Hide-and-Seek and the Non-Resignability of the BUFF Transform

Jelle Don[1], Serge Fehr[1,2], Yu-Hsuan Huang[1], Jyun-Jie Liao[3], and Patrick Struck[4]

[1] Centrum Wiskunde & Informatica (CWI), Amsterdam, The Netherlands
[2] Mathematical Institute, Leiden University, Leiden, The Netherlands
[3] Cornell University, Ithaca, NY, USA
[4] University of Konstanz, Konstanz, Germany
{jelle.don, serge.fehr, yhh}@cwi.nl,
jjliao@cs.cornell.edu, patrick.struck@uni.kn

Abstract. The BUFF transform, due to Cremers et al. (S&P'21), is a generic transformation for digital signature scheme, with the purpose of obtaining additional security guarantees beyond unforgeability: exclusive ownership, message-bound signatures, and non-resignability. Non-resignability (which essentially challenges an adversary to re-sign an unknown message for which it only obtains the signature) turned out to be a delicate matter, as recently Don et al. (CRYPTO'24) showed that the initial definition is essentially unachievable; in particular, it is *not* achieved by the BUFF transform. This led to the introduction of new, weakened versions of non-resignability, which are (potentially) achievable. In particular, it was shown that a *salted* variant of the BUFF transform does achieves some weakened version of non-resignability. However, the salting requires additional randomness and leads to slightly larger signatures. Whether the *original* BUFF transform also achieves some meaningful notion of non-resignability remained a natural open question.
In this work, we answer this question in the affirmative. We show that the BUFF transform satisfies the (almost) strongest notions of non-resignability one can hope for, facing the known impossibility results. Our results cover both the statistical and the computational case, and both the classical and the quantum setting. At the core of our analysis lies a new security game for random oracles that we call *Hide-and-Seek*. While seemingly innocent at first glance, it turns out to be surprisingly challenging to rigorously analyze.

1 Introduction

Digital Signatures and the BUFF Transform. Digital signatures are at the very heart of modern cryptography. The gold standard security notion for digital signature schemes is (strong) unforgeability against chosen message attacks. However, in certain applications, additional security properties are desirable, or even necessary. For example, [JCCS19] showed that the *"Dynamically Recreatable Key"* protocol [KBJ+14] is insecure if the signature scheme used in the protocol does not additionally offer some sort of non-malleability property that, informally, requires it to be hard to turn a signature for an unknown message into a signature for the same message but under a different public key (with a possibly known secret key). This property was named *non-resignability* in [JCCS19], and formally defined later in [CDF+21], along with two more properties: *exclusive ownership* and *message-bound signatures*. On top, [CDF+21] introduced a generic transformation, the *BUFF transform*, which can be applied to any signature scheme, and it was argued that the transformed signature scheme then satisfies these three additional properties (in the random oracle model). The transform is very simple: instead of signing the message m, a BUFF-transformed signature scheme signs the hash $H(\mathsf{pk}, m)$ of the public key and the message, and this hash value is also appended to the signature.

Motivated by the fact that the NIST call for additional post-quantum signatures [NIST22] explicitly mentioned the above as "*additional desirable security properties*", several of the NIST post-quantum signature submissions have the BUFF transform built in, or mention the possibility of applying the BUFF transform to the proposed scheme.

Recent Development. Somewhat surprisingly given the apparently clear situation around the BUFF transform, the recent work [DFHS24] showed that the question of defining and achieving *non-resignability*

is actually more subtle. Concretely, it was shown that non-resignability, as defined in [CDF+21], is almost unachievable as a matter of fact, both in the plain model and in the random oracle model.[5] In particular, it follows that the BUFF transform does *not* achieve non-resignability (as defined in [CDF+21]). The apparent contradiction to the positive claim from [CDF+21] comes from the fact that the proof in [CDF+21] relied on a non-malleability claim for the random oracle that was taken from [BFS11], and which turned out to be false.

Towards showing a positive result, [DFHS24] introduced $\mathsf{NR}^{H,\perp}$, a weaker version of the original definition of non-resignability (in the ROM), and they showed that a *salted version* of the BUFF transform satisfies $\mathsf{NR}^{H,\perp}$. The situation is actually more complicated in that the non-resignability definition involves an entropy condition, of which one can consider a statistical or a computational variant. While the impossibility of [DFHS24] holds for both, the positive result on $\mathsf{NR}^{H,\perp}$ for the salted BUFF transform holds for the statistical variant only, and provably not for the computational variant.[6]

In reaction to the negative results from (an early version of) [DFHS24], the authors of [CDF+21] updated their paper to [CDF+23] by weakening their definition of non-resignability and tried to argue that the (original) BUFF transform satisfies their weakened definition; however, their argument relies on an assumption that is shown to be false in [DFHS24].

Thus, the bottom line is that the following question has remained open:

Does the BUFF transform satisfy some non-trivial notion(s) of non-resignability?

Our Results. In this work, we answer the above question in the affirmative. Concretely, we introduce yet another variant of non-resignability, $\mathsf{sNR}^{H,\perp}$, and we show that the (original) BUFF transform satisfies $\mathsf{sNR}^{H,\perp}$, both in the statistical setting, where the entropy condition holds statistically and adversaries may be computationally unbounded, and in the computational setting, where the entropy condition holds computationally and adversaries have bounded computing power only.

In the statistical setting, $\mathsf{sNR}^{H,\perp}$ is strictly stronger than $\mathsf{NR}^{H,\perp}$; in the computationally setting, the two notions are (probably) incomparable, yet $\mathsf{sNR}^{H,\perp}$ is strictly stronger than the notion considered in [CDF+23]. Therefore, given that [DFHS24] showed that the BUFF transform does not satisfy $\mathsf{NR}^{H,\perp}$ in the computationally setting, our results appear to be the best we can hope for towards proving positive results on the non-resignability of the BUFF transform.

Our approach is inspired by the proof in [DFHS24] for the salted BUFF transform. Indeed, on the technical level, we can recycle and adjust some of the arguments, although we avoid the detour via some tailor-made non-malleability property for the random oracle. The crucial part of course is when [DFHS24] exploits the salt that originates from the salted BUFF transform, which we cannot do, given that we consider the original, unsalted variant. Instead, we capture the crucial, missing piece in the form of a particular, simple game in the random oracle model, which we call *Hide-and-Seek*, and we reduce the non-resignability property of the BUFF transform to the hardness of winning Hide-and-Seek. In essence, the game asks to find x when given $H(x)$ and query-bounded access to H, where x may be chosen arbitrarily *dependent* on H subject to the condition that it is hard to guess when given access to H only, i.e., without being given $H(x)$.

Despite its simplicity and harmless appearance, this game turns out to be surprisingly tricky to analyze. Thus, the technical core of this work is in analyzing Hide-and-Seek and showing that it is hard to win, both in the statistical and in the computational setting, and both in the classical and in the quantum ROM.

Related Work. The relevance of the BUFF security notions can be traced to attacks [Aye15,JCCS19], which exploit the absence of additional security properties like exclusive ownership, message-bound signatures, and non-resignability. The former security notion (exclusive ownership) was first mentioned by Pornin and Stern [PS05] which can further be traced back to [MS04,BWM99]. Along with defining exclusive ownership,

[5] There are hypothetical signature schemes to which the attack from [DFHS24] does not apply, though we are not aware of any natural scheme for which that is the case.

[6] We note that the statistic and the computational variants of $\mathsf{NR}^{H,\perp}$ are incomparable: in the computational case, the adversary is restricted in its computational power but is bound to a weaker entropy condition.

Pornin and Stern also give three generic transformations that achieve exclusive ownership. The other security notions (message-bound signatures and non-resignability) were formalized in [CDF+21].

In very recent work, Aulbach et al. [ADM+24] analyzed the BUFF security of the schemes submitted to the recent NIST standardization process for post-quantum signature schemes [NIST22], though considering an even weaker notion of non-resignability than $\mathsf{NR}^{H,\perp}$ (where there is no auxiliary information at all).

Also very recently, Düzlü et al. [DFF24] reconsider the BUFF security notions for Falcon [PFH+20], exploiting the particular form of a Falcon signature, and they argue that all that is needed is to replace the hash $H(r, m)$ in a Falcon signature computation by $H(r, \mathsf{pk}, m)$; thus, the hash can be "recycled" (this was argued in [CDF+21] already), but also, it does *not* need to be appended to the signature as in the BUFF transform (in line with the lighter transform by Pornin and Stern [PS05]). Regarding non-resignability, they consider the variant from [CDF+23], which is weaker than $\mathsf{NR}^{H,\perp}$, but relax the HILL entropy requirement to a bound on the *computational unpredictability*, which makes the definition stronger in that aspect. Thus, strictly speaking, the considered variant is incomparable with the computational versions of $\mathsf{NR}^{H,\perp}$ and $\mathsf{sNR}^{H,\perp}$.

2 Preliminary

We start by briefly spelling out the notions of guessing probability and min-entropy, and recalling the random oracle model. Then, we introduce $\mathsf{sNR}^{H,\perp}$, the variant of non-resignability we consider in this paper. Finally, we recall the BUFF transform, as introduced in [CDF+21].

2.1 Guessing probability and min-entropy

For a random variable X over a finite set $\mathcal{X}$, the *guessing probability* and the *min-entropy* are respectively defined as

$$\mathsf{guess}(X) := \max_x \Pr[X = x] \qquad \text{and} \qquad \mathsf{H}_\infty(X) := -\log\big(\mathsf{guess}(X)\big)$$

where here and in the remainder, log is the binary logarithm. For random variables X and Y over respective finite sets $\mathcal{X}$ and $\mathcal{Y}$, the *conditional* guessing probability is defined as

$$\mathsf{guess}(X \mid Y) := \sum_y \Pr[Y = y] \max_x \Pr[X = x \mid Y = y] \,.$$

It is well known that $\mathsf{guess}(X \mid Y) = \max_f \Pr[X = f(Y)]$, where the maximization is over all (deterministic or randomized) functions $f : \mathcal{Y} \to \mathcal{X}$. In line with the unconditional case above, the *conditional* min-entropy is then given by $\mathsf{H}_\infty(X \mid Y) := -\log\big(\mathsf{guess}(X \mid Y)\big)$.

