BUFFing FALCON without Increasing the Signature Size

Samed Düzlü[1], Rune Fiedler[2], Marc Fischlin[2]

[1] Universität Regensburg, Germany `samed.duzlu@ur.de`
[2] Technische Universität Darmstadt, Germany
`firstname.lastname@cryptoplexity.de`

Abstract. This work shows how FALCON can achieve the Beyond UnForgeability Features (BUFF) introduced by Cremers et al. (S&P'21) more efficiently than by applying the generic BUFF transform. Specifically, we show that applying a transform of Pornin and Stern (ACNS'05), dubbed PS-3 transform, already suffices for FALCON to achieve BUFF security. For FALCON, this merely means to include the public key in the hashing step in signature generation and verification, instead of hashing only the nonce and the message; the other signature computation steps and the signature output remain untouched. In comparison to the BUFF transform, which appends a hash value to the final signature, the PS-3 transform therefore achieves shorter signature sizes, without incurring additional computations.

Keywords: BUFF, Post-Quantum Cryptography, FALCON, (Q)ROM

1 Introduction

In 2017, NIST has initiated a standardization process [11] for post-quantum signature schemes to resist the threat of (future) quantum computers. At the end of round 3 [1], the hash-based SPHINCS+ [9] and the lattice-based Dilithium [10] and FALCON [16] were selected for standardization. Draft standards for the first two schemes [13,14] are available, while the draft standard for FALCON is expected this year. To further diversify the assumptions on which the signature schemes are based, NIST has started an additional call for signature schemes [12].

In their additional call, NIST has declared the Beyond UnForgeability Features, or BUFF for short, formalized by Cremers et al. [4], as *desired features*. The BUFF properties add resilience against maliciously generated (public) keys. The three properties are: Exclusive Ownership (EO, can a signature verify under several public keys?), Message-Bound Signatures (MBS, can a signature verify several messages?), and Non-Resignability (NR, given a signature for an unknown message, can the adversary engineer another signature under its own key for this unknown message?).

Three different signature transforms related to exclusive ownership notions were first introduced by Pornin and Stern [15]. In [4], Cremers et al. show that neither of those transforms generically suffice to ensure security with respect to all BUFF notions. This is why they introduced the BUFF transform, which generically adds BUFF security to any signature scheme. The transform appends a hash of the message and the public key to the signature (over this hash value). In [4], it was shown that the original version of FALCON merely satisfies message-bound signatures, but neither exclusive ownership, nor non-resignability. Hence, the BUFF transform seems necessary for FALCON, and indeed, FALCON-BUFF achieves all BUFF properties.

1.1 BUFFing FALCON Directly

In this work we argue that the third transformation of [15], dubbed PS-3 transform by [4], already suffices for FALCON to achieve all BUFF properties. For signatures, FALCON computes a hash value $\mathsf{H}(r\|m)$ of a nonce r and the message m and then finds a close-by lattice point to this hash value with help of the secret key, described by a value s. The signature then consists of $\sigma = (r, s)$. The verifier also computes $\mathsf{H}(r\|m)$ and verifies that s describes a sufficiently close lattice point to the hash value with help of the public key.

The PS-3 transform adds the public key to the hash evaluation, $\mathsf{H}(r\|\mathsf{pk}\|m)$, both for signing and verifying. The signature, however, remains as before. In particular, one of the pleasing features of FALCON, the short signature size, is not changed at all. Table 1 compares FALCON, as is, FALCON with the BUFF transform (called FALCON-BUFF), and FALCON with the PS-3 transform (called FALCON-PS-3). The table compares the achieved BUFF properties and signature sizes.

Scheme	Sig. target	Sig. format	M-S-UEO	MBS	NR	Size (B)
FALCON	$\mathsf{H}(r\|m)$	(r, s)	✗ [4]	✓ [4]	✗ [4]	1280
FALCON-BUFF	$\mathsf{H}(r\|\mathsf{pk}\|m)$	$(r, s, \mathsf{H}(r\|\mathsf{pk}\|m))$	✓ [4]	✓ [4]	✓ [4]	1344
FALCON-PS-3	$\mathsf{H}(r\|\mathsf{pk}\|m)$	(r, s)	✓ Prop. 14	✓ Prop. 16	✓ Prop. 18	1280

Table 1. The impact of the PS-3 and BUFF transform on FALCON: Signature target, signature format, BUFF properties, and signature size (in bytes, for security level V). ✓ indicates that a property holds and ✗ indicates an attack.

Both, the standard notion of existential unforgeability and our proof of the BUFF security of FALCON-PS-3, make use of the same assumptions as FALCON, namely, the random oracle model and the same underlying lattice hardness assumption. Unforgeability of FALCON follows the GPV framework [7] and is proven in the (Q)ROM and reduces to the SIS problem for NTRU lattices. We show that unforgeability of FALCON-PS-3 is reduced to unforgeability of FALCON directly, without requiring additional assumptions, and give the proofs for the BUFF security of FALCON-PS-3 in the random oracle model (while the original BUFF transform refers to standard assumptions like collision resistance). For the two weaker variants S-CEO and S-DEO of M-S-UEO and NR we discuss security in the quantum setting under the quantum random oracle assumption.

In our view, the short signature size of FALCON-PS-3, together with the fact that the important unforgeability property is not weakened by the PS-3 transform, and that the random oracle assumption for the BUFF properties appears to be necessary in FALCON anyway, makes FALCON-PS-3 the better candidate for standardization than FALCON-BUFF.

1.2 Related Work

Aulbach et al. [2] have analyzed the BUFF security of the additional signature candidates for the NIST standardization process that are based on codes, isogenies, lattices, and multivariate equations. Their analysis considers an exclusive-ownership flavor called S-UEO, which guarantees exclusive ownership for honestly generated key pairs and signatures, as well as a weaker version of non-resignability called wNR. In particular, they observe that the PS-3 transform is sufficient to secure against S-UEO, MBS, and wNR for many of the alternative candidates, but not *in general*. The results regarding FALCON here are inspired by their analysis.

Recently, Don et al. [6] noted that the original non-resignability notion in [4] is unachievable if one allows for arbitrary auxiliary data about the unknown message. The definition is easy to fix by demanding that the auxiliary data is computationally independent of the message, as already pointed out in the latest version of [4]. We discuss the different definitions of non-resignability in Section 2.3.

2 Preliminaries

We briefly introduce our notation, review the syntax and unforgeability of signature schemes, followed by the BUFF properties and two of the transformations proposed to achieve (some of) the BUFF properties.

2.1 Notation

We write $y \leftarrow A(x)$ for a deterministic algorithm A with input x and output y, and $y \leftarrow_{\$} A(x)$ for the random output y of the probabilistic algorithm A on input x. Here, the probability is over A's internal coin tosses. In cryptography, algorithms usually receive the security parameter in unary, 1^n, as additional input; we often omit this extra input for sake of brevity and write for instance $\mathsf{KGen}()$ instead of $\mathsf{KGen}(1^n)$. Similarly, when algorithms operate in presence of an oracle, typically a random oracle H, then we also omit writing oracle access explicitly and simply write $y \leftarrow_{\$} A(x)$ instead of $y \leftarrow_{\$} A^{\mathsf{H}}(x)$. In particular, if A is adversarial and quantum, then it has quantum access to the random oracle H. Yet, algorithms of schemes and protocols proceed classically, such that oracle access is still classical for such algorithms. Vice versa, the adversary, too, only has classical oracle access to such procedures, e.g., to the signing oracle $\mathsf{Sign}(\mathsf{sk}, \cdot)$ in the unforgeability definition.

We write the concatenation of two strings a and b as $a\|b$. If we want to emphasize that encoding of two strings via concatenation is reversible with context information, e.g., if it is clear that a is of fixed length, then we often write (a, b) instead of $a\|b$.

2.2 Syntax and unforgeability

Definition 1 (Signatures). *A signature scheme $\Sigma = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ with associated message space $\mathcal{M}$ is a triple of three* PPT *algorithms:*

- $\mathsf{KGen}() \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$ *outputs a key pair consisting of a public key* pk *and a secret key* sk
- $\mathsf{Sign}(\mathsf{sk}, m) \xrightarrow{\$} \sigma$ *takes a secret key* sk *and a message m as input and outputs a signature σ*
- $\mathsf{Verify}(\mathsf{pk}, m, \sigma) \to d$ *takes a public key* pk*, a message m, and a signature σ as input and outputs a decision bit $d \in \{0, 1\}$. If $d = 1$ we say the signature is valid.*

A signature scheme Σ is δ-correct, if for every $(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{KGen}()$ and every $m \in \mathcal{M}$ it holds that $\Pr[\mathsf{Verify}(\mathsf{pk}, m, \mathsf{Sign}(\mathsf{sk}, m)) = 1] \geq 1 - \delta$. It is correct if it is 0-correct.

