

Practical Group Signatures from Tag-Based NTRU Sampler

Corentin Jeudy

corentin.jeudy@orange.com

Orange Research, Applied Crypto Group, Cesson-Sévigné, France

Abstract. The post-quantum migration for key agreements and signatures being well underway, the focus naturally shifts to other properties and primitives that still lack efficient solutions. One such area is that of privacy-enhanced primitives, with a growing number of post-quantum constructions. Among the most fundamental are group signatures, which represent an important milestone of anonymity and accountability towards more involved designs. However, despite recent progress, most compact lattice constructions are either computationally intensive or suffer large key materials, or both. In this paper, we introduce a new lattice group signature scheme that reaches compact signatures and keys, while being runtime-efficient in all its procedures. Our key element is a new tag-based gadget sampler that properly interfaces with NTRU, then benefiting from the desirable properties of such trapdoor frameworks for advanced constructions while leveraging the compactness of NTRU. Although applied to group signatures, our sampler may be of independent interest and used for more expressive privacy-driven primitives like anonymous credentials. We also introduce two assumptions, which can be seen as hybrid versions of NTRU and ISIS, that independently underly the security of our group signature. Despite their similarities with the recently introduced NTRU-ISIS₊ assumption, they appear strictly harder and even admit reductions from R-ISIS in some parameter settings.

Keywords: Lattice-Based Cryptography · Group Signatures · Privacy · NTRU · Tag-Based Sampler

1 Introduction

Digital authentication is now ubiquitous in our daily use cases, realized using electronic certificates issued by an authority to attest of some rights. This simple and efficient approach however clashes with both the need for migrating current solutions to quantum-safe alternatives and the growing concern for decentralization and user privacy. This has led researchers to design post-quantum privacy-preserving mechanisms to lay the groundwork for the post-quantum transition in this area. One of the fundamental such primitives are group signatures. Coined by Chaum and van Heyst [CvH91], they offer flexibility between privacy and accountability, making them a pillar for many more advanced protocols [CL01,BCC04,BL07]. In a group signature, each user owns an individual

signing key received from a group manager, while the public verification key is common to all users. The signing keys allow users to authenticate while remaining anonymous among the group members. This naturally covers scenarios where one gives a set of individuals some shared rights, e.g., the group of users subscribed to public transportation or the group of citizens of a country, while giving them a way to remain private.

A typical conundrum of such anonymous signatures is to enforce accountability to avoid misbehaving users and possibly revoke them in case of abuses. In group signatures, this function is called opening and is entrusted to the group manager, which has the ability to lift the anonymity of any signature. If we take the example of a user claiming they are a European citizen to a border control agent, group signatures would allow to do so while staying anonymous among the group of European citizens, but the agent could still request the group manager (and its opening capability) as a safeguard to prevent revoked (i.e., unlawful) citizens from passing. In this case, the agent would learn nothing beyond that the user owns a Europe-delivered credential and whether it is blacklisted or not. Although, they were later enhanced to ensure a better trust model [BCC04] and even decentralized revocation [BL07,ST21], the main functionalities of group signatures for combining anonymity with accountability keep making them a fundamental primitive.

Far from being a theoretical solution, group signatures have had meaningful industrial impacts [ISO13,TCG15,Int16]. They thus represent an important milestone towards more expressive protocols. The lack of efficient post-quantum constructions is however alarming. Despite recent efforts in the lattice setting, for example [dPLS18,LNPS21,LNP22,LN22,DFJS26], the proposed solutions either suffer from large keys and/or signatures, or intensive computations. Designing a lattice-based group signature reaching small keys, small signatures, while being computationally efficient for key generation, signing, verification and opening then remains open.

1.1 Current Techniques

Lattice group signature designs are almost all based on gadget-based trapdoors and their associated preimage samplers. Originally introduced by Micciancio and Peikert [MP12], they provide a simple and efficient trapdoor generation procedure and are versatile enough to enable tag-based constructions, which is very suited to construct group signatures. Indeed, the tags represent the user identities, which allows for compactly differentiating user signing keys, and said signing key is then a simple preimage (for a given tag) of a public syndrome. Group membership can then be proven by using a non-interactive zero-knowledge (NIZK) proof of the preimage and tag, while the opening is done using an encryption of the tag under the group manager’s public encryption key.

The remaining desirable feature is efficiency. In their initial form, the trapdoors were designed to be used in LWE-based systems. As such, they unfortunately hardly compete with more compact trapdoor frameworks based on NTRU [DLP14,PFH⁺20]. Concretely, in [MP12], the trapdoor function is rep-

represented by $\mathbf{A}_t = [\mathbf{I}|\mathbf{A}|t\mathbf{G} - (\mathbf{R}_1 + \mathbf{A}\mathbf{R}_2)] \bmod q$ for a given tag t , where $(\mathbf{A}, \mathbf{B} = \mathbf{R}_1 + \mathbf{A}\mathbf{R}_2)$ is the (structured) LWE instance, and $\mathbf{G}$ the base- b gadget matrix. Hence, the dimension of preimages is $n(2 + \lceil \log_b q \rceil)$, which drastically hinders performance compared to NTRU trapdoors used in Falcon [PFH⁺20] where the dimension is only $2n$. Given that lattice-based NIZK, e.g., [LNP22], are very sensitive to the witness dimension, opting for a smaller dimension readily improves the group signature size. One differentiating feature is that gadget-based constructions interface well with tags, making advanced cryptographic primitives like group signatures their main applications, whereas NTRU trapdoors are so far limited to important but very specific use-cases such as standard digital signatures. Several directions have then been explored to improve group signatures.

Improving Gadget Samplers. A natural direction is to improve the efficiency by reducing the dimension of preimages produced by gadget samplers.

Approximate Trapdoors. A series of work focused on reducing said dimension by attacking the gadget component. The idea, initiated by Chen, Genise and Mukherjee [CGM19], is to rely on an *upper* base- b decomposition defined by $\mathbf{G}_H = [b^\ell \mathbf{I} \dots |b^{k-1} \mathbf{I}]$ with $k = \lceil \log_b q \rceil$, instead of the full one $\mathbf{G} = [\mathbf{G}_L | \mathbf{G}_H]$, with $\mathbf{G}_L = [\mathbf{I} | b\mathbf{I} \dots |b^{\ell-1} \mathbf{I}]$ being the *lower* part. The dimension reduction however does not come for free as the discarded lower part introduces an error on the preimage. But as long as this error remains small, one is able to preserve meaningful security assurances. The truncation parameter ℓ thus allows for trading security (due to a larger error) for efficiency (due to a reduced preimage dimension). This idea was later re-used with small variations [YJW23, JRS24], leading to new compact gadget samplers. Unfortunately, all these works were only able to prove the sampler secure on average, i.e., when the inverted syndrome is a random oracle output. This requirement prevents the use of such samplers in most advanced cryptographic designs, including group signatures, as they usually rely on inverting arbitrary (and sometimes even adversarially-controlled) syndromes.

Recently, Jeudy and Sanders [JS25b] proposed a new sampling approach while keeping the trapdoor structure, and proved it secure in the worst-case, i.e., for arbitrary and possibly adversarial syndromes. It generalized the original sampler from [MP12] while fixing gaps in the security proof. Most importantly, it unlocked the use of truncated gadgets in the main applications of gadget samplers, advanced primitives. The same authors later improved it [JS26] by designing a gadget structure and sampler which interfaces better with tags, an essential component to prove security on standard assumptions. By leveraging smooth moduli, they are able to choose smaller parameters, leading to more compact sizes.

iNTRU Trapdoors. Another angle to limit the preimage dimension is to take inspiration from the compactness of NTRU trapdoors. At a high level, LWE-based systems must somehow contain the LWE instance $(a, r_1 + ar_2 \bmod q)$ which explains why the matrix $\mathbf{A}_t$ contains both blocks $(\mathbf{A}$ and $\mathbf{B} = \mathbf{R}_1 + \mathbf{A}\mathbf{R}_2)$. Meanwhile, NTRU-based systems must also feature the NTRU instance $h =$

$ef^{-1} \bmod q$, but notice the latter contains fewer elements. Naturally, the latter produces more compact trapdoor functions. The caveat with NTRU trapdoors as in Falcon [PFH⁺20] is that it requires completing $[e| - f]^T$ into a full short basis of the NTRU lattice to allow for sampling preimages for $[1|h]$. The trapdoor generation is thus fairly complex. To get the (relative) best of both worlds, Genise and Li [GL20] introduced *inhomogeneous* NTRU (iNTRU) trapdoors. The idea is to change the definition of h to hide a gadget. More concretely, h is replaced by the vector $\mathbf{h} = (\mathbf{g} - \mathbf{e})f^{-1} \bmod q$, where $\mathbf{g}^T = [1|b|\dots|b^{k-1}]$ is the (one-dimensional) gadget from above. As a result, the trapdoor matrix is now $[1|(\mathbf{g}^T - \mathbf{e}^T)f^{-1}]$ which gives a preimage dimension of $n(1+k)$ (when the algebraic ring embeds to $\mathbb{Z}^n$). This essentially saves the block $\mathbf{A}$ from the previous matrix $\mathbf{A}_t$. The authors also enriched this iNTRU trapdoor with gadget truncation, resulting in $\mathbf{a}_t^T = [1|(\mathbf{g}_H^T + \mathbf{e}^T)f^{-1}]$ and preimages of dimension $n(1+k-\ell)$. It was pushed further in [YJW23] using their compact gadget sampler to reach dimension $2n$.

Although these constructions are more compact, they suffer from two important limitations, making them unamenable for group signatures: (1) they are only proven secure on average (as opposed to [MP12, JS25b, JS26] for LWE-based gadgets or [Kle00] for NTRU trapdoors), and (2) they are not particularly suited for tag-based constructions (just like NTRU trapdoors). Regarding (2), the trapdoors indeed do not preserve the initial structure of [MP12] of $t\mathbf{G} + \mathbf{pk}$ (or more generally since the work of [JS26] as $\mathbf{G}_{H,t} + \mathbf{pk}$). Although the authors of [GL20] mention their second construction can be instantiated with tags, they do not describe how. To explain why tags can be tricky, we note that defining $\mathbf{a}_t^T = [1|(t\mathbf{g}_H^T - \mathbf{e}^T)f^{-1}]$, to mimic the tag structure of [MP12], is likely insecure. In the context of group signatures, each user would receive some tag t and a preimage $\mathbf{x}$ of some public syndrome u . As they should not know the issuing key $(\mathbf{e}, f)$, this definition of $\mathbf{a}_t$ prevents them from computing the matrix from the iNTRU instance $(\mathbf{g}_H^T - \mathbf{e}^T)f^{-1}$ alone. Hence, the issuer would have to send $\mathbf{h}_t^T = (t\mathbf{g}_H^T - \mathbf{e}^T)f^{-1}$ for each different tag. However, two colluding users, i.e., with different tags $t \neq t'$, could then recover f^{-1} by computing $[\mathbf{h}_t^T - \mathbf{h}_{t'}^T]_0 \cdot (t - t')^{-1}$, thus recovering the issuer secret key. They could then use it to register new group members without the group manager, and thus break traceability. Another attempt could be to define $\mathbf{a}_t^T = [1|t(\mathbf{g}_H^T - \mathbf{e}^T)f^{-1}]$, but this makes efficiency unclear. Indeed, in typical tag-based group signatures, one must ensure the effective tag used during sampling remains hidden to the adversary. This allows for replacing $\mathbf{pk} = \mathbf{R}_1 + \mathbf{A}\mathbf{R}_2$ by $\mathbf{pk}' = \mathbf{R}_1 + \mathbf{A}\mathbf{R}_2 + t^+\mathbf{G}_H$ in LWE-type constructions so that the tag used in the sampling is $t - t^+$, while later allowing a security reduction to SIS¹. With this definition of $\mathbf{a}_t$, the tag contribution is now multiplicative with respect to the public key. As such, one would need to define $\mathbf{pk}' = \mathbf{pk}/t^+$. One would then either use an effective tag t/t^+ which can be extremely large (rendering the gadget truncation vacuous as

¹ This indeed allows for cancelling the gadget part when $t = t^+$ so that we would end up with $[\mathbf{I}|\mathbf{A}] - [\mathbf{I}|\mathbf{A}]\mathbf{R} = [\mathbf{I}|\mathbf{A}][\mathbf{I}|\mathbf{R}^T]^T$. The value t^+ is a guess on the (tag of the) honest user targetted in the traceability.

the preimage in this case depends on the size of the effective tag), or use an effective trapdoor ($\mathbf{e}, f' = ft^+$), which vastly deteriorates the parameters as t^+f is likely much bigger than f .

Overall, iNTRU trapdoors are so far unsuited for group signature constructions, let alone more advanced primitives.

Using Falcon in OR-proofs. Making NTRU trapdoors amenable for group signatures was recently explored by Lyubashevsky and Nguyen [LN22]. They introduced a group signature scheme directly based on Falcon signatures combined with an OR-proof of their design. They circumvent the problems of needing worst-case security of the sampler and that of having a tag-based construction by designing a scheme directly based on the security of Falcon. To do so, they however produce the signing keys as Falcon signature of the hashed user identity. At first glance, this would require the NIZK to prove this hash is properly evaluated, which leads to notoriously complex and inefficient proofs. To avoid such complex proofs, the authors resort to an OR-proof, by proving knowledge that the signing key (s_1, s_2) , i.e., a Falcon preimage, satisfies $s_1 + hs_2 \in \{\mathcal{H}(\text{id}_i); i \in [N]\}$, that is one of the hashed identities satisfies the Falcon verification equation, N being the group size.

Although they showcase remarkable sizes, generating such an OR-proof requires performing a number of fairly complex computations that scales linearly with the size of the group. This can thus quickly become hard to manage for larger groups, and then goes in the opposite direction of our initial goal to have a concretely efficient scheme.

Delegating Gadget Trapdoors. A recent construction of static group signature [DFJS26] attempted at rehabilitating the use of delegated trapdoors in such cryptographic primitives. The authors indeed greatly insist on the need to change paradigm from prior works [dPLS18, LNPS21, LNP22, LN22, JS25b], which are based on the approach of [BMW03], in order to reach more compact group signatures. They revisit the approach from [GKV10] of delegating trapdoors, but in a much more compact way. Using the tag-friendly sampler of [JS26], each user is given a tag t_i and a short basis $\mathbf{S}_i$ of the lattice $\mathcal{L}_q^\perp(\mathbf{A}_{t_i})$ where $\mathbf{A}_{t_i} = [\mathbf{I}|\mathbf{A}|q_L t_i \mathbf{G}_H - \mathbf{B}|\mathbf{a}_3]$. The group public key then only contains $\mathbf{A}, \mathbf{B}, \mathbf{a}_3$ ², which is common to all the $\mathbf{A}_{t_i}$ and thus constant in N . From $\mathbf{S}_i$, a user produces a preimage $\mathbf{v}$ of the first half of a Dual-Regev encryption $\mathbf{A}\mathbf{s} + \mathbf{e}$. They add the (very small) second half to enable opening and only have to hide $(\mathbf{s}, \mathbf{e}, t_i)$ in the NIZK (with low-dimensional extra randomness), while giving $\mathbf{v}$ in clear. Their construction relies on neat observations to add the least amount of complexity to allow for the opening to work. In particular, by using ideas from the Naor-Yung with shared randomness paradigm [BMV16], they show how to efficiently turn their construction into a CCA-anonymous one, with only a very small overhead.

At first glance, the construction of [DFJS26] seems to properly address limitations from prior works and in an efficient way. Indeed, as the preimage for

² The extra slot represented by $\mathbf{a}_3$ is actually common to all prior lattice group signatures. Using the trapdoor switching proof technique of [JS26] however allows this extra trapdoor slot to be only 1 column.

$\mathbf{A}_t$ is no longer hidden in the NIZK, it seems the dimension issue from above no longer applies. In addition, their construction does not seem to suffer from a large signing runtime. A closer inspection however highlights two shortcomings. First, because $\mathbf{v}$ must still be transmitted, reducing the dimension of $\mathbf{A}_t$ is still relevant. But most importantly, as $\mathbf{v}$ is produced from a delegated trapdoor of worse quality, their scheme requires larger parameters. Albeit better in sizes, it results in much larger keys, and likely in worse runtimes than showcased in [JS26]. The most concerning however lies in their key generation routine. The process to construct the basis $\mathbf{S}_i$ is indeed quite expensive. At a high level, the group manager first uses the sampler of [JS26] to construct a square Gaussian matrix $\mathbf{U}_i$ such that $\mathbf{A}_{t_i} \mathbf{U}_i = \mathbf{0} \bmod q$ (by successively sampling preimages of $\mathbf{0}$), but in such a way that the columns form a full-rank set of $\text{Span}_{\mathbb{R}}(\mathcal{L}_q^\perp(\mathbf{A}_{t_i}))$. Outputting $\mathbf{S}_i = \mathbf{U}_i$ is however insufficient as the lattice generated by $\mathbf{U}_i$ is only a sparse sublattice of $\mathcal{L}_q^\perp(\mathbf{A}_{t_i})$. This is why the authors have to resort to an algorithm ToBasis [MG02] to turn $\mathbf{U}_i$ into a basis $\mathbf{S}_i$ of same quality in polynomial time. Unfortunately, this last step involves computing an HNF of the coefficient matrix of $\mathbf{U}_i$, which is relatively high-dimensional ($D > 9000$ in [DFJS26]) and with relatively high coefficients. This algorithm has time complexity $\tilde{O}(D^4)$ which can quickly become cumbersome, especially if one has to repeat it $N \geq 2^{32}$ times to generate the whole set of group member keys.

Although the key generation is run once and this level of computations can be acceptable in practical deployments (e.g., key generation run in factory by the chip manufacturer), avoiding such intensive computations is preferable. What is less practically acceptable is the also fairly large user key size (i.e., $\mathbf{S}_i$) as it is aimed at being stored on possibly constrained secure elements (e.g., smart cards). Practical considerations then call for a faster key generation, smaller user signing keys, while preserving the benefits of [DFJS26], namely compactness, signing efficiency, and strong security guarantees (CCA-anonymity and full-traceability).

1.2 Our Contributions

In this paper, we first propose a new gadget-based sampler which offers tag-friendly features similar to those of [JS26] while benefiting from the improved compactness of NTRU. We show our sampler is secure in the worst-case, a necessary requirement for most advanced constructions, which we leverage to construct a practical group signature scheme. Our scheme has a faster key generation, smaller user and group keys, but also smaller signatures while preserving the assets of [DFJS26]. In doing so, we show that the classic approach of giving users a single certificate that they later show in zero-knowledge is actually more efficient than anticipated in [DFJS26]. We also show that the Naor-Yung with shared randomness method to reach CCA-anonymity applies independently of their trapdoor delegation approach. This in particular allows for choosing an independent and smaller encryption modulus, which reduces the ciphertexts size and boosts its security. This rehabilitates the “*certificate, encryption, zero-knowledge proof*” approach of [BMW03] over [DFJS26], shedding light on its attractive features that are crucial for practical use cases.

