Hash your Keys before Signing
BUFF Security of the Additional NIST PQC Signatures

Thomas Aulbach[1], Samed Düzlü[1], Michael Meyer[1], Patrick Struck[2], and Maximiliane Weishäupl[1]

[1] Universität Regensburg
{thomas.aulbach,samed.duzlu,maximiliane.weishaeupl}@ur.de
michael@random-oracles.org
[2] Universität Konstanz
patrick.struck@uni-konstanz.de

Abstract. In this work, we analyze the so-called Beyond UnForgeability Features (BUFF) security of the submissions to the current standardization process of additional signatures by NIST. The BUFF notions formalize security against maliciously generated keys and have various real-world use cases, where security can be guaranteed despite misuse potential on a protocol level. Consequently, NIST declared the security against the BUFF notions as *desirable features*. Despite NIST's interest, only 6 out of 40 schemes consider BUFF security at all, but none give a detailed analysis. We close this gap by analyzing the schemes based on codes, isogenies, lattices, and multivariate equations. The results vary from schemes that achieve neither notion (e.g., Wave) to schemes that achieve all notions (e.g., Prov). In particular, we dispute certain claims by Squirrels and Vox regarding their BUFF security. Resulting from our analysis, we observe that two schemes (Cross and Prov) achieve BUFF security without having the hash of public key and message as part of the signature, as BUFF transformed schemes would have. Prov essentially uses the lighter PS-3 transform by Pornin and Stern (ACNS'05). We further point out whether this transform suffices for the other schemes to achieve the BUFF notions, with both positive and negative results.

Keywords: Signature Schemes · BUFF · Additional Security Properties

⋆ **Errata:** The proceedings version of this paper published at PQCrypto 2024 contains an incorrect analysis of the lattice-based signature scheme Hawk [9]. The issue was identified in [17], which also provides a detailed description of the underlying mistakes. More recently, a severe attack against Hawk was discovered [38], prompting the Hawk team to withdraw the scheme from the NIST standardization process (https://csrc.nist.gov/Projects/pqc-dig-sig/round-3-additional-signatures). Given these developments, we have decided to remove the Hawk analysis entirely from our paper. The results for all other signature schemes considered in the paper remain unaffected.

1 Introduction

Nowadays, digital signature schemes are fundamental cryptographic primitives. They allow a signer Alice to generate a signature $\mathtt{sig}$ of a message $\mathtt{msg}$, using her private key $\mathtt{sk}$, such that anybody, using Alice's public key $\mathtt{pk}$, can verify the validity of the signature. *Existential unforgeability under chosen message attacks* (EUF-CMA) has become the standard security notion for digital signature schemes. EUF-CMA secure schemes come with the guarantee that an adversary, seeing several message-signature pairs generated by Alice, cannot generate a new message-signature pair that is accepted as a signature by Alice.

Unforgeability is essential for digital signature schemes and in most use cases also sufficient. However, the EUF-CMA security notion only covers scenarios where Alice's key pair is honestly generated. Depending on the use case of a digital signature scheme, other attacks are possible which are not ruled out by using a signature scheme that is unforgeable. This led to the development of additional security notions: *exclusive ownership*, *message-bound signatures*, and *non-resignability*. In the following, we give high-level descriptions of these notions, covering the gist of each.

The first security notion, *exclusive ownership*, provides the adversary with a valid message-signature pair $(\mathtt{msg}, \mathtt{sig})$ under a public key $\mathtt{pk}$ and asks it to find a different public key $\overline{\mathtt{pk}}$ under which $(\mathtt{msg}, \mathtt{sig})$ remains a valid message-signature pair. The lack of exclusive ownership allows an adversary to "claim" signatures as its own by providing $\overline{\mathtt{pk}}$. The relevance can be seen by the real-world attack against the Let's Encrypt protocol, where an adversary can exploit such claimed signatures to obtain certificates for domains despite not owning them [1]. The notion comes in two flavors: the one just described, which is called *conservative exclusive ownership* (S-CEO), and *destructive exclusive ownership* (S-DEO), where the adversary needs to find a different message.

The second security notion, *message-bound signatures* (MBS), asks the adversary to come up with two messages $\mathtt{msg} \neq \overline{\mathtt{msg}}$, a signature $\mathtt{sig}$, and a public key $\mathtt{pk}$, such that both $(\mathtt{msg}, \mathtt{sig})$ and $(\overline{\mathtt{msg}}, \mathtt{sig})$ are valid message-signature pairs under $\mathtt{pk}$. Absence of this property allows adversaries to bypass non-repudiation: when the adversary is accused of having signed $\mathtt{msg}$, it can claim to have signed $\overline{\mathtt{msg}}$ instead. At the first glance, it seems that this should already be covered by standard EUF-CMA—finding $\overline{\mathtt{msg}}$ immediately yields a forged signature. The difference is that EUF-CMA is limited to honestly generated keys whereas the notion we describe here is more permissive by letting the adversary output an arbitrary public key, in particular, not constrained to be the outcome of the key generation algorithm.

The third security notion, *non-resignability* (NR), provides the adversary with a signature $\mathtt{sig}$ of an unknown message $\mathtt{msg}$ under some public key $\mathtt{pk}$ and asks the adversary for a different public key $\overline{\mathtt{pk}}$ and signature $\overline{\mathtt{sig}}$, such that $\overline{\mathtt{sig}}$ verifies correctly under $\overline{\mathtt{pk}}$ for the unknown[3] message $\mathtt{msg}$. Jackson et al. [28]

[3] This part is crucial. If the adversary was to know the message $\mathtt{msg}$, it could generate a new key pair $(\overline{\mathtt{sk}}, \overline{\mathtt{pk}})$ and sign $\mathtt{msg}$ using $\overline{\mathtt{sk}}$ to obtain $\overline{\mathtt{sig}}$ and output $(\overline{\mathtt{pk}}, \overline{\mathtt{sig}})$.

showed that a resignable signature scheme, i.e., one for which the adversary can find $\overline{\texttt{pk}}$ and $\overline{\texttt{sig}}$ as described above, allows for attacks against the "Dynamically Recreatable Key" (DRKey) protocol [29]. Here, the adversary has to re-sign a message which contains a—to the adversary unknown—symmetric key.

The additional security properties exclusive ownership, message-bound signatures, and non-resignability were formalized in [16], which also provides a generic transformation—called the BUFF transform—to achieve them. Furthermore, the authors of [16] analyzed the signature schemes selected to be standardized by NIST: Dilithium [31], Falcon [37], and Sphincs$^+$ [27]. Dilithium was shown to achieve the notions and while Falcon does not, the authors of Falcon announced to deploy the BUFF transform in the next update. For Sphincs$^+$ it is informally argued that it achieves the additional security properties. While the notions are not a requirement in the ongoing NIST standardization process for digital signature schemes [33], the call-for-algorithms mentions them as "additional desirable security properties beyond standard unforgeability". Despite this, only six out of 40 submissions mention these security properties at all, but none give a detailed analysis. Thus, there is a gap with respect to the security achieved by the signature schemes submitted to the NIST standardization process. A gap that we (partially) close in this work.

A Note on Non-Resignability. Note that the initial definition of non-resignability in [16] was identified to be flawed in [22]. The problem lies in the auxiliary information which allowed for an (arguably artificial) attack. New proposals for the definition of non-resignability are given in [22] and an updated version of [16]. However, it is unclear which definition will ultimately define non-resignability, and if the BUFF transform achieves either notion. Given these problems, we opt for a weaker form of non-resignability (wNR) in which there is no auxiliary information—thus considering a weaker notion than the one introduced in [22]. Nevertheless, we provide concrete attacks against most schemes. Thus, they are also vulnerable to any stronger form of non-resignability, in particular, to the existing ones [16,22].

1.1 Our Contribution

We analyze the submissions to the NIST standardization process for post-quantum signatures [33]. We focus on the submissions that are based on either codes, isogenies, lattices, or multivariate equations—excluding those for which attacks against EUF-CMA have been identified. More precisely, we analyze four code-based schemes (Cross [3], Less [2], Meds [13], Wave [4]), the sole isogeny-based scheme (SQIsign [11]), four lattice-based schemes (Haetae [12], HuFu [40], Raccoon [20], Squirrels [23]), and seven multivariate schemes (Mayo [6], Prov [26], Qr-Uov [24], Snova [39], Tuov [21], Uov [7], Vox [35]). The results are summarized in Table 1.

In the following, we describe the main results. First, we remark that MBS is almost always satisfied and the security can be traced back to the security of the hash function. In the two cases of Squirrels and Wave, where MBS is

not satisfied, the reason is the scheme-dependent construction of a public key that allows multiple messages to verify under the same signature. Note that the specification of Squirrels claims MBS security, which our analysis refutes.

Secondly, we note that all schemes—except for SQIsign, Meds and Less—satisfy either both S-CEO and S-DEO, or neither. Despite the general separation by [16], our results indicate that in practice, these two notions often behave similarly. In fact, both proofs and attacks usually use the same idea for S-CEO and S-DEO, where for S-DEO, one needs to be slightly more careful in the choices. One group among the schemes that satisfy these exclusive ownership notions achieves them by hashing the public key, together with the message, to generate a target (resp. challenge), which the signature of the message corresponding to the given public key solves. In this way, any modification of the public key uncontrollably changes the target in a random manner. Then, the signature, which is required to be the same as the given one, cannot solve the new target, hence rendering the scheme secure. All schemes that do not satisfy exclusive ownership security are attacked by explicitly constructing new public keys which are compatible with the target generated independently of the public key, and the given signature. Differences between S-CEO and S-DEO can arise, when the message, but not the public key, is used to derive the target. Then it depends on the inherent properties of the scheme if different public keys can be constructed for the same (S-CEO) or a new (S-DEO) target value. An exception to the above rule is Cross, where the security reduces to solving an underdetermined system of linear equations.

Finally, we consider non-resignability. All schemes that satisfy wNR are also secure with respect to both exclusive ownership and MBS. However, there are schemes (SQIsign, Meds and Less), that satify S-DEO but not wNR. Indeed, we see a relationship to their exclusive ownership security: While fixing a signature fixes the public key in a certain sense, one can attack non-resignability by modifying both in a compatible manner, which does not require any knowledge about the message being signed. For the schemes that satisfy wNR, we see a similar argument as for exclusive ownership, namely that producing the target using a hash of the public key and the message, makes the target untraceable, even if one can control the signature. The exception, again, is Cross, where the security results from the Merkle tree structure and an underdetermined system of linear equations. The other schemes that do not satisfy wNR are attacked, as in the case of exclusive ownership, by explicit constructions. Neither of those attacks rely on any auxiliary information about the unknown message, which an adversary is provided in stronger versions of non-resignability.

From our results, we can deduce the following interesting conjecture. Even though [16] shows that in general the BUFF transform is necessary to achieve full BUFF security, it turns out that in practice, it is most often sufficient to use the PS-3 transform as suggested in [36]. That means, instead of using a mere hash-and-sign paradigm, one needs to hash the message *and* the public key, and then sign the hash value. The PS-3 transform is more lightweight than the BUFF transform as the latter requires to also append the hash value to the signature.

Table 1: Overview of our results. A ✓ indicates that a signature scheme achieves a security notion, while a ✗ indicates that there is an attack. A ✦ indicates that we identified an attack that seems not to be relevant in practice. A superscript † indicates that the result disproves a claim made for the scheme. For LESS and MEDS, the results for S-CEO depend on the parameter sets.

Scheme	S-CEO	S-DEO	MBS	wNR	Type
CROSS [3]	✓	✓	✓	✓	
LESS [2]	✓ | ✗	✓	✓	✗	Code (Sect. 3)
MEDS [13]	✓ | ✗	✓	✓ | ✗	✗	
WAVE [4]	✗	✗	✗	✗	
SQISIGN [11]	✦	✓	✗	✗	Isogeny (Sect. 4)
HAETAE [12]	✓	✓	✓	✓	
HUFU [40]	✗	✗	✓	✗	Lattice
RACCOON [20]	✓	✓	✓	✓	(Sect. 5)
SQUIRRELS [23]	✗	✗	✗†	✗	
MAYO [6]	✗	✗	✓	✗	
PROV [26]	✓	✓	✓	✓	
QR-UOV [24]	✗	✗	✓	✗	Multivariate
SNOVA [39]	✗	✗	✓	✗	(Sect. 6)
TUOV [21]	✗	✗	✓	✗	
UOV [7]	✗	✗	✓	✗	
VOX [35]	✗†	✗†	✓	✗†	

One important caveat in this regard is that it is often *not* sufficient to hash only a part of the public key. Important examples where such an approach does not help to satisfy BUFF security are given by various multivariate schemes, e.g., VOX, where this approach is used explicitly to gain BUFF security, but is not sufficient.

Structure of the Analyses. The analyses presented in this work follow a common structure, which we explain briefly. To analyze the BUFF security, the relevant information is the structure of the public key and signature, and the verification algorithm. Those are introduced at the beginning of each section, followed by the analysis of S-CEO, S-DEO, MBS, and wNR. In Section 6 on multivariate schemes, we give a more detailed general outline and give a generic proof of MBS and a generic attack on wNR, as the schemes allow such an all-encompassing formulation. The remaining analyses in the section follow the same structure.

1.2 Related Work

Unforgeability notions can be traced back to [25]. Exclusive ownership originates from [8,32], which introduces a specialized version under the name *Duplicate-Signature Key Selection*. A generalized version was developed in [36] which also

coins the term exclusive ownership. Non-resignability was first mentioned in [28] though without a formal definition. Eventually, formal definitions of all beyond unforgeability properties (exclusive ownership, message-bound signatures, and non-resignability) were developed in [16], which also gives two generic transformations to achieve them.

2 Preliminaries

2.1 Notation

For integers m, n with $m < n$, we write $[m]$ and $[m, n]$ for the sets $\{1, 2, \ldots, m\}$ and $\{m, m+1, \ldots, n\}$, respectively. Throughout this work, $\mathtt{H}$ will denote a hash function (optionally with a subscript if multiple hash functions are used) which is often modeled as a random oracle [5]. For a matrix M, we denote the entries by m_{ij}. Similarly, for a vector x_i, its entries are denoted by $x_{i,j}$. We use ϑ to denote a generic bound (used for the lattice-based schemes).

2.2 Signature Schemes and Security Notions

A signature scheme Σ consists of three efficient algorithms:

`KGen`: the key generation gets a security parameter 1^λ as input and outputs a secret key `sk` along with a public key `pk`.

`Sign`: the signing algorithm gets a secret key `sk` and a message `msg` as input and outputs a signature `sig`.

`Verify`: the verification algorithm takes as input a public key `pk`, a message `msg`, and a signature `sig`, and it outputs a bit `v`.

A signature scheme is correct if, for any key pair $(\mathtt{sk}, \mathtt{pk}) = \mathtt{KGen}(1^\lambda)$, we have $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{Sign}(\mathtt{sk}, \mathtt{msg})) = 1$ with overwhelming probability in the security parameter 1^λ.

In this work, we are using the security notions conservative/destructive exclusive ownership and message-bound signatures as formalized in [16], as well as a weaker form of non-resignability. Below we give the definitions. The corresponding security games S-CEO, S-DEO, MBS, and wNR, are shown in Fig. 1.

For conservative exclusive ownership, the adversary can obtain signatures for arbitrary messages and is then challenged to find a different public key that verifies one of the received message-signature pairs. Destructive exclusive ownership is similar to conservative exclusive ownership. The difference is that the adversary needs to find not just a different public key but also a different message that verify using one of the received signatures. The message-bound signature property guarantees that it is hard to find a signature that verifies two different messages under the same public key.