The conventional security requirement for signature schemes is unforgeability under chosen message attacks (just unforgeability for short).

Definition 2 (Unforgeability). *Let Σ be a signature scheme. We say that Σ provides* existential unforgeability under chosen-message attacks (EUF-CMA) *if, for every* PPT *algorithm $\mathcal{A}$, there exists a negligible function $\mu\colon \mathbb{N} \to \mathbb{R}$ such that, for every $n \in \mathbb{N}$,*

$$\boldsymbol{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Sigma,\mathcal{A}}(n) := \Pr[\mathbf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\Sigma,\mathcal{A}}(n)] \leq \mu(n),$$

where $\mathbf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\Sigma,\mathcal{A}}(n)$ is defined on the left-hand side in Figure 1.

2.3 BUFF properties

Exclusive ownership ensures that a signature does not verify messages under two distinct public keys, where the two messages are identical (constructive EO), distinct (destructive EO), or arbitrary (universal EO). The prefixes indicate that the adversary chooses the keys itself (malicious) and does not need to output a secret key (strong). Note that M-S-UEO implies S-UEO, which in turn is equivalent to the combination of S-CEO and S-DEO.

```
$\mathbf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\Sigma,\mathcal{A}}(n)$:
    $\mathcal{Q} \leftarrow \emptyset$
    $(\mathsf{sk}, \mathsf{pk}) \leftarrow\!\!\$\ \mathsf{KGen}(1^n)$
    $(m', \sigma') \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
    $d \leftarrow \mathsf{Verify}(\mathsf{pk}, m', \sigma')$
    $\mathbf{return}\ \left[d = 1 \ \wedge\ m' \notin \mathcal{Q}\right]$

$\mathsf{Sign}(\mathsf{sk}, m)$:
    $\sigma \leftarrow\!\!\$\ \mathsf{Sign}(\mathsf{sk}, m)$
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
    $\mathbf{return}\ \sigma$
```

Fig. 1. Definition of the experiment $\mathbf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\Sigma,\mathcal{A}}(n)$ from Definition 2.

```
$\mathbf{Exp}^{\mathsf{S\text{-}CEO}}_{\Sigma,\mathcal{A}}(n)$:
    $\mathcal{Q} \leftarrow \emptyset$
    $(\mathsf{sk}, \mathsf{pk}) \leftarrow\!\!\$\ \mathsf{KGen}(1^n)$
    $(m', \sigma', \mathsf{pk}') \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
    $d \leftarrow \mathsf{Verify}(\mathsf{pk}', m', \sigma')$
    $\mathbf{return}\ \left[d = 1 \ \wedge\ (m', \sigma') \in \mathcal{Q} \ \wedge\ \mathsf{pk}' \neq \mathsf{pk}\right]$

$\mathsf{Sign}(\mathsf{sk}, m)$:
    $\sigma \leftarrow\!\!\$\ \mathsf{Sign}(\mathsf{sk}, m)$
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(m, \sigma)\}$
    $\mathbf{return}\ \sigma$

$\mathbf{Exp}^{\mathsf{S\text{-}DEO}}_{\Sigma,\mathcal{A}}(n)$:
    $\mathcal{Q} \leftarrow \emptyset$
    $(\mathsf{sk}, \mathsf{pk}) \leftarrow\!\!\$\ \mathsf{KGen}(1^n)$
    $(m', \sigma', \mathsf{pk}') \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
    $d \leftarrow \mathsf{Verify}(\mathsf{pk}', m', \sigma')$
    $\mathbf{return}\ \left[d = 1 \ \wedge\ \left(\exists m^* \neq m' \colon (m^*, \sigma') \in \mathcal{Q}\right) \ \wedge\ \mathsf{pk}' \neq \mathsf{pk}\right]$
```

Fig. 2. Definition of the experiments $\mathbf{Exp}^{\mathsf{S\text{-}CEO}}_{\Sigma,\mathcal{A}}(n)$ and $\mathbf{Exp}^{\mathsf{S\text{-}DEO}}_{\Sigma,\mathcal{A}}(n)$ from Definitions 3 and 4, respectively.

Definition 3 (Strong Constructive Exclusive Ownership [4]). *Let Σ be a digital signature scheme. We say that Σ provides* strong constructive exclusive ownership ($\mathsf{S\text{-}CEO}$) *if, for every* PPT *algorithm $\mathcal{A}$, there exists a negligible function $\mu\colon \mathbb{N} \to \mathbb{R}$ such that, for every $n \in \mathbb{N}$,*

$$\boldsymbol{Adv}^{\mathsf{S\text{-}CEO}}_{\Sigma,\mathcal{A}}(n) := \Pr[\mathbf{Exp}^{\mathsf{S\text{-}CEO}}_{\Sigma,\mathcal{A}}(n)] \leq \mu(n),$$

where $\mathbf{Exp}^{\mathsf{S\text{-}CEO}}_{\Sigma,\mathcal{A}}(n)$ is defined on the top in Figure 2.

Definition 4 (Strong Destructive Exclusive Ownership [4]). *Let Σ be a digital signature scheme. We say that Σ provides* strong destructive exclusive ownership ($\mathsf{S\text{-}DEO}$) *if, for every* PPT *algorithm $\mathcal{A}$, there exists a negligible function $\mu\colon \mathbb{N} \to \mathbb{R}$ such that, for every $n \in \mathbb{N}$,*

$$\boldsymbol{Adv}^{\mathsf{S\text{-}DEO}}_{\Sigma,\mathcal{A}}(n) := \Pr[\mathbf{Exp}^{\mathsf{S\text{-}DEO}}_{\Sigma,\mathcal{A}}(n)] \leq \mu(n),$$

where $\mathbf{Exp}^{\mathsf{S\text{-}DEO}}_{\Sigma,\mathcal{A}}(n)$ is defined on the bottom in Figure 2.

```
$\mathbf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\Sigma,\mathcal{A}}(n)$:
61 : $(m_1, m_2, \sigma, \mathsf{pk}_1, \mathsf{pk}_2) \leftarrow_{\$} \mathcal{A}(1^n)$
62 : $d_1 \leftarrow \mathsf{Verify}(\mathsf{pk}_1, m_1, \sigma)$
63 : $d_2 \leftarrow \mathsf{Verify}(\mathsf{pk}_2, m_2, \sigma)$
64 : return $[d_1 = 1 \wedge d_2 = 1$
65 :     $\wedge\ \mathsf{pk}_1 \neq \mathsf{pk}_2]$

$\mathbf{Exp}^{\mathsf{MBS}}_{\Sigma,\mathcal{A}}(n)$:
71 : $(m_1, m_2, \sigma, \mathsf{pk}) \leftarrow_{\$} \mathcal{A}(1^n)$
72 : $d_1 \leftarrow \mathsf{Verify}(\mathsf{pk}, m_1, \sigma)$
73 : $d_2 \leftarrow \mathsf{Verify}(\mathsf{pk}, m_2, \sigma)$
74 : return $[d_1 = 1\ \wedge\ d_2 = 1$
75 :     $\wedge\ m_1 \neq m_2]$
```

Fig. 3. Definition of the experiments $\mathbf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\Sigma,\mathcal{A}}(n)$ and $\mathbf{Exp}^{\mathsf{MBS}}_{\Sigma,\mathcal{A}}(n)$ from Definitions 5 and 6, respectively.

Definition 5 (Malicious-Strong Universal Exclusive Ownership [4]). *Let Σ be a digital signature scheme. We say that Σ provides* malicious-strong universal exclusive ownership ($\mathsf{M\text{-}S\text{-}UEO}$) *if, for every* PPT *algorithm $\mathcal{A}$, there exists a negligible function $\mu\colon \mathbb{N} \to \mathbb{R}$ such that, for every $n \in \mathbb{N}$,*

$$\boldsymbol{Adv}^{\mathsf{M\text{-}S\text{-}UEO}}_{\Sigma,\mathcal{A}}(n) := \Pr[\mathbf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\Sigma,\mathcal{A}}(n)] \leq \mu(n),$$

where $\mathbf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\Sigma,\mathcal{A}}(n)$ is defined on the left-hand side in Figure 3.

Message-bound signatures ensure that a signature does not verify two messages under the same public key.

Definition 6 (Message-Bound Signatures [4]). *Let Σ be a digital signature scheme. We say that Σ provides* message-bound signatures (MBS) *if, for every*

PPT *algorithm $\mathcal{A}$, there exists a negligible function $\mu\colon \mathbb{N} \to \mathbb{R}$ such that, for every $n \in \mathbb{N}$, it holds that*

$$\boldsymbol{Adv}^{\mathsf{MBS}}_{\Sigma,\mathcal{A}}(n) := \Pr[\mathbf{Exp}^{\mathsf{MBS}}_{\Sigma,\mathcal{A}}(n)] \le \mu(n),$$

where $\mathbf{Exp}^{\mathsf{MBS}}_{\Sigma,\mathcal{A}}(n)$ is defined on the right-hand side in Figure 3.