To showcase the practicality of our approach, we propose an implementation of our NTRU sampler in our group signature use-case. We show that it reaches similar preimage generation runtimes as that of [JS26], but features a key generation and verification that are orders of magnitude faster. We also give a second implementation which replaces the very complex arbitrary-precision Flint backend for floating-points operations by one that only relies on C native double precision floating-points. This optimization alone improves the signing runtime by a factor of 30 to 60. The detailed performance comparison is given in Section 6.2. We now describe the main technical ideas behind our sampler and group signature construction.

Compact Tag-Friendly Sampler from NTRU. Our first contribution is to propose a gadget sampler from NTRU that is proven in the worst-case and amenable to tag-based construction like group signatures. As opposed to [GL20] whose trapdoor framework is based on the inhomogeneous version of NTRU learning³, ours relies on the standard NTRU problem. More concretely, we define the vector $\mathbf{h} = \mathbf{e} \oslash \mathbf{f} \bmod q$ where $\mathbf{h}$ matches the dimensions of the gadget $\mathbf{g}_H$, and $\oslash$ denotes the entrywise division. Said differently, $\mathbf{h}$ is a vector of k_H independent NTRU instances, where $\mathbf{e}$ and $\mathbf{f}$ are very small polynomial vectors, i.e., ternary. Our trapdoor matrix is then $\mathbf{a}_t^T = [1|q_L t \mathbf{g}_H^T - \mathbf{h}^T]$, where q_L divides q as in the sampler of [JS26]. Taking inspiration from the latter, we design a Gaussian sampler for this matrix that is proven in the worst-case. Most gadget-based trapdoor matrices have a very similar form, and the complexity lies in proving the security of the sampler which determines its quality. We give more details on the inner workings and security of our sampler in Section 3.1. Additionally, thanks to the structure $t \times \text{gadget} + \text{pk}$, we do not suffer from the same limitations as [GL20] in the handling of tags. In particular, by extending the matrix into $\mathbf{a}_t^T = [1|q_L t \mathbf{g}_H^T - \mathbf{h}^T | a_3] \bmod q\mathcal{R}$ with a single polynomial a_3 , we show that the trapdoor switching technique of [JS26] can be applied to our case, which is a crucial feature for the security proof of our group signature. As mentioned above, we propose two efficient implementations of our sampler showing it is highly competitive and outperforms that of [JS26] in key generation, preimage sampling, and verification.

Compact and Runtime-Efficient Group Signature. We then follow the approach of [BMW03] to construct a group signature from our sampler. More precisely, each user receives a tag t_i corresponding to its identity, and a preimage $\mathbf{v}_i$ such that $\mathbf{a}_{t_i}^T \mathbf{v}_i = u \bmod q\mathcal{R}$. To sign, the user computes a Dual-Regev encryption of t_i and a NIZK of $(t_i, \mathbf{v}_i)$ and the ciphertext randomness such that $\mathbf{a}_{t_i}^T \mathbf{v}_i = u \bmod q\mathcal{R}$ and that the ciphertext is well-formed. We extend this construction to a CCA-anonymous one by showing the Naor-Yung with shared randomness paradigm [BMV16] used in [DFJS26] can be adapted to our case and is independent of their trapdoor delegation construction. The anonymity then relies on a similar assumption, that is a hint variant of M-LWE which reduces

³ The problem where several NTRU samples share the same denominator, i.e., $h_1 = e_1 f^{-1}, \dots, h_d = e_d f^{-1}$, has been referred to as NTRU learning [Pei16, Def. 4.4.4] or Tuple-NTRU [PS21, Def. 3.1] in the literature.

from the standard version of M-LWE. Our traceability proof differs from previous constructions due to the peculiarities of our sampler. As in previous proofs, we guess the identity t^+ targetted by the adversary, and change the public key $\mathbf{h}$ to $\mathbf{h} + q_L t^+ \mathbf{g}_H$. By carefully laying out the trapdoor switching argument of our sampler, we show that this change can be done unbeknownst to the adversary under NTRU. Then, extracting the NIZK from the forged group signature, we obtain some $\mathbf{v}^*$ such that $[1|\mathbf{h}^T|a_3]\mathbf{v}^* = u \bmod q\mathcal{R}$, provided that our guess on the identity is correct. At first glance, this would reduce from R-ISIS by changing $\mathbf{h}$ to a uniform vector under NTRU and using $\mathbf{v}^*$ as the R-ISIS solution. But doing so removes the NTRU trapdoor we need to produce the signing keys of other users (and respond to the adversary corruption and signing queries). Relying directly on a hybrid version of NTRU and R-ISIS, where the matrix is $\mathbf{a}^T = [1|\mathbf{h}^T|a_3]$ allowing us to hide an NTRU structure for $\mathbf{h} = \mathbf{e} \oslash \mathbf{f}$, would yield the same issue as embedding this challenge matrix prevents the challenger from producing the user signing keys. By instead relying on a q -type assumption we call T-NTRU-ISIS, consisting of solving R-ISIS on the matrix $\mathbf{a}^T$ while being given preimages $\mathbf{v}_i$ of the R-ISIS syndrome u for $\mathbf{a}_{t_i}^T = [1|q_L t_i \mathbf{g}_H^T - \mathbf{h}^T|a_3]$, we are able to conclude the traceability proof. Although this resembles the NTRU-ISIS_f assumption introduced in [BLNS23], we explain in details how our assumption differs from the latter in Section 4. In particular, we observe that we can remove the q -type and avoid giving the preimages by simply giving $\mathbf{e}, \mathbf{f}$ to the adversary of the assumption. Doing this in NTRU-ISIS_f would readily break the assumption as the latter is defined so that the NTRU trapdoor can be used to sample preimages of $[1|\mathbf{h}^T|a_3]$ directly. In our case, the NTRU keys are much smaller and then incompatible with samplers like Falcon's, preventing such trivial attacks. In particular, we show that in specific parameter settings, this assumption which we call Known-NTRU-ISIS admits a reduction from regular R-ISIS, which is not known for NTRU-ISIS_f. Although we step out of these parameter settings for efficiency, this gives more confidence in the security assumptions underpinning our group signature.

In the end, our group signature improves over previous constructions in that it is computationally efficient in all its routines (key generation, signing, verification, opening), with small keys, while reaching compact signature sizes of under 40 KB, which is roughly 30% smaller than the recent construction of [DFJS26], and about twice as small as the previous runtime-efficient construction of [JS25b]. A detailed comparison with prior group signatures mentioned in Section 1.1 is given in Section 6.3.

2 Preliminaries

For integers $m \leq n$, we define $[m, n] = \{m, \dots, n\}$, and $[n] = [1, n]$. The group of units in a ring R is denoted by $R^\times$, and we extend the notation to $\mathbb{N}$ by defining $\mathbb{N}^\times = \mathbb{N} \setminus \{0\}$. Further, for $q \in \mathbb{N}^\times$, we let $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ which can be identified by the set of representatives $[0, q-1]$ or $[-q/2, q/2) \cap \mathbb{Z}$. When the centered representation of some $a \in \mathbb{Z}_q$ is needed, we use $a \bmod^\pm q$. Vectors are written

in bold lower case $\mathbf{a}$ and are considered column vectors, while matrices are in bold upper case $\mathbf{A}$. The transpose operator of a matrix or vector is denoted by $\mathbf{A}^T$ or $\mathbf{a}^T$. The j -th entry of a vector $\mathbf{a}$ is sometimes denoted by $[\mathbf{a}]_j$. For a ring R and $\mathbf{a}, \mathbf{b} \in R^d$, we denote by $\mathbf{c} = \mathbf{a} \odot \mathbf{b}$ the entrywise product of $\mathbf{a}$ and $\mathbf{b}$, i.e., $[\mathbf{c}]_j = [\mathbf{a}]_j [\mathbf{b}]_j$ for all j . When $\mathbf{b} \in (R^\times)^d$, we denote by $\mathbf{c} = \mathbf{a} \oslash \mathbf{b}$ the entrywise division of $\mathbf{a}$ by $\mathbf{b}$, i.e., $[\mathbf{c}]_j = [\mathbf{a}]_j ([\mathbf{b}]_j)^{-1}$ for all j . Finally, $\|\mathbf{a}\|_p$ denotes the standard ℓ^p norm of $\mathbf{a} \in \mathbb{R}^n$, and $\|\mathbf{A}\|_2 = \max_{\mathbf{x} \neq \mathbf{0}} \|\mathbf{A}\mathbf{x}\|_2 / \|\mathbf{x}\|_2$ the spectral norm of matrix $\mathbf{A}$.

2.1 Algebraic Number Theory

We give the necessary background in algebraic number theory needed to comprehend our contributions, and refer to the literature for more details. Although some of our result extend to more general number fields, we only consider power-of-two cyclotomic fields for the sake of simplicity. Formally, for $n \in \mathbb{N}^\times$ a power of two, we define the number field $\mathcal{K}$ through its quotient ring isomorphic description as $\mathcal{K} = \mathbb{Q}[x]/\langle x^n + 1 \rangle$. We then define the ring $\mathcal{K}_{\mathbb{R}} = \mathbb{R}[x]/\langle x^n + 1 \rangle$, and the ring of integers $\mathcal{R} = \mathbb{Z}[x]/\langle x^n + 1 \rangle$. For an integer q , we define $\mathcal{R}_q = \mathcal{R}/q\mathcal{R} = \mathbb{Z}_q[x]/\langle x^n + 1 \rangle$. The following notions are detailed for the ring of integers but apply to $\mathcal{K}, \mathcal{K}_{\mathbb{R}}, \mathcal{R}_q$ in the natural way by changing the underlying polynomial ring.

Embedding in Subrings. As described for example in [LNPS21, AGJ+24], the ring $\mathcal{R}$ can be embedded into modules over its subrings. This is particularly useful to offer more flexibility over the choice of rings in cryptographic constructions with several components having different requirements. We formally describe it here for completeness but refer to [LNPS21, AGJ+24] for ampler discussions.

Consider two power-of-two cyclotomic rings $\mathcal{R}_2 \subseteq \mathcal{R}_1$ of respective degree $n_2 | n_1$, and define $k = n_1/n_2$. We write $\otimes_i$ the multiplication in $\mathcal{R}_i$ respectively to avoid confusion. We define the subring embedding $\theta : \mathcal{R}_1 \rightarrow \mathcal{R}_2^k$ by

$$\theta \left(\sum_{j=0}^{n_1-1} a_j x^j \right) = \left[\sum_{j=0}^{n_2-1} a_{kj+i} x^j \right]_{i \in [0, k-1]} \in \mathcal{R}_2^k.$$

For $\mathbf{b} = [b_i]_{i \in [0, k-1]}$, the inverse embedding $\theta^{-1}(\mathbf{b})$ is then defined by $\theta^{-1}(\mathbf{b}) = \sum_{i \in [0, k-1]} b_i (x^k) \otimes_1 x^i$. We note that for $b \in \mathcal{R}_2$, we can simply define the element $a = b(x^k) \in \mathcal{R}_1$. For this element, we have $\theta(a) = [b|0| \dots |0]^T \in \mathcal{R}_2^k$.

Multiplication over $\mathcal{R}_1$ maps to a structured matrix-vector multiplication over $\mathcal{R}_2^k$ through the embedding θ . Formally, there is a ring homomorphism $M_\theta : \mathcal{R}_1 \rightarrow \mathcal{R}_2^{k \times k}$ defined by $M_\theta(a) = [\theta(a)|\theta(a \otimes_1 x)| \dots |\theta(a \otimes_1 x^{k-1})]$, such that for any $a, b \in \mathcal{R}$, $\theta(a \otimes_1 b) = M_\theta(a)\theta(b)$ with the matrix-vector product being performed with $\otimes_2$. We extend θ to vectors over $\mathcal{R}_1$ by concatenation, and M_θ to matrices entrywise, thus defining a block matrix $M_\theta([a_{i,j}]_{i,j}) = [M_\theta(a_{i,j})]_{i,j}$.

Coefficient, Canonical Embeddings, and Conjugate. The coefficient embedding of $\mathcal{R}$, which we denote by τ , is a specific case of the prior subring

embedding where $\mathcal{R}_1 = \mathcal{R}$ and $\mathcal{R}_2 = \mathbb{Z}$. It then simplifies to $\tau(\sum_i a_i x^i) = [a_0 | \dots | a_{n-1}]^T$. The multiplication matrix map also simplifies to $M_\tau(a)$ defined as the negacirculant matrix over $\mathbb{Z}^{n \times n}$ with first column $\tau(a)$, i.e.,

$$M_\tau(a) = \begin{bmatrix} a_0 & -a_{n-1} & \dots & -a_1 \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & -a_{n-1} \\ a_{n-1} & a_{n-2} & \dots & a_0 \end{bmatrix}.$$

We also denote by σ the canonical embedding of $\mathcal{K}$ characterized by mapping x to the primitive $2n$ -th roots of unity in $\mathbb{C}$. Hence, σ maps $\mathcal{R}$ to $\mathbb{C}^n$. More details can be found in the literature. The set of elements a for which the $\sigma(a)$ are positive reals is denoted $\mathcal{K}_{\mathbb{R}}^{++}$.

We extend the ℓ^p norms to $\mathcal{R}$ through the coefficient embedding τ as $\|a\|_p = \|\tau(a)\|_p$. We also define the conjugate operator by $\tau(a^*) = [a_0 | -a_{n-1} | \dots | -a_1]^T$. It then entails $M_\tau(a^*) = M_\tau(a)^T$. The conjugate operator extends to vectors and matrices just like the usual Hermitian operator, by conjugating each entry and transposing the vector or matrix.

Units. The NTRU assumption as well as our sampler requires inverting ring elements modulo q or factors of q . It means we must ensure certain short ring elements are modular units. For that, we use the result from [LS18, Cor. 1.2] to argue that for moduli splitting properly, all non-zero short elements are modular units. We adapt it slightly to be formulated modulo arbitrary moduli rather than primes, which we later need due to our smooth modulus.

Lemma 2.1 ([LS18, Cor. 1.2]). *Let $\mathcal{R}$ be a power-of-two cyclotomic of degree n , and let $\kappa \in [2, n]$ be a power of two. Let $p_1, k_1, \dots, p_r, k_r$ be in $\mathbb{N}^\times$ such that the p_i are distinct, prime, and satisfy $p_i = 2\kappa + 1 \pmod{4\kappa}$. Then, define $q = \prod_{i \in [r]} p_i^{k_i}$. For any $y \in \mathcal{R} \setminus \{0\}$ such that $\|y\|_\infty < \frac{1}{\kappa} (\min_i p_i)^{1/\kappa}$, it holds $y \bmod q\mathcal{R} \in \mathcal{R}_q^\times$.*

Proof. Using [LS18, Cor. 1.2] for each p_i , it holds that such an element y verifies $y \bmod p_i \mathcal{R} \in \mathcal{R}_{p_i}^\times$ for all $i \in [r]$. By the Chinese Remainder Theorem, the $p_i^{k_i}$ being coprime, y being invertible modulo $q\mathcal{R}$ is equivalent to y being invertible modulo each $p_i^{k_i} \mathcal{R}$.

It yet holds that for a prime p , invertibility modulo $p^k \mathcal{R}$ is equivalent to invertibility modulo $p\mathcal{R}$. Indeed, let $a \in \mathcal{R}$ and assume $a \bmod p^k \mathcal{R} \in \mathcal{R}_{p^k}^\times$. Then, there exists $b, c \in \mathcal{R}$ such that $ab = 1 + p^k c$ in $\mathcal{R}$. Hence, $ab \bmod p\mathcal{R} = 1$, and as such $a \bmod p\mathcal{R}$ is in $\mathcal{R}_p^\times$. Reciprocally, assume $a \bmod p\mathcal{R} \in \mathcal{R}_p^\times$. There exists $b, c \in \mathcal{R}$ such that $ab = 1 - pc$ in $\mathcal{R}$. As a result, we get

$$a \cdot b \sum_{i=0}^{k-1} (pc)^i = (1 - pc) \sum_{i=0}^{k-1} (pc)^i = 1 - p^k c^k.$$

It thus means that, by defining $b' = b \sum_{i=0}^{k-1} (pc)^i$, we have $ab' \bmod p^k \mathcal{R} = 1$. It shows $a \bmod p^k \mathcal{R}$ is invertible, but also gives an expression of its inverse based on its inverse modulo $p\mathcal{R}$ and the lifting term c .

Applying the above successively to $p = p_i, k = k_i$ and using that $y \bmod p_i \mathcal{R} \in \mathcal{R}_{p_i}^\times$ gives that y is a unit modulo each $p_i^{k_i} \mathcal{R}$. Hence, $y \bmod q\mathcal{R} \in \mathcal{R}_q^\times$ by the Chinese Remainder Theorem mentioned above. $\square$

(Structured) Lattices. A (full-rank) n -dimensional lattice $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^n$, which can be defined by a basis $\mathbf{B} \in \mathbb{R}^{n \times n}$ as $\mathcal{L} = \mathcal{L}(\mathbf{B}) = \mathbf{B}\mathbb{Z}^n$. The dual of a lattice $\mathcal{L}$ is defined by $\mathcal{L}^\vee = \{\mathbf{x} \in \text{Span}_{\mathbb{R}}(\mathcal{L}) : \forall \mathbf{y} \in \mathcal{L}, \mathbf{x}^T \mathbf{y} \in \mathbb{Z}\}$. When $\mathcal{L} = \mathcal{L}(\mathbf{B})$ is full-rank, we can derive a basis of $\mathcal{L}^\vee$ as $\mathbf{B}^\vee = \mathbf{B}(\mathbf{B}^T \mathbf{B})^{-1} = \mathbf{B}^{-T}$, i.e., $\mathcal{L}^\vee = \mathcal{L}(\mathbf{B}^\vee) = \mathbf{B}^{-T} \mathbb{Z}^n$.

For $\mathbf{A} \in \mathcal{R}_q^{d \times m}$ and $\mathbf{u} \in \mathcal{R}_q^d$, we define the q -ary structured lattice coset $\mathcal{L}_q^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{x} \in \mathcal{R}_q^m : \mathbf{A}\mathbf{x} = \mathbf{u} \bmod q\mathcal{R}\}$. Strictly speaking, this is a lattice coset when embedded into $\mathbb{R}^n$ (via τ for example). The corresponding lattice coset would be $\tau(\mathcal{L}_q^{\mathbf{u}}(\mathbf{A})) = \mathcal{L}_q^{\tau(\mathbf{u})}(M_\tau(\mathbf{A})) = \{\mathbf{x} \in \mathbb{Z}^{nm} : M_\tau(\mathbf{A})\mathbf{x} = \tau(\mathbf{u}) \bmod q\mathbb{Z}\}$. We similarly define the corresponding structured lattice by $\mathcal{L}_q^\perp(\mathbf{A}) = \mathcal{L}_q^{\mathbf{0}}(\mathbf{A})$.