Definition 1. *A signature scheme* $\Sigma = (\mathtt{KGen}, \mathtt{Sign}, \mathtt{Verify})$ *is said to have* conservative exclusive ownership*, destructive exclusive ownership, and* message-bound signatures *if for any efficient adversary* $\mathcal{A}$*, its probability in winning game* S-CEO*,* S-DEO*, and* MBS*, respectively, is negligible.*

Game S-CEO/S-DEO

$\mathcal{Q} \leftarrow \emptyset$
$(\mathtt{sk}, \mathtt{pk}) \leftarrow \mathtt{KGen}()$
$(\overline{\mathtt{pk}}, \overline{\mathtt{msg}}, \overline{\mathtt{sig}}) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathtt{sk},\cdot)}(\mathtt{pk})$
$\mathtt{v}_1 \leftarrow \mathtt{Verify}(\overline{\mathtt{pk}}, \overline{\mathtt{msg}}, \overline{\mathtt{sig}})$
$\mathtt{v}_2 \leftarrow \mathsf{Valid}(\overline{\mathtt{msg}}, \overline{\mathtt{sig}})$
return $(\mathtt{v}_1 = 1 \wedge \mathtt{v}_2 = 1 \wedge \overline{\mathtt{pk}} \neq \mathtt{pk})$

Game wNR

$(\mathtt{sk}, \mathtt{pk}) \leftarrow \mathtt{KGen}()$
$\mathtt{msg} \leftarrow \mathcal{A}_0(\mathtt{pk})$
$\mathtt{sig} \leftarrow \mathtt{Sign}(\mathtt{sk}, \mathtt{msg})$
$(\overline{\mathtt{sig}}, \overline{\mathtt{pk}}) \leftarrow \mathcal{A}_1(\mathtt{pk}, \mathtt{sig})$
$\mathtt{v} \leftarrow \mathtt{Verify}(\overline{\mathtt{pk}}, \mathtt{msg}, \overline{\mathtt{sig}})$
return $(\overline{\mathtt{pk}} \neq \mathtt{pk} \wedge \mathtt{v} = 1)$

Game MBS

$(\mathtt{msg}, \overline{\mathtt{msg}}, \mathtt{sig}, \mathtt{pk}) \leftarrow \mathcal{A}()$
$\mathtt{v}_1 \leftarrow \mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig})$
$\mathtt{v}_2 \leftarrow \mathtt{Verify}(\mathtt{pk}, \overline{\mathtt{msg}}, \mathtt{sig})$
return $(\mathtt{msg} \neq \overline{\mathtt{msg}} \wedge \mathtt{v}_1 = 1 \wedge \mathtt{v}_2 = 1)$

Oracle $\mathsf{Sign}(\mathtt{sk}, \mathtt{msg})$

$\mathtt{sig} \leftarrow \mathtt{Sign}(\mathtt{sk}, \mathtt{msg})$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(\mathtt{msg}, \mathtt{sig})\}$
return sig

$\mathsf{Valid}(\overline{\mathtt{msg}}, \overline{\mathtt{sig}})$ in S-CEO

if $(\overline{\mathtt{msg}}, \overline{\mathtt{sig}}) \in \mathcal{Q}$
  return 1
return 0

$\mathsf{Valid}(\overline{\mathtt{msg}}, \overline{\mathtt{sig}})$ in S-DEO

if $\exists \mathtt{msg} \neq \overline{\mathtt{msg}}$ **s.t.** $(\mathtt{msg}, \overline{\mathtt{sig}}) \in \mathcal{Q}$
  return 1
return 0

Fig. 1: Security games S-CEO, S-DEO, MBS, and wNR, for signature schemes.

Non-resignability provides the adversary with a signature of an unknown message and asks to find a different public key and (not necessarily different) signature that verify the unknown message. We consider a slightly weaker form of non-resignability, which we call weak NR (wNR), which does not grant the adversary auxiliary information about the message. Note that for the majority of signature schemes we give attacks against wNR which are also valid attacks against any stronger form of non-resignability, in particular, those including auxiliary information for the adversary.

Definition 2. *A signature scheme* $\Sigma = (\mathtt{KGen}, \mathtt{Sign}, \mathtt{Verify})$ *is said to have* non-resignability *if for any efficient adversary* $(\mathcal{A}_0, \mathcal{A}_1)$*, where* $\mathcal{A}_0$ *outputs uniformly random message, its probability in winning game* wNR *is negligible.*

We say that a signature scheme Σ has full BUFF security, if it satisfies S-CEO, S-DEO, MBS, and wNR.

2.3 Transformations

There are several generic transformations that turn a signature scheme and a hash function into a signature scheme that achieves the aforementioned BUFF notions. For this work, we mainly need two: The BUFF transform [16] (cf. Fig. 3)

$\mathtt{KGen}^*()$	$\mathtt{Sign}^*(\mathtt{sk},\mathtt{msg})$	$\mathtt{Verify}^*(\mathtt{pk},\mathtt{msg},\mathtt{sig})$
$(\mathtt{sk},\mathtt{pk}) \leftarrow \mathtt{KGen}()$	$\mathtt{h} \leftarrow \mathtt{H}(\mathtt{msg},\mathtt{pk})$	$\overline{\mathtt{h}} \leftarrow \mathtt{H}(\mathtt{msg},\mathtt{pk})$
return $(\mathtt{sk},\mathtt{pk})$	$\mathtt{sig} \leftarrow \mathtt{Sign}(\mathtt{sk},\mathtt{h})$	$\mathtt{v} \leftarrow \mathtt{Verify}(\mathtt{pk},\overline{\mathtt{h}},\mathtt{sig})$
	return $\mathtt{sig}$	**return** $\mathtt{v} = 1$

Fig. 2: The signature scheme PS-3$[\mathtt{H},\Sigma] = (\mathtt{KGen}^*,\mathtt{Sign}^*,\mathtt{Verify}^*)$ constructed from a hash function $\mathtt{H}$ and a signature scheme $\Sigma = (\mathtt{KGen},\mathtt{Sign},\mathtt{Verify})$.

$\mathtt{KGen}^*()$	$\mathtt{Sign}^*(\mathtt{sk},\mathtt{msg})$	$\mathtt{Verify}^*(\mathtt{pk},\mathtt{msg},(\mathtt{sig},\mathtt{h}))$
$(\mathtt{sk},\mathtt{pk}) \leftarrow \mathtt{KGen}()$	$\mathtt{h} \leftarrow \mathtt{H}(\mathtt{msg},\mathtt{pk})$	$\overline{\mathtt{h}} \leftarrow \mathtt{H}(\mathtt{msg},\mathtt{pk})$
return $(\mathtt{sk},\mathtt{pk})$	$\mathtt{sig} \leftarrow \mathtt{Sign}(\mathtt{sk},\mathtt{h})$	$\mathtt{v} \leftarrow \mathtt{Verify}(\mathtt{pk},\overline{\mathtt{h}},\mathtt{sig})$
	return $(\mathtt{sig},\mathtt{h})$	**return** $(\mathtt{v} = 1 \wedge \mathtt{h} = \overline{\mathtt{h}})$

Fig. 3: The signature scheme BUFF$[\mathtt{H},\Sigma] = (\mathtt{KGen}^*,\mathtt{Sign}^*,\mathtt{Verify}^*)$ constructed from a hash function $\mathtt{H}$ and a signature scheme $\Sigma = (\mathtt{KGen},\mathtt{Sign},\mathtt{Verify})$.

and the PS-3 transform [36] (cf. Fig. 2). The former was shown to achieve all the BUFF notions—based on the assumptions on the used hash function. The latter was shown to not achieve all notions, due to a property that [16] calls *weak keys*, i.e., public keys that verify multiple messages. Both transformations work by first computing the hash of the public key and message. This hash value is then signed[4] by the signature scheme. The difference is that the BUFF transform additionally appends this hash value to the signature (which PS-3 does not).

3 Code-based schemes

In this section, we analyze the code-based signature schemes. They rely on two distinct code-related problems: the more classical syndrome decoding problem (CROSS and WAVE), and the fairly new code equivalence problem (MEDS and LESS). Although they are based on the same underlying problem, CROSS and WAVE are still very different, and while CROSS satisfies all BUFF properties, we show that WAVE is vulnerable with respect to each of the notions. WAVE fails to satisfy full BUFF security even after the PS-3 transform. We analyze CROSS in Section 3.1 and WAVE in Section 3.4. The two schemes based on code equivalences (MEDS and LESS) are very similar. We therefore only present MEDS in full detail (in Section 3.2), as the analysis of LESS (in Section 3.3) is almost verbatim the same. A surprising result of this analysis is that exclusive ownership notions are satisfied due to the inherent structure of the code equivalence problem. Indeed, for a given signature, there can essentially only be a single public key that verifies

[4] Typically, the signature scheme itself first hashes the message. It is understood that in this case, the transformed scheme would in fact replace this hash operation, i.e., it signs $\mathtt{H}(\mathtt{msg},\mathtt{pk})$ instead of $\mathtt{H}(\mathtt{H}(\mathtt{msg},\mathtt{pk}))$.

the message correctly. As this does not suffice to satisfy wNR, we show that using the PS-3 transform ensures full BUFF security for MEDS and LESS. Moreover, we note that PS-3-transformed MEDS and LESS can be considered to implement the full BUFF transform.

3.1 CROSS

CROSS is a code-based signature scheme based on a zero-knowledge identification protocol, the security of which relies on the NP-complete *restricted syndrome decoding problem*. To increase the soundness of the Fiat-Shamir transform, CROSS incorporates Merkle trees into its signature definition. There are two variants of CROSS, R-SDP and R-SDP(G), where the latter restricts the problem to a subgroup G, to achieve shorter signature sizes. As the analysis regarding BUFF security is the same for both versions, we will only consider CROSS-R-SDP(G).

The protocol uses integers k, m, n, t, w, λ, prime numbers p and z, and an element $g \in \mathbb{F}_p^*$ of order z. The cyclic subgroup generated by g is denoted by $\mathbb{E} \subseteq \mathbb{F}_p^*$ and G denotes a subgroup of $\mathbb{E}^n$. Further, a pseudorandom number generator PRNG is used, which we assume to be ideal throughout our analysis, i.e., the outputs are random.

Key Pair. The public key consists of a tuple $(\mathtt{seed}_{\mathtt{pk}}, s)$ for $\mathtt{seed}_{\mathtt{pk}} \in \{0,1\}^\lambda$ and $s \in \mathbb{F}_p^{n-k}$. The *secret key* is given by $\mathtt{seed}_{\mathtt{sk}} \in \{0,1\}^\lambda$.

Signature. The signature of a message $\mathtt{msg}$ consists of

$$\mathtt{salt}||d_{01}||d_b||\mathtt{MerkleProofs}||\mathtt{SeedPath}||\mathtt{rsp}_0||\mathtt{rsp}_1$$

for $d_{01}, d_b \in \{0,1\}^\lambda$, $\mathtt{MerkleProofs} \in \{0,1\}^{l_m}$, $\mathtt{SeedPath} \in \{0,1\}^{l_s}$ with

$$l_m = 2\lambda \left(1 + (t-w)\log_2\left(\frac{t}{t-w}\right)\right),\ l_s = \lambda(t-w)\log_2\left(\frac{t}{t-w}\right),$$

$\mathtt{rsp}_0 \in (\mathbb{F}_p^n \times \mathbb{F}_z^m)^{t-w}$, and $\mathtt{rsp}_1 \in (\{0,1\}^\lambda)^{t-w}$.

Verify. Given a public key $\mathtt{pk} = (\mathtt{seed}_{\mathtt{pk}}, s)$, a message $\mathtt{msg}$, and a signature $\mathtt{sig} = (\mathtt{salt}||d_{01}||d_b||\mathtt{MerkleProofs}||\mathtt{SeedPath}||\mathtt{rsp}_0||\mathtt{rsp}_1)$, the verification algorithm is shown in Fig. 4.

S-CEO. Given a public key $\mathtt{pk} = (\mathtt{seed}_{\mathtt{pk}}, s)$, a message $\mathtt{msg}$, and a signature $\mathtt{sig}$ such that $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$, we need to find a different public key $\overline{\mathtt{pk}} = (\mathtt{seed}_{\overline{\mathtt{pk}}}, \overline{s})$ such that $\mathtt{Verify}(\overline{\mathtt{pk}}, \mathtt{msg}, \mathtt{sig}) = 1$. Note that for $b[i] = 0$, the values t_i are computed as $x_i H^\top - \beta[i]s$, then hashed to $\mathtt{cmt}_0[i]$.

First, one sees that a change in the t_i leads to a change of $\mathtt{cmt}_0[i]$, consequently a change in d_0' and d_{01}', hence finally an invalid signature. Here, we use that changing the values in the Merkle tree results in another root, as long as the hash function is collision-resistant. Thus, any change of the public key that results in a change in any of the t_i will not be accepted in the verification.

$\texttt{Verify}(\texttt{pk}, \texttt{msg}, \texttt{sig})$

$(\texttt{seed}_{\texttt{pk}}, s) \leftarrow \texttt{pk}$

$(\texttt{salt}, d_{01}, d_b, \texttt{MerkleProofs}, \texttt{SeedPath}, \texttt{rsp}_0, \texttt{rsp}_1) \leftarrow \texttt{sig}$

$H, M_G \leftarrow \text{PRNG}(\texttt{seed}_{\texttt{pk}})$ // $H \in \mathbb{F}_p^{(n-k)\times k},\ M_G \in \mathbb{F}_z^{m\times n}$

$\beta \leftarrow \text{PRNG}(\texttt{H}(\texttt{H}(\texttt{msg})||d_{01}||\texttt{salt}))$ // $\beta \in (\mathbb{F}_p^*)^t$

$b \leftarrow \text{PRNG}(d_b)$ // $b \in \{0,1\}^t$ with hamming weight w

$(\texttt{seed}_0, \ldots, \texttt{seed}_{t-1}) \leftarrow \texttt{RebuildSeedTreeLeaves}(\texttt{SeedPath}, b, \texttt{salt})$

$j \leftarrow 0$

for $i = 0, \ldots, t-1$ **do**

 if $b[i] = 1$

  $(\texttt{cmt}_1[i], y_i) \leftarrow F_{b1}(\texttt{seed}_i, \texttt{salt}, M_G, \beta[i], i)$

 else

  $(\texttt{cmt}_0[i], \texttt{cmt}_1[i], y_i) \leftarrow F_{b0}(\texttt{rsp}_0[j], \texttt{rsp}_1[j], M_G, H, \beta[i], s, \texttt{salt}, i, j)$

  $j \leftarrow j + 1$

$d'_0 \leftarrow \texttt{RecomputeMerkleRoot}(\texttt{cmt}_0, \texttt{MerkleProofs}, b)$

$d'_1 \leftarrow \texttt{H}(\texttt{cmt}_1[i], \ldots, \texttt{cmt}_1[t-1]),\ d'_{01} = \texttt{H}(d'_0||d'_1),\ d'_b = \texttt{H}(y_0|| \ldots ||y_{t-1})$

if $(d_{01} = d'_{01} \wedge d_b = d'_b)$

  return 1

return 0

$F_{b1}(\texttt{seed}_i, \texttt{salt}, M_G, \beta[i], i)$

$\texttt{cmt}_1[i] \leftarrow \texttt{H}(\texttt{seed}_i||\texttt{salt}||i)$

$(u_i, \xi_i) \leftarrow \text{PRNG}(\texttt{seed}_i)$

// $u_i \in \mathbb{F}_p^n, \xi_i \in \mathbb{F}_z^m$

$\eta_i \leftarrow \xi_i M_G$

$e_i \leftarrow (g^{\eta_i[1]}, \ldots, g^{\eta_i[n]})$

return $(\texttt{cmt}_1[i], y_i = u_i + \beta[i]e_i)$

$F_{b0}(\texttt{rsp}_0[j], \texttt{rsp}_1[j], M_G, H, \beta[i], s, \texttt{salt}, i, j)$

$(y_i, \delta_i) \leftarrow \texttt{rsp}_0[j]$ // $(y_i, \delta_i) \in \mathbb{F}_p^n \times \mathbb{F}_z^m$

verify $\delta_i \in G$

$\sigma_i \leftarrow \delta_i M_G$

$v_i \leftarrow (g^{\sigma_i[1]}, \ldots, g^{\sigma_i[n]})$

$x_i \leftarrow v_i \star y_i$ // component-wise multiplication

$t_i = x_i H^\top - \beta[i]s$

$\texttt{cmt}_0[i] = \texttt{H}(t_i||\delta_i||\texttt{salt}||i)$

$\texttt{cmt}_1[i] = \texttt{rsp}_1[j]$

return $(\texttt{cmt}_0[i], \texttt{cmt}_1[i], y_i)$

Fig. 4: The verification algorithm of Cross. Note that the PRNG generation of H, M_G, u_i and ξ_i is depicted in a simplified fashion; further observe that $\texttt{RecomputeMerkleRoot}$ only needs the subset $\{\texttt{cmt}_0[i] \mid i \text{ s.t. } b[i] = 0\}$ of commitments. We do not provide definitions for functions that are not relevant for the BUFF analysis.