The third BUFF property is called non-resignability. Intuitively, it guarantees the following: Given a signature on an unknown message, an adversary cannot produce a signature for the same (unknown) message that verifies under a new public key. This leaves some wiggle room to formalize that an adversary may have circumstantial knowledge about the message. The original definition of [4] did this with auxiliary info that has sufficient min-entropy. However, as [6] points out, this definition is not achievable since the auxiliary info can directly contain a signature under a fresh key pair. Instead, [6] propose a different formalization of non-resignability, where message sampling and generation of auxiliary info is split into two algorithms, where the auxiliary info has to be generated without access to the random oracle. To circumvent this issue, [2] has proposed a variant named *weak non-resignability (*wNR*)*, which does not use any auxiliary info at all. In Remark 19 we discuss the relation between our new definition and wNR in more detail.

Since our results are in the random oracle model and in the quantum random oracle model, we chose to define the notion more directly via unpredictability:

Definition 7 (Unpredictability). *For (quantum) random oracle H we say that a message distribution $\mathcal{D}$ is* unpredictable with respect to KGen*, if for any algorithm $\mathcal{A}$ there exists a negligible function $\mu\colon \mathbb{N} \to \mathbb{R}$ such that, for every $n \in \mathbb{N}$, it holds that*

$$\boldsymbol{Adv}^{\mathsf{pred}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{A}}(n) := \Pr[\mathbf{Exp}^{\mathsf{pred}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{A}}(n)] \le \mu(n),$$

where $\mathbf{Exp}^{\mathsf{pred}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{A}}(n)$ is defined on the left-hand side in Figure 4.

Note that the above definition can be used with both classical as well as quantum algorithms (with superposition oracle queries), and with efficient and potentially also unbounded algorithms $\mathcal{A}$. It comes close in spirit to the notion of unpredictability entropy [8].

As additional tool we need computational indistinguishability of the auxiliary data (following [5], the updated full version of [4]).

Definition 8 (Computational Independence of auxiliary data). *We say that a message distribution $\mathcal{D}$ has* computationally independent auxiliary data with respect to key generator KGen*, if for every PPT algorithm $\mathcal{A}$ there exists a negligible function $\mu\colon \mathbb{N} \to \mathbb{R}$ such that, for every $n \in \mathbb{N}$, it holds that*

$$\boldsymbol{Adv}^{\mathsf{CI\text{-}aux}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{A}}(n) := \Pr[\mathbf{Exp}^{\mathsf{CI\text{-}aux}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{A}}(n)] - \frac{1}{2} \le \mu(n),$$

where $\mathbf{Exp}^{\mathsf{CI\text{-}aux}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{A}}(n)$ is defined on the right-hand side in Figure 4.

```
$\mathbf{Exp}^{\mathsf{pred}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{A}}(n)$:
81 : $(\mathsf{pk},\mathsf{sk}) \leftarrow_\$ \mathsf{KGen}(1^n)$
82 : $(m,\mathsf{aux}) \leftarrow_\$ \mathcal{D}(\mathsf{pk})$
83 : $m' \leftarrow_\$ \mathcal{A}(\mathsf{pk},\mathsf{sk},\mathsf{aux})$
84 : return $[m = m']$
```

```
$\mathbf{Exp}^{\mathsf{CI\text{-}aux}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{A}}(n)$:
91 : $(\mathsf{pk},\mathsf{sk}) \leftarrow_\$ \mathsf{KGen}(1^n)$
92 : $b \leftarrow_\$ \{0,1\}$
93 : $(m_0,\mathsf{aux}_0) \leftarrow_\$ \mathcal{D}(1^n,\mathsf{pk})$
94 : $(m_1,\mathsf{aux}_1) \leftarrow_\$ \mathcal{D}(1^n,\mathsf{pk})$
95 : $b' \leftarrow_\$ \mathcal{A}^{\mathsf{H}}(\mathsf{pk},\mathsf{sk},m_0,\mathsf{aux}_b)$
96 : return $[b' = b]$
```

Fig. 4. On the left-hand side, the definition of the experiment $\mathbf{Exp}^{\mathsf{pred}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{A}}(n)$ from Definition 7, on the right-hand side, the definition of the experiment $\mathbf{Exp}^{\mathsf{CI\text{-}aux}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{A}}(n)$ from Definition 8.

Definition 9 (Non-Resignability). *Let Σ be a digital signature scheme. We say that Σ is* non-resignable (NR) *if, for every* PPT *algorithms $\mathcal{A}$ and $\mathcal{D}$, where $\mathcal{D}$ is unpredictable with computationally-independent auxiliary data for* KGen*, there exists a negligible function $\mu\colon \mathbb{N} \to \mathbb{R}$ such that, for every $n \in \mathbb{N}$, it holds that*

$$\boldsymbol{Adv}^{\mathsf{NR}}_{\Sigma,\mathcal{A},\mathcal{D}}(n) := \Pr[\mathbf{Exp}^{\mathsf{NR}}_{\Sigma,\mathcal{A},\mathcal{D}}(n)] \leq \mu(n),$$

where $\mathbf{Exp}^{\mathsf{NR}}_{\Sigma,\mathcal{A},\mathcal{D}}(n)$ is defined in Figure 5.

```
$\mathbf{Exp}^{\mathsf{NR}}_{\Sigma,\mathcal{A},\mathcal{D}}(n)$:
101 : $(\mathsf{pk},\mathsf{sk}) \leftarrow_\$ \mathsf{KGen}(1^n)$
102 : $(m,\mathsf{aux}) \leftarrow_\$ \mathcal{D}(1^n,\mathsf{pk})$
103 : $\sigma \leftarrow_\$ \mathsf{Sign}(\mathsf{sk},m)$
104 : $(\sigma',\mathsf{pk}') \leftarrow_\$ \mathcal{A}(\mathsf{pk},\sigma,\mathsf{aux})$
105 : $d \leftarrow \mathsf{Verify}(\mathsf{pk}',m,\sigma')$
106 : return $[d = 1 \wedge \mathsf{pk}' \neq \mathsf{pk}]$
```

Fig. 5. The definition of the experiment $\mathbf{Exp}^{\mathsf{NR}}_{\Sigma,\mathcal{A},\mathcal{D}}(n)$ from Definition 9.

2.4 Transformations

We briefly review the BUFF transformation and the PS-3 transformation with the help of Figure 6: Based on a signature scheme Σ and a hash function H, we change the signing and verification algorithms as follows: Signing first computes a hash digest of the message and the public key, which it then signs with the original scheme. It returns the signature and — only for the BUFF transformation — also the hash digest. Verification unwraps the signature, re-computes the hash

```
$\mathsf{KGen}^*(1^n)$:
11 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow_\$ \mathsf{KGen}(1^n)$
12 : **return** $(\mathsf{sk}, \mathsf{pk})$

$\mathsf{Sign}^*(\mathsf{sk}, m)$:
21 : $h \leftarrow \mathsf{H}(m, \mathsf{pk})$
22 : $\sigma \leftarrow_\$ \mathsf{Sign}(\mathsf{sk}, h)$
23 : $\sigma^* \leftarrow (\sigma, \boxed{h})$
24 : **return** $\sigma^*$

$\mathsf{Verify}^*(\mathsf{pk}, m, \sigma^*)$:
31 : $(\hat{\sigma}, \boxed{\hat{h}}) \leftarrow \sigma^*$
32 : $h \leftarrow \mathsf{H}(m, \mathsf{pk})$
33 : $d \leftarrow \mathsf{Verify}(\mathsf{pk}, h, \hat{\sigma})$
34 : **return** $\left[d = 1 \ \wedge \ \boxed{\hat{h} = h}\right]$
```

Fig. 6. Applying the BUFF [4] or PS-3 [15] transformation to a signature scheme $\Sigma = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ with a hash function H, yielding a transformed signature scheme $\Sigma^* = (\mathsf{KGen}^*, \mathsf{Sign}^*, \mathsf{Verify}^*)$. $\boxed{\text{Boxed lines}}$ are exclusive to the BUFF transformation.

digest, and verifies the signature with the original verification algorithm for the digest as message. For the BUFF transformation, verification additionally checks that the re-computed digest and the digest in the signature match.

Many signatures schemes, e.g., FALCON, begin the signing procedure with hashing the message (possibly together with some other input). We can combine this hash function call with the hash function call of the transformation (shown in line 21 of Figure 6), yielding a more compact notation.

