A Closer Look at Falcon

Phillip Gajland[1,2,3], Jonas Janneck[2], and Eike Kiltz[2]

[1] Max Planck Institute for Security and Privacy
[2] Ruhr University Bochum
[3] IBM Research – Zurich

15th November 2024

Abstract Falcon is a winner of NIST's six-year post-quantum cryptography standardisation competition. Based on the celebrated full-domain-hash framework of Gentry, Peikert and Vaikuntanathan (GPV) (STOC'08), Falcon leverages NTRU lattices to achieve the most compact signatures among lattice-based schemes.

Its security hinges on a Rényi divergence-based argument for Gaussian samplers, a core element of the scheme. However, the GPV proof, which uses statistical distance to argue closeness of distributions, fails when applied naively to Falcon due to parameter choices resulting in statistical distances as large as 2^{-34}. Additional implementation-driven deviations from the GPV framework further invalidate the original proof, leaving Falcon without a security proof despite its selection for standardisation.

This work takes a closer look at Falcon and demonstrates that introducing a few minor, conservative modifications allows for the first formal proof of the scheme in the random oracle model. At the heart of our analysis lies an adaptation of the GPV framework to work with the Rényi divergence, along with an optimised method for parameter selection under this measure. Furthermore, we obtain a provable version of the GPV framework over NTRU rings. Both these tools may be of independent interest.

Unfortunately, our analysis shows that despite our modification of Falcon-512 and Falcon-1024 we do not achieve *strong unforgeability* for either scheme. For *plain unforgeability* we are able to show that our modifications to Falcon-512 barely satisfy the claimed 120-bit security target and for Falcon-1024 we confirm the claimed security level. As such we recommend revisiting Falcon and its parameters.

Contents

1 Introduction

Among the 69 submissions to the NIST post-quantum cryptography standardisation process in 2016 [Kim16], Falcon [PFH$^+$20] was selected as one of four winning algorithms in 2022. Currently, NIST is in the process of drafting the corresponding FIPS standard. Falcon is a signature scheme based on the full-domain-hash (FDH) paradigm [BR96], commonly known as *"hash-and-sign"*. In this framework, the public verification key is a trapdoor permutation f and the signing key is the inverse f^{-1}. To sign a message m, one first hashes m to some point $y = \mathsf{H}(m)$ in the range of f, then outputs the signature $\sigma = f^{-1}(y)$. Verification consists of checking that $f(\sigma) = \mathsf{H}(m)$. Falcon, like most of the selected algorithms such as Kyber [SAB$^+$22] and Dilithium [LDK$^+$22], relies on the hardness of lattice problems. Its design follows the FDH framework over lattices, as formalised in the celebrated work of Gentry, Peikert and Vaikuntanathan (GPV) [GPV08], which generalised the FDH paradigm to work with *preimage sampleable trapdoor functions*, rather than solely permutations. Concretely, GPV signatures σ are sampled from $f^{-1}(\mathsf{H}(m))$. By leveraging NTRU lattices, introduced by Hoffstein, Pipher, and Silverman [HPS98, HHP$^+$03], Falcon benefits from their ring structure, allowing a reduction in public keys by a factor of $\mathcal{O}(n)$ and accelerating many computations by a factor of $\mathcal{O}(n/\log n)$. More importantly, [DLP14] showed that, by choosing appropriate parameters, the length of NTRU trapdoors can be within a small constant factor of the theoretical optimal, achieving the most compact signatures among lattice-based schemes. Compared to other signature schemes selected for standardisation by NIST, such as Dilithium [LDK$^+$22] and Sphincs+ [HBD$^+$22], Falcon stands out for its compactness, minimising both public key and signature sizes.

While the GPV framework was originally proven [GPV08] under the plain (unstructured) Short Integer Solution (SIS) assumption [Ajt96], adapting it to the (structured) NTRU-SIS setting is described in the Falcon specification as *"straightforward"*. The GPV proof relies on the Leftover Hash Lemma [HILL99] to argue that the simulation of the random oracle is statistically close to uniform. While this statistical argument can be adapted using a regularity lemma for rings [SS11, LPR13, RSW18], a more critical issue arises: applying this argument with Falcon parameters leads to statistical distances as large as 2^{-34}. Moreover, Falcon deviates from the GPV framework by relying on the Rényi divergence instead of statistical distance. This choice, made to achieve tighter parameters and smaller signature sizes, introduces additional uncertainty about the scheme's security guarantees. For instance, [LAZ19, Sec. 2.3] states that the parameters used in Falcon are not supported by the GPV proof.

Given the importance of thoroughly understanding schemes intended for mass deployment, and in light of recent classical attacks on post-quantum schemes [Beu22, CD23, MMP$^+$23, Rob23], careful security analysis is paramount. Despite successfully progressing through all three stages of the NIST process and being selected for standardisation, a formal proof of Falcon remains elusive raising the following pertinent question.

Can Falcon be proven secure? If so, what is its concrete security?

1.1 Contributions

This work provides the first formal security analysis of Falcon-type signature schemes in the GPV framework. Our main contributions are:

Extending the GPV framework to Rényi divergence. We extend the GPV framework to incorporate the Rényi divergence, adapting key lemmata to support the Rényi divergence and NTRU rings. This result is broadly applicable to other constructions including [GJK24, EFG$^+$22, ENS$^+$23, YJW23]. We also develop tools for optimally selecting parameters for Rényi divergence. For example, while Falcon recommends using a Rényi divergence of order $a = 2\lambda$, this results in a 60-bit security loss for the Falcon-1024 parameter set. Our tools reduce this loss to just 16 bits.

Falcon$^+$: modifications to Falcon for provable security. While our extensions to the GPV framework and parameter optimisation tools improve the security analysis, we were not able to prove the security of Falcon without modifications. To this end, we introduce Falcon$^+$, a minor modification of Falcon, that can easily be justified at this late stage of the standardisation process. The differences to

Falcon are sketched in Fig. 4. Besides hashing the public key (which is standard cryptographic practice and allows for a tighter multi-user security proof), Falcon$^+$ crucially samples a random salt and samples a preimage of the hash of the message/salt pair *within* the repeat loop of signing, i.e., until a sufficiently short preimage is found. In contrast, Falcon picks a fixed random salt *outside* of the repeat loop and then samples the preimage.[4] This modification incurs minimal additional cost since the loop is executed only once or twice in expectation. Furthermore, the costs associated with Gaussian sampling within the loop far outweigh the hashing and FFT costs, even for large messages.

```
$\mathsf{Sgn}(sk, m)$
Sample salt $r$
repeat
    $\boldsymbol{s} \xleftarrow{\$} f^{-1}(\mathsf{H}(r, m))$
until $\|\boldsymbol{s}\|_2 \leq \beta$
$\sigma := (r, \boldsymbol{s})$

$\mathsf{Sgn}^+(sk, m)$
repeat
    Sample salt $r$
    $\boldsymbol{s} \xleftarrow{\$} f^{-1}(\mathsf{H}(pk, r, m))$
until $\|\boldsymbol{s}\|_2 \leq \beta$
$\sigma := (r, \boldsymbol{s})$
```

Figure 1. Signing (simplified) of original Falcon (left) and our modification Falcon$^+$ (right). Sampling from $f^{-1}(\cdot)$ is done using sk.

Security analysis and recommendations. We provide a thorough security analysis of Falcon$^+$ in the random oracle model. Using our new tools, we obtain concrete security bounds derived from our theorems focusing on minimising the bit security loss due to Rényi divergence arguments. Our findings show that using our Rényi divergence tools combined with existing techniques, neither Falcon$^+$-512 (NIST Level 1) nor Falcon$^+$-1024 (NIST Level 5) provide *strong unforgeability*. For comparison, schemes like Dilithium [LDK+22] already meet strong unforgeability. In the case of *plain unforgeability*, Falcon$^+$-512 achieves only 100 bits of provable security. By reducing the number of allowed signing queries from 2^{64} to 2^{57}, we increase this to 118 bits, nearing the claimed security level. For Falcon$^+$-1024, we prove that it meets 256 bits of security for *plain unforgeability*. An overview of the provable bit security is shown in Table 1.

Note that we do not present concrete attacks against Falcon, nor do we claim that a better security proof is impossible. Rather, we show that our tools combined with currently known proof techniques are insufficient to fully justify the target security claims of Falcon and Falcon$^+$. In fact, Falcon in its current state has no provable security.

Scheme	**UF-CMA**	**SUF-CMA**
Falcon$^+$-512 ($Q_s = 2^{64}$)	100	75
Falcon$^+$-512 ($Q_s = 2^{57}$)	118	93
Falcon$^+$-1024 ($Q_s = 2^{64}$)	256	0

Table 1. Provable bit security levels of Falcon$^+$-512 and Falcon$^+$-1024, where Q_s is the maximal number of allowed signing queries.

[4] Note that SQUIRRELS [ENST23], a scheme submitted to the first round of the *NIST Call for Additional Post-Quantum Signature Schemes*, suffers from the same shortcoming.

1.2 Technical Overview

The Gentry-Peikert-Vaikuntanathan Framework. The GPV framework [GPV08] provides a method for constructing secure lattice-based signature schemes via the full-domain-hash (FDH) paradigm [BR96], commonly known as *"hash-and-sign"*. Central to this framework is a *"preimage sampleable trapdoor function"* defined as $f_{\boldsymbol{A}}(s) := \boldsymbol{A}s \mod q$ where $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$. Here, each signature essentially corresponds to a short preimage of the hash of a message. More specifically, the public key pk is a full-rank matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ (with $n \leq m$) which defines a q-ary lattice $\boldsymbol{\Lambda}$. The secret key (or trapdoor) sk is a matrix $\boldsymbol{B} \in \mathbb{Z}_q^{m \times m}$ generating the lattice orthogonal to $\boldsymbol{\Lambda} \mod q$, enabling the efficient inversion of $f_{\boldsymbol{A}}$. A signature on a message m is a short Gaussian vector $s \in \mathbb{Z}_q^m$ such that $\mathsf{H}(m) = \boldsymbol{A}s \mod q$, where $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_q^n$ is a hash function. Verification involves checking both the shortness of s and that $f_{\boldsymbol{A}}(s) = \mathsf{H}(m)$.

The GPV Proof. The GPV framework was proven secure in both the random oracle model [BR93,GPV08] and the quantum random oracle model [BDF+11] under the plain (unstructured) Short Integer Solution (SIS) assumption [Ajt96]. Security can be establish in two ways: (1) via *collision resistance*, reducing to SIS, or (2) via *one-wayness*, reducing to ISIS. The original work [GPV08] presented a *tight* proof of *strong unforgeability* for FDH, leveraging collision resistance.

Suppose, for the sake of contradiction, that an adversary A breaks the *strong unforgeability* of the signature scheme producing a forgery $(m^\star, s^*)$ where s^* is short and $\mathsf{H}(m^*) = \boldsymbol{A}s^* \mod q$. We construct an adversary B that breaks SIS by finding a collision in $f_{\boldsymbol{A}}(s)$. Given a SIS instance $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$, B runs adversary A on the public key $pk = \boldsymbol{A}$ and simulates the random oracle H and signing oracle as follows.

- The random oracle is programmed as follows: for each fresh query to $\mathsf{H}(m)$, B samples a Gaussian s_m and returns $\mathsf{H}(m) := \boldsymbol{A}s_m \mod q$ to A. Crucially, by the leftover hash lemma [HILL99] the simulated random oracle output is statistically close to uniform.
- Whenever A makes a signing query on m, B retrieves (m, s_m) from the hash table and returns s_m as the signature. It can be shown again that the distribution of the signature is statistically close to the real one.
- Finally, upon receiving a forgery from A, B can use this to find a collision. That is, two different preimages that map to the same hash value giving a solution to SIS. When A produces the forgery (m^*, s^*), B looks up (m^*, s_{m^*}) in its hash table and outputs $(s^* - s_{m^*})$ as a SIS solution to $\boldsymbol{A}$. This is a valid solution to SIS because $\mathsf{H}(m^*) = \boldsymbol{A}s^* \mod q$ and $\mathsf{H}(m^*) = \boldsymbol{A}s_{m^*} \mod q$, and $\boldsymbol{A}(s^* - s_{m^*}) = 0 \mod q$ and $\|s^* - s_{m^*}\|$ is small.

Plain unforgeability can also be proven with a reduction to one-wayness (inhomogenous SIS). This proof is looser but enjoys better SIS parameters.

Falcon Instantiation of the GPV Framework. The design of Falcon prioritises compactness, minimising the combined size of $|pk| + |\sigma|$. To achieve this, Falcon relies on the class of NTRU lattices introduced by Hoffstein, Pipher, and Silverman [HPS98, HHP+03], which come with an additional ring structure that reduces the public key size by a factor of $\mathcal{O}(n)$ and accelerates many computations by a factor of at least $\mathcal{O}(n/\log n)$. Among structured lattices, NTRU lattices are particularly efficient, with public keys represented as a single polynomial $\boldsymbol{h} \in \mathcal{R} = \mathbb{Z}_q[X]/(X^n + 1)$. Falcon instantiates a randomized version of the GPV framework with the NTRU-based preimage sampleable trapdoor $f_{\boldsymbol{h}}$ [HPS98, DLP14, PFH+22]. Specifically, $f_{\boldsymbol{h}}$ maps two ring elements $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ to $\boldsymbol{s}_1 + \boldsymbol{h} * \boldsymbol{s}_2 \mod q$. Observe that $f_{\boldsymbol{h}}$ is a special case of the GPV trapdoor function $f_{\boldsymbol{A}}(s) = \boldsymbol{A}s \bmod q$. A valid signature on message m consists of a tuple $(\boldsymbol{s}_1, \boldsymbol{s}_2) \in \mathcal{R}^2$ and a random salt $r \in \{0,1\}^k$ satisfying

$$\mathsf{H}(m, r) = \boldsymbol{s}_1 + \boldsymbol{h} * \boldsymbol{s}_2 \mod q \quad \wedge \quad \|(\boldsymbol{s}_1, \boldsymbol{s}_2)\|_2 \leq \beta.$$

This adaptation requires the standard *"randomised GPV"* proof to be based on an *"NTRU-SIS"* assumption, a process described as *"straightforward"* in the Falcon specification [PFH+22].

Repeated Sampling and Salting. One key difference in Falcon compared to the GPV framework is that signatures are not directly output from the preimage sampling procedure, as they may fail verification if their norms are too large – something that occurs with small probability of about 2^{-14}. To eliminate this correctness error, signatures are checked for shortness, and if the norm exceeds some threshold, a new preimage is sampled repeatedly until one with a sufficiently small norm is found. This introduces a complication for simulating signing queries, as the process involves conditional distributions. The signing oracle outputs preimages conditioned on having sufficiently small norm, whereas programming the random oracle with this constraint and arguing about the uniformity of outputs seems to be challenging.

In the current Falcon specification, the random salt r is chosen before the preimage sampling loop and therefore does not help mitigate the issue of conditional distributions. In our modified scheme, Falcon$^+$, we propose drawing a new salt each time the preimage sampling process results in too large signatures. This modification allows the reduction to continue programming the random oracle with large preimages, while still being able to produce valid signatures. If a sampled preimage is too large, the reduction can simply choose a new salt, yielding a new random oracle output and a new preimage. This change incurs only a minor constant overhead in the security bound, corresponding to the maximum number of repetitions. In practice, the efficiency impact is minimal, as preimage sampling remains the dominant computational cost in both the original and modified schemes.

Rényi Divergence in Falcon. Another issue is that Falcon relies on the Rényi divergence, whereas the GPV framework uses the statistical distance to prove the closeness of the sampler and a Gaussian. Citing [Pre17, Lem. 6] and the analysis of the Klein sampler [Kle00], Falcon claims that for suitable parameters, the Rényi divergence between the FFO sampler's output and an ideal Gaussian is bounded by $1 + \mathcal{O}(1)/Q_s$, incurring a loss of at most $\mathcal{O}(1)$ bits of security. However, we are interested in the concrete bounds. To address this, we modify the GPV framework to the handle Rényi divergence, enabling the simulation of signing queries.

Furthermore, the GPV framework uses a second statistical argument, the Leftover Hash Lemma, to show that the programmed output of the random oracle is close to uniform. However, two challenges arise. First, the argument, originally stated for unstructured lattices, must be adapted to the ring setting, which can be done using a regularity lemma from [SS11, Sec. 3.3] or [LPR13, Sec. 4]. More critically, applying such a statistical argument to the Falcon parameters yields statistical distances as large as 2^{-34}, for each simulated random oracle output. As a result, further modifications to the GPV framework are necessary to argue that the random oracle's output is Rényi-close to uniform. That is, we require a lemma showing that $\mathsf{H}(m, r) := \boldsymbol{s}_1 + \boldsymbol{h} * \boldsymbol{s}_2 \mod q$ is Rényi close to uniform for Gaussian $\boldsymbol{s}_1, \boldsymbol{s}_2$. However, the Rényi divergence arguments are highly sensitive to the number of signing queries $Q_s = 2^{64}$ and random oracle queries $Q_\mathsf{H} = 2^{96}$. This is primarily because the Falcon parameters are specifically tuned to accommodate signing queries rather than random oracle queries. Thus, these tools cannot be applied directly in the random oracle model, requiring us to carefully program only those random oracle queries originating from signing queries. The downside is that for a reduction to collision resistance we need to guess a specific random oracle query, leading to a loss of Q_H.

Concrete Security of Falcon$^+$. Table 2 summarises our concrete security bounds of Falcon$^+$, our modified version of Falcon. A key observation is that achieving *strong unforgeability* requires a norm bound of 2β when reducing to SIS, while *plain unforgeability* requires a norm bound of β in the reduction to ISIS. Since SIS and ISIS become easier with larger β, this explains why neither Falcon$^+$-512 nor Falcon$^+$-1024 satisfy *strong unforgeability* as indicated in Table 1. The parameters of Falcon have been carefully chosen such that the Rényi bound $r_u^{Q_s} = (1 + \delta_u)^{Q_s}$ is a small constant for $Q_s = 2^{64}$ signing queries. However, the Rényi bound $r_u^{Q_\mathsf{H}} = (1+\delta_u)^{Q_\mathsf{H}}$ can be extremely large for $Q_\mathsf{H} = 2^{96} \gg 2^{64}$ random oracle queries. Theorem 1 has a loose Rényi bound (i.e., term $r_u^{Q_\mathsf{H}}$). The reason is that in the proof, the random oracle needs to be programmed for every query (i.e., Q_H times) to $\mathsf{H}(m, r) = \boldsymbol{s}_1 + \boldsymbol{h} * \boldsymbol{s}_2$ for known $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ such that the reduction can construct a collision from one of the programmed preimages and the forgery. Due to the loss of $r_u^{Q_\mathsf{H}}$, this theorem is not applicable to the proposed Falcon parameters. In contrast, Theorem 2 has a tighter Rényi bound (i.e., term $r_u^{Q_s}$) but comes with an additional multiplicative security loss of Q_H. The idea of the proof is to guess the challenge random oracle query, i.e., the query corresponding to the output forgery, which incurs a multiplicative loss of Q_H. In turn, the random oracle only needs to be programmed to

$\mathsf{H}(m,r) = \boldsymbol{s}_1 + \boldsymbol{h} * \boldsymbol{s}_2$ on that one query and in addition on at most Q_s positions to answer signing queries. This leads to a smaller Rényi loss of $r_u^{Q_s+1} \approx r_u^{Q_s}$ which is applicable to the proposed Falcon parameters.

The resulting bit security for Falcon$^+$-512 (NIST Level I) and Falcon$^+$-1024 (NIST Level V) are shown in Table 1 on page 4. These values are derived from Theorem 2, taking into account the Rényi loss for the Falcon parameter sets, and using the *"lattice-estimator"* [APS15b, APS15a] to estimate the hardness of (I)SIS. However, we choose to ignore the $Q_\mathsf{H} \approx 2^{96}$ loss stemming from the random oracle guessing of the proof, as this is necessary to obtain any level of meaningful security. Furthermore, this loss is often considered a proof artifact, which is why it is *sometimes* set aside from a practical standpoint. An example is the loss arising from rewinding, which is also viewed as a proof artifact and ignored in the analyses of Dilithium [LDK$^+$22] and EdDSA [BDL$^+$12]. In contrast, losses from information-theoretic terms, such as Rényi divergence and statistical distance, are not proof artifacts; they are directly linked to the parameter choices of the scheme. For example, to reduce the loss factor $r_u^{Q_\mathsf{H}}$, one would need to increase the scheme's standard deviation s, which in turn results in larger signatures. Ignoring these Rényi losses while setting arbitrarily small parameters can lead to an insecure scheme.

Security	**Multiplicative loss**	**Assumption**
SUF-CMA (Th. 1)	$r_u^{Q_\mathsf{H}} \cdot r_p^{Q_s}$	$\mathcal{R}$-$\mathbf{SIS}_{2\beta}$
SUF-CMA (Th. 2)	$r_u^{Q_s} \cdot r_p^{Q_s} \cdot Q_\mathsf{H}$	$\mathcal{R}$-$\mathbf{SIS}_{2\beta}$
UF-CMA (Th. 2)	$r_u^{Q_s} \cdot r_p^{Q_s} \cdot Q_\mathsf{H}$	$\mathcal{R}$-$\mathbf{ISIS}_{\beta}$

Table 2. Concrete security loss (simplified) for Falcon$^+$ in the random oracle model. Constants $r_u = 1 + \delta_u$ and $r_p = 1 + \delta_p$ are Rényi divergences related to the uniformity of a NTRU evaluation on Gaussian inputs (r_u) and the preimage sampler (r_p). Q_s and Q_H denote the number of signing and random oracle queries, respectively.

Multi-User Security. For signatures, and most other primitives, the single-user setting is typically considered as it is simpler to analyse and due to the existence of a standard reduction [GMLS02] from multi-user to single-user security. Specifically, for any signature scheme, the attacker's success probability in the multi-user setting with N public keys is at most N times higher than in the single-user setting. However, this reduction is unsatisfactory, as with $N = 2^{40}$ public keys, the attacker's success probability increases by a factor of 2^{40}. For instance, if the single-user security is 2^{-128}, the multi-user security would only be 2^{-88}, which is insufficient considering current threats. Similar to the Q_H loss from random oracle guessing discussed before, in practice such multi-user reduction losses are often ignored. But from a provable security stand point taking this into account would result in significantly larger parameters in the signature scheme for the same level of security. Interestingly we are able to exploit the fact that the hardness of (I)SIS is independent of the number of samples, enabling us to give a *tight* proof of multi-user security. However, for this we require the public key to be included in the hash to program the random oracle output with respect to a specific user. To the best of our knowledge, this technique does not appear applicable to other PQC schemes such as Dilithium [LDK$^+$22]. Further details and the formal proof can be found in Appendix D.

1.3 Open Problems

We do not present specific attacks against Falcon, nor do we claim that a stronger security proof is impossible. It may be possible to improve the analysis, and we consider this an open problem worth exploring. It would be particularly interesting to remove the loss factor Q_H in Theorem 2, while maintaining the tighter Rényi bounds (cf. Table 2). Falcon relies on [Pre17, Lem. 6] and the analysis of the Klein sampler [Kle00], to claim that the Rényi divergence between the FFO sampler's output and an ideal Gaussian is small. An important area for future work is to conduct a similar analysis for the FFO sampler as well as the key generation procedure. Finally, we leave open as an interesting future direction the potential application of the the Rényi divergence in the quantum random oracle model (QROM). We remark that HAWK [BBD+23] was analysed in the QROM [FH23] but its proof only requires programming Q_s fixed positions in the QROM and no direct queries, unlike ours. Furthermore, they do not need to program the random oracle beyond responding to signing queries because they reduce to a different, "one-more"-type hardness assumption.

2 Preliminaries

We introduce some relevant notation and definitions used throughout the paper.

2.1 Notations

SETS AND ALGORITHMS. We write $s \xleftarrow{\$} \mathcal{S}$ to denote the uniform sampling of s from the finite set $\mathcal{S}$ and by $\mathcal{U}(\mathcal{S})$ the uniform distribution ober $\mathcal{S}$. For an integer n, we define $[n] \coloneqq \{1, \ldots, n\}$. The notation $[\![b]\!]$, where b is a boolean statement, evaluates to 1 if the statement is true and 0 otherwise. We use uppercase letters $\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}$ to denote algorithms. Unless otherwise stated, algorithms are probabilistic, and we write $(y_1, \ldots) \xleftarrow{\$} \mathsf{A}(x_1, \ldots)$ to denote that A returns $(y_1, \ldots)$ when run on input $(x_1, \ldots)$. We write A^{B} to denote that A has oracle access to B during its execution. The support of a discrete random variable X is defined as $\sup(X) \coloneqq \{x \in \mathbb{R} \mid \Pr[X = x] > 0\}$. For two polynomials $\boldsymbol{f}, \boldsymbol{g}$, we denote the polynomial multiplication of $\boldsymbol{f}$ and $\boldsymbol{g}$ by $\boldsymbol{f} * \boldsymbol{g}$. By "log" we denote the logarithm to base 2, by "ln" to base e.