Hence, we have to find $\mathtt{pk} = (\mathtt{seed}_{\mathtt{pk}}, s) \neq (\mathtt{seed}_{\mathtt{pk}}, s) = \overline{\mathtt{pk}}$ such that $\overline{t}_i = t_i$ holds for all i with $b[i] = 0$. Note that we can assume that b has roughly $t/2$ bits equal to 0 as it is generated with the PRNG. Then the problem corresponds to solving the system $t_i = \overline{x}_i \overline{H}^\top - \beta[i]\overline{s}$ of $t/2$ random equations in the single indeterminate $\overline{s}$. If we choose $\mathtt{seed}_{\overline{\mathtt{pk}}} = \mathtt{seed}_{\mathtt{pk}}$, we have $\overline{H} = H$ and $\overline{x}_i = x_i$, thus there is no other solution than s. If we choose $\mathtt{seed}_{\overline{\mathtt{pk}}} \neq \mathtt{seed}_{\mathtt{pk}}$, we obtain a different pseudorandom matrix $\overline{H} \neq H$ and vector $\overline{x}_i \neq x_i$ and the probability that the resulting system is solvable is $\left(1/p^{n-k}\right)^{t/2}$. For all parameter sets of Cross, this is less than $2^{-20\,000}$. Therefore, Cross fulfills S-CEO security.

S-DEO. Given a public key $\mathtt{pk}$, a message $\mathtt{msg}$, and a signature $\mathtt{sig}$ such that $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$, we need to find a second public key $\overline{\mathtt{pk}} \neq \mathtt{pk}$ and a second message $\overline{\mathtt{msg}} \neq \mathtt{msg}$ such that $\mathtt{Verify}(\overline{\mathtt{pk}}, \overline{\mathtt{msg}}, \mathtt{sig}) = 1$. Here, the same argument as in the S-CEO analysis is applicable. Even though the message can be changed, this brings no advantage to an adversary as it is directly hashed, so that the value of $\overline{\beta}$ cannot be controlled. Thus, the situation is again that $\overline{s}$ needs to be chosen such that $\overline{s} = (\overline{x}_i \overline{H}^\top - t_i) \cdot \overline{\beta}[i]^{-1}$ holds for all i with $b[i] = 0$. With the same argument as above, this implies that Cross is S-DEO-secure.

MBS. One needs to find a public key $\mathtt{pk}$, two distinct messages $\overline{\mathtt{msg}} \neq \mathtt{msg}$, and a signature $\mathtt{sig}$, such that $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$ and $\mathtt{Verify}(\mathtt{pk}, \overline{\mathtt{msg}}, \mathtt{sig}) = 1$. For different messages, but the same signature and public key, only the values for β differ in the computation of t_i for i such that $b[i] = 0$. This implies $\overline{t_i} \neq t_i$ and hence the verification fails, as long as the hash function is collision-resistant. Therefore, MBS security is given.

wNR. Given a public key $\mathtt{pk}$ and a signature $\mathtt{sig}$ to an unknown message $\mathtt{msg}$, one has to find another public key $\overline{\mathtt{pk}} \neq \mathtt{pk}$, and a signature $\overline{\mathtt{sig}}$ such that $\mathtt{Verify}(\overline{\mathtt{pk}}, \mathtt{msg}, \overline{\mathtt{sig}}) = 1$. For unknown messages, the values of β are also unknown. Thus, even though the public key and signature can be chosen freely, an attacker cannot know what to set them to, making this problem as hard as a random search of two hash values, each of size at least 256 bits, depending on the security level. Hence, the success probability is at most 2^{-512}. We conclude that Cross is wNR-secure.

3.2 MEDS

Meds is a signature scheme based on the difficulty of finding equivalences of matrix codes in the rank metric. It is constructed from a zero-knowledge identification protocol and involves a technique to increase the soundness and thereby reduce the signature size. The protocol uses integers m, n, s, t, a prime power q, and the field $\mathbb{F}_q$ with q elements. The hash function $\mathtt{H}$ maps to $\{0, \ldots, s\}^t$, its entries are denoted h_i. The standard form of a code is the unique generator matrix in row-reduced echelon form.

Key Pair. The public key consists of matrices $G_0, \ldots, G_s \in \mathbb{F}_q^{k \times nm}$, all in standard form. For $i = 0, \ldots, s$, let C_i denote the code generated by G_i. The secret key consists of code equivalence maps $\pi_{A_i,B_i} \colon C_0 \to C_i$ for $i = 1, \ldots, s$, where A_i and B_i are square matrices of the appropriate sizes. It holds that G_i is the standard form of $A_i G_0 B_i$.

Signature. The signature of a message $\mathtt{msg}$ to a public key $(G_0, \ldots, G_s)$ consists of (h, π_{i,h_i}), where $h = \mathtt{H}(\tilde{G}_0, \ldots, \tilde{G}_t, \mathtt{msg}) \in \{0,1\}^t$, and $\pi_{i,h_i} \colon G_{h_i} \to \tilde{G}_i$ are code equivalences, for $i = 1, \ldots, t$. The matrices $\tilde{G}_i$ are constructed as $\tilde{A}_i G_0 \tilde{B}_i$ using random matrices $\tilde{A}_i$ and $\tilde{B}_i$, for $i = 1, \ldots, t$.

Verify. The verification algorithm computes $\tilde{G}_i$ using $\pi_{i,h_i} G_{h_i}$ and checks if $h = \mathtt{H}(\tilde{G}_1, \ldots, \tilde{G}_t, \mathtt{msg})$ holds.

S-CEO. Let $(G_j)_j$ be a public key and $\mathtt{msg}$ be a message with signature $\mathtt{sig} = (h, \pi_{i,h_i})$. Fix an index i and set $j = h_i$. Then, $\pi_{i,h_i} G_j$ and $\tilde{G}_i$ define the same code. Thus, if $(G'_j)_j$ would be another public key accepting the same signature for the message $\mathtt{msg}$, we find that $\pi_{i,h_i}^{-1} \tilde{G}_i$ and G'_j both define the same code as G_j. Hence, $G_j = G'_j$ by normalization. Thus, a message-signature pair cannot be attacked if the following assumption holds: For each j there is an index i such that $h_i = j$. Conversely, suppose j^* is an index such that $h_i \neq j^*$ for all i. Then we may pick an arbitrary G'_{j^*} different from G_{j^*}, while setting $G'_j = G_j$ for all $j \neq j^*$. As G_{j^*} or G'_{j^*} are not used, the verification succeeds. If an index j^* exists, the new public key is constructed in constant time.

We conclude that a message-signature pair is vulnerable to an S-CEO attack, if and only if for the corresponding h there is such an index j which is not one of the components of h. Assuming that h is uniformly random, this translates to picking uniformly maps $\{1, \ldots, t\} \to \{0, \ldots, s-1\}$ which are non-surjective.

As any such choice depends on a query to a signature oracle, we bound the number of queries by 2^{64}, cf. [33, Section 4.B.2]. We say a parameter set is vulnerable against an S-CEO attacker if, with less than 2^{64} queries, the probability of finding a non-surjective mapping exceeds 50%. Conversely, we declare a parameter set to be secure if, after 2^{64} queries, the probability of finding a non-surjective map is still negligible.

To compute these probabilities, we define A_ℓ as the event that after ℓ queries, no non-surjective map has been found. It is easy to see that

$$1 - \left(1 - \left(\frac{s-1}{s}\right)^t\right)^\ell \leq \mathbb{P}(A_\ell) \leq 1 - \left(1 - s\left(\frac{s-1}{s}\right)^t\right)^\ell.$$

Using standard formulas and approximations for logarithm, we find that for

$$q \approx \frac{\log(2)}{\left(\frac{s-1}{s}\right)^t},$$

the probability of finding non-surjective maps exceeds $\frac{1}{2}$. As can be seen in Table 2, this shows that all but two parameter sets of MEDS are vulnerable to attacks. For the remaining two parameter sets, we can use the upper bound

$$\mathbb{P}(A_{2^\lambda}) \leq 2^\lambda s \left(\frac{s-1}{s}\right)^t,$$

which is valid if $s\left(\frac{s-1}{s}\right)^t$ is sufficiently small. The bounds are given in the final row of Table 2.

Table 2: The third row denotes the number of queries q such that the attack probability is above 50%. The probability in the final row denotes the chance of finding a message-signature pair that is vulnerable after 2^{64} queries.

Security Level	I	I	III	III	V	V
s	4	5	4	5	5	6
t	1152	192	608	160	192	112
Lower bound $\log_2(q)$	477	61	251	50	61	28
Success probability after 2^{64} queries	2^{-412}	≈ 1	2^{-186}	≈ 1	≈ 1	≈ 1

S-DEO. MEDS satisfies S-DEO as any change in the message yields a change in the hash h that is part of the signature, unless a second preimage of the hash h is found.

MBS. MEDS satisfies MBS security trivially, if the hash function is collision-resistant, as distinct messages yield distinct hashes, contained in the signature. However, MEDS uses a 256-bit hash function for all security levels, yielding 128-bit MBS-security. For the security levels III and V, this falls short of the target security.

wNR. MEDS does not satisfy wNR security. Indeed, given a public key $(G_i)_i$ and a signature $(h, (\pi_{i,h_i}))$ that verify an unknown message $\mathtt{msg}$, we can adapt the public key and the signature as follows. Pick arbitrary matrices $\overline{A}, \overline{B}$ of the correct size, apply to G_1 the transformation $\pi_{\overline{A},\overline{B}}$, and update this new generator matrix $\overline{G}_1$ as the first component in the public key. For each i such that $h_i = 1$, modify the function $\pi_{i,1}$ to $\pi_{i,1} \circ \pi^{-1}_{\overline{A},\overline{B}}$. The verification will succeed, as by construction, $\pi_{i,1} \circ \pi^{-1}_{\overline{A},\overline{B}} \overline{G}_1 = \pi_{i,1} G_1 = \tilde{G}_i$. Note that h in the signature is unchanged.

Remark 3. The signature scheme MEDS would additionally satisfy wNR, if in the signing process, h would be redefined as $h := \mathtt{H}(\tilde{G}_1, \ldots, \tilde{G}_t, \mathtt{msg}, \mathtt{pk})$, which corresponds to an application of PS-3. Indeed, as h itself is part of the signature,

this change can be viewed as applying the BUFF transform to MEDS. In order to achieve MBS security, a larger hash output would be required to guarantee 192-bit resp. 256-bit collision resistance in security levels III and V.

3.3 LESS

LESS is, like MEDS, a signature scheme that relies on the code-equivalence problem and is based on a zero-knowledge identification protocol. Due to the strong similarity with MEDS, we do not provide all details. In short, LESS does not satisfy wNR, but satisfies S-DEO and MBS.[5] Like in the analysis of MEDS, S-CEO security depends on the parameter set. The detailed results can be found in Table 3. Note that the second parameter set requires fewer queries than the security parameter and after 2^{64} queries, the success probability of an attack is 2^{-35}. While 2^{100} signature queries are too many, this parameter set seems to be an edge case which we cannot safely declare to be secure. Adding the public key in the hash computation makes LESS BUFF secure, as this is essentially the BUFF transform.

Table 3: The third row denotes the number of queries q such that the attack probability is above 50%. The probability in the final row denotes the chance of finding a message-signature pair that is vulnerable after 2^{64} queries.

Security Level	I	I	I	III	III	V	V
s	2	4	8	2	3	2	3
t	247	244	198	759	895	1352	907
Lower bound $\log_2(q)$	246	100	37	758	523	inf	530
Success probability after 2^{64} queries	2^{-182}	2^{-35}	≈ 1	2^{-694}	2^{-457}	≈ 0	2^{-464}

3.4 WAVE

WAVE is a code-based signature scheme using the Hamming weight over the field $\mathbb{F}_3$. The security of WAVE relies on the syndrome decoding problem and a scheme-specific problem regarding the indistinguishability of the public key.

The Hamming weight of a vector over $\mathbb{F}_3$ is denoted $|_|$. WAVE uses integer parameters n and k, which are the length and dimension of the codes, and ω, a target Hamming weight.

[5] In contrast to MEDS, LESS uses a collision-resistant hash function for all security levels, providing MBS security.

Key Pair. The public key is a matrix $M = M(R) \in \mathbb{F}_3^{k\times(n-k)}$, where $R \in \mathbb{F}_3^{(n-k)\times k}$ is a matrix and $M(R)$ is defined row-wise by

$$\begin{aligned}\mathsf{row}(M, 2i) &= \mathsf{col}(R, 2i) + \mathsf{col}(R, 2i+1)\\ \mathsf{row}(M, 2i+1) &= \mathsf{col}(R, 2i) - \mathsf{col}(R, 2i+1),\end{aligned}$$

for $0 \le i < \frac{k-1}{2}$, and if k is odd, then $\mathsf{row}(M, k-1) = -\mathsf{col}(R, k-1)$.

Signature. A signature $\mathtt{sig} = (\mathtt{salt}, s)$ consists of an element $s \in \mathbb{F}_3^k$ and a random value $\mathtt{salt}$. It defines a valid signature for a message $\mathtt{msg}$ and the public key $M = M(R)$, if and only if

$$|s| + |\mathtt{H}(\mathtt{msg}||\mathtt{salt}) + Rs| = \omega\,, \tag{1}$$

where $\mathtt{H}$ is a hash function that maps to $\mathbb{F}_3^{n-k}$.

Verify. The verification algorithm checks whether Equation (1) holds.

S-CEO. Given a public key $M = M(R)$, any message $\mathtt{msg}$, and a signature $\mathtt{sig} = (\mathtt{salt}, s)$, we pick a matrix $\overline{R}$ such that $\overline{R}s = Rs$ but $\overline{R} \neq R$, for instance by extending s to a basis and defining $\overline{R}$ on the other basis vectors randomly. Then, Equation (1) holds trivially with $\overline{R}$. Setting $\overline{M} = M(\overline{R})$ yields the new public key.

S-DEO. Given a public key $M = M(R)$, any message $\mathtt{msg}$, and a signature $\mathtt{sig} = (\mathtt{salt}, s)$, we randomly pick a new message $\overline{\mathtt{msg}} \neq \mathtt{msg}$ and compute $\overline{\mathtt{h}} \coloneqq \mathtt{H}(\overline{\mathtt{msg}}||\mathtt{salt})$. We pick a vector $t \in \mathbb{F}_3^{n-k}$ such that $|\overline{\mathtt{h}} - t| = \omega - |s| =: \omega_s$, which can be done by choosing a random t' with hamming weight ω_s and setting $t = \overline{\mathtt{h}} - t'$. Then we choose $\overline{R}$ such that $\overline{R}s = t$ and set $\overline{M} = M(\overline{R})$. We find that Equation (1) is satisfied, indeed, $|s| + |\overline{\mathtt{h}} - \overline{R}s| = |s| + |t'| = \omega$.

MBS. The MBS security of Wave can be attacked as follows. First, we pick random messages $\mathtt{msg} \neq \overline{\mathtt{msg}}$, and a random $\mathtt{salt}$. We compute $h = \mathtt{H}(\mathtt{msg}||\mathtt{salt})$ and $\overline{h} = \mathtt{H}(\overline{\mathtt{msg}}||\mathtt{salt})$. Then we need to find $t \in \mathbb{F}_3^{n-k}$, such that

$$\omega' \coloneqq |h - t| = |\overline{h} - t| < \omega\,.$$

Indeed, if we have found such a t, we define s such that $|s| = \omega - \omega'$ and R such that $Rs = t$. Then, both messages are verified with the signature $\mathtt{sig} = (\mathtt{salt}, s)$ under the public key $M = M(R)$, as Equation (1) is satisfied for both.

A simple but tedious combinatorial construction shows that such a t can be found in almost all cases.[6]

[6] A Python script is provided at `https://git.uni-regensburg.de/buff/wave_mbs`.

wNR. The attack against the S-CEO security of Wave applies to wNR, as no information about the message is required.

