m)) = 0] \le \mathrm{negl}(\lambda)\,.$$

Furthermore, we insist on the key generation to produce a public key pk that has negligible guessing probability

$$\mathsf{guess}_{(sk,pk)\leftarrow\mathtt{KGen}(1^\lambda)}(pk) \le \mathrm{negl}(\lambda)\,;$$

this is obviously necessary for the scheme to be *secure* in any meaningful way. For a signature scheme in the random oracle model (i.e., where $\mathtt{KGen}$ may query the random oracle H), we require $\mathtt{guess}(pk \,|\, \mathrm{H})$ to be negligible.

In expressions similar to those above, we will sometimes write (sk, pk) instead $(sk, pk) \leftarrow \mathtt{KGen}(\lambda)$, taking the generation as understood.

2.3 Non-Resignability and Φ-Non-Malleability

For a signature scheme $\mathrm{S} = (\mathtt{KGen}, \mathtt{Sign}, \mathtt{Vfy})$, *non-resignability* is defined via game NR, given in Fig. 1, which involves an *adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ and a (possibly randomized) auxiliary function $\mathtt{aux}$. We say that $\mathcal{A}$ is PPT if $\mathcal{A}_0$ and $\mathcal{A}_1$ are. In spirit, the goal of the adversary is to turn a signature σ for an unknown message m into a signature for the same message, but under a different, adversarially chosen, public key. We write

$$\mathbf{Adv}^{\mathsf{NR}}_{\mathrm{S}}(\lambda, \mathcal{A}, \mathtt{aux}) = \Pr[1 \leftarrow \mathsf{NR}_{\mathrm{S}}]$$

for the probability that the NR_{S} game outputs 1 when instantiated with signature scheme S and with adversary $\mathcal{A}$ and auxiliary function $\mathtt{aux}$. Similarly, for variants of NR and for other games that we will consider. For simplicity, we will often leave the security parameter λ implicit.

Game NR_{S}

$(sk, pk) \leftarrow \mathtt{KGen}(1^{\lambda})$
$m \leftarrow \mathcal{A}_0(pk)$
$h \coloneqq \mathtt{aux}(m, pk)$
$\sigma \leftarrow \mathtt{Sign}(sk, m)$
$(\overline{\sigma}, \overline{pk}) \leftarrow \mathcal{A}_1(pk, \sigma, h)$
$v \coloneqq \mathtt{Vfy}(\overline{pk}, m, \overline{\sigma})$
return $(v = 1 \wedge \overline{pk} \neq pk)$

Fig. 1. Security game NR_{S} (in the plain model) with an explicit hint function $\mathtt{aux}$ and a signature scheme $\mathrm{S} = (\mathtt{KGen}, \mathtt{Sign}, \mathtt{Vrfy})$. The original definition in [CDF+21] had both m and h produced by $\mathcal{A}_0(pk)$. This change in the definition only makes our negative result stronger (since it is a restriction on how h is produced), and will be convenient later on when trying to restore positive results.

As pointed out in [CDF+21], an adversary $\mathcal{A}$ can easily win this game if $\mathcal{A}_1$ can compute m; indeed, it can then just sign m under a public-key $\overline{pk}$ for which it knows the secret key. Thus, for this to be a potentially hard game, we need to enforce an entropy requirement on m. In this work, we consider two variants, by requiring the *statistical* entropy $H_{\infty}(m \mid pk, h)$ or the *computational* entropy $\mathsf{HILL}_{\infty}(m \mid pk, h)$ to be lower bounded, where m, pk and h are chosen as in NR.

Following [CDF+21] (subject to this minor change in the game NR mentioned in Fig. 1), non-resignability is defined as follows.

Definition 2. *A signature scheme* $\mathrm{S} = (\mathtt{KGen}, \mathtt{Sign}, \mathtt{Vfy})$ *is called non-resignable if for any PPT adversary* $\mathcal{A}$ *and any PPT function* $\mathtt{aux}$ *that satisfy the computational entropy condition*

$$\mathsf{HILL}_{\infty}\limits_{\substack{(sk,pk)\leftarrow\mathtt{KGen}(1^{\lambda})\\ m\leftarrow\mathcal{A}_0(pk)}} \big(m \,\big|\, pk, \mathtt{aux}(m, pk)\big) \geq \omega(\log \lambda) \tag{1}$$

it holds that $\mathbf{Adv}^{\mathsf{NR}}_{\mathrm{S}}(\lambda, \mathcal{A}, \mathtt{aux}) \leq \mathrm{negl}(\lambda)$.

A related notion, which is used in [CDF+21] towards proving non-resignability of the BUFF transform (see Sect. 2.4), is Φ-non-malleability, first introduced in [BCFW09], for a (keyed) hash function, specified by a pair $(\mathtt{KGen}, \mathtt{Eval})$ of PPT key-generation and evaluation algorithms.

The definition is via game $\Phi\text{-}\mathsf{NM}_{\mathcal{F}}$, given in Fig. 2, and, as for non-resignability, we need to require a certain amount of statistical entropy $H_{\infty}(x \mid hk, h)$ or computational entropy $\mathsf{HILL}_{\infty}(x \mid hk, h)$, for the game to be non-trivial.

Game Φ-$\mathsf{NM}_{\mathcal{F}}$

$hk \leftarrow \mathtt{KGen}(1^\lambda)$
$x \leftarrow \mathcal{A}_0(hk)$
$h \leftarrow \mathtt{aux}(hk, x)$
$y \coloneqq \mathcal{F}_{hk}(x)$
$(\overline{y}, \phi) \leftarrow \mathcal{A}_1(hk, y, h)$
return $(\mathcal{F}_{hk}(\phi(x)) = \overline{y} \wedge \phi(x) \neq x)$

Fig. 2. Security game Φ-$\mathsf{NM}_{\mathcal{F}}$ (in the plain model) with explicit hint function $\mathtt{aux}$ and a keyed hash function $\mathcal{F} = (\mathtt{KGen}, \mathtt{Eval})$, where we denote $\mathcal{F}_{hk}(x) \coloneqq \mathtt{Eval}(hk, x)$.

Following [BFS11,CDF+21], for a family Φ of functions, Φ-non-malleability is defined as follows. Looking ahead, of particular interest is the case where x consists of two parts, conveniently written as $x = (pk, m)$, and Φ consists of shifts of pk but leaves m untouched.

Definition 3. *A keyed hash function $\mathcal{F} = (\mathtt{KGen}, \mathtt{Eval})$ is called* Φ-non-malleable *if for any PPT adversary $\mathcal{A}$ that satisfies the computational entropy condition*

$$\mathsf{HILL}^{\infty}_{\substack{hk \leftarrow \mathtt{KGen}(1^\lambda) \\ x \leftarrow \mathcal{A}_0(hk)}} \left(x \,\middle|\, hk, \mathtt{aux}(hk, x)\right) \geq \omega(\log \lambda) \tag{2}$$

it holds that $\mathbf{Adv}^{\Phi\text{-}\mathsf{NM}}_{\mathcal{F}}(\mathcal{A}) \leq \mathrm{negl}(\lambda)$.

2.4 The BUFF Transform

The BUFF transform [CDF+21] is a generic transform for signature schemes to achieve additional security properties beyond standard unforgeability. The transformation comes in two variants—BUFF and BUFF-lite. Throughout this work, we only consider the former, stronger variant[5] which is displayed in Fig. 3.

$\mathtt{KGen}(1^\lambda)$	$\mathtt{Sign}(sk, m)$	$\mathtt{Vfy}(pk, m, \sigma)$
$(sk_{\mathrm{S}}, pk_{\mathrm{S}}) \leftarrow \mathrm{S}.\mathtt{KGen}(1^\lambda)$	$(sk_{\mathrm{S}}, hk) \coloneqq sk$	$(pk_{\mathrm{S}}, hk) \leftarrow pk$
$hk \leftarrow \mathtt{KGen}(1^\lambda)$	$y \coloneqq \mathcal{F}_{hk}(m, pk)$	$(\sigma_{\mathrm{S}}, y) \leftarrow \sigma$
$sk \coloneqq (sk_{\mathrm{S}}, hk)$	$\sigma_{\mathrm{S}} \leftarrow \mathrm{S}.\mathtt{Sign}(sk_{\mathrm{S}}, y)$	$\overline{y} \coloneqq \mathrm{H}_{hk}(m, pk)$
$pk \coloneqq (pk_{\mathrm{S}}, hk)$	$\sigma \coloneqq (\sigma_{\mathrm{S}}, y)$	$d \coloneqq \mathrm{S}.\mathtt{Vfy}(pk_{\mathrm{S}}, y, \sigma_{\mathrm{S}})$
return (sk, pk)	**return** $(\overline{\sigma}, \overline{pk})$	**return** $d = 1 \wedge \overline{y} = y$

Fig. 3. The BUFF transform.

[5] The weaker one, BUFF-lite, does not achieve non-resignability, which is the focus of our work.

The idea of the BUFF transform is to sign the hash of the message and the public key, and to append this hash to the signature. This "binds" the public key to the signature, which ensures that no other keys can be generated that verify such a signature, thus ensuring what is known as *exclusive ownership*. The idea behind *non-resignability* is as follows. In order to turn a signature into a new signature for the same message but under a different public key $\overline{pk}$, the adversary needs to produce the hash value $\overline{y} \coloneqq \mathcal{F}_{hk}(m, \overline{pk})$ for the modified public key $\overline{pk}$ and the unknown message m, which should be hard by the Φ-non-malleability of the hash function. Indeed, formally, the following is proven (where we omit the claims regarding further security properties that are not relevant to our work).

Theorem 4 ([CDF+21, Theorem 5.5]). *Let* S *be an EUFCMA-secure signature scheme. Then the application of the BUFF transformation produces an EUFCMA-secure signature scheme* $\mathrm{BUFF}[\mathrm{S}, \mathrm{H}]$ *that additionally provides [...] NR if* H *is* Φ*-non-malleable where* $\Phi = \{\phi_{\overline{pk}} \mid \overline{pk} \in \mathcal{K}\}$ *and* $\phi_{\overline{pk}}(pk, m) = (\overline{pk}, m)$.

In combination with the claim on the random oracle being Φ-non-malleable for this choice Φ from [BFS11], the authors of [CDF+21] then conclude non-resignability of the Buff transform in the ROM.

3 On the Impossibility of Non-Resignability

In this section, we show strong negative results on the non-resignability property in general, and on the BUFF transform in particular.

First, we consider the plain model, where we show, by means of a simple attack, that non-resignability is not satisfied when applied to a signature scheme with the property that the message has high (computational) entropy when given its signature (and the public key).[6] Since the BUFF transform, when applied to any signature scheme, satisfies the considered entropy requirement (assuming the hash function to be compressing), it follows directly that the BUFF transform does *not* satisfy non-resignability in the plain model, regardless of the hash function used. We stress that not only is there no proof for the non-resignability of the BUFF transform in the plain model, but there is also an attack that breaks it.

These negative results from the plain model carry over to the random oracle model (ROM) when considering the *natural extension* of the non-resignability property to the ROM (prior works did not rigorously specify the property in the ROM). Thus, also in the ROM the BUFF transform does not (necessarily) satisfy non-resignability, invalidating the positive results claimed in [CDF+21] in that respect. In essence, the claim on the random oracle being Φ-non-malleable, made in [CDF+21,BFS11], is false.

[6] In the other extreme, if the message can be efficiently computed from its signature, non-resignability is also not satisfied, as already pointed out in [CDF+21].

3.1 Non-Resignability and BUFF Transform in the Plain Model

It is clear that the NR_{S} game (Fig. 1) is easy to win if the message m can be efficiently computed from its signature σ. In the following theorem, we show that if, on the other hand, the signature scheme is such that the message m has high computational entropy given its signature (and the public key), then another attack applies.[7]

Theorem 5. *Let* S *be a signature scheme such that for message* $\overline{m} \leftarrow \mathcal{M}$ *and key-pair* $(\overline{sk}, \overline{pk}) \leftarrow \mathtt{KGen}(1^\lambda)$ *we have* $\mathsf{HILL}_\infty(\overline{m} \mid \overline{pk}, \mathtt{Sign}(\overline{sk}, m)) \geq \omega(\log \lambda)$. *Then there exists a PPT adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ *and a PPT function* $\mathtt{aux}$ *such that the computational entropy condition* (1) *is satisfied, yet*

$$\mathbf{Adv}^{\mathsf{NR}}_{\mathrm{S}}(\lambda, \mathcal{A}, \mathtt{aux}) \geq 1 - \mathrm{negl}(\lambda)\,.$$

The attack is surprisingly simple. Instead of burdening $\mathcal{A}_1$ with finding $\overline{\sigma}$, which, intuitively, is hard since $\mathcal{A}_1$ does not know m, we simply let $\mathtt{aux}$ compute $\overline{\sigma}$ and hand it over to $\mathcal{A}_1$ as auxiliary information h. The entropy condition on the signature scheme then ensures that this is an eligible attack. The proof below spells out the details.

Proof. We construct the adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ and function $\mathtt{aux}$ as shown in Fig. 4. In the first stage, $\mathcal{A}_0$ samples a message uniformly at random and outputs it. The function $\mathtt{aux}$, which receives m as an input, generates a new key pair $(\overline{sk}, \overline{pk}) \leftarrow \mathtt{KGen}(1^\lambda)$, computes $\overline{\sigma} \leftarrow \mathtt{Sign}(\overline{sk}, m)$, and outputs the hint $h := (\overline{\sigma}, \overline{pk})$. In the second stage, $\mathcal{A}_1$ receives as input the public key pk, the signature $\sigma \leftarrow \mathtt{Sign}(sk, m)$, and the hint $h = (\overline{\sigma}, \overline{pk})$, and it outputs $(\overline{\sigma}, \overline{pk})$.

By the completeness property, we have $\Pr[\mathtt{Vfy}(\overline{pk}, m, \overline{\sigma}) = 1] \geq 1 - \mathrm{negl}(\lambda)$. We further have $\Pr[\overline{pk} \neq pk] \geq 1 - \mathrm{negl}(\lambda)$ due to the high min-entropy of key generation.