2.2 The Random-Oracle model

Throughout, we consider the random oracle model (ROM) [BR93], i.e., we consider a uniformly randomly function $H \colon \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are suitably chosen, finite sets, and algorithms are (only) given *oracle* access to H. By default, we consider algorithms to be *classical* and thus make classical queries to H; however, we also consider the quantum setting, in which case we then explicitly refer to *quantum queries* and/or the *quantum* random oracle model (QROM) [BDF+11]. In some case, we also consider an algorithm that can make an unbounded number of queries to H, in which case it then is irrelevant if these are classical or quantum.

2.3 Non-resignability

Let $\mathcal{S} = (\mathsf{KGen}^H, \mathsf{Sign}^H, \mathsf{Vrfy}^H)$ be a signature scheme, where we make explicit that we consider schemes in the random oracle model, and thus key-generation, signing, and verification are given oracle access to H. As

usual, we require KGen^H, Sign^H, and Vrfy^H to be PPT, and it is understood that KGen^H takes the unary representation of λ as input, where λ is the security parameter. By default, we denote the message space by $\mathcal{M}$ and the public-key and secret-key spaces by $\mathcal{PK}$ and $\mathcal{SK}$, respectively. Without loss of generality, we assume that the public key pk can be efficiently computed from its corresponding secret key sk.

In this work, we consider a new variant of *non-resignability*, denoted $\mathsf{sNR}^{H,\perp}$. It is similar in spirit as $\mathsf{NR}^{H,\perp}$ introduced in [DFHS24]; in particular, a crucial aspect is that aux is not given access to H, but we additionally provide the adversary with the secret key sk, and we adjust the entropy condition correspondingly (see below for a more detailed comparison). The security game is shown in Fig. 1. It is played by randomized oracle algorithms[7],

$$\mathcal{D}^H : \mathcal{SK} \to \mathcal{M} \qquad \text{and} \qquad \mathcal{A}^H : \mathcal{SK} \times \mathcal{SGN} \times \mathcal{AUX} \to \mathcal{PK} \times \mathcal{SGN}$$

given query access to H, referred to as *adversaries*, and a randomized algorithm $\mathsf{aux} : \mathcal{SK} \times \mathcal{M} \to \mathcal{AUX}$ with no access to H, referred to as *hint function*.[8]

$\mathsf{sNR}_{\mathcal{S}}^{H,\perp}(\mathcal{D}, \mathcal{A}, \mathsf{aux})$:
1: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}^H$
2: $m \leftarrow \mathcal{D}^H(\mathsf{sk})$
3: $\sigma \leftarrow \mathsf{Sign}^H(\mathsf{sk}, m)$
4: $(\mathsf{pk}', \sigma') \leftarrow \mathcal{A}^H(\mathsf{sk}, \sigma, \mathsf{aux}(m, \mathsf{sk}))$
5: **return** $\mathsf{pk} \neq \mathsf{pk}' \wedge \mathsf{Vrfy}^H(\mathsf{pk}', m, \sigma') = 1$

Fig. 1. Our new variant of the non-resignability game $\mathsf{sNR}^{H,\perp}$.

While playing $\mathsf{sNR}^{H,\perp}$, we consider restricted ($\mathcal{S}$-dependent) classes of adversaries with a give bound h on the entropy

$$\mathsf{H}_{\infty}\limits_{\substack{(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{KGen}^H \\ m \leftarrow \mathcal{D}^H(\mathsf{sk})}} (m \mid H, \mathsf{sk}, \mathsf{aux}(\mathsf{sk}, m)) \geq h\,. \tag{1}$$

For now we only consider the statistical variant, where we take an arbitrary but fixed security parameter for $\mathcal{S}$, where $\mathcal{D}$, $\mathcal{A}$ and aux may be computationally unbounded and we only limit their query complexity, and where the entropy requirement holds statistically, i.e., as in (1). The computational setting is handled later in Section 5; there, $\mathcal{D}, \mathcal{A}$ and aux are restricted to be (uniform or non-uniform) PPT algorithms, and the entropy requirement is expressed via HILL entropy (which causes some complications given that (1) conditions on the entire function table of H).

Informally, we say that a signature scheme $\mathcal{S} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ is *non-resignable* if for all $\mathcal{D}$, $\mathcal{A}$ and any hint function aux that satisfy the statistical entropy condition (1) for sufficiently large h, the probability of winning the $\mathsf{sNR}^{H,\perp}$ game, i.e.,

$$\mathbf{Adv}^{\mathsf{sNR}^{H,\perp}}(\mathcal{D}, \mathcal{A}, \mathsf{aux}) := \Pr\left[1 = \mathsf{sNR}_{\mathcal{S}}^{H,\perp}(\mathcal{D}, \mathcal{A}, \mathsf{aux})\right]\,,$$

is small.

The recent developments have shown that formalizing non-resignability is a non-trivial task, and different weaker variants of the original (unachievable) version have been proposed. We quickly discuss here how $\mathsf{sNR}^{H,\perp}$ relates to those variants; namely, we show that is stronger than the versions proposed in [DFHS24] and [CDF+23].

[7] Here and in the remainder, we borrow from set notation to indicate the input and output space of (oracle) algorithms. In case of an algorithm that takes no input, we write the singleton set $\{\perp\}$ as domain.

[8] The hint function may be randomized, but we refer to it as a function for convenience.

Comparison with Non-Resignability from [DFHS24]. The difference to $\mathsf{NR}^{H,\perp}$ as defined in [DFHS24] is that $\mathsf{sNR}^{H,\perp}$ provides the $\mathcal{D}$, $\mathcal{A}$ and the hint function aux with the secret key sk, whereas $\mathsf{NR}^{H,\perp}$ only provides the public key pk (recall that we assume that pk can be computed from sk). This of course gives more power to the adversary. The other difference lies in the entropy requirement: for $\mathsf{NR}^{H,\perp}$, the message is required to have high entropy conditioned on pk (and aux) only, i.e.,

$$\mathsf{H}_\infty(m \mid H, \mathsf{pk}, \mathsf{aux}(\mathsf{pk}, m)) \geq h$$

whereas $\mathsf{sNR}^{H,\perp}$ requires (1) to hold, which conditions on sk instead; this seems to be a stronger restriction, but we observe that for $m \leftarrow \mathcal{D}(\mathsf{pk})$, produced by a $\mathcal{D}$ that only gets the public key as input (as in $\mathsf{NR}^{H,\perp}$),

$$\mathsf{H}_\infty(m \mid H, \mathsf{pk}, \mathsf{aux}(\mathsf{pk}, m)) = \mathsf{H}_\infty(m \mid H, \mathsf{sk}, \mathsf{aux}(\mathsf{pk}, m))$$

since $\mathsf{sk} \to (H, \mathsf{pk}, \mathsf{aux}(\mathsf{pk}, m)) \to m$ forms a Markov chain then. This implies that any attack against $\mathsf{NR}^{H,\perp}$ can be cast as an attack against $\mathsf{sNR}^{H,\perp}$ with the same entropy bound, making the latter a stronger security notion.

Comparison with Non-Resignability from [CDF+23]. We first note that [CDF+23] defines non-resignability only in the computational setting, so we compare it with the computational version of $\mathsf{sNR}^{H,\perp}$. While we have postponed the exact definition to Section 5, the high level reasoning can still be understood. First of all, in [CDF+23] the side information on m (given by aux in our case) is required to be *computationally independent* of m, which is equivalent to allowing *no side information at all* when considering computationally bounded adversaries. Furthermore, in line with $\mathsf{sNR}^{H,\perp}$, the entropy condition (though phrased in terms of HILL entropy) is required to hold when conditioning on the secret key sk. But on the other hand and in the spirit of $\mathsf{NR}^{H,\perp}$, the adversaries are only given pk as input, and not sk. Altogether, this makes their notion weaker than our computational version of $\mathsf{sNR}^{H,\perp}$, which provided sk as input to the adversaries.

2.4 BUFF Transformation

The BUFF transform, as proposed in [CDF+21], transforms any signature scheme $\mathcal{S}$ into another signature scheme $\mathsf{BUFF}[\mathcal{S}, H]$. The transformation is described in Fig. 2; in essence, $\mathsf{BUFF}[\mathcal{S}, H]$ signs a message m by signing the hash $H(\mathsf{pk}, m)$ and additionally appending this hash value to the signature.

KGen'^H:
1: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}^H$
2: **return** $(\mathsf{sk}, \mathsf{pk})$

$\mathsf{Sign}'^H(\mathsf{sk}, m)$:
1: $y \coloneqq (\mathsf{pk}, m)$
2: $\sigma \leftarrow \mathsf{Sign}^H(\mathsf{sk}, y)$
3: $\sigma' \coloneqq (\sigma, y)$
4: **return** σ'

$\mathsf{Vrfy}'^H(\mathsf{pk}, m, \sigma')$:
1: $(\sigma, y) \coloneqq \sigma'$
2: **return** $\mathsf{Vrfy}^H(\mathsf{pk}, y, \sigma) \wedge$
3: $\qquad y = H(\mathsf{pk}, m)$

Fig. 2. The signature scheme $\mathsf{BUFF}[\mathcal{S}, H] = (\mathsf{KGen}'^H, \mathsf{Sign}'^H, \mathsf{Vrfy}'^H)$, obtained from applying the BUFF transform to $\mathcal{S} = (\mathsf{KGen}^H, \mathsf{Sign}^H, \mathsf{Vrfy}^H)$.

Here and in the remainder when considering the BUFF transform, we take it as understood that the random oracle $H\colon \mathcal{X} \to \mathcal{Y}$ has fitting domain and range, i.e., $\mathcal{X} \supseteq \mathcal{K} \times \mathcal{M}$ and $\mathcal{Y} \subseteq \mathcal{M}$, so that the BUFF transform is well defined.

3 Hide-and-Seek and the Non-resignability of BUFF

Our goal is to prove the non-resignability (in the sense of $\mathsf{sNR}^{H,\perp}$) of the BUFF transform, which signs a message m by signing $H(\mathsf{pk}, m)$, with the hash value then appended to the signature. Clearly, for this

non-resignability to hold, it must *necessarily* be hard to recover m from $H(\mathsf{pk}, m)$. This hardness may look trivial at first glance, since H is (typically) compressing, and modeled as a random oracle; however, it turns out to be not trivial at all. The reason is that in the $\mathsf{sNR}^{H,\perp}$ game, m is produced arbitrarily and dependent on H, with the only promise being that m is hard to guess from scratch (i.e., when $H(\mathsf{pk}, m)$ is not given).