Remark 4. For Wave, we can show that applying the PS-3 transform *does not suffice* to achieve full BUFF security. Let us suppose that the value h in the signature is set to $\mathtt{H}(\mathtt{msg}||\mathtt{pk}||\mathtt{salt})$, and a signature $(\mathtt{salt}, s)$ is valid, if Equation (1) holds with this h. Then, the resulting signature scheme is not MBS secure. Indeed, we begin by picking a matrix R from which $\mathtt{pk}$ is deduced and for which we know an efficient decoding algorithm $\mathcal{G}$. We set $\mathtt{salt}$ randomly. We pick random messages $\mathtt{msg}$ and $\overline{\mathtt{msg}}$ and compute $\mathtt{h} = \mathtt{H}(\mathtt{msg}||\mathtt{pk}||\mathtt{salt})$ and $\overline{\mathtt{h}}(\overline{\mathtt{msg}}||\mathtt{pk}||\mathtt{salt})$. As in the attack against MBS security for the original Wave scheme, we can find t such that $\omega' \coloneqq |\mathtt{h} - t| = |\overline{\mathtt{h}} - t|$. We set $d = \omega - \omega'$ and run $\mathcal{G}$ with target vector t and Hamming weight d to obtain s. Then, $\mathtt{sig} = (\mathtt{salt}, s)$ is a valid signature for both $\mathtt{msg}$ and $\overline{\mathtt{msg}}$ under the public key $\mathtt{pk}$.

Despite this, the PS-3-transformed version of Wave does satisfy S-CEO, S-DEO, and wNR.

4 Isogeny-based schemes

In this section, we analyze the BUFF security of SQIsign [11], the sole isogeny-based signature scheme submitted to the NIST standardization process. We first give some background and notation that we require for the analysis.

Background and notation. For elliptic curves E, E' over a finite field $\mathbb{F}_q$, an isogeny is a non-constant morphism $\varphi : E \to E'$ such that $\varphi(\infty_E) = \infty_{E'}$ for the respective points at infinity on E resp. E'. A subgroup G of order m uniquely (up to composition with isomorphisms) determines an isogeny $\varphi : E \to E/G$, where the kernel $\ker(\varphi) = G$ and the degree of φ is m. Such a subgroup can be described by a single point $K \in E$ of order m, i.e., $G = \langle K \rangle$. SQIsign uses a compressed representation of subgroups: Given a deterministic basis (P, Q) of the m-torsion subgroup $E[m]$, we can represent a suitable point as $K = P + [s]Q$ or $K = [s]P + Q$ for an $s \in \mathbb{Z}/m\mathbb{Z}$. Hence, given s and a decision bit $b \in \{0, 1\}$, we can compute $K = P + [s]Q$, where b indicates whether P and Q need to swapped prior to computing K. All occurring values s and b (with indices) will be of this form, and we refer to this computation as $\mathtt{Decompress}_{P,Q}(s, b)$, where b can be omitted if no point swap is necessary. Each isogeny $\varphi : E \to E'$ has a unique dual isogeny $\widehat{\varphi} : E' \to E$ such that the composition $\widehat{\varphi} \circ \varphi$ resp. $\varphi \circ \widehat{\varphi}$ is the multiplication-by-m map on E resp. E'. We will only use supersingular curves E over $\mathbb{F}_{p^2}$ for a large prime p.

4.1 SQIsign

SQIsign applies the Fiat-Shamir transform to an identification protocol based on isogenies. Following Fig. 5, we define a public starting curve E_0, and the prover computes a secret isogeny $\varphi_A : E_0 \to E_A$, where E_A is published. The

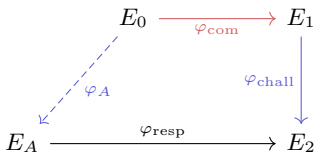


Fig. 5: The SQISIGN protocol with three phases: commitment φ_{com}, challenge φ_{chall}, and response φ_{resp}.

Verify$(\mathtt{pk}, \mathtt{msg}, \mathtt{sig})$

1 : $(b, s^{(1)}, \ldots, s^{(n)}, r, b_2, s_2, b_3, s_3) \leftarrow \mathtt{sig}$
2 : $E^{(1)} \leftarrow \mathtt{pk}$
3 : $(P^{(1)}, Q^{(1)}) \leftarrow \mathtt{FindBasis}_{2^f}(E^{(1)})$
4 : $K^{(1)} \leftarrow \mathtt{Decompress}_{P^{(1)},Q^{(1)}}(s^{(1)}, b)$
5 : **for** $j = 1, \ldots, n-1$ **do**
6 : $\quad \varphi^{(j)} : E^{(j)} \to E^{(j+1)} = E/\langle K^{(j)} \rangle$
7 : $\quad Q^{(j+1)} \leftarrow \varphi^{(j)}(Q^{(j)})$
8 : $\quad P^{(j+1)} \leftarrow \mathtt{CompleteBasis}_{2^f}(E^{(j+1)}, Q^{(j+1)})$
9 : $\quad K^{(j+1)} \leftarrow \mathtt{Decompress}_{P^{(j+1)},Q^{(j+1)}}(s^{(j+1)})$ ▮
10 : $\varphi^{(n)} : E^{(n)} \to E^{(n+1)} = E/\langle K^{(n)} \rangle$
11 : $Q', E_1 = \mathtt{Decompress}-$
$\quad \mathtt{Challenge}(E^{(n+1)}, b_2, s_2, b_3, s_3)$
12 : **if** $[r]Q' = \mathcal{H}(\mathtt{msg}, E_1)$
13 : $\quad$ **return** 1
14 : **return** 0

Fig. 6: Verification algorithm of SQISIGN.

prover commits to the codomain E_1 of the commitment isogeny $\varphi_{\text{com}} : E_0 \to E_1$, followed by the challenger providing a challenge isogeny $\varphi_{\text{chall}} : E_1 \to E_2$. The prover answers with an isogeny $\varphi_{\text{resp}} : E_A \to E_2$. For the computation and the zero-knowledge property of φ_{resp} we refer to the SQISIGN specification [11]. The standard Fiat-Shamir transform turns this protocol into a non-interactive signature scheme. We note that, due to the exponentially large challenge space, a single round of the protocol suffices.

Key Pair. For a fixed supersingular curve E_0 over $\mathbb{F}_{p^2}$ of known endomorphism ring, a secret key is an isogeny $\varphi_A : E_0 \to E_A$. The public key is given by E_A.

Signature. A signature consists of compressed descriptions of the isogenies φ_{resp} and φ_{chall}. For fixed positive integers e, f, g, n with $e = nf$ it is of the form

$$\mathtt{sig} = (b, s^{(1)}, \ldots, s^{(n)}, r, b_2, s_2, b_3, s_3),$$

where $b, b_2, b_3 \in \{0, 1\}$, $s^{(j)}, s_2 \in \mathbb{Z}/2^f\mathbb{Z}$, $s_3 \in \mathbb{Z}/3^g\mathbb{Z}$, and $r \in \mathbb{Z}/2^f 3^g\mathbb{Z}$, following the notation from [14].

Verify. The verification algorithm, described in Fig. 6, consists of three parts. The most relevant part for the following discussion is the recomputation of

$\varphi_{\text{resp}} : E_A \to E_2$ through a chain of n isogenies $\varphi^{(j)}$ of degree 2^f. Each isogeny $\varphi^{(j)}$ is determined by a kernel generator $K^{(j)}$. We compute these $K^{(j)}$ by deterministically sampling a basis $(P^{(j)}, Q^{(j)})$ of $E^{(j)}[2^f]$ through `FindBasis` if no point is given resp. `CompleteBasis` if $Q^{(j)}$ is given, and running `Decompress` with input $s^{(j)}$ (and b for $j = 1$). In particular, for $j > 1$, only $P^{(j)}$ is sampled, while we get $Q^{(j)} = \varphi^{(j-1)}(Q^{(j-1)})$, such that $Q^{(j)}$ generates the kernel of the dual isogenies $\widehat{\varphi^{(j-1)}}$. Therefore, we compute φ_{resp} through the following chain:

$$E_A = E^{(1)} \xrightarrow{\varphi^{(1)}} E^{(2)} \xrightarrow{\varphi^{(2)}} E^{(3)} \xrightarrow{\varphi^{(3)}} \cdots \xrightarrow{\varphi^{(n)}} E^{(n+1)} = E_2$$

In the second step, summarized in `DecompressChallenge`, we recompute the dual $\widehat{\varphi_{\text{chall}}} : E_2 \to E_1$ of order $D_{\text{chall}} = 2^f 3^g$ using `FindBasis` and `Decompress` with input (b_2, s_2, b_3, s_3). For a deterministically sampled point $Q'' \in E_2$ of order D_{chall} that is linearly independent of $\ker(\widehat{\varphi_{\text{chall}}})$, it computes $Q' \leftarrow \widehat{\varphi_{\text{chall}}}(Q'')$. Furthermore, this function verifies that the composition $\widehat{\varphi_{\text{chall}}} \circ \varphi_{\text{resp}}$ is cyclic.

The final step verifies that $[r]Q'$ corresponds to the kernel generator of the challenge isogeny, i.e. $[r]Q' = \mathcal{H}(\texttt{msg}, E_1)$. The function $\mathcal{H}$ is defined to first compute $a = \texttt{H}(\texttt{msg}, j(E_1)) \in \mathbb{Z}/D_{\text{chall}}\mathbb{Z}$ for a hash function `H` and the j-invariant $j(E_1)$, and output $R_1 + [a]S_1$ with a deterministic basis (R_1, S_1) of $E_1[D_{\text{chall}}]$.

S-CEO. Let `sig` be a valid signature for $\texttt{pk} = E_A$ and `msg`, i.e., $\texttt{Verify}(\texttt{pk}, \texttt{msg}, \texttt{sig}) = 1$. Our aim is to construct a public key $\overline{\texttt{pk}} = E_{A'} \neq E_A$ such that $\texttt{Verify}(\overline{\texttt{pk}}, \texttt{msg}, \texttt{sig}) = 1$. This amounts to finding $E_{A'}$ for which the compression in `sig` describes an isogeny $\psi_{\text{resp}} : E_{A'} \to E_2$ that has the same codomain E_2. In this case, the second and third step of the verification are the same as when running $\texttt{Verify}(\texttt{pk}, \texttt{msg}, \texttt{sig})$.

A naive way to find such a $E_{A'}$ is to compute random 2^e-isogenies $\psi' : E_2 \to E_{A'}$ and check if $(b, s^{(1)}, \ldots, s^{(n)})$ generates an isogeny $\psi_{\text{resp}} : E_{A'} \to E_2$ mapping to the correct E_2. However, the fact that we know several curves on the path between E_A and E_2 from `sig` allows for an easier S-CEO attack as follows:

1. Find $\widetilde{\psi}^{(1)} : E^{(2)} \to E_{A'}$ of degree 2^f with `FindBasis` and $\texttt{Decompress}(s^{(1)}, b)$ generating the 2^f-isogeny $\psi^{(1)} : E_{A'} \to E^{(2)}$ with the desired codomain.
2. Ensure that the following 2^f-isogenies satisfy $\psi^{(j)} = \varphi^{(j)}$ for $j > 1$, and hence ψ_{resp} maps to E_2.

Explicitly generating $E_{A'}$ in the first step seems infeasible, hence we resort to a search approach, going through all 2^f suitable isogenies $\widetilde{\psi}$. We require that the deterministic basis $(\widetilde{P}, \widetilde{Q})$ of $E_{A'}[2^f]$ and $b, s^{(1)} \in \texttt{sig}$ construct a suitable kernel generator $\widetilde{K}$ such that $\psi^{(1)} : E_{A'} \to E_{A'}/\langle \widetilde{K} \rangle = E^{(2)}$. Since there are $3 \cdot 2^{f-1}$ isogenies of degree 2^f starting from $E_{A'}$ and `sig` determines exactly one of these, the success probability for this step, given `sig`, is $1/(3 \cdot 2^{f-1})$. Thus, we can expect to find a suitable curve $E_{A'}$ with a probability of roughly 50%.

Assuming we found a suitable $E_{A'}$, we obtain a basis $(P^{(2)}, \widetilde{Q}^{(2)})$ of $E^{(2)}[2^f]$, where $\widetilde{Q}^{(2)} = \psi^{(1)}(\widetilde{Q})$. In contrast, a verification starting from E_A obtains the basis $(P^{(2)}, Q^{(2)})$ with the same sampled point $P^{(2)}$, but $Q^{(2)} = \varphi^{(1)}(Q^{(1)})$. Since

the dual isogenies of $\varphi^{(1)}$ and $\psi^{(1)}$ are not equal, we have $\widetilde{Q}^{(2)} \notin \langle Q^{(2)} \rangle$. For the second attack step, we require for $j > 1$ that

$$\langle P^{(j)} + [s^{(j)}]Q^{(j)} \rangle = \langle P^{(j)} + [s^{(j)}]\widetilde{Q}^{(j)} \rangle.$$

All following steps trivially succeed if $s^{(j)} = 0$ for all $j > 1$. Furthermore, if $[2^k]Q^{(2)} = [2^k]\widetilde{Q}^{(2)}$ for $0 < k < f$, we succeed if $s^{(j)} \equiv 0 \mod 2^k$ for all $j > 1$. Even though the attack can only succeed if the signature values $s^{(j)}$ for $j > 1$ have a very special shape, it appears infeasible to enumerate all such possibilities, and compute an explicit success probability.

Instead, we implemented this attack using the AprèsSQI software [14], which closely follows the NIST submission of SQIsign.[7] For reduced parameters that allow feasible running times, i.e., a 36-bit prime p and $f \in \{7, 8, 9, 10\}$, our implementation suggests that the probability of a given $(s^{(1)}, \ldots, s^{(n)})$ to be vulnerable to this attack is below 2^{-f}. If we conjecture that this behavior scales to the SQIsign parameter sizes featuring $f = 75, 97, 145$ for NIST-I/III/V, this means that for each given $\mathtt{sig}$, the S-CEO attack has a search complexity of $O(2^f)$ and success probability of 2^{-f}. Although we can conjecture that this attack does not break S-CEO security, we emphasize that better attack avenues might exist, and our probability estimations can only be viewed as a lower bound.

Remark 5. The probability and effort for a possible attack depend on the size of f and how φ_{resp} is verified. E.g., the SQIsign variant AprèsSQI [14] proposes much larger values of f, which push the success probability below the probability of breaking EUF-CMA. On the other hand, earlier parameter proposals use smaller values of f [18,15,19,10], therefore simplifying the described attack.

AprèsSQI further proposes a variant that samples both $P^{(j)}$ and $Q^{(j)}$ deterministically instead of obtaining $Q^{(j)}$ though an isogeny evaluation. Hence, the second step in the S-CEO attack automatically succeeds, pushing the overall success probability for a given signature to 50% with search complexity $O(2^f)$.

S-DEO. In contrast to S-CEO, we additionally need to find a message $\overline{\mathtt{msg}} \neq \mathtt{msg}$ such that $\mathtt{Verify}(\overline{\mathtt{pk}}, \overline{\mathtt{msg}}, \mathtt{sig}) = 1$. Thus, we can only repeat the S-CEO attack above if the challenge curves E_2 resp. E_2' when signing $\mathtt{msg}$ with $\mathtt{pk}$ resp. $\overline{\mathtt{msg}}$ with $\overline{\mathtt{pk}}$ are equal, requiring $\mathcal{H}(\mathtt{msg}, E_1) = \mathcal{H}(\overline{\mathtt{msg}}, E_1)$, and therefore finding a second preimage of the form $(\overline{\mathtt{msg}}, E_1)$ of $\mathcal{H}(\mathtt{msg}, E_1)$.

If $\mathcal{H}(\mathtt{msg}, E_1) \neq \mathcal{H}(\overline{\mathtt{msg}}, E_1)$, i.e. $E_2' \neq E_2$, this attack is not available, hence we can only pick a random public key $\overline{\mathtt{pk}}$. During verification, after recomputing ψ_{resp} and running $\mathtt{DecompressChallenge}$, we end up at $E_1' \neq E_1$, such that the check $[r]Q' = \mathcal{H}(\overline{\mathtt{msg}}, E_1')$ only succeeds with negligible probability $1/D_{\text{chall}}$. Therefore SQIsign is S-DEO-secure.