It remains to argue that $\mathcal{A}$ satisfies the entropy condition (1). It holds that

$$\begin{aligned}\mathsf{HILL}_\infty\big(m \,\big|\, pk, \mathtt{aux}(m, pk)\big) &= \mathsf{HILL}_\infty\big(m \,\big|\, \mathtt{aux}(m, pk)\big) \\ &= \mathsf{HILL}_\infty\big(m \,\big|\, \overline{pk}, \mathtt{Sign}(\overline{sk}, m)\big) \geq \omega(\log \lambda)\,,\end{aligned}$$

where the first equality holds by the independence of pk and $(m, \mathtt{aux}(m, pk))$, the second equality holds by the construction of $\mathtt{aux}$, and the last inequality holds from the entropy requirement. Taking all of this together, we get that $\mathcal{A}$ is a valid adversary playing NR_{S} such that

$$\mathbf{Adv}^{\mathsf{NR}}_{\mathrm{S}}(\lambda, \mathcal{A}, \mathtt{aux}) \geq 1 - \mathrm{negl}(\lambda)\,.$$

This concludes the proof. □

[7] This leaves open only a very small, artificial gap for signature schemes that may potentially satisfy non-resignability: the message must be hard to compute from its signature while having low conditional HILL entropy.

Adversary $\mathcal{A}_0(pk)$	Adversary $\mathcal{A}_1(pk, \sigma, h)$	Function $\mathtt{aux}(m, pk)$
1 : $m \leftarrow \mathcal{M}$	1 : $(\overline{\sigma}, \overline{pk}) \leftarrow h$	1 : $(\overline{sk}, \overline{pk}) \leftarrow \mathtt{KGen}(1^\lambda)$
2 : **return** m	2 : **return** $(\overline{\sigma}, \overline{pk})$	2 : $\overline{\sigma} \leftarrow \mathrm{S}.\mathtt{Sign}(\overline{sk}, m)$
		3 : **return** $(\overline{\sigma}, \overline{pk})$

Fig. 4. Adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ and function $\mathtt{aux}$ used in the proof of Theorem 5.

Having established Theorem 5, it then follows as an immediate corollary that the BUFF-transform does not achieve non-resignability, no matter what hash function is used, as long as it is compressing, so that there is entropy in the message m when given $\mathrm{H}(m, pk)$ (here, the entropy is even statistical).

Corollary 6. *Let* S *be a signature scheme, and let* $\mathrm{BUFF}[\mathrm{S}, \mathcal{F}]$ *be the signature scheme obtained via the BUFF transform obtained by using a (keyed) hash function* $\mathcal{F}$ *that compresses by at least the size of the public key plus* $\omega(\log \lambda)$ *bits. Then there exists a PPT adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ *and a PPT function* $\mathtt{aux}$ *such that the entropy condition* (1) *is satisfied, yet*

$$\mathbf{Adv}^{\mathsf{NR}}_{\mathrm{S}}(\lambda, \mathcal{A}, \mathtt{aux}) \geq 1 - \mathrm{negl}(\lambda)\,.$$

Clearly, a non-compressing hash function avoids this particular attack; however, it is unclear if security would be restored (in particular in the light of Footnote 7). Also, from a practical point of view, hash functions used in the BUFF transform will be compressing.

3.2 Non-Resignability and the BUFF Transform in the ROM

When considering the random oracle model, things become somewhat subtle. First, we note that no definition of non-resignability *in the ROM* has been explicitly provided in the previous literature; the definitions given in [CDF+21] are in the plain model. When switching to the ROM, one needs to specify who is given access to the random oracle. Clearly, considering a signature scheme "in the ROM", we give $\mathtt{KGen}$, $\mathtt{Sign}$, and $\mathtt{Vfy}$ oracle access to the random oracle H.[8] Also, by default, the attacker is given oracle access to the random oracle. Thus, looking at Fig. 1, this means we certainly want to give $\mathcal{A}_0$ and $\mathcal{A}_1$ oracle access to the random oracle. But, say, what about the function $\mathtt{aux}$? Given that in the original definition in [CDF+21], the auxiliary information h is actually computed by $\mathcal{A}_0$ (and our re-writing in terms of a function $\mathtt{aux}$ is for later convenience), it is natural to then also allow the function $\mathtt{aux}$ to have oracle access the random oracle. We make this explicit in Fig. 5, where we give the resulting security game for non-resignability *in the ROM*.

However, there is another subtle matter in the definition of non-resignability when switching to the ROM. Namely, the entropy condition (1), as well as its unconditional counterpart $H_\infty(m \mid pk, \mathtt{aux}^{\mathrm{H}}(m, pk)) \geq \omega(\log \lambda)$, are not sufficient

[8] We make this explicit by writing $\mathtt{KGen}^{\mathrm{H}}$ etc.

anymore for the definition to be meaningful. Indeed, $\mathcal{A}_0^{\mathrm{H}}$ could simply choose m to be the hash of 0. In the ROM, this will be of high entropy and independent of pk; yet, $\mathcal{A}_1^{\mathrm{H}}$ can easily recover it (as the hash of 0), and then honestly sign it using his secret key. For this reason, in the definition of non-resignability in the ROM, we change the entropy requirement on the message to hold when additionally conditioning on the random oracle, i.e., we require

$$H_\infty\limits_{\substack{(sk,pk)\leftarrow \mathtt{KGen}^{\mathrm{H}}(1^\lambda)\\ m\leftarrow \mathcal{A}_0^{\mathrm{H}}(pk)}} \left(m \,\middle|\, pk, \mathtt{aux}^{\mathrm{H}}(m,pk), \mathrm{H}\right) \geq \omega(\log\lambda)\,. \tag{3}$$

We stress that we consider the *statistical* variant of the entropy condition here; with H an exponentially large function table, switching to the computational HILL variant will bring up further issues, which we want to avoid—for now (though we will look into this issue in Section 4.4). Furthermore, having this more stringent requirement on the attacker only makes our negative result stronger.

Game $\mathsf{NR}_{\mathrm{S}}^{\mathrm{H}}$

$(sk, pk) \leftarrow \mathtt{KGen}^{\mathrm{H}}(1^\lambda)$
$m \leftarrow \mathcal{A}_0^{\mathrm{H}}(pk)$
$h \coloneqq \mathtt{aux}^{\mathrm{H}}(m, pk)$
$\sigma \leftarrow \mathtt{Sign}^{\mathrm{H}}(sk, m)$
$(\overline{\sigma}, \overline{pk}) \leftarrow \mathcal{A}_1^{\mathrm{H}}(pk, \sigma, h)$
$v \coloneqq \mathtt{Vfy}^{\mathrm{H}}(\overline{pk}, m, \overline{\sigma})$
return $(v = 1 \wedge \overline{pk} \neq pk)$

Fig. 5. Security game $\mathsf{NR}_{\mathrm{S}}^{\mathrm{H}}$ for the signature $\mathrm{S}^{\mathrm{H}} = (\mathtt{KGen}^{\mathrm{H}}, \mathtt{Sign}^{\mathrm{H}}, \mathtt{Vfy}^{\mathrm{H}})$ in the random oracle model. In this variant, both the adversary and the function $\mathtt{aux}$ are granted access to the random oracle H.

The following theorem shows that, considering the above natural definition of non-resignability in the ROM, the impossibility result from the plain model (Theorem 5) carries over pretty much in the obvious way.

Theorem 7. *Let* H *be a random oracle and* $\mathrm{S}^{\mathrm{H}} = (\mathtt{KGen}^{\mathrm{H}}, \mathtt{Sign}^{\mathrm{H}}, \mathtt{Vrfy}^{\mathrm{H}})$ *be a signature scheme given query access to* H *such that for message* $\overline{m} \leftarrow \mathcal{M}$ *and key-pair* $(\overline{sk}, \overline{pk}) \leftarrow \mathtt{KGen}^{\mathrm{H}}(1^\lambda)$ *we have* $H_\infty(\overline{m} \mid \overline{pk}, \mathtt{Sign}^{\mathrm{H}}(\overline{sk}, m), \mathrm{H}) \geq \omega(\log\lambda)$. *Then there exists a PPT adversary* $\mathcal{A}^{\mathrm{H}} = (\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ *and a PPT function* $\mathtt{aux}^{\mathrm{H}}$ *with access to* H *such that the entropy condition* (3) *is satisfied, yet*

$$\mathbf{Adv}_{\mathrm{S}}^{\mathsf{NR}^{\mathrm{H}}}(\lambda, \mathcal{A}, \mathtt{aux}) \geq 1 - \mathrm{negl}(\lambda)\,.$$

The proof follows essentially from the proof of Theorem 5 with some obvious adjustments regarding the random oracle. The full proof is given in Appendix C.

Adversary $\mathcal{A}_0^{\mathrm{H}}(pk)$	Adversary $\mathcal{A}_1^{\mathrm{H}}(pk, \sigma, h)$	Function $\mathtt{aux}^{\mathrm{H}}(m, pk)$
1 : $m \leftarrow \mathcal{M}$	1 : $(\overline{\sigma}, \overline{pk}) \leftarrow h$	1 : $(\overline{sk}, \overline{pk}) \leftarrow \mathtt{KGen}^{\mathrm{H}}(1^\lambda)$
2 : **return** m	2 : **return** $(\overline{\sigma}, \overline{pk})$	2 : $\overline{\sigma} \leftarrow \mathrm{S}.\mathtt{Sign}^{\mathrm{H}}(\overline{sk}, m)$
		3 : **return** $(\overline{\sigma}, \overline{pk})$

Fig. 6. Adversary $\mathcal{A}^{\mathrm{H}} = (\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ and function $\mathtt{aux}$ used in the proof of Theorem 7.

We can leverage Theorem 7 to show that a BUFF-transformed signature scheme does not achieve non-resignability in the ROM. This is formalized in the following corollary.

Corollary 8. *Let* H *be a random oracle,* $\mathcal{F}^{\mathrm{H}}$ *be a PPT hash function given query access to* H*, compressing by at least the size of the public-key plus* $\omega(\log \lambda)$ *bits,* S^{H} *be a signature scheme given query access to* H*, and* $\mathrm{BUFF}[\mathrm{S}, \mathcal{F}]$ *be the signature scheme obtained via the BUFF transform with* $\mathcal{F}^{\mathrm{H}}$*. Then there exists a PPT adversary* $\mathcal{A}^{\mathrm{H}} = (\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ *and a PPT function* $\mathtt{aux}$ *such that the entropy condition* (3) *is satisfied, yet*

$$\mathbf{Adv}^{\mathsf{NR}^{\mathrm{H}}}_{\mathrm{BUFF}[\mathrm{S},\mathcal{F}]}(\lambda, \mathcal{A}, \mathtt{aux}) \geq 1 - \mathrm{negl}(\lambda)\,.$$

The above negative claim on the BUFF transform contradicts [CDF+21], which claims that the BUFF transform does satisfy non-resignability in the ROM (though without being explicit about the definition in the ROM). We discuss below the source of the false positive claim.

3.3 Φ-Non-Malleability in the ROM

[CDF+21] argues the non-resignability of the BUFF transform (in the ROM) in two steps. First, they prove the security of the BUFF transform *in the plain model* under the assumption that the hash function (family) satisfies the notion of Φ-non malleability (for a certain class Φ of functions). The formal statement is given in [CDF+21, Theorem 5.5] (cf. Theorem 4). Then, the following is remarked in [CDF+21, page 9], from which it is then concluded that the BUFF transform satisfies non-resignability in the ROM.

> *We note that if we model* H *as a random oracle then the hash function satisfies the definition of Φ-non-malleability for any class Φ where the functions ϕ preserve sufficient entropy in x, as will be the case for our results. The reason is that the adversary can only output a related random oracle value $\overline{y}$ if it has queried the random oracle about $\phi(x)$ before. But this is infeasible if $\phi(x)$ still contains enough entropy.*

Note that this claim originates from [BFS11], where a similar argument is made.

We show that this claim on the Φ-non-malleability of the random oracle is incorrect (under some mild assumption on Φ). As a matter of fact, the same

kind of attack as for the non-resignability of signature schemes applies here as well: we can simply let $\mathtt{aux}$ compute the mauled hash value. This bypasses the argument that the adversary has to make this particular query to the random oracle.

First, we explicitly spell out in Fig. 7 the security game of Φ-non-malleability *in the ROM*, for any PPT hash function $\mathcal{F}^{\mathrm{H}}$ with query access to the random oracle H. Then, we state the negative result in Theorem 9 below. Note that the latter in particular implies that the random oracle itself, i.e., when setting $\mathcal{F}^{\mathrm{H}} = H$, is not Φ-non-malleable.

Game $\Phi\text{-}\mathsf{NM}^{\mathrm{H}}_{\mathcal{F}}$

$x \leftarrow \mathcal{A}_0^{\mathrm{H}}()$

$h \coloneqq \mathtt{aux}^{\mathrm{H}}(x)$

$y \coloneqq \mathcal{F}^{\mathrm{H}}(x)$

$(\overline{y}, \phi) \leftarrow \mathcal{A}_1^{\mathrm{H}}(y, h)$

return $(\mathcal{F}^{\mathrm{H}}(\phi(x)) = \overline{y} \wedge \phi(x) \neq x)$

Fig. 7. Security game $\Phi\text{-}\mathsf{NM}^{\mathrm{H}}_{\mathcal{F}}$ for the hash function $\mathcal{F}^{\mathrm{H}}$ in the ROM and an arbitrary function family Φ.