3 Description of FALCON and its Transforms

FALCON is a lattice-based signature scheme over NTRU that makes use of the GPV framework [7]. Figure 7 provides an algorithmic description of FALCON and its transformed variants, following the exposition of [4]. The public keys and all values that appear in arithmetic operations in FALCON are polynomials in $\mathbb{Z}[x]/(q, \phi)$, where ϕ is a polynomial of degree n and with $q = 12\,289$. The values are chosen such that addition and multiplication in $\mathbb{Z}[x]/(q, \phi)$ correspond to addition and multiplication in $\mathbb{Z}_q^n$, so we use these two rings interchangeably, often without explicitly mentioning the isomorphism. FALCON comes with two parameter sets. For our purpose, the parameters we require are $n = 512, 1024$ and β which is given respectively as $\lfloor \beta \rfloor^2 = 34\,034\,726$ and $\beta^2 = 70\,265\,242$. Rounding up, we set β to be 5834 and 8383, respectively, which is sufficient for us.

To create a signature, we compute a target value c and use the trapdoor in the secret key to compute a short vector s that defines a lattice point which is close to c. The bound is for the resulting size is given by β. Verification recomputes the target c and checks that the element s in the signature is short and its corresponding lattice point is close to c. For further information on how these algorithms (and the subroutines FalconTree, NTRUGen, FFT, and FFSampling) work we refer the reader to [4] or [16].

It is worth noting that the hashing computation H, called HashToPoint in [16], is via iterations of SHAKE-256 and outputs values from $\mathbb{Z}_q^n$. To this end, the hashing algorithm first hashes the entire input string via the injection procedure of SHAKE and then iteratively extracts values $c_j \in \mathbb{Z}_q$ via the extraction

```
$\mathsf{KGen}(1^n)$
11 : $(f, g, F, G) \leftarrow_\$ \mathsf{NTRUGen}(\phi, q)$
12 : $\mathbf{B} \leftarrow \begin{bmatrix} g & -f \\ G & -F \end{bmatrix}$
13 : $\hat{\mathbf{B}} \leftarrow \mathsf{FFT}(\mathbf{B})$
14 : $T \leftarrow \mathsf{FalconTree}(\hat{\mathbf{B}})$
15 : $\mathsf{sk} \leftarrow (\hat{\mathbf{B}}, T)$
16 : $h \leftarrow gf^{-1}$
17 : $\mathsf{pk} \leftarrow h$
18 : $\mathbf{return}\ (\mathsf{sk}, \mathsf{pk})$
```

```
$\mathsf{Sign}(\mathsf{sk}, \mathsf{pk}, m)$
21 : $h \leftarrow \mathsf{pk}$
22 : $(\hat{\mathbf{B}}, T) \leftarrow \mathsf{sk}$
23 : $r \leftarrow_\$ \{0,1\}^{320}$
24 : $c \leftarrow \mathsf{H}(r \| m)$            // FALCON
     $c \leftarrow \mathsf{H}(r \| h \| m)$       // FALCON-PS-3
25 : $\mathbf{t} \leftarrow (\mathsf{FFT}(c), \mathsf{FFT}(0)) \cdot \hat{\mathbf{B}}^{-1}$
26 : $\mathbf{s} \leftarrow_\$ \mathsf{FFSampling}(\mathbf{t}, T, \lfloor \beta^2 \rfloor)$
27 : $(s_1, s_2) \leftarrow \mathsf{FFT}^{-1}(\mathbf{s})$
28 : $s \leftarrow \mathsf{Compress}(s_2)$
29 : $\sigma \leftarrow (r, s)$
30 : $\mathbf{return}\ \sigma$
```

```
$\mathsf{Verify}(\mathsf{pk}, m, \sigma)$
31 : $h \leftarrow \mathsf{pk}$
32 : $(r, s) \leftarrow \sigma$
33 : $c \leftarrow \mathsf{H}(r \| m)$            // FALCON
     $c \leftarrow \mathsf{H}(r \| h \| m)$       // FALCON-PS-3
34 : $s_2 \leftarrow \mathsf{Decompress}(s)$
35 : $s_1 \leftarrow c - s_2 h$
36 : $\mathbf{return}\ [\|(s_1, s_2)\|^2 \leq \lfloor \beta^2 \rfloor]$
```

Fig. 7. Algorithmic description of FALCON and FALCON-PS-3.

procedure of SHAKE to create each entry in the final output $c = \sum c_j x^j \in \mathbb{Z}_q^n$. For each scalar c_j the algorithm repeatedly extracts the next 16 bits output from SHAKE, views it as an integer t, checks that $t < kq$ for $k = \lfloor k/2^{16} \rfloor$ and, if so, sets $c_j \leftarrow t \bmod q$. This means that, if one assumes that SHAKE behaves like an extendable-output random oracle, that the output value c is uniform. We may thus assume that H behaves like a random oracle.

Regarding the encoding of values, we assume that $h = \mathsf{pk}$ is of fixed length (for given n) and that keys for the different security parameters $n = 512$ and $n = 1024$ are differently encoded in the sense that one cannot be the prefix of the other one. This is indeed the case in FALCON, as public keys start with a header byte $0000k_1k_2k_3k_4$ where bits $k_1 \ldots k_4$ encode $\log_2 n \in \{9, 10\}$. This ensures in our case that hashing $r\|h\|m$ for fixed-length r can never coincide with encodings with other nonces r', public keys h', and messages m'. We also note that signatures σ in Falcon are encoded in such a way that the entries r, s are recoverable; we thus simply describe signatures here as tuples $\sigma = (r, s)$.

Looking at the original scheme FALCON, the scheme after applying the PS-3, called FALCON-PS-3, and the scheme after applying the BUFF transform, called FALCON-BUFF, we notice the following differences: After applying either transform, the target c additionally depends on the public key (see lines 24 and 33). FALCON-BUFF additionally carries a hash digest of $r\|\mathsf{pk}\|m$ in the signature and checks the re-computed digest during verification (not depicted in the figure, would affect lines 29, 32, 33). We can use the first 512 bits squeezed from SHAKE-256 as digest to sidestep including the complete output c of HashToPoint, which is of size approximately $n \log q$ (corresponding to roughly 6900 bits for $n = 512$ or 13800 bits for $n = 1024$), in the signature.

4 Analysis of FALCON-PS-3

We first show that FALCON-PS-3 retains unforgeability of FALCON.

Proposition 10 (Unforgeability). *Assuming* FALCON *is unforgeable, then the advantage of an adversary $\mathcal{A}$ against* EUF-CMA *of* FALCON-PS-3 *is at most*

$$\boldsymbol{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{FALCON\text{-}PS\text{-}3},\mathcal{A}}(n) \le \boldsymbol{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{FALCON},\mathcal{B}}(n).$$

for some adversary $\mathcal{B}$ where $\mathcal{B}$ runs in approximately the same time as $\mathcal{A}$.

Proof. We construct an adversary $\mathcal{B}$ against FALCON from an adversary $\mathcal{A}$ against FALCON-PS-3 as follows: The reduction $\mathcal{B}$ is started with a public key $\mathsf{pk} = h$ as input, which it passes on to $\mathcal{A}$. For any signature query m_i of $\mathcal{A}$, the reduction queries $h\|m_i$, i.e., the public key concatenated with the message, to its own signing oracle. Hence, the signing oracle computes the target c as $\mathsf{H}(r\|h\|m_i)$. This results in a valid signature for m_i under FALCON-PS-3 and a valid signature for $h\|m_i$ under FALCON. Once $\mathcal{A}$ returns a message and signature (m^*, σ^*) as forgery, the reduction outputs $(h\|m^*, \sigma^*)$. If σ^* verifies m^* under FALCON-PS-3, it also verifies $h\|m^*$ under FALCON and if m^* has not been signed in $\mathcal{A}$'s simulation before, then neither has $h\|m^*$ in $\mathcal{B}$'s attack. Hence, the reduction wins if $\mathcal{A}$ wins. $\square$

Preparation for BUFF security of FALCON-PS-3. To analyze the BUFF security of FALCON-PS-3, we introduce some notation and give some probability estimations that are useful in the proofs later.

Let $R > 0$ be some bound. We define $S_R \coloneqq \{x \in \mathbb{Z}[x]/(q, \phi) \mid \|x\| \le R\}$ as the set of elements from $\mathbb{Z}[x]/(q, \phi)$ of norm at most R and $s_R \coloneqq \#S_R$ the cardinality of this set. On the other hand, we set $B(R) \coloneqq \{x \in \mathbb{R}^n \mid \|x\| < R\}$ to be the Euclidean ball of elements from $\mathbb{R}^n$ of norm less than R. We let $\mathsf{vol}(B(R))$ denote the volume of this ball.