SECURITY GAMES. We use standard code-based security games [BR04]. A *game* G is a probability experiment in which an adversary A interacts with an implicit challenger that answers oracle queries issued by A. The game G has one *main procedure* and an arbitrary amount of additional *oracle procedures* which describe how these oracle queries are answered. We denote the (binary) output b of game G between a challenger and an adversary A as $\mathsf{G}(\mathsf{A}) \Rightarrow b$. A is said to *win* G if $\mathsf{G}^{\mathsf{A}} \Rightarrow 1$, or shortly $\mathsf{G} \Rightarrow 1$. Unless otherwise stated, the randomness in the probability term $\Pr[\mathsf{G}(\mathsf{A}) \Rightarrow 1]$ is over all the random coins in game G. To provide a cleaner description and avoid repetitions, we sometimes refer to procedures of different games. To call the oracle procedure $\texttt{Oracle}$ of game G on input x, we shortly write $\mathsf{G}.\texttt{Oracle}(x)$. If a game is aborted the output is 0. For our analysis we rely on the commonly used main difference lemma or the multiplicative difference lemma for independent events.

2.2 Signatures

We recall the syntax and standard security notions of signatures.

Definition 1 (Signature Scheme). A *signature scheme* Sig is defined as a tuple $(\mathsf{Gen}, \mathsf{Sgn}, \mathsf{Ver})$ of the following three algorithms.

$(sk, pk) \xleftarrow{\$} \mathsf{Gen}$**:** The probabilistic key generation algorithm returns a secret key sk and a corresponding public key pk, where pk defines a message space $\mathcal{M}$.

$\sigma \xleftarrow{\$} \mathsf{Sgn}(sk, m)$**:** Given a secret key sk and a message $m \in \mathcal{M}$, the probabilistic signing algorithm Sgn returns a signature σ.

$b \leftarrow \mathsf{Ver}(pk, m, \sigma)$**:** Given a public key pk, a message m, and a signature σ, the deterministic verification algorithm Ver returns a bit b, such that $b = 1$ if and only if σ is a valid signature on m and $b = 0$ otherwise.

Sig has ε*-correctness error* if for all $(sk, pk) \in \sup(\mathsf{Gen})$ and any $m \in \mathcal{M}$ $\Pr[\mathsf{Ver}(pk, m, \mathsf{Sgn}(sk, m)) \neq 1] \leq \varepsilon$, where the probability is taken over the random choices of Gen and Sgn.

Definition 2 ((Strong) Unforgeability). The notions of *(strong) existential unforgeability under chosen message attacks* are formalised via the games Q_s-$\mathbf{UF\text{-}CMA}_{\mathsf{Sig}}(\mathsf{A})$ and Q_s-$\mathbf{SUF\text{-}CMA}_{\mathsf{Sig}}(\mathsf{A})$. Both are depicted in Figure 2, where Q_s is the maximum number of the adversary's signing queries. We define the advantage functions of adversary A as

$$\begin{aligned}\mathrm{Adv}_{\mathsf{Sig},\mathsf{A}}^{Q_s\text{-}\mathbf{UF\text{-}CMA}} &:= \Pr[Q_s\text{-}\mathbf{UF\text{-}CMA}_{\mathsf{Sig}}(\mathsf{A}) \Rightarrow 1],\\ \mathrm{Adv}_{\mathsf{Sig},\mathsf{A}}^{Q_s\text{-}\mathbf{SUF\text{-}CMA}} &:= \Pr[Q_s\text{-}\mathbf{SUF\text{-}CMA}_{\mathsf{Sig}}(\mathsf{A}) \Rightarrow 1].\end{aligned}$$

```
Games $Q_s$-UF-CMA$_{\mathsf{Sig}}(\mathsf{A})$/$Q_s$-SUF-CMA$_{\mathsf{Sig}}(\mathsf{A})$
$\mathcal{Q} \leftarrow \emptyset$
$(sk, pk) \xleftarrow{\$} \mathsf{Gen}$
$(m^\star, \sigma^\star) \xleftarrow{\$} \mathsf{A}^{\mathsf{Sgn}(\cdot)}(pk)$
return $[\![\mathsf{Ver}(m^\star, \sigma^\star) = 1 \wedge (m^\star, \cdot) \notin \mathcal{Q}]\!]$              // UF-CMA
return $[\![\mathsf{Ver}(pk, m^\star, \sigma^\star) = 1 \wedge (m^\star, \sigma^\star) \notin \mathcal{Q}]\!]$        // SUF-CMA
Oracle $\mathtt{Sgn}(m)$
$\sigma \xleftarrow{\$} \mathsf{Sgn}(sk, m)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(m, \sigma)\}$
return $\sigma$
```

Figure 2. Games defining **UF-CMA** and **SUF-CMA** for a signature scheme $\mathsf{Sig} = (\mathsf{Gen}, \mathsf{Sgn}, \mathsf{Ver})$ and adversary A making at most Q_s queries to $\mathtt{Sgn}$.

2.3 Lattices

Rings and Norms. In this work, we work with polynomial rings of the form $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$ for $n = 2^k$ and $k, q \in \mathbb{N}$. For a polynomial $\boldsymbol{f} \in \mathcal{R}_q$, let $f \in \mathbb{Z}_q^n$ denote the coefficient embedding of $\boldsymbol{f}$, and $f_i \in \mathbb{Z}_q$ the i^{th} coefficient.

Definition 3 (Anticirculant Matrix). For a polynomial $\boldsymbol{f} \in \mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$, the anticirculant matrix of $\boldsymbol{f}$ is defined as

$$\mathcal{A}(\boldsymbol{f}) = \begin{bmatrix} f_0 & f_1 & \cdots & f_{n-1} \\ -f_{n-1} & f_0 & \cdots & f_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ -f_1 & -f_2 & \cdots & f_0 \end{bmatrix} \in \mathbb{Z}^{n \times n}.$$

Anticirculant matrices satisfy the following useful properties.

Lemma 1. *Let $\boldsymbol{f}, \boldsymbol{g} \in \mathcal{R}$. Then $\mathcal{A}(\boldsymbol{f}) + \mathcal{A}(\boldsymbol{g}) = \mathcal{A}(\boldsymbol{f} + \boldsymbol{g})$ and $\mathcal{A}(\boldsymbol{f}) \cdot \mathcal{A}(\boldsymbol{g}) = \mathcal{A}(\boldsymbol{f} * \boldsymbol{g})$.*

This implies an isomorphism between $\mathcal{R}$ and the anticirculant matrices over $\mathbb{Z}^{n \times n}$, $\mathcal{R}_q$ and $\mathbb{Z}_q^{n \times n}$ respectively. Sometimes we overload the notation and write $\mathcal{A}(f)$ for the coefficient embedding $f \in \mathbb{Z}^n$ of $\boldsymbol{f}$ instead of $\mathcal{A}(\boldsymbol{f})$.

Let the ℓ_2-norm for $\boldsymbol{f} = f_0 + f_1 X + \ldots + f_{n-1} X^{n-1} \in \mathcal{R}_q$ be defined as $\|\boldsymbol{f}\|_2 := \sqrt{\sum_{i=0}^{n-1} |f_i|^2}$. For two polynomials $\boldsymbol{f}, \boldsymbol{g} \in \mathcal{R}_q$ we use the notation

$$\|(\boldsymbol{f}, \boldsymbol{g})\|_2 := \sqrt{\sum_{i=0}^{n-1} \left(|f_i|^2 + |g_i|^2\right)}.$$

Lattices. A lattice $\boldsymbol{\Lambda} \subseteq \mathbb{R}^n$ is a discrete additive subgroup of $\mathbb{R}^n$.

Definition 4 (Lattice). A rank m lattice in $\mathbb{R}^n$ is defined via the set $b_1, \ldots, b_m \in \mathbb{R}^n$ of *linearly independent* vectors that form a basis $\boldsymbol{B} = \{b_1, \ldots, b_m\}$ for the lattice

$$\boldsymbol{\Lambda} \coloneqq \boldsymbol{\Lambda}(\boldsymbol{B}) = \boldsymbol{\Lambda}(b_1, \ldots, b_m) = \left\{ \sum_{i=1}^{m} c_i b_i \mid c_1, \ldots, c_m \in \mathbb{Z} \right\}.$$

If $m = n$, then $\boldsymbol{\Lambda}$ is a full-rank lattice.

The *determinant* of a lattice $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}(\boldsymbol{B}) \subseteq \mathbb{R}^n$ for some basis $\boldsymbol{B} \in \mathbb{R}^{n \times m}$ is defined as $\det(\boldsymbol{\Lambda}) = \sqrt{\det(\boldsymbol{B}^\top \boldsymbol{B})}$. The *orthogonal* lattice for $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ is defined as $\boldsymbol{\Lambda}^\perp(\boldsymbol{A}) \coloneqq \{e \in \mathbb{Z}^m \mid \boldsymbol{A}e = 0 \mod q\}$. For an n-dimensional lattice $\boldsymbol{\Lambda}$, a lattice $\boldsymbol{\Lambda}' \subseteq \boldsymbol{\Lambda}$ is called a sublattice of $\boldsymbol{\Lambda}$. One can define the following quotient group $\boldsymbol{\Lambda}/\boldsymbol{\Lambda}' \coloneqq \{v + \boldsymbol{\Lambda}' \mid v \in \boldsymbol{\Lambda}\}$, which forms a group under the addition of cosets $v + \boldsymbol{\Lambda}'$.

Definition 5 (NTRU Lattice). Let $n = 2^k$ for $k \in \mathbb{Z}$, q prime, $\boldsymbol{f}, \boldsymbol{g} \in \mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$, and $\boldsymbol{h} = \boldsymbol{g} * \boldsymbol{f}^{-1} \mod q$. The NTRU lattice parameterised by $\boldsymbol{h}$ and q is a lattice of volume q^n in $\mathbb{R}^{2n}$ in the coefficient embedding of the following module

$$\boldsymbol{\Lambda}_{\boldsymbol{h},q} \coloneqq \{(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{R}^2 \mid \boldsymbol{u} + \boldsymbol{v} * \boldsymbol{h} = \boldsymbol{0} \mod q\}.$$

Equivalently, for $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$, an NTRU lattice is a full-rank submodule lattice of $\mathcal{R}^2$ generated by the columns of a matrix of the form

$$\boldsymbol{B}_{\boldsymbol{h}} = \begin{bmatrix} -\boldsymbol{h} & \boldsymbol{q} \\ \boldsymbol{1} & \boldsymbol{0} \end{bmatrix}$$

for prime q, $\boldsymbol{q} = q \cdot \boldsymbol{1}$, and some $\boldsymbol{h} \in \mathcal{R}$ coprime to q. A trapdoor for this lattice is a relatively short basis

$$\boldsymbol{B}_{\boldsymbol{f},\boldsymbol{g}} = \begin{bmatrix} \boldsymbol{g} & \boldsymbol{G} \\ -\boldsymbol{f} & -\boldsymbol{F} \end{bmatrix}$$

where the basis vectors $(\boldsymbol{f}, \boldsymbol{g}) \in \mathcal{R}^2$ and $(\boldsymbol{F}, \boldsymbol{G}) \in \mathcal{R}^2$ are not much larger than $\sqrt{\det \boldsymbol{B}_{\boldsymbol{h}}} = \sqrt{q}$ and $\boldsymbol{f} * \boldsymbol{G} - \boldsymbol{g} * \boldsymbol{F} = q \mod (X^n + 1)$.

Gaussians and Preimage Sampling. We define discrete Gaussians and state some of their useful properties.

Definition 6 (Discrete Gaussian Distribution over $\boldsymbol{\Lambda}$). The n-dimensional *Gaussian function* $\rho_{s,c} \colon \mathbb{R}^n \to (0, 1]$ on $\mathbb{R}^n$ centred at $c \in \mathbb{R}^n$ with standard deviation $s > 0$ is defined by

$$\rho_{s,c}(x) \coloneqq \exp\left(-\frac{\|x - c\|_2^2}{2s^2}\right).$$

For any $c \in \mathbb{R}^n$, $s \in \mathbb{R}^+$, and lattice $\boldsymbol{\Lambda}$, the *discrete Gaussian distribution over* $\boldsymbol{\Lambda}$ is defined as

$$\forall\, x \in \boldsymbol{\Lambda}, \quad \mathcal{D}_{\boldsymbol{\Lambda},s,c} \coloneqq \frac{\rho_{s,c}(x)}{\sum_{z \in \boldsymbol{\Lambda}} \rho_{s,c}(z)}.$$

We sometimes use the following notation $\rho_{s,c}(\boldsymbol{\Lambda}) = \sum_{x \in \boldsymbol{\Lambda}} \rho_{s,c}(x)$. We omit the subscript c when the Gaussian is centred at 0 and subscript $\boldsymbol{\Lambda}$ when the Gaussian is over $\mathbb{Z}^n$. We use $\boldsymbol{f} \sim \mathcal{D}_{\mathcal{R}}$ to denote the polynomial $\boldsymbol{f} \coloneqq \sum_{i=0}^{n-1} f_i X^i \mod (X^n + 1)$ for $f \sim \mathcal{D}_{\mathbb{Z}^n}$.

For bounding the probability that a random variable deviates a long way from the mean, we will use the following tail bound from [Ban93, Lyu12].

Lemma 2 (Gaussian Tailbound [Lyu12, Lem. 4.4 (3)]). Let $n, k > 1$ and $s > 0$. then it holds

$$\Pr_{z \leftarrow \mathcal{D}_{\mathbb{Z}^n,s}} [\|z\|_2 > ks\sqrt{n}] < k^n e^{\frac{n}{2}(1-k^2)}.$$

Lemma 3 (Min-Entropy of Gaussian [Lyu12, Lem. 4.4 (2)]). Let $n \in \mathbb{N}$, $\mathbf{\Lambda} \in \mathbb{Z}^n$ and $s \geq 3/\sqrt{2\pi}$. Then for any $c \in \mathbb{R}^n$,

$$\forall\, x \in \mathbf{\Lambda}, \quad \mathcal{D}_{\mathbf{\Lambda},s,c}(x) \leq 2^{-n}.$$

Definition 7 (Gram-Schmidt Norm [GPV08, DLP14]). For a finite basis $\boldsymbol{B} = (\boldsymbol{b}_i)_{i\in I}$, let $\tilde{\boldsymbol{B}} = (\tilde{\boldsymbol{b}}_i)_{i\in I}$ be its Gram-Schmidt orthogonalization. Then the Gram-Schmidt norm of $\boldsymbol{B}$ is the value $\|\boldsymbol{B}\|_{GS} \coloneqq \max_{i\in I} \left\|\tilde{\boldsymbol{b}}_i\right\|$.

Lemma 4 (NTRU Trapdoor Generation [HPS98, Pre15]). An NTRU Trapdoor Generation algorithm $\mathsf{TpdGen}(\mathcal{R}, \alpha, q)$, given a ring $\mathcal{R}$, a target quality $\alpha \geq 1$, and a modulus q, returns a public key $\boldsymbol{h} \in \mathcal{R}_q \setminus \{\boldsymbol{0}\}$ and the trapdoor $(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{F}, \boldsymbol{G}) \in \mathcal{R}^4$, such that $\boldsymbol{B_h}$ and $\boldsymbol{B_{f,g}}$ form a basis of the same lattice. Furthermore, $\|\boldsymbol{B_{f,g}}\|_{GS} \leq \alpha\sqrt{q}$.

Let $\mathbf{\Lambda}$ be an n-dimensional lattice and $\epsilon > 0$, the (scaled) smoothing parameter $\eta_\epsilon(\mathbf{\Lambda})$ is the smallest $s > 0$ such that $\rho_{1/s}(\mathbf{\Lambda}^* \setminus 0) \leq \epsilon$, where $\mathbf{\Lambda}^*$ denotes the dual lattice (the exact definition of the dual is not required for this work). We will use the following upper bound on the smoothing parameter.

Lemma 5 (Special Case of [MR07, Lem. 4.4]). For any $\epsilon \in (0, 1)$ it holds

$$\eta_\epsilon\left(\mathbb{Z}^{2n}\right) \leq \frac{1}{\pi} \cdot \sqrt{\frac{\ln(4n(1 + 1/\epsilon))}{2}}.$$

The following lemma appears implicitly in [MR04, Lem. 4.4].

Lemma 6. *For any n-dimensional lattice $\mathbf{\Lambda}$, vector $c \in \mathbb{R}^n$, and reals $0 < \epsilon < 1$, $s \geq \eta_\epsilon(\mathbf{\Lambda})$, we have*

$$\rho_{s,c}(\mathbf{\Lambda}) \in [1 - \epsilon, 1 + \epsilon] \cdot \frac{s^n}{\det(\mathbf{\Lambda})}.$$

2.4 Rényi Divergence

Definition 8 (Rényi Divergence [Rén61, BLL$^+$15, Pre17]). Let $\mathcal{P}, \mathcal{Q}$ be two distributions such that $\text{sup}(\mathcal{P}) \subseteq \text{sup}(\mathcal{Q})$. For $a \in (1, \infty)$, we define the Rényi divergence of order a as

$$R_a(\mathcal{P}||\mathcal{Q}) = \left(\sum_{x\in\text{sup}(\mathcal{P})} \frac{\mathcal{P}(x)^a}{\mathcal{Q}(x)^{a-1}}\right)^{\frac{1}{a-1}}.$$

In addition, we define the Rényi divergence of order $+\infty$ as

$$R_\infty(\mathcal{P}||\mathcal{Q}) = \max_{x\in\text{sup}(\mathcal{P})} \frac{\mathcal{P}(x)}{\mathcal{Q}(x)}.$$

Note that it is not symmetric and does not verify a triangle inequality. When the Rényi divergence is finite, which it will be for all our applications, we can think of it as a value $1 + \delta$ for $\delta \geq 0$. A smaller δ indicates that the distributions are closer.

Lemma 7 (Properties of the Rényi Divergence [BLL$^+$15, Lem. 2.7]). *Let $a \in (1, \infty)$, $\mathcal{P}$ and $\mathcal{Q}$ two distributions with $\text{sup}(\mathcal{P}) \subseteq \text{sup}(\mathcal{Q})$, and $(\mathcal{P}_i)_i, (\mathcal{Q}_i)_i$ two families of distributions. Then the following properties hold.*

- ***Data Processing Inequality:*** *For a function f, let $\mathcal{P}^f$ (resp. $\mathcal{Q}^f$) denote the distribution of $f(x)$ where $x \leftarrow \mathcal{P}$ (resp. $x \leftarrow \mathcal{Q}$). Then for any function f, $R_a(\mathcal{P}^f \mid\mid \mathcal{Q}^f) \leq R_a(\mathcal{P} \mid\mid \mathcal{Q})$.*
- ***Multiplicativity:*** *$R_a(\prod_i \mathcal{P}_i \mid\mid \prod_i \mathcal{Q}_i) = \prod_i R_a(\mathcal{P}_i \mid\mid \mathcal{Q}_i)$.*
- ***Probability Preservation:*** *For any event $E \subseteq \text{sup}(\mathcal{Q})$,*

$$\mathcal{Q}(E) \geq \mathcal{P}(E)^{\frac{a}{a-1}} / R_a(\mathcal{P} \,||\, \mathcal{Q}),$$
$$\mathcal{Q}(E) \geq \mathcal{P}(E) / R_\infty(\mathcal{P} \,||\, \mathcal{Q}).$$

Definition 9 (Relative Error [MW17]). *Let $\mathcal{P}$ and $\mathcal{Q}$ be two discrete probability distributions over the same countable set $\mathcal{X}$. The relative error of $\mathcal{P}$ and $\mathcal{Q}$ is defined as*

$$\delta_{RE}(\mathcal{P}, \mathcal{Q}) \coloneqq \max_{x \in \sup(\mathcal{P})} \frac{|\mathcal{P}(x) - \mathcal{Q}(x)|}{\mathcal{P}(x)}.$$

The relative error can be used to bound the Renyi Divergence.

Lemma 8 (Relative Error [Pre17, Lem. 3]). Let $\mathcal{P}, \mathcal{Q}$ be two distributions such that $\sup(\mathcal{P}) = \sup(\mathcal{Q})$ and $\delta_{RE} > 0$. Then for all $a \in (1, +\infty)$

$$R_a(\mathcal{P} \,||\, \mathcal{Q}) \lesssim 1 + \frac{a\delta_{RE}^2}{2}.$$

The Klein Sampler [Kle00, GPV08] was analyzed in [Pre17] with respect to its relative error and Rényi divergence.

Lemma 9 (Relative Error of Klein Sampler [Pre15, Pre17]). Let n be a positive integer and $\epsilon \in (0, 1/4)$. Then the *relative error* of the Klein Sampler $\mathsf{PreSmp}(\boldsymbol{B}, s, \boldsymbol{c})$ for any basis $\boldsymbol{B}$, standard deviation $s \geq \eta_\epsilon(\mathbb{Z}^{2n}) \cdot \|\boldsymbol{B}\|_{GS}$, and arbitrary syndrome $\boldsymbol{c} \in \mathcal{R}_q$ is bounded by

$$\delta_{RE}\left(\mathsf{PreSmp}(\boldsymbol{B}, s, \boldsymbol{c}), \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}), s, \boldsymbol{c}}\right) \leq \left(\frac{1 + \epsilon/n}{1 - \epsilon/n}\right)^n - 1 \approx 2\epsilon.$$

Corollary 1 (Rényi Divergence of Klein Sampler). Let n be a be a positive integer, $a > 1$, and $\epsilon \in (0, 1/4)$. Then for the Klein Sampler $\mathsf{PreSmp}(\boldsymbol{B}, s, \boldsymbol{c})$, for any basis $\boldsymbol{B}$, standard deviation $s \geq \eta_\epsilon(2^{2n}) \cdot \|\boldsymbol{B}\|_{GS}$, and arbitrary syndrome $\boldsymbol{c} \in \mathcal{R}_q$, the *Rényi divergence* is bounded by

$$R_a\left(\mathsf{PreSmp}(\boldsymbol{B}, s, \boldsymbol{c}), \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}), s, \boldsymbol{c}}\right) \lesssim 1 + 2a\epsilon^2.$$

2.5 Hardness Assumptions

We will use the following definitions of the $\mathcal{R}$-**SIS** and $\mathcal{R}$-**ISIS** problems over NTRU lattices.

Definition 10 ($\mathcal{R}$-SIS, $\mathcal{R}$-ISIS). The *Ring Short Integer Solution* problem and the *Ring Inhomogeneous Short Integer Solution* problem relative to the NTRU trapdoor algorithm TpdGen with parameters $m, q > 0$ and $\alpha, B > 0$ are defined via the games $\mathcal{R}$-**SIS** and $\mathcal{R}$-**ISIS**, depicted in Figure 3. We define the advantages of A as

$$\mathrm{Adv}^{\mathcal{R}\text{-}\mathbf{SIS}}_{m,q,\alpha,B,\mathsf{A}} \coloneqq \Pr[\mathcal{R}\text{-}\mathbf{SIS}_{m,q,\alpha,B}(\mathsf{A}) \Rightarrow 1],$$
$$\mathrm{Adv}^{\mathcal{R}\text{-}\mathbf{ISIS}}_{m,q,\alpha,B,\mathsf{A}} \coloneqq \Pr[\mathcal{R}\text{-}\mathbf{ISIS}_{m,q,\alpha,B}(\mathsf{A}) \Rightarrow 1].$$

According to [LM06], $\mathcal{R}$-$\mathbf{SIS}_{m,q,\alpha,B}$ and $\mathcal{R}$-$\mathbf{ISIS}_{m,q,\alpha,B}$ are as hard as $\mathbf{SVP}_\gamma$ for $\gamma = \tilde{O}(nB)$. In particular, its hardness is independent of m. Note that we defined (I)SIS with respect to an NTRU key instead of a uniformly random element since (I)SIS is not believed to become easier in this case. However, if this should turn out to be wrong, the advantage of our definition can be trivially upper bounded by the sum of the decisional NTRU advantage and the usual ring (I)SIS definition.

```
Game $\mathcal{R}$-$\mathbf{SIS}_{m,q,\alpha,B}(\mathsf{A})$
for $i \in [m]$
    $(\boldsymbol{h}_i, \cdot, \cdot) \leftarrow \mathsf{TpdGen}(q, \alpha)$
$(\boldsymbol{u}, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_m) \xleftarrow{\$} \mathsf{A}(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_m)$
return $[\![\boldsymbol{u} + \sum_{i\in[m]} \boldsymbol{h}_i * \boldsymbol{v}_i = 0]\!] \wedge [\![0 < \|(\boldsymbol{u}, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)\|_2 \le B]\!]$
Game $\mathcal{R}$-$\mathbf{ISIS}_{m,q,\alpha,B}(\mathsf{A})$
for $i \in [m]$
    $(\boldsymbol{h}_i, \cdot, \cdot) \leftarrow \mathsf{TpdGen}(q, \alpha)$
$\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$
$(\boldsymbol{u}, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_m) \xleftarrow{\$} \mathsf{A}(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_m, \boldsymbol{c})$
return $[\![\boldsymbol{u} + \sum_{i\in[m]} \boldsymbol{h}_i * \boldsymbol{v}_i = \boldsymbol{c}]\!] \wedge [\![\|(\boldsymbol{u}, \boldsymbol{v}_1, \cdots \boldsymbol{v}_m)\|_2 \le B]\!]$
```

Figure 3. Games defining $\mathcal{R}$-$\mathbf{SIS}_{m,q,\alpha,B}$ and $\mathcal{R}$-$\mathbf{ISIS}_{m,q,\alpha,B}$.

3 New Security Arguments Using The Rényi Divergence

We introduce new techniques for applying Rényi arguments to prove the security of Falcon-type schemes. These general results may be useful for a broader class of schemes that rely on the Rényi divergence, with potential applications to works such as [EFG+22, ENS+23, GJK24, YJW23].