In this section, we formally capture (a particular formulation of) this hardness via a game, which we call *Hide-and-Seek*, and we show that hardness of winning Hide-and-Seek is *sufficient* for proving the non-resignability of the BUFF transform. The main technical challenge then lies in proving that Hide-and-Seek is hard to win, which we do in Sect. 4.

Throughout the remainder, let $\mathcal{X}, \mathcal{Y}$, and $\mathcal{Z}$ be finite non-empty sets, and let $H\colon \mathcal{X} \to \mathcal{Y}$ be the random oracle.

3.1 The Hide-and-Seek Game

The Hide-and-Seek game is played by two adversaries $\mathcal{D}$ and $\mathcal{A}$: the (possibly query-unbounded) *hider* $\mathcal{D}^H\colon \{\perp\} \to \mathcal{X} \times \mathcal{Z}$, and the query-bounded *seeker* $\mathcal{A}^H\colon \mathcal{Y} \times \mathcal{Z} \to \mathcal{X}$ that is allowed to make at most q queries to H. First, $\mathcal{D}^H$ chooses a challenge $x \in \mathcal{X}$ together with a hint $z \in \mathcal{Z}$ and "hides" x as $H(x)$, and then $\mathcal{A}^H$ is supposed to find x from $H(x)$ and z. The game is formally specified as follows:

$\mathsf{HnS}^H(\mathcal{D}, \mathcal{A})$:
1: $(x, z) \leftarrow \mathcal{D}^H$
2: **return** $x = \mathcal{A}^H(H(x), z)$

In line with the entropy condition in $\mathsf{sNR}^{H,\perp}$, we require x to be statistically hidden given H and z. I.e., we require that

$$\mathsf{guess}(x \mid H, z) \leq \epsilon \tag{2}$$

for some small $\epsilon > 0$. Informally, we say that the random oracle H satisfies the Hide-and-Seek property, or HnS^H for short, if for every such pair of $\mathcal{D}, \mathcal{A}$ as above, the winning probability, given as

$$\mathbf{Adv}^{\mathsf{HnS}^H}(\mathcal{D}, \mathcal{A}) \coloneqq \Pr\left[1 = \mathsf{HnS}^H(\mathcal{D}, \mathcal{A})\right] = \Pr_{(x,z)\leftarrow\mathcal{D}^H}\left[x = \mathcal{A}^H(H(x), z)\right],$$

is small.

As mentioned above already, what is tricky about this game is that x (and z) may depend arbitrarily on H, subject to the bound (2) on the guessing probability. Because of this, known results on inverting the random oracle do not apply, and it may not be fully clear whether we can actually expect it to be hard to win, i.e., that there is no sneaky way to win the game. We discuss this in more detail in Sect. 4, where we then analyze Hide-and-Seek and prove that it *is* hard to win after all.

3.2 Reducing $\mathsf{sNR}^{H,\perp}$ of BUFF to Hide-and-Seek

In the following statement, we reduce the $\mathsf{sNR}^{H,\perp}$ security of the BUFF transform $\mathsf{BUFF}[\mathcal{S}, H]$ of a signature scheme $\mathcal{S} = (\mathsf{KGen}^H, \mathsf{Sign}^H, \mathsf{Vrfy}^H)$ to the hardness of winning the Hide-and-Seek game HnS^H. In the lemma statement, the parameters q_K and q_S refer to (an upper bound on) the number of queries to H that KGen^H and Sign^H perform.

Lemma 1. *Let $\mathcal{D}^H : \mathcal{SK} \to \mathcal{M}$ and $\mathcal{A}^H : \mathcal{SK} \times \mathcal{SGN} \times \mathcal{AUX} \to \mathcal{PK} \times \mathcal{SGN}$ be $\mathsf{sNR}^{H,\perp}$-adversaries against $\mathsf{BUFF}[\mathcal{S}, H]$ for some $\mathsf{aux} : \mathcal{SK} \times \mathcal{M} \to \mathcal{AUX}$, making at most q_D and q_A queries to H, respectively. Then there exists a hider $\bar{\mathcal{D}} : \{\perp\} \to \mathcal{X} \times \mathcal{Z}$ and a seeker $\bar{\mathcal{A}} : \mathcal{Y} \times \mathcal{Z} \to \mathcal{X}$ with $\mathcal{Z} = \mathcal{SK} \times \mathcal{AUX}$, where $\bar{\mathcal{A}}$ makes at most $q_A + q_S$ queries to H, and such that*

$$\mathsf{H}_\infty^{(x,z)\leftarrow\bar{\mathcal{D}}^H}(x \mid H, z) = \mathsf{H}_\infty^{\substack{(\mathsf{sk},\mathsf{pk})\leftarrow\mathsf{KGen}^H \\ m\leftarrow\mathcal{D}^H(\mathsf{sk})}}(m \mid H, \mathsf{sk}, \mathsf{aux}(\mathsf{sk}, m)) \tag{3}$$

and

$$\mathbf{Adv}^{\mathsf{sNR}^{H,\perp}}_{\mathsf{BUFF}[\mathcal{S},H]}(\mathcal{D},\mathcal{A},\mathsf{aux}) \leq (q_A+q_S)\cdot\mathbf{Adv}^{\mathsf{HnS}^H}(\bar{\mathcal{D}},\bar{\mathcal{A}}) + q_K\epsilon + \frac{q_D+1}{|\mathcal{Y}|}\ . \tag{4}$$

In the case $\mathcal{A}$ makes quantum queries to H, then

$$\mathbf{Adv}^{\mathsf{sNR}^{H,\perp}}_{\mathsf{BUFF}[\mathcal{S},H]}(\mathcal{D},\mathcal{A},\mathsf{aux}) \leq 2(q_A+q_S)\cdot\sqrt{\mathbf{Adv}^{\mathsf{HnS}^H}(\bar{\mathcal{D}},\bar{\mathcal{A}})} + q_K\epsilon + \frac{q_D+1}{|\mathcal{Y}|}\ , \tag{5}$$

holds in place of (4)*, and $\bar{\mathcal{A}}$ then makes quantum queries as well.*

Furthermore, in the computational setting when considering a non-fixed security parameter and PPT algorithms $\mathcal{D}$ and $\mathcal{A}$, then $\bar{\mathcal{D}}$ and $\bar{\mathcal{A}}$ are PPT as well.

The intuition behind the proof is as follows. Consider the $\mathsf{sNR}^{H,\perp}$ game. Due to the assumed hardness of Hide-and-Seek, $\mathcal{A}$ cannot recover m from its input and thus makes no query to H that has m as suffix. But then it cannot gather any information on $H(\mathsf{pk}',m)$ for any pk', and thus it will not be able to output $y'=H(\mathsf{pk}',m)$ for any pk'. Formally, we have to make sure that $\mathcal{A}$ gets no information on y' via its input, which is controlled by KGen and $\mathcal{D}$, which may query H on $H(\mathsf{pk}',m)$ for any pk'. This is taken care of in our formal proof below.

Proof. In Fig. 3 we define a hybrid sequence reducing the $\mathsf{sNR}^{H,\perp}$ property of $\mathsf{BUFF}[\mathcal{S},H]$ to the HnS^H property of H. To start with, we note that the adversary $(\mathcal{D},\mathcal{B})$ playing G_0 is identical to $(\mathcal{D},\mathcal{A})$ playing $\mathsf{sNR}^{H,\perp}_{\mathsf{BUFF}[\mathcal{S},H]}$.

The G_0 to G_1 hop. The only difference between G_0 and G_1 is whether $\mathcal{B}$ is given oracle access to the original random oracle H, or the reprogrammed oracle $H[(\cdot,m)\mapsto\perp]$ and replies with $\perp$ to any query that has suffix m.

Consider the hider $\bar{\mathcal{D}}$ and seeker $\bar{\mathcal{A}}$, where $\bar{\mathcal{D}}$ samples $(\mathsf{sk},\mathsf{pk})\leftarrow\mathsf{KGen}^H$ and $m\leftarrow\mathcal{D}^H(\mathsf{sk})$ and returns

$$x\coloneqq(\mathsf{pk},m)\qquad\text{and}\qquad z\coloneqq(\mathsf{sk},\mathsf{aux}(\mathsf{sk},m))\,,$$

and on input $H(x)=H(\mathsf{pk},m)$ and z, the seeker $\bar{\mathcal{A}}$ samples a random index $i\leftarrow[q_A+q_S]$, runs

$$\mathcal{B}^H(\mathsf{sk},H(\mathsf{pk},m),\mathsf{aux}(\mathsf{sk},m)) = \mathcal{A}^H\big(\mathsf{sk},\big(H(\mathsf{pk},m),\mathsf{Sign}^H(\mathsf{sk},y)\big),\mathsf{aux}(\mathsf{sk},m)\big)$$

internally, but then looks at / does a full measurement of the i-th query to obtain (pk_i^*,m_i^*), and returns (pk,m_i^*). It is clear by construction that $z\in\mathcal{SK}\times\mathcal{AUX}$, and (3) immediately follows from the fact that pk can be derived from sk, and so

$$\mathsf{H}_\infty(x\mid H,z)=\mathsf{H}_\infty(\mathsf{pk},m\mid H,\mathsf{sk},\mathsf{aux}(\mathsf{sk},m))=\mathsf{H}_\infty(m\mid H,\mathsf{sk},\mathsf{aux}(\mathsf{sk},m)) \tag{6}$$

as claimed. It also follows from construction that $\bar{\mathcal{D}}$ and $\bar{\mathcal{A}}$ preserve the efficiency of $\mathcal{D}$ and $\mathcal{A}$. In terms of query complexity, $\bar{\mathcal{A}}$ makes at most q_A+q_S queries to H.