MBS. Assume that we have a valid signature $\mathtt{sig}$ for $\mathtt{pk}$ and $\mathtt{msg}$, i.e., $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$, and a message $\overline{\mathtt{msg}} \neq \mathtt{msg}$ such that $\mathtt{Verify}(\mathtt{pk}, \overline{\mathtt{msg}}, \mathtt{sig}) = 1$. In

[7] The implementation is available at `https://git.uni-regensburg.de/buff/sqisign_ceo`.

both verification runs, the first and second step that recompute φ_{resp} and $\widehat{\varphi_{\text{chall}}}$ are equal. In the last step, both runs compute $Q' \leftarrow \widehat{\varphi_{\text{chall}}}(Q'')$ and verify that $[r]Q' = \mathcal{H}(\mathtt{msg}, E_1)$ resp. $[r]Q' = \mathcal{H}(\overline{\mathtt{msg}}, E_1)$. However, if verification for $\mathtt{msg}$ and $\overline{\mathtt{msg}}$ succeeds, we have $\mathtt{H}(\mathtt{msg}, j(E_1)) = \mathtt{H}(\overline{\mathtt{msg}}, j(E_1))$, yielding a hash collision of $\mathtt{H}$ modulo D_{chall}. For security level I, III, resp. V, $\mathtt{H}$ has an output length of $\lambda \in \{128, 192, 256\}$ bits, yielding a collision finding complexity of $O(2^{\lambda/2})$. Hence, SQIsign only achieves 64-, 96-, resp. 128-bit security against MBS attacks for security levels I, III, resp. V.

wNR. Given a public key $\mathtt{pk}$ and a signature $\mathtt{sig}$ for an unknown message $\mathtt{msg}$, an attacker has to find a public key $\overline{\mathtt{pk}} \neq \mathtt{pk}$ and a signature $\overline{\mathtt{sig}}$ such that $\mathtt{Verify}(\overline{\mathtt{pk}}, \mathtt{msg}, \overline{\mathtt{sig}}) = 1$. To construct $\overline{\mathtt{pk}}$ and $\overline{\mathtt{sig}}$, we run the first step of $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig})$ to obtain the curve E_2. We choose a random 2^f-isogeny $\widehat{\psi}^{(n)} : E_2 \to \widetilde{E}^{(n)}$ such that the composition $\varphi_{\text{chall}} \circ \widehat{\psi}^{(n)}$ is cyclic. Starting from $j = n - 1$ in decreasing order, we then construct random 2^f-isogenies $\widehat{\psi}^{(j)} : \widetilde{E}^{(j+1)} \to \widetilde{E}^{(j)}$ such that the composition $\widehat{\psi} = \widehat{\psi}^{(1)} \circ \cdots \circ \widehat{\psi}^{(n)}$ is cyclic. For each of the $\widehat{\psi}^{(j)}$, we pick a point $R \in \widetilde{E}^{(j+1)}$ of order 2^f such that R is linearly independent of $\ker(\widehat{\psi}^{(j)})$. Therefore, $K^{(j)} = \widehat{\psi}^{(j)}(R)$ generates the kernel of the dual isogeny $\psi^{(j)}$ of $\widehat{\psi}^{(j)}$.

For the signature $\overline{\mathtt{sig}}$ we use $\psi_{\text{resp}} = \psi^{(n)} \circ \cdots \circ \psi^{(1)}$ and the public key $\overline{\mathtt{pk}} = E_{A'} = \widetilde{E}^{(1)}$. For a valid signature the kernel generator points $K^{(j)}$ have to be represented in a compressed form. To compute this representation, we follow the approach of SQIsign. For the deterministic basis $(\widetilde{P}^{(1)}, \widetilde{Q}^{(1)})$ this allows us to find $\widetilde{s}^{(1)}$ to get a suitable kernel generator $\widetilde{P}^{(1)} + [\widetilde{s}^{(1)}]\widetilde{Q}^{(1)})$ for $\psi^{(1)}$, potentially swapping $\widetilde{P}^{(1)}$ and $\widetilde{Q}^{(1)}$ by setting $\widetilde{b} = 1$, and $\widetilde{b} = 0$ otherwise. The following steps proceed analogously, computing $\widetilde{s}^{(j)}$ through discrete logarithms without requiring to swap points.

Since E_2 is the codomain of ψ and $\widehat{\varphi_{\text{chall}}} \circ \psi$ is cyclic by construction, we can reuse the values r, b_2, s_2, b_3, s_3 for the second and third step of the verification of $\overline{\mathtt{sig}}$. Hence, we have constructed a public key $\overline{\mathtt{pk}} \neq \mathtt{pk}$ and signature $\overline{\mathtt{sig}} \neq \mathtt{sig}$ of the form $\overline{\mathtt{sig}} = (\widetilde{b}, \widetilde{s}^{(1)}, \ldots, \widetilde{s}^{(n)}, r, b_2, s_2, b_3, s_3)$ such that $\mathtt{Verify}(\overline{\mathtt{pk}}, \mathtt{msg}, \overline{\mathtt{sig}}) = 1$ without requiring knowledge of $\mathtt{msg}$.

Remark 6. For SQIsign, the PS-3 transform suffices to achieve S-CEO, S-DEO, and wNR security. In this case, the signer computes the challenge generator through $\mathcal{H}(\mathtt{msg}, \mathtt{pk}, E_1)$, which uses the hash value $a = \mathtt{H}(\mathtt{msg}, \mathtt{pk}, j(E_1)) \in \mathbb{Z}/D_{\text{chall}}\mathbb{Z}$ as described above. This means that $r \in \mathtt{sig}$, which satisfies $[r]Q' = \mathcal{H}(\mathtt{msg}, \mathtt{pk}, E_1)$ for a deterministic point Q', can be viewed as an encoding of the hash value $\mathtt{H}(\mathtt{msg}, \mathtt{pk}, j(E_1))$, resembling the BUFF transform.

In this case, the problem to solve S-CEO is equivalent to the description of C-DEO above. For wNR, the PS-3 transform implies that the curve E_2 in $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig})$ is not a valid challenge curve in $\mathtt{Verify}(\overline{\mathtt{pk}}, \mathtt{msg}, \overline{\mathtt{sig}})$. Attacking wNR thus requires to pick $\overline{\mathtt{pk}}, \overline{\mathtt{sig}}$ and hope for $[r]Q' = \mathcal{H}(\overline{\mathtt{msg}}, E_1')$ to hold for the chosen r, which has negligible success probability.

However, we note that in order to achieve MBS security, a collision resistant hash function is required.

$\texttt{Verify}(\texttt{pk}, \texttt{msg}, \texttt{sig})$

1 : $(\texttt{salt}, s) \leftarrow \texttt{sig}$
2 : $(x_1, x_2) \leftarrow \texttt{Decompress}(s)$
3 : $(\texttt{seed}_{\hat{A}}, B) \leftarrow \texttt{pk}$
4 : $u \leftarrow \texttt{H}(\texttt{msg}, \texttt{salt})$
5 : $\hat{A} \leftarrow \texttt{XOF}(\texttt{seed}_{\hat{A}})$
6 : $x_0 \leftarrow (u - \hat{A}x_1 - Bx_2) \mod Q$
7 : **if** $||(x_0, x_1, x_2)|| \leq \vartheta$
8 : **return** 1
9 : **return** 0

Fig. 7: Verification algorithm of HuFu. Note that (x_0, x_1, x_2) denotes the vector obtained from concatenating x_0, x_1, and x_2 and $||\cdot||$ is the l_2-norm.

5 Lattice-based schemes

The lattice-based schemes we deal with in this section can be divided into two groups: Raccoon and Haetae, which are closely related to Dilithium; HuFu and Squirrels, which follow a GPV-like approach. For Raccoon and Haetae, we give an outline of the analysis from [16] in Section 5.3. The cases of HuFu (Section 5.1) and Squirrels (Section 5.2) are quite different, and we do a hands-on analysis. The results turn out to differ for the two schemes. While HuFu only lacks wNR security, Squirrels is insecure with respect to all notions. We remark that a PS-3-transformed HuFu would satisfy all BUFF security notions. Finally, Squirrels is vulnerable even after the PS-3 transform and only the full BUFF transform could achieve all notions.

5.1 HuFu

HuFu applies the GPV approach. It uses unstructured lattices and is based on the short integer solution and learning with errors problems.

Key Pair. Consider a distribution χ over $\mathbb{Z}$, $m, n \in \mathbb{N}$, and $Q = pq$ for p, q some powers of 2. The secret key is a tuple of matrices $\texttt{sk} = (S, E, L_{22}, L_{32}, L_{33})$ for $(S, E) \leftarrow \chi^{n \times m} \times \chi^{m \times m}$ and $L_{22} \in \mathbb{R}^{n \times n}$, $L_{32} \in \mathbb{R}^{m \times n}$, and $L_{33} \in \mathbb{R}^{m \times m}$. The public key is a pair $\texttt{pk} = (\text{seed}_{\hat{A}}, B = p \cdot I - (\hat{A}S + E))$ for $\hat{A} \in \mathbb{Z}_Q^{m \times n}$ generated using $\text{seed}_{\hat{A}}$.

Signature. The signature $\texttt{sig}$ of a message $\texttt{msg}$ consists of a tuple $(\texttt{salt}, s)$ for $s = \texttt{Compress}(x_1, x_2)$, where $x_1 \in \mathbb{Z}^n$ and $x_2 \in \mathbb{Z}^m$.

Verify. Given a public key $\texttt{pk} = (\text{seed}_{\hat{A}}, B)$, a message $\texttt{msg}$, and a signature $\texttt{sig} = (\texttt{salt}, s)$, the verification algorithm is shown in Fig. 7.

S-CEO. Given a public key $\texttt{pk} = (\texttt{seed}_{\hat{A}}, B)$, a message $\texttt{msg}$ and a signature $\texttt{sig} = (\texttt{salt}, s)$ such that $\texttt{Verify}(\texttt{pk}, \texttt{msg}, \texttt{sig}) = 1$, we need to find a second public key $\overline{\texttt{pk}} = (\overline{\texttt{seed}}_{\hat{A}}, \overline{B})$ such that $\texttt{Verify}(\overline{\texttt{pk}}, \texttt{msg}, \texttt{sig}) = 1$. We choose $\overline{\texttt{seed}}_{\hat{A}} = \texttt{seed}_{\hat{A}}$, which expand to the same matrix $\hat{A}$. As $(\texttt{salt}, s)$ is a valid

signature, we know that $||(x_0, x_1, x_2)|| \leq \vartheta$, where $x_0 = (u - \hat{A}x_1 - Bx_2) \mod Q$. Thus, if we find $\overline{B}$ s.t. $\overline{x_0} = x_0$, we obtain $||(\overline{x_0}, x_1, x_2)|| = ||(x_0, x_1, x_2)|| \leq \vartheta$, which shows that S-CEO security is not given. In order to construct such a $\overline{B}$, first note that we can assume that there is at least one i such that $x_{2,i} \neq 0$, as otherwise one can trivially choose $\overline{B} \neq B$ with the desired properties. Without loss of generality, we assume $x_{2,1} \neq 0$. Then we define $\overline{B} \neq B$ as follows: $\overline{b}_{11} = (b_{11} + x_{2,2})$, $\overline{b}_{12} = (b_{12} - x_{2,1})$, and $\overline{b}_{ij} = b_{ij}$ for all other i, j. It holds that $(\overline{B}x_2)_1 = (Bx_2)_1$. Thus $\overline{B}x_2 = Bx_2$ as only the first row differs for $\overline{B}$ and B. This implies $\overline{x_0} = x_0$.

S-DEO. Given a public key $\mathtt{pk} = (\mathtt{seed}_{\hat{A}}, B)$, a message $\mathtt{msg}$, and a signature $\mathtt{sig} = (\mathtt{salt}, s)$ s.t. $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$, we need to find a second public key $\overline{\mathtt{pk}} \neq \mathtt{pk}$ and a second message $\overline{\mathtt{msg}} \neq \mathtt{msg}$ s.t. $\mathtt{Verify}(\overline{\mathtt{pk}}, \overline{\mathtt{msg}}, \mathtt{sig}) = 1$. We choose again $\overline{\mathtt{seed}}_{\hat{A}} = \mathtt{seed}_{\hat{A}}$, which yield the same matrix $\hat{A}$. Further we choose $\overline{\mathtt{msg}} \neq \mathtt{msg}$ randomly and compute u and $\overline{u}$. If we find $\overline{B}$ such that $\overline{x_0} = \overline{u} - \hat{A}x_1 - \overline{B}x_2 = 0 \mod Q$, we obtain $||(\overline{x_0}, x_1, x_2)|| \leq ||(x_0, x_1, x_2)|| \leq \vartheta$. Then, we have $\mathtt{Verify}(\overline{\mathtt{pk}}, \overline{\mathtt{msg}}, \mathtt{sig}) = 1$ for $\overline{\mathtt{pk}} = (\mathtt{seed}_{\hat{A}}, \overline{B})$, which gives an attack against S-DEO security. A matrix $\overline{B}$ such that $\overline{B}x_2 = \overline{u} - \hat{A}x_1$ can be constructed if $\gcd(x_{2,i}) = 1$.[8] As $m \geq 768$, the coefficients of $x_2 \in \mathbb{Z}^m$ are coprime with overwhelming probability given by $\zeta(m)^{-1} \approx 1$.

MBS. One needs to find a public key $\mathtt{pk} = (\mathtt{seed}_{\hat{A}}, B)$, two distinct messages $\mathtt{msg} \neq \overline{\mathtt{msg}}$, and a signature $\mathtt{sig} = (\mathtt{salt}, s)$ such that $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$ and $\mathtt{Verify}(\mathtt{pk}, \overline{\mathtt{msg}}, \mathtt{sig}) = 1$. Assume, we have found $\mathtt{pk}, \mathtt{msg}, \overline{\mathtt{msg}}$, and $\mathtt{sig} = (\mathtt{salt}, s)$ with these properties. Then $||(x_0, x_1, x_2)||, ||(\overline{x_0}, x_1, x_2)|| \leq \vartheta$ and hence in particular $||x_0||, ||\overline{x_0}|| \leq \vartheta$. Observe that this implies $||u - \overline{u}|| = ||u - (\hat{A}x_1 - Bx_2) + (\hat{A}x_1 - Bx_2) - \overline{u}|| = ||x_0 - \overline{x_0}|| \leq ||x_0|| + ||\overline{x_0}|| \leq 2\vartheta$. As $u = \mathtt{H}(\mathtt{msg}, r)$ and $\overline{u} = \mathtt{H}(\overline{\mathtt{msg}}, r)$, the probability to find messages that yield u and $\overline{u}$ which are close to each other is negligible (near-collision resistance of the hash function).[9]

wNR. Given a public key $\mathtt{pk} = (\mathtt{seed}_{\hat{A}}, B)$ and a signature $\mathtt{sig} = (\mathtt{salt}, s)$ to an unknown message $\mathtt{msg}$, one has to find another public key $\overline{\mathtt{pk}} \neq \mathtt{pk}$, and a signature $\overline{\mathtt{sig}} = (\overline{\mathtt{salt}}, \overline{s})$ such that $\mathtt{Verify}(\overline{\mathtt{pk}}, \mathtt{msg}, \overline{\mathtt{sig}}) = 1$. To do this, we can proceed exactly as we did for S-CEO. Note that for the attack it is not necessary to know the message and we can choose $\overline{\mathtt{sig}} = (\overline{\mathtt{salt}}, \overline{s}) = (\mathtt{salt}, s) = \mathtt{sig}$.

Remark 7. We showed that HuFu only achieves MBS security. We observe, however, that by applying the PS-3 transform, i.e., changing the computation of $u = \mathtt{H}(\mathtt{msg}, \mathtt{salt})$ to $u = \mathtt{H}(\mathtt{msg}, \mathtt{pk}, \mathtt{salt})$, full BUFF security can be achieved. This is the case, as the above change prevents an attacker to control x_0 by their choice of $\mathtt{pk}$—any change to $\mathtt{pk}$ also changes the value of u and hence h in an uncontrollable way. Using this, S-DEO, S-CEO, and wNR security can be proven, while the proof for MBS security given for unmodified HuFu still applies.