First, we extend [GPV08, Cor. 2.8], originally stated in terms of statistical distance, to accommodate the Rényi divergence. Such a lemma for Rényi order ∞ was stated in [BLL+15, Lem. 2.10]. Lemma 10 shows that a Gaussian sample over $\mathbf{\Lambda}$ is distributed almost-uniformly modulo a sublattice $\mathbf{\Lambda}'$, provided the standard deviation exceeds the smoothing parameter of $\mathbf{\Lambda}'$.

Lemma 10 (Rényi Divergence of Gaussian Sample over $\mathbf{\Lambda}/\mathbf{\Lambda}'$ (adapted from [GPV08, Cor. 2.8])). Let $\mathbf{\Lambda}, \mathbf{\Lambda}'$ be n-dimensional full-rank lattices with $\mathbf{\Lambda}' \subseteq \mathbf{\Lambda}$. Then for any $a \in (1, \infty)$, $\epsilon \in (0, \frac{1}{2})$, any $s \geq \eta_\epsilon(\mathbf{\Lambda}')$, and any $c \in \mathbb{R}^n$,

$$R_a\left(\mathcal{U}(\mathbf{\Lambda}/\mathbf{\Lambda}') \mid\mid \mathcal{D}_{\mathbf{\Lambda}/\mathbf{\Lambda}',s,c}\right) \lesssim 1 + \frac{2a\epsilon^2}{(1-\epsilon)^2}.$$

Proof. The quotient group $\mathbf{\Lambda}/\mathbf{\Lambda}'$ is defined as the additive group of cosets $x + \mathbf{\Lambda}', x \in \mathbf{\Lambda}$. Sampling from a discrete Gaussian over this quotient group we obtain that for any $x \in \mathbf{\Lambda}$

$$\mathcal{D}_{\mathbf{\Lambda}/\mathbf{\Lambda}',s,c}(x) = \frac{\rho_{s,c}(x + \mathbf{\Lambda}')}{\rho_{s,c}(\mathbf{\Lambda})}.$$

By assumption $\mathbf{\Lambda}' \subseteq \mathbf{\Lambda}$ which implies $\eta_\epsilon(\mathbf{\Lambda}) \leq \eta_\epsilon(\mathbf{\Lambda}') \leq s$. Therefore, we can apply Lemma 6 and get

$$\rho_{s,c}(\mathbf{\Lambda}) \in [1-\epsilon, 1+\epsilon] \cdot \frac{s^n}{\det(\mathbf{\Lambda})}.$$

Again, since $s \geq \eta_\epsilon(\mathbf{\Lambda})$

$$\rho_{s,c}(x + \mathbf{\Lambda}') \in [1-\epsilon, 1+\epsilon] \cdot \frac{s^n}{\det(\mathbf{\Lambda}')}.$$

Combining these results yields

$$\mathcal{D}_{\mathbf{\Lambda}/\mathbf{\Lambda}',s,c} \in \left[\frac{1-\epsilon}{1+\epsilon}, \frac{1+\epsilon}{1-\epsilon}\right] \cdot \frac{\det(\mathbf{\Lambda})}{\det(\mathbf{\Lambda}')}.$$

Since $\mathbf{\Lambda}$ and $\mathbf{\Lambda}'$ are full rank, their spans are the same ($\mathbb{R}^n$) and hence the size of their quotient group $\mathbf{\Lambda}/\mathbf{\Lambda}'$ is finite. Therefore, by [DD18, Lem. 10] we get that $|\mathbf{\Lambda}/\mathbf{\Lambda}'| = \frac{\det(\mathbf{\Lambda}')}{\det(\mathbf{\Lambda})}$. Computing the relative error between the Gaussian distribution and the uniform distribution $\mathcal{U}(\mathbf{\Lambda}/\mathbf{\Lambda}')(x) = \frac{1}{|\mathbf{\Lambda}/\mathbf{\Lambda}'|}$ gives

$$\frac{\mathcal{U}(|\mathbf{\Lambda}/\mathbf{\Lambda}'|)(x)}{\mathcal{D}_{\mathbf{\Lambda}/\mathbf{\Lambda}',s,c}(x)} \in \left[\frac{1-\epsilon}{1+\epsilon}, \frac{1+\epsilon}{1-\epsilon}\right] = \left[1 - \frac{2\epsilon}{1-\epsilon}, 1 + \frac{2\epsilon}{1-\epsilon}\right].$$

Applying Lemma 8 with $\delta = \frac{2\epsilon}{1-\epsilon}$, we obtain

$$R_a\left(\mathcal{U}(\mathbf{\Lambda}/\mathbf{\Lambda}') \mid\mid \mathcal{D}_{\mathbf{\Lambda}/\mathbf{\Lambda}',s,c}\right) \lesssim 1 + \frac{2a\epsilon^2}{(1-\epsilon)^2}.$$

This completes the proof. ■

Similarly, we extend [GPV08, Lem. 5.2], also originally stated in terms of statistical distance, to work with the Rényi divergence. The following Lemma states that an error vector taken from an appropriate Discrete Gaussian over $\mathbb{Z}^m$ corresponds to a nearly-uniform syndrome.

Lemma 11 (Rényi divergence (adapted from [GPV08, Lem 5.2])). If the columns of $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ generate $\mathbb{Z}_q^n$, $a \in (1,\infty)$, $\epsilon \in (0, \frac{1}{2})$, and $s \geq \eta_\epsilon(\mathbf{\Lambda}^\perp(\boldsymbol{A}))$; then for $e \sim \mathcal{D}_{\mathbb{Z}^m,s}$, the distribution $\mathcal{P} = \mathcal{U}(\mathbb{Z}_q^n)$, and the distribution $\mathcal{Q}$ of the syndromes $u = \boldsymbol{A}e \mod q$, it holds

$$R_a(\mathcal{P} \mid\mid \mathcal{Q}) \lesssim 1 + \frac{2a\epsilon^2}{(1-\epsilon)^2}.$$

Proof. For simplicity we denote $\mathbf{\Lambda}^\perp = \mathbf{\Lambda}^\perp(\boldsymbol{A})$. By assumption the set of all syndromes of $\boldsymbol{A}$ equals $\mathbb{Z}_q^N$, i.e. $\{\boldsymbol{A}e \mod q \mid e \in \mathbb{Z}^m\} = \mathbb{Z}_q^N$. Consider the quotient group $(\mathbb{Z}^m/\mathbf{\Lambda}^\perp)$ which is defined as the group of all cosets, i.e. $\{e + \mathbf{\Lambda}^\perp \mid e \in \mathbb{Z}^m\}$. This quotient group is isomorphic to the set of syndromes of $\boldsymbol{A}$ via the mapping $e + \mathbf{\Lambda}^\perp \mapsto \boldsymbol{A}e \mod q$, where $e \in \mathbb{Z}^m$. Hence, we have $\mathcal{P} \simeq \mathcal{U}(\mathbb{Z}^m/\mathbf{\Lambda}^\perp)$. Further, the distribution $\mathcal{D}_{\mathbb{Z}^m/\mathbf{\Lambda}^\perp,s} = \mathcal{D}_{\mathbb{Z}^m,s} \mod \mathbf{\Lambda}^\perp$ is the distribution of $e \sim \mathcal{D}_{\mathbb{Z}^m,s}$ reduced modulo $\mathbf{\Lambda}^\perp$. That is, the coset $e+\mathbf{\Lambda}^\perp$ for $e \sim \mathcal{D}_{\mathbb{Z}^m,s}$. Applying the above isomorphism, this distribution is isomorphic to distribution $\mathcal{Q}$. Finally we can apply Lemma 10 with $\mathbf{\Lambda} = \mathbb{Z}^m$, $\mathbf{\Lambda}' = \mathbf{\Lambda}^\perp$ and $c = 0$ to obtain the claim. ■

Corollary 2 (Rényi uniformity for NTRU). Let q be prime, $\boldsymbol{h} \in \mathcal{R}_q \setminus \{\mathbf{0}\}$, $a \in (1,\infty)$, $\epsilon \in (0, \frac{1}{2})$, $s \geq \eta_\epsilon(\mathbf{\Lambda}_{\boldsymbol{h},q})$, $\mathcal{P} = \mathcal{U}(\mathcal{R}_q)$, and $\mathcal{Q}$ the distribution of $\boldsymbol{u} + \boldsymbol{v} * \boldsymbol{h} \mod q$ where $\boldsymbol{u}, \boldsymbol{v} \sim \mathcal{D}_{\mathcal{R},s}$. Then it holds

$$R_a(\mathcal{P} \mid\mid \mathcal{Q}) \lesssim 1 + \frac{2a\epsilon^2}{(1-\epsilon)^2}.$$

Proof. Elements in $\mathcal{R}$ are polynomials of degree n that can be described via their anticirculant matrix $\mathcal{A}(\cdot) \in \mathbb{Z}^{n\times n}$. For q prime and $\boldsymbol{h} \in \mathcal{R}_q \setminus \{\mathbf{0}\}$, we consider matrix $\boldsymbol{A} = \left[I_N | \mathcal{A}(\boldsymbol{h})\right] \in \mathbb{Z}^{n\times 2n}$ that defines the NTRU lattice $\mathbf{\Lambda}_{\boldsymbol{h},q} = \mathbf{\Lambda}^\perp(\boldsymbol{A})$. By Lemma 1 the anticiruclant matrices with matrix addition and multiplication form a ring that is isomorphic to $\mathcal{R}$. In particular, this holds for the anticirculant of samples $e = (e_1, e_2)$ with $e_i \sim \mathcal{D}_{\mathbb{Z}^n,s}$ and $(\boldsymbol{u}, \boldsymbol{v})$ with $\boldsymbol{u}, \boldsymbol{v} \sim \mathcal{D}_{\mathcal{R},s}$ as well as for the resulting distributions $\boldsymbol{A} \cdot \mathcal{A}(e) \mod q$ and the distribution of $\boldsymbol{z}$ such that $\mathcal{A}(\boldsymbol{z}) = \boldsymbol{A}\begin{bmatrix}\mathcal{A}(\boldsymbol{u}) \\ \mathcal{A}(\boldsymbol{v})\end{bmatrix} = \mathcal{A}(\boldsymbol{u}) + \mathcal{A}(\boldsymbol{h}) \cdot \mathcal{A}(\boldsymbol{v}) \mod q$. The latter distribution is equivalent to $\mathcal{Q}$. Finally, due to its special structure with identity I_N on the left, $\boldsymbol{A}$ generates $\mathbb{Z}_q^n$ such that we can apply Lemma 11 to conclude the proof. ■

The next lemma shows that the tailbounds of two distributions with a small relative error are close.[5]

Lemma 12 (Rényi Divergence for Tailbounds). Let $\mathcal{P}$ and $\mathcal{Q}$ be two distributions with $\sup(\mathcal{P}) = \sup(\mathcal{Q}) = \mathbb{Z}^n$ such that their relative error is bounded by $\frac{\mathcal{P}}{\mathcal{Q}} \leq 1 + \delta$. Then for any $\beta \geq 0$,

$$\Pr_{x\leftarrow\mathcal{P}}[\|x\|_2 > \beta] \leq \Pr_{x\leftarrow\mathcal{Q}}[\|x\|_2 > \beta] \cdot (1 + \delta).$$

[5] Note that the relative error is related to Rényi arguments via Lemma 8.

Proof. We can use the relative error to upper bound the Rényi divergence of order ∞:

$$R_\infty(\mathcal{P} \mid\mid \mathcal{Q}) = \max_{x \in \sup(\mathcal{P})} \frac{\mathcal{P}(x)}{\mathcal{Q}(x)} \leq (1+\delta).$$

Further, let E be the event that the drawn value x fulfils $\|x\|_2 > \beta$. Applying the probability preservation for R_∞ (Lemma 7) we obtain

$$\Pr_{x \leftarrow \mathcal{Q}}[\|x\|_2 > \beta] = \mathcal{Q}(E) \geq \frac{\mathcal{P}(E)}{R_\infty(\mathcal{P} \mid\mid \mathcal{Q})} \geq \Pr_{x \leftarrow \mathcal{P}}[\|x\|_2 > \beta]/(1+\delta).$$

■

For the Rényi divergence, the order a can take any value in $(1, \infty)$, where a smaller a offers better efficiency, and a larger a enables a tighter proof. The description of the lemma is chosen to match statements usually occurring in a security bound (compare for example Section 4.2). For two events $\mathcal{E}_1$ and $\mathcal{E}_2$, Lemma 13 states the minimal number of bits that are lost when moving from $\mathcal{E}_1$ to $\mathcal{E}_2$. Optimising the Rényi order was previously considered in [TT15].

Lemma 13 (Optimal Rényi Order). For $\lambda \in \mathbb{N}$, let $\mathcal{E}_1, \mathcal{E}_2$ be two events such that $\Pr[\mathcal{E}_1] \geq 2^{-\lambda}$. Assume that for any $Q \in \mathbb{N}$ and $a \in (1, \infty)$ the Rényi divergence between two arbitrary distributions is at most $R_a \in [1, \infty)$, and

$$\Pr[\mathcal{E}_2] \leq R_a^Q \cdot \Pr[\mathcal{E}_1]^{\frac{a-1}{a}} \quad \forall\, a > 1.$$

Then

$$-\log(\Pr[\mathcal{E}_2]) \leq -\log(\Pr[\mathcal{E}_1]) - \min_{a>1}\left\{Q \log R_a + \frac{\lambda}{a}\right\}.$$

Proof. By assumption it holds $\Pr[\mathcal{E}_1] \geq 2^{-\lambda}$. Minimising for $a > 1$ yields

$$\begin{aligned}\Pr[\mathcal{E}_2] &\leq \min_{a>1}\left\{R_a^Q \cdot \Pr[\mathcal{E}_1]^{\frac{a-1}{a}}\right\} = \min_{a>1}\left\{R_a^Q \cdot \Pr[\mathcal{E}_1]^{-1/a}\right\} \cdot \Pr[\mathcal{E}_1] \\ &\leq \min_{a>1}\left\{R_a^Q \cdot 2^{\lambda/a}\right\} \cdot \Pr[\mathcal{E}_1].\end{aligned}$$

In other words, this gives at least

$$-\log(\Pr[\mathcal{E}_1]) - \min_{a>1}\left\{Q \log R_a + \frac{\lambda}{a}\right\}$$

bits success probability for $\mathcal{E}_2$. ■

4 CoreFalcon$^+$: A Framework for Falcon

Let n be a power of 2, q prime, and $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$. Let $\alpha \in \mathbb{R}^{>1}$ (basis quality), $\beta \in \mathbb{R}^{>0}$ (signature norm bound), $s \in \mathbb{R}^{>0}$ (Gaussian standard deviation), and $k \in \mathbb{N}$ (size of seed) be fixed parameters. Let $\mathsf{TpdGen} : \mathcal{R} \times \mathbb{R} \times \mathbb{Z} \to \mathcal{R}^4$ be a trapdoor generation algorithm, let $\mathsf{PreSmp} : \mathbb{Z}^{2n \times 2n} \times \mathbb{R} \times \mathcal{R}^2 \to \mathcal{R}^2$ be a preimage sampling algorithm, and $\mathsf{H} : \mathcal{R}_q \times \{0,1\}^k \times \mathcal{M} \to \mathcal{R}_q$ be a hash function. The defining algorithms of signature schemes CoreFalcon$^+$ and CoreFalcon are given in Figure 4.

Note that CoreFalcon$^+$ is a slight modification of CoreFalcon: In signing Sgn^+ of CoreFalcon$^+$, picking the random seed r and computing the ring element $\boldsymbol{c} = \mathsf{H}(pk, r, m)$ is performed inside of the repeat loop (lines 15-19), whereas CoreFalcon picks one fixed seed r. This modification is not only conceptual, see the discussion below.

```
$\underline{\mathsf{Gen}}$
$(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{F}, \boldsymbol{G}) \xleftarrow{\$} \mathsf{TpdGen}(\mathcal{R}, \alpha, q)$
$\boldsymbol{B} := \left[\begin{array}{c|c} \mathcal{A}(\boldsymbol{g}) & \mathcal{A}(\boldsymbol{G}) \\ \hline -\mathcal{A}(\boldsymbol{f}) & -\mathcal{A}(\boldsymbol{F}) \end{array}\right] \in \mathbb{Z}^{2n \times 2n}$
$\boldsymbol{h} := \boldsymbol{g} * \boldsymbol{f}^{-1} \in \mathcal{R}_q$
$\mathbf{return}\ (sk := \boldsymbol{B}, pk := \boldsymbol{h})$

$\underline{\mathsf{Sgn}(sk = \boldsymbol{B}, m)}$
$r \xleftarrow{\$} \{0,1\}^k$
$\boldsymbol{c} := \mathsf{H}(pk, r, m) \in \mathcal{R}_q$
$\mathbf{repeat}$
    $(\boldsymbol{s}_1, \boldsymbol{s}_2) \xleftarrow{\$} \mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}, \boldsymbol{0}))$
$\mathbf{until}\ \|(\boldsymbol{s}_1, \boldsymbol{s}_2)\|_2 \leq \beta$
$\sigma := (r, \boldsymbol{s}_2) \in \{0,1\}^k \times \mathcal{R}$
$\mathbf{return}\ \sigma$
```

```
$\underline{\mathsf{Ver}(pk = \boldsymbol{h}, m, \sigma = (r, \boldsymbol{s}_2))}$
$\boldsymbol{c} := \mathsf{H}(pk, r, m)$
$\boldsymbol{s}_1 := \boldsymbol{c} - \boldsymbol{s}_2 * \boldsymbol{h} \mod q$
$\mathbf{return}\ [\![\|(\boldsymbol{s}_1, \boldsymbol{s}_2)\|_2 \leq \beta]\!]$

$\underline{\mathsf{Sgn}^+(sk = \boldsymbol{B}, m)}$
$\mathbf{repeat}$
    $r \xleftarrow{\$} \{0,1\}^k$
    $\boldsymbol{c} := \mathsf{H}(pk, r, m) \in \mathcal{R}_q$
    $(\boldsymbol{s}_1, \boldsymbol{s}_2) \xleftarrow{\$} \mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}, \boldsymbol{0}))$
$\mathbf{until}\ \|(\boldsymbol{s}_1, \boldsymbol{s}_2)\|_2 \leq \beta$
$\sigma := (r, \boldsymbol{s}_2) \in \{0,1\}^k \times \mathcal{R}$
$\mathbf{return}\ \sigma$
```

Figure 4. Construction of the CoreFalcon = (Gen, Sgn, Ver) and CoreFalcon$^+$ = (Gen, Sgn$^+$, Ver) signature schemes.

The NIST Falcon signature schemes, Falcon-512 and Falcon-1024, can be seen as specific instantiations of CoreFalcon.[6] Unfortunately, we are not able to analyse the security of CoreFalcon since picking the random seed r outside of the repeat loop crucially affects the distribution of the signature in a way we are not able to simulate. Instead, in the next section, we will provide a general security analysis of the CoreFalcon$^+$ framework and derive concrete security levels of the modifications Falcon$^+$-512 and Falcon$^+$-1024.

Note that our modular analysis can be applied to CoreFalcon$^+$ variants that use alternative samplers or key generation procedures, including recent approaches like [ENS$^+$23], which eliminate the need for floating-point arithmetic.

4.1 Falcon Parameter Sets

As discussed above, Falcon can be seen as CoreFalcon with two parameter sets [PFH$^+$22]; a smaller set with ring degree $n = 512$ (Falcon-512) targeting NIST security level I, and a larger set with ring degree $n = 1024$ (Falcon-1024), targeting NIST security level V. Both sets use the same modulus $q = 12289$. The smoothing parameter error is defined as $\epsilon = 1/\sqrt{Q_s \cdot \lambda}$, where Q_s represents the recommend maximum number of signing queries, set to 2^{64}, and λ is the security parameter, set to 128 for NIST level I and 256 for NIST level V. Given ϵ, the standard deviation is given by

$$s = \frac{1}{\pi}\sqrt{\frac{\ln(4n(1+1/\epsilon))}{2}} \cdot 1.17\sqrt{q}.$$

Thus, the standard deviation of signatures is lower bounded by the smoothing parameter multiplied by the Gram-Schmidt norm of the trapdoor. The maximum signature norm bound β is set using a fixed tailcut rate

[6] In the signing process for Falcon-512 and Falcon-1024, a (public) compression technique is applied to the signature, and the loop is repeated until the signature reaches the desired compression level. This modification is mainly conceptual, as with the parameters of Falcon, the compressed signature typically reaches a sufficiently small size with high probability. Furthermore, CoreFalcon includes the public key in the hash function H, whereas Falcon-512 and Falcon-1024 do not. Including the public key in the hash function to make it key-contributory is generally considered good cryptographic engineering, as it enhances multi-user security bounds. Moreover, including the public key in the hash, as done in the Pornin-Stern transformation [PS05], has been shown to provide additional security properties beyond unforgeability [CDF$^+$21, DFF24].

$\tau = 1.1$, resulting in $\beta = \tau s \sqrt{2n}$. An overview of the relevant parameters of Falcon-512 and Falcon-1024 can be found in Table 3. We define Falcon^{+}-512 and Falcon^{+}-1024 using the CoreFalcon^{+} framework, instantiated with the parameters from Table 3.

Falcon uses the FFO sampler to instantiate the preimage sampler PreSmp. Since a formal analysis of the FFO sampler is lacking, we base our analysis and security estimation on the Klein sampler's [Kle00] analysis from [Pre17] which is expected to closely approximate the FFO sampler. We stress that future work is needed to analyse the FFO sampler, which may slightly alter the results presented here. However, this only affects the parameter selection, in particular the optimisation of the Rényi order. The rest of our analysis remains unchanged, as it is modular, general, and parameterised by the sampler.

Parameter	**NIST Level**	**Falcon-512**	**Falcon-1024**
	Description	**I**	**V**
n	Degree of ring $\mathcal{R}$	512	1024
q	Modulus	12289	
ϵ	Smoothing parameter error	$2^{-35.5}$	2^{-36}
s	Standard deviation	165.736617183	168.388571447
τ	Tailcut rate	1.1	
β	Max. signature norm bound	5833.93	8382.44
k	Bit size of the salt	320	

Table 3. Parameter sets for Falcon-512/Falcon^{+}-512 and Falcon-1024/Falcon^{+}-1024 [PFH^{+}22, Tab. 3.3].

4.2 Security Bounds for CoreFalcon^{+}

In this section we present two theorems that quantify the concrete security of CoreFalcon^{+} in the random oracle model. We refer to Theorem 1 as *tight* because the bound is independent of the number of queries to the random oracle. However, it does involve some loose Rényi terms, which limits its applicability to the proposed Falcon parameters. Conversely, Theorem 2 offers a tighter Rényi bound, making it suitable for analysing the Falcon parameters. However, it is considered *non-tight* due to the multiplicative security loss in the number of queries to the random oracle.

Theorem 1 (Tight Unforgeability). For any adversary A, making at most Q_s signing queries and Q_{H} random oracle queries, against the **SUF-CMA** security of CoreFalcon^{+} (Figure 4) in the random oracle model, there exists adversary B against $\mathcal{R}$-**SIS** with $t_{\mathsf{A}} \approx t_{\mathsf{B}}$ such that for all constants $C_s \in \mathbb{N}^{\geq 1}$ and $a_u, a_p \in \mathbb{R}^{>1}$ it holds

$$\begin{aligned}
&\mathrm{Adv}^{Q_s\text{-}\mathbf{SUF\text{-}CMA}}_{\textsc{CoreFalcon}^{+},\mathsf{A}} \leq \\
&(1 - p_{\mathsf{PreSmp},\beta})^{-1} \left(r_u^{C_s Q_s + Q_{\mathsf{H}}} \cdot \left(r_p^{C_s Q_s + 1} \cdot \left(\mathrm{Adv}^{\mathcal{R}\text{-}\mathbf{SIS}}_{1,q,\alpha,2\beta,\mathsf{B}} + 2^{-n} \right) \right)^{\frac{a_p - 1}{a_p}} \right)^{\frac{a_u - 1}{a_u}} \\
&+ Q_s \cdot p^{C_s}_{\mathsf{PreSmp},\beta} + \frac{Q_s (C_s Q_s + Q_{\mathsf{H}})}{2^k} \quad ,
\end{aligned}$$

where

- $p_{\mathsf{PreSmp},\beta} := \max_{\boldsymbol{c} \in \mathcal{R}_q} \Pr_{(\boldsymbol{s}_1, \boldsymbol{s}_2) \xleftarrow{\$} \mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}, \boldsymbol{0}))} [\|(\boldsymbol{s}_1, \boldsymbol{s}_2)\|_2 > \beta]$,
- $r_u = \max_{\boldsymbol{h} \neq \boldsymbol{0}} R_{a_u}(\mathcal{P} \parallel \mathcal{Q}_{\boldsymbol{h}})$ with $\mathcal{P} = \mathcal{U}(\mathcal{R}_q)$ and $\mathcal{Q}_{\boldsymbol{h}}$ the distribution of $\boldsymbol{u} + \boldsymbol{v} * \boldsymbol{h} \mod q$, where $\boldsymbol{u}, \boldsymbol{v} \sim \mathcal{D}_{\mathcal{R},s}$,

– $r_p = R_{a_p}(\mathsf{PreSmp}(\boldsymbol{B}, s, \cdot) \,||\, \mathcal{D}_{\mathcal{R},s})$.