We begin by estimating the probability of a randomly chosen element from $\mathbb{Z}[x]/(q, \phi)$ to lie in the set S_R of such elements with some norm bound.

Lemma 11. *Let $R + \frac{\sqrt{n}}{2} \leq q$ and $n = 2^k$. Then, for uniformly random $c \in \mathbb{Z}[x]/(q, \phi)$, we have*

$$\Pr[c \in S_R] < 2^{(5-k)\frac{n}{2}} \left(\frac{R + \frac{\sqrt{n}}{2}}{q} \right)^n \leq 2^{(5-k)\frac{n}{2}}.$$

Proof. First, we can estimate the number of elements of S_R as

$$s_R \leq \mathsf{vol}\left(B(R + \tfrac{\sqrt{n}}{2}) \right).$$

Indeed, let H_x be the n-dimensional hypercube with edge length 1 centered at x. Note, $\mathsf{vol}(H_x) = 1$. Further, for $x \neq y \in \mathbb{Z}^n$, we have $H_x \cap H_y = \emptyset$. Finally, note that the diameter of any H_x is $\sqrt{n}$ so that the maximum distance of x to any element of H_x is less than $\frac{\sqrt{n}}{2}$. Therefore, $\bigcup_{x \in S_R} H_x \subseteq B(R + \frac{\sqrt{n}}{2})$. Using these facts, we have

$$s_R = \sum_{x \in S_R} \mathsf{vol}(H_x) \leq \mathsf{vol}\left(B(R + \tfrac{\sqrt{n}}{2}) \right).$$

The formula for the volume of a ball of radius $R + \frac{\sqrt{n}}{2}$ in dimension n is given by

$$\mathsf{vol}\left(B(R + \tfrac{\sqrt{n}}{2}) \right) = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2} + 1\right)} \left(R + \frac{\sqrt{n}}{2} \right)^n.$$

Here, Γ is Euler's Gamma function, which we bound using Stirling's formula[3] to get

$$\Gamma\left(\frac{n}{2} + 1\right) > \sqrt{\pi n}(\tfrac{n}{2e})^{n/2}.$$

For $n = 2^k$ we get, using $\pi, e < 4$,

$$\frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2} + 1\right)} < \frac{\pi^{(n-1)/2}}{n^{1/2}} \left(\frac{2e}{n} \right)^{n/2} < 2^{n-1-\frac{k}{2}+(3-k)\frac{n}{2}} < 2^{(5-k)\frac{n}{2}}.$$

Thus, we have

$$s_R < 2^{(5-k)\frac{n}{2}} \left(R + \frac{\sqrt{n}}{2} \right)^n.$$

Assuming $R + \frac{\sqrt{n}}{2} \leq q$, we have

$$\Pr[c \in S_R] = \frac{s_R}{q^n} < 2^{(5-k)\frac{n}{2}} \left(\frac{R + \frac{\sqrt{n}}{2}}{q} \right)^n \leq 2^{(5-k)\frac{n}{2}}$$

as was claimed. □

Table 2 illustrates this probability for all security levels of FALCON using the estimation of Lemma 11.

[3] Here, $\Gamma(m+1) > \sqrt{2\pi m}\left(\frac{m}{e}\right)^m e^{1/(12m+1)} \geq \sqrt{2\pi m}\left(\frac{m}{e}\right)^m$

Parameter	level I	level V
n	512	1024
β	5834	8383
q	12289	12289
$\Pr[c \in S_R]$	$< 2^{-1024}$	$< 2^{-2560}$

Table 2. The table shows the FALCON parameters relevant for the BUFF analysis of FALCON-PS-3 for $n = 512 = 2^9$ and $n = 1024 = 2^{10}$. Additionally, the row $\Pr[c \in S_R]$ gives an upper bound on the probability of a random element being of some size $\leq R$, as long as $R + \frac{\sqrt{n}}{2} \leq q$, using the estimation of Lemma 11.

For M-S-UEO and MBS, an adversary is required to find two messages that are verified with the same signature. This translates to the situation where two uniformly random c_1, c_2 are close to a certain lattice, which we define now.

Associated to a pair $h_1, h_2 \in \mathbb{Z}[x]/(q, \phi)$ that play the role of public keys, we define the lattice Λ_{h_1,h_2} by

$$\{(zh_1, zh_2) \mid z \in \mathbb{Z}[x]/(q, \phi)\}.$$

It is important to note that Λ_{h_1,h_2} is a lattice of rank n in the Euclidean space of dimension $2n$.

The bound $R = \sqrt{2}\beta$ in the following statement will be used in the proof of the Propositions 14 and 17 below. Namely, one signature (r, s) for public keys h_1 and h_2 defines an element $(sh_1, sh_2) \in \Lambda_{h_1,h_2}$. The condition that two values c_1, c_2 computed as hashes of random salt, public key, and message verify under the signature, implies that $c = (c_1, c_2)$ is close to Λ_{h_1,h_2} within the distance $\sqrt{2}\beta$.

In these two security games M-S-UEO and MBS, $h_1 \neq h_2$ or $m_1 \neq m_2$, respectively, for successful attacks, so that the values $c_\ell = \mathsf{H}(r \| h_\ell \| m_\ell)$ for $\ell \in \{1, 2\}$ are independent and uniformly random in either case. We continue by estimating the probability that the bound $\sqrt{2}\beta$ is satisfied for a randomly chosen pair $c = (c_1, c_2)$.

Lemma 12. *For any h_1, h_2 and uniformly random $c = (c_1, c_2)$ with $c_i \in \mathbb{Z}[x]/(q, \phi)$, and $\sqrt{2}\beta + \frac{\sqrt{n}}{2} < q$, we have*

$$\Pr\left[\mathsf{dist}\left(\Lambda_{h_1,h_2}, c\right) \leq \sqrt{2}\beta\right] < 2^{(5-k)\frac{n}{2}},$$

where $\mathsf{dist}(\Lambda_{h_1,h_2}, c)$ is the shortest distance of c to any element of Λ_{h_1,h_2}.

Note that $\sqrt{2}\beta + \frac{\sqrt{n}}{2} < q$ applies for either parameter set of FALCON.

Proof. Up to rotation, we may assume that $\Lambda_{h_1,h_2} \subseteq \mathbb{R}^n \times \{0\}$, as the rank of Λ_{h_1,h_2} is n. Note also that a rotation does not change the norm, nor the distribution of the random points, nor the number of elements in the standard lattice that satisfy some bound R. For the distance, we then have $\mathsf{dist}(\Lambda_{h_1,h_2}, c) \geq$

$\|c_2\|$, since it suffices to measure the distance of the second component from the rotated-to-0 point. Hence,

$$\Pr\Big[\mathsf{dist}\,(\Lambda_{h_1,h_2}, c) \leq \sqrt{2}\beta\Big] \leq \Pr\Big[\|c_2\| \leq \sqrt{2}\beta\Big].$$

It thus suffices to bound the right hand side, where we are in the situation of Lemma 11 with $R = \sqrt{2}\beta$. Using Lemma 11, we have

$$\begin{aligned}\Pr\Big[\mathsf{dist}\,(\Lambda_{h_1,h_2}, c) \leq \sqrt{2}\beta\Big] &\leq \Pr\Big[\|c_2\| \leq \sqrt{2}\beta\Big] \\ &< 2^{(5-k)\frac{n}{2}} \left(\frac{\sqrt{2}\beta + \frac{\sqrt{n}}{2}}{q}\right)^n \\ &\leq 2^{(5-k)\frac{n}{2}},\end{aligned}$$

as was claimed. □

The proof of the lemma can been seen as a bound to estimate the probability that two random hash values can be matched to a single signature. But the lemma reveals another useful property which we will take advantage of later: For any given h_1, h_2 and any (sufficiently close) *given* c_1, the probability that a random c_2 is close to the lattice is small. This can be seen as a bound to estimate that, for a given valid signature, another independent hash value most likely cannot be matched to this signature (or even to an adapted one for a different value s'):

Corollary 13. *For any h_1, h_2 and any c_1 with $\|c_1 - sh_1\| \leq \beta$ for some $s \in \mathbb{Z}[x]/(q, \phi)$, for uniformly random c_2 from $\mathbb{Z}[x]/(q, \phi)$, and $\sqrt{2}\beta + \frac{\sqrt{n}}{2} < q$, we have*

$$\Pr\Big[\mathsf{dist}\,(\Lambda_{h_1,h_2}, (c_1, c_2)) \leq \sqrt{2}\beta\Big] < 2^{(5-k)\frac{n}{2}}.$$

Proof. Rotate the lattice Λ_{h_1,h_2} as in the proof of the lemma to $\mathbb{R}^n \times \{0\}$ and note that we bound the probability of a short distance via the probability of a small norm of the random c_2, $\Pr\big[\|c_2\| \leq \sqrt{2}\beta\big]$. □

M-S-UEO security of FALCON-PS-3. We can now show the M-S-UEO security of FALCON-PS-3.