2.2 Probabilities

For a probability distribution $\mathcal{P}$ over S , we use $x \leftarrow \mathcal{P}$ for the action of sampling $x \in S$ from $\mathcal{P}$, and $x \sim \mathcal{P}$ to say x is a random variable following the distribution $\mathcal{P}$. We use $\mathcal{P}_0 \times \mathcal{P}_1$ to denote the joint distribution of $\mathcal{P}_0$ and $\mathcal{P}_1$, and $\mathcal{P}^n$ for the joint distribution of n copies of $\mathcal{P}$. For a possibly randomized function f from S to S' , we call $f(\mathcal{P})$ the distribution obtained by sampling X from $\mathcal{P}$ and outputting $f(X)$. Finally, for two distributions $\mathcal{P}_0, \mathcal{P}_1$ over the same support S , we use the notation $\mathcal{P}_0 \approx_\delta \mathcal{P}_1$ to denote that for all $x \in S$, $\mathcal{P}_0(x) \in [\delta^{-1}, \delta] \mathcal{P}_1(x)$.

Uniform and Binomial Distributions. The uniform distribution over a discrete finite set S is denoted by $U(S)$. We also define ψ_1 to be the centered binomial distribution of parameter 1, i.e., $\psi_1 = U(\{0, 1\}) - U(\{0, 1\})$. We define it over rings as $\mathcal{B}_1 = \tau^{-1}(\psi_1^n)$. Our construction uses different rings, and thus different degrees n . To avoid overloading notations, we use the same notation $\mathcal{B}_1$ for the centered binomial distribution over different rings, but it should be clear from the context in which ring each element lies in. We may use $\{0\}$ to denote the deterministic distribution always outputting 0.

Relative Error Lemma. We use the following relative error lemma of [Pre17] to relate the probabilities of an event under two relatively close distributions. We use Lemma 2.2 to thoroughly track the loss of our trapdoor switching hybrid argument in our security proof.

Lemma 2.2 ([Pre17, Lem. 3] adapted). *Let $\mathcal{P}_0, \mathcal{P}_1$ be two distributions over the same support S . Let $\Delta \geq 0$ such that for all $x \in S$, $\mathcal{P}_0(x) \in [1 - \Delta, 1 + \Delta] \mathcal{P}_1(x)$. It then holds that for any $a > 1$, any $N \in \mathbb{N}^\times$, any (possibly randomized) function f with domain S^Q and range S' , and any event $E \subseteq S'$, we*

have

$$f(\mathcal{P}_0^N)(E) \leq \left(1 + \frac{a(a-1)\Delta^2}{2(1-\Delta)^{a+1}}\right)^{N/a} f(\mathcal{P}_1^N)(E)^{1-1/a}.$$

Discrete Gaussian Measures. We define the Gaussian function of parameter $\mathbf{S} \in \mathbb{R}^{n \times n}$ (positive definite) and center $\mathbf{c} \in \mathbb{R}^n$ by $\rho_{\sqrt{\mathbf{S}}, \mathbf{c}}(\mathbf{x}) = \exp(-\pi(\mathbf{x} - \mathbf{c})^T \mathbf{S}^{-1}(\mathbf{x} - \mathbf{c}))$. Then, for a lattice coset $\mathcal{L} + \mathbf{x} \subset \mathbb{R}^n$, we define the probability mass function of the discrete Gaussian distribution over $\mathcal{L} + \mathbf{x}$, parameter $\mathbf{S}$ and center $\mathbf{c}$ as $\mathcal{D}_{\mathcal{L}+\mathbf{x}, \sqrt{\mathbf{S}}, \mathbf{c}}(\cdot) = \rho_{\sqrt{\mathbf{S}}, \mathbf{c}}(\cdot) \mathbf{1}_{\mathcal{L}+\mathbf{x}}(\cdot) / \rho_{\sqrt{\mathbf{S}}, \mathbf{c}}(\mathcal{L} + \mathbf{x})$. The function $\mathbf{1}_{\mathcal{L}+\mathbf{x}}$ is the indicator of $\mathcal{L} + \mathbf{x}$, while $\rho_{\sqrt{\mathbf{S}}, \mathbf{c}}(\mathcal{L} + \mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{L}+\mathbf{x}} \rho_{\sqrt{\mathbf{S}}, \mathbf{c}}(\mathbf{y})$. When $\mathbf{c} = \mathbf{0}$, we omit the subscript and when $\mathbf{S} = s^2 \mathbf{I}_n$, we simply write $\mathcal{D}_{\mathcal{L}+\mathbf{x}, s, \mathbf{c}}$. We extend it to Gaussians over $\mathcal{R}$ -modules $M \subseteq \mathcal{R}^d$ using τ , by defining $\mathcal{D}_{M, \sqrt{\mathbf{S}}, \mathbf{c}} = \tau^{-1}(\mathcal{D}_{\tau(M), \sqrt{\mathbf{S}}, \tau(\mathbf{c})})$.

Above a certain threshold for the parameter s , discrete Gaussian distributions benefit from similar properties to their continuous counterparts, in particular in terms of regularity, concentration, and convolution properties. This threshold, called the smoothing parameter by Micciancio and Regev [MR07], essentially smoothes out the discrete behavior. The smoothing parameter of a lattice $\mathcal{L}$ is defined by $\eta_\varepsilon(\mathcal{L}) = \min\{s > 0 : \rho_{1/s}(\mathcal{L}^\vee) = 1 + \varepsilon\}$ for any $\varepsilon > 0$. For Gaussians with more general parameters $\mathbf{S}$, we say that $\sqrt{\mathbf{S}} \geq \eta_\varepsilon(\mathcal{L})$ whenever $\rho_{\sqrt{\mathbf{S}}^{-1}}(\mathcal{L}^\vee) \leq 1 + \varepsilon$. The latter is equivalent to $\mathbf{S} \geq \eta_\varepsilon(\mathcal{L})^2 \mathbf{I}_n$ for the Loewner order. The smoothing parameter can be bounded using the minima of the lattice, but tighter approximations exist for remarkable lattices like $\mathbb{Z}^n$.

Lemma 2.3 ([EWY23, Lem. 5]). *For any $n \in \mathbb{N}^\times$, it holds that $\eta_\varepsilon(\mathbb{Z}^n) \approx \sqrt{\ln(2n/\varepsilon)}/\pi$.*

Among the desired properties beyond the smoothing parameter, it first holds that the Gaussian mass of a lattice coset is roughly that of the lattice itself.

Lemma 2.4 ([MR07] adapted). *Let n be in $\mathbb{N}^\times$, $\mathcal{L} \subset \mathbb{R}^n$ be a full-rank lattice, $\varepsilon > 0$, and $\mathbf{S} \in \mathbb{R}^{n \times n}$ be positive definite such that $\sqrt{\mathbf{S}} \geq \eta_\varepsilon(\mathcal{L})$. Then, for all $\mathbf{c} \in \mathbb{R}^n$, we have $\rho_{\sqrt{\mathbf{S}}}(\mathcal{L} + \mathbf{c}) \in [\frac{1-\varepsilon}{1+\varepsilon}, 1] \rho_{\sqrt{\mathbf{S}}}(\mathcal{L})$.*

We also have the following concentration bound for centered discrete Gaussians, which does not require the parameter to be above the smoothing threshold. For an implicit tailcut probability p , we define $c_N = \inf\{c > 1/\sqrt{2\pi} : (c\sqrt{2\pi}ee^{-\pi c^2})^N \leq p\}$.

Lemma 2.5 ([Ban93, Lem. 1.5]). *Let n be in $\mathbb{N}^\times$, and $\mathcal{L} \subset \mathbb{R}^n$ be a full-rank lattice. For all $s > 0$ and $c > 1/\sqrt{2\pi}$, $\mathbb{P}_{\mathbf{x} \sim \mathcal{D}_{\mathcal{L}, s}}[\|\mathbf{x}\|_2 > cs\sqrt{n}] \leq (c\sqrt{2\pi}ee^{-\pi c^2})^n$.*

2.3 Hardness Assumptions

This work considers several assumptions, for which we quantify the hardness using their hardness bound. For a problem P where the advantage of an adversary $\mathcal{A}$ is called $\text{Adv}_P[\mathcal{A}]$, we define the hardness bound of P by $\varepsilon_P =$

$\sup_{\mathcal{A}} \text{Adv}_{\text{PPT}}[\mathcal{A}]$, where the parameters of the problem are clear from the context. We now recall in this section the definitions of the lattice assumptions underlying the security of our group signature construction. In particular, we first recall the definitions of the decisional *NTRU* problem introduced in [HPS98], as well as the *Module Learning With Errors* [LS15]. We recall their definition here.

Definition 2.1 (NTRU). Let $\mathcal{R}$ be a power-of-two cyclotomic ring of degree n . Let q be in $\mathbb{N}^\times$ and $\mathcal{D}$ be a distribution on $\mathcal{R}$. The *NTRU* problem $\text{NTRU}_{n,q,\mathcal{D}}$ asks to distinguish between $\mathcal{P}_0 = U(\mathcal{R}_q)$ and the distribution $\mathcal{P}_1$ of $ef^{-1} \bmod q\mathcal{R}$, where $f \sim \mathcal{D}$ conditioned on $f \bmod q\mathcal{R} \in \mathcal{R}_q^\times$ and $e \sim \mathcal{D}$. The advantage is $\text{Adv}_{\text{NTRU}}[\mathcal{A}] = |\mathbb{P}_{h \sim \mathcal{P}_0}[\mathcal{A}(h) = 1] - \mathbb{P}_{h \sim \mathcal{P}_1}[\mathcal{A}(h) = 1]|$.

Definition 2.2 (M-LWE). Let $\mathcal{R}$ be a power-of-two cyclotomic ring of degree n . Let $d, m, p \in \mathbb{N}^\times$ and $\mathcal{D}$ be a distribution over $\mathcal{R}^{d+m}$. The *Module Learning With Errors* problem, denoted $\text{M-LWE}_{n,d,m,p,\mathcal{D}}$ asks to distinguish between $\mathcal{P}_0 = U(\mathcal{R}_p^{m \times d} \times \mathcal{R}_p^m)$ and the distribution $\mathcal{P}_1$ of $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod p\mathcal{R})$, where $\mathbf{A} \sim U(\mathcal{R}_p^{m \times d})$ and $\mathbf{x} = [\mathbf{s}^T | \mathbf{e}^T]^T \sim \mathcal{D}$. We define $\text{Adv}_{\text{M-LWE}}[\mathcal{A}] = |\mathbb{P}_{(\mathbf{A}, \mathbf{b}) \sim \mathcal{P}_0}[\mathcal{A}(\mathbf{A}, \mathbf{b}) = 1] - \mathbb{P}_{(\mathbf{A}, \mathbf{b}) \sim \mathcal{P}_1}[\mathcal{A}(\mathbf{A}, \mathbf{b}) = 1]|$ as the advantage of a PPT adversary in solving M-LWE.

The security of our scheme also relies on a specific *hint* variant of M-LWE. The latter, called *Matrix-Hint M-LWE*, was introduced in [ENP24], and generalized to arbitrary matrices and multiple secrets in [DFJS26]. In addition to $\mathbf{A}, \mathbf{b}$, it considers hints of the form $\mathbf{z}_i = \mathbf{y}_i + \mathbf{M}_i \mathbf{x}$ (or on the coefficient vector of $\mathbf{x}$ for more granular hints). We use a specific setting of that of [DFJS26] by considering only one hint and one secret, which we formalize here.

Definition 2.3 (MH-M-LWE). Let $\mathcal{R}$ be a power-of-two cyclotomic ring of degree n . Let $d, m, p, k \in \mathbb{N}^\times$ and $\mathcal{D}_s, \mathcal{D}_y, \mathcal{M}$ be distributions over $\mathcal{R}^{d+m}, \mathcal{R}^k$, and $\mathcal{R}^{k \times (m+d)}$ respectively. The *Matrix-Hint Module Learning With Errors* problem, denoted $\text{MH-M-LWE}_{n,d,m,p,\mathcal{D}_s,\mathcal{D}_y,\mathcal{M}}$ asks to distinguish between the following distributions.

$\mathcal{P}_0$	Sample $\mathbf{A} \leftarrow U(\mathcal{R}_p^{m \times d})$, $\mathbf{x} \leftarrow \mathcal{D}_s$, $\mathbf{M} \leftarrow \mathcal{M}$, $\mathbf{y} \leftarrow \mathcal{D}_y$. Sample $\mathbf{b} \leftarrow U(\mathcal{R}_p^m)$, compute $\mathbf{z} = \mathbf{y} + \mathbf{M}\mathbf{x}$, and output $(\mathbf{A}, \mathbf{b}, \mathbf{M}, \mathbf{z})$.
$\mathcal{P}_1$	Sample $\mathbf{A} \leftarrow U(\mathcal{R}_p^{m \times d})$, $\mathbf{x} \leftarrow \mathcal{D}_s$, $\mathbf{M} \leftarrow \mathcal{M}$, $\mathbf{y} \leftarrow \mathcal{D}_y$. Compute $\mathbf{b} = [\mathbf{A} \mathbf{I}_m] \mathbf{x} \bmod p\mathcal{R}$, $\mathbf{z} = \mathbf{y} + \mathbf{M}\mathbf{x}$, and output $(\mathbf{A}, \mathbf{b}, \mathbf{M}, \mathbf{z})$.

We then define the advantage $\text{Adv}_{\text{MH-M-LWE}}[\mathcal{A}]$ of PPT distinguisher $\mathcal{A}$ as $|\mathbb{P}_{(\mathbf{A}, \mathbf{b}, \mathbf{M}, \mathbf{z}) \sim \mathcal{P}_0}[\mathcal{A}(\mathbf{A}, \mathbf{b}, \mathbf{M}, \mathbf{z}) = 1] - \mathbb{P}_{(\mathbf{A}, \mathbf{b}, \mathbf{M}, \mathbf{z}) \sim \mathcal{P}_1}[\mathcal{A}(\mathbf{A}, \mathbf{b}, \mathbf{M}, \mathbf{z}) = 1]|$. We recover the standard version $\text{M-LWE}_{n,d,m,p,\mathcal{D}_s}$ when $\mathcal{D}_y = \{\mathbf{0}\}$ and $\mathcal{M} = \{\mathbf{0}\}$, as this leads to $\mathbf{M} = \mathbf{0}$ and $\mathbf{z} = \mathbf{0}$.

The authors of [ENP24, DFJS26] prove an efficient reduction from M-LWE to MH-M-LWE when the distributions $\mathcal{D}_s, \mathcal{D}_y$ are discrete Gaussians and where $\mathcal{M}$ is bounded for the spectral norm. Although they prove it for more general settings, it directly applies to the assumption of Definition 2.3, and we state the hardness result below.

Lemma 2.6 ([DFJS26, Thm. 3.1] adapted). *Let $\mathcal{R}$ be a power-of-two cyclotomic ring of degree n . Let d, m, p, k be in $\mathbb{N}^\times$, and $\mathcal{M}$ be a distribution over $\mathcal{R}^{k \times (m+d)}$ such that $\sup_{\mathbf{M} \in \text{Supp}(\mathcal{M})} \|\mathbf{M}\|_2 \leq \beta$. Let $\mathbf{S}, \mathbf{S}_y$ be positive definite in $\mathbb{R}^{n(d+m) \times n(d+m)}$ and $\mathbb{R}^{nk \times nk}$, such that $s' = (\|\mathbf{S}^{-1}\|_2 + \beta^2 \|\mathbf{S}_y^{-1}\|_2)^{-1/2} / \sqrt{2}$ satisfies $s' \geq \sqrt{2} \eta_\varepsilon(\mathcal{R}^{d+m})$ for some $\varepsilon \in (0, 1)$. There is a reduction from $\text{M-LWE}_{n,d,m,p,\mathcal{D}_{\mathcal{R}^{d+m},s'}}$ to $\text{MH-M-LWE}_{n,d,m,p,\mathcal{D}_{\mathcal{R}^{d+m},\sqrt{s}},\mathcal{D}_{\mathcal{R}^k,\sqrt{s_y}},\mathcal{M}}$ which shows $\varepsilon_{\text{MH-M-LWE}} \leq \varepsilon_{\text{M-LWE}} + \varepsilon/(1 - \varepsilon)$, for the respective hardness bounds.*

2.4 Group Signatures

We recall the syntax and security model of static group signatures, formalized in [BMW03]. A group signature scheme is composed of the following five algorithms:

- **GSetup**($1^\lambda, 1^N$). On input the security parameter λ and the group size⁴ N , it generates the public parameters, which contain a description of the set of identities $\mathcal{I}$ (of size N), which may differ from the set of indices $[N]$.
- **GKeyGen**(pp). On input pp , it generates the group public key gpk , the opening key gmsk and the set of user signing keys $\text{gsk} = (\text{gsk}[i])_{i \in [N]}$.
- **GSign**($\text{gsk}[i], m, \text{gpk}, \text{pp}$). On input the signing key of user i , a message m , the group public key gpk and the public parameters pp , it generates a group signature gsig .
- **GVerify**($\text{gpk}, m, \text{gsig}, \text{pp}$). On input the group public key gpk , a message m , a group signature gsig , and the public parameters pp , it returns 1 if gsig is valid and 0 otherwise.
- **GOpen**($\text{gmsk}, m, \text{gsig}, \text{gpk}, \text{pp}$). On input the opening key gmsk , a message m , a group signature gsig , and the public parameters, it returns the identity $\text{id} \in \mathcal{I}$ that produced gsig , or a failure symbol $\perp$.

A group signature scheme should be correct, that is a legitimately generated group signature should verify and trace back to the user identity that generated it. Beyond correctness, we expect a group signature to be traceable and anonymous. First, traceability ensures that an adversary, able to adaptively corrupt group members, cannot produce a group signature that makes the opening fail or that opens to an honest (i.e., uncorrupted) member. In addition to being able to corrupt users through an oracle $\mathcal{O}_{\text{Cor}}$ (which returns $\text{gsk}[i]$ on input $i \in [N]$), the adversary is also allowed to request group signatures from any member on any message using an oracle $\mathcal{O}_{\text{Sign}}$ (returning **GSign**($\text{gsk}[i], m, \text{gpk}, \text{pp}$) on input (i, m)). For each query, the game stores $(\text{id}, m, \text{gsig})$ in a set $\mathcal{S}$ where id is the identity associated to i , and gsig the returned signature. In the end, the adversary must produce a group signature gsig^* on a message m that verifies, opens to id corresponding to a non-existent ($\perp$) or uncorrupted member, and was not obtained in a prior query (i, m) .

⁴ The group size N may also be defined from other parameters and not given as input explicitly.