Remark. Note that the bound of Theorem 1 holds for all choices of constants $C_s \in \mathbb{N}^{\geq 1}$ and $a_u, a_p \in \mathbb{R}^{>1}$. Will will refer to them as *proof constants*. Later, in Section 5 we will derive optimal choices for the proof constants that minimize the security loss for concrete relevant instantiations of CoreFalcon$^+$.

The proof of Theorem 1 can be found in Section A.

Theorem 2 (Theorem 2). For any adversary A, making at most Q_s signing queries and Q_H random oracle queries, against the **SUF-CMA/UF-CMA** security of CoreFalcon$^+$ (Figure 4) in the random oracle model, there exist adversaries B against $\mathcal{R}$-**SIS**, and C against $\mathcal{R}$-**ISIS** with $t_\mathsf{A} \approx t_\mathsf{B} \approx t_\mathsf{C}$ such that for all $C_s \in \mathbb{N}^{\geq 1}$ and $a_u, a_p \in \mathbb{R}^{>1}$ it holds

$$\begin{aligned}
&\mathrm{Adv}^{Q_s\text{-}\mathbf{SUF\text{-}CMA}}_{\textsc{CoreFalcon}^+,\mathsf{A}} \leq \\
&\quad \frac{(Q_\mathsf{H}+1)}{1-p_{\mathsf{PreSmp},\beta}} \cdot \left(r_u^{C_s Q_s + 1} \cdot \left(r_p^{C_s Q_s+1} \cdot \left(\mathrm{Adv}^{\mathcal{R}\text{-}\mathbf{SIS}}_{1,q,\alpha,2\beta,\mathsf{B}} + 2^{-n} \right) \right)^{\frac{a_p-1}{a_p}} \right)^{\frac{a_u-1}{a_u}} \\
&\quad + Q_s \cdot p^{C_s}_{\mathsf{PreSmp},\beta} + \frac{Q_s(C_s Q_s + Q_\mathsf{H})}{2^k} \quad ,
\end{aligned}$$

$$\begin{aligned}
\mathrm{Adv}^{Q_s\text{-}\mathbf{UF\text{-}CMA}}_{\textsc{CoreFalcon}^+,\mathsf{A}} &\leq (Q_\mathsf{H}+1) \cdot \left(r_u^{C_s Q_s} \cdot \left(r_p^{C_s Q_s} \cdot \mathrm{Adv}^{\mathcal{R}\text{-}\mathbf{ISIS}}_{1,q,\alpha,\beta,\mathsf{C}} \right)^{\frac{a_p-1}{a_p}} \right)^{\frac{a_u-1}{a_u}} \\
&\quad + Q_s \cdot p^{C_s}_{\mathsf{PreSmp},\beta} + \frac{Q_s(C_s Q_s + Q_\mathsf{H})}{2^k} \quad ,
\end{aligned}$$

where

- $p_{\mathsf{PreSmp},\beta} := \max_{\boldsymbol{c} \in \mathcal{R}_q} \Pr_{(\boldsymbol{s}_1,\boldsymbol{s}_2) \xleftarrow{\$} \mathsf{PreSmp}(\boldsymbol{B},s,(\boldsymbol{c},\boldsymbol{0}))}[\|(\boldsymbol{s}_1,\boldsymbol{s}_2)\|_2 > \beta]$,
- $r_u = \max_{\boldsymbol{h} \neq \boldsymbol{0}} R_{a_u}(\mathcal{P} \,||\, \mathcal{Q}_{\boldsymbol{h}})$ with $\mathcal{P} = \mathcal{U}(\mathcal{R}_q)$ and $\mathcal{Q}_{\boldsymbol{h}}$ the distribution of $\boldsymbol{u} + \boldsymbol{v} * \boldsymbol{h} \mod q$, where $\boldsymbol{u}, \boldsymbol{v} \sim \mathcal{D}_{\mathcal{R},s}$,
- $r_p = R_{a_p}(\mathsf{PreSmp}(\boldsymbol{B}, s, \cdot) \,||\, \mathcal{D}_{\mathcal{R},s})$.

The proof of Theorem 2 can be found in Appendix B.

5 Parameters and Analysing the Security Bound

In this section, we analyse the concrete security bounds for Falcon$^+$-512 and Falcon$^+$-1024 from Section 4.1. Recall that Falcon$^+$-512 and Falcon$^+$-1024 are slight modifications of Falcon-512 and Falcon-1024, respectively (with the same parameter sets), where signing includes picking the random seed inside of the repeat loop. Concretely, we will use the Theorems from Section 4.2 to derive the proof constants C_s, a_u, and a_p for an optimal tightness of the security proofs.

The Falcon specification suggest setting the Rényi order to $a_p = 2\lambda$, which is sufficient, but not ideal. As noted, achieving strong security bounds for Rényi arguments is difficult when the number of queries is larger than 2^{64} since the error ϵ of the smoothing parameter is set with respect to $Q_s = 2^{64}$. The bound of Theorem 1 involves a Rényi argument applied Q_H times, making Theorem 1 unsuitable for the current Falcon$^+$ parameters. Therefore, we focus our analysis on the alternative bound given in Theorem 2.

We proceed as follows: First, We estimate the SIS/ISIS bit security, accounting for the specific norm bound from the theorem. Next, we analyse the bound in Theorem 2, beginning with proof parameter C_s, denoting the maximal repetitions in the signing oracle. Next, based on the bit security of the (I)SIS term, we iteratively apply the Rényi arguments, carefully choosing the optimal orders a_u and a_p to minimise the security loss. Finally, we combine all results to calculate the final bit security, presenting an overview in Table 5 and Table 6, followed by a discussion of the findings.

Assumption	Bit security
$\mathcal{R}$-**ISIS**$_{B=\beta_I}$	120
$\mathcal{R}$-**SIS**$_{B=2\beta_I}$	95
$\mathcal{R}$-**ISIS**$_{B=\beta_V}$	278
$\mathcal{R}$-**SIS**$_{B=2\beta_V}$	0

Table 4. Bit security (Core-SVP) of the relevant $\mathcal{R}$-**ISIS**/$\mathcal{R}$-**SIS** instances with norm bound B.

5.1 Security of $\mathcal{R}$-(I)SIS

We estimate the security of the $\mathcal{R}$-**SIS** and $\mathcal{R}$-**ISIS** terms in our bounds. We consider the $\mathcal{R}$-**SIS** and $\mathcal{R}$-**ISIS** problem (as defined in Definition 10), parameterised by a trapdoor generation algorithm TpdGen with trapdoor quality α.

For strong unforgeability, Theorem 2 gives a reduction to $\mathcal{R}$-**SIS** with a norm bound of 2β. For plain unforgeability Theorem 2 provides a reduction to $\mathcal{R}$-**ISIS** with a norm bound of β. For the hardness of $\mathcal{R}$-**SIS** / $\mathcal{R}$-**ISIS** we use a ring dimension of $n = 512$ ($n = 1024$) and modulus $q = 12289$. The length bound $\beta = \tau s\sqrt{2n}$ results in $\beta_I = 5833.93$ for Falcon$^+$-512 and $\beta_V = 8382.44$ for Falcon$^+$-1024 (see Table 3).

We make the assumption that our $\mathcal{R}$-**SIS** and $\mathcal{R}$-**ISIS** instances are as hard as random **SIS** and **ISIS** instances. We estimate the security of **SIS** and **ISIS** using the "lattice-estimator" [APS15b, APS15a] with the `SIS.estimate.rough()` function, which computes the concrete bit security based on the *"core-SVP methodology"* from [ADPS16]. The resulting levels of bit security are summarised in Table 4. We refer to Figure 10 in Appendix C for the concrete prompts of the lattice estimator.[7]

5.2 Number Of Signing Repetitions C_s

The proof constant C_s defines the maximum number of repetitions to the signing oracle. Increasing C_s inflates all terms in the security bound, except for the term $Q_s \cdot p^{C_s}_{\mathsf{PreSmp},\beta}$. Hence, to obtain an optimal bound that fulfills the target security level λ, we have to find the smallest C_s such that $p^{C_s}_{\mathsf{PreSmp},\beta} \leq 2^{-\lambda}$. The following lemma establishes this for Falcon$^+$-512 and Falcon$^+$-1024.

Lemma 14 (Optimal C_s). For Falcon$^+$-512 and $\lambda = 128$ it holds

$$\arg\min_{C_s} \left\{ p^{C_s}_{\mathsf{PreSmp},\beta} \;\middle|\; p^{C_s}_{\mathsf{PreSmp},\beta} \leq 2^{-\lambda} \right\} = 9,$$

and for Falcon$^+$-1024 and $\lambda = 256$ it holds

$$\arg\min_{C_s} \left\{ p^{C_s}_{\mathsf{PreSmp},\beta} \;\middle|\; p^{C_s}_{\mathsf{PreSmp},\beta} \leq 2^{-\lambda} \right\} = 5.$$

5.3 Rényi Terms

Falcon builds on the work of [Pre17, Lem. 6] which suggests that setting $a_p = 2\lambda$ *"seems to be good compromise"*. Although this is true for certain problem instantiations, Lemma 13 makes this choice less ad hoc and allows us to set the order of the Rényi divergence optimally similar to [TT15]. We start with optimising the Rényi order for the unforgeability bound (Theorem 2), i.e., the reduction to $\mathcal{R}$-**ISIS**.

Falcon$^+$-512. We start with the advantage for $\mathcal{R}$-**ISIS** which gives 120 bits security, so for the inner most part of the bound we have to preserve at most $\lambda = 120$ bits of security. Since the security is already low, we

[7] The lattice estimator results in 228 bits of security for Falcon$^+$-1024 when $n = 1024$ and $B = 2\beta$. This is clearly incorrect, as $2\beta = 16764.88 > q = 12289$.

decide to choose the maximum number of signing queries to be lower than the recommended value of 2^{64} to get a tighter proof, namely $Q_s = 2^{57}$ which is the largest number to get an optimal Rényi loss of at most 1 bit.

Corollary 3 (Rényi Loss for Falcon$^+$-512 (Preimage Sampler)). For $\varepsilon \geq 2^{-\lambda} = 2^{-120}$, $r_p = R_{a_p}(\mathsf{PreSmp} \,||\, \mathcal{D})$, $C_s Q_s = 9 \cdot 2^{57}$, and the parameters for Falcon$^+$-512, the Rényi argument for

$$r_p^{C_s Q_s} \varepsilon^{\frac{a_p - 1}{a_p}}$$

loses at most 1 bit for an order $a_p \approx 275.15$.

Corollary 4 (Rényi Loss for Falcon$^+$-512 (Uniformity)). For $\varepsilon \geq 2^{-\lambda} = 2^{-119}$, $r_u = R_{a_u}(\mathcal{U}(\mathcal{R}_q) \,||\, \mathcal{U}_{\boldsymbol{h}})$, $C_s Q_s = 9 \cdot 2^{57}$, and the parameters for Falcon$^+$-512, the Rényi argument for

$$r_u^{C_s Q_s} \varepsilon^{\frac{a_u - 1}{a_u}}$$

loses at most 1 bit for an order $a_u \approx 274.01$.

Proof. Similar to the proof of Corollary 3 except that the Rényi divergence is upper bounded using Corollary 2. ■

Falcon$^+$-1024. We apply the same arguments as for Falcon$^+$-512. For the $\mathcal{R}$-**ISIS** term we obtain a security of 278 bits, i.e. we can assume that the Rényi argument of the preimage sampler needs to preserve at most $\lambda = 278$ bits.

Corollary 5 (Rényi Loss for Falcon$^+$-1024 (Preimage Sampler)). For $\varepsilon \geq 2^{-\lambda} = 2^{-278}$, $r_p = R_{a_p}(\mathsf{PreSmp} \,||\, \mathcal{D})$, $C_s Q_s = 5 \cdot 2^{64}$, and the parameters for Falcon$^+$-1024, the Rényi argument for

$$r_p^{C_s Q_s} \varepsilon^{\frac{a_p - 1}{a_p}}$$

loses at most 8 bit for an order $a_p \approx 70.24$.

Proof. The proof works as the proof of Corollary 3 with different parameters. ■

Since we lost already 8 bits when unfolding the Rényi argument for the preimage sampler, we need to apply the following corollary with a security level of only 270 bits.

Corollary 6 (Rényi Loss for Falcon$^+$-1024 (Uniformity)). For $\varepsilon \geq 2^{-\lambda} = 2^{-270}$, $r_u = R_{a_u}(\mathcal{U}(\mathcal{R}_q) \,||\, \mathcal{U}_{\boldsymbol{h}})$, $C_s Q_s = 5 \cdot 2^{64}$, and the parameters for Falcon$^+$-1024, the Rényi argument for

$$r_u^{C_s Q_s} \varepsilon^{\frac{a_u - 1}{a_u}}$$

loses at most 8 bit for an order $a_u \approx 69.22$.

Proof. The proof works as the proof of Corollary 4 with different parameters. ■

Other Bounds and Number of Signing Queries. The optimal Rényi orders for the strong unforgeability bound (Theorem 2) as well as for different choices of the maximum number of signing queries Q_s can be computed in the same way. We give an overview in the following section.

NIST Level I: $\mathcal{R} = \mathbb{Z}_{12289}[X]/(X^{512}+1)$				
Notion Parameter	**UF-CMA**		**SUF-CMA**	
Max repetitions, $C_s(128)$	9			
$\mathcal{R}$-**ISIS**/$\mathcal{R}$-**SIS** length bound B	$\beta = 5833.93$		$2\beta = 11667.86$	
Bit security (core-SVP), $\mathcal{R}$-**ISIS**$_{m=1,q=q,\alpha=1.17,B=\beta}$	120		—	
Bit security (core-SVP), $\mathcal{R}$-**SIS**$_{m=1,q=q,\alpha=1.17,B=2\beta}$	—		95	
Max Signing queries, Q_s	2^{57}	2^{64}	2^{58}	2^{64}
Rényi Order, a_p	275.15	24.32	173.11	21.64
Rényi Order, a_u	274.01	23.29	172.20	20.59
Bits lost from Rényi a_p	1	10	1	9
Bits lost from Rényi a_u	1	10	1	9
Final bit security	**118**	**100**	**93**	**75**

Table 5. Provable security level of Falcon$^+$-512.

NIST Level V: $\mathcal{R} = \mathbb{Z}_{12289}[X]/(X^{1024}+1)$		
Notion Parameter	**UF-CMA**	**SUF-CMA**
Max repetitions, $C_s(256)$	5	
$\mathcal{R}$-**ISIS**/$\mathcal{R}$-**SIS** length bound B	$\beta = 8382.44$	$2\beta = 16764.87$
Bit security (core-SVP), $\mathcal{R}$-**ISIS**$_{m=1,q=q,\alpha=1.17,B=\beta}$	278	—
Bit security (core-SVP), $\mathcal{R}$-**SIS**$_{m=1,q=q,\alpha=1.17,B=2\beta}$	—	0
Max Signing queries, Q_s	2^{64}	n/a
Rényi Order, a_p	70.24	
Rényi Order, a_u	69.22	
Bits lost from Rényi a_p	8	
Bits lost from Rényi a_u	8	
Final bit security	**256 (262)**[8]	**0**

Table 6. Provable security level of Falcon$^+$-1024.

5.4 Final Security and Discussion

To finalise the analysis of the bounds of Theorem 2, we note that the term $Q_s(C_s Q_s + Q_{\mathsf{H}})/2^k$ results in λ bits of security if $k \geq \log(Q_s) + \lambda$. Falcon$^+$ achieves this for both parameter sets by choosing $k = 320$. The multiplicative loss of $(1 - p_{\mathsf{PreSmp},\beta})^{-1}$ amounts to a fraction of a bit (see Lemma 12) and the term 2^{-n} from the strong unforgeability bound is small enough for both parameter sets since the ring dimension is at least 512. An overview of our results from the previous subsections is provided in Table 5 for Falcon$^+$-512 and in Table 6 for Falcon$^+$-1024. Note that the computational term in the bound for Falcon$^+$-1024 provides 262 bits of security but the statistical terms as described before restrict the overall security to 256 bits. Below, we address key findings and issues and suggest possible solutions.

Strong Unforgeability. While the bit security for plain unforgeability is close to the target, the bit security for strong unforgeability in Falcon$^+$-512 is insufficient. Specifically, for $Q_s = 2^{64}$, the security level is only 75 bits, offering no meaningful security guarantee. For Falcon$^+$-1024, the situation is worse, as no security can be shown at all due to the norm bound 2β exceeding the modulus q. If strong unforgeability could be reduced to one-wayness (or ISIS) as in the case of plain unforgeability, instead of

[8] 262 bits refers to the bit security of the computational term. See Section 5.4 for more information.

collision resistance (or SIS), a smaller norm bound and better security would be achievable. However, current proof techniques cannot address this problem, and we believe it is unlikely to be feasible in general. Achieving strong unforgeability for Falcon$^+$ would therefore require increasing the ring dimension and modulus, leading to larger public keys and signatures.

Number of Signing Queries. For Falcon$^+$-512 we provide bit security estimates for both reduced and full 2^{64} signing queries, as required by NIST. The reason being that, the Falcon$^+$ parameters do not account for additional queries caused by the singing procedure's repetition. Keeping 2^{64} queries increase the Rényi loss, which is problematic since the security of ISIS is already tightly set to meet the target level. For Falcon$^+$-1024, the larger security margin between ISIS and the target security compensates for higher Rényi losses, so the issue is less critical. For this reason we also present the maximum number of signing queries that can be supported while having a Rényi loss of at most 1 bit. This issues is not an artifact of our proof strategy but arises from the inherent repetition in the signing procedure and the sensitivity of the Rényi arguments. To support 2^{64} signing queries with tight Rényi bounds, the smoothing parameter error ϵ would need to account for the maximum repetitions, leading to $\epsilon = (C_s Q_s \cdot \lambda)^{-1/2}$. This would, however, increase parameters such as the signature size.

Tightness Loss. Our evaluation incorporates the complete bounds from Theorem 2, except for the tightness loss of $Q_{\mathsf{H}} + 1$. From a theoretical standpoint, including the tightness loss would reduce security by 96 bits (assuming 2^{96} hash queries), clearly causing the scheme to miss its target security levels. However, since tightness losses involving random oracle guesses are often ignored in practice (e.g., Dilithium [LDK$^+$22]), we report the bit security without this loss, but stress that is should be considered when necessary. See Section 1.2 for further details. This issue could be mitigated by using Theorem 1, which offers a tight reduction. Yet, despite the tight reduction, the (constant) Rényi loss is exorbitantly high for the concrete parameters and provide no security guarantee. This may be an artifact of our proof strategy, and could be avoided by not applying Rényi arguments in the random oracle. However, we think that this is unlikely given the reliance on collision resistance and the behaviour of the Rényi divergence. To apply Theorem 1, to concrete parameters, Falcon$^+$ would need to set parameters allowing Rényi arguments for random oracle queries, not just signing queries, resulting in a smoothing parameter error of $\epsilon = 1/\sqrt{Q_{\mathsf{H}} \cdot \lambda}$. This adjustment would also increase the signature size.

Acknowledgements. The authors thank Joel Gärtner, Thomas Prest and Damien Stehlé for helpful feedback and comments. Phillip Gajland was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2092 CASA - 390781972. Jonas Janneck was supported by the European Union (ERC AdG REWORC - 101054911). Eike Kiltz was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2092 CASA - 390781972, and by the European Union (ERC AdG REWORC - 101054911).

References

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016: 25th USENIX Security Symposium*, pages 327–343, Austin, TX, USA, August 10–12, 2016. USENIX Association. (Cited on page 19.)

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th Annual ACM Symposium on Theory of Computing*, pages 99–108, Philadephia, PA, USA, May 22–24, 1996. ACM Press. doi:10.1145/237814.237838. (Cited on pages 3 and 5.)

[APS15a] Martin R. Albrecht, Rachel Player, and Sam Scott. Lattice estimator. https://github.com/malb/lattice-estimator, 2015. Commit: 14a362513c9197dd959bc72428425abe0309779a. (Cited on pages 7 and 19.)

[APS15b] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015. URL: https://doi.org/10.1515/jmc-2015-0016 [cited 2024-05-23], doi:doi:10.1515/jmc-2015-0016. (Cited on pages 7 and 19.)

[Ban93] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, December 1993. doi:10.1007/bf01445125. (Cited on page 10.)

[BBD$^+$23] 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. (Cited on page 8.)

[BDF$^+$11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69, Seoul, South Korea, December 4–8, 2011. Springer, Berlin, Heidelberg, Germany. doi:10.1007/978-3-642-25385-0_3. (Cited on page 5.)

[BDL$^+$12] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *Journal of Cryptographic Engineering*, 2(2):77–89, September 2012. doi:10.1007/s13389-012-0027-1. (Cited on page 7.)

[Beu22] Ward Beullens. Breaking rainbow takes a weekend on a laptop. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 464–479, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Cham, Switzerland. doi:10.1007/978-3-031-15979-4_16. (Cited on page 3.)

[BLL$^+$15] Shi Bai, Adeline Langlois, Tancrède Lepoint, Damien Stehlé, and Ron Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Part I*, volume 9452 of *Lecture Notes in Computer Science*, pages 3–24, Auckland, New Zealand, November 30 – December 3, 2015. Springer, Berlin, Heidelberg, Germany. doi:10.1007/978-3-662-48797-6_1. (Cited on pages 11 and 13.)

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. doi:10.1145/168588.168596. (Cited on page 5.)

[BR96] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416, Saragossa, Spain, May 12–16, 1996. Springer, Berlin, Heidelberg, Germany. doi:10.1007/3-540-68339-9_34. (Cited on pages 3 and 5.)

[BR04] Mihir Bellare and Phillip Rogaway. Code-based game-playing proofs and the security of triple encryption. Cryptology ePrint Archive, Report 2004/331, 2004. URL: https://eprint.iacr.org/2004/331. (Cited on page 8.)

[CD23] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 423–447, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-30589-4_15. (Cited on page 3.)

[CDF$^+$21] Cas Cremers, Samed Düzlü, Rune Fiedler, Marc Fischlin, and Christian Janson. BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures. In *2021 IEEE Symposium on Security and Privacy*, pages 1696–1714, San Francisco, CA, USA, May 24–27, 2021. IEEE Computer Society Press. doi:10.1109/SP40001.2021.00093. (Cited on page 16.)

[DD18] Daniel Dadush and Léo Ducas. Determinants, packing and covering, and the minkowski theorems, 2018. URL: https://homepages.cwi.nl/~dadush/teaching/lattices-2018/notes/lecture-2.pdf. (Cited on page 13.)

[DFF24] Samed Düzlü, Rune Fiedler, and Marc Fischlin. BUFFing FALCON without increasing the signature size. Cryptology ePrint Archive, Report 2024/710, 2024. URL: https://eprint.iacr.org/2024/710. (Cited on page 16.)

[DLP14] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over NTRU lattices. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 22–41, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Springer, Berlin, Heidelberg, Germany. doi:10.1007/978-3-662-45608-8_2. (Cited on pages 3, 5, and 11.)

[EFG$^+$22] Thomas Espitau, Pierre-Alain Fouque, François Gérard, Mélissa Rossi, Akira Takahashi, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Mitaka: A simpler, parallelizable, maskable variant of falcon. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022,* *Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 222–253, Trondheim, Norway, May 30 – June 3, 2022. Springer, Cham, Switzerland. doi:10.1007/978-3-031-07082-2_9. (Cited on pages 3 and 13.)

[ENS$^+$23] Thomas Espitau, Thi Thu Quyen Nguyen, Chao Sun, Mehdi Tibouchi, and Alexandre Wallet. Antrag: Annular NTRU trapdoor generation - making mitaka as secure as falcon. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part VII*, volume 14444 of *Lecture Notes in Computer Science*, pages 3–36, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore. doi:10.1007/978-981-99-8739-9_1. (Cited on pages 3, 13, and 16.)

[ENST23] Thomas Espitau, Guilhem Niot, Chao Sun, and Mehdi Tibouchi. SQUIRRELS — Square Unstructured Integer Euclidean Lattice Signature. Technical report, National Institute of Standards and Technology, 2023. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures. (Cited on page 4.)

[FH23] Serge Fehr and Yu-Hsuan Huang. On the quantum security of HAWK. In Thomas Johansson and Daniel Smith-Tone, editors, *Post-Quantum Cryptography - 14th International Workshop, PQCrypto 2023*, pages 405–416, College Park, USA, August 16–18, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-40003-2_15. (Cited on page 8.)

[GJK24] Phillip Gajland, Jonas Janneck, and Eike Kiltz. Ring signatures for deniable AKEM: Gandalf's fellowship. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part I*, volume 14920 of *Lecture Notes in Computer Science*, pages 305–338, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-68376-3_10. (Cited on pages 3 and 13.)

[GMLS02] SD Galbraith, J Malone-Lee, and NP Smart. Public key signatures in the multi-user setting. *Information Processing Letters*, 83 (5):263 – 266, June 2002. (Cited on page 7.)

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206, Victoria, BC, Canada, May 17–20, 2008. ACM Press. doi:10.1145/1374376.1374407. (Cited on pages 3, 5, 11, 12, 13, and 14.)

[HBD$^+$22] Andreas Hülsing, Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Jean-Philippe Aumasson, Bas Westerbaan, and Ward Beullens. SPHINCS$^+$. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. (Cited on page 3.)

[HHP$^+$03] Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. NTRUSIGN: Digital signatures using the NTRU lattice. In Marc Joye, editor, *Topics in Cryptology – CT-RSA 2003*, volume 2612 of *Lecture Notes in Computer Science*, pages 122–140, San Francisco, CA, USA, April 13–17, 2003. Springer, Berlin, Heidelberg, Germany. doi:10.1007/3-540-36563-X_9. (Cited on pages 3 and 5.)