Theorem 9. *Let* H *be a random oracle,* $\mathcal{F}^{\mathrm{H}} : \mathcal{X} \to \mathcal{Y}$ *be a PPT hash function given query access to* H*, compressing by least* $\omega(\log \lambda)$ *bits, and* $\Phi \subseteq \mathcal{X}^{\mathcal{X}}$ *be such that there is a PPT algorithm* $\mathcal{D}_\Phi$ *producing* $\phi \in \Phi$ *that does not fix most points with overwhelming probability, i.e.*

$$\Pr_{\substack{\phi \leftarrow \mathcal{D}_\Phi \\ x \leftarrow \mathcal{X}}} [\phi(x) = x] \leq \mathrm{negl}(\lambda) \,. \tag{4}$$

Then, there exists a PPT adversary $\mathcal{A}^{\mathrm{H}} = (\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ *and a PPT function* $\mathtt{aux}^{\mathrm{H}}$*, both given query access to* H*, such that the entropy condition*

$$H_\infty\limits_{x \leftarrow \mathcal{A}_0^{\mathrm{H}}()} \left(x \,\middle|\, \mathtt{aux}^{\mathrm{H}}(x), \mathrm{H}\right) \geq \omega(\log \lambda)$$

is satisfied, yet

$$\mathbf{Adv}^{\Phi\text{-}\mathsf{NM}^{\mathrm{H}}}_{\mathcal{F}}(\lambda, \mathcal{A}, \mathtt{aux}) \geq 1 - \mathrm{negl}(\lambda) \,.$$

Proof. We give a PPT adversary $\mathcal{A}^{\mathrm{H}} = (\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ that wins the game $\Phi\text{-}\mathsf{NM}^{\mathrm{H}}$ with overwhelming probability. Both $\mathcal{A}^{\mathrm{H}} = (\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ and $\mathtt{aux}^{\mathrm{H}}$ are given in Fig. 8. In the first stage, adversary $\mathcal{A}_0$ chooses x uniformly at random. The auxiliary function on input $x \in \mathcal{X}$, chooses a function $\phi \leftarrow \mathcal{D}_\Phi$, computes $\overline{y} \coloneqq \mathcal{F}^{\mathrm{H}}(\phi(x))$, and returns the pair $(\overline{y}, \phi)$ as hint. In the second stage, adversary

$\mathcal{A}_1$ gets $y = \mathcal{F}^{\mathrm{H}}(x)$ along with the hint $h = (\overline{y}, \phi)$ as input and outputs the $(\overline{y}, \phi)$ from the hint. It is easy to see that $1 \leftarrow \Phi\text{-}\mathsf{NM}^{\mathrm{H}}$, unless $\phi(x) = x$, which only happens with negligible probability due to (4). Furthermore, the entropy requirement follows from the fact that x is chosen uniformly from $\mathcal{X}$ independent of (ϕ, H), and that $\mathcal{F}^{\mathrm{H}}$ is compressing by $\omega(\log \lambda)$ bits. □

Adversary $\mathcal{A}_0^{\mathrm{H}}(pk)$	Adversary $\mathcal{A}_1^{\mathrm{H}}(y, h)$	Function $\mathtt{aux}^{\mathrm{H}}(x)$
1 : $x \leftarrow \mathcal{X}$	1 : $(\overline{y}, \phi) \coloneqq h$	1 : $\phi \leftarrow \mathcal{D}_{\Phi}$
2 : **return** x	2 : **return** $(\overline{y}, \phi)$	2 : $\overline{y} \coloneqq \mathcal{F}^{\mathrm{H}}(\phi(x))$
		3 : **return** $(\overline{y}, \phi)$

Fig. 8. Adversary $\mathcal{A}^{\mathrm{H}} = (\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ and function $\mathtt{aux}^{\mathrm{H}}$ used in the proof of Theorem 9.

Remark 10. The above theorem is stated for the random oracle model. It is easy to see, though, that the result carries over to the plain model for concrete hash functions.

4 Weak Non-Resignability and Salted BUFF

In this section, we partly recover from the negative results from the previous section by considering a salted version of the BUFF transform, and showing that it satisfies a weaker variant of non-resignability in the ROM. The formal specification of the salted BUFF transform, denoted \$-BUFF, is given in Fig. 9. It matches with the original BUFF transform, except that some random salt s is added to the signature and used for the hash.

$\mathtt{KGen}(1^\lambda)$	$\mathtt{Sign}(sk, m)$	$\mathtt{Vfy}(pk, m, (\sigma, y, s))$
1 : $(sk, pk) \leftarrow \mathrm{S}.\mathtt{KGen}(1^\lambda)$	1 : $s \leftarrow \{0,1\}^\ell$	1 : $\overline{y} \leftarrow \mathcal{F}(pk, m, s)$
2 : **return** (sk, pk)	2 : $y \coloneqq \mathcal{F}(pk, m, s)$	2 : $d \coloneqq \mathrm{S}.\mathtt{Vfy}(pk, \overline{y}, \sigma)$
	3 : $\sigma \leftarrow \mathrm{S}.\mathtt{Sign}(sk, y)$	3 : **return** $(d = 1 \wedge y = \overline{y})$
	4 : **return** (σ, y, s)	

Fig. 9. The salted BUFF transform \$-BUFF$[\mathrm{S}, \mathcal{F}]$ for a signature scheme S and a hash function $\mathcal{F}$.

Our goal is to show that the salted BUFF transform satisfies the weaker variant of non-resignability in the ROM obtained by replacing the non-resignability game $\mathsf{NR}_{\mathrm{S}}^{\mathrm{H}}$ to $\mathsf{NR}_{\mathrm{S}}^{\mathrm{H},\perp}$, as given in Fig. 10 (left). The only difference is that the

function $\mathtt{aux}$, which computes the auxiliary information h, is not given access to the random oracle anymore.

Indeed, below we will prove the following. To start with, we consider the entropy condition on the message to be statistical; as explained in Sect. 3.2, here in the ROM we additionally need to condition on H (in order to avoid letting $m = \mathrm{H}(0)$). In Sect. 4.4, we then discuss the case of computational entropy.

We consider both the case of *classical* and of *quantum* queries by the adversary $\mathcal{A}$ when querying the random oracle, as well as a "semi-quantum" case where $\mathcal{A}_0$ is classical yet $\mathcal{A}_1$ may be quantum. The latter is motivated by the fact that $\mathcal{A}_0$ is typically not adversarially chosen, but determined by the considered application.[9]

Theorem 11. *Let* H *be a random oracle with co-domain denoted by* $\mathcal{Y}$*, let* $\mathrm{S}^{\mathrm{H}} = (\mathtt{KGen}, \mathtt{Sign}^{\mathrm{H}}, \mathtt{Vfy}^{\mathrm{H}})$ *be a signature scheme with a key generation that has no access to* H*, and let* $\$\text{-BUFF}[\mathrm{S}, \mathrm{H}]$ *be the signature scheme obtained by applying the salted BUFF transform (cf. Fig. 9) to* S*. Furthermore, let* $\mathcal{A}^{\mathrm{H}} = (\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ *be an computationally unbounded* $\mathsf{NR}^{\mathrm{H},\perp}$ *adversary,* $\mathtt{aux}$ *be any function, and let* $\mathtt{Sign}^{\mathrm{H}}, \mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}}$ *make at most* q_S, q_0, q_1 *queries to* H *respectively. Assuming*

$$H_{\infty}\limits_{\substack{(sk,pk)\leftarrow\mathtt{KGen}(1^{\lambda})\\ m\leftarrow\mathcal{A}_0^{\mathrm{H}}(pk)}}\left(m \,\middle|\, \mathrm{H}, pk, \mathtt{aux}(m, pk)\right) \geq \log(1/\epsilon)\,,$$

then for $\mathtt{Sign}^{\mathrm{H}}, \mathcal{A}_0, \mathcal{A}_1$ *making quantum queries in general, it holds that*

$$\mathbf{Adv}^{\mathsf{NR}^{\mathrm{H},\perp}}_{\$\text{-BUFF}[\mathrm{S},\mathrm{H}]}(\mathcal{A}, \mathtt{aux}) \leq \sqrt{q_0 \cdot 2^{-\ell}} + \frac{q_0 \cdot 2^{-\ell}}{2} + 4(q_1 + q_S)\sqrt{\epsilon} + \frac{(2q_0+1)^2}{|\mathcal{Y}|}\,, \tag{5}$$

and if $\mathcal{A}_0^{\mathrm{H}}$ *is restricted to classical queries,*

$$\mathbf{Adv}^{\mathsf{NR}^{\mathrm{H},\perp}}_{\$\text{-BUFF}[\mathrm{S},\mathrm{H}]}(\mathcal{A}, \mathtt{aux}) \leq q_0 \cdot 2^{-\ell} + 4(q_1 + q_S)\sqrt{\epsilon} + \frac{(q_0+1)}{|\mathcal{Y}|}\,. \tag{6}$$

In case where $\mathtt{Sign}^{\mathrm{H}}, \mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}}$ *are all restricted to classical queries, then we have*

$$\mathbf{Adv}^{\mathsf{NR}^{\mathrm{H},\perp}}_{\$\text{-BUFF}[\mathrm{S},\mathrm{H}]}(\mathcal{A}, \mathtt{aux}) \leq q_0 \cdot 2^{-\ell} + 2(q_1 + q_S) \cdot \epsilon + \frac{(q_0+1)}{|\mathcal{Y}|}\,. \tag{7}$$

Remark 12. In the setting where the key generation $\mathtt{KGen}^{\mathrm{H}}$ is given access to the random oracle H, it is not too hard to extend the non-resignability of $\$\text{-BUFF}$ into such a setting, by noticing that any sufficiently long portion of the random salt s in $\$\text{-BUFF}$ is hard to guess by $\mathtt{KGen}^{\mathrm{H}}$, and hence separating the domain queried by $\mathtt{KGen}^{\mathrm{H}}$ from ones queried by $\mathtt{Sign}^{\mathrm{H}}$ and $\mathtt{Vfy}^{\mathrm{H}}$ up to some negligible advantage.

[9] In the fully quantum case, we even allow the signing procedure to make quantum queries to H; this is not really relevant but obtained for free.

Our proof goes along the following blueprint. First, we consider a *weaker* and *salted* variant of the Φ-non-malleability property in the ROM (for a particular function family Φ), as specified in Fig. 10 (right), and show that if the random oracle satisfies this variant (in the ROM) then the salted BUFF transform satisfies $\mathsf{NR}^{\mathrm{H},\perp}$ (in the ROM). For this step (Sect. 4.1), we can simply recycle the original corresponding proof. The bulk of the work, and thus our main technical contribution, is then proving that the random oracle indeed satisfies this weaker and salted variant of Φ-non-malleability. This is done in Sect. 4.2 for the classical ROM, and in Sect. 4.3 for the QROM, i.e., when allowing quantum queries to the random oracle.

Finally, in Sect. 4.4, we show how the relax the requirement on the statistical min-entropy $H_\infty(m|\mathrm{H}, pk, \mathtt{aux}(m, pk))$ to a variant of the computation HILL entropy, adjusted to the ROM setting in order to avoid the conditioning on H.

Game $\mathsf{NR}_{\mathrm{S}}^{\mathrm{H},\perp}$	Game Φ-\$-$\mathsf{NM}_{\mathcal{F}}^{\mathrm{H},\perp}(pk)$
1 : $(sk, pk) \leftarrow \mathtt{KGen}^{\mathrm{H}}(1^\lambda)$	1 : $m \leftarrow \mathcal{A}_0^{\mathrm{H}}$
2 : $m \leftarrow \mathcal{A}_0^{\mathrm{H}}(pk)$	2 : $h \coloneqq \mathtt{aux}(m)$
3 : $h \coloneqq \mathtt{aux}(m)$	3 : $s \leftarrow \{0,1\}^\ell$
4 : $\sigma \leftarrow \mathtt{Sign}^{\mathrm{H}}(sk, m)$	4 : $y \coloneqq \mathcal{F}^{\mathrm{H}}(pk, m, s)$
5 : $(\overline{\sigma}, \overline{pk}) \leftarrow \mathcal{A}_1^{\mathrm{H}}(pk, \sigma, h)$	5 : $(\overline{y}, \overline{pk}, \overline{s}) \leftarrow \mathcal{A}_1^{\mathrm{H}}(y, h, s)$
6 : $v \coloneqq \mathtt{Vfy}^{\mathrm{H}}(\overline{pk}, m, \overline{\sigma})$	6 : **return** $(\mathcal{F}^{\mathrm{H}}(\overline{pk}, m, \overline{s}) = \overline{y} \wedge \overline{pk} \neq pk)$
7 : **return** $(v = 1 \wedge \overline{pk} \neq pk)$	

Fig. 10. Weaker variant of the non-resignability game for a signature scheme $\mathrm{S}^{\mathrm{H}} = (\mathtt{KGen}^{\mathrm{H}}, \mathtt{Sign}^{\mathrm{H}}, \mathtt{Vrfy}^{\mathrm{H}})$, where $\mathtt{aux}$ has no access to H (left), and the salted Φ-non-malleability game (right) for a hash function $\mathcal{F}^{\mathrm{H}}$. The game is slightly changed towards our use-case of the BUFF-transform: Instead of letting $\mathcal{A}_0$ output one value x and requiring $\mathcal{A}_1$ to output the hash value of $\phi(x)$, we split x into (pk, m), where pk is arbitrary but fixed and (only) m is chosen by $\mathcal{A}_0$. To win the game, $\mathcal{A}_1$ has to find a hash for a different pk but the same m (very much in line with finding a signature for the same message under a different public key).

4.1 Φ-\$-$\mathsf{NM}^{\mathrm{H},\perp} \Rightarrow \mathsf{NR}^{\mathrm{H},\perp}$ for \$-BUFF

The following shows that the modified BUFF transformation provides the variant of non-resignability considered in Fig. 10 (left), given that the hash function H achieves the Φ-non-malleability variant considered in Fig. 10 (right).