In the case where $\mathcal{A}$ makes classical queries, there is no difference in the two games when $\mathcal{B}$ makes no query to a point where the two oracles differ, and thus

$$\begin{aligned}\Pr[1\leftarrow\mathsf{G}_0] &\leq \Pr[1\leftarrow\mathsf{G}_1] + \Pr\left[\exists\, i\in[q_A+q_S]\text{ s.t. }m_i^*=m\right]\\ &\leq \Pr[1\leftarrow\mathsf{G}_1] + q_A\cdot\mathbf{Adv}^{\mathsf{HnS}^H}(\bar{\mathcal{D}},\bar{\mathcal{A}})\ .\end{aligned}$$

In the quantum case, the same kind of guarantee follows from the O2H lemma [AHU19, Theorem 3], which gives us that

$$\begin{aligned}\Pr[1\leftarrow\mathsf{G}_0] &\leq \Pr[1\leftarrow\mathsf{G}_1] + 2(q_A+q_S)\cdot\sqrt{\Pr[m_i^*=m]}\\ &\leq \Pr[1\leftarrow\mathsf{G}_1] + 2(q_A+q_S)\cdot\sqrt{\mathbf{Adv}^{\mathsf{HnS}^H}(\bar{\mathcal{D}},\bar{\mathcal{A}})}\ .\end{aligned}$$

G_0:
1: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}^H$; $m \leftarrow \mathcal{D}^H(\mathsf{sk})$
2: $(\mathsf{pk}', y') \leftarrow \mathcal{B}^H(\mathsf{sk}, y, \mathsf{aux}(\mathsf{sk}, m))$
3: **return** $H(\mathsf{pk}', m) = y' \ \wedge \ \mathsf{pk}' \neq \mathsf{pk}$

G_1:
1: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}^H$; $m \leftarrow \mathcal{D}^H(\mathsf{sk})$
2: $(\mathsf{pk}', y') \leftarrow \mathcal{B}^{H[(\cdot, m) \mapsto \perp]}(\mathsf{sk}, H(\mathsf{pk}, m), \mathsf{aux}(\mathsf{sk}, m))$
3: **return** $H(\mathsf{pk}', m) = y' \ \wedge \ \mathsf{pk}' \neq \mathsf{pk}$

G_2:
1: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}^H$; $m \leftarrow \mathcal{D}^H(\mathsf{sk})$
2: **abort** if KGen queried $H(\cdot, m)$
3: $(\mathsf{pk}', y') \leftarrow \mathcal{B}^{H[(\cdot, m) \mapsto \perp]}(\mathsf{sk}, H(\mathsf{pk}, m), \mathsf{aux}(\mathsf{sk}, m))$
4: **return** $H(\mathsf{pk}', m) = y' \ \wedge \ \mathsf{pk}' \neq \mathsf{pk}$

G_3^i:
1: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}^H$; $m \leftarrow \mathcal{D}^H(\mathsf{sk})$
2: **abort** if KGen queried $H(\cdot, m)$
3: $(\mathsf{pk}', y') \leftarrow \mathcal{B}^{H[(\cdot, m) \mapsto \perp]}(\mathsf{sk}, H(\mathsf{pk}, m), \mathsf{aux}(\mathsf{sk}, m))$
4: **return** $H(\mathsf{pk}', m) = y' \ \wedge \ \mathsf{pk}' \neq \mathsf{pk} \ \wedge (\mathsf{pk}', m) = (\mathsf{pk}_i, m_i)$
5: {where (pk_i, m_i) is $\mathcal{D}$'s ith query.}

G_4^i:
1: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}^H$; $m \leftarrow \mathcal{D}^H(\mathsf{sk})$
2: **abort** if KGen queried $H(\cdot, m_i)$
3: $(\mathsf{pk}', y') \leftarrow \mathcal{B}^{H[(\cdot, m_i) \mapsto \perp]}(\mathsf{sk}, H(\mathsf{pk}, m_i), \mathsf{aux}(\mathsf{sk}, m_i))$
4: **return** $H(\mathsf{pk}_i, m_i) = y' \ \wedge \ \mathsf{pk}_i \neq \mathsf{pk}$
5: {where (pk_i, m_i) is $\mathcal{D}$'s ith query.}

$\mathcal{B}^H(\mathsf{sk}, y, a)$:
1: $\sigma \leftarrow \mathsf{Sign}^H(\mathsf{sk}, y)$
2: **return** $(\mathsf{pk}', y') \leftarrow \mathcal{A}^H(\mathsf{sk}, (\sigma, y), a)$

Fig. 3. Hybrid steps reducing $\mathsf{sNR}^{H,\perp}_{\mathsf{BUFF}[\mathcal{S},H]}$ to HnS^H when $\mathcal{D}$ is classical, i.e., (5), (4). In the derivations below we drop the parameter k for notational convenience.

The G_1 to G_2 hop. The difference between G_1 and G_2 is that the latter aborts if KGen^H ever makes a query of the form $(\cdot, m)$. Given that m is produced given (H, sk) but independent of KGen's q_K queries conditioned on (H, sk), and m satisfies (1), we have

$$\Pr[1 \leftarrow \mathsf{G}_1] \leq \Pr[1 \leftarrow \mathsf{G}_2] + \Pr[\mathsf{G}_2 \text{ abort}] \leq \Pr[1 \leftarrow \mathsf{G}_2] + q_K \epsilon \ .$$

The G_2 to G_3^i hop. Assume without loss of generality that $\mathcal{D}$ never repeats its queries $(k_1, m_1), \ldots, (k_{q_D}, m_{q_D})$. Note that the queries of $\mathcal{A}$ in G_1 are blocked at $(\cdot, m)$, and the game aborts if KGen ever queries $(\cdot, m)$. Since, conditioned on KGen not querying with $(\cdot, m)$, $k' \neq k$ and $(k', m) \neq (k_i, m_i)$ for all i, the output $H(k', m)$ is uniformly random and independent of $\mathcal{A}$'s input $(\mathsf{sk}, H(\mathsf{pk}, m), \mathsf{aux}(\mathsf{sk}, m))$ together with the oracle $H[(\cdot, m) \mapsto \perp]$ it has access to, we have

$$\begin{aligned}\Pr[1 \leftarrow \mathsf{G}_2] &\leq \Pr\left[\begin{array}{c}\exists i \in [q_D] \text{ s.t. } (k', m) = (k_i, m_i) \\ 1 \leftarrow \mathsf{G}_2\end{array}\right] + \Pr\left[1 \leftarrow \mathsf{G}_2 \;\middle|\; \begin{array}{c}\mathsf{KGen} \text{ not querying } (\cdot, m) \\ (k', m) \neq (k_i, m_i)\ \forall i \in [q_D] \\ k' \neq k\end{array}\right] \\ &\leq \sum_{i \in [q_D]} \Pr\left[1 \leftarrow \mathsf{G}_3^i\right] + 1/|\mathcal{Y}| \ .\end{aligned}$$

The G_3^i to G_4^i hop. Because of the extra condition $(\mathsf{pk}', m') = (\mathsf{pk}_i, m_i)$ in G_3^i, replacing pk' with pk_i and m' with m_i as in G_4^i, does not change the winning probability. We further drop the condition $(\mathsf{pk}', m') = (\mathsf{pk}_i, m_i)$,

which does not decrease the winning probability.

Finally, it remains to upper bound the winning probability of G_4^i for each $i \in [q_D]$. By a lazy sampling argument, we note that conditioned on KGen not querying with $(\cdot, m_i)$ and $\mathsf{pk}_i \neq \mathsf{pk}$, the output $H(k_i, m_i)$ is uniform and independent of $(H[(\cdot, m_i) \mapsto \bot], k, H(k, m_i), \mathsf{aux}(m_i))$, and hence, y' generated by $\mathcal{A}^{H[(\cdot,m_i)\mapsto\bot]}(k, H(k, m_i), \mathsf{aux}(m_i))$ is equal to $H(k_i, m_i)$ with probability at most $1/|\mathcal{Y}|$, i.e.

$$\Pr\left[1 \leftarrow \mathsf{G}_4^i\right] \leq 1/|\mathcal{Y}| \,,$$

which concludes (4), (5). □

Remark 1. We point out that the claim on $\bar{\mathcal{D}}$ and $\bar{\mathcal{A}}$ be PPT if $\mathcal{D}$ and $\mathcal{A}$ are, fails to hold when aiming for a variant of Lemma 1 that considers $\mathsf{NR}^{H,\bot}$ instead of $\mathsf{sNR}^{H,\bot}$. The reason is that, on input $H(\mathsf{pk}, m)$ and z, the seeker $\bar{\mathcal{A}}$ needs to run $\mathcal{A}$ on *a signature of* $H(\mathsf{pk}, m)$, which it can do efficiently if given sk (which is part of z here, exploiting that $\mathcal{D}$ is given sk), but not if only given pk. This is the reason why in the *computational* setting, treated in Sect. 5, our proof for showing that BUFF satisfies $\mathsf{sNR}^{H,\bot}$ does not carry over to $\mathsf{NR}^{H,\bot}$ (in line with the counter example given in [DFHS24]).

3.3 Main result

By means of the above reduction to HnS and the analysis of HnS in the upcoming section, we obtain the following main result on the non-resignability of the BUFF transform $\mathsf{BUFF}[\mathcal{S}, H]$ of any signature scheme $\mathcal{S} = (\mathsf{KGen}^H, \mathsf{Sign}^H, \mathsf{Vrfy}^H)$. In the theorem statement, the parameters q_K and q_S refer to (an upper bound on) the number of queries to H that KGen^H and Sign^H perform. The theorem is obtained via plugging in Theorem 2 and 3 into Lemma 1 with some simplification to the obtained upperbounds. For completeness, we spell out its proof in Appendix A.