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. (Cited on pages 3 and 5.)

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Third Algorithmic Number Theory Symposium (ANTS)*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, June 1998. (Cited on pages 3, 5, and 11.)

[Kim16] Kevin Kimball. Announcing request for nominations for public-key post-quantum cryptographic algorithms. Technical report, National Institute of Standards and Technology, 2016. available at https://www.federalregister.gov/d/2016-30615. (Cited on page 3.)

[Kle00] Philip N. Klein. Finding the closest lattice vector when it's unusually close. In David B. Shmoys, editor, *11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 937–941, San Francisco, CA, USA, January 9–11, 2000. ACM-SIAM. (Cited on pages 6, 7, 12, and 17.)

[LAZ19] Xingye Lu, Man Ho Au, and Zhenfei Zhang. Raptor: A practical lattice-based (linkable) ring signature. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *ACNS 19: 17th International Conference on Applied Cryptography and Network Security*, volume 11464 of *Lecture Notes in Computer Science*, pages 110–130, Bogota, Colombia, June 5–7, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-21568-2_6. (Cited on page 3.)

[LDK$^+$22] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. (Cited on pages 3, 4, 7, and 22.)

[LM06] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact Knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006: 33rd International Colloquium on Automata, Languages and Programming, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155, Venice, Italy, July 10–14, 2006. Springer, Berlin, Heidelberg, Germany. doi:10.1007/11787006_13. (Cited on pages 12 and 36.)

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 35–54, Athens, Greece, May 26–30, 2013. Springer, Berlin, Heidelberg, Germany. doi:10.1007/978-3-642-38348-9_3. (Cited on pages 3 and 6.)

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Heidelberg, Germany. doi:10.1007/978-3-642-29011-4_43. (Cited on pages 10 and 11.)

[MMP$^+$23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-30589-4_16. (Cited on page 3.)

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th Annual Symposium on Foundations of Computer Science*, pages 372–381, Rome, Italy, October 17–19, 2004. IEEE Computer Society Press. doi:10.1109/FOCS.2004.72. (Cited on page 11.)

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007. arXiv:https://doi.org/10.1137/S0097539705447360, doi:10.1137/S0097539705447360. (Cited on page 11.)

[MW17] Daniele Micciancio and Michael Walter. Gaussian sampling over the integers: Efficient, generic, constant-time. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 455–485, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-63715-0_16. (Cited on page 12.)

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

[PFH$^+$22] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. (Cited on pages 5, 16, and 17.)

[Pre15] Thomas Prest. *Gaussian sampling in lattice-based cryptography.* PhD thesis, Ecole normale supérieure-ENS PARIS, 2015. (Cited on pages 11 and 12.)

[Pre17] Thomas Prest. Sharper bounds in lattice-based cryptography using the Rényi divergence. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 347–374, Hong Kong, China, December 3–7, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-70694-8_13. (Cited on pages 6, 7, 11, 12, 17, and 19.)

[PS05] Thomas Pornin and Julien P. Stern. Digital signatures do not guarantee exclusive ownership. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *ACNS 05: 3rd International Conference on Applied Cryptography and Network Security*, volume 3531 of *Lecture Notes in Computer Science*, pages 138–150, New York, NY, USA, June 7–10, 2005. Springer, Berlin, Heidelberg, Germany. doi:10.1007/11496137_10. (Cited on page 16.)

[Rén61] Alfréd Rényi. On measures of entropy and information. Proc. 4th Berkeley Symp. Math. Stat. Probab. 1, 547-561 (1961)., 1961. (Cited on page 11.)

[Rob23] Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer*

Science, pages 472–503, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-30589-4_17. (Cited on page 3.)

[RSW18] Miruna Rosca, Damien Stehlé, and Alexandre Wallet. On the ring-LWE and polynomial-LWE problems. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 146–173, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-319-78381-9_6. (Cited on page 3.)

[SAB$^+$22] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, Damien Stehlé, and Jintai Ding. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. (Cited on page 3.)

[SS11] Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 27–47, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Heidelberg, Germany. doi:10.1007/978-3-642-20465-4_4. (Cited on pages 3 and 6.)

[TT15] Katsuyuki Takashima and Atsushi Takayasu. Tighter security for efficient lattice cryptography via the Rényi divergence of optimized orders. In Man Ho Au and Atsuko Miyaji, editors, *ProvSec 2015: 9th International Conference on Provable Security*, volume 9451 of *Lecture Notes in Computer Science*, pages 412–431, Kanazawa, Japan, November 24–26, 2015. Springer, Cham, Switzerland. doi:10.1007/978-3-319-26059-4_23. (Cited on pages 15 and 19.)

[YJW23] Yang Yu, Huiwen Jia, and Xiaoyun Wang. Compact lattice gadget and its applications to hash-and-sign signatures. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 390–420, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-38554-4_13. (Cited on pages 3 and 13.)

A Proof of Theorem 1

Proof. Consider the sequence of games depicted in Figure 5.

Game G_0. This is the strong unforgeability game for CoreFalcon$^+$:

$$\Pr[\mathsf{G}_0^{\mathsf{A}} \Rightarrow 1] = \mathrm{Adv}^{Q_s\text{-}\mathbf{SUF\text{-}CMA}}_{\textsc{CoreFalcon}^+,\mathsf{A}}.$$

```
Games $\mathsf{G}_0 - \mathsf{G}_7$
$\mathcal{H}, \mathcal{Q} \leftarrow \emptyset$
$(\boldsymbol{B}, \boldsymbol{h}) \xleftarrow{\$} \mathsf{Gen}$
$(m^\star, \sigma^\star) \xleftarrow{\$} \mathsf{A}^{\mathsf{Sgn}(\cdot),\mathsf{H}(\cdot,\cdot,\cdot)}(\boldsymbol{h})$
parse $\sigma^\star \to (r^\star, \boldsymbol{s}_2^\star)$
$\boldsymbol{c}^\star := \mathsf{H}(\boldsymbol{h}, r^\star, m^\star)$                                  // $\mathsf{G}_3 - \mathsf{G}_7$
$(\boldsymbol{s}_1', \boldsymbol{s}_2') \xleftarrow{\$} \mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}^\star, \boldsymbol{0}))$                 // $\mathsf{G}_3 - \mathsf{G}_7$
$(\boldsymbol{s}_1', \boldsymbol{s}_2') \xleftarrow{\$} \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}),s,(\boldsymbol{c}^\star,\boldsymbol{0})}$                        // $\mathsf{G}_5 - \mathsf{G}_7$
find $(\boldsymbol{s}_1', \boldsymbol{s}_2') : (\boldsymbol{c}^\star, \boldsymbol{h}, r^\star, m^\star, \boldsymbol{s}_1', \boldsymbol{s}_2') \in \mathcal{H}$      // $\mathsf{G}_6 - \mathsf{G}_7$
if $\|(\boldsymbol{s}_1', \boldsymbol{s}_2')\|_2 > \beta$                                    // $\mathsf{G}_3 - \mathsf{G}_7$
  abort                                                  // $\mathsf{G}_3 - \mathsf{G}_7$
$\boldsymbol{s}_1^\star := \boldsymbol{c}^\star - \boldsymbol{s}_2^\star * \boldsymbol{h} \mod q$                                // $\mathsf{G}_7$
if $(\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star) = (\boldsymbol{s}_1', \boldsymbol{s}_2') \wedge (m^\star, \sigma^\star) \notin \mathcal{Q}$                 // $\mathsf{G}_7$
  abort                                                      // $\mathsf{G}_7$
return $[\![\mathsf{Ver}(\boldsymbol{h}, m^\star, \sigma^\star) = 1 \wedge (m^\star, \sigma^\star) \notin \mathcal{Q}]\!]$

Oracle $\mathtt{Sgn}(m)$
$cnt := 0$                                               // $\mathsf{G}_1 - \mathsf{G}_7$
repeat
  $cnt \leftarrow cnt + 1$                                     // $\mathsf{G}_1 - \mathsf{G}_7$
  if $cnt > C_s$                                           // $\mathsf{G}_1 - \mathsf{G}_7$
    abort                                                // $\mathsf{G}_1 - \mathsf{G}_7$
  $r \xleftarrow{\$} \{0,1\}^k$
  if $\exists\, \boldsymbol{c} : (\boldsymbol{c}, \boldsymbol{h}, r, m, \cdot, \cdot) \in \mathcal{H}$                       // $\mathsf{G}_2 - \mathsf{G}_7$
    abort                                                // $\mathsf{G}_2 - \mathsf{G}_7$
  $\boldsymbol{c} := \mathsf{H}(\boldsymbol{h}, r, m)$
  $(\boldsymbol{s}_1, \boldsymbol{s}_2) \xleftarrow{\$} \mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}, \boldsymbol{0}))$              // $\mathsf{G}_0 - \mathsf{G}_4$
  $(\boldsymbol{s}_1, \boldsymbol{s}_2) \xleftarrow{\$} \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}),s,(\boldsymbol{c},\boldsymbol{0})}$                      // $\mathsf{G}_5 - \mathsf{G}_7$
  $(\boldsymbol{s}_1, \boldsymbol{s}_2) \leftarrow \{(\boldsymbol{s}_1', \boldsymbol{s}_2') \mid (\boldsymbol{c}, \boldsymbol{h}, r, m, \boldsymbol{s}_1', \boldsymbol{s}_2') \in \mathcal{H}\}$  // $\mathsf{G}_6 - \mathsf{G}_7$
until $\|(\boldsymbol{s}_1, \boldsymbol{s}_2)\|_2 \le \beta$
$\sigma := (r, \boldsymbol{s}_2)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(m, \sigma)\}$
return $\sigma$

Oracle $\mathsf{H}(pk, r, m)$
if $\exists\, \boldsymbol{c} : (\boldsymbol{c}, pk, r, m, \cdot, \cdot) \in \mathcal{H}$
  return $\boldsymbol{c}$
$\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$                                           // $\mathsf{G}_0 - \mathsf{G}_3$
$\mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{c}, pk, r, m, \bot, \bot)\}$                    // $\mathsf{G}_0 - \mathsf{G}_3$
$\boldsymbol{s}_1, \boldsymbol{s}_2 \xleftarrow{\$} \mathcal{D}_{\mathcal{R},s}$                                 // $\mathsf{G}_4 - \mathsf{G}_7$
$\boldsymbol{c} \leftarrow \boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} \mod q$                          // $\mathsf{G}_4 - \mathsf{G}_7$
$\mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{c}, pk, r, m, \boldsymbol{s}_1, \boldsymbol{s}_2)\}$               // $\mathsf{G}_4 - \mathsf{G}_7$
return $\boldsymbol{c}$
```

Figure 5. Games for the proof of Theorem 1.

Game G_1. This game is identical to the previous one, except it aborts if the number of repetitions in the signing oracle exceeds threshold C_s.

Claim 1: $\left|\Pr\left[\mathsf{G}_0^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_1^{\mathsf{A}} \Rightarrow 1\right]\right| \le Q_s \cdot p_{\mathsf{PreSmp},\beta}^{C_s}$.

Proof. For one signing query the probability of an abort can be upper bounded by $p_{\mathsf{PreSmp},\beta}^{C_s}$. Since the counter is reset and the abort condition in each signing query is independent, we obtain the claim by applying a union bound over all signing queries. ■

Game G_2. This game is identical to the previous one except that it aborts in the signing oracle $\mathtt{Sgn}$ if there already exists a query to the random oracle on $\boldsymbol{h}$, the same salt r and message m (Line 22).

Claim 2: $\left|\Pr\left[\mathsf{G}_1^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_2^{\mathsf{A}} \Rightarrow 1\right]\right| \leq \frac{Q_s(C_s Q_s + Q_{\mathsf{H}})}{2^k}$.

Proof. The salt r is chosen uniformly at random from $\{0,1\}^k$ for each signing query. The total number of elements in $\mathcal{H}$ is at most $C_s Q_s + Q_{\mathsf{H}}$, as at most one element is added per query to H; and there are $C_s Q_s$ implicit queries via $\mathtt{Sgn}$ and Q_{H} direct ones. Thus, the probability that the freshly chosen salt was part of a previous query is at most $\frac{C_s Q_s + Q_{\mathsf{H}}}{2^k}$. For Q_s queries to the signing oracle $\mathtt{Sgn}$, we obtain the claimed bound. ■

Game G_3. This game is identical to the previous one except that the game computes the RO output $\boldsymbol{c}^\star$ that corresponds to the adversary's forgery. It then computes a preimage of $\boldsymbol{c}^\star$ with respect to $\boldsymbol{h}$ using the preimage sampler PreSmp with trapdoor $\boldsymbol{B}$. If the norm of the resulting preimage $(\boldsymbol{s}_1', \boldsymbol{s}_2')$ is larger than β, the game aborts.

Claim 3: $\Pr[\mathsf{G}_2^{\mathsf{A}} \Rightarrow 1] = (1 - p_{\mathsf{PreSmp},\beta})^{-1} \cdot \Pr[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1]$.

Proof. The probability that the abort event does not occur is $1 - p_{\mathsf{PreSmp},\beta}$. Since the preimage is computed after adversary A output their forgery and the winning condition is not affected by $(\boldsymbol{s}_1', \boldsymbol{s}_2')$, the abort event is independent of A's winning probability. Hence we can apply the multiplicative difference lemma to obtain the statement. ■

Game G_4. This game is the same as the previous one, except that the random oracle no longer returns a uniformly random element $\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$. Instead, it computes $\boldsymbol{c}$ as follows: It samples two elements $\boldsymbol{s}_1, \boldsymbol{s}_2$ from a Gaussian distribution $\mathcal{D}_{\mathcal{R},s}$ with standard deviation s over ring $\mathcal{R}$. Then, $\boldsymbol{c}$ is computed as $\boldsymbol{c} = \boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h}$ mod q, where $\boldsymbol{h}$ is the public key. For future use, $\boldsymbol{s}_1, \boldsymbol{s}_2$, along with the input and output to the random oracle, are stored in $\mathcal{H}$ (Line 37).

Claim 4: For distribution $\mathcal{P} := \mathcal{U}(\mathcal{R}_q)$, distribution $\mathcal{Q}_{\boldsymbol{h}}$ the distribution of $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} \mod q$ where $\boldsymbol{s}_1, \boldsymbol{s}_2 \xleftarrow{\$} \mathcal{D}_{\mathcal{R},s}$, and $a_u \in (1, \infty)$ it holds

$$\Pr[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1] \leq \left(R_{a_u}(\mathcal{P} \mid\mid \mathcal{Q}_{\boldsymbol{h}})^{C_s Q_s + Q_{\mathsf{H}}} \cdot \Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1]\right)^{\frac{a_u - 1}{a_u}}.$$

Proof. In Game G_3, the output distribution of the RO is the uniform distribution over $\mathcal{R}_q$, named $\mathcal{P}$. In Game G_4, we use the distribution of $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} \mod q$ where $\boldsymbol{s}_1, \boldsymbol{s}_2 \xleftarrow{\$} \mathcal{D}_{\mathcal{R},s}$, named $\mathcal{Q}_{\boldsymbol{h}}$ for simplicity. These underlying distributions are queried at most $Q := C_s Q_s + Q_{\mathsf{H}}$ times due to implicit RO queries from $\mathtt{Sgn}$ in Line 23 and explicit ones to H. By the probability preservation, data processing inequality, and multiplicativity of the Rényi divergence (Lemma 7) we obtain

$$\Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1] \geq \frac{\Pr[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1]^{\frac{a_u}{a_u - 1}}}{R_{a_u}(\mathsf{G}_3^{\mathsf{A}} \mid\mid \mathsf{G}_4^{\mathsf{A}})} \geq \frac{\Pr[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1]^{\frac{a_u}{a_u - 1}}}{R_{a_u}(\mathcal{P}^Q \mid\mid \mathcal{Q}_{\boldsymbol{h}}^Q)} = \frac{\Pr[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1]^{\frac{a_u}{a_u - 1}}}{R_{a_u}(\mathcal{P} \mid\mid \mathcal{Q}_{\boldsymbol{h}})^Q}.$$

This concludes the proof of Claim 4. ■

Game G_5*.* This game is identical to the previous one except that the output of the preimage sampler $\mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}, \boldsymbol{0}))$ is replaced by a Gaussian over the same lattice, standard deviation and center, namely $\mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}),s,(\boldsymbol{c},\boldsymbol{0})}$.

Claim 5: For the two distributions $\mathsf{PreSmp} := \mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}, \boldsymbol{0}))$, $\mathcal{D} := \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}),s,(\boldsymbol{c},\boldsymbol{0})}$, and for all $a_r \in (1, \infty)$ it holds

$$\Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1] \leq \left(R_{a_r}(\mathsf{PreSmp} \,||\, \mathcal{D})^{C_s Q_s + 1} \cdot \Pr[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1]\right)^{\frac{a_r - 1}{a_r}} .$$

Proof. The claims follows by analogous arguments as for the changes to Game G_4. Note that we have to replace $C_s \cdot Q_s$ queries in the signing oracle and one additional query after the output of the forgery. ■

Game G_6*.* This game is identical to the previous one except that the $\boldsymbol{s}_1, \boldsymbol{s}_2$ are not sampled from a Gaussian distribution centred at $(\boldsymbol{c}, \boldsymbol{0})$ as before. Instead, the preimage of $\boldsymbol{c}$ stored in the set $\mathcal{H}$ during the random oracle query is used. Such an element must exist due to the changes in G_4.

Claim 6: $\Pr[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1] = \Pr[\mathsf{G}_6^{\mathsf{A}} \Rightarrow 1]$.

Proof. We need to show that the distributions of the games are the same. The RO output $\boldsymbol{c}$ is the same in both games. In G_5, the singing oracle outputs $(\boldsymbol{s}_1, \boldsymbol{s}_s) \sim \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}),s,(\boldsymbol{c},\boldsymbol{0})}$. Since $\boldsymbol{\Lambda}(\boldsymbol{B})$ is the NTRU lattice for $\boldsymbol{h}$ and q and the distribution is shifted by $(\boldsymbol{c}, \boldsymbol{0})$ the output is distributed according a Gaussian $\mathcal{D}_{\mathcal{R},s}$ conditioned on $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} = \boldsymbol{c} \mod q$. The output distribution in Game G_6 is a Gaussian $\mathcal{D}_{\mathcal{R},s}$ as well where the condition $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} = \boldsymbol{c} \mod q$ is fulfilled by construction (Line 36). ■

Game G_7*.* This game is identical to the previous one except that the game aborts if the $\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star$ corresponding to the adversary's output equals the preimage with which the challenge random oracle output $\boldsymbol{c}^\star$ was computed and $(m^\star, \sigma^\star)$ is not in $\mathcal{Q}$.

Claim 7: $\left|\Pr\left[\mathsf{G}_6^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_7^{\mathsf{A}} \Rightarrow 1\right]\right| \leq 2^{-n}$.

Proof. We distinguish two cases: first, the adversary queried the signing oracle corresponding to $\boldsymbol{c}^\star$, i.e. a signing query to $m^\star$ output $(r^\star, \cdot)$ and $\mathsf{H}(\boldsymbol{h}, r^\star, m^\star) = \boldsymbol{c}^\star$. In this case, the abort event cannot trigger because if the preimages are the same, the signature must be the same as well and therefore $(m^\star, \sigma^\star) \in \mathcal{Q}$. Second, if the signing oracle was not queried corresponding to $\boldsymbol{c}^\star$, the adversary does not have any information about the preimages of $\boldsymbol{c}^\star$ except that they are Gaussian distributed. Hence, we obtain the claimed upper bound by using the min-entropy of a sample from a Gaussian distribution conditioned on $\boldsymbol{c}^\star$ from Lemma 3. ■

Reduction to G_7*.* We now can reduce $\mathcal{R}$-**SIS** to Game G_7.

Claim 8: There exists an adversary B against $\mathcal{R}$-**SIS** such that

$$\Pr[\mathsf{G}_7^{\mathsf{A}} \Rightarrow 1] \leq \mathrm{Adv}^{\mathcal{R}\text{-}\mathbf{SIS}}_{1,q,\alpha,2\beta,\mathsf{B}}.$$

Proof. Adversary B is formally constructed in Figure 6. Due to the changes in the previous games, adversary B can perfectly simulate the game for adversary A against G_7 without having the secret key for $\boldsymbol{h}$. Let us assume, that A wins game G_7, i.e. the forgery verifies and the tuple $(m^\star, \sigma^\star)$ was not queried before. That means that the output of adversary B fulfills the following conditions. First, it holds

$$(\boldsymbol{s}_1^\star - \boldsymbol{s}_1') + (\boldsymbol{s}_2^\star - \boldsymbol{s}_2') * \boldsymbol{h} = \boldsymbol{s}_1^\star + \boldsymbol{s}_2^\star * \boldsymbol{h} - (\boldsymbol{s}_1' + \boldsymbol{s}_2' * \boldsymbol{h}) = \boldsymbol{c}^\star - \boldsymbol{c}^\star = \boldsymbol{0}$$

due to the computation of $s_1^\star$ in Line 08 and the structure of elements in $\mathcal{H}$. Second, it cannot equal $\boldsymbol{0}$ due to the changes in G_7. Third, the norm bound of the output can be upperbounded by $B = 2\beta$:

$$\|(\boldsymbol{s}_1^\star - \boldsymbol{s}_1', \boldsymbol{s}_2^\star - \boldsymbol{s}_2')\|_2 \leq \|(\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star)\|_2 + \|(\boldsymbol{s}_1', \boldsymbol{s}_2')\|_2 \leq 2\beta,$$

where the last inequality follows by the winning condition of adversary A and the norm condition of preimages that do not trigger an abort in Line 07. ■

Collecting the bounds yields the stated claim. ■

```
$\mathsf{B}(\boldsymbol{h})$
$\mathcal{H}, \mathcal{Q} \leftarrow \emptyset$
$(m^\star, \sigma^\star) \xleftarrow{\$} \mathsf{A}^{\mathtt{Sgn}(\cdot),\mathsf{H}(\cdot,\cdot,\cdot)}(\boldsymbol{h})$
parse $\sigma^\star \to (r^\star, \boldsymbol{s}_2^\star)$
$\boldsymbol{c}^\star := \mathsf{H}(\boldsymbol{h}, r^\star, m^\star)$
find $(\boldsymbol{s}_1', \boldsymbol{s}_2') : (\boldsymbol{c}^\star, \boldsymbol{h}, r^\star, m^\star, \boldsymbol{s}_1', \boldsymbol{s}_2') \in \mathcal{H}$
if $\|(\boldsymbol{s}_1', \boldsymbol{s}_2')\|_2 > \beta$
    abort
$\boldsymbol{s}_1^\star := \boldsymbol{c}^\star - \boldsymbol{s}_2^\star * \boldsymbol{h} \mod q$
if $(\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star) = (\boldsymbol{s}_1', \boldsymbol{s}_2') \wedge (m^\star, \sigma^\star) \notin \mathcal{Q}$
    abort
return $(\boldsymbol{s}_1^\star - \boldsymbol{s}_1', \boldsymbol{s}_2^\star - \boldsymbol{s}_2')$

Oracle $\mathtt{Sgn}(m)$
return $\mathsf{G}_7.\mathtt{Sgn}(m)$

Oracle $\mathsf{H}(pk, r, m)$
return $\mathsf{G}_7.\mathsf{H}(pk, r, m)$
```

Figure 6. Adversary B against $\mathcal{R}$-**SIS** for the proof of Theorem 1.

B Proof of Theorem 2

Proof. Consider the sequence of games depicted in Figure 7. We are proving a bound for strong unforgeability (reducing to collision resistance or $\mathcal{R}$-**SIS**) and one for plain unforgeability (reducing to one-wayness or $\mathcal{R}$-**ISIS**) using a very similar sequence of games. This is why we depict both proofs in the same Figure with the only difference being the winning condition.

Game G_0. This is the (strong) unforgeability game for CoreFalcon$^+$ so by definition we have for winning condition CR

$$\Pr[\mathsf{G}_0^{\mathsf{A}} \Rightarrow 1] = \mathrm{Adv}^{Q_s\text{-}\mathbf{SUF\text{-}CMA}}_{\textsc{CoreFalcon}^+,\mathsf{A}},$$

and for winning condition OW

$$\Pr[\mathsf{G}_0^{\mathsf{A}} \Rightarrow 1] = \mathrm{Adv}^{Q_s\text{-}\mathbf{UF\text{-}CMA}}_{\textsc{CoreFalcon}^+,\mathsf{A}},$$

Game G_1. This game is identical to the previous one, except it aborts if the number of repetitions in the signing oracle exceeds threshold C_s.

Claim 9: $\left|\Pr\left[\mathsf{G}_0^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_1^{\mathsf{A}} \Rightarrow 1\right]\right| \leq Q_s \cdot p^{C_s}_{\mathsf{PreSmp},\beta}$.