Proposition 13. *Let* H *be a random oracle, let* S *be a signature scheme with a key generation that has no access to* H*, and let* \$-BUFF[S, H] *be the signature scheme obtained by applying the modified BUFF transform (cf. Fig. 9) to* S*.*

Then for any adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ *and function* $\mathtt{aux}$ *as in the game* $\mathsf{NR}^{\mathrm{H},\perp}$*, there exists a keyed adversary* $\mathcal{B}(sk, pk) = (\mathcal{B}_0(sk, pk), \mathcal{B}_1(sk, pk, \bullet))$ *such that*

$$\underset{\substack{(sk,pk)\\ m\leftarrow\mathcal{A}_0^{\mathrm{H}}(pk)}}{\mathsf{guess}} \left(m \,\middle|\, H, pk, \mathtt{aux}(m, pk)\right) = \underset{(sk,pk)}{\mathbb{E}} \left[\underset{m\leftarrow\mathcal{B}_0^{\mathrm{H}}(sk,pk)}{\mathsf{guess}} \left(m \,\middle|\, H, \mathtt{aux}(m, pk)\right) \right]$$

and

$$\mathbf{Adv}^{\mathsf{NR}^{\mathrm{H},\perp}}_{\$\text{-}\mathrm{BUFF}[\mathrm{S},\mathrm{H}]}(\mathcal{A}, \mathtt{aux}) \leq \underset{(sk,pk)}{\mathbb{E}} \left[\mathbf{Adv}^{\Phi\text{-}\$\text{-}\mathsf{NM}^{\mathrm{H},\perp}(pk)}_{\mathrm{H}}(\mathcal{B}(sk, pk), \mathtt{aux}(\bullet, pk)) \right] .$$

$\mathcal{B}(sk, pk)$ *is given in Fig. 11. In particular,* $\mathcal{B}_0$ *makes the same number and the same type (classical versus quantum) of queries as* $\mathcal{A}_0$*, and* $\mathcal{B}_1$ *makes the same number and the same type of queries as the composition of* $\mathrm{S}.\mathtt{Sign}$ *and* $\mathcal{A}_1$*.*

Proof. The claim on the guessing probability follows directly from the construction of $\mathcal{B}$ and the basic properties of the conditional guessing probability. Towards the claim on the advantage, note that for any pair (sk, pk), the adversary $\mathcal{B}(sk, pk)$ wins the game $\Phi\text{-}\$\text{-}\mathsf{NM}^{\mathrm{H},\perp}(pk)$ if and only if

$$\mathrm{H}(\overline{pk}, m, \overline{s}) = \overline{y} \wedge \overline{pk} \neq pk \,,$$

which, by Fig. 9, is necessary for

$$\$\text{-}\mathrm{BUFF}[\mathrm{S},\mathrm{H}].\mathtt{Vfy}^{\mathrm{H}}(\overline{pk}, m, (\overline{\sigma}, \overline{y}, \overline{s})) \wedge \overline{pk} \neq pk$$

to be satisfied, where $\overline{\sigma}$ is as specified in Fig. 11. The expectation of the latter being satisfied, taken over $(sk, pk) \leftarrow \mathtt{KGen}(1^\lambda)$, is precisely $\mathbf{Adv}^{\mathsf{NR}^{\mathrm{H},\perp}}_{\$\text{-}\mathrm{BUFF}[\mathrm{S},\mathrm{H}]}(\mathcal{A}, \mathtt{aux})$, and so this proves the claim in the advantage. □

$\underline{\mathcal{B}_0^{\mathrm{H}}(sk, pk)}$

1 : $m \leftarrow \mathcal{A}_0^{\mathrm{H}}(pk)$
2 : **return** m

$\underline{\mathcal{B}_1^{\mathrm{H}}(sk, pk, y, h, s)}$

1 : $\sigma \leftarrow \mathrm{S}.\mathtt{Sign}^{\mathrm{H}}(sk, y)$
2 : $((\overline{\sigma}, \overline{y}, \overline{s}), \overline{pk}) \leftarrow \mathcal{A}_1^{\mathrm{H}}(pk, (\sigma, y, s), h)$
3 : **return** $(\overline{y}, \overline{pk}, \overline{s})$

Fig. 11. Adversary $\mathcal{B}$ used in the proof of Theorem 13.

4.2 The Random Oracle is Φ-\$-$\mathsf{NM}^{\mathrm{H},\perp}$ in the ROM

Here, we show that the random oracle (i.e., the hash function $\mathcal{F}^{\mathrm{H}} := \mathrm{H}$) satisfies the notion of $\Phi\text{-}\$\text{-}\mathsf{NM}^{\mathrm{H},\perp}$, specified in Fig. 10 (right).

Theorem 14. *Let $(\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ be an (computationally unbounded) adversary against Φ-\$-$\mathsf{NM}^{\mathrm{H},\perp}$, with $\mathcal{A}_0^{\mathrm{H}}$ making at most q_0 classical queries and $\mathcal{A}_1^{\mathrm{H}}$ at most q_1 classical queries to the random oracle H with co-domain $\mathcal{Y}$, and let aux be a function. Set*

$$\epsilon := \underset{m \leftarrow \mathcal{A}_0^H}{\mathsf{guess}} \left(m \,\middle|\, \mathrm{H}, \mathsf{aux}(m) \right),$$

and let pk be an arbitrary valid public key. Then,

$$\mathbf{Adv}_{\mathrm{H}}^{\Phi\text{-}\$\text{-}\mathsf{NM}^{\mathrm{H},\perp}(pk)}(\mathcal{A}, \mathsf{aux}) \leq q_0 \cdot 2^{-\ell} + 2q_1 \cdot \epsilon + (q_0 + 1)/|\mathcal{Y}| \,.$$

Proof. The proof proceeds via the games $\mathsf{G}_0, \ldots, \mathsf{G}_6^i$ displayed in Fig. 12. Steps from G_0 to G_3 are symmetric, in that they argue closeness between games, while each of the rest upperbounds the winning probability of one game via another.

The closeness between games $\mathsf{G}_0 \approx \mathsf{G}_1 \approx \mathsf{G}_2 \approx \mathsf{G}_3$ is via arguing that an adversary cannot detect some reprogramming in the random oracle except with small probability. For $\mathsf{G}_0 \approx \mathsf{G}_1$, one exploits that the reprogrammed point involves a freshly chosen salt s in uniform distribution. For $\mathsf{G}_1 \approx \mathsf{G}_2 \approx \mathsf{G}_3$, one exploits that m has high entropy, conditioned on the view of the attacker throughout the execution of the intermediate game G_2.

G_0 *to* G_1 *hop.* The only difference between G_0 and G_1 is that the former computes $y \leftarrow \mathrm{H}(pk, m, s)$, while the latter does a reprogramming via $\mathrm{H}(pk, m, s) \coloneqq y \leftarrow \mathcal{Y}$. Thus, both games behave identically unless $\mathcal{A}_0$ has queried the corresponding input (pk, m, s), which happens with probability at most $q_0 \cdot 2^{-\ell}$ in either game, due to the random choice of s.

G_1 *to* G_2 *hop.* Without loss of generality, we may assume $\overline{pk} \neq pk$, in which case the reprogramming of H, done in G_1 but not in G_2, does not affect the final hash $\mathrm{H}(\overline{pk}, m, \overline{s})$. Thus, there is a difference in the two games only if $\mathcal{A}_1$ makes a query to (pk, m, s); however, in G_2, using that y and s are independent of (m, H, h) and by the entropy condition, we have that $\mathsf{guess}(m \mid H, y, h, s) = \mathsf{guess}(m \mid H, h) \leq \epsilon$, and so this happens with probability at most $q_1\epsilon$. Thus,

$$\Pr\left[1 \leftarrow \mathsf{G}_1\right] \leq \Pr\left[1 \leftarrow \mathsf{G}_2\right] + q_1\epsilon \,.$$

G_2 *to* G_3 *hop.* Similar as above, the difference between G_2 and G_3 can only be noticed when $\mathcal{A}_1$ makes a query of the form $(\cdot, m, \cdot)$, which again happens with probability at most $\mathsf{guess}(m \mid H, y, h, s) \leq \epsilon$. Therefore,

$$\Pr\left[1 \leftarrow \mathsf{G}_2\right] \leq \Pr\left[1 \leftarrow \mathsf{G}_3\right] + q_1\epsilon \,.$$

G_3 *to* G_4 *hop.* In G_4, we relax the winning condition by dropping the requirement $\overline{pk} \neq pk$; this only increases the winning probability. Furthermore, we replace $\mathcal{A}_1$ in G_3 by $\mathcal{B}_1$ in G_4, which computes $h \coloneqq \mathsf{aux}(m)$ and samples $s \leftarrow \{0,1\}^\ell$ and $y \leftarrow \mathcal{Y}$ as a first step, and then runs $\mathcal{A}_1$ on input (y, h, s); this change is only syntactically and does not affect the winning probability. Thus,

$$\Pr\left[1 \leftarrow \mathsf{G}_3\right] \leq \Pr\left[1 \leftarrow \mathsf{G}_4\right] \,.$$

Game G_0

$m \leftarrow \mathcal{A}_0^{\mathrm{H}}()$
$h \coloneqq \mathtt{aux}(m)$
$s \leftarrow \{0,1\}^{\ell}$
$y \coloneqq \mathrm{H}(pk, m, s)$
$(\overline{y}, \overline{pk}, \overline{s}) \leftarrow \mathcal{A}_1^{\mathrm{H}}(y, h, s)$
return $(\mathrm{H}(\overline{pk}, m, \overline{s}) = \overline{y}$
$\qquad \wedge\, \overline{pk} \neq pk)$

Game G_1

$m \leftarrow \mathcal{A}_0^{\mathrm{H}}()$
$h \coloneqq \mathtt{aux}(m)$
$s \leftarrow \{0,1\}^{\ell}$
$\mathrm{H}(pk, m, s) \coloneqq y \leftarrow \mathcal{Y}$
$(\overline{y}, \overline{pk}, \overline{s}) \leftarrow \mathcal{A}_1^{\mathrm{H}}(y, h, s)$
return $(\mathrm{H}(\overline{pk}, m, \overline{s}) = \overline{y}$
$\qquad \wedge\, \overline{pk} \neq pk)$

Game G_2

$m \leftarrow \mathcal{A}_0^{\mathrm{H}}()$
$h \coloneqq \mathtt{aux}(m)$
$s \leftarrow \{0,1\}^{\ell}$
$y \leftarrow \mathcal{Y}$
$(\overline{y}, \overline{pk}, \overline{s}) \leftarrow \mathcal{A}_1^{\mathrm{H}}(y, h, s)$
return $(\mathrm{H}(\overline{pk}, m, \overline{s}) = \overline{y}$
$\qquad \wedge\, \overline{pk} \neq pk)$

Game G_3

$m \leftarrow \mathcal{A}_0^{\mathrm{H}}()$
$h \coloneqq \mathtt{aux}(m)$
$s \leftarrow \{0,1\}^{\ell}$
$y \leftarrow \mathcal{Y}$
$(\overline{y}, \overline{pk}, \overline{s}) \leftarrow \mathcal{A}_1^{\mathrm{H}[(\cdot, m, \cdot) \mapsto \perp]}(y, h, s)$
return $(\mathrm{H}(\overline{pk}, m, \overline{s}) = \overline{y}$
$\qquad \wedge\, \overline{pk} \neq pk)$

Game G_4

$m \leftarrow \mathcal{A}_0^{\mathrm{H}}()$
$(\overline{y}, \overline{pk}, \overline{s}) \leftarrow \mathcal{B}_1^{\mathrm{H}[(\cdot, m, \cdot) \mapsto \perp]}(m)$
return $\mathrm{H}(\overline{pk}, m, \overline{s}) = \overline{y}$

Game G_5^i

$m \leftarrow \mathcal{A}_0^{\mathrm{H}}()$
$(\overline{y}, \overline{pk}, \overline{s}) \leftarrow \mathcal{B}_1^{\mathrm{H}[(\cdot, m, \cdot) \mapsto \perp]}(m)$
return $\mathrm{H}(\overline{pk}, m, \overline{s}) = \overline{y}$
$\qquad \wedge (\overline{pk}, m, \overline{s}) = (pk^i, m^i, s^i)$
// where (pk^i, m^i, s^i) is $\mathcal{A}_0$'s i-th query

Game G_6^i

$m \leftarrow \mathcal{A}_0^{\mathrm{H}}()$
$(\overline{y}, \overline{pk}, \overline{s}) \leftarrow \mathcal{B}_1^{\mathrm{H}[(\cdot, m^i, \cdot) \mapsto \perp]}(m^i)$
return $\mathrm{H}(pk^i, m^i, s^i) = \overline{y}$
// where (pk^i, m^i, s^i) is $\mathcal{A}_0$'s i-th query

Fig. 12. The sequence of games considered in the proof of Theorem 14. In all games, H is understood to be uniformly random. In the game G_4 etc., $\mathrm{H}[(\cdot, m, \cdot) \mapsto \perp]$ denotes the oracle that blocks queries of the form $(\cdot, m, \cdot)$, i.e., replies with some special value $\perp$ in that case, and replies with H applied to the query otherwise.

G_4 *to* G_5^i *hop.* Since $\mathcal{A}_0$ is classical, assume without loss of generality that it never repeats a query. If $(\overline{pk}, m, \overline{s})$ has never been queried by $\mathcal{A}_0$, i.e., $(\overline{pk}, m, \overline{s}) \neq (pk^j, m^j, s^j)$ for all $j \in \{1, \ldots, q_0\}$, then (using that $\mathcal{A}_1$ is blocked from queries of the form $(\cdot, m, \cdot)$) the hash $\mathrm{H}(\overline{pk}, m, \overline{s})$ is random and independent of y, in which case they are equal with probability $1/|\mathcal{Y}|$. Therefore we obtain,

$$\Pr\left[\mathsf{G}_4\right] = 1/|\mathcal{Y}| + \sum_{i\in[q_0]} \Pr\left[\begin{matrix} 1 \leftarrow \mathsf{G}_4 \\ (\overline{pk}, m, \overline{s}) = (pk^i, m^i, s^i) \end{matrix}\right] = 1/|\mathcal{Y}| + \sum_{i\in[q_0]} \Pr\left[1 \leftarrow \mathsf{G}_5^i\right] \ .$$

G_5^i *to* G_6^i *hop.* In G_5^i, due to the extra condition $m = m^i$ for winning the game, replacing m by m^i as in G_6^i has no effect on the winning probability, and dropping the requirement again then only increases the probability. Thus

$$\Pr\left[1 \leftarrow \mathsf{G}_5^i\right] \leq \Pr\left[1 \leftarrow \mathsf{G}_6^i\right] \ .$$

It remains to show that the latter probability is small. First, we may assume that $\mathcal{A}_0$'s queries (pk^j, m^j, s^j) are all distinct. Furthermore, we may assume that once $\mathcal{A}_0$ has decided on the i-th query (pk^i, m^i, s^i), it stops without making this query; the game then simply proceeds as described with running $\mathcal{B}_1$ on input m^i. This shows that $\mathcal{B}_1$'s input is independent of $\mathrm{H}(pk^i, m^i, s^i)$, and so is his output $\overline{y}$ then, given that he is blocked from queries of the form $(\cdot, m, \cdot)$. Hence

$$\Pr\left[1 \leftarrow \mathsf{G}_6^i\right] \leq 1/|\mathcal{Y}| \ .$$

Combining all the (in)equalities then concludes the proof. □

Combining Proposition 13, Theorem 14, for $\mathcal{A}, \mathtt{Sign}$ restricted to classical queries, we get

$$\begin{aligned} \mathbf{Adv}^{\mathsf{NR}^{\mathrm{H},\perp}}_{\$\text{-}\mathrm{BUFF}[\mathrm{S},\mathrm{H}]}(\mathcal{A}, \mathtt{aux}) &\leq \mathop{\mathbb{E}}_{(sk,pk)} \left[q_0 \cdot 2^{-\ell} + 2(q_1 + q_S) \cdot \epsilon_{sk,pk} + (q_0 + 1)/|\mathcal{Y}|\right] \\ &\leq q_0 \cdot 2^{-\ell} + 2(q_1 + q_S) \cdot \epsilon + (q_0 + 1)/|\mathcal{Y}| \ , \end{aligned}$$

where

$$\epsilon_{sk,pk} \coloneqq \mathop{\mathsf{guess}}_{m \leftarrow \mathcal{A}_0^{\mathrm{H}}(pk)} \left(m \,\middle|\, \mathrm{H}, pk, \mathtt{aux}(m, pk)\right) \ ,$$

and the second inequality is via linearity. This concludes (7).