[8] If $\gcd(x_{2,i}) = 1$, then $\langle x_2 \rangle$ is saturated. Equivalently, $\mathbb{Z}^m / \langle x_2 \rangle$ is free, hence x_2 is part of a basis, on which $\overline{B}$ can be defined according to the requirement.

[9] Near-collision resistance is a stronger form of collision resistance, where it is even hard to find inputs whose hash values are close (with respect to some norm).

$\texttt{Verify}(\texttt{pk}, \texttt{msg}, \texttt{sig})$

1 : $(\texttt{salt}, s) \leftarrow \texttt{sig}$
2 : $h \leftarrow \texttt{H}(\texttt{msg}||\texttt{salt})$
3 : $s' \leftarrow \texttt{Decompress}(s)$
4 : $c \leftarrow s' + h$
5 : **if** $c_n = \sum_{i=1}^{n-1} v_i \cdot c_i \mod p \quad \forall p \in P_\Delta$
6 : **if** $\|s'\|^2 \leq \lfloor \vartheta^2 \rfloor$
7 : **return** 1
8 : **return** 0

Fig. 8: Verification algorithm of SQUIRRELS.

5.2 Squirrels

SQUIRRELS incorporates a GPV-like approach. It is based on unstructured lattices and uses lattices modulo various distinct primes simultaneously. The public key is composed of a single vector which is used to check if a target is contained in the lattice modulo each of the primes. Let n and q be positive integers. The target determinant is denoted by $\Delta = \prod_{p \in P_\Delta} p$, for P_Δ a set of primes in $[2^{30}, 2^{31}]$. The hash function $\texttt{H}$ maps to $[0, \ldots, q-1]^{n-1} \times \{0\}$ viewed as an element in $\mathbb{Z}^n$ with last component being 0.

Key Pair. The SQUIRRELS secret key consists of a matrix $B \in \mathbb{Z}^{n \times n}$, which, by design, has a Hermite normal form

$$\mathrm{HNF}(B) = \begin{pmatrix} I_{n-1} & v_i^\mathsf{T} \\ 0 & \Delta \end{pmatrix}.$$

The resulting vector $v \coloneqq (v_i)_{i=1,\ldots,n-1}$ is the public key.

Signature. The signature of a message $\texttt{msg}$ for a public key v consists of $(\texttt{salt}, s)$ where $\texttt{salt}$ is a random string and $s = \texttt{Compress}(s')$ with $s' \in \mathbb{Z}^n$.

Verify. Given a public key $\texttt{pk} = v$, a message $\texttt{msg}$, and a signature $\texttt{sig} = (\texttt{salt}, s)$, the verification algorithm is described in Fig. 8.

In the analysis below, we write $c' \coloneqq (c_1, \ldots, c_{n-1})^\mathsf{T}$ for $c = (c_1, \ldots, c_n)^\mathsf{T}$ and $\langle \cdot, \cdot \rangle$ for the standard inner product. Note that in the search for elements $v \in \mathbb{Z}^{n-1}$ that satisfy a certain algebraic condition modulo Δ, it suffices to give $v \mod p$ for each $p \in P_\Delta$, by making use of the Chinese Remainder Theorem. We make use of this argument, without explicitly stating it again.

S-CEO. Given a public key $\texttt{pk} = v$, a message $\texttt{msg}$ and a signature $\texttt{sig} = (\texttt{salt}, s)$ such that $\texttt{Verify}(\texttt{pk}, \texttt{msg}, \texttt{sig}) = 1$, we need to find a distinct public key $\overline{\texttt{pk}} = \overline{v}$ such that $\texttt{Verify}(\overline{\texttt{pk}}, \texttt{msg}, \texttt{sig}) = 1$. This translates to finding $\overline{v}$, which is in the kernel of $\langle c', \cdot \rangle - c_n : \mathbb{F}_p^{n-1} \to \mathbb{F}_p$ for all $p \in P_\Delta$. Note that $\dim_{\mathbb{F}_p}(\ker(\langle c', \cdot \rangle) - c_n) = n - 2$. Hence, for each p one can find an element $\overline{v}_p$ such that $\langle c', \overline{v}_p \rangle - c_n = 0 \mod p$. Then, $\overline{\texttt{pk}} = \overline{v}$ is given by the $\overline{v}_p$.

S-DEO. Given a public key $\mathtt{pk} = v$, a message $\mathtt{msg}$, and a signature $\mathtt{sig} = (\mathtt{salt}, s)$ such that $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$, we need to find a second public key $\overline{\mathtt{pk}} = \overline{v} \neq v$ and a second message $\overline{\mathtt{msg}} \neq \mathtt{msg}$ such that $\mathtt{Verify}(\overline{\mathtt{pk}}, \overline{\mathtt{msg}}, \mathtt{sig}) = 1$. For this, we choose a random $\overline{\mathtt{msg}} \neq \mathtt{msg}$ and compute $\overline{c} = s' + \mathtt{H}(\overline{\mathtt{msg}}||\mathtt{salt})$. Hence it is left to find $\overline{v}$ such that $\langle \overline{v}, \overline{c}' \rangle - \overline{c}_n = 0 \mod p$ holds for all $p \in P_\Delta$. For this, the same argument as for the S-CEO attack applies.

MBS. One needs to find a public key $\mathtt{pk} = v$, two distinct messages $\mathtt{msg} \neq \overline{\mathtt{msg}}$, and a signature $\mathtt{sig} = (\mathtt{salt}, s)$ such that $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$ and $\mathtt{Verify}(\mathtt{pk}, \overline{\mathtt{msg}}, \mathtt{sig}) = 1$. For this, we choose s' such that $\|s'\|^2 < \lfloor \vartheta^2 \rfloor$ holds and compute $s = \mathtt{Compress}(s')$. We then set $\mathtt{sig} = (\mathtt{salt}, s)$ for some randomly chosen $\mathtt{salt}$. Further we consider two random messages $\overline{\mathtt{msg}} \neq \mathtt{msg}$ and compute $c = s' + \mathtt{H}(\mathtt{msg}||\mathtt{salt})$ and $\overline{c} = s' + \mathtt{H}(\overline{\mathtt{msg}}||\mathtt{salt})$. Hence it is left to find v such that $\langle v, c' \rangle - c_n = 0 \mod p$ and $\langle v, \overline{c}' \rangle - \overline{c}_n = 0 \mod p$ holds for all $p \in P_\Delta$. Consider for $p \in P_\Delta$ the map

$$f : \mathbb{F}_p^{n-1} \to \mathbb{F}_p^2,\ x \mapsto (\langle x, c' \rangle, \langle x, \overline{c}' \rangle)$$

and observe that $\dim_{\mathbb{F}_p}(\ker(f)) = n - 3$. Hence, we can find v_p with the desired properties, which constitutes v.

Remark 8. In the SQUIRRELS specification it is claimed that MBS security is fulfilled, which the above disproves. While their claim is based on the similarity to FALCON, the MBS security of FALCON still holds. The subtle differences between SQUIRRELS and FALCON are thus important, when it comes to BUFF security.

wNR. Given $\mathtt{pk} = v$, and a signature $(\mathtt{salt}, s)$ which verifies an unknown message $\mathtt{msg}$, we can find a new public key $\overline{\mathtt{pk}} = \overline{v}$ and a new signature $(\mathtt{salt}, \overline{s})$ that verifies $\mathtt{msg}$ as follows. Let $s' = \mathtt{Decompress}(s)$. We can assume that with large probability, s'_n is divisible by a prime ϖ which is not in P_Δ. E.g., if s'_n is close to uniform, it will be even with about 0.5 probability. In this case, we set $\overline{s}'_n \coloneqq \varpi^{-1} s'_n$. Further, we set $\overline{v}_p \coloneqq \varpi^{-1} v_p$ for each p and let $\overline{v} \in \mathbb{Z}^{n-1}$ be the corresponding vector over $\mathbb{Z}$. Choosing $\overline{s}'_i = s'_i$ for $i = 1, \ldots, n-1$, and $\overline{\mathtt{salt}} = \mathtt{salt}$ yields a new public key $\overline{v}$ and signature $(\overline{\mathtt{salt}}, \overline{s})$, with $\overline{s} = \mathtt{Compress}(\overline{s}')$ that verifies the unknown message. Indeed, the hash h did not change by the procedure and for each $p \in P_\Delta$, we have $\sum_{i=1}^{n-1} \overline{v}_{i,p} c_{i,p} = \varpi^{-1} \sum_{i=1}^{n-1} v_{i,p} c_{i,p} = \varpi^{-1} c_{n,p} = \overline{s}'_n = c_{n,p}$ using that $h_n = 0$. Thereby the verification succeeds.

Remark 9. Modifying SQUIRRELS to incorporate the PS-3 transform (i.e., replacing $h \leftarrow \mathtt{H}(\mathtt{msg}||\mathtt{salt})$ by $h \leftarrow \mathtt{H}(\mathtt{msg}||\mathtt{salt}, \mathtt{pk})$) does not suffice to achieve full BUFF security. This is the case, as we can still find S-CEO/S-DEO attacks that are successful with probability greater than $\frac{1}{2^{31}}$: As above we can reduce to the case of a single $p \in P_\Delta$. While the above change to the scheme prevents an attacker to choose v in the kernel of $\langle c', \cdot \rangle - c_n : \mathbb{F}_p^{n-1} \to \mathbb{F}_p$, the probability that this holds for a random v is equal to $\frac{1}{p} \geq \frac{1}{2^{31}}$ (finding v by randomly hitting an element from a subset of size p^{n-2} contained in a set of size p^{n-1}).

5.3 Further Lattice Candidates

The remaining NIST candidates based on lattices are HAETAE and RACCOON. Both use the Fiat-Shamir with aborts framework and are based on the module versions of the learning with errors and short integer solution problems. Both schemes are similar to DILITHIUM and their BUFF analyses are analogous to the DILITHIUM analysis in [16]. In short, HAETAE signs the hash of public key and message and appends a hash value generated (among other inputs) from public key and message to the signature. Thus, HAETAE can be considered to use the BUFF transform, and if we assume the used hash function to be collision-resistant and ϕ-non-malleable (as defined in [16]), we obtain BUFF security by [16, Theorem 5.5]. This is also true for RACCOON, which is structurally very similar to DILITHIUM and hence can be viewed to implement the BUFF transform.

6 Multivariate schemes

In this section we analyze the signatures that belong to the family of multivariate (MV) schemes. After introducing the foundations and basic properties, we will give a short generic BUFF analysis, i.e., present results that hold for (nearly) all MV schemes under consideration. After this, we turn to the scheme-specific analyses: UOV, which is the basis of all remaining candidates, is treated in Section 6.1. This is followed by the analysis of MAYO in Section 6.2. While MAYO is based on UOV, its polynomials are constructed in a way that makes the analysis more involved. We present the details to show that despite the complex structure of the public key, MAYO dose not achieve full BUFF security. Both UOV and MAYO—and all MV schemes considered in this paper, except PROV—fulfill MBS as the only BUFF notion. The analysis of PROV, which achieves full BUFF security, follows in Section 6.3. For the remaining schemes QR-UOV, SNOVA, TUOV, and VOX, the BUFF analyses are similar to the one given for UOV. We provide a short outline for each scheme in Section 6.4.

Background and notation of MV schemes. The main object in multivariate cryptography is a multivariate quadratic map $\mathcal{P}\colon \mathbb{F}_q^n \to \mathbb{F}_q^m$, which consists of m homogeneous quadratic polynomials $(p^{(1)}(x), \ldots, p^{(m)}(x))$ in n variables $x = (x_1, \ldots, x_n)$. The coefficients of each of these quadratic polynomials $p^{(k)}(x)$ can be stored in a matrix $P^{(k)}$, where the (i,j)-th entry $(p^{(k)})_{i,j}$ represents the coefficient of the monomial $x_i x_j$. Thus, $p^{(k)}(x)$ can be evaluated as $x^\top P^{(k)} x$.

The task of finding a preimage $s \in \mathbb{F}_q^n$ for a given target vector $t \in \mathbb{F}_q^m$ under a given multivariate quadratic map $\mathcal{P}$ is hard in general, as it amounts to solving a system of multivariate quadratic equations, known as the $\mathcal{MQ}$-Problem. Consequently, a trapdoor needs to be included in the map $\mathcal{P}$, that allows to find such $s \in \mathbb{F}_q^m$ with $\mathcal{P}(s) = t$, which constitutes the signature `sig`. The precise realization of this trapdoor varies from scheme to scheme.

Generic BUFF analysis of MV schemes. In the following, we provide the parts of the BUFF analysis that are the same for (nearly) all multivariate schemes under consideration—namely the MBS proof and wNR attack. The arguments for these two notions will hence not be repeated in the scheme-specific sections. Furthermore, we provide a generic result on the BUFF security of the considered MV schemes using the PS-3 transform.

MBS security for MV schemes. Since the target vector $t \in \mathbb{F}_q^n$ is computed as the hash of (at least) the message $\mathtt{msg}$, multivariate schemes naturally satisfy MBS. It is not possible that a single signature $\mathtt{sig} = s$ verifies different messages $\mathtt{msg} \neq \overline{\mathtt{msg}}$, because $\mathtt{H}(\mathtt{msg}||\cdot) = \mathcal{P}(s) = \mathtt{H}(\overline{\mathtt{msg}}||\cdot)$ would imply a collision of $\mathtt{H}$.

wNR attack against MV schemes. For an wNR attack, one is given a public key $\mathtt{pk}$, from which we derive the public map $\mathcal{P}$, and a signature $\mathtt{sig} = s$ to an unknown message $\mathtt{msg}$, and has to find $\overline{\mathtt{pk}} \neq \mathtt{pk}$ and $\overline{\mathtt{sig}} = \overline{s}$ such that $\mathcal{P}(s) = \mathtt{H}(\mathtt{msg}||\cdot) = \overline{\mathcal{P}}(\overline{s})$. Firstly, note that $\mathcal{P}(s) = t = \mathtt{H}(\mathtt{msg}||\cdot)$ can be computed without knowing $\mathtt{msg}$, as s is a valid signature. Next, we generate a key pair $(\overline{\mathtt{sk}}, \overline{\mathtt{pk}})$ with $\overline{\mathtt{pk}} \neq \mathtt{pk}$ and use it to sign the target vector t. This results in a signature $\overline{s}$ that fulfills $\overline{\mathcal{P}}(\overline{s}) = t = \mathtt{H}(\mathtt{msg}||\cdot) = \mathcal{P}(s)$.

Note that this attack is not applicable for Prov, as it hashes the whole public key alongside the message, which prevents us from being able to compute the target *before* choosing the second public key $\overline{\mathtt{pk}}$. We give a proof for wNR security of Prov in Section 6.3. For all other schemes under consideration the above attack works, however, for Vox and Snova some extra care is necessary, as both schemes hash parts of the public key alongside the message. In Vox the public key consists of a seed SeedPub and the quadratic map Pub, which is generated using SeedPub. By modifying the seed for the secret key while keeping SeedPub the same, we get a new quadratic map $\overline{\mathrm{Pub}} \neq \mathrm{Pub}$. The new secret key is known to the adversary and can be used to sign to the same target. In Snova the public key is of the form $(s_{\mathrm{public}}, (P_i^{22})_{i\in[m]})$. Here, s_{public} is a seed used to generate the remaining components of the public map $\mathcal{P}$, which is done in the signing and verification algorithm. Choosing $\overline{s}_{\mathrm{public}} = s_{\mathrm{public}}$ and $\overline{\mathtt{sk}} \neq \mathtt{sk}$ guarantees $(\overline{P}_i^{22})_{i\in[m]} \neq (P_i^{22})_{i\in[m]}$ and yields a key pair $(\overline{\mathtt{sk}}, \overline{\mathtt{pk}})$ for which we can apply the above attack.