Proof. For one signing query the probability of an abort can be upper bounded by $p^{C_s}_{\mathsf{PreSmp},\beta}$. Since the counter is reset and the abort condition in each signing query is independent, we obtain the claim by applying a union bound over all signing queries. ■

Game G_2. This game is identical to the previous one except that the RO queries are counted by variable ℓ and a uniformly random RO query $\ell^\star$ is chosen from the set $[Q_{\mathsf{H}} + 1]$. Since this is only a conceptual change, it holds

$$\Pr[\mathsf{G}_1^{\mathsf{A}} \Rightarrow 1] = \Pr[\mathsf{G}_2^{\mathsf{A}} \Rightarrow 1].$$

Game G_3. This game is identical to the previous one except that it aborts in the signing oracle $\mathtt{Sgn}$ if there already exists a query to the random oracle on the same public key, salt r, and message m. To ease the depiction in further hybrids, we define a new RO H' maintaining the same set $\mathcal{H}$ as H but aborting in case of a query on the same input as a previous query. Oracle H' is then called within the signing oracle instead of H.

Claim 10: $\left|\Pr\left[\mathsf{G}_2^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1\right]\right| \leq \frac{Q_s(C_s Q_s + Q_{\mathsf{H}})}{2^k}$.

```
Games $\mathsf{G}_0 - \mathsf{G}_9$
$\mathcal{H}, \mathcal{Q} \leftarrow \emptyset$
$\ell := 0$
$\ell^\star \xleftarrow{\$} [Q_\mathsf{H} + 1]$    // $\mathsf{G}_2 - \mathsf{G}_9$
$(\boldsymbol{B}, \boldsymbol{h}) \xleftarrow{\$} \mathsf{Gen}$
$(m^\star, \sigma^\star) \xleftarrow{\$} \mathsf{A}^{\mathtt{Sgn}(\cdot), \mathsf{H}(\cdot,\cdot,\cdot)}(\boldsymbol{h})$
parse $\sigma^\star \rightarrow (r^\star, \boldsymbol{s}_2^\star)$
$\boldsymbol{c}^\star := \mathsf{H}(\boldsymbol{h}, r^\star, m^\star)$    // $\mathsf{G}_2 - \mathsf{G}_9$
if $(\boldsymbol{c}^\star, \boldsymbol{h}, r^\star, m^\star, \ell^\star, \cdot, \cdot) \notin \mathcal{H}$    // $\mathsf{G}_4 - \mathsf{G}_9$
    abort    // $\mathsf{G}_4 - \mathsf{G}_9$
$(\boldsymbol{s}_1', \boldsymbol{s}_2') \xleftarrow{\$} \mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}^\star, \boldsymbol{0}))$    // $\mathsf{G}_5 - \mathsf{G}_9$
$(\boldsymbol{s}_1', \boldsymbol{s}_2') \xleftarrow{\$} \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}), s, (\boldsymbol{c}^\star, \boldsymbol{0})}$    // $\mathsf{G}_7 - \mathsf{G}_9$
find $(\boldsymbol{s}_1', \boldsymbol{s}_2') : (\boldsymbol{c}^\star, \boldsymbol{h}, r^\star, m^\star, \cdot, \boldsymbol{s}_1', \boldsymbol{s}_2') \in \mathcal{H}$    // $\mathsf{G}_8 - \mathsf{G}_9$
if $\|(\boldsymbol{s}_1', \boldsymbol{s}_2')\|_2 > \beta$    // $\mathsf{G}_5 - \mathsf{G}_9$
    abort    // $\mathsf{G}_5 - \mathsf{G}_9$
$\boldsymbol{s}_1^\star := \boldsymbol{c}^\star - \boldsymbol{s}_2^\star * \boldsymbol{h} \mod q$    // $\mathsf{G}_9$
if $(\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star) = (\boldsymbol{s}_1', \boldsymbol{s}_2') \wedge (m^\star, \sigma^\star) \notin \mathcal{Q}$    // $\mathsf{G}_9$
    abort    // $\mathsf{G}_9$
return $[\![\mathsf{Ver}(\boldsymbol{h}, m^\star, \sigma^\star) = 1 \wedge (m^\star, \sigma^\star) \notin \mathcal{Q}]\!]$    // CR
return $[\![\mathsf{Ver}(\boldsymbol{h}, m^\star, \sigma^\star) = 1 \wedge (m^\star, \cdot) \notin \mathcal{Q}]\!]$    // OW

Oracle $\mathtt{Sgn}(m)$
$cnt := 0$    // $\mathsf{G}_1 - \mathsf{G}_9$
repeat
    $cnt \leftarrow cnt + 1$    // $\mathsf{G}_1 - \mathsf{G}_9$
    if $cnt > C_s$    // $\mathsf{G}_1 - \mathsf{G}_9$
        abort    // $\mathsf{G}_1 - \mathsf{G}_9$
    $r \xleftarrow{\$} \{0,1\}^k$
    $\boldsymbol{c} := \mathsf{H}(\boldsymbol{h}, r, m)$    // $\mathsf{G}_0 - \mathsf{G}_2$
    $(\boldsymbol{c}, \boldsymbol{s}_1, \boldsymbol{s}_2) := \mathsf{H}'(\boldsymbol{h}, r, m)$    // $\mathsf{G}_3 - \mathsf{G}_9$
    $(\boldsymbol{s}_1, \boldsymbol{s}_2) \xleftarrow{\$} \mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}, \boldsymbol{0}))$    // $\mathsf{G}_0 - \mathsf{G}_6$
    $(\boldsymbol{s}_1, \boldsymbol{s}_2) \xleftarrow{\$} \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}), s, (\boldsymbol{c}, \boldsymbol{0})}$    // $\mathsf{G}_7$
until $\|(\boldsymbol{s}_1, \boldsymbol{s}_2)\|_2 \leq \beta$
$\sigma := (r, \boldsymbol{s}_2)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(m, \sigma)\}$
return $\sigma$
```

```
Oracle $\mathsf{H}(pk, r, m)$
if $\exists\, \boldsymbol{c} : (\boldsymbol{c}, pk, r, m, \cdot, \cdot, \cdot) \in \mathcal{H}$
    return $\boldsymbol{c}$
$\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$
$\mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{c}, pk, r, m, \perp, \perp, \perp)\}$    // $\mathsf{G}_0 - \mathsf{G}_1$
$\ell := \ell + 1$    // $\mathsf{G}_2 - \mathsf{G}_9$
if $\ell = \ell^\star$    // $\mathsf{G}_6 - \mathsf{G}_9$
    $\boldsymbol{s}_1, \boldsymbol{s}_2 \leftarrow \mathcal{D}_{\mathcal{R}, s}$    // $\mathsf{G}_6 - \mathsf{G}_9$
    $\boldsymbol{c} \leftarrow \boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} \mod q$    // $\mathsf{G}_6 - \mathsf{G}_9$
    $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{c}, pk, r, m, \ell, \boldsymbol{s}_1, \boldsymbol{s}_2)\}$    // $\mathsf{G}_6 - \mathsf{G}_9$
else    // $\mathsf{G}_6 - \mathsf{G}_9$
    $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{c}, pk, r, m, \ell, \perp, \perp)\}$    // $\mathsf{G}_2 - \mathsf{G}_9$
return $\boldsymbol{c}$

Oracle $\mathsf{H}'(\boldsymbol{h}, r, m)$    // $\mathsf{G}_3 - \mathsf{G}_9$
if $\exists\, \boldsymbol{c} : (\boldsymbol{c}, \boldsymbol{h}, r, m, \cdot, \cdot, \cdot) \in \mathcal{H}$
    abort
$\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$
$(\boldsymbol{s}_1, \boldsymbol{s}_2) := (\perp, \perp)$
$\boldsymbol{s}_1, \boldsymbol{s}_2 \leftarrow \mathcal{D}_{\mathcal{R}, s}$    // $\mathsf{G}_6 - \mathsf{G}_9$
$\boldsymbol{c} \leftarrow \boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} \mod q$    // $\mathsf{G}_6 - \mathsf{G}_9$
$\mathcal{H} := \mathcal{H} \cup \{(\boldsymbol{c}, \boldsymbol{h}, r, m, \ell^\star, \boldsymbol{s}_1, \boldsymbol{s}_2)\}$
return $(\boldsymbol{c}, \boldsymbol{s}_1, \boldsymbol{s}_s)$
```

Figure 7. Games for the proof of Theorem 2.

Proof. The salt r is chosen uniformly at random from $\{0,1\}^k$ for each signing query. The total number of elements in $\mathcal{H}$ is at most $C_s Q_s + Q_\mathsf{H}$, as at most one element is added per query to H/H'; and there are $C_s Q_s$ implicit queries via `Sgn` and Q_H direct ones. Thus, the probability that the freshly chosen salt was part of a previous query is at most $\frac{C_s Q_s + Q_\mathsf{H}}{2^k}$. For Q_s queries to the signing oracle `Sgn`, we obtain the claimed bound. ■

Game G_4. This game is identical to the previous one except that it aborts if the hash query corresponding to the output forgery does not equal the $\ell^\star$-th query guessed in the beginning or originated from the signing oracle.

Claim 11: $\Pr[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1] \leq (Q_{\mathsf{H}} + 1) \cdot \Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1]$.

Proof. The probability that the abort does not trigger is at least $\frac{1}{Q_{\mathsf{H}}+1}$ and independent of winning probability in Game G_3. Hence, the mutiplicative difference yields the claim. ■

The following change is only necessary to reduce to SIS in the end, i.e. it is only needed to prove strong unforgeability. Since it does not significantly[9] change the remaining structure of the proof, we will ignore this change in the plain unforgeability bound.

Game G_5. This game is identical to the previous one except that the game computes a preimage of $\boldsymbol{c}^\star$ with respect to $\boldsymbol{h}$ using the preimage sampler PreSmp with trapdoor $\boldsymbol{B}$. If the norm of the resulting preimage $(\boldsymbol{s}_1', \boldsymbol{s}_2')$ is larger than β, the game aborts.

Claim 12: $\Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1] = (1 - p_{\mathsf{PreSmp},\beta})^{-1} \cdot \Pr[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1]$.

Proof. The probability that the abort event does not occur is $1 - p_{\mathsf{PreSmp},\beta}$. Since the preimage is computed after adversary A output their forgery and the winning condition is not affected by $(\boldsymbol{s}_1', \boldsymbol{s}_2')$, the abort event is independent of A's winning probability. Hence we can apply the multiplicative difference lemma to obtain the statement. ■

Game G_6. This game is the same as the previous one, except that random oracle H' no longer returns a uniformly random element $\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$. Instead, it computes $\boldsymbol{c}$ as follows: It samples two elements $\boldsymbol{s}_1, \boldsymbol{s}_2$ from a Gaussian distribution $\mathcal{D}_{\mathcal{R},s}$ with standard deviation s over ring $\mathcal{R}$. Then, $\boldsymbol{c}$ is computed as $\boldsymbol{c} = \boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} \mod q$, where $\boldsymbol{h}$ is the public key. For future use, $\boldsymbol{s}_1, \boldsymbol{s}_2$, along with the input and output to the random oracle, are stored in $\mathcal{H}$. This procedure is also applied in the $\ell^\star$-th query to H.

Claim 13: For distribution $\mathcal{P} := \mathcal{U}(\mathcal{R}_q)$, distribution $\mathcal{Q}_{\boldsymbol{h}}$ the distribution of $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} \mod q$ where $\boldsymbol{s}_1, \boldsymbol{s}_2 \xleftarrow{\$} \mathcal{D}_{\mathcal{R},s}$, and $a \in (1, \infty)$ it holds

$$\Pr[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1] \leq \left(R_a(\mathcal{P} \mid\mid \mathcal{Q}_{\boldsymbol{h}})^{C_s Q_s + 1} \cdot \Pr[\mathsf{G}_6^{\mathsf{A}} \Rightarrow 1]\right)^{\frac{a-1}{a}} .$$

Proof. In Game G_5, the output distribution of the RO is the uniform distribution over $\mathcal{R}_q$, named $\mathcal{P}$. In Game G_6, we use the distribution of $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} \mod q$ where $\boldsymbol{s}_1, \boldsymbol{s}_2 \xleftarrow{\$} \mathcal{D}_{\mathcal{R},s}$, named $\mathcal{Q}_{\boldsymbol{h}}$ for simplicity. These underlying distributions are queried at most $Q := C_s Q_s + 1$ times due to $C_s Q_s$ queries to H' and one query to H. By the probability preservation, data processing inequality, and multiplicativity of the Rényi divergence (Lemma 7) we obtain

$$\Pr[\mathsf{G}_6^{\mathsf{A}} \Rightarrow 1] \geq \frac{\Pr[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1]^{\frac{a}{a-1}}}{R_a(\mathsf{G}_5^{\mathsf{A}} \mid\mid \mathsf{G}_6^{\mathsf{A}})} \geq \frac{\Pr[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1]^{\frac{a}{a-1}}}{R_a(\mathcal{P}^Q \mid\mid \mathcal{Q}_{\boldsymbol{h}}^Q)} = \frac{\Pr[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1]^{\frac{a}{a-1}}}{R_a(\mathcal{P} \mid\mid \mathcal{Q}_{\boldsymbol{h}})^Q} .$$

■

Game G_7. This game is identical to the previous one except that the output of the preimage sampler $\mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}, \boldsymbol{0}))$ is replaced by a Gaussian over the same lattice, standard deviation and center, namely $\mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}),s,(\boldsymbol{c},\boldsymbol{0})}$.

Claim 14: For distributions $\mathsf{PreSmp} := \mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}, \boldsymbol{0}))$, $\mathcal{D} := \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}),s,(\boldsymbol{c},\boldsymbol{0})}$, and $a \in (1, \infty)$ it holds

$$\Pr[\mathsf{G}_6^{\mathsf{A}} \Rightarrow 1] \leq \left(R_a(\mathsf{PreSmp} \mid\mid \mathcal{D})^{C_s Q_s + 1} \cdot \Pr[\mathsf{G}_7^{\mathsf{A}} \Rightarrow 1]\right)^{\frac{a-1}{a}} .$$

Proof. The claims follows by analogous arguments as for the changes to Game G_6. Note that we have to replace $C_s \cdot Q_s$ queries in the signing oracle and one additional query after the output of the forgery. ■

[9] There will be one additional query to the underlying distributions in G_7.

Game G_8. This game is identical to the previous one except that the $\boldsymbol{s}_1, \boldsymbol{s}_2$ are not sampled from a Gaussian distribution centred at $(\boldsymbol{c}, \mathbf{0})$ as before. Instead, the preimage of $\boldsymbol{c}$ that was sampled in H' is used.

Claim 15: $\Pr[\mathsf{G}_7^{\mathsf{A}} \Rightarrow 1] = \Pr[\mathsf{G}_8^{\mathsf{A}} \Rightarrow 1]$.

Proof. We need to show that the distributions of the games are the same. The RO output $\boldsymbol{c}$ is the same in both games. In G_7, the singing oracle outputs $(\boldsymbol{s}_1, \boldsymbol{s}_s) \sim \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}),s,(\boldsymbol{c},\mathbf{0})}$. Since $\boldsymbol{\Lambda}(\boldsymbol{B})$ is the NTRU lattice for $\boldsymbol{h}$ and q and the distribution is shifted by $(\boldsymbol{c}, \mathbf{0})$ the output is distributed according a Gaussian $\mathcal{D}_{\mathcal{R},s}$ conditioned on $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} = \boldsymbol{c} \mod q$. The output distribution in Game G_8 is a Gaussian $\mathcal{D}_{\mathcal{R},s}$ as well where the condition $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} = \boldsymbol{c} \mod q$ is fulfilled by construction (Line 41 and Line 51). ■

Game G_9. This game is identical to the previous one except that the game aborts if the $\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star$ corresponding to the adversary's output equals the preimage with which the challenge random oracle output $\boldsymbol{c}^\star$ was computed and the message/signature pair is not in $\mathcal{Q}$.

Claim 16: $\left|\Pr\left[\mathsf{G}_8^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_9^{\mathsf{A}} \Rightarrow 1\right]\right| \leq 2^{-n}$.

Proof. We distinguish two cases: first, the adversary queried the signing oracle corresponding to $\boldsymbol{c}^\star$, i.e. a signing query to $m^\star$ output $(r^\star, \cdot)$ and $\mathsf{H}(\boldsymbol{h}, r^\star, m^\star) = \boldsymbol{c}^\star$. In this case, the abort event cannot trigger because if the preimages are the same, the signature must be the same as well and therefore $(m^\star, \sigma^\star) \in \mathcal{Q}$. Second, if the signing oracle was not queried corresponding to $\boldsymbol{c}^\star$, the adversary does not have any information about the preimages of $\boldsymbol{c}^\star$ except that they are Gaussian distributed. Hence, we obtain the claimed upper bound by using the min-entropy of a sample from a Gaussian distribution conditioned on $\boldsymbol{c}^\star$ from Lemma 3. ■

Reduction from $\mathcal{R}$-***SIS*** *to* G_9. We now can reduce $\mathcal{R}$-**SIS** to Game G_9.

Claim 17: There exists an adversary B against $\mathcal{R}$-**SIS** such that

$$\Pr[\mathsf{G}_9^{\mathsf{A}} \Rightarrow 1] \leq \mathrm{Adv}^{\mathcal{R}\textbf{-SIS}}_{1,q,\alpha,2\beta,\mathsf{B}}.$$

Proof. Adversary B is formally constructed in Figure 8. Due to the changes in the previous games, adversary B can perfectly simulate the game for adversary A against G_9 without having the secret key for $\boldsymbol{h}$. Let us assume, that A wins the strong unforgeability game G_9, i.e. the forgery verifies and the tuple $(m^\star, \sigma^\star)$ was not queried before. That means that the output of adversary B fulfills the following conditions. First, it holds

$$(\boldsymbol{s}_1^\star - \boldsymbol{s}_1') + (\boldsymbol{s}_2^\star - \boldsymbol{s}_2') * \boldsymbol{h} = \boldsymbol{s}_1^\star + \boldsymbol{s}_2^\star * \boldsymbol{h} - (\boldsymbol{s}_1' + \boldsymbol{s}_2' * \boldsymbol{h}) = \boldsymbol{c}^\star - \boldsymbol{c}^\star = \mathbf{0}$$

due to the computation of $s_1^\star$ in Line 12 and the structure of elements in $\mathcal{H}$. Second, it cannot equal $\mathbf{0}$ due to the changes in G_9. Third, the norm bound of the output can be upper bounded by $B = 2\beta$:

$$\|(\boldsymbol{s}_1^\star - \boldsymbol{s}_1', \boldsymbol{s}_2^\star - \boldsymbol{s}_2')\|_2 \leq \|(\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star)\|_2 + \|(\boldsymbol{s}_1', \boldsymbol{s}_2')\|_2 \leq 2\beta,$$

where the last inequality follows by the winning condition of adversary A and the norm condition of preimages passing the check in Line 11.

■

We can also reduce $\mathcal{R}$-**ISIS** to Game G_8, i.e. G_9 is only needed for the reduction from $\mathcal{R}$-**SIS**. The same holds for G_5 as we mentioned earlier.

Reduction from $\mathcal{R}$-***ISIS*** *to* G_8. In the changes to Game G_6, we did not only program RO H' but also one query to H. For the reduction to $\mathcal{R}$-**ISIS** this is not needed nor helpful why we assume that this part was not changed for the current reduction to G_8. We will also take this into account in the final bound, i.e. in the changes for G_6, $Q \coloneqq C_s Q_s$ queries are sufficient. The same holds for ignoring the bound from G_5 and the implication of an additional query in G_7.

Claim 18: There exists an adversary C against $\mathcal{R}$-**ISIS** such that

$$\Pr[\mathsf{G}_8^{\mathsf{A}} \Rightarrow 1] \leq \mathrm{Adv}^{\mathcal{R}\textbf{-ISIS}}_{1,q,\alpha,\beta,\mathsf{C}}.$$

```
$\mathsf{B}(\boldsymbol{h})$
01 $\mathcal{H}, \mathcal{Q} \leftarrow \emptyset$
02 $\ell := 0$
03 $\ell^\star \xleftarrow{\$} [Q_\mathsf{H} + 1]$
04 $(m^\star, \sigma^\star) \xleftarrow{\$} \mathsf{A}^{\mathtt{Sgn}(\cdot),\mathsf{H}(\cdot,\cdot,\cdot)}(\boldsymbol{h})$
05 parse $\sigma^\star \to (r^\star, \boldsymbol{s}_2^\star)$
06 $\boldsymbol{c}^\star := \mathsf{H}(\boldsymbol{h}, r^\star, m^\star)$
07 if $(\boldsymbol{c}^\star, \boldsymbol{h}, r^\star, m^\star, \ell^\star, \cdot, \cdot) \notin \mathcal{H}$
08     abort
09 find $(\boldsymbol{s}_1', \boldsymbol{s}_2') : (\boldsymbol{c}^\star, \boldsymbol{h}, r^\star, m^\star, \cdot, \boldsymbol{s}_1', \boldsymbol{s}_2') \in \mathcal{H}$
10 if $\|(\boldsymbol{s}_1', \boldsymbol{s}_2')\|_2 > \beta$
11     abort
12 $\boldsymbol{s}_1^\star := \boldsymbol{c}^\star - \boldsymbol{s}_2^\star * \boldsymbol{h} \mod q$
13 if $(\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star) = (\boldsymbol{s}_1', \boldsymbol{s}_2') \wedge (m^\star, \sigma^\star) \notin \mathcal{Q}$
14     abort
15 return $(\boldsymbol{s}_1^\star - \boldsymbol{s}_1', \boldsymbol{s}_2^\star - \boldsymbol{s}_2')$

Oracle $\mathsf{H}(pk, r, m)$
16 if $\exists\, \boldsymbol{c} : (\boldsymbol{c}, pk, r, m, \cdot, \cdot, \cdot) \in \mathcal{H}$
17     return $\boldsymbol{c}$
18 $\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$
19 $\ell := \ell + 1$
20 if $\ell = \ell^\star$
21     $\boldsymbol{s}_1, \boldsymbol{s}_2 \leftarrow \mathcal{D}_{\mathcal{R},s}$
22     $\boldsymbol{c} \leftarrow \boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} \mod q$
23     $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{c}, pk, r, m, \ell, \boldsymbol{s}_1, \boldsymbol{s}_2)\}$
24 else
25     $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{c}, pk, r, m, \ell, \bot, \bot)\}$
26 return $\boldsymbol{c}$

Oracle $\mathsf{H}'(\boldsymbol{h}, r, m)$
27 return $\mathsf{G}_9.\mathsf{H}'(\boldsymbol{h}, r, m)$

Oracle $\mathtt{Sgn}(m)$
28 return $\mathsf{G}_9.\mathtt{Sgn}(m)$
```

Figure 8. Adversary B against $\mathcal{R}$-**SIS** for the proof of Theorem 2.

Proof. Adversary C is formally constructed in Figure 9. Due to the changes in the previous games, adversary B can perfectly simulate the game for adversary A against G_8 without having the secret key for $\boldsymbol{h}$. Further they embed their own target in the $\ell^\star$-th query to H. Let us assume, that A wins the unforgeability game G_8, i.e. the forgery verifies and $(m^\star, \cdot)$ was not queried before. This implies that it holds $\hat{\boldsymbol{c}} = \boldsymbol{c}^\star$ because if A is winning the game, it cannot abort in Line 08 and thus the challenge RO output $\boldsymbol{c}^\star$ corresponds to the guessed query (that is exactly $\hat{\boldsymbol{c}}$) or to a signing query. If it corresponds to a signing query, there is no way that adversary A can win the game due to the freshness condition $(m^\star, \cdot) \notin \mathcal{Q}$. Hence, Line 09 ensures the first winning condition of C: $\boldsymbol{s}_1^\star + \boldsymbol{s}_2^\star * \boldsymbol{h} = \hat{\boldsymbol{c}}$. Further, the norm bound from A directly translates to the second winning condition, i.e. $\|(\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star)\|_2 \leq \beta$. ■

```
$\mathsf{C}(\boldsymbol{h}, \hat{\boldsymbol{c}})$
01 $\mathcal{H}, \mathcal{Q} \leftarrow \emptyset$
02 $\ell := 0$
03 $\ell^\star \xleftarrow{\$} [Q_\mathsf{H} + 1]$
04 $(m^\star, \sigma^\star) \xleftarrow{\$} \mathsf{A}^{\mathtt{Sgn}(\cdot),\mathsf{H}(\cdot,\cdot,\cdot)}(\boldsymbol{h})$
05 parse $\sigma^\star \to (r^\star, \boldsymbol{s}_2^\star)$
06 $\boldsymbol{c}^\star := \mathsf{H}(\boldsymbol{h}, r^\star, m^\star)$
07 if $(\boldsymbol{c}^\star, \boldsymbol{h}, r^\star, m^\star, \ell^\star, \cdot, \cdot) \notin \mathcal{H}$
08     abort
09 $\boldsymbol{s}_1^\star := \boldsymbol{c}^\star - \boldsymbol{s}_2^\star * \boldsymbol{h} \mod q$
10 return $(\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star)$

Oracle $\mathtt{Sgn}(m)$
11 return $\mathsf{G}_8.\mathtt{Sgn}(m)$

Oracle $\mathsf{H}(pk, r, m)$
12 if $\exists\, \boldsymbol{c} : (\boldsymbol{c}, pk, r, m, \cdot, \cdot, \cdot) \in \mathcal{H}$
13     return $\boldsymbol{c}$
14 $\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$
15 $\ell := \ell + 1$
16 if $\ell = \ell^\star$
17     $\boldsymbol{c} := \hat{\boldsymbol{c}}$        // embed challenge target
18 $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{c}, pk, r, m, \ell, \bot, \bot)\}$
19 return $\boldsymbol{c}$

Oracle $\mathsf{H}'(\boldsymbol{h}, r, m)$
20 return $\mathsf{G}_8.\mathsf{H}'(\boldsymbol{h}, r, m)$
```

Figure 9. Adversary C against $\mathcal{R}$-**ISIS** for the proof of Theorem 2.