Proposition 14 (M-S-UEO)**.** *Assuming* H *is a random oracle, for any adversary $\mathcal{A}$ against* M-S-UEO *security of* FALCON-PS-3 *that makes q_{H} queries to the random oracle, the advantage satisfies*

$$\boldsymbol{Adv}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{FALCON\text{-}PS\text{-}3},\mathcal{A}}(n) \leq (q_{\mathsf{H}} + 2)^2 \cdot 2^{(5-k)\frac{n}{2}}.$$

For the two parameter sets of FALCON*, the bounds are thus $(q_{\mathsf{H}} + 2)^2 \cdot 2^{-1024}$ for security level I and $(q_{\mathsf{H}} + 2)^2 \cdot 2^{-2560}$ for security level V, respectively.*

Proof. We assume that the adversary queries the hash function about its two signature output values before terminating; if not we first make these two queries, increasing the number from q_{H} to at most $q_{\mathsf{H}} + 2$ random oracle queries. In total, the adversary makes $q_{\mathsf{H}} + 2$ queries of the form (r_i, h_i, m_i), where r_i is a random salt, $h_i \in \mathbb{Z}[x]/(q, \phi)$ is a public key, and m_i is a message, to receive uniformly random c_i. An output $(h_i, h_j, m_i, m_j, (r_i, s))$ of an adversary against $\mathsf{M\text{-}S\text{-}UEO}$ can be valid, only if $h_i \neq h_j$, $r_i = r_j$, and there exists an $s \in \mathbb{Z}[x]/(q, \phi)$ with $\|(c_\ell - sh_\ell, s)\| \leq \beta$ for $\ell \in i, j$.

We show that with overwhelming probability, no such s exists. Thus, the advantage of the adversary is in the first place bounded by its chance to find an instance of (r_i, m_i, h_i) and (r_j, m_j, h_j) such that for the resulting c_i and c_j, there *exists* a single s that validates both c_i under h_i and c_j under h_j.[4]

First, note that $\|(c_\ell - sh_\ell, s)\| \leq \beta$ implies that $\|(c_i - sh_i, c_j - sh_j)\| \leq \sqrt{2}\beta$. In other words,

$$\mathsf{dist}(\Lambda_{h_i,h_j}, (c_i, c_j)) \leq \sqrt{2}\beta.$$

As $c_i, c_j \in \mathbb{Z}[x]/(q, \phi)$ are uniformly random, we apply Lemma 12 to see that the probability that this happens is bounded from above by $2^{(5-k)\frac{n}{2}}$. As there are (less than) $(q_{\mathsf{H}} + 2)^2$ pairs of distinct public keys, which can be tested by the adversary, we have the claimed result. $\square$

S-CEO and S-DEO of FALCON-PS-3 for Quantum Random Oracles. We can lift the above lower bound to the quantum case as well, exploiting known lower bounds for searching unstructured databases. But instead of showing the stronger notion of M-S-UEO we will revert to S-CEO and S-DEO. Assume that the adversary has made all q_{Sign} classical signature queries for the S-CEO or S-DEO attack upfront, resulting in signatures $\sigma_i = (r_i, s_i)$ for messages m_i. Making signature queries later can only decrease the adversary's success probability, as it limits the number of valid entries in the database. Define the function $f(h')$ for S-CEO (resp. $f(h', m')$ for S-DEO for some bounded message length, e.g., given through the run time of the adversary in question) to be 1 if there exists $i \in \{1, 2, \ldots, q_{\mathsf{Sign}}\}$ such that applying $\mathsf{H}(r_i \| h' \| m_i)$ resp. $\mathsf{H}(r_i \| h' \| m')$ yields a valid signature with sufficiently small $s'_{i,1} \leftarrow c'_i - s_{i,2}h'$ (and where, on top, $m' \neq m_i$ for S-DEO). Then, the goal of the adversary in the quantum random oracle model is to find an input h' resp. (h', m') making f evaluate to 1.

Note that for any fixed signature σ_i with hash value $c_i = \mathsf{H}(r_i \| h \| m_i)$, the hash value $c' = \mathsf{H}(r_i \| h' \| m_i)$ (resp. $c' = \mathsf{H}(r_i \| h' \| m')$ for S-DEO) is uniformly and independently distributed, if keys h, h' are different. Therefore, for each input h' resp. h', m' to function f the associated hash value c' allows for a valid signature under both keys with probability at most $2^{(5-k)n/2}$ by Corollary 13.

[4] If such an s exists, then the adversary indeed may have a good chance to mount a successful attack: the public keys may be chosen so that a trapdoor of Λ_{h_i,h_j} is known, and if a solution exists, the trapdoor may allow to compute a solution. The proof shows that under the randomness of the hash function, which the adversary cannot control, it is infeasible to find such instances in the first place.

Put differently, for each fixed signature there is a fraction of at most $2^{(5-k)n/2}$ challenges making f evaluate to 1. The number q_{Sign} of signatures increases this number by a factor q_{Sign}.

We finally need to relate the number of bad hash values to the number of bad inputs. Here, bad values refer to values which lead to a sufficiently small hash value. Let H be the size of the hash space, and N be the size of the input space of h' resp. h', m'. Note that for S-CEO the two sizes coincide because both public keys as well as hashes are from $\mathbb{Z}[x]/(q, \phi)$. For S-DEO, however, N is larger and we need to estimate how many inputs may hit a bad hash value. Observe that for each h', m' the hash value is independently distributed and hits a bad hash value with probability at most $q_{\mathsf{Sign}} \cdot 2^{(5-k)n/2}$. Then the probability that all N inputs h', m' hit twice as often such bad hash values is by the Chernoff bound[5] at most $\exp(-pN/3)$, where $p = q_{\mathsf{Sign}} \cdot 2^{(5-k)n/2}$. Hence, except with this double-exponentially small probability, we can assume (in both cases) that the number of inputs h' resp. h', m' hitting a bad hash value is at most $2q_{\mathsf{Sign}} \cdot 2^{(5-k)n/2}$. Furthermore, valid solutions are located at random positions by the randomness of H.

We can now apply the lower bound of Boyer et al. [3] for search algorithms. This bound states that, if there are t solutions with $f(x) = 1$, then the expected number of oracle evaluations of f to find a valid input x with probability ϵ, is at least $\Omega(\sqrt{\epsilon N/t})$ for the search space N. In our case we have N describe the cardinality of the set of admissible inputs h' resp. h', m', with f having at most $t \leq 2q_{\mathsf{Sign}} \cdot 2^{(5-k)n/2} \cdot N$ inputs evaluating to 1, except with exponentially small probability.

Proposition 15 (S-CEO and S-DEO for Quantum Random Oracles). *If we assume that* H *is a quantum random oracle, then the probability of a quantum adversary breaking* S-CEO *(resp.* S-DEO*) for* FALCON-PS-3 *with* q_{H} *superposition random oracle queries and at most* q_{Sign} *classical signature queries is at most*

$$\boldsymbol{Adv}^{\mathsf{S\text{-}CEO}}_{\mathsf{FALCON\text{-}PS\text{-}3},\mathcal{A}}(\mathcal{A}), \boldsymbol{Adv}^{\mathsf{S\text{-}DEO}}_{\mathsf{FALCON\text{-}PS\text{-}3},\mathcal{A}}(n) \leq O\left(q_{\mathsf{H}}^2 \cdot q_{\mathsf{Sign}} \cdot 2^{(k-5)n/2}\right).$$

For the two parameter sets of FALCON*, the bounds are thus* $O\left(q_{\mathsf{H}}^2 \cdot q_{\mathsf{Sign}} \cdot 2^{-1024}\right)$ *for security level I and* $O\left(q_{\mathsf{H}}^2 \cdot q_{\mathsf{Sign}} \cdot 2^{-2560}\right)$ *for security level V, respectively, with small constants in the O-notation.*

Proof. Assume that the hash function H behaves well in the sense that the number of possible solutions t in the algorithm of Boyer et al. [3] is bounded by $t \leq 2q_{\mathsf{Sign}} \cdot 2^{(5-k)n/2} \cdot N$, where $N \geq q^n$. The probability that this does not hold is at most

$$\exp\left(-q_{\mathsf{Sign}} \cdot 2^{(k-5)n/2} \cdot N/3\right) \leq O\left(q_{\mathsf{Sign}} \cdot 2^{(k-5)n/2}\right),$$

[5] For k independent Bernoulli variables X_i with $\Pr[X_i] = p$ and $\delta \in (0, 1]$ it holds $\Pr\left[\sum X_i \geq (1+\delta)pk\right] \leq \exp(-\delta^2 pk/3)$.

where we used a very generous estimate for the upper bound. If this is not the case, then we get an upper bound for the advantage ϵ of the adversary via the lower bound for the number of oracle queries:

$$q_{\mathsf{H}}^2 \geq \Omega\left(\epsilon \cdot q_{\mathsf{Sign}} \cdot 2^{(k-5)n/2}\right).$$

We therefore get for the adversary's success probability that, either H is not well-balanced with probability at most $O\left(q_{\mathsf{Sign}} \cdot 2^{(k-5)n/2}\right)$, or if it is, then $\epsilon \leq O\left(q_{\mathsf{H}}^2 \cdot q_{\mathsf{Sign}} \cdot 2^{(k-5)n/2}\right)$ by the search lower bound. Putting the two bounds together and subsuming the constants in the O-notation we derive the claim. $\square$

MBS security of FALCON-PS-3. The MBS security of FALCON-PS-3 translates immediately to the MBS security of FALCON, if one views the additional input h to the hashing step as part of the message $m' = h\|m$. Note that near-collision resistance is a standard model assumption, so that the following result applies to quantum adversaries.