Theorem 1. *Let $\mathcal{D}^H : \mathcal{SK} \to \mathcal{M}$ and $\mathcal{A}^H : \mathcal{SK} \times \mathcal{SGN} \times \mathcal{AUX} \to \mathcal{PK} \times \mathcal{SGN}$ be $\mathsf{sNR}^{H,\bot}$-adversaries against $\mathsf{BUFF}[\mathcal{S}, H]$ for some $\mathsf{aux} : \mathcal{SK} \times \mathcal{M} \to \mathcal{AUX}$, making at most q_D and q_A queries to H, respectively, where (1) is satisfied for $0 < \epsilon \leq \frac{1}{2}$. Then*

$$\mathbf{Adv}^{\mathsf{sNR}^{H,\bot}}_{\mathsf{BUFF}[\mathcal{S},H]}(\mathcal{D}, \mathcal{A}, \mathsf{aux}) \leq 8(q_A + q_S + 1)^2 \log\left(\frac{|\mathcal{SK}| \cdot |\mathcal{AUX}|}{\epsilon}\right)\epsilon + q_K\epsilon + \frac{q_D + 1}{|\mathcal{Y}|} \,,$$

and in the case $\mathcal{A}$ makes quantum queries to H, then

$$\mathbf{Adv}^{\mathsf{sNR}^{H,\bot}}_{\mathsf{BUFF}[\mathcal{S},H]}(\mathcal{D}, \mathcal{A}, \mathsf{aux}) \leq O\left(\sqrt{\left(\log\frac{|\mathcal{SK}| \cdot |\mathcal{AUX}|}{\epsilon} + q_A + q_S\right)(q_A + q_S)^3\epsilon}\right) + q_K\epsilon + \frac{q_D + 1}{|\mathcal{Y}|} \,,$$

where the asymptotic bound holds as $\min(1/\epsilon, q_A) \to \infty$, and the constants are absolute constants.

Remark 2. In the case where $\mathcal{D}$ makes quantum queries to H, we expect a similar argument as in the proof of [DFHS24, Theorem 15] applies.

4 Analyzing Hide-and-Seek

As explained above, the technical core of proving the non-resignability property of the BUFF transform consists of analyzing the Hide-and-Seek game. Concretely, our goal is to show that the probability

$$\Pr\left[\, x = \mathcal{A}^H(H(x), z) \,\right]$$

is small, for any query-*un*bounded algorithm $\mathcal{D}^H$ that produces a pair (x, z) such that $\mathsf{guess}(x \mid H, z) \leq \epsilon$ holds, and for any query-bounded algorithm $\mathcal{A}$.

Below in Sect. 4.2, we first consider the case of an $\mathcal{A}$ that makes classical queries to the random oracle H; later we also consider the case of quantum queries, which introduces additional challenges. We note that since $\mathcal{D}$ has unbounded query complexity, it is irrelevant whether those are classical or quantum; $\mathcal{D}$ may inspect the entire function table anyway.[9] We also emphasize that we do not restrict the computational complexity of $\mathcal{D}$ or $\mathcal{A}$.

Before jumping into the analysis though, we discuss the game a bit further, and in particular we look into the simple(r) variant where x is uniformly random and independent of H, and z is fixed.

4.1 Special Case: Uniform Challenges

What makes the game challenging to analyze is that the challenge x (and the hint z) may be arbitrarily correlated with H, as long as $\mathsf{guess}(x \mid H, z) \leq \epsilon$. For instance, given a function $H : \mathcal{X} \to \mathcal{X}$, the hider $\mathcal{D}$ can pick a challenge x that satisfies $H(x) = x$, and the seeker $\mathcal{A}$ can simply output $H(x)$. Although this is not a valid attack under the condition $\mathsf{guess}(x \mid H, z) \leq \epsilon$, because a random function $H : \mathcal{X} \to \mathcal{X}$ typically does not have many fixed points, this example suggests that one cannot argue that $H(x)$ reveals no information about x.

In the special case where x is uniform and independent of (H, z) and z is fixed, it is straightforward to show that any $\mathcal{A}$ making at most q classical queries to the random oracle H satisfies

$$\Pr\big[x = \mathcal{A}^H(H(x), z)\big] \leq \frac{(q+1)}{|\mathcal{X}|}\,.$$

In addition, even if the hint z can depend on H, tight bounds are known in the literature: the probability that a q-query seeker $\mathcal{A}$ succeeds is in the order of at most $q \log |\mathcal{Z}|/|\mathcal{X}|$ if $\mathcal{A}$ is classical [DGK17,CDGS18], or in the order of at most $q(q + \log |\mathcal{Z}|)/|\mathcal{X}|$ if $\mathcal{A}$ can make quantum queries [CGLQ20].

However, in the general case, where the only guarantee about x is that $\mathsf{guess}(x \mid H, z) \leq \epsilon$ for some $\epsilon < 1$, the strong bounds above do not apply. Nevertheless, in the remaining of this section, we will show how to reduce the tricky general case to the uniform-challenge case.

Inspired by [CGLQ20], we will actually reduce the general case to the "multi-instance" case with uniform challenges and an independent hint. In particular, consider challenges $x_1^u, \ldots, x_k^u$ that are sampled uniformly and independently from $\mathcal{X}$, and a fixed hint $z^\circ \in \mathcal{Z}$ that does not depend on $x_1^u, \ldots, x_k^u$ and H. Then for any seeker that attempts to solve all k challenges with the hint z°, it is not hard to prove the following lemma. For completeness we give the proof in Appendix B.

Lemma 2. *For every oracle algorithm $\mathcal{A}^H : \mathcal{Y} \times \mathcal{Z} \to \mathcal{X}$ that makes at most q classical queries to H,*

$$\Pr\left[\forall i \in [k] \colon x_i^u = \mathcal{A}^H(H(x_i^u), z^\circ)\right] \leq k! \frac{(q+1)^k}{|\mathcal{X}|^k}\,,$$

where $\mathcal{A}$ is independently re-executed for each i.

The case where $\mathcal{A}$ can make quantum queries to H is more involved, but has been studied in [CGLQ20].

Lemma 3 (Corollary of [CGLQ20, Lemma 5.2]). *For every oracle algorithm $\mathcal{A}^H : \mathcal{Y} \times \mathcal{Z} \to \mathcal{X}$ that makes at most q quantum queries to H,*

$$\Pr\left[\forall i \in [k] \colon x_i^u = \mathcal{A}^H(H(x_i^u), z^\circ)\right] \leq O\left(\frac{kq + q^2}{|\mathcal{X}|}\right)^k \text{ as } \min(k, q, |\mathcal{X}|) \to \infty\,,$$

where $\mathcal{A}$ is independently re-executed for each i, and the constants in the asymptotic bound are absolute constants.

[9] For the purpose of proving Thm. 1, it would be sufficient to restrict the seeker $\mathcal{D}$ to be query bounded as well; however, interestingly, we need the result for a query unbounded $\mathcal{D}$ for the computational case (see Sect. 5 and Remark 5).

4.2 The Classical Case

The following provides a bound on the Hide-and-Seek property of the random oracle for a *classical* seeker $\mathcal{A}$.

Theorem 2 (The RO satisfies HnS^H, classically). *Let $\mathcal{D} : \{\bot\} \to \mathcal{X} \times \mathcal{Z}$ and $\mathcal{A} : \mathcal{Y} \times \mathcal{Z} \to \mathcal{X}$ be HnS^H-adversaries satisfying (2) for some $0 < \epsilon < 1$, where $\mathcal{A}$ makes q classical queries to H. Then we have*

$$\mathbf{Adv}^{\mathsf{HnS}^H}(\mathcal{D}, \mathcal{A}) \leq 2(q+1)\big(\log|\mathcal{Z}| + \log(1/\epsilon) + 1\big)\epsilon + \epsilon\,.$$

Our strategy is to turn a successful HnS^H seeker $\mathcal{A}$ into a similarly successful guesser $\mathcal{G}$ that tries to guess x from H and z. Since such a successful guesser cannot exist by (2), no successful $\mathcal{A}$ can exist.

Proof. Given that $\mathcal{A}$ is classical here, we may assume it to be deterministic. For any fixed choices H° and z°, we can thus define the set

$$S(H^\circ, z^\circ) \coloneqq \{x^\circ \in \mathcal{X} \,|\, \mathcal{A}^{H^\circ}(H^\circ(x^\circ), z^\circ) = x^\circ\}$$

of all x° on which $\mathcal{A}$ succeeds.

Following the above strategy for proving the claimed statement, we consider the following guesser $\mathcal{G}$. On input H and z, it samples and outputs a uniformly random $\hat{x} \in S(H, z)$ as guess for x (with the convention that $\hat{x} = \bot$ in case S is empty). We can then lower bound the success probability of $\mathcal{G}$ as follows, for any positive $T \in \mathbb{Z}$.

$$\begin{aligned}
\Pr[\hat{x} = x] &\geq \Pr[\hat{x} = x \wedge |S| \leq T] \\
&\geq \frac{1}{T}\Pr[\mathcal{A}^H(H(x), z) = x \wedge |S| \leq T] \\
&\geq \frac{1}{T}\big(\Pr[\mathcal{A}^H(H(x), z) = x] - \Pr[|S| > T]\big)\,,
\end{aligned}$$

where for the second inequality we exploit that for any fixed choices of H, x and z, if $|S| \leq T$ then $\Pr[\hat{x} = x] = 1/T$ if $x \in S$, i.e., if $\mathcal{A}^H(H(x), z) = x$, and 0 otherwise, and so the inequality is obtained by averaging over the choices of H, x, and z. The last inequality is by union bound. Rearranging the terms, we thus have

$$\mathbf{Adv}^{\mathsf{HnS}^H}(\mathcal{D}, \mathcal{A}) \leq T \cdot \Pr[\hat{x} = x] + \Pr[|S| > T] \leq T\epsilon + \Pr[|S| > T]\,. \tag{7}$$