4.3 The Random Oracle is Φ-\$-$\mathsf{NM}^{\mathrm{H},\perp}$ in the QROM

We now prove the same result in the QROM when $\mathcal{A}_0^{\mathrm{H}}$ and $\mathcal{A}_1^{\mathrm{H}}$ are algorithms that may make quantum (i.e., superposition) queries to the random oracle.

Theorem 15. *Let* $\mathcal{A} = (\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ *be a (computationally unbounded) adversary against* Φ-\$-$\mathsf{NM}^{\mathrm{H},\perp}$*, with* $\mathcal{A}_0^{\mathrm{H}}$ *making at most* q_0 quantum *queries and* $\mathcal{A}_1^{\mathrm{H}}$ *at*

most q_1 quantum *queries to the random oracle* H *with co-domain* $\mathcal{Y}$*, and let* aux *be a function. Set*

$$\epsilon := \underset{m \leftarrow \mathcal{A}_0^H}{\mathsf{guess}} \left(m \,\middle|\, H, \mathsf{aux}(m) \right),$$

and let pk *be an arbitrary public key. Then,*

$$\mathbf{Adv}_{\mathrm{H}}^{\Phi\text{-}\$\text{-}\mathsf{NM}^{\mathrm{H},\perp}(pk)}(\mathcal{A}, \mathsf{aux}) \leq \sqrt{q_0 \cdot 2^{-\ell}} + \frac{q_0 \cdot 2^{-\ell}}{2} + 4q_1\sqrt{\epsilon} + (2q_0+1)^2/|\mathcal{Y}|\,.$$

Proof. The proof of Theorem 15 is identical to that of Theorem 14 up to some small changes in the argumentation for the first four game hops, and a more significant change of strategy in the last two. Indeed, we reuse the first four games and define modified versions of G_5^i and G_6^i, as specified in Fig. 13. Namely, in case of superposition queries by $\mathcal{A}_0$, we cannot define (pk^i, m^i, s^i) as the i-th query; instead, rather naturally, we introduce (pk^i, m^i, s^i) by *measuring* the i-th query. Furthermore, for technical reasons, we then reprogram H on (pk^i, m^i, s^i) by a random value Θ, from this or the next query onwards.

Game G_5^i	Game G_6^i
$m \leftarrow \mathcal{A}_0^{\mathrm{H}_i^\Theta}()$	$m \leftarrow \mathcal{A}_0^{\mathrm{H}_i^\Theta}()$
#measure ith query: (pk^i, m^i, s^i)	#measure ith query: (pk^i, m^i, s^i)
$(\overline{y}, \overline{pk}, \overline{s}) \leftarrow \mathcal{B}_1^{\mathrm{H}[(\cdot,m,\cdot)\mapsto\perp]}(m)$	$(\overline{y}, \overline{pk}, \overline{s}) \leftarrow \mathcal{B}_1^{\mathrm{H}[(\cdot,m^i,\cdot)\mapsto\perp]}(m^i)$
return $\mathrm{H}_i^\Theta(\overline{pk}, m, \overline{s}) = \overline{y}$	**return** $\mathrm{H}_i^\Theta(pk^i, m^i, s^i) = \overline{y}$
$\wedge(\overline{pk}, m, \overline{s}) = (pk^i, m^i, s^i)$	

Fig. 13. The modified games G_5^i and G_6^i for the proof of Theorem 15. In both games, H is understood to be uniformly random. H_i^Θ is the oracle that implements H until just before the i-th query, then measures that query in the computational basis to obtain (pk^i, m^i, s^i), and subsequently answers queries from $\mathcal{A}_0$ with $\mathrm{H}[(pk^i, m^i, s^i) \mapsto \Theta]$, i.e., with H but reprogrammed to a random value Θ at (pk^i, m^i, s^i), either from the i-th or the $i+1$-th query onward, with this choice being made uniformly at random as well.

G_0 *to* G_1 *hop.* The only difference between G_0 and G_1 is that the former computes $y \leftarrow \mathrm{H}(pk, m, s)$, while the latter reprograms $\mathrm{H}(pk, m, s) := y \leftarrow \mathcal{Y}$. With s being uniformly random chosen, this is a direct application of the *adaptive reprogramming lemma* (Theorem 1 in [GHHM21]) to bound the distinguishing probability:

$$\Pr\left[1 \leftarrow \mathsf{G}_0\right] \leq \Pr\left[1 \leftarrow \mathsf{G}_1\right] + \sqrt{q_0 \cdot 2^{-\ell}} + \frac{q_0 \cdot 2^{-\ell}}{2}\,.$$

G_1 *to* G_2 *hop.* Without loss of generality, we may assume $\overline{pk} \neq pk$, in which case the reprogramming of H, done in G_1 but not in G_2, does not affect the final

hash $\mathrm{H}(\overline{pk}, m, \overline{s})$. Thus, the only difference between the two games is that $\mathcal{A}_1$ interacts with the original H in G_2, and with H that is reprogrammed to $\perp$ at the point (pk, m, s) in G_1.

This is a direct application for O2H ([AHU19, Theorem 3]). We note that in game G_2, $\mathcal{A}_1$ has access to H, y, h and s. Using that y and s are independent of (m, H, h), we obtain $\mathsf{guess}(m \mid H, y, h, s) = \mathsf{guess}(m \mid H, h) = \epsilon$ (where the latter equality is given by the entropy condition). Thus, measuring a random query of $\mathcal{A}_1$ in G_2 yields (pk, m, s) with probability at most ϵ. Therefore, by O2H,

$$\Pr\left[1 \leftarrow \mathsf{G}_1\right] \leq \Pr\left[1 \leftarrow \mathsf{G}_2\right] + 2q_1\sqrt{\epsilon}\,.$$

G_2 *to* G_3 *hop.* Again by O2H - arguing as above that the measurement outcome when measuring a random query of $\mathcal{A}_1$ in G_2 is of the form $(\cdot, m, \cdot)$ with probability at most ϵ - we obtain

$$\Pr\left[1 \leftarrow \mathsf{G}_2\right] \leq \Pr\left[1 \leftarrow \mathsf{G}_3\right] + 2q_1\sqrt{\epsilon}\,.$$

G_3 *to* G_4 *hop.* Here we argue precisely as in the classical case: we relax the winning condition, and we do a syntactical change by introducing $\mathcal{B}_1$, which does the computation of h and the sampling of s and y locally, before it runs $\mathcal{A}_1$. Thus, also here

$$\Pr\left[1 \leftarrow \mathsf{G}_3\right] \leq \Pr\left[1 \leftarrow \mathsf{G}_4\right]\,.$$

G_4 *to* G_5^i *hop.* In G_5^i, the oracle for $\mathcal{A}_0$ is replaced by H_i^{Θ}, which implements H until just before the i-th query, then measures that query in the computational basis to obtain (pk^i, m^i, s^i), and subsequently switches to $\mathrm{H}[(pk^i, m^i, s^i) \mapsto \Theta]$ either from the i-th or the $i+1$-th query onward, with this choice being made uniformly at random.

The goal here is to use the measure-and-reprogram technique from [DFM20] to control the effect of this change. For this purpose, we consider the oracle algorithm $\mathcal{C}^{\mathrm{H}}(\mathcal{A}_0, \mathcal{B}_1)$, which simply runs $m \leftarrow \mathcal{A}_0^{\mathrm{H}}()$ followed by $(\overline{y}, \overline{pk}, \overline{s}) \leftarrow \mathcal{B}_1^{\mathrm{H}[(\cdot,m,\cdot)\mapsto\perp]}(m)$.

We allow $\mathcal{C}$ conditional superposition query access to the random-oracle H, which it uses to forward all queries from $\mathcal{A}_0$ unconditionally and queries x from $\mathcal{B}_1$ conditional on $x \neq (\cdot, m, \cdot)$, returning $\perp$ for $x = (\cdot, m, \cdot)$. $\mathcal{C}^{\mathrm{H}}$ is thus an oracle algorithm with query complexity $q_0 + q_1$ and such that in its second phase the only query inputs with non-zero amplitude are of the form $x \neq (\cdot, m, \cdot)$. At the end of its run, $\mathcal{C}^{\mathrm{H}}(\mathcal{A}_0, \mathcal{B}_1)$ outputs (x, z) with $x := (\overline{pk}, m, \overline{s})$ and $z := \overline{y}$.

Furthermore, we define the verification predicate $V(x, y, z)$ that is 1 if and only if $y = z$. Then, $V(x, \mathrm{H}(x), z) = 1$ if and only if $\mathrm{H}(\overline{pk}, m, \overline{s}) = \overline{y}$, which is the verification condition in G_4. Thus,

$$\Pr[V(x, H(x), z) = 1 : (x, z) \leftarrow \mathcal{C}^{\mathrm{H}}((\mathcal{A}_0, \mathcal{B}_1))] = \Pr\left[1 \leftarrow \mathsf{G}_4(\mathcal{A}_0, \mathcal{B}_1)\right]\,.$$

We are now in a situation where we can apply a modified version of the measure-and-reprogram technique from [DFM20]. Theorem 22 in the Appendix

ensures the existence of a "simulator" $\mathcal{S}^{\mathcal{C}}$ such that, for a random Θ,

$$\frac{\Pr[V(x,H(x),z)=1:(x,z)\leftarrow\mathcal{C}^{\mathrm{H}}]}{(2q+1)^2} \leq \Pr[V(x,\Theta,z)=1\wedge x=x':(x',x,z,i)\leftarrow\langle\mathcal{S}^{\mathcal{C}}(Q),\Theta\rangle]\,,$$

where Q is a set of queries where $\mathcal{S}$ has non-zero probability of success, and $q=|Q|$. We need to describe $\mathcal{S}$ in a bit more detail before we can determine Q.

In the following, let $Q_0 := \{0,\ldots,q_0-1\}$ and $Q_1 := \{q_0,\ldots,q_0+q_1-1\}$ (we let $\mathcal{C}$'s queries start at 0 and run until q_0+q_1-1). The algorithm $\langle\mathcal{S}^{\mathcal{C}},\Theta\rangle$ works as follows: it measures the i-th query of $\mathcal{C}^{\mathrm{H}}$ for a random $i\in Q_0\cup Q_1\cup\{q_0+q_1\}$, with the measurement outcome being x', and then reprograms future queries to input x' by Θ (starting from this or the next query, chosen uniformly at random).[10] Finally, $\mathcal{S}$ outputs x' along with i and the final output (x,z) of $\mathcal{C}$.

In the case of our algorithm $\mathcal{C}$ it is easy to determine Q; $\mathcal{C}$ knows m by the end of its first phase, and by construction 1. never queries any input of the form $(\cdot,m,\cdot)$ from that point on and 2. at the end of its run outputs $x=(\overline{pk},m,\overline{s})$. Hence, for $i\in Q_1$ we have $x\neq x'$ with certainty and thus

$$\Pr_{(x',x,z,i)\leftarrow\langle\mathcal{S}^{\mathcal{C}},\Theta\rangle}[V(x,\Theta,z)=1\wedge x=x'|i\in Q_1]=0\,.$$

It follows that $Q=Q_0$ and therefore $q=q_0$.

Thus, *conditioned* on $i\in Q_0$, $\langle\mathcal{S}^{\mathcal{C}},\Theta\rangle$ works as $\mathcal{A}_0^{\mathrm{H}_i^{\Theta}}()$ followed by $\mathcal{B}_1^{\mathrm{H}[(\cdot,m,\cdot)\mapsto\perp]}(m)$ in G_5^i for a random $i\in Q_0$, and the event $V(x,\Theta,z)=1\wedge x=x'$ matches with the winning condition of G_5^i.

Hence, omitting the specification $(x',x,z,i)\leftarrow\langle\mathcal{S}^{\mathcal{C}},\Theta\rangle$ of the probability space and writing V as a shorthand for $V(x,\Theta,z)$ in the expressions to simplify notation, we obtain

$$\begin{aligned}\Pr[V=1\wedge x=x'] &= \Pr[i\in Q_0]\Pr\left[V=1\wedge x=x'\mid i\in Q_0\right]\\ &\quad+\Pr[i\notin Q_0]\Pr\left[V=1\wedge x=x'\mid i\notin Q_0\right]\\ &= \frac{q_0}{q_0+1}\Pr\left[1\leftarrow\mathsf{G}_5^i\mid i\in Q_0\right]\\ &\quad+\frac{1}{q_0+1}\Pr\left[V=1\wedge x=x'\mid i\notin Q_0\right]. \qquad (8)\end{aligned}$$

Finally, we argue that for $(x',x,z,i)\leftarrow\langle\mathcal{S}^{\mathcal{C}},\Theta\rangle$

$$\Pr[V(x,\Theta,z)=1\wedge x=x'\mid i\notin Q_0]=\frac{1}{|\mathcal{Y}|}\,.$$

Consider $i\notin Q_0$, i.e. $\mathcal{S}$ measures the final output of $\mathcal{C}$. Then $\mathcal{B}_1$ learns no information on Θ before producing its output, and so $V(x,\Theta,z)=1$ with probability $\frac{1}{|\mathcal{Y}|}$.