Proposition 16 (MBS). *Assuming near-collision resistance (cf. [4, Assumption V.2]) of the hash function* H*,* FALCON-PS-3 *satisfies* MBS*.*

Using Lemma 11 we can give another proof of MBS which does not use the near-collision resistance assumption, but is in the random oracle model instead.

Proposition 17 (MBS, Random Oracle Version). *Assuming* H *is a random oracle, for any adversary $\mathcal{A}$ that makes q_{H} queries to the random oracle, we have*

$$\boldsymbol{Adv}^{\mathsf{MBS}}_{\mathsf{FALCON\text{-}PS\text{-}3},\mathcal{A}}(n) \leq (q_{\mathsf{H}} + 2)^2 \cdot 2^{(5-k)\frac{n}{2}}.$$

Proof. We set $\Lambda_{h,h}$ the rank n lattice in $\mathbb{R}^{2n}$ as defined in earlier. If $(\mathsf{pk}, m_1, m_2, \sigma)$ with $h \leftarrow \mathsf{pk}$, $(r, s) \leftarrow \sigma$, is a successful attack against MBS of FALCON-PS-3, by setting $c_\ell = \mathsf{H}(r\|h\|m_\ell)$ for $\ell \in \{1, 2\}$, we find that $\mathsf{dist}(\Lambda_{h,h}, (c_1, c_2)) \leq \sqrt{2}\beta$. Note that as $m_1 \neq m_2$, the values c_1 and c_2 are independent and uniformly random. By Lemma 12, the probability that $\mathsf{dist}(\Lambda_{h,h}, (c_1, c_2)) \leq \sqrt{2}\beta$ holds is less than $2^{(5-k)\frac{n}{2}}$.

Now, let $\mathcal{A}$ be an adversary making q_{H} queries. By increasing this value by 2, we may assume that $\mathcal{A}$ has queried $r\|h\|m_1$ and $r\|h\|m_2$, where $(h, m_1, m_2, (r, s))$ is its final output. We let $r_\ell\|h_\ell\|m_\ell$ denote all queries of $\mathcal{A}$ with hash values c_ℓ. Then, any c_i and c_j can only be part of a successful MBS attack if $h_i = h_j$, $r_i = r_j$, and $m_i \neq m_j$. Note that then c_i and c_j are independent and uniformly random. As there are at most $(q_{\mathsf{H}} + 2)^2$ such pairs, and for each the probability that there exists a valid s is bounded as shown above, we conclude the result. $\square$

NR security of FALCON-PS-3. Finally we show that FALCON-PS-3 satisfies non-resignability in the random oracle model, as long as the challenge sampler $\mathcal{D}$ in the game produces an output $(m, \mathsf{aux}) \leftarrow_{\$} \mathcal{D}(\mathsf{pk})$ where the message part m is unpredictable, given aux and the key pair. We note that the NR property can be shown along the lines of [6] since the hash value $\mathsf{H}(r\|h\|m)$ in FALCON-PS-3 is salted. However, the security proof in [6] in the ROM and QROM does not consider auxiliary data and is carried out indirectly via non-malleability. The former can be patched via computationally indistinguishability, and we can give a more direct proof without having to revert to non-malleability. We outline this in the classical ROM setting:

Proposition 18 (Non-Resignability). *Assuming* H *is a random oracle, the probability of an adversary $(\mathcal{A}, \mathcal{D})$, where $\mathcal{D}$ is unpredictable with computationally-independent auxiliary data with respect to* KGen *of* FALCON-PS-3*, making at most $q_{\mathcal{D}}$ random oracle queries, breaking* NR *of* FALCON-PS-3 *with* q_{H} *random oracle queries is at most*

$$\begin{aligned}\boldsymbol{Adv}^{\mathsf{NR}}_{\mathsf{FALCON\text{-}PS\text{-}3},\mathcal{A},\mathcal{D}}(n) \leq 2 \cdot \boldsymbol{Adv}^{\mathsf{CI\text{-}aux}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{B}}(n) + (q_{\mathsf{H}} + q_{\mathcal{D}}) \cdot \boldsymbol{Adv}^{\mathsf{pred}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{C}}(n) \\ + (q_{\mathsf{H}} + q_{\mathcal{D}}) 2^{(5-k)\frac{n}{2}}.\end{aligned}$$

where $\mathcal{B}$ and $\mathcal{C}$ have roughly the same running time as $\mathcal{A}$.

Proof. An attacker against NR of FALCON-PS-3 is given a public key pk, a signature $\sigma \leftarrow (r, s)$ that verifies under this public key pk for a message m (that is supposedly largely unknown to the attacker), as well as circumstantial knowledge aux about the message. We argue that it is infeasible for an adversary, given σ, pk, and aux to find a public key pk' and a signature σ' which verifies for the message m.

We proceed via game hopping. Let Game_0 be the original attack of $\mathcal{A}$ against the scheme. We denote by $\Pr[\mathsf{Game}_i]$ the probability that the ith game returns 1. In particular, $\Pr[\mathsf{Game}_0]$ then denotes $\mathcal{A}$'s success probability in the attack.

In the hop to game Game_1 we simply let the game execute another sampling step for distribution $\mathcal{D}$ for the given public key pk. For sake of distinction we denote this sampling step as $(\overline{m}, \overline{\mathsf{aux}}) \leftarrow\$ \overline{\mathcal{D}}(\mathsf{pk})$, although $\overline{\mathcal{D}}$ is the same algorithm as $\mathcal{D}$. The rest of the game remains unchanged. It obviously follows that $\Pr[\mathsf{Game}_0] = \Pr[\mathsf{Game}_1]$.

In the hop to game Game_2 we replace the auxiliary information aux when running $\mathcal{A}$ in Game_1 by the sample $\overline{\mathsf{aux}}$ picked independently as $(\overline{m}, \overline{\mathsf{aux}}) \leftarrow\$ \overline{\mathcal{D}}(\mathsf{pk})$ in the extra sampling step introduced in the previous game. That is, in Game_2 we measure the probability of $\mathcal{A}$ winning the NR experiment when receiving $\overline{\mathsf{aux}}$ instead of aux. By the computational indistinguishability this cannot decrease the success probability significantly. Specifically, we can build an adversary $\mathcal{B}$ against the indistinguishability of the sampler $\mathcal{D}$ running the entire experiment for the given input data (m, aux_b) where $\mathsf{aux}_b = \mathsf{aux}$ or $\mathsf{aux}_b = \overline{\mathsf{aux}}$. In particular, algorithm $\mathcal{B}$ receives the key pair $(\mathsf{pk}, \mathsf{sk})$ according to the scheme and the challenger's input (m, aux_b), creates a signature σ for m with sk, and runs $\mathcal{A}$ on input $(\mathsf{pk}, \sigma, \mathsf{aux}_b)$. All oracle queries of $\mathcal{A}$ to H are forwarded to $\mathcal{B}$'s oracle H. Eventually, $\mathcal{B}$ verifies if $\mathcal{A}$ succeeded in the non-resignability game. This is possible since $\mathcal{B}$ knows the message m.