Definition 2.4. A group signature is ε_{tr} -traceable if $\sup_{\mathcal{A}} \text{ppT Adv}_{tr}[\mathcal{A}] \leq \varepsilon_{tr}$, $\text{Adv}_{tr}[\mathcal{A}] = \mathbb{P}[\text{Game}_{\mathcal{A}}^{tr}(1^\lambda, 1^N) = 1]$ being the advantage of $\mathcal{A}$ in the game of Figure 2.1.

```

Game $_{\mathcal{A}}^{tr}(1^\lambda, 1^N)$ :
1  $\mathcal{C}, \mathcal{S} \leftarrow \emptyset$ 
2  $\text{pp} \leftarrow \text{GSetup}(1^\lambda, 1^N)$ 
3  $(\text{gpk}, \text{gmsk}, \text{gsk}) \leftarrow \text{GKeyGen}(\text{pp})$ 
4  $(m, \text{gsig}^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{Cor}}, \mathcal{O}_{\text{Sign}}}(\text{pp}, \text{gpk}, \text{gmsk})$ 
5  $\text{id} \leftarrow \text{GOpen}(\text{gmsk}, m, \text{gsig}^*, \text{gpk}, \text{pp})$ 
7 return  $\llbracket \text{GVerify}(\text{gpk}, m, \text{gsig}^*, \text{pp}) = 1 \rrbracket \wedge \llbracket \text{id} \notin \mathcal{C} \rrbracket \wedge \llbracket (\text{id}, m, \text{gsig}^*) \notin \mathcal{S} \rrbracket$ 

```

Fig. 2.1. Traceability game

Then, anonymity ensures a group signature does not reveal the identity of the user that produced it, unless to the opener which is able to lift the anonymity of every signature. The strongest form of anonymity [BMW03] gives the adversary the ability to request the opening of any signature but the challenge one, through an oracle $\mathcal{O}_{\text{Open}}$. Weaker notions such as CPA-anonymity [BBS04] (where opening queries are not permitted) or selfless-anonymity [BS04] (where the adversary cannot corrupt the two challenge users) have been introduced to obtain more efficient schemes, but provide weaker anonymity guarantees.

Definition 2.5. A group signature is ε_{an} -anonymous if $\sup_{\mathcal{A}} \text{ppT Adv}_{an}[\mathcal{A}] \leq \varepsilon_{an}$, where $\text{Adv}_{an}[\mathcal{A}] = |\mathbb{P}[\text{Game}_{\mathcal{A}}^{an}(1^\lambda, 1^N) = 1] - 1/2|$ is the advantage of $\mathcal{A}$ in the game of Figure 2.2.

```

Game $_{\mathcal{A}}^{an}(1^\lambda, 1^N)$ :
1  $\text{pp} \leftarrow \text{GSetup}(1^\lambda, 1^N)$ 
2  $(\text{gpk}, \text{gmsk}, \text{gsk}) \leftarrow \text{GKeyGen}(\text{pp})$ 
3  $(\text{st}, i_0, i_1, m) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{Open}}}(\text{pp}, \text{gpk}, \text{gsk})$ 
4  $b \leftarrow U(\{0, 1\})$ 
5  $\text{gsig}^* \leftarrow \text{GSign}(\text{gsk}[i_b], m, \text{gpk}, \text{pp})$ 
6  $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\text{Open}}}(\text{st}, \text{pp}, \text{gpk}, \text{gsk}, \text{gsig}^*)$ 
7 if  $\llbracket i_0 = i_1 \rrbracket$  or  $(m, \text{gsig}^*)$  was queried to  $\mathcal{O}_{\text{Open}}$ , return 0
8 return  $\llbracket b = b' \rrbracket$ 

```

Fig. 2.2. Anonymity game. The variable st refers to a state $\mathcal{A}$ may keep for the second stage after having received the challenge.

3 Tag-Friendly NTRU Sampler

We first present our new tag-friendly sampler based on NTRU. Our sampler departs from [MP12] and its recent improvements [JS25b, JS26] by changing the structure of the matrix we use to hide the gadget component. At a high level, the latter works used an M-LWE instance $(\mathbf{A}, [\mathbf{I}|\mathbf{A}]\mathbf{R})$ to hide the gadget, whereas we use NTRU instances $(e_0 f_0^{-1}, \dots, e_{k-1} f_{k-1}^{-1})$. One of the upside is that we have fewer blocks in the trapdoor matrix as we no longer need the matrix $\mathbf{A}$. As opposed to iNTRU gadget samplers which do not interface well with tags, as described in Section 1.1, ours preserve the tag structure of the recent sampler of [JS26]. In particular, it allows for proving its security in the worst-case, as opposed to iNTRU-based samplers [GL20, YJW23], and for performing the trapdoor switching argument in the security proof of our group signature in a similar way as in the LWE-based framework of [JS26].

3.1 Description

We first give a high-level description of our sampler, which works over $\mathcal{R} = \mathbb{Z}[x]/\langle x^n + 1 \rangle$. We let q_L, b_H, k_H be in $\mathbb{N}^\times$, and we define $q_H = b_H^{k_H}$ and $q = q_L q_H$. Then, for any $\mathbf{t} = (t_0, \dots, t_{k_H-1}) \in (\mathcal{R}_{b_H}^\times)^{k_H}$, we define the tagged gadget $\mathbf{g}_{H,\mathbf{t}}^T = [t_0 | t_1 b_H | \dots | t_{k_H-1} b_H^{k_H-1}]$ in a similar way as in [JS26]. Then, for NTRU instances $\mathbf{h}^T = [h_0 | \dots | h_{k_H-1}]$ with h_i defined by $h_i = e_i f_i^{-1} \bmod q\mathcal{R}$, our sampler securely produces short Gaussian vectors $\mathbf{v}$ such that $\mathbf{a}_{\mathbf{t}}^T \mathbf{v} = u \bmod q\mathcal{R}$ for any $u \in \mathcal{R}_q$, where

$$\mathbf{a}_{\mathbf{t}}^T = [1 | q_L \mathbf{g}_{H,\mathbf{t}}^T - \mathbf{h}^T] \bmod q\mathcal{R}.$$

To describe how we sample such $\mathbf{v}$, let us first note that $\mathbf{h} = \mathbf{e} \odot \mathbf{f}$ where $\mathbf{e} = [e_0 | \dots | e_{k_H-1}]^T$ and $\mathbf{f} = [f_0 | \dots | f_{k_H-1}]^T$, with the division by $\mathbf{f}$ is performed modulo $q\mathcal{R}$. As a result, we get

$$\mathbf{a}_{\mathbf{t}}^T \underbrace{\begin{bmatrix} 1 & \mathbf{e}^T \\ \mathbf{0} & \text{diag}(\mathbf{f}) \end{bmatrix}}_{\mathbf{L}} = [1 | q_L \mathbf{g}_{H,\mathbf{t}}^T \odot \mathbf{f}] \bmod q\mathcal{R} = [1 | q_L \mathbf{g}_{H,\mathbf{t} \odot \mathbf{f}}^T].$$

This is then reminiscing of the tagged gadget of [JS26], albeit with $d = 1$, $\mathbf{g}_L = [1]$, and with effective tag $\mathbf{t} \odot \mathbf{f}$ instead of $\mathbf{t}$. Leveraging the smoothness of our modulus q , we can then resort to a similar “sampling-then-lifting” method. Naively, if we sample a short $\mathbf{z}$ such that $[1 | q_L \mathbf{g}_{H,\mathbf{t} \odot \mathbf{f}}^T] \mathbf{z} = u$, then $\mathbf{v} = \mathbf{L} \mathbf{z}$ is a short preimage of u for $\mathbf{a}_{\mathbf{t}}$. If we can do so, we can make it secure by sampling a perturbation $\mathbf{p}$ with a carefully-crafted covariance so that we instead invert $u' = u - \mathbf{a}_{\mathbf{t}}^T \mathbf{p}$ and output $\mathbf{v} = \mathbf{p} + \mathbf{L} \mathbf{z}$, which by convolution does not leak information on $\mathbf{t}$ nor the sampling key $(\mathbf{e}, \mathbf{f})$.

To sample $\mathbf{z}$ to be a preimage of u' for $[1 | q_L \mathbf{g}_{H,\mathbf{t} \odot \mathbf{f}}^T]$, we start by tackling the low component. For that, deterministic lattice decoding as in [YJW23] is insufficient as it thwarts the Gaussian convolution argument we need to prove

the sampler secure. Instead, we use randomized lattice decoding by sampling a Gaussian with the same low-order bits. Concretely, we define $\ell = u' \bmod q_L \mathcal{R}$, and sample z_L from $\mathcal{D}_{q_L \mathcal{R} + \ell, s_L}$. This already improves over [JS26] which works iteratively with a gadget $\mathbf{G}_L$ in base b_L (and $q_L = b_L^{k_L}$), rather than q_L directly. In our case, we observe that the optimal setting is to choose $b_L = q_L$ as it leads to less Gaussian sampling, as well as more compact parameters.

From z_L , we compute the lifting term $\ell_0 = (u' - z_L)/q_L \bmod q_H \mathcal{R}$ which belongs to $\mathcal{R}_{q_H}$. We now iteratively sample some $z_{H,0}, \dots, z_{H,k_H-1}$ as follows. We compute $c_i = f_i^{-1} t_i^{-1} \ell_i \bmod b_H \mathcal{R}$, and sample $z_{H,i}$ from $\mathcal{D}_{b_H \mathcal{R} + c_i, s_H}$. We then update ℓ_i to $\ell_{i+1} = (\ell_i - f_i t_i z_{H,i})/b_H$ which is in $\mathcal{R}$ based on how $z_{H,i}$ is sampled. In the end, we have $z_L + q_L f_0 t_0 z_{H,0} + \dots + q_L b_H^{k_H-1} f_{k_H-1} t_{k_H-1} z_{H,k_H-1} = u' - q_L b_H^{k_H} \ell_{k_H} = u' \bmod q \mathcal{R}$. Said differently, defining $\mathbf{z} = [z_L | z_{H,0} | \dots | z_{H,k_H-1}]^T$ gives $[1 | q_L \mathbf{g}_{H,\mathbf{t}}^T] \mathbf{z} = u' \bmod q \mathcal{R}$.

The remaining aspect is to identify the correct perturbation distribution so that $\mathbf{p} + \mathbf{Lz}$ does not leak secret information. By studying the spectral properties following the methodology of [JS25b, JS26], we are able to find tight parameters for our sampler to be secure. We now define formally our sampler **NTRU-TSampler** and prove it secure in the worst-case in Theorem 3.1.

Algorithm 3.1: NTRU-TSampler($\mathbf{e}, \mathbf{f}, u, \mathbf{t}, \mathbf{S}, s_L, s_H$)

Input: trapdoor vectors $\mathbf{e}, \mathbf{f} \in \mathcal{R}^{k_H}$ with $\mathbf{f} \bmod b_H \mathcal{R} \in (\mathcal{R}_{b_H}^\times)^{k_H}$, syndrome $u \in \mathcal{R}_q$, tag $\mathbf{t} \in (\mathcal{R}_{b_H}^\times)^{k_H}$, covariance matrices $\mathbf{S} \in \mathcal{S}_{n(1+k_H)}^{++}(\mathbb{R})$, $s_L, s_H > 0$.

1. $\mathbf{p} \leftarrow \mathcal{D}_{\mathcal{R}^{1+k_H}, \sqrt{\mathbf{S}_p}}$ with $\mathbf{S}_p = \mathbf{S} - M_\tau(\mathbf{L}) \text{diag}(s_L^2 \mathbf{I}_n, s_H^2 \mathbf{I}_{nk_H}) M_\tau(\mathbf{L})^T$
2. $u' \leftarrow u - [1 | q_L \mathbf{g}_{H,\mathbf{t}}^T - (\mathbf{e} \oslash \mathbf{f})^T] \mathbf{p} \bmod q \mathcal{R}$
3. $\ell \leftarrow u' \bmod q_L \mathcal{R}$
4. $z_L \leftarrow \mathcal{D}_{q_L \mathcal{R} + \ell, s_L}$
5. $\ell_0 \leftarrow (u' - z_L)/q_L \bmod q_H \mathcal{R}$
6. **For** $i = 0, \dots, k_H - 1$ **do**
7. $c_i \leftarrow f_i^{-1} t_i^{-1} \cdot \ell_i \bmod b_H \mathcal{R}$
8. $z_{H,i} \leftarrow \mathcal{D}_{b_H \mathcal{R} + c_i, s_H}$
9. $\ell_{i+1} \leftarrow (\ell_i - f_i t_i z_{H,i})/b_H \in \mathcal{R}$
10. $\mathbf{z} \leftarrow [z_L | z_{H,0} | \dots | z_{H,k_H-1}]^T$
11. $\mathbf{v} \leftarrow \mathbf{p} + \mathbf{Lz}$

$$\triangleright \mathbf{L} = \begin{bmatrix} 1 & \mathbf{e}^T \\ \mathbf{0} & \text{diag}(\mathbf{f}) \end{bmatrix}$$

Output: $\mathbf{v} \in \mathcal{L}_q^u(\mathbf{a}_t^T)$

3.2 Security of NTRU-TSampler

We now show that the output distribution of **NTRU-TSampler** is statistically close to $\mathcal{D}_{\mathcal{L}_q^u(\mathbf{a}_t^T), \sqrt{\mathbf{S}}}$ for any given $u \in \mathcal{R}_q$. In particular, it shows the preimages statistically hide the trapdoor elements $(\mathbf{e}, \mathbf{f})$ as well as the effective tag that is used in the sampling mechanism $(\mathbf{t} \oslash \mathbf{f})$.

Theorem 3.1. *Let $\mathcal{R}$ be a cyclotomic ring of degree n . Let q_L, b_H, k_H be in $\mathbb{N}^\times$ and define $q_H = b_H^{k_H}$ and $q = q_L q_H$. Let $\mathbf{S} \in \mathbb{R}^{n(1+k_H) \times n(1+k_H)}$ be positive definite, $s_L, s_H > 0$ and define $\mathbf{S}_g = \text{diag}(s_L^2 \mathbf{I}_n, s_H^2 \mathbf{I}_{nk_H})$. Let $\mathbf{e}, \mathbf{f}$ be in $\mathcal{R}^{k_H}$ such that $\mathbf{f} \bmod q \mathcal{R} \in (\mathcal{R}_q^\times)^{k_H}$, and $\mathbf{t} = [t_i]_{i \in [0, k_H-1]} \in (\mathcal{R}_{b_H}^\times)^{k_H}$.*

Then, define $\mathbf{g}_{H,\mathbf{t}} = [t_i b_H^i]_{i \in [0, k_H-1]}$. We denote by $\mathcal{P}_p$ the distribution outputted by a given choice of sampler in step 1 of **NTRU-TSampler**, and assume $\mathcal{P}_p \approx_\alpha \mathcal{D}_{\mathcal{R}^{1+k_H}, \sqrt{\mathbf{S}_p}}$. We finally denote by $\mathcal{P}$ the distribution of the outputs of **NTRU-TSampler**($\mathbf{e}, \mathbf{f}, u, \mathbf{t}, \mathbf{S}, s_L, s_H$) and assume the following conditions, $\mathbf{L}$ being defined from $\mathbf{e}, \mathbf{f}$ as in **NTRU-TSampler** and $\varepsilon, \varepsilon_L, \varepsilon_H$ being in $(0, 1)$.

- ① $s_L \geq q_L \eta_{\varepsilon_L}(\mathbb{Z}^n), s_L > \eta_\varepsilon(\mathbb{Z}^{n(1+k_H)})$
- ② $s_H \geq b_H \eta_{\varepsilon_L}(\mathbb{Z}^n), s_H > \eta_\varepsilon(\mathbb{Z}^{n(1+k_H)})$
- ③ $\mathbf{S} \geq M_\tau(\mathbf{L}) \text{diag} \left(\frac{s_L^4}{s_L^2 - \eta_\varepsilon(\mathbb{Z}^{n(1+k_H)})^2} \mathbf{I}_n, \frac{s_H^4}{s_H^2 - \eta_\varepsilon(\mathbb{Z}^{n(1+k_H)})^2} \mathbf{I}_{nk_H} \right) M_\tau(\mathbf{L})^T$

It then holds that $\text{Supp}(\mathcal{P}) = \mathcal{L}_q^u(\mathbf{a}_\mathbf{t}^T)$, and that $\mathcal{P} \approx_\delta \mathcal{D}_{\mathcal{L}_q^u(\mathbf{a}_\mathbf{t}^T), \sqrt{\mathbf{S}}}$ for

$$\delta = \alpha^2 \cdot \frac{1+\varepsilon}{1-\varepsilon} \cdot \frac{1+\varepsilon_L}{1-\varepsilon_L} \cdot \left(\frac{1+\varepsilon_H}{1-\varepsilon_H} \right)^{k_H},$$

where $\mathbf{a}_\mathbf{t}^T = [1|q_L \mathbf{g}_{H,\mathbf{t}}^T - (\mathbf{e} \oslash \mathbf{f})^T] \bmod q\mathcal{R}$.

Proof. We follow the main steps of the proof of [JS26], but tackle differently the several subtleties that arise due to our different trapdoor structure. In particular, using NTRU instances rather than LWE ones requires the effective tagged gadget to feature $\mathbf{f}$, and our matrix $\mathbf{L}$ is actually invertible which avoids the peculiarities of pseudoinverses. Note that $\mathbf{f} \bmod q\mathcal{R} \in (\mathcal{R}_q^\times)^{k_H}$ implies $\mathbf{f} \bmod b_H \mathcal{R} \in (\mathcal{R}_{b_H}^\times)^{k_H}$ as needed by **NTRU-TSampler**.

Correctness. We start by showing that the support is indeed the whole lattice coset in Lemma 3.1.