In order to control $\Pr[|S| > T]$, we introduce

$$\sigma(H^\circ, z^\circ) \coloneqq \Pr\Big[x^u = \mathcal{A}^{H^\circ}(H^\circ(x^u), z^\circ)\Big] = \frac{|S(H^\circ, z^\circ)|}{|\mathcal{X}|} \tag{8}$$

where $x^u \leftarrow \mathcal{X}$, and we observe that, for any positive $k \in \mathbb{Z}$,

$$\sigma(H^\circ, z^\circ)^k = \Pr\Big[x_i^u = \mathcal{A}^{H^\circ}(H^\circ(x_i^u), z^\circ)\ \forall i \in [k]\Big]$$

where $x_1^u, \ldots, x_k^u \leftarrow \mathcal{X}$. What we are actually interested in is the average over the choice of H and z. Towards this end, we note that

$$\begin{aligned}
\mathbb{E}[\sigma(H, z)^k] &= \Pr\Big[x_i^u = \mathcal{A}^H(H(x_i^u), z)\ \forall i \in [k]\Big] \\
&= \sum_{z^\circ} \Pr\Big[z = z^\circ \wedge x_i^u = \mathcal{A}^H(H(x_i^u), z^\circ)\ \forall i \in [k]\Big] \\
&\leq \sum_{z^\circ} \Pr\Big[x_i^u = \mathcal{A}^H(H(x_i^u), z^\circ)\ \forall i \in [k]\Big] \qquad (9) \\
&\leq |\mathcal{Z}| \cdot \frac{k!(q+1)^k}{|\mathcal{X}|^k}\,,
\end{aligned}$$

where the last inequality is by Lemma 2. Thus

$$\mathbb{E}[|S(H,z)|^k] = |\mathcal{X}|^k \cdot \mathbb{E}[\sigma(H,z)^k] \le |\mathcal{Z}| \cdot k!(q+1)^k\,,$$

and so by Markov's inequality,

$$\Pr\big[|S(H,z)| > 2k(q+1)\big] \le \frac{\mathbb{E}\big[|S(H,z)|^k\big]}{\big(2k(q+1)\big)^k} \le \frac{|\mathcal{Z}|}{2^k} \le \epsilon$$

where the final inequality is achieved by choosing $k = \big\lceil \log|\mathcal{Z}| + \log(1/\epsilon)\big\rceil$. Thus, setting $T = 2k(q+1)$ and plugging into (7) we obtain that

$$\begin{aligned}\mathbf{Adv}^{\mathsf{HnS}^H}(\mathcal{D},\mathcal{A}) &\le T\epsilon + \Pr[|S| > T] \\ &\le 2(q+1)\big(\log|\mathcal{Z}| + \log(1/\epsilon) + 1\big)\epsilon + \epsilon\,.\end{aligned}$$

This proves the claim. □

4.3 A Bound for the Quantum Case

The following provides a bound on the Hide-and-Seek property of the random oracle for a *quantum* seeker $\mathcal{A}$.

Theorem 3 (The RO satisfies HnS^H, quantumly). *Let $\mathcal{D} : \{\bot\} \to \mathcal{X} \times \mathcal{Z}$ and $\mathcal{A} : \mathcal{Y} \times \mathcal{Z} \to \mathcal{X}$ be HnS^H-adversaries satisfying (2) for some $0 < \epsilon < 1$, where $\mathcal{A}$ makes q quantum queries to H. Then we have*

$$\mathbf{Adv}^{\mathsf{HnS}^H}(\mathcal{D},\mathcal{A}) \le O\big((\log|\mathcal{Z}| + \log(1/\epsilon) + q)q\epsilon\big)$$

as $\min(1/\epsilon, |\mathcal{Z}|, q) \to \infty$, where the constants in the asymptotic bound are absolute constants.

The proof here follows very closely the proof for the classical case, except that we use Lemma 3 to bound the multi-instance game for a quantum algorithm. Furthermore, some additional changes are needed since we cannot assume $\mathcal{A}$ to be deterministic anymore.

Proof. Here, for any H° and z°, we define the following "weighted set"

$$S^*(H^\circ, z^\circ) \coloneqq \big\{\big(x^\circ, w_{H^\circ,z^\circ}(x^\circ)\big) \,\big|\, x^\circ \in \mathcal{X}\big\}\,,$$

where each element x° comes with a weight, given by

$$w_{H^\circ,z^\circ}(x^\circ) := \Pr\big[x^\circ = \mathcal{A}^{H^\circ}(H^\circ(x^\circ), z^\circ)\big]\,.$$

The total weight of $S^*(H^\circ, z^\circ)$ is defined as $W(S^*(H^\circ, z^\circ)) := \sum_{x^\circ} w_{H^\circ,z^\circ}(x^\circ)$.

Here, we consider the guesser $\mathcal{G}$ that, on input H and z, chooses its guess $\hat{x}$ by picking it from $\mathcal{X}$ according to the renormalized weights, i.e., according to the distribution

$$p_{H,z}(\hat{x}) := \frac{w_{H,z}(\hat{x})}{W(S^*(H,z))}\,.$$

We observe that this generalizes the approach in the previous section where $\mathcal{A}$ may assumed to be deterministic. All weights are then 0 or 1, giving rise to the set S in the proof of Theorem 2 when keeping only the elements with weight 1, and the total weight of S^* then matches up with $|S|$, and $\hat{x}$ is then uniformly random in S.

We proceed by following that approach, with obvious changes. Namely, first we note that

$$\begin{aligned}\Pr[\hat{x} = x] &\ge \Pr[\hat{x} = x \,\wedge\, W(S^*(H,z)) \le T] \\ &\ge \frac{1}{T}\Pr[\mathcal{A}^H(H(x),z) = x \,\wedge\, W(S^*(H,z)) \le T] \\ &\ge \frac{1}{T}\big(\Pr[\mathcal{A}^H(H(x),z) = x] - \Pr[W(S^*(H,z)) > T]\big)\,,\end{aligned}$$

where here, for the second inequality, we exploit that for any fixed choices of H, x and z, if $W(S^*(H,z)) \le T$ then $\Pr[\hat{x} = x] = p_{H,z}(x) \ge w_{H,z}(x)/T$, and so the inequality is obtained by averaging over these choices. Rearranging the terms, we have

$$\mathbf{Adv}^{\mathsf{HnS}^H}(\mathcal{D}, \mathcal{A}) \le T\epsilon + \Pr[W(S^*(H,z)) > T]\,. \tag{10}$$

In order to control $\Pr[W(S^*(H,z)) > T]$, we introduce

$$\sigma(H^\circ, z^\circ) \coloneqq \Pr\Big[x^u = \mathcal{A}^{H^\circ}(H^\circ(x^u), z^\circ)\Big] = \frac{W(S^*(H^\circ, z^\circ))}{|\mathcal{X}|} \tag{11}$$

where $x^u \leftarrow \mathcal{X}$. Recycling the line of reasoning in the previous section, we observe that, for any positive $k \in \mathbb{Z}$,

$$\sigma(H^\circ, z^\circ)^k = \Pr\Big[x_i^u = \mathcal{A}^{H^\circ}(H^\circ(x_i^u), z^\circ)\ \forall i \in [k]\Big]$$

where $x_1^u, \ldots, x_k^u \leftarrow \mathcal{X}$, and that

$$\begin{aligned}
\mathbb{E}[\sigma(H,z)^k] &= \Pr\Big[x_i^u = \mathcal{A}^H(H(x_i^u), z)\ \forall i \in [k]\Big] \\
&= \sum_{z^\circ} \Pr\Big[z = z^\circ \ \wedge\ x_i^u = \mathcal{A}^H(H(x_i^u), z^\circ)\ \forall i \in [k]\Big] \\
&\le \sum_{z^\circ} \Pr\Big[x_i^u = \mathcal{A}^H(H(x_i^u), z^\circ)\ \forall i \in [k]\Big] \\
&\le |\mathcal{Z}| \cdot C^k \frac{(k+q)^k q^k}{|\mathcal{X}|^k}\,,
\end{aligned} \tag{12}$$

for some absolute constant C, and $k, q, |\mathcal{X}|$ large enough, where the last inequality is now by Lemma 3, given that $\mathcal{A}$ is quantum. Thus

$$\mathbb{E}[W(S^*(H,z))^k] = |\mathcal{X}|^k \cdot \mathbb{E}[\sigma(H,z)^k] \le |\mathcal{Z}| \cdot C^k (k+q)^k q^k\,,$$

and so by Markov inequality,

$$\Pr\big[W(S^*(H,z)) > 2C(k+q)q\big] \le \frac{\mathbb{E}\big[W(S^*(H,z))^k\big]}{\big(2C(k+q)q\big)^k} \le \frac{|\mathcal{Z}|}{2^k} \le \epsilon$$

where the final inequality is achieved by choosing the minimum possible $k \ge \log|\mathcal{Z}| + \log(1/\epsilon)$. Thus, setting $T = 2C(k+q)q$ and plugging into (10) we obtain that

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{HnS}^H}(\mathcal{D}, \mathcal{A}) &\le T\epsilon + \Pr[W(S^*(H,z)) > T] \\
&\le O\big((\log|\mathcal{Z}| + \log(1/\epsilon) + q)q\epsilon\big)\,.
\end{aligned}$$

This proves the claim. □

5 Non-Resignability in the Computational Setting

Here, we want to extend our result on non-resignability of the BUFF transform to the computational setting, where $\mathcal{D}, \mathcal{A}$ and aux are polynomially bounded, and where the entropy requirement (1) holds computationally only; the latter is the reason why the computational case does not follow directly from the statistical case. In order to capture the entropy requirement (1) in the computational setting via HILL entropy, we need the notion of the HILL entropy *in the ROM*, as introduced in [DFHS24], which we briefly recall below.

Here and for the remainder of this section, we take it as understood that the domain and co-domain of $H : \mathcal{X} \to \mathcal{Y}$ may depend on the security parameter λ; for simplicity, we leave this dependency implicit. Moreover, we assume the co-domain of H to be super-polynomially large, i.e., $|\mathcal{Y}| \ge \lambda^{\omega(1)}$. For simplicity, we restrict to asymptotic bounds below.

5.1 HILL Entropy in the ROM

The HILL entropy [HILL99,HLR07] is introduced as a computational analogue of min-entropy. For a pair of random variables (X, Y), we say that X has high conditional HILL entropy given Y, if there is another random variable Z, such that (X, Y) and (Z, Y) are computationally indistinguishable, and yet Z has high min-entropy given Y.

However, expressing (1) naively using HILL entropy is problematic, since H, which is conditioned on, is too large for a computationally bounded distinguisher to even read. Because of this reason, [DFHS24] introduced the notion of HILL entropy *in the ROM*, where instead of conditioning on H, the distinguisher (that tries to distinguish (X, Y) and (Z, Y)) is given bounded oracle access to H. We recall (the asymptotic version of) the formal definition.

Definition 1. *Let (X_λ, Y_λ) be a pair of (possibly H-dependent) random variables for each λ. We say that $X = \{X_\lambda\}_\lambda$ has $k(\lambda)$ bits of* conditional HILL entropy *given $Y = \{Y_\lambda\}_\lambda$* in the ROM*, denoted by*

$$\mathsf{HILL}^{H}_{\infty}(X \mid Y) \geq k(\lambda)\,,$$

if for every λ there exists a random variable Z_λ with $\mathsf{H}_\infty(Z_\lambda \mid Y_\lambda, H) \geq k(\lambda)$, and so that $\{(X_\lambda, Y_\lambda)\}_\lambda$ and $\{(Z_\lambda, Y_\lambda)\}_\lambda$ are computationally indistinguishable for oracle algorithms.