If $\mathsf{aux}_b = \mathsf{aux}$ then the simulation of $\mathcal{B}$ corresponds to Game_1. If $\mathsf{aux}_b = \overline{\mathsf{aux}}$ then it corresponds to Game_2. Taking into account a factor of 2 for switching to a guessing experiment we obtain:

$$\Pr[\mathsf{Game}_1] \leq \Pr[\mathsf{Game}_2] + 2 \cdot \mathbf{Adv}^{\mathsf{CI\text{-}aux}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{B}}(n).$$

In the third hop to Game_3 we declare $\mathcal{A}$ to lose if it ever makes a random oracle query to H about the message $*\|*\|m$ sampled by the challenger $\mathcal{D}$ in game Game_2, or if $\overline{\mathcal{D}}$ does so when sampling the independent value $(\overline{m}, \overline{\mathsf{aux}})$ in the second execution. We remark that we can recover the message part form a query to H via the expected encoding of sound queries, i.e., expecting a leading nonce of 320 bits and fixed-length encoding of the key. We can bound the probability of $\mathcal{A}$ or $\overline{\mathcal{D}}$ making such a query in Game_2 by the (un)predictability of the sampler $\mathcal{D}(\mathsf{pk})$. For this we construct a predictor $\mathcal{C}(\mathsf{pk}, \mathsf{sk}, \mathsf{aux})$, receiving a key pair $(\mathsf{pk}, \mathsf{sk})$ generated by KGen and auxiliary data aux generated by $\mathcal{D}(\mathsf{pk})$, but not the message m. We let $\mathcal{C}$ first pick an index i between 1 and $q_{\mathsf{H}} + q_{\mathcal{D}}$ at the beginning. Then $\mathcal{C}$ samples $(\overline{m}, \overline{\mathsf{aux}}) \leftarrow_{\$} \overline{\mathcal{D}}(\mathsf{pk})$ according to the game to create some auxiliary input, forwarding random oracle queries of $\overline{\mathcal{D}}$ to its own oracle. Next, it generates a challenge signature σ with the help of the input sk, but using a random c instead of the actual hash value $\mathsf{H}(r\|\mathsf{pk}\|m)$ for the unknown message m. It finally runs $\mathcal{A}$ on input $(\mathsf{pk}, \sigma, \overline{\mathsf{aux}})$, relaying random oracle queries of $\mathcal{A}$ to its own random oracle. In both simulations, if either $\overline{\mathcal{D}}$ or $\mathcal{A}$ makes the ith query $*\| * \|m_i$ to the random oracle, counting queries of both algorithms together, algorithm $\mathcal{C}$ outputs the message-entry m_i in this query as a prediction.

Note that the probability of $\overline{\mathcal{D}}$ or $\mathcal{A}$ making a query about m to H in Game_2 is identical to the one in Game_3 up to the point where $\overline{\mathcal{D}}$ or $\mathcal{A}$ makes such a query (and possibly spots that c used in the simulated signature does not match the correct H-value). Therefore, algorithm $\mathcal{C}$ guesses the first of such queries (if it exists) with probability $1/(q_{\mathsf{H}} + q_{\mathcal{D}})$ and in this case predicts m correctly if $\overline{\mathcal{D}}$ or $\mathcal{A}$ would pose such a query. We can thus bound this event from above by $(q_{\mathsf{H}} + q_{\mathcal{D}}) \cdot \mathbf{Adv}^{\mathsf{pred}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{C}}(n)$. It follows that

$$\Pr[\mathsf{Game}_2] \leq \Pr[\mathsf{Game}_3] + (q_{\mathsf{H}} + q_{\mathcal{D}}) \cdot \mathbf{Adv}^{\mathsf{pred}}_{\mathcal{D},\mathsf{H},\mathsf{KGen},\mathcal{C}}(n).$$

Note that now adversary $\mathcal{A}$ loses if it queries the random oracle H about input $*\| * \|m$ according to Game_3, and $\overline{\mathsf{aux}}$ is independently distributed of $\mathsf{H}(*\| * \|m)$ since $\overline{\mathcal{D}}$ never makes a query about m to its random oracle. Furthermore, in $\mathsf{FALCON\text{-}PS\text{-}3}$, key generation does not query the random oracle to compute the key pair $(\mathsf{pk}, \mathsf{sk})$. Thus, the only information about hash values involving m is (implicitly) via the challenge signature σ but for public key $\mathsf{pk} = h$, i.e., about $\mathsf{H}(r\|\mathsf{pk}\|m)$. Yet, the adversary needs to succeed for a different public key $\mathsf{pk}' = h' \neq \mathsf{pk} = h$. Hence, we can imagine to sample the hash value $\mathsf{H}(r'\|\mathsf{pk}'\|m)$ for verifying $\mathcal{A}$'s signature $\sigma' = (r', s')$ for the public key $\mathsf{pk}' = h'$ only *after* $\mathcal{A}$ has output the values pk', σ'. As $c' = \mathsf{H}(r'\|\mathsf{pk}'\|m)$ is uniformly distributed, the same holds for $c' - s'h'$ with s' and h' being determined. By Lemma 11, the probability that the norm of $c' - s'h'$ is bounded by β is less than $2^{(5-k)\frac{n}{2}}$. This shows the proposition. $\square$

Remark 19. Note that the first game hops in the proof of Proposition 18 are not tied to FALCON-PS-3 in particular. Rather, equivalence of wNR without auxiliary data, as formalized in [2], and NR with computationally-independent auxiliary data appears to hold for any signature scheme. Moreover, this remains true for the quantum setting, where the potential distinguisher $\mathcal{B}$ has quantum access to the random oracle, then we merely need to work with a notion of quantum indistinguishability.

Then, observe that in the classical setting the fact that neither the message sampler $\overline{\mathcal{D}}$ nor the adversary $\mathcal{A}$ queries the random oracle about the unknown message, due to unpredictability, does not use specifics of FALCON-PS-3, but also holds in general.

In FALCON-PS-3 the first game hops allow us to conclude that the adversary produces an output which is independent of the hash value of m, also because key generation in FALCON-PS-3 does not query the random oracle at all. A technical trick for general schemes, where KGen calls the random oracle, could resolve this part as well: One may formally put all hash queries made during key generation into the secret key sk, and then let our guessing adversary $\mathcal{C}$ —receiving sk as input— also potentially pick from this query list, such that one can conclude that key generation neither queries about m.

5 Conclusion

Our analysis shows that, under reasonable assumptions about the hash function, for FALCON it suffices to apply the PS-3 transform to achieve all beyond-unforgeability features. In comparison to the full BUFF transform, the PS-3 transform does not increase the signature size. From our point of view, this suggests to consider FALCON-PS-3 as a preferred candidate for standardization of a BUFFed version of FALCON.

6 Acknowledgements

S.D. was funded by the Deutsche Forschungsgemeinschaft (DFG) – SFB 1119 – 236615297. R.F. was supported by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE. Finally, S.D. is thankful to Patrick Struck for his helpful comments on early versions of this work.

References

1. Gorjan Alagic, David Cooper, Quynh Dang, Thinh Dang, John M. Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl A. Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, and Daniel Apon. Status report on the third round of the nist post-quantum cryptography standardization process, 07 2022. 1

2. Thomas Aulbach, Samed Düzlü, Michael Meyer, Patrick Struck, and Maximiliane Weishäupl. Hash your keys before signing: Buff security of the additional nist pqc signatures. Cryptology ePrint Archive, Paper 2024/591, 2024. `https://eprint.iacr.org/2024/591`. 3, 7, 20
3. Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4–5):493–505, June 1998. 16
4. 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 (SP)*, 2021. 1, 2, 3, 6, 7, 9, 17
5. Cas Cremers, Samed Düzlü, Rune Fiedler, Marc Fischlin, and Christian Janson. Buffing signature schemes beyond unforgeability and the case of post-quantum signatures. Cryptology ePrint Archive, Paper 2020/1525, 2023. `https://eprint.iacr.org/2020/1525`, version 1.4. 7
6. Jelle Don, Serge Fehr, Yu-Hsuan Huang, and Patrick Struck. On the (in)security of the buff transform. Cryptology ePrint Archive, Paper 2023/1634, 2023. `https://eprint.iacr.org/2023/1634`. 3, 7, 18
7. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008. 2, 9
8. Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In Moni Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 169–186. Springer, Heidelberg, May 2007. 7
9. 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, 2022. available at `https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022`. 1
10. 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, 2022. available at `https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022`. 1
11. NIST. Post-quantum cryptography. `https://csrc.nist.gov/projects/post-quantum-cryptography-standardization`, 01 2017. 1
12. NIST. Call for additional digital signature schemes for the post-quantum cryptography standardization process, 09 2022. 1
13. NIST. Module-lattice-based digital signature standard, August 2023. FIPS 204 (draft). `https://doi.org/10.6028/NIST.FIPS.204.ipd`. 1
14. NIST. Stateless hash-based digital signature standard, August 2023. FIPS 205 (draft). `https://doi.org/10.6028/NIST.FIPS.205.ipd`. 1
15. 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. 1, 2, 9
16. 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, 2022. available at `https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022`. 1, 9