Lemma 3.1. *It holds that $\text{Supp}(\mathcal{P}) = \mathcal{L}_q^u(\mathbf{a}_\mathbf{t}^T)$*

Proof (Lemma 3.1). Let $\mathbf{v} = \mathbf{p} + \mathbf{L}\mathbf{z}$ be an output of the sampler. We start by observing that

$$\mathbf{a}_\mathbf{t}^T \mathbf{L} = [1|q_L \mathbf{g}_{H,\mathbf{t}}^T \odot \mathbf{f}^T] \bmod q\mathcal{R} = [1|q_L \mathbf{g}_{H,\mathbf{t} \odot \mathbf{f}}^T] \bmod q\mathcal{R}$$

It then yields $\mathbf{a}_\mathbf{t}^T \mathbf{v} = \mathbf{a}_\mathbf{t}^T \mathbf{p} + [1|q_L \mathbf{g}_{H,\mathbf{t} \odot \mathbf{f}}^T] \mathbf{z} \bmod q\mathcal{R}$. We now focus on the second term. First, we note that ℓ_0 is well defined in $\mathcal{R}_{q_H}$. Indeed, we have that $z_L \in \ell + q_L \mathcal{R}$ by the way z_L is sampled, and $u' \in \ell + q_L \mathcal{R}$ by simple Euclidean division. Hence, $u' - z_L \in q_L \mathcal{R}$. We check similarly by induction that all the subsequent ℓ_i are well-defined in $\mathcal{R}$. Because $f_i t_i z_{H,i} \bmod b_H \mathcal{R} = f_i t_i c_i = \ell_i$, it holds $f_i t_i z_{H,i} \in \ell_i + b_H \mathcal{R}$. Hence, $\ell_i - f_i t_i z_{H,i} \in b_H \mathcal{R}$ as desired. Then, we can easily see that

$$\begin{aligned} \ell_0 &= f_0 t_0 z_{H,0} + b_H \ell_1 = f_0 t_0 z_{H,0} + b_H f_1 t_1 z_{H,1} + b_H^2 \ell_2 \\ &= \dots \\ &= \sum_{j=0}^{k_H-1} b_H^j f_j t_j z_{H,j} + b_H^{k_H} \ell_{k_H}. \end{aligned}$$

This proves $\mathbf{g}_{H,t \odot \mathbf{f}}^T \mathbf{z}_H = \ell_0 - q_H \ell_{k_H}$. Multiplying by q_L , adding z_L , and reducing modulo $q = q_L q_H$ then gives $[1|q_L \mathbf{g}_{H,t \odot \mathbf{f}}^T] \mathbf{z} = u' \bmod q\mathcal{R}$, and thus $\mathbf{a}_t^T \mathbf{v} = u \bmod q\mathcal{R}$. We have then shown $\text{Supp}(\mathcal{P}) \subseteq \mathcal{L}_q^u(\mathbf{a}_t^T)$.

Reciprocally, let $\mathbf{v} \in \mathcal{L}_q^u(\mathbf{a}_t^T)$. Let $\mathbf{z} \in \mathcal{R}^{1+k_H}$ and define $\mathbf{p} = \mathbf{v} - \mathbf{L}\mathbf{z}$. Then $\mathbf{p} \in \mathcal{R}^{1+k_H}$ because $\mathbf{L}$ has coefficients in $\mathcal{R}$ and $\mathbf{x}$ does too. Then, $\mathbf{a}_t^T \mathbf{L}\mathbf{z} = \mathbf{a}_t^T \mathbf{v} - \mathbf{a}_t^T \mathbf{p} = u - \mathbf{a}_t^T \mathbf{p} \bmod q\mathcal{R}$. As $\mathbf{a}_t^T \mathbf{L} = [1|q_L \mathbf{g}_{H,t \odot \mathbf{f}}^T] \bmod q\mathcal{R}$, we deduce

$$z_L + q_L \mathbf{g}_{H,t \odot \mathbf{f}} \mathbf{z}_H = u - \mathbf{a}_t^T \mathbf{p} \bmod q\mathcal{R}$$

We then have $z_L \bmod q_L \mathcal{R} = u' \bmod q_L \mathcal{R} = \ell$. Hence, $z_L \in \ell + q_L \mathcal{R}$ as requested. Then, we have $\mathbf{g}_{H,t \odot \mathbf{f}}^T \mathbf{z}_H = (u' - z_L)/q_L = \ell_0$. Following the iterative process, we have that each $z_{H,i}$ is in the proper coset of $b_H \mathcal{R}$. Because $\mathbf{p}, \mathbf{z}_L, \mathbf{z}_H$ are in the support of their respective distributions, it holds $\mathbf{v}$ is in the support of $\mathcal{P}$. $\square$

Smoothing Condition. We now compute the relevant covariance elements and show they satisfy the proper smoothing conditions we need to prove the distribution $\mathcal{P}$ is close to the desired Gaussian. We denote by η the quantity $\eta_\varepsilon(\mathbb{Z}^{n(1+k_H)})$. We first define

$$\mathbf{M} = M_\tau(\mathbf{L}) = \begin{bmatrix} \mathbf{I}_n & M_\tau(e_0) & \dots & M_\tau(e_{k_H-1}) \\ & M_\tau(f_0) & & \\ \mathbf{0} & & \ddots & \\ & & & M_\tau(f_{k_H-1}) \end{bmatrix}.$$

Because all the f_i are invertible modulo b_H , it implies that $\det(M_\tau(f_i)) \bmod b_H \mathbb{Z} \neq 0$, and therefore $\det(M_\tau(f_i)) \neq 0$. Hence all the f_i are units in $\mathcal{K}$ which most importantly entails that $\mathbf{M}$ is invertible.

We then define the lattice $\mathcal{L} = \mathcal{L}(\mathbf{M}) = \mathbf{M}\mathbb{Z}^{n(1+k_H)}$, as well as $\mathbf{S}_y = \mathbf{M}\mathbf{S}_g\mathbf{M}^T$ and $\mathbf{S}_p = \mathbf{S} - \mathbf{S}_y$. Because $\mathbf{S}_g$ is invertible, it then holds that $\mathbf{S}_y$ is too. By conditions ① and ②, the diagonal matrix of condition ③ is well-defined. In particular, as $s_x^4/(s_x^2 - \eta^2) > s_x^2$, this diagonal matrix, call it $\mathbf{D}$, satisfies $\mathbf{D} > \mathbf{S}_g$ for the Loewner order. Because $\mathbf{M}$ is invertible, we then have $\mathbf{MDM}^T > \mathbf{MS}_g\mathbf{M}^T$ preserving the strict inequality. Hence, condition ③ shows $\mathbf{S} \geq \mathbf{MDM}^T > \mathbf{MS}_g\mathbf{M}^T$ which means $\mathbf{S}_p = \mathbf{S} - \mathbf{MS}_g\mathbf{M}^T$ is positive definite and thus invertible. We can then define $\mathbf{S}_3 = (\mathbf{S}_p^{-1} + \mathbf{S}_y^{-1})^{-1}$, and rewrite it as $\mathbf{S}_3 = \mathbf{M}(\mathbf{M}^T \mathbf{S}_p^{-1} \mathbf{M} + \mathbf{S}_g^{-1})^{-1} \mathbf{M}^T$. We now show that $\mathbf{S}_3$ satisfies the following smoothing condition.

Lemma 3.2. *It holds that $\sqrt{\mathbf{S}_3} \geq \eta_\varepsilon(\mathcal{L})$.*

Proof (Lemma 3.2). We observe the matrix $\mathbf{D}$ defined above and appearing in condition ③ can be re-expressed as $\mathbf{D} = \mathbf{S}_g + (\eta^{-2} \mathbf{I}_{n(1+k_H)} - \mathbf{S}_g^{-1})^{-1}$. As a result, the latter condition can then be written as

$$\mathbf{S}_p \geq \mathbf{M}(\eta^{-2} \mathbf{I}_{n(1+k_H)} - \mathbf{S}_g^{-1})^{-1} \mathbf{M}^T.$$

The compatibility of the inverse with the Loewner order then gives $\mathbf{S}_p^{-1} \leq \mathbf{M}^{-T}(\eta^{-2}\mathbf{I}_{n(1+k_H)} - \mathbf{S}_g^{-1})\mathbf{M}^{-1}$, and therefore $\mathbf{M}^T\mathbf{S}_p^{-1}\mathbf{M} + \mathbf{S}_g^{-1} \leq \eta^{-2}\mathbf{I}_{n(1+k_H)}$. Using the inverse again and multiplying by $\mathbf{M}$ and $\mathbf{M}^T$ then gives $\mathbf{S}_3 \geq \eta^2\mathbf{M}\mathbf{M}^T$. We then have

$$\begin{aligned}
1 + \varepsilon &= \sum_{\mathbf{y} \in \mathbb{Z}^{n(1+k_H)}} \exp(-\pi\eta^2\mathbf{y}^T\mathbf{y}) \\
&= \sum_{\mathbf{y} \in \mathbb{Z}^{n(1+k_H)}} \exp(-\pi\eta^2\mathbf{y}^T\mathbf{M}^{-1}\mathbf{M}\mathbf{M}^T\mathbf{M}^{-T}\mathbf{y}) \\
&= \sum_{\mathbf{y} \in \mathcal{L}^\vee} \exp(-\pi\eta^2\mathbf{y}^T\mathbf{M}\mathbf{M}^T\mathbf{y}) \quad (\text{as } \mathcal{L}^\vee = \mathbf{M}^{-T}\mathbb{Z}^{n(1+k_H)}) \\
&\geq \sum_{\mathbf{y} \in \mathcal{L}^\vee} \exp(-\pi\mathbf{y}^T\mathbf{S}_3\mathbf{y}),
\end{aligned}$$

which then proves $\rho_{\sqrt{\mathbf{S}_3^{-1}}}(\mathcal{L}^\vee) \leq 1 + \varepsilon$ thus giving $\sqrt{\mathbf{S}_3} \geq \eta_\varepsilon(\mathcal{L})$. $\square$

Distribution Analysis. We now study the distribution $\mathcal{P}$ of **NTRU-TSampler**. As of Lemma 3.1, we only know that the support of $\mathcal{P}$ matches that of its ideal distribution. Let $\mathbf{v}$ be an output of **NTRU-TSampler**. There then exists $\mathbf{p} \in \mathcal{R}^{1+k_H}$, z_L in the right coset of $q_L\mathcal{R}$ and some $z_{H,0}, \dots, z_{H,k_H-1}$ in cosets of $b_H\mathcal{R}$ such that $\mathbf{v} = \mathbf{p} + \mathbf{L}\mathbf{z}$ for $\mathbf{z} = [z_L | z_{H,0} | \dots | z_{H,k_H-1}]^T$. The cosets that the entries of $\mathbf{z}$ belong to depend on previously sampled elements (including $\mathbf{p}$) due to our iterative sampling. Note that because $\mathbf{L}$ is invertible (as $\mathbf{M} = M_\tau(\mathbf{L})$ is), $\mathbf{z}$ is entirely determined by $\mathbf{p}$ and $\mathbf{v}$ as $\mathbf{z} = \mathbf{L}^{-1}(\mathbf{v} - \mathbf{p})$. Also, for a given $\mathbf{v}$, $\mathbf{p} \in \mathbf{v} + \mathbf{L}\mathcal{R}^{1+k_H}$ and therefore $\tau(\mathbf{p}) \in \tau(\mathbf{v}) + \mathcal{L}$. For conciseness, we abuse notation and write $\mathbf{p} \in \mathbf{v} + \mathcal{L}$ but this should be interpreted through the coefficient embedding.

As a result, by the chain rule of conditional probabilities, we have

$$\mathcal{P}(\mathbf{v}) = \sum_{\mathbf{p} \in \mathbf{v} + \mathcal{L}} \mathcal{P}_p(\mathbf{p}) \mathbb{P}[z_L | \mathbf{p}] \mathbb{P}[z_{H,0} | \mathbf{p}, z_L] \dots \mathbb{P}[z_{H,k_H-1} | \mathbf{p}, z_L, z_{H,0}, \dots, z_{H,k_H-2}].$$

The z terms can be expressed as

$$\begin{aligned}
&\mathbb{P}[z_L | \mathbf{p}] \mathbb{P}[z_{H,0} | \mathbf{p}, z_L] \dots \mathbb{P}[z_{H,k_H-1} | \mathbf{p}, z_L, z_{H,0}, \dots, z_{H,k_H-2}] \\
&= \frac{\rho_{s_L}(z_L)}{\rho_{s_L}(q_L\mathcal{R} + \ell)} \cdot \prod_{i=0}^{k_H-1} \frac{\rho_{s_H}(z_{H,i})}{\rho_{s_H}(b_H\mathcal{R} + c_i)} \\
&= \frac{\rho_{s_L}(z_L) \rho_{s_H}(\mathbf{z}_H)}{\rho_{s_L}(q_L\mathcal{R} + \ell) \prod_{i=0}^{k_H-1} \rho_{s_H}(b_H\mathcal{R} + c_i)},
\end{aligned}$$

where the cosets ℓ, c_i depend on $\mathbf{p}$ and prior elements as in **NTRU-TSampler**. We denote the denominator by $D(\mathbf{p})$ to clarify its dependency in $\mathbf{p}$. We can also express the numerator as $\rho_{\sqrt{\mathbf{S}_g}}(\mathbf{z})$ which can be re-written as follows.

$$\rho_{\sqrt{\mathbf{S}_g}}(\mathbf{z}) = \exp(-\pi\tau(\mathbf{z})^T\mathbf{S}_g^{-1}\tau(\mathbf{z}))$$

$$\begin{aligned}
&= \exp(-\pi \tau(\mathbf{z})^T \mathbf{M}^T \mathbf{S}_y^{-1} \mathbf{M} \tau(\mathbf{z})) \\
&= \exp(-\pi \tau(\mathbf{Lz})^T \mathbf{S}_y^{-1} \tau(\mathbf{Lz})) \\
&= \exp(-\pi \tau(\mathbf{v} - \mathbf{p})^T \mathbf{S}_y^{-1} \tau(\mathbf{v} - \mathbf{p})) \\
&= \rho_{\sqrt{\mathbf{S}_y}}(\mathbf{v} - \mathbf{p}).
\end{aligned}$$

We then have

$$\mathcal{P}(\mathbf{v}) = \sum_{\mathbf{p} \in \mathbf{v} + \mathcal{L}} \mathcal{P}_p(\mathbf{p}) \frac{\rho_{\sqrt{\mathbf{S}_y}}(\mathbf{p} - \mathbf{v})}{D(\mathbf{p})} \in \left[\frac{1}{\alpha}, \alpha \right] \sum_{\mathbf{p} \in \mathbf{v} + \mathcal{L}} \frac{\rho_{\sqrt{\mathbf{S}_p}}(\mathbf{p}) \rho_{\sqrt{\mathbf{S}_y}}(\mathbf{p} - \mathbf{v})}{\rho_{\sqrt{\mathbf{S}_p}}(\mathcal{R}^{1+k_H}) D(\mathbf{p})}.$$

We can now use the identity proven in Appendix A of the full version of [JS25b], which fixes [MP12, Fact 5.6]. However, because we are in the more favorable case where $\mathbf{S}_p, \mathbf{S}_y$ are positive *definite* and thus invertible, we can directly rely on the same result but for non-singular covariances proven in [Pei10, Fact 2.1]. As opposed to [MP12, Fact 5.6] that oversaw some of the peculiarities of pseudoinverses which led to a mistaken statement, the non-singular case [Pei10, Fact 2.1] is much simpler and correctly stated and proven. The latter then gives

$$\rho_{\sqrt{\mathbf{S}_p}}(\mathbf{p}) \rho_{\sqrt{\mathbf{S}_y}}(\mathbf{p} - \mathbf{v}) = \rho_{\sqrt{\mathbf{S}}}(\mathbf{v}) \rho_{\sqrt{\mathbf{S}_3}}(\mathbf{p} - \mathbf{c}).$$

where $\mathbf{c} = \tau^{-1}(\mathbf{S}_3 \mathbf{S}_y^{-1} \tau(\mathbf{v}))$. Going back to our computation of $\mathcal{P}(\mathbf{v})$, we have

$$\begin{aligned}
\mathcal{P}(\mathbf{v}) &\in \left[\frac{1}{\alpha}, \alpha \right] \sum_{\mathbf{p} \in \mathbf{v} + \mathcal{L}} \frac{\rho_{\sqrt{\mathbf{S}}}(\mathbf{v}) \rho_{\sqrt{\mathbf{S}_3}}(\mathbf{p} - \mathbf{c})}{\rho_{\sqrt{\mathbf{S}_p}}(\mathcal{R}^{1+k_H}) D(\mathbf{p})} \\
&\in \left[\frac{1}{\alpha}, \alpha \right] \mathcal{D}_{\mathcal{L}_q^u(\mathbf{a}_t^T), \sqrt{\mathbf{S}}}(\mathbf{v}) \frac{\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}_q^u(\mathbf{a}_t^T))}{\rho_{\sqrt{\mathbf{S}_p}}(\mathcal{R}^{1+k_H})} \sum_{\mathbf{p} \in \mathbf{v} + \mathcal{L}} \frac{\rho_{\sqrt{\mathbf{S}_3}}(\mathbf{p} - \mathbf{c})}{D(\mathbf{p})}
\end{aligned}$$

We now smooth out the cosets from $D(\mathbf{p})$ to simplify the sum. Using condition ①, we have $s_L \geq q_L \eta_{\varepsilon_L}(\mathbb{Z}^n) = \eta_{\varepsilon_L}(q_L \mathcal{R})$. Hence, Lemma 2.4 gives $\rho_{s_L}(q_L \mathcal{R} + \ell) \in [\frac{1-\varepsilon_L}{1+\varepsilon_L}, 1] \rho_{s_L}(q_L \mathcal{R})$. Similarly, condition ② and Lemma 2.4 combined give $\rho_{s_H}(b_H \mathcal{R} + c_i) \in [\frac{1-\varepsilon_H}{1+\varepsilon_H}, 1] \rho_{s_H}(b_H \mathcal{R})$ as $s_H \geq b_H \eta_{\varepsilon_H}(\mathbb{Z}^n) = \eta_{\varepsilon_H}(b_H \mathcal{R})$. We then get

$$D(\mathbf{p}) \in \left[\frac{1-\varepsilon_L}{1+\varepsilon_L} \left(\frac{1-\varepsilon_H}{1+\varepsilon_H} \right)^{k_H}, 1 \right] \rho_{s_L}(q_L \mathcal{R}) \rho_{s_H}(b_H \mathcal{R})^{k_H},$$

which leads to

$$\mathcal{P}(\mathbf{v}) \in \left[\frac{1}{\alpha}, \alpha' \right] \mathcal{D}_{\mathcal{L}_q^u(\mathbf{a}_t^T), \sqrt{\mathbf{S}}}(\mathbf{v}) \frac{\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}_q^u(\mathbf{a}_t^T)) \rho_{\sqrt{\mathbf{S}_3}}(\mathcal{L} + \mathbf{v} - \mathbf{c})}{\rho_{\sqrt{\mathbf{S}_p}}(\mathcal{R}^{1+k_H}) \rho_{s_L}(q_L \mathcal{R}) \rho_{s_H}(b_H \mathcal{R})^{k_H}},$$

for $\alpha' = \alpha \frac{1+\varepsilon_L}{1-\varepsilon_L} \left(\frac{1+\varepsilon_H}{1-\varepsilon_H} \right)^{k_H}$. Using Lemma 3.2 and Lemma 2.4, we can finally smooth out the shift $\mathbf{v} - \mathbf{c}$ in the numerator. It gives

$$\mathcal{P}(\mathbf{v}) \in [a, b] \mathcal{D}_{\mathcal{L}_q^u(\mathbf{a}_t^T), \sqrt{\mathbf{S}}}(\mathbf{v}) \underbrace{\frac{\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}_q^u(\mathbf{a}_t^T)) \rho_{\sqrt{\mathbf{S}_3}}(\mathcal{L})}{\rho_{\sqrt{\mathbf{S}_p}}(\mathcal{R}^{1+k_H}) \rho_{s_L}(q_L \mathcal{R}) \rho_{s_H}(b_H \mathcal{R})^{k_H}}}_c,$$

where $a = \frac{1}{\alpha} \cdot \frac{1-\varepsilon}{1+\varepsilon}$, $b = \alpha'$ and c is a constant independent of $\mathbf{v}$. Summing over the entire support of $\mathcal{P}$, that is $\mathcal{L}_q^u(\mathbf{a}_t^T)$ by Lemma 3.1, gives $ac \leq 1 \leq bc$ and thus $c \in [b^{-1}, a^{-1}]$. Hence, $\mathcal{P} \approx_{b/a} \mathcal{D}_{\mathcal{L}_q^u(\mathbf{a}_t^T), \sqrt{\mathbf{S}}}$. We conclude by expressing b/a as

$$\delta := \frac{b}{a} = \alpha^2 \cdot \frac{1+\varepsilon}{1-\varepsilon} \cdot \frac{1+\varepsilon_L}{1-\varepsilon_L} \cdot \left(\frac{1+\varepsilon_H}{1-\varepsilon_H} \right)^{k_H},$$

to get the desired claim. $\square$

We can now give a concrete choice of $\mathbf{S}$ so as to tightly match condition (3). Applying the following lemma with $a_x = s_x^4 / (s_x^2 - \eta_\varepsilon(\mathbb{Z}^{n(1+k_H)})^2)$ then implies condition (3).