Collecting the bounds yields the stated claim. ■

C Appendix for Section 5

```
sage: SIS.estimate.rough(SIS.Parameters(n=512,q=12289,length_bound=5833.93,norm=2,m=2*512))
lattice :: rop: ≈2^121.2, red: ≈2^121.2, δ: 1.003882, β: 415, d: 1024, tag: euclidean
sage: SIS.estimate.rough(SIS.Parameters(n=512,q=12289,length_bound=2*5833.93,norm=2,m=2*512))
lattice :: rop: ≈2^95.8, red: ≈2^95.8, δ: 1.004561, β: 328, d: 1024, tag: euclidean
sage: SIS.estimate.rough(SIS.Parameters(n=1024,q=12289,length_bound=8382.44,norm=2,m=2*1024))
lattice :: rop: ≈2^279.2, red: ≈2^279.2, δ: 1.002114, β: 956, d: 2048, tag: euclidean
```

Figure 10. SIS hardness estimates for ring dimension $n = 512$, $n = 1024$ and length bound β, 2β.

C.1 Proof of Lemma 14

Proof. We begin with Falcon$^+$-512 and $\lambda = 128$. By Lemma 12 and Lemma 9, we obtain

$$\begin{aligned} p_{\mathsf{PreSmp},\beta} &= \max_{\boldsymbol{c}\in\mathcal{R}_q} \Pr_{(\boldsymbol{s}_1,\boldsymbol{s}_2)\xleftarrow{\$}\mathsf{PreSmp}(\boldsymbol{B},s,(\boldsymbol{c},\boldsymbol{0}))} [\|(\boldsymbol{s}_1,\boldsymbol{s}_2)\|_2 > \beta] \\ &\le \Pr_{(\boldsymbol{s}_1,\boldsymbol{s}_2)\leftarrow\mathcal{D}_{\mathbb{Z}^{2n},s}} [\|(\boldsymbol{s}_1,\boldsymbol{s}_2)\|_2 > \beta] \cdot (1+2\epsilon). \end{aligned}$$

Using Lemma 2 and $\epsilon = (2^{64}\cdot 128)^{-1/2}, \beta = 1.1\cdot s\sqrt{2n}$, and $n = 512$ for Falcon$^+$-512 yields

$$p_{\mathsf{PreSmp},\beta} \le 1.1^{1024} e^{256\cdot(1-1.1^2)} \cdot (1 + 2(2^{64}\cdot 128)^{-1/2}) \quad (\approx 2^{-14.31})$$

The smallest C_s such that $p_{\mathsf{PreSmp},\beta}^{C_s} \le 2^{-128}$ is 9.

The second claim follows analogously with $\epsilon = (2^{64}\cdot 256)^{-1/2}$ and $n = 1024$ for Falcon$^+$-1024. ■

C.2 Proof of Corollary 3

Proof. By Lemma 13 we need to solve

$$\min_{a_p>1} C_s Q_s \log\left(R_{a_p}(\mathsf{PreSmp} \,||\, \mathcal{D})\right) + \lambda \cdot a_p^{-1}.$$

By Corollary 1 we can upper bound R_{a_p}

$$\min_{a_p>1} C_s Q_s \log\left(1 + 2a_p\epsilon^2\right) + \lambda \cdot a_p^{-1}.$$

Computing the derivative of the above function gives

$$C_s Q_s \frac{2\epsilon^2}{(1+2a_p\epsilon^2)\ln(2)} - \frac{\lambda}{a_p^2}.$$

Setting the result to 0 and rearranging the terms yields

$$0 = C_s Q_s 2\epsilon^2 a_p^2 - 2\cdot\lambda\epsilon^2\ln(2)a_p - \lambda\ln(2).$$

With the condition $a_p > 1$ the solution of the quadratic equation is

$$a_p = \frac{2\ln(2)\cdot\lambda\epsilon^2 + \sqrt{4\ln(2)^2\cdot\lambda^2\epsilon^4 + 8C_sQ_s\epsilon^2\lambda\ln(2)}}{4C_sQ_s\epsilon^2}.$$

Plugging in the parameters gives

$$a_p \approx 275.15.$$

and thus a bit loss of at most

$$C_s Q_s \log(1 + 2 \cdot 275.15\epsilon^2) + \frac{128}{275.15} \leq 0.87224.$$

■

D Tight Multi-User Security

Another modification of CoreFalcon$^+$ compared to the Falcon specification is hashing the public key which we use to show that CoreFalcon$^+$ achieves multi-user security without any additional tightness loss. There reason is based on two key aspects. First, in CoreFalcon$^+$, the public key is hashed which allows the reduction to program the random oracle for the user of the input public key without the need of guessing the attacked user. Second, eventually security can be reduced to (I)SIS for N samples where N is the number of users. Note that the reduction to SVP is independent of the number of samples [LM06] and thus the security does not decrease with a larger N. In Theorem 3 we prove the multi-user variant of Theorem 1 and in Theorem 4 we prove the multi-user variant of the unforgeability claim of Theorem 2. We chose the two bounds to show that all proof techniques we used are adaptable to the multi-user setting. Since the strong unforgeability bound of Theorem 2 is essentially a combination of techniques from the other two bounds, we omit the proof. In the following proofs we only prove new claims, i.e. claims that are not exactly the same as for the proof's single-user counterpart. Note that we obtain the same bounds as for the single-user counterparts and thus the analysis from Section 5 applies.

Definition 11 (Multi-user (Strong) Unforgeability). The *multi-user* notions of *(strong) existential unforgeability under chosen message attacks* are formalised via the games (N, Q_s)-$\mathbf{MU\text{-}UF\text{-}CMA}_{\mathsf{Sig}}(\mathsf{A})$ and (N, Q_s)-$\mathbf{MU\text{-}SUF\text{-}CMA}_{\mathsf{Sig}}(\mathsf{A})$, respectively. Both are depicted in Figure 11, where N denotes the number of users and Q_s the maximum number of the adversary's signing queries. We define the advantage functions of adversary A as

$$\begin{aligned} \mathrm{Adv}^{(N,Q_s)\text{-}\mathbf{MU\text{-}UF\text{-}CMA}}_{\mathsf{Sig},\mathsf{A}} &:= \Pr[(N, Q_s)\text{-}\mathbf{MU\text{-}UF\text{-}CMA}_{\mathsf{Sig}}(\mathsf{A}) \Rightarrow 1], \\ \mathrm{Adv}^{(N,Q_s)\text{-}\mathbf{MU\text{-}SUF\text{-}CMA}}_{\mathsf{Sig},\mathsf{A}} &:= \Pr[(N, Q_s)\text{-}\mathbf{MU\text{-}SUF\text{-}CMA}_{\mathsf{Sig}}(\mathsf{A}) \Rightarrow 1]. \end{aligned}$$

Theorem 3 (Multi-User Strong Unforgeability). For any adversary A, making at most Q_s signing queries and Q_{H} random oracle queries, against the **MU-SUF-CMA** security of CoreFalcon$^+$ (Figure 4) in the random oracle model, there exists adversary B against $\mathcal{R}$-**SIS** with $t_{\mathsf{A}} \approx t_{\mathsf{B}}$ such that for all constants $C_s \in \mathbb{N}^{\geq 1}$ and $a_u, a_p \in \mathbb{R}^{>1}$ it holds

$$\begin{aligned} &\mathrm{Adv}^{(\mathrm{N},Q_s)\text{-}\mathbf{MU\text{-}SUF\text{-}CMA}}_{\textsc{CoreFalcon}^+,\mathsf{A}} \leq \\ &(1 - p_{\mathsf{PreSmp},\beta})^{-1} \left(r_u^{C_s Q_s + Q_{\mathsf{H}}} \cdot \left(r_p^{C_s Q_s + 1} \cdot \left(\mathrm{Adv}^{\mathcal{R}\text{-}\mathbf{SIS}}_{N,q,\alpha,2\beta,\mathsf{B}} + 2^{-n} \right) \right)^{\frac{a_p - 1}{a_p}} \right)^{\frac{a_u - 1}{a_u}} \\ &+ Q_s \cdot p^{C_s}_{\mathsf{PreSmp},\beta} + \frac{Q_s(C_s Q_s + Q_{\mathsf{H}})}{2^k} \quad , \end{aligned}$$

where

- $p_{\mathsf{PreSmp},\beta} := \max_{\boldsymbol{c} \in \mathcal{R}_q, (\boldsymbol{B}, \cdot) \in \sup(\mathsf{Gen})} \Pr_{(\boldsymbol{s}_1, \boldsymbol{s}_2) \xleftarrow{\$} \mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}, \mathbf{0}))} [\|(\boldsymbol{s}_1, \boldsymbol{s}_2)\|_2 > \beta]$,
- $r_u = \max_{\boldsymbol{h} \neq \mathbf{0}} R_{a_u}(\mathcal{P} \mid\mid \mathcal{Q}_{\boldsymbol{h}})$ with $\mathcal{P} = \mathcal{U}(\mathcal{R}_q)$ and $\mathcal{Q}_{\boldsymbol{h}}$ the distribution of $\boldsymbol{u} + \boldsymbol{v} * \boldsymbol{h} \mod q$, where $\boldsymbol{u}, \boldsymbol{v} \sim \mathcal{D}_{\mathcal{R},s}$,
- $r_p = \max_{(\boldsymbol{B}, \cdot) \in \sup(\mathsf{Gen})} R_{a_p}(\mathsf{PreSmp}(\boldsymbol{B}, s, \cdot) \mid\mid \mathcal{D}_{\mathcal{R},s})$.

Proof. Consider the sequence of games depicted in Figure 12.

```
Games $(N, Q_s)$-MU-UF-CMA$_{\mathsf{Sig}}(\mathsf{A})/(N, Q_s)$-MU-SUF-CMA$_{\mathsf{Sig}}(\mathsf{A})$
$\mathcal{Q} \leftarrow \emptyset$
for $i \in [N]$
    $(sk_i, pk_i) \xleftarrow{\$} \mathsf{Gen}$
$(i^\star, m^\star, \sigma^\star) \xleftarrow{\$} \mathsf{A}^{\mathtt{Sgn}(\cdot,\cdot)}(pk_1, \dots, pk_N)$
return $[\![\mathsf{Ver}(pk_{i^\star}, m^\star, \sigma^\star) = 1 \wedge (i^\star, m^\star, \cdot) \notin \mathcal{Q}]\!]$        // MU-UF-CMA
return $[\![\mathsf{Ver}(pk_{i^\star}, m^\star, \sigma^\star) = 1 \wedge (i^\star, m^\star, \sigma^\star) \notin \mathcal{Q}]\!]$       // MU-SUF-CMA

Oracle $\mathtt{Sgn}(i \in [N], m)$
$\sigma \xleftarrow{\$} \mathsf{Sgn}(sk_i, m)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(i, m, \sigma)\}$
return $\sigma$
```

Figure 11. Games defining **MU-UF-CMA** and **MU-SUF-CMA** for a signature scheme $\mathsf{Sig} = (\mathsf{Gen}, \mathsf{Sgn}, \mathsf{Ver})$ and adversary A making at most Q_s queries to $\mathtt{Sgn}$.

Game G_0. This is the multi-user strong unforgeability game for CoreFalcon$^+$:

$$\Pr[\mathsf{G}_0^{\mathsf{A}} \Rightarrow 1] = \mathrm{Adv}^{(N,Q_s)\text{-}\mathbf{MU\text{-}SUF\text{-}CMA}}_{\textsc{CoreFalcon}^+,\mathsf{A}}.$$

Game G_1. This game is identical to the previous one, except it aborts if the number of repetitions in the signing oracle exceeds threshold C_s.

Claim 19: $\left|\Pr\left[\mathsf{G}_0^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_1^{\mathsf{A}} \Rightarrow 1\right]\right| \leq Q_s \cdot p_{\mathsf{PreSmp},\beta}^{C_s}$.

Game G_2. This game is identical to the previous one except that it aborts in the signing oracle $\mathtt{Sgn}$ if there already exists a query to the random oracle on $\boldsymbol{h}_i$, the same salt r and message m (Line 24).

Claim 20: $\left|\Pr\left[\mathsf{G}_1^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_2^{\mathsf{A}} \Rightarrow 1\right]\right| \leq \frac{Q_s(C_s Q_s + Q_{\mathsf{H}})}{2^k}$.

Game G_3. This game is identical to the previous one except that the game computes the RO output $\boldsymbol{c}^\star$ that corresponds to the adversary's forgery. It then computes a preimage of $\boldsymbol{c}^\star$ with respect to the challenged user $\boldsymbol{h}^\star = \boldsymbol{h}_{i^\star}$ using the preimage sampler PreSmp with trapdoor $\boldsymbol{B}_{i^\star}$. If the norm of the resulting preimage $(\boldsymbol{s}_1', \boldsymbol{s}_2')$ is larger than β, the game aborts.

Claim 21: $\Pr[\mathsf{G}_2^{\mathsf{A}} \Rightarrow 1] = (1 - p_{\mathsf{PreSmp},\beta})^{-1} \cdot \Pr[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1]$.

Game G_4. This game is the same as the previous one, except that the random oracle no longer returns a uniformly random element $\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$ if the public key pk input to the RO is honest. Instead, it computes $\boldsymbol{c}$ as follows: It samples two elements $\boldsymbol{s}_1, \boldsymbol{s}_2$ from a Gaussian distribution $\mathcal{D}_{\mathcal{R},s}$ with standard deviation s over ring $\mathcal{R}$. Then, $\boldsymbol{c}$ is computed as $\boldsymbol{c} = \boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} \mod q$, where $\boldsymbol{h} = pk$ is the public key input to the RO. For future use, $\boldsymbol{s}_1, \boldsymbol{s}_2$, along with the input and output to the random oracle, are stored in $\mathcal{H}$ (Line 40).

Claim 22: For distribution $\mathcal{P} := \mathcal{U}(\mathcal{R}_q)$, distribution $\mathcal{Q}_{\boldsymbol{h}}$ the distribution of $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} \mod q$ where $\boldsymbol{s}_1, \boldsymbol{s}_2 \xleftarrow{\$} \mathcal{D}_{\mathcal{R},s}$, and $a \in (1, \infty)$ it holds

$$\Pr[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1] \leq \left(\max_{\boldsymbol{h} \neq \boldsymbol{0}} R_a(\mathcal{P} \,||\, \mathcal{Q}_{\boldsymbol{h}})^{C_s Q_s} \cdot \Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1]\right)^{\frac{a-1}{a}}.$$

Proof. In Game G_3, the output distribution of the RO is the uniform distribution over $\mathcal{R}_q$, named $\mathcal{P}$. In Game G_4, we use the distribution of $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h}_i \mod q$ where $\boldsymbol{s}_1, \boldsymbol{s}_2 \xleftarrow{\$} \mathcal{D}_{\mathcal{R},s}$, named $\mathcal{Q}_{\boldsymbol{h}_i}$ for simplicity. These underlying distributions are queried at most $Q := C_s Q_s + Q_{\mathsf{H}}$ times due to implicit RO queries from $\mathtt{Sgn}$ and explicit ones to H. Since for each query, distribution $\mathcal{Q}$ can be parametrized by a different $\boldsymbol{h}_i$ we

```
Games $\mathsf{G}_0 - \mathsf{G}_7$
$\mathcal{H}, \mathcal{Q} \leftarrow \emptyset$
for $i \in [N]$
    $(\boldsymbol{B}_i, \boldsymbol{h}_i) \xleftarrow{\$} \mathsf{Gen}$
$(i^\star, m^\star, \sigma^\star) \xleftarrow{\$} \mathsf{A}^{\mathtt{Sgn}(\cdot,\cdot),\mathsf{H}(\cdot,\cdot,\cdot)}(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N)$
parse $\sigma^\star \to (r^\star, \boldsymbol{s}_2^\star)$
$\boldsymbol{h}^\star := \boldsymbol{h}_{i^\star}$
$\boldsymbol{c}^\star := \mathsf{H}(\boldsymbol{h}^\star, r^\star, m^\star)$                                   // $\mathsf{G}_3 - \mathsf{G}_7$
$(\boldsymbol{s}_1', \boldsymbol{s}_2') \xleftarrow{\$} \mathsf{PreSmp}(\boldsymbol{B}_{i^\star}, s, (\boldsymbol{c}^\star, \boldsymbol{0}))$               // $\mathsf{G}_3 - \mathsf{G}_7$
$(\boldsymbol{s}_1', \boldsymbol{s}_2') \xleftarrow{\$} \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}_{i^\star}),s,(\boldsymbol{c}^\star,\boldsymbol{0})}$                        // $\mathsf{G}_5 - \mathsf{G}_7$
find $(\boldsymbol{s}_1', \boldsymbol{s}_2') : (\boldsymbol{c}^\star, \boldsymbol{h}^\star, r^\star, m^\star, \boldsymbol{s}_1', \boldsymbol{s}_2') \in \mathcal{H}$          // $\mathsf{G}_6 - \mathsf{G}_7$
if $\|(\boldsymbol{s}_1', \boldsymbol{s}_2')\|_2 > \beta$                                      // $\mathsf{G}_3 - \mathsf{G}_7$
    abort                                                  // $\mathsf{G}_3 - \mathsf{G}_7$
$\boldsymbol{s}_1^\star := \boldsymbol{c}^\star - \boldsymbol{s}_2^\star * \boldsymbol{h}^\star \mod q$                              // $\mathsf{G}_7$
if $(\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star) = (\boldsymbol{s}_1', \boldsymbol{s}_2') \wedge (i^\star, m^\star, \sigma^\star) \notin \mathcal{Q}$             // $\mathsf{G}_7$
    abort                                                       // $\mathsf{G}_7$
return $[\![\mathsf{Ver}(\boldsymbol{h}^\star, m^\star, \sigma^\star) = 1 \wedge (i^\star, m^\star, \sigma^\star) \notin \mathcal{Q}]\!]$
```

```
Oracle $\mathtt{Sgn}(i \in [N], m)$
$cnt := 0$                                                 // $\mathsf{G}_1 - \mathsf{G}_7$
repeat
    $cnt \leftarrow cnt + 1$                                     // $\mathsf{G}_1 - \mathsf{G}_7$
    if $cnt > C_s$                                           // $\mathsf{G}_1 - \mathsf{G}_7$
        abort                                              // $\mathsf{G}_1 - \mathsf{G}_7$
    $r \xleftarrow{\$} \{0,1\}^k$
    if $\exists\, \boldsymbol{c} : (\boldsymbol{c}, \boldsymbol{h}_i, r, m, \cdot, \cdot) \in \mathcal{H}$                          // $\mathsf{G}_2 - \mathsf{G}_7$
        abort                                              // $\mathsf{G}_2 - \mathsf{G}_7$
    $\boldsymbol{c} := \mathsf{H}(\boldsymbol{h}_i, r, m)$
    $(\boldsymbol{s}_1, \boldsymbol{s}_2) \xleftarrow{\$} \mathsf{PreSmp}(\boldsymbol{B}_i, s, (\boldsymbol{c}, \boldsymbol{0}))$                  // $\mathsf{G}_0 - \mathsf{G}_4$
    $(\boldsymbol{s}_1, \boldsymbol{s}_2) \xleftarrow{\$} \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}_i),s,(\boldsymbol{c},\boldsymbol{0})}$                         // $\mathsf{G}_5 - \mathsf{G}_7$
    $(\boldsymbol{s}_1, \boldsymbol{s}_2) \leftarrow \{(\boldsymbol{s}_1', \boldsymbol{s}_2') \mid (\boldsymbol{c}, \boldsymbol{h}_i, r, m, \boldsymbol{s}_1', \boldsymbol{s}_2') \in \mathcal{H}\}$    // $\mathsf{G}_6 - \mathsf{G}_7$
until $\|(\boldsymbol{s}_1, \boldsymbol{s}_2)\|_2 \leq \beta$
$\sigma := (r, \boldsymbol{s}_2)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(m, \sigma)\}$
return $\sigma$
```

```
Oracle $\mathsf{H}(pk, r, m)$
if $\exists\, \boldsymbol{c} : (\boldsymbol{c}, pk, r, m, \cdot, \cdot) \in \mathcal{H}$
    return $\boldsymbol{c}$
$\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$                                            // $\mathsf{G}_0 - \mathsf{G}_2$
$\mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{c}, pk, r, m, \perp, \perp)\}$                  // $\mathsf{G}_0 - \mathsf{G}_2$
if $pk \in \{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N\}$
    $\boldsymbol{s}_1, \boldsymbol{s}_2 \xleftarrow{\$} \mathcal{D}_{\mathcal{R},s}$                                  // $\mathsf{G}_4 - \mathsf{G}_7$
    $\boldsymbol{c} \leftarrow \boldsymbol{s}_1 + \boldsymbol{s}_2 * pk \mod q$                      // $\mathsf{G}_4 - \mathsf{G}_7$
$\mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{c}, pk, r, m, \boldsymbol{s}_1, \boldsymbol{s}_2)\}$              // $\mathsf{G}_4 - \mathsf{G}_7$
return $\boldsymbol{c}$
```

Figure 12. Games for the proof of Theorem 3.

define a vector $\mathbf{H} := (\mathbf{H}_1, \ldots, \mathbf{H}_Q) \in \mathcal{R}_q^Q$ such that $\mathbf{H}_j$ equals the public key $\boldsymbol{h}_i$ used in the j-th query. Further we define $\mathcal{Q}_{\mathbf{H}}$ as the sequence of distributions $\mathcal{Q}_{\mathbf{H}_i}$. By the probability preservation, data processing inequality, and multiplicativity of the Rényi divergence (Lemma 7) we obtain

$$\begin{aligned}\Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1] &\geq \frac{\Pr[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1]^{\frac{a}{a-1}}}{R_a(\mathsf{G}_3^{\mathsf{A}} \mid\mid \mathsf{G}_4^{\mathsf{A}})} \geq \frac{\Pr[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1]^{\frac{a}{a-1}}}{R_a(\mathcal{P}^Q \mid\mid \mathcal{Q}_{\mathbf{H}})} \\ &= \frac{\Pr[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1]^{\frac{a}{a-1}}}{R_a(\mathcal{P}^Q \mid\mid \prod_{j=1}^{Q} \mathcal{Q}_{\mathbf{H}_j})} \geq \frac{\Pr[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1]^{\frac{a}{a-1}}}{\max_{\boldsymbol{h} \neq \boldsymbol{0}}\ R_a(\mathcal{P} \mid\mid \mathcal{Q}_{\boldsymbol{h}})^Q}.\end{aligned}$$

Note that this includes cases in which the adversary queries the RO on inputs for which the first element is not an honest public key or even a ring element because the Rényi divergence is 1 in that case and thus does not contribute to the loss. ∎

Game G_5*.* This game is identical to the previous one except that the output of the preimage sampler $\mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}, \mathbf{0}))$ is replaced by a Gaussian over the same lattice, standard deviation and center, namely $\mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}),s,(\boldsymbol{c},\mathbf{0})}$.

Claim 23: For $r = \max_{(\boldsymbol{B},\cdot)\in\mathsf{sup}(\mathsf{Gen})} R_a(\mathsf{PreSmp}(\boldsymbol{B}, s, \cdot) \,||\, \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}),s,\cdot})$ and $a \in (1, \infty)$ it holds

$$\Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1] \leq \left(r^{C_s Q_s + 1} \cdot \Pr[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1]\right)^{\frac{a-1}{a}} .$$

Proof. The claims follows by analogous arguments as for Game G_4. Note that we have to replace $C_s \cdot Q_s$ queries in the signing oracle and one additional query after the output of the forgery. ∎

Game G_6*.* This game is identical to the previous one except that the $\boldsymbol{s}_1, \boldsymbol{s}_2$ are not sampled from a Gaussian distribution centred at $(\boldsymbol{c}, \mathbf{0})$ as before. Instead, the preimage of $\boldsymbol{c}$ stored in the set $\mathcal{H}$ during the random oracle query is used. Such an element must exist due to the changes in G_4.

Claim 24: $\Pr[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1] = \Pr[\mathsf{G}_6^{\mathsf{A}} \Rightarrow 1]$.

Proof. We need to show that the distributions of the games are the same. The RO output $\boldsymbol{c}$ is the same in both games. In G_5, the singing oracle outputs $(\boldsymbol{s}_1, \boldsymbol{s}_s) \sim \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}_i),s,(\boldsymbol{c},\mathbf{0})}$. Since $\boldsymbol{\Lambda}(\boldsymbol{B}_i)$ is the NTRU lattice for $\boldsymbol{h}_i$ and q and the distribution is shifted by $(\boldsymbol{c}, \mathbf{0})$ the output is distributed according a Gaussian $\mathcal{D}_{\mathcal{R},s}$ conditioned on $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h}_i = \boldsymbol{c} \mod q$. The output distribution in Game G_6 is a Gaussian $\mathcal{D}_{\mathcal{R},s}$ as well where the condition $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h}_i = \boldsymbol{c} \mod q$ is fulfilled by construction (Line 39). ∎

Game G_7*.* This game is identical to the previous one except that the game aborts if the $\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star$ corresponding to the adversary's output equals the preimage with which the challenge random oracle output $\boldsymbol{c}^\star$ was computed and $(i^\star, m^\star, \sigma^\star)$ is not in $\mathcal{Q}$.

Claim 25: $\left|\Pr\left[\mathsf{G}_6^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_7^{\mathsf{A}} \Rightarrow 1\right]\right| \leq 2^{-n}$.