[10] The choice $i=q_0+q_1$ indicates that the final output $(\overline{pk},m,\overline{s})$ of $\mathcal{C}$ is measured, instead of one of its queries.

Putting all together, we obtain that

$$\frac{\Pr\left[1 \leftarrow \mathsf{G}_4\right]}{(2q_0+1)^2} \leq \frac{q_0}{q_0+1} \Pr\left[1 \leftarrow \mathsf{G}_5^i \mid i \in Q_0\right] + \frac{1}{(q_0+1)\cdot|\mathcal{Y}|} \ .$$

G_5^i *to* G_6^i *hop.* Let $i \in Q_0$ now be fixed. As in the classical case, due to the extra condition in G_5^i, we may replace the occurrence of m with m^i without affecting the output of the game. Thus

$$\Pr\left[1 \leftarrow \mathsf{G}_5^i\right] = \Pr\left[1 \leftarrow \mathsf{G}_6^i\right] \ .$$

Similarly (but not identically) to the classical case, we can argue the latter probability to be small. Indeed, we may assume that $\mathcal{A}_0$ stops after having produced the i-th query, which is then measured. This then means, given that $H[(\cdot, m^i, \cdot) \mapsto \bot])$ blocks the query that would reveal Θ, the output $(\overline{y}, \overline{pk}, \overline{s})$ produced by $\mathcal{B}_1$ is independent of Θ. Thus, the probability that $\overline{y} = \Theta$ is at most $1/|\mathcal{Y}|$, showing that

$$\Pr\left[1 \leftarrow \mathsf{G}_6^i\right] = 1/|\mathcal{Y}| \qquad \text{for all fixed } i \in Q_0.$$

Substituting terms in Equation 8, we obtain that

$$\Pr\left[1 \leftarrow \mathsf{G}_4\right] \leq \frac{(2q_0+1)^2}{|\mathcal{Y}|} \ .$$

Combining the above bounds concludes the proof. □

With this, we have all ingredients to conclude (5), i.e., Theorem 11 for quantum adversaries. Similar to the bottom of Section 4.2, we combine Proposition 13 and Theorem 15, except now replacing the use of linearity to Jensen's inequality.

Given that, in typical applications, $\mathcal{A}_0$ is not adversarially chosen but determined by the environment, and typical applications take place in a classical environment, it makes sense to also consider the "semi-quantum case" where we restrict $\mathcal{A}_0$ to classical queries, but we still allow $\mathcal{A}_1$ to be quantum. By an appropriate mix-and-match of the classical and the (fully) quantum proof, we obtain the following bound then.[11]

Porism 16. *Let* $\mathcal{A} = (\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ *be a computationally unbounded adversary against* Φ-$\$$-$\mathsf{NM}^{\mathrm{H},\bot}$*, with* $\mathcal{A}_0^{\mathrm{H}}$ *making at most* q_0 classical *queries and* $\mathcal{A}_1^{\mathrm{H}}$ *at most* q_1 quantum *queries to the random oracle* H *with co-domain* $\mathcal{Y}$*, and let* aux *be a function. Set*

$$\epsilon := \underset{m \leftarrow \mathcal{A}_0^H}{\mathsf{guess}} \left(m \,\middle|\, H, \mathsf{aux}(m)\right),$$

and let pk *be an arbitrary public key. Then,*

$$\mathbf{Adv}_{\mathrm{H}}^{\Phi\text{-}\$\text{-}\mathsf{NM}^{\mathrm{H},\bot}(pk)}(\mathcal{A}, \mathsf{aux}) \leq q_0 \cdot 2^{-\ell} + 4q_1\sqrt{\epsilon} + (q_0+1)/|\mathcal{Y}| \,.$$

Again, combining Proposition 13 and Porism 16, we conclude (6), i.e. the semi-quantum case of Theorem 11.

[11] The most challenging part of the proof is the superposition queries by $\mathcal{A}_0$.

4.4 *Computational* Entropy in the ROM

Previous definitions of non-resignability (and similar for Φ-non-malleability) in the plain model involved a premise on the min-entropy of the signed message, making it hard for the adversary $\mathcal{A}_1$ to guess it. In the random oracle model, however, the random oracle $\mathrm{H}\colon \mathcal{X} \to \mathcal{Y}$ itself becomes a source of randomness. Thus, by choosing the message to be the hash of a fixed string, it has high entropy but is still easy to guess, by just making one query to the random oracle. So far, we dealt with this by considering the statistical min-entropy and additionally conditioning on the (function table of the) random oracle. However, conditioning on the exponentially large function table of the random oracle is problematic when considering the computational HILL entropy.

Below, we consider a natural way to overcome this issue by introducing a notion of HILL entropy in the ROM. In spirit, a random variable X, which may be correlated with the random oracle, has high HILL entropy, if it is indistinguishable from a random variable that has high statistical entropy, and where indistinguishability must hold for efficient oracle algorithms that may query the random oracle. Looking ahead, unfortunately, our positive result from above nevertheless does not carry over to the computational setting, despite this new notion of HILL entropy in the ROM.

Formally, for two random variables X, Y that may depend on the random oracle H, we define the computational distance relative to H as

$$\delta^H_{s,q}(X,Y) := \max_C \left|\Pr\left[1 \leftarrow C^H(X)\right] - \Pr\left[1 \leftarrow C^H(Y)\right]\right| ,$$

where the maximum is taken over all circuits C of size s and additionally given at most q queries to H. The definition of the (conditional) HILL entropy HILL^H_∞ relative to H is then in line with Definition 1.

Definition 17. *For a pair of random variables (X, Z), possibly dependent on the random oracle H, the conditional HILL entropy (with parameters δ, s and q) relative to the random oracle is defined as*

$$^{\delta,s,q}\mathsf{HILL}^H_\infty(X|Z) := \max_Y H_\infty(Y|Z,H) ,$$

where the max *is over all random variables Y with $\delta^H_{s,q}((X,Z),(Y,Z)) \leq \delta$.*

Note that we can also consider a variant where the indistinguishability is captured via *quantum* circuits, but for the sake of simplicity, here we consider only the classical variant of HILL entropy. Furthermore, similarly to the remark in Section 2.1, we may also use an asymptotic notation and omit the parameters.

Below, we show that our positive result on the salted BUFF transform in the ROM does not carry over to the computational setting when considering the entropy requirement to be computational, i.e., captured by the above notion of HILL entropy in the ROM. Concretely, we show that under the computational Diffie-Hellman (CDH) assumption, there exists a (contrived) signature scheme that is secure in the standard sense (and thus a meaningful signature scheme), but for which the salted BUFF transformation does not provide the computational variant of $\mathsf{NR}^{\mathrm{H},\perp}$. Formally, this is summarized in Theorem 18.

Theorem 18. *Let* H *be a random oracle with co-domain* $\mathcal{Y}$*. Assuming CDH is hard, there exists a signature scheme* $\mathrm{S}^{\mathrm{H}} = (\mathtt{KGen}, \mathtt{Sign}^{\mathrm{H}}, \mathtt{Vfy}^{\mathrm{H}})$ *in the ROM, with a key generation that has no access to* H*, for which* \$-BUFF[S, H]*, obtained by applying the salted BUFF transform (see Fig. 9), satisfies the following:*

There exists a PPT adversary $\mathcal{A}^{\mathrm{H}} := (\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ *given query access to* H*, and a PPT function* aux *without any query to* H *such that*

$$\mathsf{HILL}^{\mathrm{H}}_{\infty}\big(m \,\big|\, pk, \mathtt{aux}(m, pk)\big) \geq \log(|\mathcal{Y}|)\ ,$$
$$\scriptstyle (sk,pk)\leftarrow \$\text{-BUFF}[\mathrm{S},\mathrm{H}].\mathtt{KGen}(1^\lambda)\quad m\leftarrow \mathcal{A}_0^{\mathrm{H}}(pk)$$

and yet they win the game $\mathsf{NR}^{\mathrm{H},\perp}$ *against* \$-BUFF[S, H] *with certainty, i.e.,*

$$\mathbf{Adv}^{\mathsf{NR}^{\mathrm{H},\perp}}_{\$\text{-BUFF}[\mathrm{S},\mathrm{H}]}(\mathcal{A}, \mathtt{aux}) = 1\ .$$

Moreover, S^{H} *is strongly unforgeable under chosen message attacks.*

Let $S_\circ^{\mathrm{H}}$ be an arbitrary CDH-based (strongly unforgeable) signature scheme, with a key generation that does not query H. Define S to be as $S_\circ$, but modified as follows. The key generation additionally produces a pair (a, g^a), and attaches a to the secret key and g^a to the public key. Furthermore, signing attaches a to the actual signature (but will be ignored by the verification). Then, we consider an attacker that produces the message m as $m := (\mathrm{H}(g^{ab}), g^b)$, and the auxiliary function $\mathtt{aux}(m, pk) := g^b$. Then we have

$$\mathsf{HILL}^{\mathrm{H}}_{\infty}\big(m \,\big|\, pk, \mathtt{aux}(m, pk)\big) \geq \mathsf{HILL}^{\mathrm{H}}_{\infty}\big(\mathrm{H}(g^{ab}) \,\big|\, g^a, g^b\big)\ ,$$

which is at least as large as $\log(|\mathcal{Y}|)$ by the CDH assumption; yet when given the signature of m, which includes a (be it BUFF transformed or not), the attacker can compute all of m and so produce a new signature by freshly signing m, which breaks the $\mathsf{NR}^{\mathrm{H},\perp}$ security of \$-BUFF[S, H]. For completeness, the detailed proof is spelled out in Appendix D.

5 Conclusion

Non-resignability was considered to be a meaningful, possibly desirable, additional security property for digital signature schemes, beyond standard unforgeability, and the BUFF transform was believed to be a generic transformation that turns any signature scheme into one that satisfies this additional security property (in the random oracle model). Our work shows that the situation is actually much more negative; the above, incorrect understanding is largely due to prior work not spelling out formal definitions, statements, and proofs rigorously enough, leading to incorrect claims—or to claims that are not defined but easily incorrectly interpreted.

We also show that there is room for positive results, but more work is necessary in that respect. In particular, it is conceivable that the *original* BUFF

transform satisfies our weaker notion of non-resignability in the ROM when having a bound on the statistical entropy, i.e., that Theorem 11 holds (with adjusted bounds) for the original BUFF transform. Achieving our weaker notion of non-resignability in the ROM, by means of a generic transformation, when considering the *computational* variant of the entropy requirement is another challenging open problem.

References

AHU19. Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 269–295. Springer, Heidelberg, August 2019.

BBD+23. Joppe Bos, Olivier Bronchain, Léo Ducas, Serge Fehr, Yu-Hsuan Huang, Thomas Pornin, Eamonn Postlethwaite, Thomas Prest, Ludo Pulles, and Wessel van Woerden. Hawk. Technical report, National Institute of Standards and Technology, 2023. Available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures.

BCFW09. Alexandra Boldyreva, David Cash, Marc Fischlin, and Bogdan Warinschi. Foundations of non-malleable hash and one-way functions. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 524–541. Springer, Heidelberg, December 2009.

BFS11. Paul Baecher, Marc Fischlin, and Dominique Schröder. Expedient non-malleability notions for hash functions. In Aggelos Kiayias, editor, *CT-RSA 2011*, volume 6558 of *LNCS*, pages 268–283. Springer, Heidelberg, February 2011.

CDF+21. Cas Cremers, Samed Düzlü, Rune Fiedler, Marc Fischlin, and Christian Janson. BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures. In *2021 IEEE Symposium on Security and Privacy*, pages 1696–1714. IEEE Computer Society Press, May 2021. Cryptology ePrint Archive version available at https://eprint.iacr.org/archive/2020/1525/20230116:141028 (Version 1.3).

CDF+23. Cas Cremers, Samed Düzlü, Rune Fiedler, Marc Fischlin, and Christian Janson. BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures, 2023. An updated version (Version 1.4) of [CDF+21], available at https://eprint.iacr.org/archive/2020/1525/20231020:082812.

DFM20. Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 602–631. Springer, Heidelberg, August 2020.

DH76. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

dPEK+23. Rafael del Pino, Thomas Espitau, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, Mélissa Rossi, and Markku-Juhani Saarinen. Racoon. Technical report, National Institute of Standards and Technology, 2023. Available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures.

ENST23. Thomas Espitau, Guilhem Niot, Chao Sun, and Mehdi Tibouchi. Squirrels. Technical report, National Institute of Standards and Technology, 2023. Available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures.

FHK+22. Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon - whats next? https://csrc.nist.gov/csrc/media/Presentations/2022/falcon-update/images-media/session-1-prest-falcon-pqc2022.pdf, 2022.

GCF+23. Louis Goubin, Benoît Cogliati, Jean-Charles Faugére, Pierre-Alain Fouque, Robin Larrieu, Gilles Macario-Rat, Brice Minaud, and Jacques Patarin. Prov. Technical report, National Institute of Standards and Technology, 2023. Available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures.

GHHM21. Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the QROM. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 637–667. Springer, Heidelberg, December 2021.

HBD+20. Andreas Hülsing, Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Jean-Philippe Aumasson, Bas Westerbaan, and Ward Beullens. SPHINCS+. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

JCCS19. Dennis Jackson, Cas Cremers, Katriel Cohn-Gordon, and Ralf Sasse. Seems legit: Automated analysis of subtle attacks on protocols that use signatures. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 2165–2180. ACM Press, November 2019.

KBJ+14. Tiffany Hyun-Jin Kim, Cristina Basescu, Limin Jia, Soo Bum Lee, Yih-Chun Hu, and Adrian Perrig. Lightweight source authentication and path validation. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, pages 271–282, 2014.

LDK+20. Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

LZ23. Dongxi Liu and Raymond Zhao. emle. Technical report, National Institute of Standards and Technology, 2023. Available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures.