BUFF security using PS-3 *transform.* Our analysis reveals that from the family of multivariate schemes only Prov satisfies full BUFF security. The main design feature that contributes to this is the hashing of the public key alongside the message. As all multivariate schemes considered in this paper verify signatures by comparing $\mathtt{H}(\mathtt{msg}, \cdot)$ to $\mathcal{P}(s)$, we can achieve BUFF security for all of them, by adding the *complete*[10] public key alongside the message into the hash function. To prove this, the same arguments as for Prov apply—note that in the analysis of Prov, we use little scheme-specific details except for the size of the domain of $\mathcal{P}$. This approach is very similar to applying the PS-3 transform, except for the fact

[10] Vox and Snova hash parts of the public key, which is insufficient for BUFF security.

that an application of PS-3 would result in an additional hash computation (see Fig. 2) that can be avoided by modifying the existing computation of $\mathtt{H}(\mathtt{msg}||\cdot)$ instead. In the following we write PS-3, but it is understood that the simpler modification described above is applied if possible.

Proposition 10. *For* $\Sigma \in \{\text{MAYO}, \text{QR-UOV}, \text{SNOVA}, \text{TUOV}, \text{UOV}, \text{VOX}\}$ *and* $\mathtt{H}$ *a random oracle, the transformed scheme* PS-3$[\mathtt{H}, \Sigma]$ *fulfills BUFF security.*

6.1 UOV

The unbalanced oil and vinegar (UOV) signature scheme is the oldest candidate and the foundation of the remaining multivariate schemes, [34,30]. The trapdoor information in UOV is a secret linear m-dimensional subspace, the so-called oil space O, which is annihilated by the public key map $\mathcal{P}$, i.e., $\mathcal{P}(o) = 0$ for all $o \in O$. The dimension of the oil space m needs to equal the number of quadratic equations and the number of variables n usually satisfies $n \approx 2.5m$. We introduce the algorithms of classic UOV here, instead of the compressed versions. The analysis holds for all variants similarly, as we argue below.

Key Pair. The public key $\mathtt{pk} = \{P_i\}_{i\in[m]}$ consists of m matrices

$$P_i = \begin{pmatrix} P_i^{(1)} & P_i^{(2)} \\ 0 & P_i^{(3)} \end{pmatrix},$$

where $P_i^{(1)} \in \mathbb{F}_q^{v\times v}, P_i^{(2)} \in \mathbb{F}_q^{v\times m}$ and $P_i^{(3)} \in \mathbb{F}_q^{m\times m}$. Here, the matrices $P_i^{(1)}$ and $P_i^{(2)}$ are generated randomly from a seed and $P_i^{(3)}$ is computed via

$$P_i^{(3)} = -O^\top P_i^{(1)} O - O^\top P_i^{(2)},$$

with a randomly generated oil space $O \in \mathbb{F}_q^{v\times m}$.

The secret key $\mathtt{sk} = (\mathtt{seed}_{\mathtt{sk}}, O, \{P_i^{(1)}, S_i\}_{i\in[m]})$ consists of a seed $\mathtt{seed}_{\mathtt{sk}}$, the oil space O, a part of the public key matrices $\{P_i^{(1)}\}_{i\in[m]}$, and some auxiliary matrices $\{S_i\}_{i\in[m]}$ needed for signing, given by $S_i = (P_i^{(1)} + P_i^{(1)\top})O + P_i^{(2)}$.

Signature. The signature is given by $\mathtt{sig} = (s, \mathtt{salt})$, containing a vector $s \in \mathbb{F}_q^n$ and a random $\mathtt{salt}$.

Verify. Given a public key $\mathtt{pk} = (P_i^{(1)}, P_i^{(2)}, P_i^{(3)})$, a message $\mathtt{msg}$, and a signature $\mathtt{sig} = (s, \mathtt{salt})$, the verification algorithm is shown in Fig. 9.

S-CEO. Given a public key $\mathtt{pk}$, a message $\mathtt{msg}$, and a signature $\mathtt{sig} = (s, \mathtt{salt})$ such that $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$, we need to find a second public key $\overline{\mathtt{pk}} \neq \mathtt{pk}$ such that for the corresponding public key map $\overline{\mathcal{P}}$ it holds that $\overline{\mathcal{P}}(s) = \mathtt{H}(\mathtt{msg}||\mathtt{salt})$. Let $p_{i,j}^{(k)}$ be the (i,j)-th entry of the public key matrix P_k coming

$\mathsf{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig})$

$\mathcal{P} \leftarrow \mathsf{Map}(\mathtt{pk})$
$t \leftarrow \mathsf{Target}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig})$
if $\mathcal{P}(\mathtt{sig}) = t$
  return 1
return 0

$\mathsf{Target}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig})$

$(\cdot, \mathtt{salt}) \leftarrow \mathtt{sig}$
$t \leftarrow \mathtt{H}(\mathtt{msg}||\mathtt{salt})$ // UOV
$t \leftarrow \mathtt{H}(\mathtt{H}(\mathtt{msg})||\mathtt{salt})$ // MAYO
$t \leftarrow \mathtt{H}_2(\mathtt{H}_1(\mathtt{pk})||\mathtt{msg}||\mathtt{salt})$ // PROV
return t

$\mathsf{Map}(\mathtt{pk})$ in UOV

$(P_i^{(1)}, P_i^{(2)}, P_i^{(3)}) \leftarrow \mathtt{pk}$
$P_i \leftarrow \begin{pmatrix} P_i^{(1)} & P_i^{(2)} \\ 0 & P_i^{(3)} \end{pmatrix}$
return $\mathcal{P}$

$\mathsf{Map}(\mathtt{pk})$ in PROV

$(\mathtt{seed}_{\mathtt{pk}}, (P_i^{(3)})) \leftarrow \mathtt{pk}$
$(P_i^{(1)}, P_i^{(2)}) \leftarrow \mathtt{E}(\mathtt{seed}_{\mathtt{pk}})$
$P_i = \begin{pmatrix} P_i^{(1)} & P_i^{(2)} \\ 0 & P_i^{(3)} \end{pmatrix}$
return $\mathcal{P}$

$\mathsf{Map}(\mathtt{pk})$ in MAYO

$(\mathtt{seed}_{\mathtt{pk}}, \{P_i^{(3)}\}) \leftarrow \mathtt{pk}$
$\{P_i^{(1)}, P_i^{(2)}\} \leftarrow \mathtt{E}(\mathtt{seed}_{\mathtt{pk}})$
return $\mathcal{P}^*$

Fig. 9: Verification algorithm of UOV, MAYO, and PROV. Recall that the public map $\mathcal{P}$ consists of m homogeneous quadratic polynomials $(p^{(1)}(x), \ldots, p^{(m)}(x))$, and can be computed from $P_1, \ldots, P_k$ using the relation $p^i(x) = x^\top P_i x$. For MAYO the larger map $\mathcal{P}^*$ is used, which can be computed from $\mathcal{P}$ as described in Equation (2). Lastly, note that $\mathtt{E}(\cdot)$ is used as an abbreviation for $\mathtt{Expand}(\cdot)$.

from $\mathtt{pk}$. We define $\overline{p}_{i,j}^{(k)}$, the coefficients of $\overline{P}_k$ from $\overline{\mathtt{pk}}$ as $p_{i,j}^{(k)}$, except for the following adjustment. We pick an arbitrary $i \in [v+1, n-1]$ and change $\overline{p}_{i,i}^{(k)}$ and $\overline{p}_{i+1,i+1}^{(k)}$ s.t. $p_{i,i}^{(k)} s_i^2 + p_{i+1,i+1}^{(k)} s_{i+1}^2 = \overline{p}_{i,i}^{(k)} s_i^2 + \overline{p}_{i+1,i+1}^{(k)} s_{i+1}^2$. Keeping all other coefficients, we get $\overline{P}_k(s) = P_k(s)$ for all k, hence verification succeeds for $\overline{\mathtt{pk}}$.

S-DEO. Given a public key $\mathtt{pk}$, a message $\mathtt{msg}$, and a signature $\mathtt{sig} = (s, \mathtt{salt})$ such that $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$, we need to find a second public key $\overline{\mathtt{pk}} \neq \mathtt{pk}$ and a second message $\overline{\mathtt{msg}} \neq \mathtt{msg}$ such that $\overline{\mathcal{P}}(s) = \overline{h} = (\overline{h}_k) = \mathtt{H}(\overline{\mathtt{msg}}||\mathtt{salt})$. We take some index $l \in [v+1, n]$, with $s_l \neq 0$. For each $k \in [m]$, set $\overline{p}_{i,i}^k = (\overline{h}_k - \sum_{i<j,(i,j)\neq(l,l)} p_{i,j}^k s_i s_j)/(s_l^2)$. Then we found $\overline{\mathcal{P}}$ with $\overline{\mathcal{P}}(s) = \overline{h}$.

Variants. The statements also hold for the compressed variants pkc and pkc+skc. For these, the public key does not consist of the matrices $\{P_i\}_{i\in[m]}$, but only of the submatrices $\{P_i^{(3)}\}_{i\in[m]}$ and a seed that is used to generate $\{P_i^{(1)}\}_{i\in[m]}$ and $\{P_i^{(2)}\}_{i\in[m]}$. The results of our analysis only require a change of the $P_i^{(3)}$, so that the attacks work for the compressed versions as well.

6.2 MAYO

In MAYO, the public key map $\mathcal{P}$ has the same structure as in UOV, but it is publicly whipped up to a k-fold larger map $\mathcal{P}^* : \mathbb{F}_q^{kn} \to \mathbb{F}_q^m$ via

$$\mathcal{P}^*(s_1, \ldots, s_k) = \sum_{i=1}^{k} E_{ii}\mathcal{P}(s_i) + \sum_{i=1}^{k}\sum_{j=i+1}^{k} E_{ij}\mathcal{P}'(s_i, s_j)\,, \tag{2}$$

where $E_{ij} \in \mathbb{F}_q^{m\times m}$ are system parameters and $\mathcal{P}'$ is the bilinear map associated to $\mathcal{P}$, i.e., component-wise $P'_l(s_i, s_j) = s_i^\top(P_l + P_l^\top)s_j$, for each l. The benefit of this approach is a smaller public key size at the expense of a slightly larger signature and an additional security assumption: the *Multi-Target Whipped MQ* problem [6, Section 5.1].

Key Pair. The secret key is given by a private seed $\mathtt{sk} = \mathtt{seed}_{\mathtt{sk}}$. It is used to derive a public seed $\mathtt{seed}_{\mathtt{pk}}$ and the secret linear oil space $O \in \mathbb{F}_q^{(n-o)\times o}$. The public key is given by $\mathtt{pk} = (\mathtt{seed}_{\mathtt{pk}}, \{P_i^{(3)}\}_{i\in[m]})$, where

$$P_i^{(3)} = -O^\top P_i^{(1)} O - O^\top P_i^{(2)} \in \mathbb{F}_q^{o\times o}\,.$$

Hereby, $P_i^{(1)} \in \mathbb{F}_q^{(n-o)\times(n-o)}$ and $P_i^{(2)} \in \mathbb{F}_q^{(n-o)\times o}$ are expanded from $\mathtt{seed}_{\mathtt{pk}}$.

Signature. The signature is given by $\mathtt{sig} = (s_1, \ldots, s_k, \mathtt{salt})$, with $s_i \in \mathbb{F}_q^n$.

Verify. Given a public key $\mathtt{pk} = (\mathtt{seed}_{\mathtt{pk}}, \{P_i^{(3)}\}_{i\in[m]})$, a message $\mathtt{msg}$, and a signature $\mathtt{sig} = (s_1, \ldots, s_k, \mathtt{salt})$, the verification is shown in Fig. 9.

S-CEO. Given a public key $\mathtt{pk} = (\mathtt{seed}_{\mathtt{pk}}, \{P_i^{(3)}\}_{i\in[m]})$, a message $\mathtt{msg}$, and a signature $\mathtt{sig} = (s_1, \ldots, s_k, \mathtt{salt})$, such that $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$, we need to find a second public key $\overline{\mathtt{pk}} \neq \mathtt{pk}$ such that $\overline{\mathcal{P}}^*(s_1, \ldots, s_k) = t = \mathtt{H}(\mathtt{H}(\mathtt{msg})||\mathtt{salt})$ holds, where $\overline{\mathcal{P}}^*$ is derived from $\overline{\mathtt{pk}}$. The main observation to tackle this task, is that the map $\mathcal{P}^*$ is linear with respect to the entries of its corresponding public key matrices P_i.

The strategy is now to generate various $\tilde{\mathtt{pk}}_a$, where we always use the same $\mathtt{seed}_{\mathtt{pk}}$, but randomly generated $(\{P_{i,a}^{(3)}\}_{i\in[m]})_a$ for $a \in \{1, 2, \ldots\}$. Denote by $\tilde{\mathcal{P}}_a$ the quadratic map associated to this public key. Then, we consecutively compute $\tilde{\mathcal{P}}_a^*(s_1, \ldots, s_k) = x_a$ until we gathered m linearly independent vectors x_a. Thus, we find $\lambda_a \in \mathbb{F}_q$, such that $t = \sum_{a=1}^m \lambda_a \cdot x_a$. Now we add up the randomly generated matrices accordingly and define $\overline{pk} = (\mathtt{seed}_{pk}, \{\overline{P}_i^{(3)}\}_{i\in[m]})$, where $\overline{P}_i^{(3)} \coloneqq \sum_{a=1}^m \lambda_a (P_i^{(3)})_a$ for all $i \in [m]$. Due to the linearity we have

$$\overline{\mathcal{P}}^*(s_1, \ldots, s_k) = \sum_{a=1}^{m} \lambda_a \tilde{\mathcal{P}}_a^*(s_1, \ldots, s_k) = \sum_{a=1}^{m} \lambda_a x_a = t\,.$$

Thus, an attacker is able to find a different public key $\overline{\mathtt{pk}} \neq \mathtt{pk}$, such that $\mathtt{Verify}(\overline{\mathtt{pk}}, \mathtt{msg}, \mathtt{sig}) = 1$ and MAYO is not S-CEO-secure.

S-DEO. Given a public key $\mathtt{pk} = (\mathtt{seed}_{\mathtt{pk}}, \{P_i^{(3)}\}_{i\in[m]})$, a message $\mathtt{msg}$, and a signature $\mathtt{sig} = (s_1, \ldots, s_k, \mathtt{salt})$ such that $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$, we need to find a second public key $\overline{\mathtt{pk}} \neq \mathtt{pk}$ and message $\overline{\mathtt{msg}} \neq \mathtt{msg}$ such that $\overline{\mathcal{P}}^*(s_1, \ldots, s_k) = t = \mathtt{H}(\mathtt{H}(\overline{\mathtt{msg}})||\mathtt{salt})$. Since the vectors x_a we generated in the S-CEO analysis give a basis for the complete vector space $\mathbb{F}_q^m$, an attacker can compute $\bar{t} = \mathtt{H}(\mathtt{H}(\overline{\mathtt{msg}})||\mathtt{salt})$ and find $\overline{\lambda}_a$ such that $\bar{t} = \sum \overline{\lambda}_a x_a$ for some randomly chosen message $\overline{\mathtt{msg}} \neq \mathtt{msg}$. Thus, the same attack that was developed to analyze S-CEO, works here and MAYO is not S-DEO-secure.

6.3 PROV

Key Pair. Let $\mathbb{F}$ denote the finite field $\mathbb{F}_{2^8}$ and $\delta := o - m$. The public key $\mathtt{pk}$ is a pair $(\mathtt{seed}_{\mathtt{pk}}, (P_i^{(3)})_{i\in[m]})$ where $P_i^{(3)} \in \mathbb{F}^{(m+\delta)\times(m+\delta)}$ for all i. From $\mathtt{seed}_{\mathtt{pk}}$ the matrices $(P_i^{(1)}, P_i^{(2)})_{i\in[m]}$ with $P_i^{(1)} \in \mathbb{F}^{(n-m-\delta)\times(n-m-\delta)}$ and $P_i^{(2)} \in \mathbb{F}^{(n-m-\delta)\times(m+\delta)}$ for all i, are generated. We denote by P_i the matrix

$$\begin{pmatrix} P_i^{(1)} & P_i^{(2)} \\ 0 & P_i^{(3)} \end{pmatrix}.$$

The secret key is the triple $(\mathtt{seed}_{\mathtt{pk}}, \mathtt{seed}_{\mathtt{sk}}, \mathtt{H}(\mathtt{pk}))$. From $\mathtt{seed}_{\mathtt{sk}}$ the matrix $O \in \mathbb{F}^{(n-m-\delta)\times(m+\delta)}$ is generated.