Reduction to G_7*.* We now can reduce $\mathcal{R}$-**SIS** to Game G_7.

Claim 26: There exists an adversary B against $\mathcal{R}$-**SIS** such that

$$\Pr[\mathsf{G}_7^{\mathsf{A}} \Rightarrow 1] \leq \mathrm{Adv}^{\mathcal{R}\text{-}\mathbf{SIS}}_{N,q,\alpha,2\beta,\mathsf{B}}.$$

Proof. Adversary B is formally constructed in Figure 13. Due to the changes in the previous games, adversary B can perfectly simulate the game for adversary A against G_7 without having the trapdoors for $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N$. Let us assume, that A wins game G_7, i.e. the forgery verifies and $(i^\star, m^\star, \sigma^\star)$ was not queried before. That means that the output of adversary B fulfills the following conditions. First, it holds

$$\begin{aligned}(\boldsymbol{s}_1^\star - \boldsymbol{s}_1') + \sum_{i=1}^{N} \boldsymbol{s}_i * \boldsymbol{h}_i &= (\boldsymbol{s}_1^\star - \boldsymbol{s}_1') + (\boldsymbol{s}_{i^\star} - \boldsymbol{s}_2') * \boldsymbol{h}_{i^\star} \\ &= \boldsymbol{s}_1^\star + \boldsymbol{s}_2^\star * \boldsymbol{h}^\star - (\boldsymbol{s}_1' + \boldsymbol{s}_2' * \boldsymbol{h}^\star) = \boldsymbol{c}^\star - \boldsymbol{c}^\star = \mathbf{0}\end{aligned}$$

due to the structure of the output computed in Line 12 and Line 13 and the structure of elements in $\mathcal{H}$. Second, it cannot equal $\mathbf{0}$ due to the changes in G_7. Third, the norm bound of the output can be upper bounded by $B = 2\beta$ using the fact that 0 elements do not contribute to the norm:

$$\begin{aligned}\|(\boldsymbol{s}_1^\star - \boldsymbol{s}_1', \boldsymbol{s}_1, \ldots, \boldsymbol{s}_N)\|_2 &= \|(\boldsymbol{s}_1^\star - \boldsymbol{s}_1', \boldsymbol{s}_2^\star - \boldsymbol{s}_2')\|_2 \\ &\leq \|(\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star)\|_2 + \|(\boldsymbol{s}_1', \boldsymbol{s}_2')\|_2 \leq 2\beta,\end{aligned}$$

where the last inequality follows by the winning condition of adversary A and the norm condition of preimages that do not trigger an abort in Line 08. ∎

```
B$(\boldsymbol{h}_1, \dots, \boldsymbol{h}_N)$
$\mathcal{H}, \mathcal{Q} \leftarrow \emptyset$
$(i^\star, m^\star, \sigma^\star) \xleftarrow{\$} \mathsf{A}^{\mathtt{Sgn}(\cdot,\cdot),\mathsf{H}(\cdot,\cdot,\cdot)}(\boldsymbol{h}_1, \dots, \boldsymbol{h}_N)$
parse $\sigma^\star \to (r^\star, \boldsymbol{s}_2^\star)$
$\boldsymbol{h}^\star := \boldsymbol{h}_{i^\star}$
$\boldsymbol{c}^\star := \mathsf{H}(\boldsymbol{h}^\star, r^\star, m^\star)$
find $(\boldsymbol{s}_1', \boldsymbol{s}_2') : (\boldsymbol{c}^\star, \boldsymbol{h}^\star, r^\star, m^\star, \boldsymbol{s}_1', \boldsymbol{s}_2') \in \mathcal{H}$
if $\|(\boldsymbol{s}_1', \boldsymbol{s}_2')\|_2 > \beta$
    abort
$\boldsymbol{s}_1^\star := \boldsymbol{c}^\star - \boldsymbol{s}_2^\star * \boldsymbol{h}^\star \mod q$
if $(\boldsymbol{s}_1^\star, \boldsymbol{s}_2^\star) = (\boldsymbol{s}_1', \boldsymbol{s}_2') \wedge (i^\star, m^\star, \sigma^\star) \notin \mathcal{Q}$
    abort
$\boldsymbol{s}_i := 0 \quad \forall\, i \in [N] \setminus \{i^\star\}$
$\boldsymbol{s}_{i^\star} := \boldsymbol{s}_2^\star - \boldsymbol{s}_2'$
return $(\boldsymbol{s}_1^\star - \boldsymbol{s}_1', \boldsymbol{s}_1, \dots, \boldsymbol{s}_N)$

Oracle Sgn$(i \in [N], m)$
return $\mathsf{G}_7$.Sgn$(i, m)$

Oracle H$(pk, r, m)$
return $\mathsf{G}_7$.H$(pk, r, m)$
```

Figure 13. Adversary B against $\mathcal{R}$-**SIS** for the proof of Theorem 3.

Collecting the bounds yields the stated claim. ■

Theorem 4 (Multi-User Unforgeability). For any adversary A, making at most Q_s signing queries and Q_H random oracle queries, against the **UF-CMA** security of CoreFalcon$^+$ (Figure 4) in the random oracle model, there exist adversaries B against $\mathcal{R}$-**ISIS** with $t_\mathsf{A} \approx t_\mathsf{B}$ such that for all $C_s \in \mathbb{N}^{\geq 1}$ and $a_u, a_p \in \mathbb{R}^{>1}$ it holds

$$\mathrm{Adv}^{(\mathrm{N},Q_s)\text{-}\mathbf{UF\text{-}CMA}}_{\textsc{CoreFalcon}^+,\mathsf{A}} \leq (Q_\mathsf{H}+1) \cdot \left(r_u^{C_s Q_s} \cdot \left(r_p^{C_s Q_s} \cdot \mathrm{Adv}^{\mathcal{R}\text{-}\mathbf{ISIS}}_{N,q,\alpha,\beta,\mathsf{B}} \right)^{\frac{a_p-1}{a_p}} \right)^{\frac{a_u-1}{a_u}} + Q_s \cdot p^{C_s}_{\mathsf{PreSmp},\beta} + \frac{Q_s(C_s Q_s + Q_\mathsf{H})}{2^k} \quad ,$$

where

- $p_{\mathsf{PreSmp},\beta} := \max_{\boldsymbol{c} \in \mathcal{R}_q, (\boldsymbol{B},\cdot) \in \sup(\mathsf{Gen})} \Pr_{(\boldsymbol{s}_1,\boldsymbol{s}_2) \xleftarrow{\$} \mathsf{PreSmp}(\boldsymbol{B},s,(\boldsymbol{c},\boldsymbol{0}))}[\|(\boldsymbol{s}_1, \boldsymbol{s}_2)\|_2 > \beta]$,
- $r_u = \max_{\boldsymbol{h} \neq \boldsymbol{0}} R_{a_u}(\mathcal{P} \mid\mid \mathcal{Q}_{\boldsymbol{h}})$ with $\mathcal{P} = \mathcal{U}(\mathcal{R}_q)$ and $\mathcal{Q}_{\boldsymbol{h}}$ the distribution of $\boldsymbol{u} + \boldsymbol{v} * \boldsymbol{h} \mod q$, where $\boldsymbol{u}, \boldsymbol{v} \sim \mathcal{D}_{\mathcal{R},s}$,
- $r_p = \max_{(\boldsymbol{B},\cdot) \in \sup(\mathsf{Gen})} R_{a_p}(\mathsf{PreSmp}(\boldsymbol{B}, s, \cdot) \mid\mid \mathcal{D}_{\mathcal{R},s})$.

Proof. Consider the sequence of games depicted in Figure 14. We are proving a bound for strong unforgeability (reducing to collision resistance or $\mathcal{R}$-**SIS**) and one for plain unforgeability (reducing to one-wayness or $\mathcal{R}$-**ISIS**) using a very similar sequence of games. This is why we depict both proofs in the same Figure with the only difference being the winning condition.

Game G_0. This is the unforgeability game for CoreFalcon$^+$ so by definition we have

$$\Pr[\mathsf{G}_0^\mathsf{A} \Rightarrow 1] = \mathrm{Adv}^{(N,Q_s)\text{-}\mathbf{MU\text{-}UF\text{-}CMA}}_{\textsc{CoreFalcon}^+,\mathsf{A}} .$$

Game G_1. This game is identical to the previous one, except it aborts if the number of repetitions in the signing oracle exceeds threshold C_s.

Claim 27: $\left|\Pr\left[\mathsf{G}_0^\mathsf{A} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_1^\mathsf{A} \Rightarrow 1\right]\right| \leq Q_s \cdot p^{C_s}_{\mathsf{PreSmp},\beta}$.

```
Games $\mathsf{G}_0 - \mathsf{G}_7$
$\mathcal{H}, \mathcal{Q} \leftarrow \emptyset$
$\ell := 0$
$\ell^\star \xleftarrow{\$} [Q_\mathsf{H} + 1]$                                  // $\mathsf{G}_2 - \mathsf{G}_7$
for $i \in [N]$
    $(\boldsymbol{B}_i, \boldsymbol{h}_i) \xleftarrow{\$} \mathsf{Gen}$
$(i^\star, m^\star, \sigma^\star) \xleftarrow{\$} \mathsf{A}^{\mathtt{Sgn}(\cdot,\cdot),\mathsf{H}(\cdot,\cdot,\cdot)}(\boldsymbol{h}_1, \dots, \boldsymbol{h}_N)$
parse $\sigma^\star \rightarrow (r^\star, \boldsymbol{s}_2^\star)$
$\boldsymbol{c}^\star := \mathsf{H}(\boldsymbol{h}_{i^\star}, r^\star, m^\star)$                           // $\mathsf{G}_2 - \mathsf{G}_7$
if $(\boldsymbol{c}^\star, \boldsymbol{h}_{i^\star}, r^\star, m^\star, \ell^\star, \cdot, \cdot) \notin \mathcal{H}$               // $\mathsf{G}_4 - \mathsf{G}_7$
    abort                                                  // $\mathsf{G}_4 - \mathsf{G}_7$
return $[\![\mathsf{Ver}(\boldsymbol{h}_{i^\star}, m^\star, \sigma^\star) = 1 \wedge (i^\star, m^\star, \cdot) \notin \mathcal{Q}]\!]$

Oracle $\mathtt{Sgn}(i \in [N], m)$
$cnt := 0$                                                 // $\mathsf{G}_1 - \mathsf{G}_7$
repeat
    $cnt \leftarrow cnt + 1$                                   // $\mathsf{G}_1 - \mathsf{G}_7$
    if $cnt > C_s$                                         // $\mathsf{G}_1 - \mathsf{G}_7$
        abort                                              // $\mathsf{G}_1 - \mathsf{G}_7$
    $r \xleftarrow{\$} \{0,1\}^k$
    $\boldsymbol{c} := \mathsf{H}(\boldsymbol{h}_i, r, m)$                                   // $\mathsf{G}_0 - \mathsf{G}_2$
    $(\boldsymbol{c}, \boldsymbol{s}_1, \boldsymbol{s}_2) := \mathsf{H}'(\boldsymbol{h}_i, r, m)$                         // $\mathsf{G}_3 - \mathsf{G}_7$
    $(\boldsymbol{s}_1, \boldsymbol{s}_2) \xleftarrow{\$} \mathsf{PreSmp}(\boldsymbol{B}_i, s, (\boldsymbol{c}, \boldsymbol{0}))$               // $\mathsf{G}_0 - \mathsf{G}_5$
    $(\boldsymbol{s}_1, \boldsymbol{s}_2) \xleftarrow{\$} \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}_i),s,(\boldsymbol{c},\boldsymbol{0})}$                          // $\mathsf{G}_6$
until $\|(\boldsymbol{s}_1, \boldsymbol{s}_2)\|_2 \leq \beta$
$\sigma := (r, \boldsymbol{s}_2)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(m, \sigma)\}$
return $\sigma$

Oracle $\mathsf{H}(pk, r, m)$
if $\exists\, \boldsymbol{c} : (\boldsymbol{c}, pk, r, m, \cdot, \cdot, \cdot) \in \mathcal{H}$
    return $\boldsymbol{c}$
$\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$
$\mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{c}, pk, r, m, \bot, \bot, \bot)\}$       // $\mathsf{G}_0 - \mathsf{G}_1$
$\ell := \ell + 1$                                         // $\mathsf{G}_2 - \mathsf{G}_7$
$\mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{c}, pk, r, m, \ell, \bot, \bot)\}$        // $\mathsf{G}_2 - \mathsf{G}_7$
return $\boldsymbol{c}$

Oracle $\mathsf{H}'(\boldsymbol{h}_i, r, m)$                                 // $\mathsf{G}_3 - \mathsf{G}_7$
if $\exists\, \boldsymbol{c} : (\boldsymbol{c}, \boldsymbol{h}_i, r, m, \cdot, \cdot, \cdot) \in \mathcal{H}$
    abort
$\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$
$(\boldsymbol{s}_1, \boldsymbol{s}_2) := (\bot, \bot)$
$\boldsymbol{s}_1, \boldsymbol{s}_2 \leftarrow \mathcal{D}_{\mathcal{R},s}$                               // $\mathsf{G}_5 - \mathsf{G}_7$
$\boldsymbol{c} \leftarrow \boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h}_i \mod q$                        // $\mathsf{G}_5 - \mathsf{G}_7$
$\mathcal{H} := \mathcal{H} \cup \{(\boldsymbol{c}, \boldsymbol{h}_i, r, m, \ell^\star, \boldsymbol{s}_1, \boldsymbol{s}_2)\}$
return $(\boldsymbol{c}, \boldsymbol{s}_1, \boldsymbol{s}_s)$
```

Figure 14. Games for the proof of Theorem 2.

Game G_2. This game is identical to the previous one except that the RO queries are counted by variable ℓ and a uniformly random RO query $\ell^\star$ is chosen from the set $[Q_\mathsf{H} + 1]$. Since this is only a conceptual change, it holds

$$\Pr[\mathsf{G}_1^\mathsf{A} \Rightarrow 1] = \Pr[\mathsf{G}_2^\mathsf{A} \Rightarrow 1].$$

Game G_3. This game is identical to the previous one except that it aborts in the signing oracle $\mathtt{Sgn}$ if there already exists a query to the random oracle on the same public key, salt r, and message m. To ease the depiction in further hybrids, we define a new RO H' maintaining the same set $\mathcal{H}$ as H but aborting in case of a query on the same input as a previous query. Oracle H' is then called within the signing oracle instead of H.

Claim 28:

$$\left|\Pr\left[\mathsf{G}_2^\mathsf{A} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_3^\mathsf{A} \Rightarrow 1\right]\right| \leq \frac{Q_s(C_s Q_s + Q_\mathsf{H})}{2^k}.$$

Game G_4. This game is identical to the previous one except that it aborts if the hash query corresponding to the output forgery does not equal the $\ell^\star$-th query guessed in the beginning or originated from the signing oracle.

Claim 29:

$$\Pr[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1] \leq (Q_{\mathsf{H}} + 1) \cdot \Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1].$$

Game G_5. This game is the same as the previous one, except that random oracle H' no longer returns a uniformly random element $\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$. Instead, it computes $\boldsymbol{c}$ as follows: It samples two elements $\boldsymbol{s}_1, \boldsymbol{s}_2$ from a Gaussian distribution $\mathcal{D}_{\mathcal{R},s}$ with standard deviation s over ring $\mathcal{R}$. Then, $\boldsymbol{c}$ is computed as $\boldsymbol{c} = \boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h}_i \mod q$, where $\boldsymbol{h}_i$ is the public key input to H'. For future use, $\boldsymbol{s}_1, \boldsymbol{s}_2$, along with the input and output to the random oracle, are stored in $\mathcal{H}$.

Claim 30: For distribution $\mathcal{P} \coloneqq \mathcal{U}(\mathcal{R}_q)$, distribution $\mathcal{Q}_{\boldsymbol{h}}$ the distribution of $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h} \mod q$ where $\boldsymbol{s}_1, \boldsymbol{s}_2 \xleftarrow{\$} \mathcal{D}_{\mathcal{R},s}$, and $a \in (1, \infty)$ it holds

$$\Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1] \leq \left(\max_{\boldsymbol{h} \neq \boldsymbol{0}} R_a(\mathcal{P} \mid\mid \mathcal{Q}_{\boldsymbol{h}})^{C_s Q_s} \cdot \Pr[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1] \right)^{\frac{a-1}{a}}.$$

Proof. In Game G_4, the output distribution of the RO is the uniform distribution over $\mathcal{R}_q$, named $\mathcal{P}$. In Game G_5, we use the distribution of $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h}_i \mod q$ where $\boldsymbol{s}_1, \boldsymbol{s}_2 \xleftarrow{\$} \mathcal{D}_{\mathcal{R},s}$, named $\mathcal{Q}_{\boldsymbol{h}_i}$ for simplicity. These underlying distributions are queried at most $Q \coloneqq C_s Q_s$ times due to $C_s Q_s$ queries to H'. Since for each query, distribution $\mathcal{Q}$ can be parametrized by a different $\boldsymbol{h}_i$ we define a vector $\mathbf{H} \coloneqq (\mathbf{H}_1, \ldots, \mathbf{H}_Q) \in \mathcal{R}_q^Q$ such that $\mathbf{H}_j$ equals the public key $\boldsymbol{h}_i$ used in the j-th query. Further we define $\mathcal{Q}_{\mathbf{H}}$ as the sequence of distributions $\mathcal{Q}_{\mathbf{H}_i}$. By the probability preservation, data processing inequality, and multiplicativity of the Rényi divergence (Lemma 7) we obtain

$$\begin{aligned}\Pr[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1] \geq \frac{\Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1]^{\frac{a}{a-1}}}{R_a(\mathsf{G}_4^{\mathsf{A}} \mid\mid \mathsf{G}_5^{\mathsf{A}})} &\geq \frac{\Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1]^{\frac{a}{a-1}}}{R_a(\mathcal{P}^Q \mid\mid \mathcal{Q}_{\mathbf{H}})} \\ &= \frac{\Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1]^{\frac{a}{a-1}}}{R_a(\mathcal{P}^Q \mid\mid \prod_{j=1}^{Q} \mathcal{Q}_{\mathbf{H}_j})} \geq \frac{\Pr[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1]^{\frac{a}{a-1}}}{\max_{\boldsymbol{h} \neq \boldsymbol{0}} R_a(\mathcal{P} \mid\mid \mathcal{Q}_{\boldsymbol{h}})^Q}.\end{aligned}$$

■

Game G_6. This game is identical to the previous one except that the output of the preimage sampler $\mathsf{PreSmp}(\boldsymbol{B}, s, (\boldsymbol{c}, \boldsymbol{0}))$ is replaced by a Gaussian over the same lattice, standard deviation and center, namely $\mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}),s,(\boldsymbol{c},\boldsymbol{0})}$.

Claim 31: For $r = \max_{(\boldsymbol{B},\cdot) \in \sup(\mathsf{Gen})} R_a(\mathsf{PreSmp}(\boldsymbol{B}, s, \cdot) \mid\mid \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}),s,\cdot})$ and $a \in (1, \infty)$ it holds

$$\Pr[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1] \leq \left(r^{C_s Q_s} \cdot \Pr[\mathsf{G}_6^{\mathsf{A}} \Rightarrow 1] \right)^{\frac{a-1}{a}}.$$

Proof. The claims follows by analogous arguments as for Game G_5. ■

Game G_7. This game is identical to the previous one except that the $\boldsymbol{s}_1, \boldsymbol{s}_2$ are not sampled from a Gaussian distribution centred at $(\boldsymbol{c}, \boldsymbol{0})$ as before. Instead, the preimage of $\boldsymbol{c}$ that was sampled in H' is used.

Claim 32:

$$\Pr[\mathsf{G}_6^{\mathsf{A}} \Rightarrow 1] = \Pr[\mathsf{G}_7^{\mathsf{A}} \Rightarrow 1].$$

Proof. We need to show that the distributions of the games are the same. The RO output $\boldsymbol{c}$ is the same in both games. In G_6, the singing oracle outputs $(\boldsymbol{s}_1, \boldsymbol{s}_s) \sim \mathcal{D}_{\boldsymbol{\Lambda}(\boldsymbol{B}_i),s,(\boldsymbol{c},\boldsymbol{0})}$. Since $\boldsymbol{\Lambda}(\boldsymbol{B}_i)$ is the NTRU lattice for $\boldsymbol{h}_i$ and q and the distribution is shifted by $(\boldsymbol{c}, \boldsymbol{0})$ the output is distributed according a Gaussian $\mathcal{D}_{\mathcal{R},s}$ conditioned on $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h}_i = \boldsymbol{c} \mod q$. The output distribution in Game G_7 is a Gaussian $\mathcal{D}_{\mathcal{R},s}$ as well where the condition $\boldsymbol{s}_1 + \boldsymbol{s}_2 * \boldsymbol{h}_i = \boldsymbol{c} \mod q$ is fulfilled by construction (Line 38). ■

*Reduction from $\mathcal{R}$-**ISIS**.* We now can reduce $\mathcal{R}$-**ISIS** to Game G_7.

Claim 33: There exists an adversary B against $\mathcal{R}$-**ISIS** such that

$$\Pr[\mathsf{G}_7^{\mathsf{A}} \Rightarrow 1] \le \mathrm{Adv}^{\mathcal{R}\text{-}\mathbf{ISIS}}_{N,q,\alpha,\beta,\mathsf{B}}.$$

Proof. Adversary B is formally constructed in Figure 15. Due to the changes in the previous games, adversary B can perfectly simulate the game for adversary A against G_7 without having any secret keys for $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N$. Further they embed their own target in the $\ell^\star$-th query to H. Let us assume, that A wins the unforgeability game G_7, i.e. the forgery verifies for user $i^\star$ and $(i^\star, m^\star, \cdot)$ was not queried before. This implies that it holds $\hat{\boldsymbol{c}} = \boldsymbol{c}^\star$ because if A is winning the game, it cannot abort in Line 08 and thus the challenge RO output $\boldsymbol{c}^\star$ corresponds to the guessed query (that is exactly $\hat{\boldsymbol{c}}$) or to a signing query. If it corresponds to a signing query, there is no way that adversary A can win the game due to the freshness condition $(i^\star, m^\star, \cdot) \notin \mathcal{Q}$. In the following note that $\boldsymbol{s}_{i^\star} = \boldsymbol{s}_2^\star$. Hence, Line 10 and Line 11 ensure the first winning condition of B:

$$\boldsymbol{s}_1^\star + \sum_{i \in [N]} \boldsymbol{s}_i * \boldsymbol{h}_i = \boldsymbol{s}_1^\star + \boldsymbol{s}_{i^\star} * \boldsymbol{h}_{i^\star} = \hat{\boldsymbol{c}}.$$

Further, the norm bound of the signature given by A directly translates to the second winning condition since all elements except for the $i^\star$-th are set to 0:

$$\|(\boldsymbol{s}_1^\star, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_N)\|_2 = \|(\boldsymbol{s}_1^\star, \boldsymbol{s}_{i^\star})\|_2 \le \beta.$$

■

```
B(h_1, ..., h_N, ĉ)
01  $\mathcal{H}, \mathcal{Q} \leftarrow \emptyset$
02  $\ell := 0$
03  $\ell^\star \xleftarrow{\$} [Q_{\mathsf{H}} + 1]$
04  $(i^\star, m^\star, \sigma^\star) \xleftarrow{\$} \mathsf{A}^{\mathtt{Sgn}(\cdot,\cdot),\mathsf{H}(\cdot,\cdot,\cdot)}(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N)$
05  parse $\sigma^\star \to (r^\star, \boldsymbol{s}_2^\star)$
06  $\boldsymbol{c}^\star := \mathsf{H}(\boldsymbol{h}_{i^\star}, r^\star, m^\star)$
07  if $(\boldsymbol{c}^\star, \boldsymbol{h}_{i^\star}, r^\star, m^\star, \ell^\star, \cdot, \cdot) \notin \mathcal{H}$
08      abort
09  $\boldsymbol{s}_1^\star := \boldsymbol{c}^\star - \boldsymbol{s}_2^\star * \boldsymbol{h}_{i^\star} \mod q$
10  $\boldsymbol{s}_i := \boldsymbol{0} \quad \forall\, i \in [N] \setminus \{i^\star\}$
11  $\boldsymbol{s}_{i^\star} := \boldsymbol{s}_2^\star$                    // embed solution
12  return $(\boldsymbol{s}_1^\star, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_N)$

Oracle H(pk, r, m)
13  if $\exists\, \boldsymbol{c} : (\boldsymbol{c}, pk, r, m, \cdot, \cdot, \cdot) \in \mathcal{H}$
14      return $\boldsymbol{c}$
15  $\boldsymbol{c} \xleftarrow{\$} \mathcal{R}_q$
16  $\ell := \ell + 1$
17  if $\ell = \ell^\star$
18      $\boldsymbol{c} := \hat{\boldsymbol{c}}$                    // embed challenge
19  $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{c}, pk, r, m, \ell, \bot, \bot)\}$
20  return $\boldsymbol{c}$

Oracle Sgn(i ∈ [N], m)
21  return $\mathsf{G}_7.\mathtt{Sgn}(i, m)$

Oracle H'(h_i, r, m)
22  return $\mathsf{G}_7.\mathsf{H}'(\boldsymbol{h}_i, r, m)$
```

Figure 15. Adversary B against $\mathcal{R}$-**ISIS** for the proof of Theorem 4.

Collecting the bounds yields the stated claim. ■