NIST22. National Institute of Standards and Technology. Call for additional digital signature schemes for the post-quantum cryptography standardization process. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf, 2022.

PCF+23. Jacques Patarin, Benoît Cogliati, Jean-Charles Faugére, Pierre-Alain Fouque, Louis Goubin, Robin Larrieu, Gilles Macario-Rat, and Brice Minaud. Vox. Technical report, National Institute of Standards and Technology, 2023. Available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures.

PFH+20. Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

PS05. Thomas Pornin and Julien P. Stern. Digital signatures do not guarantee exclusive ownership. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *ACNS 05*, volume 3531 of *LNCS*, pages 138–150. Springer, Heidelberg, June 2005.

RSA78. Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, February 1978.

ZBPB17. Jean Karim Zinzindohoué, Karthikeyan Bhargavan, Jonathan Protzenko, and Benjamin Beurdouche. HACL*: A verified modern cryptographic library. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1789–1806. ACM Press, October / November 2017.

A Unachievability of Φ-Non-Malleability as in [CDF+23]

Following the (updated) definition in [CDF+23, Def. 2.4], a hash function H is Φ-non-malleable if for every pair $(\mathcal{A}_0, \mathcal{A}_1)$ of PPT algorithms for which the Hill entropy $\mathsf{HILL}_\infty(x|\mathsf{state})$ is sufficiently large for $(\mathcal{X}, \mathsf{state}) \leftarrow \mathcal{A}_0$ and $x \leftarrow \mathcal{X}$, the probability of winning the game in Fig. 14 is negligible. In case of a hash function family, the hash key is given as input to $\mathcal{A}_0$ and the HILL entropy is then also conditioned on the hash key. In case of H a random oracle, $\mathcal{A}_0$ is given query access to H and the entropy requirement is then on the statistical min-entropy $H_\infty(x|\mathrm{H}, \mathsf{state})$, where one additionally conditions on the (function table of) the random oracle.

The relevant choice of Φ for the non-resignability claim in [CDF+23] is

$$\Phi = \{\phi_{\overline{pk}} : (pk, m) \mapsto (\overline{pk}, m) \,|\, \overline{pk} \in \mathcal{K}\}\,,$$

where $\mathcal{K}$ is the space of all public keys. In more detail, [CDF+23, Lemma 5.7] shows that the considered NR property of the (original) BUFF transform is satisfied *if* the considered hash function H (or hash function family) satisfies the above notion of Φ-non-malleability for this particular choice of Φ.

We show here a simple attack against this notion of Φ-non-malleability for this choice of Φ. The attack applies to *any* hash function (family) H, including the random oracle, and so renders the non-resignability claim in [CDF+23, Lemma 5.7], and in [CDF+23, Theorem 5.5], vacuous.

Φ-NM_0:

1: $(\mathcal{X}, \mathsf{state}) \leftarrow \mathcal{A}_0$
2: $x \leftarrow \mathcal{X}$
3: $y := \mathrm{H}(x)$
4: $(\overline{y}, \phi) \leftarrow \mathcal{A}_1(y, \mathsf{state})$
5: **return** $(\,\mathrm{H}(\phi(x)) = \overline{y} \,\wedge\, \phi(x) \neq x\,)$

Fig. 14. The Φ-non-malleability game, as considered in [CDF+23], but for a fixed hash function H. $\mathcal{X}$ is an efficiently sampleable distribution. The subscript in Φ-NM_0 here is meant to distinguish it from the original definition, which considers some additional auxiliary information.

The attack works as follows. $\mathcal{A}_0$ outputs the distribution $\mathcal{X}$ that samples a random $pk \in \mathcal{K}$ and outputs $x = (pk, 0)$, and $\mathcal{A}_1$ ignores its input $y = \mathrm{H}(pk, 0)$ and simply outputs $(\mathrm{H}(\overline{pk}, 0), \phi_{\overline{pk}})$ for an arbitrary (fixed) $\overline{pk} \in \mathcal{K}$. Note that there is no state information state here, and the entropy condition is satisfied (assuming $\mathcal{K}$ to be sufficiently large). Thus, this is a valid attack that succeeds with probability almost 1; it only fails when the random $pk \in \mathcal{K}$ happens to be $\overline{pk}$.

B A modified measure-and-reprogram lemma

In [DFM20] we find the "measure-and-reprogram technique 2.0" (Theorem 2):

Theorem 19 (Measure-and-reprogram). *Let $\mathcal{X}$ and $\mathcal{Y}$ be finite non-empty sets. There exists a black-box two-stage quantum algorithm $\mathcal{S}$ with the following property. Let $\mathcal{A}$ be an arbitrary oracle quantum algorithm that makes q queries to a uniformly random $H : \mathcal{X} \to \mathcal{Y}$ and that outputs some $x \in \mathcal{X}$ and a (possibly quantum) output z. Then, the two-stage algorithm $\mathcal{S}^{\mathcal{A}}$ outputs some $x \in \mathcal{X}$ in the first stage and, upon a random $\Theta \in \mathcal{Y}$ as input to the second stage, a (possibly quantum) output z, so that for any $x_\circ \in \mathcal{X}$ and any (possibly quantum) predicate V:*

$$\Pr_{H,\Theta}\big[x{=}x_\circ \wedge V(x,\Theta,z) : (x,z) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \Theta\rangle\big] \geq \frac{1}{(2q+1)^2} \Pr_{H}\big[x{=}x_\circ \wedge V(x,H(x),z) : (x,z) \leftarrow \mathcal{A}^{H}\big]\,.$$

Furthermore, $\mathcal{S}$ runs in time polynomial in q, $\log|\mathcal{X}|$, and $\log|\mathcal{Y}|$.

Here $\langle \mathcal{S}^{\mathcal{A}}, \Theta\rangle$ works as follows: First, one of the $q+1$ queries of $\mathcal{A}$ (also counting the final output in register X) is measured, and the measurement outcome x is output by (the first stage of) $\mathcal{S}$. Each of the q actual queries is picked with probability $\frac{2}{2q+1}$, while the final output is picked with probability $\frac{1}{2q+1}$. Then this very query of $\mathcal{A}$ is answered either using the original H *or* using the repro-

grammed oracle $H*\Theta x$, with the choice being made at random[12], while all the remaining queries of $\mathcal{A}$ are answered using oracle $H*\Theta x$. Finally, (the second stage of) $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs.

The theorem follows directly from the above definition of $\mathcal{S}$ and a technical lemma. Let first $|\phi_i\rangle$ be defined as $\mathcal{A}$'s state right before making its i+1st query—with the special case $|\phi_q\rangle$ denoting the final output state—to which we add the superscript $\mathcal{O} \in \{H, H*\Theta x\}$ when all previous queries have been answered using $\mathcal{O}$. Next, we use $\mathcal{A}^{\mathcal{O}}_{i\to j}$ to denote the unitary that brings $\mathcal{A}$ from $|\phi_i\rangle$ to $|\phi_j\rangle$, using $\mathcal{O}$ from the i-th query on. Finally, we use the shorthand $X := |x\rangle\langle x|$. The lemma then reads:

Lemma 20. *Let $\mathcal{A}$ be a q-query oracle quantum algorithm. Then, for any function $H : \mathcal{X} \to \mathcal{Y}$, any $x \in \mathcal{X}$ and $\Theta \in \mathcal{Y}$, and any projection $\Pi_{x,\Theta}$, it holds that*

$$\mathbb{E}_{i,b}\left[\left\|(X\otimes\Pi_{x,\Theta})\left(\mathcal{A}^{H*\Theta x}_{i+b\to q}\right)\left(\mathcal{A}^{H}_{i\to i+b}\right)X\left|\phi_i^H\right\rangle\right\|_2^2\right] \geq \frac{\left\|(X\otimes\Pi_{x,\Theta})\left|\phi_q^{H*\Theta x}\right\rangle\right\|_2^2}{(2q+1)^2},$$

where the expectation is over uniform $(i,b) \in (\{0,\ldots,q-1\}\times\{0,1\})\cup\{(q,0)\}$.

Here the left-hand side corresponds to the success probability of $\mathcal{S}$ with respect to V and Θ, while (in expectation over Θ) the right-hand side is equal to the success probability of the adversary in a normal run, now with respect to V and the original oracle output $H(x)$.

A first observation is that the technical lemma actually proves something slightly stronger than Theorem 19; If we let $\mathcal{S}$ additionally output the measurement outcome x, we get the condition $x = x'$ for free (since the same projector X is used on the query as well as the final output state). On the other hand, for our application it suffices to use a slightly weaker statement (in a different respect) that we obtain by summing over all $x_\circ \in \mathcal{X}$:

$$\begin{aligned}&\Pr_{\Theta}\left[x=x' \wedge V(x,\Theta,z) : (x,x',z) \leftarrow \langle\mathcal{S}^{\mathcal{A}},\Theta\rangle\right]\\ &\geq \frac{1}{(2q+1)^2}\Pr_{H}\left[V(x,H(x),z) : (x,z)\leftarrow\mathcal{A}^H\right].\end{aligned}$$

Next, we will reduce q to account for only those queries where $\mathcal{S}$ has a non-zero probability of measuring the same $x' = x$ that will eventually be output by $\mathcal{A}$, while also satisfying the quantum predicate. The probability here is over the choice of H, Θ and the measurement outcome x', we thus define:

$$\begin{aligned}Q_{\min} := \{i\in\{0,\ldots,q-1\} \mid \exists H\in\mathcal{Y}^{\mathcal{X}}, \exists\Theta\in\mathcal{Y}, \exists x\in\mathcal{X}, \exists b\in\{0,1\} \text{ s.t.}\\ \left\|(X\otimes\Pi_{x,\Theta})\left(\mathcal{A}^{H*\Theta x}_{i+b\to q}\right)\left(\mathcal{A}^{H}_{i\to i+b}\right)X\left|\phi_i^H\right\rangle\right\|_2 \neq 0\}.\end{aligned}$$

Let furthermore Q be any subset of queries such that $Q_{\min} \subseteq Q \subseteq \{0,\ldots,q-1\}$. It will now be easy to prove the following modified lemma:

[12] If it is the final output that is measured then there is nothing left to reprogram, so no choice has to be made.

Lemma 21. *Let $\mathcal{A}$ be a q-query oracle quantum algorithm, with Q as defined above. Then, for any function $H\colon \mathcal{X} \to \mathcal{Y}$, any $x \in \mathcal{X}$ and $\Theta \in \mathcal{Y}$, and any projection $\Pi_{x,\Theta}$, it holds that*

$$\mathop{\mathbb{E}}_{i,b}\left[\left\|(X \otimes \Pi_{x,\Theta})\left(\mathcal{A}^{H*\Theta x}_{i+b\to q}\right)\left(\mathcal{A}^{H}_{i\to i+b}\right)X\left|\phi^H_i\right\rangle\right\|_2^2\right] \geq \frac{\left\|(X\otimes\Pi_{x,\Theta})\left|\phi^{H*\Theta x}_q\right\rangle\right\|_2^2}{(2|Q|+1)^2}\,,$$

where the expectation is over uniform $(i,b) \in (Q\times\{0,1\})\cup\{(q,0)\}$.

Note that the only difference to Lemma 20 is in the expectation on the left-hand side and the denominator on the right-hand side, as indicated in blue. The proof is largely taken from [DFM20], with a small modification which we highlight with a gray background.

Proof. For any $0 \leq i \leq q$, inserting a resolution of the identity and exploiting that

$$\left(\mathcal{A}^{H*\Theta x}_{i+1\to q}\right)\left(\mathcal{A}^{H}_{i\to i+1}\right)(\mathbb{I}-X)\left|\phi^H_i\right\rangle = \left(\mathcal{A}^{H*\Theta x}_{i\to q}\right)(\mathbb{I}-X)\left|\phi^H_i\right\rangle\,,$$

we can write

$$\begin{aligned}
&\left(\mathcal{A}^{H*\Theta x}_{i+1\to q}\right)\left|\phi^H_{i+1}\right\rangle\\
&\quad= \left(\mathcal{A}^{H*\Theta x}_{i+1\to q}\right)\left(\mathcal{A}^{H}_{i\to i+1}\right)(\mathbb{I}-X)\left|\phi^H_i\right\rangle \quad + \left(\mathcal{A}^{H*\Theta x}_{i+1\to q}\right)\left(\mathcal{A}^{H}_{i\to i+1}\right)X\left|\phi^H_i\right\rangle\\
&\quad= \left(\mathcal{A}^{H*\Theta x}_{i\to q}\right)(\mathbb{I}-X)\left|\phi^H_i\right\rangle \quad + \left(\mathcal{A}^{H*\Theta x}_{i+1\to q}\right)\left(\mathcal{A}^{H}_{i\to i+1}\right)X\left|\phi^H_i\right\rangle\\
&\quad= \left(\mathcal{A}^{H*\Theta x}_{i\to q}\right)\left|\phi^H_i\right\rangle - \left(\mathcal{A}^{H*\Theta x}_{i\to q}\right)X\left|\phi^H_i\right\rangle \quad + \left(\mathcal{A}^{H*\Theta x}_{i+1\to q}\right)\left(\mathcal{A}^{H}_{i\to i+1}\right)X\left|\phi^H_i\right\rangle
\end{aligned}$$

Rearranging terms, applying $G^{\Theta}_x = (X\otimes\Pi_{x,\Theta})$ and using the triangle equality, we can thus bound

$$\begin{aligned}
\left\|G^{\Theta}_x\left(\mathcal{A}^{H*\Theta x}_{i\to q}\right)\left|\phi^H_i\right\rangle\right\|_2 \leq{}& \left\|G^{\Theta}_x\left(\mathcal{A}^{H*\Theta x}_{i+1\to q}\right)\left|\phi^H_{i+1}\right\rangle\right\|_2\\
&+ \left\|G^{\Theta}_x\left(\mathcal{A}^{H*\Theta x}_{i\to q}\right)X\left|\phi^H_i\right\rangle\right\|_2\\
&+ \left\|G^{\Theta}_x\left(\mathcal{A}^{H*\Theta x}_{i+1\to q}\right)\left(\mathcal{A}^{H}_{i\to i+1}\right)X\left|\phi^H_i\right\rangle\right\|_2.
\end{aligned}$$