Remark 3. Following the standard definition, computationally indistinguishability holds for *non-uniform* PPT distinguishers; this then allows us to consider *non-uniform* PPT (oracle) algorithms $\mathcal{D}$ and $\mathcal{A}$ below. If instead we consider computationally indistinguishability for *uniform* PPT distinguishers only then below $\mathcal{D}$ and $\mathcal{A}$ need to be restricted to *uniform* PPT algorithms as well. Similarly, if we allow the distinguisher to be quantum, then $\mathcal{D}$ and $\mathcal{A}$ below may be quantum as well.

5.2 Achieving $\mathsf{sNR}^{H,\perp}$ in the Computational Setting

Here, we consider the computational variant of $\mathsf{sNR}^{H,\perp}$, where we restrict $\mathcal{D}^H, \mathcal{A}^H$ and aux to be PPT (oracle) algorithms. Furthermore, the entropy requirement (1) is replaced by

$$\mathsf{HILL}^{H}_{\infty}\left(m \mid \mathsf{sk}, \mathsf{aux}(\mathsf{sk}, m)\right) \geq \omega(\log \lambda)\,, \tag{13}$$
$$(\mathsf{sk}, pk) \leftarrow \mathsf{KGen}^H,\ m \leftarrow \mathcal{D}^H(\mathsf{sk})$$

and we then naturally demand that the game $\mathsf{sNR}^{H,\perp}$ can be won with negligible probability $\mathsf{negl}(\lambda)$ only.

Remark 4. Interestingly, and maybe somewhat surprisingly, in the computational setting $\mathsf{sNR}^{H,\perp}$ does not imply $\mathsf{NR}^{H,\perp}$, in contrast to the statistical setting, as explained in Sect. 2.3. Indeed, [DFHS24] showed that the BUFF transform does in general not satisfy $\mathsf{NR}^{H,\perp}$ in the computational setting, while below we show that it does satisfy $\mathsf{sNR}^{H,\perp}$. See Remark 1 for why our proof does not carry over to $\mathsf{NR}^{H,\perp}$. We suspect that the two notions are incomparable in the computational setting.

We get the following positive result on the computational $\mathsf{sNR}^{H,\perp}$ security of the BUFF transform $\mathsf{BUFF}[\mathcal{S}, H]$.

Theorem 4. *Let $\mathcal{S} = (\mathsf{KGen}, \mathsf{Sign}^H, \mathsf{Vrfy}^H)$ be a signature scheme in ROM, where KGen makes no query to H, and let $\mathsf{BUFF}[\mathcal{S}, H]$ be the signature scheme obtained by applying the BUFF transform. Then for every PPT hint function aux, and for any PPT adversaries $\mathcal{D}^H, \mathcal{A}^H$ against $\mathsf{sNR}^{H,\perp}_{\mathsf{BUFF}[\mathcal{S},H]}$ that satisfy (13), we have*

$$\mathbf{Adv}^{\mathsf{sNR}^{H,\perp}}_{\mathsf{BUFF}[\mathcal{S},H]}(\mathcal{D}, \mathcal{A}, \mathsf{aux}) \leq \mathsf{negl}(\lambda)\,.$$

In spirit, we can recycle Lemma 1 to reduce the computational variant of $\mathsf{sNR}^{H,\perp}$ to the computational variant of Hide-and-Seek, and then we show in Lemma 4 that the latter is hard as well, which follows rather directly from the statistical hardness and the definition of the HILL entropy.

Proof. Take $(\bar{\mathcal{D}}, \bar{\mathcal{A}})$ as in Lemma 1, for which

$$\mathbf{Adv}^{\mathsf{sNR}^{H,\perp}}_{\mathsf{BUFF}[\mathcal{S},H]} \leq \mathsf{poly}(\lambda) \cdot \mathbf{Adv}^{\mathsf{HnS}^H}(\bar{\mathcal{D}}, \bar{\mathcal{A}}) + \mathsf{negl}(\lambda) \ ,$$

where we exploit that the numbers of queries made by Sign, $\mathcal{D}$, and $\mathcal{A}$ are bounded by their (polynomial) running times, respectively, and that the additive term $q_K\epsilon$ in (5) vanishes due to the assumption that KGen makes no query to H. Hence it suffices to control the HnS^H advantage of $(\bar{\mathcal{D}}, \bar{\mathcal{A}})$.

Towards this end, we first note that, by inspecting the construction of $\bar{\mathcal{D}}$ with $x = (\mathsf{pk}, m)$ and $z = (\mathsf{sk}, \mathsf{aux}(\mathsf{sk}, m))$, the HILL entropy variant of (3) follows:

$$\mathsf{HILL}^H_\infty_{(x,z)\leftarrow\bar{\mathcal{D}}^H}(x \mid z) \geq k(\lambda) \iff \mathsf{HILL}^H_\infty_{\substack{(\mathsf{sk},pk)\leftarrow\mathsf{KGen}^H \\ m\leftarrow\mathcal{D}^H(\mathsf{sk})}}(m \mid \mathsf{sk}, \mathsf{aux}(\mathsf{sk}, m)) \geq k(\lambda) \ ,$$

where the equivalence is due to the public key pk being *efficiently derivable* from its corresponding secret key sk, and so (6) also holds for the HILL entropy. Combining the above with (13), we obtain

$$\mathsf{HILL}^H_\infty_{(x,z)\leftarrow\bar{\mathcal{D}}^H}(x \mid z) \geq \omega(\log\lambda) \ .$$

Moreover, by Lemma 1, $\bar{\mathcal{D}}\colon \{\perp\} \to \mathcal{X} \times \mathcal{Z}$ with $\mathcal{Z} = \mathcal{SK} \times \mathcal{AUX}$. Hence

$$\log|\mathcal{Z}| = \log|\mathcal{SK}| + \log|\mathcal{AUX}| \leq \mathsf{poly}(\lambda)$$

due to both KGen and aux being poly-time. Finally, Lemma 1 ensures that $\bar{\mathcal{A}}$ is PPT whenever $\mathcal{A}$ is, which is satisfied by assumption. Thus, the assumptions for Lemma 4 below (the hardness of Hide-and-Seek in the computational setting) are all satisfied, and so

$$\mathbf{Adv}^{\mathsf{HnS}^H}(\bar{\mathcal{D}}, \bar{\mathcal{A}}) \leq \mathsf{negl}(\lambda) \ ,$$

which concludes the proof. □

The following provided the computational hardness of Hide-and-Seek.

Lemma 4. *Let $\mathcal{D}^H\colon \{\perp\} \to \mathcal{X} \times \mathcal{Z}$ and $\mathcal{A}^H : \mathcal{Y} \times \mathcal{Z} \to \mathcal{X}$ be adversaries against HnS^H, with $\mathcal{A}$ being PPT, $\log|\mathcal{Z}| < \mathsf{poly}(\lambda)$, and*

$$\mathsf{HILL}^H_\infty_{(x,z)\leftarrow\mathcal{D}^H}(x \mid z) \geq \omega(\log\lambda) \ .$$

Then $\mathbf{Adv}^{\mathsf{HnS}^H}(\mathcal{D}, \mathcal{A}) \leq \mathsf{negl}(\lambda)$.

Proof. Let $(x, z) \leftarrow \mathcal{D}^H$. Via the entropy condition, there is an (H-dependent) random variable $x^* \in \mathcal{X}$ such that $\mathsf{guess}(x^* \mid H, z) \leq \mathsf{negl}(\lambda)$ moreover (x^*, z) and (x, z) are computationally indistinguishable. Without loss of generality, we may assume (x^*, z) is sampled via a (possibly unbounded) hider $\mathcal{D}^{*H}$. Now, inspect the displayed games $\mathsf{HnS}^H(\mathcal{D}, \mathcal{A})$ and $\mathsf{HnS}^H(\mathcal{D}^*, \mathcal{A})$ below.

$\mathsf{HnS}^H(\mathcal{D}, \mathcal{A})$
1: $(x, z) \leftarrow \mathcal{D}^H$
2: **return** $x = \mathcal{A}^H(H(x), z)$

$\mathsf{HnS}^H(\mathcal{D}^*, \mathcal{A})$:
1: $(x^*, z) \leftarrow \mathcal{D}^{*H}$
2: **return** $x^* = \mathcal{A}^H(H(x^*), z)$

By the computational indistinguishability, it follows that

$$|\mathbf{Adv}^{\mathsf{HnS}^H}(\mathcal{D}, \mathcal{A}) - \mathbf{Adv}^{\mathsf{HnS}^H}(\mathcal{D}^*, \mathcal{A})| \leq \mathsf{negl}(\lambda) \ .$$

Finally, we can apply Theorem 3 to the HnS^H adversaries $\mathcal{D}^*$ and $\mathcal{A}$, which satisfy the statistical entropy condition, and so we have $\mathbf{Adv}^{\mathsf{HnS}}(\mathcal{D}^*, \mathcal{A}) \leq \mathsf{negl}(\lambda)$. This concludes the proof. □

Remark 5. Interestingly, towards proving $\mathsf{sNR}^{H,\perp}$ of the BUFF transform in the *statistical* setting, as we did earlier in the paper, it would have been sufficient to show that the random oracle satisfies (the statistical variant of) HnS for a *query bounded* hider $\mathcal{D}$. However, for the above line of reasoning in the computational setting, it is essential that Theorem 2 holds for a query unbounded hider; indeed, above, x^* may be arbitrarily dependent on H, and so might not be producible by a query bounded hider $\mathcal{D}^*$.

6 Conclusion

In the light of recent negative result on the notion of non-resignability in general, and the non-resignability of the BUFF transform in particular, we re-establish the non-resignability property for the original BUFF transform for the (almost) strongest notions of non-resignability that do not contradict any negative result. Our results cover both the statistical and the computational case, and both the classical and the quantum setting. This answers the pressing question left open in the recent works on the non-resignability of the BUFF transform.