Signature. A signature is given by $\mathtt{sig} = (\mathtt{salt}, s)$ for $s \in \mathbb{F}^n$.

Verify. Given a public key $\mathtt{pk} = (\mathtt{seed}_{\mathtt{pk}}, (P_i^{(3)})_{i\in[m]})$, a message $\mathtt{msg}$, and a signature $\mathtt{sig} = (\mathtt{salt}, s)$, the verification is shown in Fig. 9.

S-CEO. Given a public key $\mathtt{pk}$, a message $\mathtt{msg}$, and a signature $\mathtt{sig} = (\mathtt{salt}, s)$ such that $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$, we need to find a different public key $\overline{\mathtt{pk}} = (\overline{\mathtt{seed}}_{\mathtt{pk}}, (\overline{P}_i^3)_{i\in[m]})$ such that $(\bar{t}_i)_{i\in[m]} = \overline{h}$ and hence $(s^\top \overline{P}_i s)_{i\in[m]} = \mathtt{H}_2(\mathtt{H}_1(\overline{\mathtt{pk}})||\mathtt{msg}||\mathtt{salt})$. As both sides of the latter equation depend on $\overline{\mathtt{pk}}$ and the value on the right is random (assuming $\mathtt{H}_1$ and $\mathtt{H}_2$ to be random oracles), the probability to find a suitable $\overline{\mathtt{pk}}$ is $\frac{1}{|\mathbb{F}^m|} = \frac{1}{2^{8m}} \leq 2^{-368}$, for all proposed variants.

S-DEO. Given a public key $\mathtt{pk}$, a message $\mathtt{msg}$, and a signature $\mathtt{sig} = (\mathtt{salt}, s)$ such that $\mathtt{Verify}(\mathtt{pk}, \mathtt{msg}, \mathtt{sig}) = 1$, we need to find a second public key $\overline{\mathtt{pk}} \neq \mathtt{pk}$ and a second message $\overline{\mathtt{msg}} \neq \mathtt{msg}$ such that $\mathtt{Verify}(\overline{\mathtt{pk}}, \overline{\mathtt{msg}}, \mathtt{sig}) = 1$. This is not feasible by the same argument that was used for S-CEO security, as changing the message only influences the hash value $\overline{h} = \mathtt{H}_2(\mathtt{H}_1(\overline{\mathtt{pk}})||\mathtt{msg}||\mathtt{salt})$.

wNR. Given a public key $\mathtt{pk}$ and a signature $\mathtt{sig}$ to an unknown message $\mathtt{msg}$, one has to find another public key $\overline{\mathtt{pk}} \neq \mathtt{pk}$, and a signature $\overline{\mathtt{sig}}$ such that $\mathtt{Verify}(\overline{\mathtt{pk}}, \mathtt{msg}, \overline{\mathtt{sig}}) = 1$. This is not feasible as one would have to find $\overline{\mathtt{pk}}$ such that $(\bar{t}_i)_{i\in[m]} = \overline{h}$, where $\overline{h} = \mathtt{H}_2(\mathtt{H}_1(\overline{\mathtt{pk}})||\mathtt{msg}||\overline{\mathtt{salt}})$ is unknown as $\mathtt{msg}$ is. Note that we can compute $h = (s^\top P_i s)_i$ but not $\mathtt{msg}$ and hence not $\overline{h}$. Thus, the probability for the equality $(\bar{t}_i)_{i\in[m]} = \overline{h}$ to hold is $\frac{1}{2^{8m}}$ and therefore less than 2^{-368} for all variants.

6.4 Further Multivariate Candidates

The remaining NIST signature candidates based on multivariate polynomials are QR-UOV, SNOVA, TUOV and VOX. For all of them, the BUFF analysis follows the same idea as the one given for UOV in Section 6.1. We provide a short overview over the main arguments in the following.

The main difference between QR-UOV and UOV is that the public key matrices $P_1^{(i)}, P_2^{(i)}$, and $P_3^{(i)}$ of QR-UOV are block matrices, where each component $\Phi_g^f \in \mathbb{F}_q^{l\times l}$ corresponds to an element g of the quotient ring $\mathbb{F}_q[x]/(f)$, with an irreducible polynomial $f \in \mathbb{F}_q[x]$ of degree l. The polynomial matrices of the subalgebra $\mathcal{A}_f := \{\Phi_g^f \in \mathbb{F}_q^{l\times l} \mid g \in \mathbb{F}_q[x]/(f)\}$ are defined entry-wise such that $(\Phi_g^f)_{ij}$ is the coefficient of x^{i-1} in $x^{j-1} \cdot g$. In the S-CEO/S-DEO analysis for QR-UOV we cannot modify single entries $p_{i,j}^{(k)}$ of the matrices $P_k^{(3)}$ that were used to control the values $y_k = s^\top P_k s$ in the analysis of UOV. Instead, we can only alter one coefficient (or more) of the polynomials $g = \sum_{i=0}^{l-1} a_i x^i \in \mathcal{A}_f$ that are stored in the $P_k^{(3)}$ part of the public key $\mathtt{pk}$. This will change l values in the corresponding block $\Phi_g^f \in \mathbb{F}_q^{l\times l}$ of $P_i^{(3)}$. However, we can still dictate the result $r_k = s_l^\top \Phi_g^f s_l$ by choosing the coefficients of g accordingly. Here s_l denote the l entries of the vector $s \in \mathbb{F}^n$ that are multiplied with this block.

SNOVA differs from UOV in the fact that it works over the non-commutative ring $R = \mathbb{F}_q^{l\times l}$ instead of $\mathbb{F}_q$. Further, SNOVA computes the target vector as $t = \mathtt{H}(\text{seed}_p||\mathtt{H}(\mathtt{msg})||\mathtt{salt})$ for $\mathtt{pk} = (\text{seed}_p, \{P_i^{22}\}_{i\in[m]})$ with $P_i^{22} \in R^{o\times o}$, while for UOV we have $t = \mathtt{H}(\mathtt{msg}||\mathtt{salt})$. However, for neither of the BUFF security notions, the adversary has to provide honestly generated keys, hence it can choose two different public keys $\overline{\mathtt{pk}} \neq \mathtt{pk}$ that have the same seed, which then result in the same target t. Then, S-CEO and S-DEO insecurity follows by using the concrete parameters provided in SNOVA to prove systems of equations solvable.

The TUOV analysis is completely analogous to the UOV analysis, as the additional affine transformation $S\colon \mathbb{F}_q^m \to \mathbb{F}_q^m$ has no impact on the analysis.

VOX is a UOV-based scheme that incorporates the quotient ring technique. Despite their claim to achieve BUFF security, VOX only satisfies MBS. S-CEO and S-DEO can be attacked as in UOV. In short the attack proceeds as follows: One keeps the part SeedPub of the public key $\mathtt{pk} = (\text{SeedPub}, \text{Pub})$ unchanged. The Pub part can be changed independently and is chosen as in the attack against UOV. Note that VOX uses the quotient ring technique, however, the problem of defining Pub is still the same as in UOV, just over the extension field. The wNR security can be attacked as described at the beginning of this section.

7 Conclusion

In this work, we analyzed the signature schemes based on codes, isogenies, lattices, and multivariate polynomials submitted to the additional NIST PQC standardization effort for signatures regarding their BUFF security. Besides the analysis of the original schemes, we included comments on the BUFF security after a light

transform, the so-called PS-3 transform. In fact, we see that often, the PS-3 transform suffices to ensure BUFF security, despite the fact that the PS-3 transform is not sufficient for generic schemes. This gap between the general statement and the empirical evidence on practical schemes can be analyzed further.

In the NIST competition, there are even more signature schemes, which we have not analyzed in this work. An interesting future work is to analyze those. In particular, this would give a chance to assess the empirical evidence regarding the relation of BUFF security and the PS-3 transform.

We considered a weaker form of non-resignability (wNR) as the initial definition turned out to be unachievable—the problem being the auxiliary information. The majority of our results regarding wNR—attacks against 12 out of 16 signature schemes—remain relevant regardless of how the auxiliary information issue gets resolved eventually. The reason is that neither attack relies on any auxiliary information. On the other hand, our positive results only guarantee security against non-resignability in this restricted form. Once the matter of defining non-resignability is completely resolved, our positive results given here should be re-evaluated. Note, however, that for the 5 positive results, the schemes implicitly use either the PS-3 or the BUFF transform. Hence, if either the PS-3 or BUFF transform can be shown to generally satisfy a new definition of NR, the results would apply to the 5 positive results presented here.

Acknowledgements

We thank Jonathan Komada Eriksen, Rune Fiedler, and Krijn Reijnders for helpful comments and discussions. This work has been funded by the Deutsche Forschungsgemeinschaft (DFG – German Research Foundation) – 505500359 and SFB 1119 – 236615297, by the German Federal Ministry of Education and Research (BMBF) under the projects 6G-RIC (16KISK033) and Quant-ID (16KISQ111), and by the Hector Foundation II.

References

1. 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.
2. Marco Baldi, Alessandro Barenghi, Luke Beckwith, Jean-Francois Biasse, Andre Esser, Kris Gaj, Kamyar Mohajerani, Gerardo Pelosi, Edoardo Persichetti, Markku-Juhani Saarinen, Paolo Santini, and Robert Wallace. LESS. Technical report, National Institute of Standards and Technology, 2023.
3. Marco Baldi, Alessandro Barenghi, Sebastian Bitzer, Patrick Karl, Felice Manganiello, Alessio Pavoni, Gerardo Pelosi, Paolo Santini, Jonas Schupp, Freeman Slaughter, Antonia Wachter-Zeh, and Violetta Weger. CROSS. Technical report, National Institute of Standards and Technology, 2023.
4. Gustavo Banegas, Kévin Carrier, André Chailloux, Alain Couvreur, Thomas Debris-Alazard, Philippe Gaborit, Pierre Karpman, Johanna Loyer, Ruben Niederhagen, Nicolas Sendrier, Benjamin Smith, and Jean-Pierre Tillich. Wave. Technical report, National Institute of Standards and Technology, 2023.

5. 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.
6. Ward Beullens, Fabio Campos, Sophía Celi, Basil Hess, and Matthias J. Kannwischer. MAYO. Technical report, National Institute of Standards and Technology, 2023.
7. Ward Beullens, Ming-Shing Chen, Jintai Ding, Boru Gong, Matthias J. Kannwischer, Jacques Patarin, Bo-Yuan Peng, Dieter Schmidt, Cheng-Jhih Shih, Chengdong Tao, and Bo-Yin Yang. UOV. Technical report, National Institute of Standards and Technology, 2023.
8. 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.
9. Joppe W. Bos, Olivier Bronchain, Léo Ducas, Serge Fehr, Yu-Hsuan Huang, Thomas Pornin, Eamonn W. Postlethwaite, Thomas Prest, Ludo N. Pulles, and Wessel van Woerden. HAWK. Technical report, National Institute of Standards and Technology, 2023. available at `https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures`.
10. Giacomo Bruno, Maria Corte-Real Santos, Craig Costello, Jonathan Komada Eriksen, Michael Meyer, Michael Naehrig, and Bruno Sterner. Cryptographic smooth neighbors. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part VII*, volume 14444 of *LNCS*, pages 190–221. Springer, Heidelberg, December 2023.
11. Jorge Chavez-Saab, Maria Corte-Real Santos, Luca De Feo, Jonathan Komada Eriksen, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Michael Meyer, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Francisco Rodríguez Henríquez, Sina Schaeffler, and Benjamin Wesolowski. SQIsign. Technical report, National Institute of Standards and Technology, 2023.
12. Jung Hee Cheon, Hyeongmin Choe, Julien Devevey, Tim Güneysu, Dongyeon Hong, Markus Krausz, Georg Land, Junbum Shin, Damien Stehlé, and MinJune Yi. HAETAE. Technical report, National Institute of Standards and Technology, 2023.
13. Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Lars Ran, Tovohery Hajatiana Randrianarisoa, Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. MEDS. Technical report, National Institute of Standards and Technology, 2023.
14. Maria Corte-Real Santos, Jonathan Komada Eriksen, Michael Meyer, and Krijn Reijnders. AprèsSQI: Extra Fast Verification for SQIsign Using Extension-Field Signing. Cryptology ePrint Archive, Paper 2023/1559, 2023.
15. Craig Costello, Michael Meyer, and Michael Naehrig. Sieving for twin smooth integers with solutions to the prouhet-tarry-escott problem. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 272–301. Springer, Heidelberg, October 2021.
16. 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.
17. Quang Dao. Weak keys break the BUFF security of HAWK. Cryptology ePrint Archive, Paper 2026/1298, 2026.
18. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: Compact post-quantum signatures from quaternions and

isogenies. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 64–93. Springer, Heidelberg, December 2020.
19. Luca De Feo, Antonin Leroux, Patrick Longa, and Benjamin Wesolowski. New algorithms for the deuring correspondence - towards practical and secure SQISign signatures. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 659–690. Springer, Heidelberg, April 2023.
20. Rafael del Pino, Thomas Espitau, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, Mélissa Rossi, and Markku-Juhani Saarinen. Raccoon. Technical report, National Institute of Standards and Technology, 2023.
21. Jintai Ding, Boru Gong, Hao Guo, Xiaoou He, Yi Jin, Yuansheng Pan, Dieter Schmidt, Chengdong Tao, Danli Xie, and Ziyu Yang, Bo-Yin Zhao. TUOV. Technical report, National Institute of Standards and Technology, 2023.
22. Jelle Don, Serge Fehr, Yu-Hsuan Huang, and Patrick Struck. On the (in)security of the BUFF transform. *IACR Cryptol. ePrint Arch.*, 2023:1634, 2023.
23. Thomas Espitau, Guilhem Niot, Chao Sun, and Mehdi Tibouchi. SQUIRRELS. Technical report, National Institute of Standards and Technology, 2023.
24. Hiroki Furue, Yasuhiko Ikematsu, Fumitaka Hoshino, Tsuyoshi Takagi, Kan Yasuda, Toshiyuki Miyazawa, Tsunekazu Saito, and Akira Nagai. QR-UOV. Technical report, National Institute of Standards and Technology, 2023.
25. Shafi Goldwasser, Silvio Micali, and Ronald Rivest. A digital signature scheme secure against adaptive chosen-message attacks. In *SIAM Journal on Computing*, 1988.
26. Louis Goubin, Benoît Cogliati, Jean-Charles Faugére, Pierre-Alain Fouque, Robin Larrieu, Gilles Macario-Rat, Brice Minaud, and Jacques Patarin. PROV. Technical report, National Institute of Standards and Technology, 2023.
27. Andreas Hülsing, Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Jean-Philippe Aumasson, Bas Westerbaan, and Ward Beullens. SPHINCS$^+$. Technical report, National Institute of Standards and Technology, 2020.
28. 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.
29. 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*, 2015.
30. Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar signature schemes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 206–222. Springer, Heidelberg, May 1999.
31. Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2020.
32. Alfred Menezes and Nigel P. Smart. Security of signature schemes in a multi-user setting. *DCC*, 33(3):261–274, 2004.
33. 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.

34. Jacques Patarin. The oil and vinegar signature scheme, 1997.
35. Jacques Patarin, Benoît Cogliati, Jean-Charles Faugére, Pierre-Alain Fouque, Louis Goubin, Robin Larrieu, Gilles Macario-Rat, and Brice Minaud. VOX. Technical report, National Institute of Standards and Technology, 2023.
36. 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.
37. 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.
38. Zygimantas Straznickas and Stephen A. Weis. HAWK-n key recovery reduces to SVP in dimension $n/2 + 1$. Cryptology ePrint Archive, Paper 2026/1593, 2026.
39. Lih-Chung Wang, Chun-Yen Chou, Jintai Ding, Yen-Liang Kuan, Ming-Siou Li, Bo-Shu Tseng, Po-En Tseng, and Chia-Chun Wang. SNOVA. Technical report, National Institute of Standards and Technology, 2023.
40. Yang Yu, Huiwen Jia, Leibo Li, Delong Ran, Zhiyuan Qiu, Shiduo Zhang, Xiuhan Lin, and Xiaoyun Wang. HuFu. Technical report, National Institute of Standards and Technology, 2023.