Lemma 3.3. *Let $\mathcal{R}$ be a cyclotomic ring of degree n . Let k_H be in $\mathbb{N}^\times$, and $a_L, a_H, s_1, s_2 > 0$ to define $\mathbf{S} = \text{diag}(s_1^2 \mathbf{I}_n, s_2^2 \mathbf{I}_{nk_H})$. Let $\mathbf{e}, \mathbf{f} = [f_0 | \dots | f_{k_H-1}]^T$ be in $\mathcal{R}^{k_H}$. We let $\mathbf{L} = \begin{bmatrix} 1 & \mathbf{e}^T \\ \mathbf{0} & \text{diag}(\mathbf{f}) \end{bmatrix}$. For $i \in [0, k_H - 1]$, we use the singular value decomposition $M_\tau(f_i) = \mathbf{U}_i \mathbf{D}_i \mathbf{V}_i^T$ and define $\mathbf{V} = \text{diag}(\mathbf{V}_0, \dots, \mathbf{V}_{k_H})$. Then, $\mathbf{S} \geq M_\tau(\mathbf{L}) \text{diag}(a_L \mathbf{I}_n, a_H \mathbf{I}_{nk_H}) M_\tau(\mathbf{L})^T$ if and only if*

$$s_1^2 \geq a_L + a_H \|M_\tau(\mathbf{e}^T) \mathbf{V} \mathbf{D} \mathbf{V}^T M_\tau(\mathbf{e}^T)^T\|_2, \text{ and } s_2^2 > a_H \max_{0 \leq i < k_H} \|M_\tau(f_i)\|_2^2,$$

where, for $\mathfrak{s}_j(\cdot)$ denoting the j -th singular value, we have

$$\mathbf{D} = \text{diag} \left(\text{diag} \left(\frac{s_2^2}{s_2^2 - a_H \mathfrak{s}_j(M_\tau(f_0))^2} \right), \dots, \text{diag} \left(\frac{s_2^2}{s_2^2 - a_H \mathfrak{s}_j(M_\tau(f_{k_H-1}))^2} \right) \right).$$

A sufficient setting is to choose, for any $\gamma > 1$,

$$s_1 = \sqrt{a_L + a_H \frac{\gamma^2}{\gamma^2 - 1} \|\sigma(\mathbf{e}^* \mathbf{e})\|_\infty}, \text{ and } s_2 = \gamma \sqrt{a_H} \max_{0 \leq i < k_H} \|\sigma(f_i)\|_\infty,$$

Proof. We start by defining $\mathbf{E}^T = M_\tau(\mathbf{e}^T) = [M_\tau(e_0) | \dots | M_\tau(e_{k_H-1})]$, and $\mathbf{F} = M_\tau(\text{diag}(\mathbf{f})) = \text{diag}(M_\tau(f_0), \dots, M_\tau(f_{k_H-1}))$. This simplifies the expression of $M_\tau(\mathbf{L})$, and leads to $\mathbf{S}' = \mathbf{S} - M_\tau(\mathbf{L}) \text{diag}(a_L \mathbf{I}_n, a_H \mathbf{I}_{nk_H}) M_\tau(\mathbf{L})^T$ being equal to

$$\begin{bmatrix} \text{A} & \text{B} \\ (s_1^2 - a_L) \mathbf{I}_n - a_H \mathbf{E}^T \mathbf{E} & -a_H \mathbf{E}^T \mathbf{F}^T \\ -a_H \mathbf{F} \mathbf{E} & s_2^2 \mathbf{I}_{nk_H} - a_H \mathbf{F} \mathbf{F}^T \\ \text{B}^T & \text{C} \end{bmatrix}$$

It then holds that $\mathbf{S}' \geq 0$ if and only if $\mathbf{C} > 0$ and $\mathbf{S}'/\mathbf{C} = \mathbf{A} - \mathbf{B} \mathbf{C}^{-1} \mathbf{B}^T \geq 0$. For i in $[0, k_H - 1]$, we let $\mathbf{U}_i, \mathbf{V}_i$ be in $O_n(\mathbb{R})$ and $\mathbf{D}_i = \text{diag}(\mathfrak{s}_{1,i}, \dots, \mathfrak{s}_{n,i})$ be such that $M_\tau(f_i) = \mathbf{U}_i \mathbf{D}_i \mathbf{V}_i^T$, with $\mathfrak{s}_{1,i} \geq \dots \geq \mathfrak{s}_{n,i} > 0$ the singular values of $M_\tau(f_i)$ (which are positive as $M_\tau(f_i)$ is invertible). Writing $\mathbf{U} = \text{diag}(\mathbf{U}_0, \dots, \mathbf{U}_{k_H-1})$, $\mathbf{V} = \text{diag}(\mathbf{V}_0, \dots, \mathbf{V}_{k_H-1})$ and $\mathbf{D}' = \text{diag}(\mathbf{D}_0, \dots, \mathbf{D}_{k_H-1})$, we then have $\mathbf{F} = \mathbf{U} \mathbf{D}' \mathbf{V}^T$.

Hence $\mathbf{C} = \mathbf{U}(s_2^2 \mathbf{I}_{nk_H} - a_H \mathbf{D}'^2) \mathbf{U}^T$, which means $\mathbf{C} > 0$ if and only if $s_2^2 > a_H \max_i \mathfrak{s}_{1,i}^2$. We also have $\mathbf{C}^{-1} = \mathbf{U}(s_2^2 \mathbf{I}_{nk_H} - a_H \mathbf{D}'^2)^{-1} \mathbf{U}^T$. Then, it yields $\mathbf{B} \mathbf{C}^{-1} \mathbf{B} = a_H^2 \mathbf{E}^T \mathbf{V} \mathbf{D}'^2 (s_2^2 \mathbf{I}_{nk_H} - a_H \mathbf{D}'^2)^{-1} \mathbf{V}^T \mathbf{E}$. In the end, we have

$$\mathbf{S}'/\mathbf{C} = (s_1^2 - a_L) \mathbf{I}_n - a_H \mathbf{E}^T \mathbf{V} \underbrace{(\mathbf{I}_{nk_H} + a_H \mathbf{D}'^2 (s_2^2 \mathbf{I}_{nk_H} - a_H \mathbf{D}'^2)^{-1})}_{\mathbf{D}} \mathbf{V}^T \mathbf{E}.$$

By simplifying the expression of $\mathbf{D}$, we get that of the lemma statement. Then, $\mathbf{S}'/\mathbf{C} \geq 0$ is equivalent to $s_1^2 - a_L \geq a_H \|\mathbf{E}^T \mathbf{V} \mathbf{D} \mathbf{V}^T \mathbf{E}\|_2$ as claimed.

Let us now look at the sufficient condition. First, because $\|\mathbf{E}^T \mathbf{V} \mathbf{D} \mathbf{V}^T \mathbf{E}\|_2 \leq \|\mathbf{E}\|_2^2 \|\mathbf{D}\|_2$, and $\|\mathbf{D}\|_2 = \max_{i,j} s_2^2 / (s_2^2 - a_H \mathfrak{s}_{j,i}^2) = s_2^2 / (s_2^2 - a_H \max_i \mathfrak{s}_{1,i}^2)$, it holds that the condition on s_1 is implied by the simpler condition

$$s_1^2 \geq a_L + a_H \frac{s_2^2}{s_2^2 - a_H \max_i \mathfrak{s}_{1,i}^2} \|\mathbf{E}\|_2^2.$$

We observe that $\|M_\tau(f_i)\|_2 = \|\sigma(f_i)\|_\infty$ by [BJRW23, Lem. 2.3] where σ is the canonical embedding of $\mathcal{R}$. We also have that $\|\mathbf{E}\|_2^2 = \lambda_{\max}(M_\tau(\mathbf{e}^T) M_\tau(\mathbf{e}^T)^T) = \lambda_{\max}(M_\tau(\mathbf{e}^* \mathbf{e})) = \|M_\tau(\mathbf{e}^* \mathbf{e})\|_2$, which by [BJRW23, Lem. 2.3] again is equal to $\|\sigma(\mathbf{e}^* \mathbf{e})\|_\infty$. Then, let $\gamma > 1$, and set s_1, s_2 as in the statement. It thus holds that $s_2^2 = \gamma^2 a_H \max_i \mathfrak{s}_{1,i}^2 > a_H \max_i \mathfrak{s}_{1,i}^2$. We also have that $s_2^2 / (s_2^2 - a_H \max_i \mathfrak{s}_{1,i}^2) = \gamma^2 / (\gamma^2 - 1)$. It thus means that $s_1^2 = a_L + a_H \frac{s_2^2}{s_2^2 - a_H \max_i \mathfrak{s}_{1,i}^2} \|\mathbf{E}\|_2^2$, which thus satisfies the simpler condition above. $\square$

3.3 Perturbation Sampler

We now propose a perturbation sampler for the first step of **NTRU-TSampler**, thus instantiating $\mathcal{P}_p$ from Theorem 3.1. For that we follow the approach from prior implementation works [AGJ⁺24, JS25b, JS26] by using an iterative Schur complements sampler with precomputations. Consider the following structured covariance matrix, i.e., over $\mathcal{K}_{\mathbb{R}}$.

$$\mathbf{S}_p = \begin{bmatrix} s_1^2 - s_L^2 - s_H^2 \mathbf{e}^* \mathbf{e} & -s_H^2 \mathbf{e}^T \odot \mathbf{f}^* \\ -s_H^2 \mathbf{f} \odot \mathbf{e}^{T*} & s_2^2 \mathbf{I}_{k_H} - s_H^2 \text{diag}(\mathbf{f} \odot \mathbf{f}^{T*}) \end{bmatrix}.$$

Observe that the bottom right block is equal to $\text{diag}(s_2^2 - s_H^2 f_0 f_0^*, \dots, s_2^2 - s_H^2 f_{k_H-1} f_{k_H-1}^*)$. All these elements are in $\mathcal{K}_{\mathbb{R}}^{++}$, which means the associated components of the perturbation can be sampled using the **SampleFz** algorithm from [GM18]. The covariance elements, namely the $g_i = s_2^2 - s_H^2 f_i f_i^*$ can be precomputed at key generation. We thus sample the perturbation $\mathbf{p}_2 \in \mathcal{R}^{k_H}$ using k_H calls to **SampleFz**. We can then compute the Schur complement $s_p = \mathbf{S}_p / (s_2^2 \mathbf{I}_{k_H} - s_H^2 \text{diag}(\mathbf{f} \odot \mathbf{f}^{T*}))$ and new center c as

$$s_p = s_1^2 - s_L^2 - s_H^2 s_2^2 \sum_{i=0}^{k_H-1} e_i e_i^* (s_2^2 - s_H^2 f_i f_i^*)^{-1} \quad \text{and} \quad c = \mathbf{f}'^T \mathbf{p}_2$$

with $\mathbf{f}' = -s_H^2 [e_i f_i^* (s_2^2 - s_H^2 f_i f_i^*)^{-1}]_{i \in [0, k_H-1]}$.

Note that s_p is a single element of $\mathcal{K}_{\mathbb{R}}^{++}$, and that both s_p and $\mathbf{f}'$ can be pre-computed at key generation. We then need to sample from $\mathcal{D}_{\mathcal{R}, \sqrt{M_{\tau}(s_p)}, c}$, which can be done in one call to **SampleFz**. We then obtain the following perturbation sampler which maximizes precomputations.

Algorithm 3.2: PerturbationSampler(e, f)

Precomputed: $(g_i = s_2^2 - s_H^2 f_i f_i^*)_{i \in [0, k_H-1]}$, $\mathbf{f}' = [-s_H^2 e_i f_i^* (s_2^2 - s_H^2 f_i f_i^*)^{-1}]_{i \in [0, k_H-1]}$ and $s_p = s_1^2 - s_L^2 - s_H^2 s_2^2 \sum_{i \in [0, k_H-1]} e_i e_i^* (s_2^2 - s_H^2 f_i f_i^*)^{-1}$.

1. $c \leftarrow 0$
2. **For** $i = 0, \dots, k_H - 1$ **do**
3. $p_{2,i} \leftarrow \text{SampleFz}(g_i, 0)$ $\triangleright p_{2,i} \sim \mathcal{D}_{\mathcal{R}, \sqrt{M_{\tau}(g_i)}, c}$ [GM18]
4. $c \leftarrow c + [\mathbf{f}']_i \cdot p_{2,i}$
5. $p_1 \leftarrow \text{SampleFz}(s_p, c)$ $\triangleright p_1 \sim \mathcal{D}_{\mathcal{R}, \sqrt{M_{\tau}(s_p)}, c}$ [GM18]

Output: $\mathbf{p} = [p_1 | p_{2,0} | \dots | p_{2,k_H-1}]^T$ (close to $\mathcal{D}_{\mathcal{R}^{1+k_H}, \sqrt{M_{\tau}(\mathbf{s}_p)}}$)

Such precomputations make our perturbation sampler simpler than those of prior works [GM18, AGJ⁺24, JS25b, JS26], but its security can still be obtained by adapting the security argument of [GM18, Thm. 4.1] to our case.

Lemma 3.4 ([GM18, Thm. 4.1] adapted). *It holds that the distribution $\mathcal{P}_p$ of **PerturbationSampler(e, f)** verifies $\mathcal{P}_p \approx_{\alpha_p} \mathcal{D}_{\mathcal{R}^{1+k_H}, \sqrt{M_{\tau}(\mathbf{s}_p)}}$ with $\alpha_p = ((1 + \varepsilon)/(1 - \varepsilon))^{(1+k_H)(n-1)+1} \approx 1 + (2(1 + k_H)(n - 1) + 2)\varepsilon$.*

3.4 Trapdoor Switching for **NTRU-TSampler**

We designed our trapdoor sampler so as to use independent NTRU instances to mask each component of the gadget $\mathbf{g}_{H,\mathbf{t}}$. Far from being an unnecessary choice, if we had considered an NTRU learning instance $\mathbf{h} = \mathbf{e}f^{-1}$, i.e., with $f_0 = \dots = f_{k_H-1}$, this would prevent us from changing the tag from $\mathbf{t}$ to $\mathbf{t}'$ component by component in the security proof of our group signature. We would have had to change it all at once, which would in turn require extending the matrix $\mathbf{a}_{\mathbf{t}}^T$ by a full block $q_L \mathbf{g}_H^T - \mathbf{e}'^T f'^{-1}$. In our case, we can change it componentwise, thus only requiring one extra element to append to $\mathbf{a}_{\mathbf{t}}^T$. This extra element is then leveraged in a hybrid argument (exactly as $\mathbf{A}_3$ or $\mathbf{a}_3$ in [AGJ⁺24, JS26]) to successively embed the partial trapdoor element $q_L b_H^j - e'_j f_j'^{-1}$. The hybrid argument then relies on the following trapdoor switching result, which essentially states that one can use **NTRU-TSampler** in two different ways for the extended matrix

$$\bar{\mathbf{a}}_{\mathbf{t}}^T = [1 | q_L \mathbf{g}_{H,\mathbf{t}}^T - (\mathbf{e} \oslash \mathbf{f})^T | q_L b_H^j t'_j - e'_j f_j'^{-1}].$$

It either (1) samples v_3 with Gaussian parameter s_2 , and computes a preimage of $u - (q_L b_H^j t'_j - e'_j f_j'^{-1})v_3$ using the regular matrix $\mathbf{a}_{\mathbf{t}}^T$ with trapdoor $(\mathbf{e}, \mathbf{f})$, or (2) samples $v_{2,j}$ with parameter s_2 , and computes a preimage of $u - (q_L b_H^j t_j -$

$e_j f_j^{-1})v_{2,j}$ with trapdoor $(\mathbf{e}', \mathbf{f}')$ obtained by replacing e_j, f_j by e'_j, f'_j . This essentially swaps the sampling roles of v_3 and $v_{2,j}$. We formalize it in the trapdoor switching lemma below.