Summing up the respective sides of the inequality over $i = 0,\dots,q-1$, dropping (some of) the zero terms in the summand[13], we get

$$\left\|G^{\Theta}_x\left|\phi^{H*\Theta x}_q\right\rangle\right\|_2 \;\leq\; \left\|G^{\Theta}_x\left|\phi^H_q\right\rangle\right\|_2 + \sum_{\substack{i\,\in\, Q\\ b\in\{0,1\}}}\left\|G^{\Theta}_x\left(\mathcal{A}^{H*\Theta x}_{i+b\to q}\right)\left(\mathcal{A}^{H}_{i\to i+b}\right)X\left|\phi^H_i\right\rangle\right\|_2.$$

By squaring both sides, dividing by $2|Q|+1$ (i.e., the number of terms on the right-hand side), and using Jensen's inequality on the right-hand side, we obtain

$$\frac{\left\|G^{\Theta}_x\left|\phi^{H*\Theta x}_q\right\rangle\right\|_2^2}{2|Q|+1} \leq \left\|G^{\Theta}_x\left|\phi^H_q\right\rangle\right\|_2^2 + \sum_{\substack{0\leq i<q\\ b\in\{0,1\}}}\left\|G^{\Theta}_x\left(\mathcal{A}^{H*\Theta x}_{i+b\to q}\right)\left(\mathcal{A}^{H}_{i\to i+b}\right)X\left|\phi^H_i\right\rangle\right\|_2^2$$

[13] At most (if $Q = Q_{\min}$) we drop all terms that are zero for every choice of b, H, Θ, and x.

and thus, noting that we can write $\left\| G_x^{\Theta} \left| \phi_q^H \right\rangle \right\|_2^2$ as

$$\left\| G_x^{\Theta} \left(\mathcal{A}_{i+b\to q}^{H*\Theta x} \right) \left(\mathcal{A}_{i\to i+b}^{H} \right) X \left| \phi_i^H \right\rangle \right\|_2^2$$

with $i = q$ and $b = 0$,

$$\frac{\left\| G_x^{\Theta} \left| \phi_q^{H*\Theta x} \right\rangle \right\|_2^2}{(2|Q|+1)^2} \leq \mathop{\mathbb{E}}_{i\in Q,b} \left[\left\| G_x^{\Theta} \left(\mathcal{A}_{i+b\to q}^{H*\Theta x} \right) \left(\mathcal{A}_{i\to i+b}^{H} \right) X \left| \phi_i^H \right\rangle \right\|_2^2 \right] .$$

This concludes the proof. □

The corresponding theorem reads as follows:

Theorem 22 (Measure-and-reprogram with stingy simulator). *Let $\mathcal{X}$ and $\mathcal{Y}$ be finite non-empty sets. There exists a black-box two-stage quantum algorithm $\mathcal{S}$ with the following property. Let $\mathcal{A}$ be an arbitrary oracle quantum algorithm that makes q queries to a uniformly random $H : \mathcal{X} \to \mathcal{Y}$ and that outputs some $x \in \mathcal{X}$ and a (possibly quantum) output z, and let V be a (possibly quantum) predicate. For $i \in \{0, \ldots, q-1\}$, define the two-stage algorithm $\mathcal{S}_i^{\mathcal{A}}$ as follows: In the first stage $\mathcal{S}_i$ measures the i-th query of $\mathcal{A}$, and outputs the measurement outcome x'. Then, upon a random $\Theta \in \mathcal{Y}$ as input to the second stage, this very query of $\mathcal{A}$ is answered either using the original H* or *using the reprogrammed oracle $H*\Theta x$, with the choice being made at random, while all the remaining queries of $\mathcal{A}$ are answered using oracle $H*\Theta x$. At the end of its run $\mathcal{S}_i$ then outputs whatever $\mathcal{A}$ outputs (along with i). Now let $Q \subseteq \{0, \ldots, q-1\}$ be such that for all $i \notin Q$ we have $\Pr_{H,\Theta}\left[x' = x \wedge V(x,\Theta,z) : (x',x,z) \leftarrow \langle \mathcal{S}_i^{\mathcal{A}}, \Theta \rangle\right] = 0$. Define $\mathcal{S}(Q)$ to be the algorithm that with probability $\frac{2|Q|}{2|Q|+1}$ picks i uniformly at random from Q and then runs $\mathcal{S}_i$, and with probability $\frac{1}{2|Q|+1}$ chooses $i = q$ and just simulates $\mathcal{A}$ without any measurement or reprogramming, and again outputs whatever $\mathcal{A}$ outputs (along with $x' := x$ and i). We then have*

$$\begin{aligned} &\Pr_{H,\Theta}\left[x' = x \wedge V(x,\Theta,z) : (x',x,z,i) \leftarrow \langle \mathcal{S}^{\mathcal{A}}(Q), \Theta \rangle\right] \\ &\geq \frac{1}{(2|Q|+1)^2} \Pr_H\left[V(x,H(x),z) : (x,z) \leftarrow \mathcal{A}^H\right] . \end{aligned}$$

Furthermore, $\mathcal{S}$ runs in time polynomial in q, $\log|\mathcal{X}|$, and $\log|\mathcal{Y}|$.

C Proof of Theorem 7

Proof. For the signature scheme $\mathrm{S}^{\mathrm{H}} = (\mathtt{KGen}^{\mathrm{H}}, \mathtt{Sign}^{\mathrm{H}}, \mathtt{Vrfy}^{\mathrm{H}})$, we give an adversary $\mathcal{A}^{\mathrm{H}} = (\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ that wins game $\mathsf{NR}_{\mathrm{S}}^{\mathrm{H}}$ with overwhelming probability. The adversary $\mathcal{A}^{\mathrm{H}} = (\mathcal{A}_0^{\mathrm{H}}, \mathcal{A}_1^{\mathrm{H}})$ and the function $\mathtt{aux}^{\mathrm{H}}$ are given in Fig. 6. In the first stage, adversary $\mathcal{A}_0^{\mathrm{H}}$ chooses a random message m that it will output. The auxiliary function, which receives the message m as input, first generates a new key pair $(\overline{sk}, \overline{pk}) \leftarrow \mathtt{KGen}^{\mathrm{H}}(1^\lambda)$, computes $\overline{\sigma} \leftarrow \mathtt{Sign}^{\mathrm{H}}(\overline{sk}, \overline{y})$, and outputs

$h \leftarrow (\overline{pk}, \overline{\sigma})$. In the second state, $\mathcal{A}_1^{\mathrm{H}}$ gets the public key pk, the signature σ (of message m using secret key sk), and the hint h (consisting of $\overline{pk}$ and $\overline{\sigma}$) as input and simply outputs $h = (\overline{pk}, \overline{\sigma})$. It is easy to see that $\mathsf{NR}_{\mathrm{S}}^{\mathrm{H}}$ outputs 1 with overwhelming probability. This is because, by the correctness of S, we have that $\overline{\sigma}$ is a valid signature of m under $\overline{pk}$ with overwhelming probability, and by the high min-entropy of a public key conditioned on H, we have $\overline{pk} \neq pk$ with overwhelming probability.

The remaining is to argue that $\mathcal{A}$ satisfies the entropy condition (3). It holds that

$$\begin{aligned} H_\infty(m \mid pk, \mathtt{aux}^{\mathrm{H}}(m, pk), \mathrm{H}) &= H_\infty(m \mid \mathtt{aux}^{\mathrm{H}}(m, pk), \mathrm{H}) \\ &= H_\infty(m \mid \overline{pk}, \mathtt{Sign}^{\mathrm{H}}(\overline{sk}, m), \mathrm{H}) \geq \omega(\log \lambda)\,. \end{aligned}$$

The first equality holds by noticing that $m \to (\mathtt{aux}^{\mathrm{H}}(m, pk), \mathrm{H}) \to pk$ forms a Markov chain, the second equality holds by the construction of $\mathtt{aux}^{\mathrm{H}}$, and the last inequality holds from the entropy requirement. Collecting the above yields

$$\mathbf{Adv}_{\mathrm{S}}^{\mathsf{NR}^{\mathrm{H}}}(\lambda, \mathcal{A}, \mathtt{aux}) \geq 1 - \mathrm{negl}(\lambda)\,,$$

which concludes the proof. □

D Prove of Theorem 18

Let G be a cyclic group of order n generated by g, for which CDH is hard. The core of this negative result is a generic compiler RS_g as defined in Fig. 15. The compiler RS_g appends $a \leftarrow \{1, \ldots, n\}$ to sk and g^a to pk during key generation, and appends a to the signature during signing, transforming any signature scheme $\mathrm{S}_\circ^{\mathrm{H}}$ into another $\mathsf{RS}_g[\mathrm{S}_\circ, H]$ that can easily be re-signed, as will be shown later. This transformation does not make additional queries to H during key generation. Moreover, as can be clearly seen by construction, RS_g preserves strong (resp. weak) unforgeability, and additionally commutes with \$-BUFF, i.e.

$$\$\text{-BUFF}[\mathsf{RS}_g[\mathrm{S}_\circ, \mathrm{H}], \mathrm{H}] \equiv \mathsf{RS}_g[\$\text{-BUFF}[\mathrm{S}_\circ, \mathrm{H}], \mathrm{H}]\,, \tag{9}$$

where we use "$\equiv$" to denote that signatures on both sides behave identically (up to output re-formatting). As a direct consequence, the salted BUFF applied to any such compiled signature $\mathsf{RS}_g[\mathrm{S}_\circ, \mathrm{H}]$ yields another RS_g-transformed signature, thus can still be re-signed.

Now we are ready to spell out the attack $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ and $\mathtt{aux}$ in Fig. 16 against $\mathsf{NR}^{\mathrm{H},\perp}$ in Fig. 10. From the construction, we immediate see that they win the game with certainty, i.e.

$$\mathbf{Adv}_{\mathsf{RS}_g[\mathrm{S}_\circ, H]}^{\mathsf{NR}^{\mathrm{H},\perp}}(\mathcal{A}, \mathtt{aux}) = 1\,. \tag{10}$$

Then, it follows from Lemma 23 that the above attack indeed satisfies the HILL entropy requirement, namely

$$\mathsf{HILL}_\infty^{\mathrm{H}}{}_{\substack{(sk,pk) \leftarrow \mathsf{RS}_g[\mathrm{S}_\circ, \mathrm{H}].\mathtt{KGen}(1^\lambda) \\ m \leftarrow \mathcal{A}_0^{\mathrm{H}}(pk)}} \left(m \;\middle|\; pk, \mathtt{aux}(m, pk)\right) \geq \log(|\mathcal{Y}|)\,.$$

$\mathtt{KGen}(1^\lambda)$	$\mathtt{Sign}^{\mathrm{H}}(sk, m)$	$\mathtt{Vfy}^{\mathrm{H}}(pk', m, \sigma')$
$(\overline{sk}, \overline{pk}) \leftarrow \mathrm{S}_\circ.\mathtt{KGen}(1^\lambda)$	$(a, \overline{sk}) := sk$	$(g^a, pk) := pk$
$a \leftarrow \{1, \ldots, n\}$	$\overline{\sigma} \leftarrow \mathrm{S}_\circ.\mathtt{Sign}(sk, m)$	$(a, \overline{\sigma}) := \sigma$
$sk := (a, sk)$	**return** $\sigma := (a, \overline{\sigma})$	**return** $\mathrm{S}_\circ.\mathtt{Vfy}(\overline{pk}, m, \overline{\sigma})$
$pk := (g^a, pk)$		
return (sk, pk)		

Fig. 15. The compiler $\mathsf{RS}_g[\mathrm{S}_\circ, H]$ for a signature scheme $\mathrm{S}_\circ^{\mathrm{H}}$ and a hash function H.

Putting things together, we conclude Theorem 18.

$\mathcal{A}_0^{\mathrm{H}}(pk)$	$\mathtt{aux}(m, pk)$	$\mathcal{A}_1^{\mathrm{H}}(pk, \sigma, h)$
$(g^a, \overline{pk}) := pk$	$(-, g^b) := m$	$(a, -) := \sigma$
$b \leftarrow [n]$	**return** $h := g^b$	$g^b := h$
return $m := (\mathrm{H}(g^{ab}), g^b)$		**return** g^{ab}

Fig. 16. Attacker against $\mathsf{NR}^{\mathrm{H},\perp}$ in terms of HILL entropy

Lemma 23. *Let* H *be a random oracle with co-domain $\mathcal{Y}$, and G be a cyclic group of order n generated by g, for which CDH is hard. Then we have*

$$\mathsf{HILL}^{\mathrm{H}}_{\substack{\infty \\ a,b \leftarrow [n]}}\left(H(g^{ab}) \mid g^a, g^b\right) \geq \log(|\mathcal{Y}|) \ .$$

Proof. Let $y \leftarrow \mathcal{Y}$ be a fresh randomness, and C^{H} be a q-query polynomial-circuit-size oracle algorithm where q is polynomially bounded. To distinguish between $(H(g^{ab}), g^a, g^b)$ and (y, g^a, g^b), the algorithm C^{H} has to query g^{ab} in one of its queries, i.e.

$$\begin{aligned}&\left|\Pr\left[1 \leftarrow C^H(H(g^{ab}), g^a, g^b)\right] - \Pr\left[1 \leftarrow C^H(y, g^a, g^b)\right]\right| \\ &\leq \sum_{i \in [q]} \Pr\left[g^{ab} \leftarrow C_i(g^a, g^b)\right] \ ,\end{aligned}$$

where the circuit C_i simulates C^H and its queries to H via lazy sampling, and then simply output the ith query to H. Since each C_i is still polynomial-circuit-size, it is unable to produce g^{ab} except with negligible probability, i.e.

$$\Pr_{a,b \leftarrow [n]}[g^{ab} \leftarrow C_i(g^a, g^b)] \leq \mathrm{negl}(\lambda) \ ,$$

for some security parameter λ that has been kept implicit. Putting things together, we obtain

$$\mathsf{HILL}_{\substack{\infty \\ a,b \leftarrow [n]}}\left(H(g^{ab}) \mid g^a, g^b\right) \geq H_\infty(y \mid H, g^a, g^b) = \log(|\mathcal{Y}|) \ ,$$

where the last equality follows from the independence between y and (H, g^a, g^b). □