One small gap that remains open from our work is to weaken the HILL entropy requirement in the computational setting to *computational unpredictability*, as considered in [DFF24]. Having large HILL entropy implies computational unpredictability, but not the other way round. Thus, whether the BUFF transform satisfies the computational variant of $\mathsf{sNR}^{H,\perp}$ when the HILL entropy requirement is relaxed to computational unpredictability, remains open.

Acknowledgments

Yu-Hsuan Huang is supported by the Dutch Research Agenda (NWA) project HAPKIDO (Project No. NWA.1215.18.002), which is financed by the Dutch Research Council (NWO). Jyun-Jie Liao is supported by Eshan Chattopadhyay's NSF CAREER award 2045576. Patrick Struck acknowledges funding by the Hector Foundation II.

References

ADM+24. Thomas Aulbach, Samed Düzlü, Michael Meyer, Patrick Struck, and Maximiliane Weishäupl. Hash your keys before signing: BUFF security of the additional NIST PQC signatures. In *PQCrypto 2024*, 2024.

AHU19. Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 269–295, Cham, 2019. Springer International Publishing.

Aye15. Andrew Ayer. Duplicate signature key selection attack in let's encrypt. `https://www.agwa.name/blog/post/duplicate_signature_key_selection_attack_in_lets_encrypt`, 2015.

BDF+11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011.

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.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

BWM99. Simon Blake-Wilson and Alfred Menezes. Unknown key-share attacks on the station-to-station (STS) protocol. In Hideki Imai and Yuliang Zheng, editors, *PKC'99*, volume 1560 of *LNCS*, pages 154–170. Springer, Heidelberg, March 1999.

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 1.4.1) of [CDF+21], available at `https://eprint.iacr.org/archive/2020/1525/20231023:114351`.

CDGS18. Sandro Coretti, Yevgeniy Dodis, Siyao Guo, and John P. Steinberger. Random oracles and non-uniformity. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018*, volume 10820 of *Lecture Notes in Computer Science*. Springer, 2018.

CGLQ20. Kai-Min Chung, Siyao Guo, Qipeng Liu, and Luowen Qian. Tight quantum time-space tradeoffs for function inversion. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 673–684, 2020.

DFF24. Samed Düzlü, Rune Fiedler, and Marc Fischlin. BUFFing FALCON without increasing the signature size. Cryptology ePrint Archive, Paper 2024/710, 2024. `https://eprint.iacr.org/2024/710`.

DFHS24. Jelle Don, Serge Fehr, Yu-Hsuan Huang, and Patrick Struck. On the (in)security of the BUFF transform. To appear in CRYPTO 2024, *Lecture Notes in Computer Science*, 2024. Cryptology ePrint Archive version available at `https://eprint.iacr.org/2023/1634`.

DGK17. Yevgeniy Dodis, Siyao Guo, and Jonathan Katz. Fixing cracks in the concrete: Random oracles with auxiliary input, revisited. In *EUROCRYPT 2017*, volume 10211 of *Lecture Notes in Computer Science*, 2017.

HILL99. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

HLR07. Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In Moni Naor, editor, *Advances in Cryptology - EUROCRYPT 2007*, pages 169–186, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

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.

MS04. Alfred Menezes and Nigel Smart. Security of signature schemes in a multi-user setting. In *Designs, Codes and Cryptography*, 2004.

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.

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.

A Proof of Theorem 1

Proof. In the classical case, combining Lemma 1 and Theorem 2, for $\mathcal{Z} = \mathcal{SK} \times \mathcal{AUX}$ and $q = q_A + q_S$ we obtain

$$\begin{aligned}\mathbf{Adv}^{\mathsf{sNR}^{H,\perp}}_{\mathsf{BUFF}[\mathcal{S},H]}(\mathcal{D},\mathcal{A},\mathsf{aux}) &\le q \cdot 4(q+1)(\log|\mathcal{Z}| + \log(1/\epsilon) + 1)\epsilon + q_K\epsilon + \frac{q_D+1}{|\mathcal{Y}|} \\ &\le 8(q+1)^2\big(\log|\mathcal{Z}| + \log(1/\epsilon)\big)\epsilon + q_K\epsilon + \frac{q_D+1}{|\mathcal{Y}|}\,,\end{aligned}$$

where the second inequality exploits that $\log(1/\epsilon) \ge 1$. This concludes the classical bound.

Similarly, in the quantum case, combining Lemma 1 and Theorem 3, for $\mathcal{Z} = \mathcal{SK} \times \mathcal{AUX}$ and $q = q_A + q_S$ we obtain

$$\begin{aligned}&\mathbf{Adv}^{\mathsf{sNR}^{H,\perp}}_{\mathsf{BUFF}[\mathcal{S},H]}(\mathcal{D},\mathcal{A},\mathsf{aux}) \le 2q \cdot \sqrt{O\big((\log|\mathcal{Z}| + \log(1/\epsilon) + q)q\epsilon\big)} + q_K\epsilon + \frac{q_D+1}{|\mathcal{Y}|} \\ &\le O\big(\sqrt{(\log|\mathcal{Z}| + \log(1/\epsilon) + q)q^3\epsilon}\big) + q_K\epsilon + \frac{q_D+1}{|\mathcal{Y}|} \text{ as } \min(1/\epsilon, |\mathcal{Z}|, q) \to \infty,\end{aligned}$$

where the constants in the asymptotic bounds are absolute constants. Hence, there are absolute constants $n, C \geq 2$ such that

$$\mathbf{Adv}^{\mathsf{sNR}^{H,\perp}}_{\mathsf{BUFF}[\mathcal{S},H]}(\mathcal{D},\mathcal{A},\mathsf{aux}) \leq C\sqrt{(\log|\mathcal{Z}|+\log(1/\epsilon)+q)q^3\epsilon} + q_K\epsilon + \frac{q_D+1}{|\mathcal{Y}|}\ ,$$

whenever $\min(1/\epsilon,|\mathcal{Z}|,q) \geq n$. In order to get a bound even when $|\mathcal{Z}| < n$, we increase $|\mathcal{AUX}|$ to $n|\mathcal{AUX}|$ without actually changing the algorithm aux, and so get

$$\begin{aligned}\mathbf{Adv}^{\mathsf{sNR}^{H,\perp}}_{\mathsf{BUFF}[\mathcal{S},H]}(\mathcal{D},\mathcal{A},\mathsf{aux}) &\leq C\cdot\sqrt{\left(\log\frac{|\mathcal{SK}|\cdot n|\mathcal{AUX}|}{\epsilon}+q\right)q^3\epsilon} + q_K\epsilon + \frac{q_D+1}{|\mathcal{Y}|}\\ &\leq C\sqrt{2}\cdot\sqrt{\left(\log\frac{|\mathcal{SK}|\cdot|\mathcal{AUX}|}{\epsilon}+q\right)q^3\epsilon} + q_K\epsilon + \frac{q_D+1}{|\mathcal{Y}|}\end{aligned}$$

whenever $\min(1/\epsilon,q) \geq n$, where the second inequality is via $q+\log n \leq q+n \leq 2q$. Finally, since $q \geq q_A$, the boundary condition of the above inequality can be relaxed to $\min(1/\epsilon,q_A) \geq n$. This concludes the proof. □

B Proof of Lemma 2

Proof. First, we note that the input z° can be omitted, as it can be hardwired into $\mathcal{A}$.

For the case $k=1$, consider H' to be a fresh random oracle, independent of H. Then, the distributions of $\mathcal{A}^H(H(x^u))$ and $\mathcal{A}^{H'}(H(x^u))$ coincide, unless a query of $\mathcal{A}$ to H happens to be a query on x^u, which happens with probability at most $\frac{q}{|\mathcal{X}|}$. Thus

$$\Pr\left[x^u = \mathcal{A}^H(H(x^u))\right] \leq \Pr\left[x^u = \mathcal{A}^{H'}(H(x^u))\right] + \frac{q}{|\mathcal{X}|} \leq \frac{q+1}{|\mathcal{X}|}\,.$$

For the case $k>1$, instead of considering $\mathcal{A}^H(H(x^u_k))$, the run of $\mathcal{A}$ on the k-th instance, we consider a run of $\mathcal{A}^H_k(H(x^u_k),T_{k-1})$, specified as follows. $\mathcal{A}_k$ is given as additional input the collection T_{k-1} of transcripts of the runs of $\mathcal{A}$ on the previous instances $x^u_1,\ldots,x^u_{k-1}$; this includes each instance x^u_i and its hash $H(x^u_i)$, as well as all the hash queries and responses of these $k-1$ runs of $\mathcal{A}$. $\mathcal{A}_k$ then simply runs $\mathcal{A}$, but whenever $\mathcal{A}$ is about to query H on an input that is contained in T_{k-1}, it reads out the hash from there, instead of querying H. $\mathcal{A}^H_k(H(x^u_k),T_{k-1})$ then obviously behaves identically to $\mathcal{A}^H(H(x^u_k))$. Furthermore, conditioned on any fixed T_{k-1}, the distributions of $\mathcal{A}^H_k(H(x^u_k),T_{k-1})$ and $\mathcal{A}^{H'}_k(H'(x^u_k),T_{k-1})$ coincide, where again H' is a fresh random oracle, unless x^u_k happens to be contained in T_{k-1}, which happens with probability $\frac{(k-1)(q+1)}{\mathcal{X}}$. Thus,

$$\begin{aligned}&\Pr\left[x^u_k = \mathcal{A}^H(H(x^u_k)) \mid x^u_i = \mathcal{A}^H(H(x^u_i))\,\forall\, i<k\right]\\ &\quad= \Pr\left[x^u_k = \mathcal{A}^H_k(H(x^u_k),T_{k-1}) \,\middle|\, x^u_i = \mathcal{A}^H(H(x^u_i))\,\forall\, i<k\right]\\ &\quad\leq \Pr\left[x^u_k = \mathcal{A}^{H'}_k(H'(x^u_k),T_{k-1}) \,\middle|\, x^u_i = \mathcal{A}^H(H(x^u_i))\,\forall\, i<k\right] + (k-1)\frac{q+1}{|\mathcal{X}|}\\ &\quad\leq k\frac{q+1}{|\mathcal{X}|}\end{aligned}$$

where the last inequality follows from the fact for any fixed choice of T_{k-1}, we are back to the case $k=1$ due to the freshness of H'. Multiplying these probability gives the claimed bound. □