Lemma 3.5. *Let q_L, k_H be in $\mathbb{N}^\times$, $j \in [0, k_H - 1]$, and $\varepsilon_L, \varepsilon_H, \varepsilon$ in $(0, 1)$. Then, define $q_H = b_H^{k_H}$ and $q = q_L q_H$. Let $u \in \mathcal{R}_q$, $\mathbf{e}, \mathbf{f}$ be in $\mathcal{R}^{k_H}$ such that $\mathbf{f} \bmod q\mathcal{R} \in (\mathcal{R}_q^\times)^{k_H}$, and $\mathbf{t} = [t_i]_{i \in [0, k_H - 1]} \in (\mathcal{R}_{b_H}^\times)^{k_H}$. Then, define $\mathbf{g}_{H, \mathbf{t}} = [t_i b_H^i]_{i \in [0, k_H - 1]}$. Let e'_j, f'_j be in $\mathcal{R}$ with $f'_j \bmod q\mathcal{R} \in \mathcal{R}_q^\times$, and $t'_j \in \mathcal{R}_{b_H}^\times$. We define $\mathbf{e}', \mathbf{f}', \mathbf{t}'$ from $\mathbf{e}, \mathbf{f}, \mathbf{t}$ by replacing the $j + 1$ -th entry by e'_j, f'_j, t'_j respectively.*

Let $s_L, s_H > 0$ be such that $s_L \geq q_L \eta_{\varepsilon_L}(\mathcal{R})$, $s_H \geq b_H \eta_{\varepsilon_H}(\mathcal{R})$ and $s_L, s_H > \eta_\varepsilon(\mathcal{R}^{1+k_H})$. Then, let $\gamma > 1$ and

$$s_1 = \sqrt{\frac{s_L^4}{s_L^2 - \eta^2} + \frac{\gamma^2}{\gamma^2 - 1} \cdot \frac{s_H^4}{s_H^2 - \eta^2} B_e^2}, \quad \text{and} \quad s_2 = \gamma \frac{s_H^2}{\sqrt{s_H^2 - \eta^2}} B_f$$

where $B_e^2 \geq \max(\|\sigma(\mathbf{e}^* \mathbf{e})\|_\infty, \|\sigma(\mathbf{e}'^* \mathbf{e}')\|_\infty)$ and $B_f \geq \max(\max_i \|\sigma([\mathbf{f}]_i)\|_\infty, \|\sigma(f'_j)\|_\infty)$. We define $\bar{\mathbf{a}}^T = [1 | q_L \mathbf{g}_{H, \mathbf{t}}^T - (\mathbf{e} \oslash \mathbf{f})^T | q_L b_H^j t'_j - e'_j f_j'^{-1}]$, and the following distributions.

$$\begin{aligned} \mathcal{P}_0 & \quad \text{Sample } v_3 \leftarrow \mathcal{D}_{\mathcal{R}, s_2}, \text{ compute } u' = u - (q_L t'_j b_H^j - e'_j f_j'^{-1}) v_3 \bmod q\mathcal{R}, \text{ sample } \\ & \quad \mathbf{v} \leftarrow \text{NTRU-TSampler}(\mathbf{e}, \mathbf{f}, u', \mathbf{t}, s_1, s_2, s_L, s_H), \text{ and output } (\mathbf{v}, v_3). \\ \mathcal{P}_1 & \quad \text{Sample } v_{2,j} \leftarrow \mathcal{D}_{\mathcal{R}, s_2}, \text{ compute } u' = u - (q_L t_j b_H^j - e_j f_j^{-1}) v_{2,j} \bmod q\mathcal{R}, \\ & \quad \text{sample } \mathbf{w} \leftarrow \text{NTRU-TSampler}(\mathbf{e}', \mathbf{f}', u', \mathbf{t}', s_1, s_2, s_L, s_H). \text{ Parse } \mathbf{w} \\ & \quad \text{as } (v_1, v_{2,0}, \dots, v_{2,j-1}, v_3, v_{2,j+1}, \dots, v_{2,k_H-1}) \text{ and compute } \mathbf{v} = \\ & \quad [v_1 | v_{2,0} | \dots | v_{2,k_H-1}]^T. \text{ Output } (\mathbf{v}, v_3). \end{aligned}$$

It holds that $\mathcal{P}_0$ and $\mathcal{P}_1$ have the same support and that $\mathcal{P}_0 \approx_{\delta^2} \mathcal{P}_1$ and $\mathcal{P}_1 \approx_{\delta^2} \mathcal{P}_0$ where δ is defined in Theorem 3.1.

Proof. For a given v_3 , we call $u'(v_3) = u - (q_L t'_j b_H^j - e'_j f_j'^{-1}) v_3$. For any $u' \in \mathcal{R}_q$, we also denote by $\mathcal{P}_0^{u'}$ the distribution of $\text{NTRU-TSampler}(\mathbf{e}, \mathbf{f}, u', \mathbf{t}, s_1, s_2, s_L, s_H)$. By Theorem 3.1, it holds that for any such u' , $\mathcal{P}_0^{u'} \approx_\delta \mathcal{D}_{\mathcal{L}_q^{u'}(\mathbf{a}_t^T), \sqrt{\mathbf{S}}}$, where $\mathbf{a}_t^T = [1 | q_L \mathbf{g}_{H, \mathbf{t}}^T - (\mathbf{e} \oslash \mathbf{f})^T] \bmod q\mathcal{R}$ and $\mathbf{S} = \text{diag}(s_1^2 \mathbf{I}_n, s_2^2 \mathbf{I}_{nk_H})$. For any $(\bar{\mathbf{v}}, \bar{v}_3)$, we first condition on $v_3 = \bar{v}_3$ and have

$$\mathcal{P}_0(\bar{\mathbf{v}}, \bar{v}_3) = \mathcal{D}_{\mathcal{R}, s_2}(\bar{v}_3) \cdot \mathcal{P}_0^{u'(\bar{v}_3)}(\bar{\mathbf{v}}) \in [\delta^{-1}, \delta] \mathcal{D}_{\mathcal{R}, s_2}(\bar{v}_3) \cdot \mathcal{D}_{\mathcal{L}_q^{u'(\bar{v}_3)}(\mathbf{a}_t^T), \sqrt{\mathbf{S}}}(\bar{\mathbf{v}}).$$

Yet, the distribution of $\mathbf{v} \sim \mathcal{D}_{\mathcal{L}_q^{u'}(\mathbf{a}_t^T), \sqrt{\mathbf{S}}}$ exactly matches the conditional distribution “ $\mathcal{D}_{\mathcal{R}^{1+k_H}, \sqrt{\mathbf{S}}}$ conditioned on $\mathbf{a}_t^T \mathbf{v} = u' \bmod q\mathcal{R}$ ”. As such, we have

$$\begin{aligned} \mathcal{P}_0(\bar{\mathbf{v}}, \bar{v}_3) & \in [\delta^{-1}, \delta] \mathcal{D}_{\mathcal{R}, s_2}(\bar{v}_3) \cdot \left(\mathcal{D}_{\mathcal{R}^{1+k_H}, \sqrt{\mathbf{S}}} \left| \bar{\mathbf{a}}^T \begin{bmatrix} \mathbf{v} \\ \bar{v}_3 \end{bmatrix} = u \bmod q\mathcal{R} \right. \right) (\bar{\mathbf{v}}) \\ & = [\delta^{-1}, \delta] \left(\mathcal{D}_{\mathcal{R}^{2+k_H}, \sqrt{\mathbf{S}'}} \left| \bar{\mathbf{a}}^T \begin{bmatrix} \mathbf{v} \\ v_3 \end{bmatrix} = u \bmod q\mathcal{R} \right. \right) (\bar{\mathbf{v}}, \bar{v}_3) \\ & = [\delta^{-1}, \delta] \mathcal{D}_{\mathcal{L}_q^u(\bar{\mathbf{a}}^T), \sqrt{\mathbf{S}'}}(\bar{\mathbf{v}}, \bar{v}_3) \end{aligned}$$

where $\mathbf{S}' = \text{diag}(\mathbf{S}, s_2^2 \mathbf{I}_n)$. A similar calculation for $\mathcal{P}_1$ but by conditioning on $v_{2,j} = \overline{v_{2,j}}$ and re-arranging gives $\mathcal{P}_1 \approx_\delta \mathcal{D}_{\mathcal{L}_q^u(\overline{\mathbf{a}}^T), \sqrt{\mathbf{S}'}}$ as well. Combining both then yields $\mathcal{P}_0 \approx_{\delta^2} \mathcal{P}_1$ and $\mathcal{P}_1 \approx_{\delta^2} \mathcal{P}_0$, then concluding the proof. $\square$

4 Tag-NTRU-ISIS and Known-NTRU-ISIS Assumptions

This section introduces two weaker variants of the NTRU-ISIS_f assumption, designed to interact with our sampler rather than Falcon's. As we shall see, these two assumptions naturally arise when proving security of advanced signatures based on our sampler, and our group signature in particular. The NTRU-ISIS_f assumption was introduced in [BLNS23] as a way to design lattice anonymous credentials more efficiently. Current constructions indeed rely on zero-knowledge proofs of a signature on a set of hidden attributes (the message). Reaching compact sizes *and* computational efficiency then discards regular signatures like ML-DSA or Falcon due to their reliance on cryptographic hashing of the message. The idea behind NTRU-ISIS_f is to replace the hash in Falcon by an algebraically structured function $\mathbf{f}$, then turning Falcon into a proof-friendly signature suited for anonymous credentials. The assumption then essentially argues that Falcon remains secure in this case, and without the random oracle model. Concretely, it states an adversary being given random $x_1, \dots, x_k$ and Gaussian preimages $\mathbf{v}_i$ of the $\mathbf{f}(x_i)$ for the matrix $\mathbf{a}^T = [1|ef^{-1}|\mathbf{a}_3^T]$ (for some short e, f defining a Falcon trapdoor, and $\mathbf{a}_3$ uniform) cannot produce a fresh $(x^*, \mathbf{v}^*)$ such that $\mathbf{a}^T \mathbf{v}^* = \mathbf{f}(x^*)$ with $\mathbf{v}^*$ short and x^* in the desired space. As mentioned above, when $\mathbf{f}$ is a random function, NTRU-ISIS_f corresponds to the security of Falcon and thus reduces from NTRU and R-ISIS. Its unsuitability for anonymous credentials pushed the authors to instead consider a binary encoding function $\mathbf{f}$. Although the assumption looses its reduction to well-known problems, it seems to remain robust despite opening to other attack methodologies, as we describe in Section 4.1.

Our first assumption, called *Tag NTRU Inhomogeneous Short Integer Solution* (T-NTRU-ISIS), is inspired by NTRU-ISIS_f but departs from it in a number of ways which we believe make it harder. We give the adversary a vector $\mathbf{h} = \mathbf{e} \oslash \mathbf{f}$ of k_H NTRU instances and some uniform a_3 , as well as preimages of a random target $u \in \mathcal{R}_q$, but for *tagged* matrices. The adversary then has to produce a preimage of u for the zero-tag matrix, i.e., $[1|\mathbf{h}^T|a_3]$. More precisely, the adversary is given $(\mathbf{h}, a_3, u)$ and N pairs $(t_i, \mathbf{v}_i)$ such that $[1|q_L t_i \mathbf{g}_H^T - \mathbf{h}^T|a_3] \mathbf{v}_i = u \bmod q\mathcal{R}$ for tags t_i in a designated tag space, and must compute some short $\mathbf{v}^*$ such that $[1|\mathbf{h}^T|a_3] \mathbf{v}^* = u \bmod q\mathcal{R}$. As opposed to NTRU-ISIS_f, the security of our assumption does not rely on the careful choice of a function $\mathbf{f}$, and is closer in essence to a regular version of ISIS as we still ask for a short preimage of a perfectly random target. The sole advantage an adversary could have over ISIS may come from the preimages, but they are given for distinct tags, thus changing the matrix and underlying lattice for each one. We discuss below some cryptanalytic intuitions and first formally define the assumption here.

Definition 4.1 (T-NTRU-ISIS). Let $\mathcal{R}$ be a power-of-two cyclotomic ring of degree n . Let q, q_L, b_H, k_H be in $\mathbb{N}^\times$, and $\beta > 0$. We define $\mathbf{g}_H^T = [1|b_H|\dots|b_H^{k_H-1}]$. Let $\mathcal{D}$ be a distribution on $\mathcal{R}$, and $\mathcal{T} \subset \mathcal{R} \setminus \{0\}$ a tag space. The Tag NTRU Inhomogeneous Short Integer Solution problem, denoted by $\text{T-NTRU-ISIS}_{n,q,k_H,\mathcal{D},\mathcal{T},\beta}$, is defined by the experiment of Figure 4.1. The advantage of a PPT adversary $\mathcal{A}$ is $\text{Adv}_{\text{T-NTRU-ISIS}}[\mathcal{A}] = \mathbb{P}[\text{Game}_{\mathcal{A}}^{\text{T-NTRU-ISIS}}() = 1]$.

```

Game $\mathcal{A}$ T-NTRU-ISIS():
1  $\mathbf{f} \leftarrow \mathcal{D}^{k_H}$  conditioned on each  $[\mathbf{f}]_i$  being a unit modulo  $q$ , and  $\mathbf{e} \leftarrow \mathcal{D}^{k_H}$ 
2  $a_3, u \leftarrow U(\mathcal{R}_q)$ 
3  $\mathbf{h} \leftarrow \mathbf{e} \odot \mathbf{f} \bmod q\mathcal{R}$  and  $\mathbf{a}^T \leftarrow [1|\mathbf{h}^T|a_3]$ 
4 For distinct  $t_i \in \mathcal{T}$  ( $i = 1, \dots, |\mathcal{T}|$ ) do
5    $\mathbf{a}_{t_i}^T \leftarrow [1|q_L t_i \mathbf{g}_H^T - \mathbf{h}^T|a_3]$ 
6    $\mathbf{v}_i \leftarrow \mathbf{a}_{t_i}^{-1}(u)$   $\triangleright$  verifies  $\mathbf{a}_{t_i}^T \mathbf{v}_i = u \bmod q\mathcal{R}$  and  $\|\mathbf{v}_i\|_2 \leq \beta$ 
7    $\mathbf{v}^* \leftarrow \mathcal{A}(\mathbf{a}, u, (t_i, \mathbf{v}_i)_{i \in [|\mathcal{T}|]})$ 
8 return  $[\mathbf{a}^T \mathbf{v}^* = u \bmod q\mathcal{R}] \wedge [\|\mathbf{v}^*\|_2 \leq \beta]$ 

```

Fig. 4.1. The T-NTRU-ISIS game.

Observe that if one sets the parameters as in Section 3, the assumption is falsifiable. Indeed, if $q = q_L b_H^{k_H}$, then step 6 of $\text{Game}_{\mathcal{A}}^{\text{T-NTRU-ISIS}}$ in Figure 4.1 can be instantiated using **NTRU-TSampler**. One simply has to sample some $v_{3,i}$ from $\mathcal{D}_{\mathcal{R},s_2}$, and use **NTRU-TSampler** on the updated syndrome $u - a_3 v_{3,i} \bmod q\mathcal{R}$.

4.1 Cryptanalytic Perspectives and Comparison with NTRU-ISIS_f

The robustness of T-NTRU-ISIS crucially depends on whether the preimages procure the adversary a particular advantage. Indeed, without them, the assumption would naturally reduce from NTRU and R-ISIS by changing $\mathbf{h}$ by some uniform vector under NTRU. We however note that using Theorem 3.1, the preimages do not actually leak the underlying NTRU structure of $\mathbf{h}$. As a result, said preimages do not seem to procure $\mathcal{A}$ any significant advantage over the NTRU+R-ISIS approach, besides trying to combine them in a non-trivial way. We thus survey methods to do so and how our situation differs from that of NTRU-ISIS_f [BLNS23].

Lattice/Combinatorial Attacks. In [BLNS23], the authors use NTRU-ISIS_f with $\mathbf{f}(x) = \tau^{-1}(\mathbf{B} \cdot \text{bin}(x-1))$, where $\mathbf{B}$ is an integer matrix, and $\text{bin}(x-1)$ is the binary decomposition of $x-1 \in [0, N-1]$. As a result, the given preimages $\mathbf{v}_i$ satisfy $M_\tau(\mathbf{a}^T)\tau(\mathbf{v}_i) = \mathbf{B} \cdot \text{bin}(x_i-1)$, and therefore $\mathbf{s}_i = [\tau(\mathbf{v}_i)^T | \text{bin}(x_i-1)^T]^T$ is short and verifies $\mathbf{A}\mathbf{s}_i = \mathbf{0} \bmod q\mathbb{Z}$, with $\mathbf{A} = [M_\tau(\mathbf{a}^T) | -\mathbf{B}]$. Packing these vectors into a matrix $\mathbf{S}$ gives $\mathbf{A}\mathbf{S} = \mathbf{0} \bmod q\mathbb{Z}$. Finding a short vector $\mathbf{x}$ in $\mathcal{L}(\mathbf{S})$ then seems sufficient. However, one has to ensure that $\mathbf{x}$ is not within the columns of $\mathbf{S}$ while still having a bottom part that is binary (so that the latter corresponds to some $\text{bin}(x^*-1)$ with $x^* \in [N]$). Satisfying this binary

constraint while ensuring $\mathbf{x}$ passes the acceptance bound β (which is close or equal to the bound on the $\mathbf{s}_i$) renders this attack complex. One can actually interpret this attack as some kind of combinatorial attack consisting in finding a non-trivial linear combination $(\alpha_{i_1}, \dots, \alpha_{i_\ell})$ and some $x^* \in [N]$ such that $\sum_{j \in [\ell]} \alpha_{i_j} \mathbf{f}(x_{i_j}) = \mathbf{f}(x^*)$, so that $\mathbf{v}^* = \sum_{j \in [\ell]} \alpha_{i_j} \mathbf{v}_{i_j}$ verifies $\mathbf{a}^T \mathbf{v}^* = \mathbf{f}(x^*)$. This is fairly easy, even with only few ternary α_{i_j} , but the security of NTRU-ISIS_f actually lies in the difficulty of finding such a linear combination that does not blow up the norm of $\mathbf{v}^*$ beyond the acceptance bound β .

The non-linearity of our assumption (as t_i multiplies the preimage $\mathbf{v}_i$) thwarts the previous combinatorial attack even more. Our situation indeed differs in several key aspects. First, the target cannot be adaptively chosen by the adversary through an $\mathbf{f}(x^*)$ as it needs to find a preimage of a given u for $\mathbf{a}^T = [1 | \mathbf{h}^T | a_3]$, i.e., the zero-tag. Second, due to the non-linearity, one would need to find some combination (α_{i_j}) such that $\sum_j \alpha_{i_j} t_{i_j} \mathbf{v}_{2,i_j} = \mathbf{0}$ and $\sum_j \alpha_{i_j} = 1$. In that case, building $\mathbf{v}^* = \sum_j \alpha_{i_j} \mathbf{v}_{i_j}$ would give $\mathbf{a}^T \mathbf{v}^* = u - q_L \mathbf{g}_H^T \sum_j \alpha_{i_j} t_{i_j} \mathbf{v}_{2,i_j} = u$. One also has to ensure $\mathbf{v}^*$ is as short as the $\mathbf{v}_i$. The attack thus seems much more complex than for NTRU-ISIS_f, as finding such a combination corresponds to a more restrictive and higher-dimensional subset sum problem. We can still interpret it as a constrained lattice attack. Concretely, each given preimages satisfies $[1 | q_L \mathbf{g}_H^T] - \mathbf{h}^T [a_3 | u] \mathbf{s}_i = 0 \bmod q\mathcal{R}$, where $\mathbf{s}_i = [v_{1,i} | t_i \mathbf{v}_{2,i}^T | \mathbf{v}_{2,i}^T | v_{3,i} - 1]^T$. We then obtain a similar relation of the form $\bar{\mathbf{a}}^T \mathbf{S} = \mathbf{0} \bmod q\mathcal{R}$. Deriving a solution of T-NTRU-ISIS can then be done by finding some $\mathbf{x} = [x_1 | \widetilde{\mathbf{x}}_2 | \mathbf{x}_2 | x_3 | x_u]^T \in \mathcal{L}(\mathbf{S}) \setminus \{[\mathbf{S}]_i; i \in [k]\}$ such that $\|\mathbf{x}\|_2 \leq \sqrt{\beta^2 + 1}$, but most importantly such that $x_u = -1$, and $\widetilde{\mathbf{x}}_2 = \mathbf{0}$. The latter two equations correspond to the constraints $\sum_j \alpha_{i_j} t_{i_j} \mathbf{v}_{2,i_j} = \mathbf{0}$ and $\sum_j \alpha_{i_j} = 1$.

4.2 Revealing the NTRU structure

Our assumption and parameter setting also vastly differ from that of NTRU-ISIS_f. In the latter, solving the underlying NTRU instance readily breaks the assumption. As we mentioned above, the latter is indeed used in constructions based on Falcon-like trapdoors [PFH+20], whose NTRU distribution $\mathcal{D}$ is compatible with the Falcon preimage sampler by design. Recovering the underlying short e, f such that $ef^{-1} = h$ is then sufficient to build a full NTRU trapdoor, which is by design of the same quality as that of Falcon preimages. From (e, f) , one can complete the Falcon key generation deterministically, then giving the adversary the ability to find any short enough preimage $\mathbf{v}^*$ of any $\mathbf{f}(x^*)$. It means that NTRU-ISIS_f is no harder than the NTRU assumption used to define its instance.

In our case, breaking NTRU does not seem sufficient to solve T-NTRU-ISIS. Let us assume an adversary broke the first NTRU instance h_1 (without loss of generality) and recovered the underlying e_1, f_1 . Ignoring the preimages of the T-NTRU-ISIS assumption, it could be tempted to set $\mathbf{v}^* = [v_1 | v_2 | 0 | \dots | 0]^T$, where (v_1, v_2) would be sampled as a preimage of u for the matrix $[1 | h_1]$ using a sampler as in Falcon. However, such a sampler would not give a sufficiently short vector. Our case is indeed very different due to our distribution $\mathcal{D}$ of

NTRU keys which is closer to NTRUEncrypt-like keys than Falcon keys. Indeed, in our sampler and group signature, we choose the polynomials e_i, f_i of our NTRU instances to be drawn from a centered binomial distribution, leading to ternary polynomials which are then much shorter than in Falcon. Being able to sample Falcon preimages from (e_1, f_1) requires completing it into a short enough basis of the NTRU lattice $\mathcal{L}_q^\perp([1|h_1])$. As studied and shown in [DLP14], completing the basis from $(-e_1, f_1)$ would lead to a Gram-Schmidt norm of roughly $\max(\|[-e_1|f_1]\|_2, q/\|[-e_1|f_1]\|_2)$. Hence the balanced setting of Falcon to choose e, f close to $\sqrt{q}$ so that the complete basis reaches a quality of roughly $\sqrt{q}$. In our case, e_1, f_1 are binomial, which means the completed basis would have Gram-Schmidt norm close to $\max(B_{ef}, q/B_{ef}) = q/B_{ef}$, where $B_{ef} \approx 1.15\sqrt{n}$ is the binomial concentration bound of a vector of size $2n$. As a result, Gaussian preimages generated from such a trapdoor would yield $\|\mathbf{v}^*\|_2 = \|[v_1|v_2]^T\|_2 \approx \frac{1}{\sqrt{2\pi}} \cdot \eta_\varepsilon(\mathbb{Z}^{2n}) \frac{q}{B_{ef}} \cdot \sqrt{2n} > q$. Our parameter choice however chooses $\beta < q$, which then discards such an attack. It is therefore unclear whether solving NTRU even gives an advantage in solving T-NTRU-ISIS.

Said differently, as opposed to NTRU-ISIS_f, the ISIS matrix $\mathbf{a}^T$ the adversary has to target in T-NTRU-ISIS is, to our knowledge, incompatible with preimage sampling. In particular, leaking the NTRU instances may still result in a secure variant of R-ISIS. We thus introduce another assumption, which we call Known-NTRU-ISIS. The latter avoids q-type assumptions as T-NTRU-ISIS where the challenger is required to give preimages $\mathbf{v}_i$. Concretely, the adversary is still given $\mathbf{a}^T = [1|\mathbf{h}^T|a_3]$ for $\mathbf{h} = \mathbf{e} \oslash \mathbf{f} \bmod q\mathcal{R}$, and u , and asked to find a short $\mathbf{v}^*$ such that $\mathbf{a}^T \mathbf{v}^* = u \bmod q\mathcal{R}$. But we also give the adversary the NTRU structure of $\mathbf{h}$ by giving it $(\mathbf{e}, \mathbf{f})$, hence the *known* NTRU part of the name.

Definition 4.2 (Known-NTRU-ISIS). *Let $\mathcal{R}$ be a power-of-two cyclotomic ring of degree n . Let q, k_H be in $\mathbb{N}^\times$, and let $\beta > 0$. Let $\mathcal{D}$ be a short distributions on $\mathcal{R}$. The Known NTRU Inhomogeneous Short Integer Solution problem Known-NTRU-ISIS _{$n, q, k_H, \mathcal{D}, \beta$} is defined by the experiment of Figure 4.2. We define $\text{Adv}_{\text{Known-NTRU-ISIS}}[\mathcal{A}] = \mathbb{P}[\text{Game}_{\mathcal{A}}^{\text{Known-NTRU-ISIS}}() = 1]$ to be the advantage of a PPT adversary $\mathcal{A}$.*

```

Game $\mathcal{A}$ Known-NTRU-ISIS():
  1  $\mathbf{f} \leftarrow \mathcal{D}^{k_H}$  conditioned on each  $[\mathbf{f}]_i$  being a unit modulo  $q$ , and  $\mathbf{e} \leftarrow \mathcal{D}^{k_H}$ 
  2  $a_3, u \leftarrow U(\mathcal{R}_q)$ 
  3  $\mathbf{v}^* \leftarrow \mathcal{A}(\mathbf{e}, \mathbf{f}, a_3, u)$ 
  4 return  $\mathbb{I}[1|(\mathbf{e} \oslash \mathbf{f})^T|a_3]\mathbf{v}^* = u \bmod q\mathcal{R}] \wedge \mathbb{I}[\|\mathbf{v}^*\|_2 \leq \beta]$ 

```

Fig. 4.2. The Known-NTRU-ISIS game.

The Known-NTRU-ISIS assumption naturally reduces to T-NTRU-ISIS, showing it is no harder than the latter. If one has access to the NTRU structure, they can use NTRU-TSampler to sample all the preimages $\mathbf{v}_i$ for the tags t_i

and thus simulate the T-NTRU-ISIS challenger. Although Known-NTRU-ISIS is then easier than T-NTRU-ISIS, it provides a clearer path for the cryptanalysis of T-NTRU-ISIS. We can indeed observe the existence of a reduction from T-NTRU-ISIS to the combination problem “Known-NTRU-ISIS + search NTRU”. Given $\mathbf{h}$, one would use the search-NTRU oracle to obtain the underlying $\mathbf{e}, \mathbf{f}$, and feed them to the Known-NTRU-ISIS oracle to solve T-NTRU-ISIS. As a result, a successful attack on Known-NTRU-ISIS could possibly lead to one on T-NTRU-ISIS if one is able to break the NTRU key $\mathbf{h} = \mathbf{e} \odot \mathbf{f}$.

Reduction from R-ISIS. To give more intuition on why the Known-NTRU-ISIS assumption is reasonable in our case and on our differences with NTRU-ISIS_f, we give a straightforward, albeit likely not tight on the parameters, reduction from R-ISIS. We have shown that in the parameterization of NTRU-ISIS_f from [BLNS23], giving out the NTRU key (e, f) directly breaks NTRU-ISIS_f. As described above, our case differs in the distribution of the NTRU key which is much shorter, and incompatible with preimage samplers à la Falcon. We can actually show Known-NTRU-ISIS is no easier than R-ISIS with a larger bound, the overhead depending on the size of the NTRU distribution.

Lemma 4.1. *Let $\mathcal{R}$ be a power-of-two cyclotomic ring of degree n . Let $q, k_H \in \mathbb{N}^\times$ and $\beta > 0$. Let $\mathcal{D}$ be a distribution on $\mathcal{R}$ and define*

$$B = \sup_{\mathbf{e}, \mathbf{f} \sim \mathcal{D}^{k_H}} \left\| M_\tau \left(\left[\prod_i [\mathbf{f}]_i \mid \mathbf{e} \odot \left[\prod_{j \neq i} [\mathbf{f}]_j \right]_i \right] \right) \right\|_2.$$

There is an efficient reduction from $\text{R-ISIS}_{n,q,2,B\beta}$ to $\text{Known-NTRU-ISIS}_{n,q,k_H,\mathcal{D},\beta}$, showing $\varepsilon_{\text{Known-NTRU-ISIS}} \leq \varepsilon_{\text{R-ISIS}}$ for the respective hardness bounds. For example, $B = \sup_{e,f \sim \mathcal{D}} \|M_\tau([f|e])\|_2$ when $k_H = 1$, which for $\mathcal{D} = \mathcal{B}_1$ leads to $B = O(\sqrt{2n})$.

Proof. Let $\mathcal{A}$ be an adversary solving $\text{Known-NTRU-ISIS}_{n,q,k_H,\mathcal{D},\beta}$. We construct an adversary $\mathcal{A}'$ for $\text{R-ISIS}_{n,q,2,B\beta}$ for a matrix $[1|a'_3]$, as follows. Given the R-ISIS instance $([1|a'_3], u')$, $\mathcal{A}'$ samples $\mathbf{e}, \mathbf{f}$ from $\mathcal{D}^{k_H}$ with $[\mathbf{f}]_i$ being a unit modulo q . Then, $\mathcal{A}'$ computes

$$f = \prod_{i=0}^{k_H-1} [\mathbf{f}]_i, \text{ and } \forall i, e'_i = [\mathbf{e}]_i \cdot \prod_{\substack{j=0 \\ j \neq i}}^{k_H-1} [\mathbf{f}]_j, \text{ and } \mathbf{e}' = [e'_i]_{i \in [0, k_H-1]}.$$

It uses f to define $a_3 = a'_3 f^{-1} \bmod q\mathcal{R}$ and $u = u' f^{-1} \bmod q\mathcal{R}$, and calls $\mathcal{A}$ to obtain $\mathbf{x} = [x_1 | \mathbf{x}_2 | x_3]^T \leftarrow \mathcal{A}(\mathbf{e}, \mathbf{f}, a_3, u)$. The adversary $\mathcal{A}'$ then returns $\mathbf{x}' = [fx_1 + \mathbf{e}'^T \mathbf{x}_2 | x_3]^T$ as the R-ISIS solution.

First, because a'_3 and u' are uniform and f is a unit modulo q , it holds that a_3, u are still uniform and correctly distributed. The instance given to $\mathcal{A}$ is therefore a well-distributed instance of $\text{Known-NTRU-ISIS}_{n,q,k_H,\mathcal{D},\beta}$. Hence, $\mathcal{A}$ returned $\mathbf{x}$ such that $[1 | (\mathbf{e} \odot \mathbf{f})^T | a_3] \mathbf{x} = u \bmod q\mathcal{R}$ and $\|\mathbf{x}\|_2 \leq \beta$. The first

equation, when multiplied by f , can then be written as

$$\begin{aligned} u' &= fx_1 + \sum_{i=0}^{k_H-1} [\mathbf{e}]_i \cdot [\mathbf{f}]_i^{-1} \cdot f \cdot [\mathbf{x}_2]_i + a'_3 x_3 \bmod q\mathcal{R} \\ &= \left(fx_1 + \sum_{i=0}^{k_H-1} e'_i \cdot [\mathbf{x}_2]_i \right) + a'_3 x_3 \bmod q\mathcal{R} = [1|a'_3]\mathbf{x}' \bmod q\mathcal{R} \end{aligned}$$

Additionally, we have $\|\mathbf{x}'\|_2^2 \leq \|M_\tau([f|\mathbf{e}'^T])\|_2^2 \|x_1|\mathbf{x}_2^T\|_2^2 + \|x_3\|_2^2 \leq B^2\beta^2$. It then yields

$$\varepsilon_{\text{R-ISIS}} \geq \text{Adv}_{\text{R-ISIS}}[\mathcal{A}'] \geq \text{Adv}_{\text{Known-NTRU-ISIS}}[\mathcal{A}],$$

and thus $\varepsilon_{\text{R-ISIS}} \geq \varepsilon_{\text{Known-NTRU-ISIS}}$. $\square$

When $\mathcal{D} = \mathcal{B}_1$, the overhead to the R-ISIS bound is rather small when $k_H = 1$ ($B = O(\sqrt{2n})$). When $k_H > 1$, evaluating B is more complex but it remains independent of q (as opposed to Falcon key distributions which scale with $\sqrt{q}$). Hence, at the expense of a larger q and thus deteriorated parameters, one can obtain a reduction from R-ISIS. To our knowledge, no similar reduction is known for NTRU-ISIS_f.

Our group signature, and likely most applications of our sampler, uses $k_H = 2$ (at least). The reduction then requires much larger parameters as $B = O(2n)$. We depart from these parameter settings to reach more compact sizes, but it however gives more confidence in Known-NTRU-ISIS (and T-NTRU-ISIS by extension) by showing that giving out the NTRU structure for very small NTRU keys does not fundamentally make the problem significantly easier, as opposed to NTRU-ISIS_f. In the end, we have the following hierarchy.

$$\begin{array}{ccccccc} & & & & \text{NTRU-ISIS}_f & \longrightarrow & \text{NTRU} \\ \text{R-ISIS} & \xrightarrow{\beta' = B\beta} & \text{Known-NTRU-ISIS} & \longrightarrow & \text{T-NTRU-ISIS} & \longrightarrow & \text{"Known-NTRU-ISIS} + \text{NTRU"} \end{array}$$

We show in Section 5 that our group signature can be proven secure based on either assumption. This means our security can be based on the rather simple Known-NTRU-ISIS assumption, as well as our trapdoor switching argument (itself based on NTRU), but if Known-NTRU-ISIS turns out to be significantly easier than T-NTRU-ISIS, our group signature would remain secure on the latter. The cryptanalysis of both assumptions (beyond attacking NTRU/R-ISIS) or different reductions from standard assumptions are left as interesting research directions.

5 Practical Group Signature from NTRU

We now describe our group signature. Despite the incentive put forth in [DFJS26] of changing paradigm in the design of group signatures, we go back to the original approach of [BMW03]. Albeit elegant, the approach of [DFJS26] presents certain complexities which are unsuitable for concrete applications. The mechanism needed to produce a delegated basis for each user seems extremely cumbersome in high dimensions which prevents large-scale deployment of their solution. Additionally, relying on delegated trapdoor naturally deteriorates the quality of

the preimages, thus requiring larger parameters and as such larger user secret keys, group public keys, but also signatures. This increase in parameters unfortunately outweighs the benefit of pulling the preimage out of the zero-knowledge proof component. In our paper, we rehabilitate the solution of a group manager producing short certificates and users proving validity of their certificate in zero-knowledge. In particular, we have a very efficient key generation procedure, and are able to have small user and group keys.

For the opening, we also go back to using an encryption independent of the certificate component. This allows for choosing independent parameters and thus be more flexible to match a given security target. We also show that the approach of recycling randomness to obtain a CCA-anonymous group signature is not specific to the scheme of [DFJS26]. We indeed use a similar idea to enhance our group signature to the highest level of anonymity, with a relatively small overhead. In particular, we rely on the same hint-type assumption MH-M-LWE. Also, as the encryption component is independent of the certificate one, this shared randomness technique does not impact the proof of traceability, as opposed to [DFJS26]. In their case, they need to rely on the hint assumption with Q_s secrets for the traceability which required increasing the parameters to reach the desired security. In our construction, we only need the assumption for the anonymity proof, which only requires one hint for one secret. We are thus able to use smaller parameters which improves the size and security of the encryption.

5.1 Specifications

The formal description of our group signature can be found below. Our scheme leverages a tower of power-of-two cyclotomic rings $\mathcal{R}_\pi \subseteq \mathcal{R}_{\text{op}} \subseteq \mathcal{R}$ of degrees $n_\pi \leq n_{\text{op}} \leq n$. This allows for decorrelating the different components (signature, encryption, zero-knowledge proof) to allow for a finer parameter selection, tighter security, and smaller sizes. We discuss it further below. We concatenate the subscripts and use $\mathcal{R}_{\text{op},p}$ to denote $\mathcal{R}_{\text{op}}/p\mathcal{R}_{\text{op}}$. We present our CCA-anonymous construction, but note that the opening parameters can be simplified if one aims at a CPA-anonymous variant. In particular, in the CPA-anonymous case, one need not to use Gaussian distributions for the ciphertext randomness and can use parameters similar to an optimized public-key encryption such as Kyber [BDK⁺18].

Algorithm 5.1: GSetup($1^\lambda, 1^N$)

1. Choose $n, n_{\text{op}}, n_\pi, \kappa$ to be powers of two such that $n_\pi \leq n_{\text{op}} \leq n$ and $\kappa \leq n$
 2. $k_\pi \leftarrow n/n_\pi$ and $k_{\text{op},\pi} \leftarrow n_{\text{op}}/n_\pi$
- Parameters of NTRU-TSampler.**
3. Choose $b_H, q_L > \sqrt{\kappa}^\kappa$ to be prime congruent to $2\kappa + 1$ modulo 4κ
 4. Choose $k_H \in \mathbb{N}^\times$
 5. $q_H \leftarrow b_H^{k_H}$
 6. $q \leftarrow q_L q_H$
 7. Choose $w \in \mathbb{N}^\times$ s.t. $\binom{n_\pi}{w} \geq N$ and set $N \leftarrow \binom{n_\pi}{w}$
 8. $\mathcal{I} \leftarrow \{\text{id} \in \mathcal{R}_\pi : \tau(\text{id}) \in \{0, 1\}^{n_\pi}, \text{ and } \|\text{id}\|_1 = w\}$
 9. $B_e \leftarrow \sqrt{n} + \sqrt{n k_H}$
 10. $B_f \leftarrow 2\sqrt{n}$

11. Choose $\varepsilon \in (0, 1)$ and $\gamma > 1$ ▷ typically $\varepsilon \approx 2^{-40}$ and $\gamma = \sqrt{2}$
 12. $s_L \leftarrow q_L \cdot \sqrt{\ln(2n/\varepsilon)/\pi}$ ▷ $s_L \approx q_L \eta_\varepsilon(\mathcal{R})$ by Lemma 2.3
 13. $s_H \leftarrow b_H \cdot \sqrt{\ln(2nk_H/\varepsilon)/\pi}$ ▷ $s_H \approx b_H \eta_\varepsilon(\mathcal{R}^{k_H}) \geq b_H \eta_{\varepsilon/k_H}(\mathcal{R})$ by Lemma 2.3
 14. $\eta \leftarrow \sqrt{\ln(2n(1+k_H)/\varepsilon)/\pi}$ ▷ $\eta \approx \eta_\varepsilon(\mathbb{Z}^{n(1+k_H)})$ by Lemma 2.3
 15. $s_1 \leftarrow \sqrt{s_L^4/(s_L^2 - \eta^2) + \gamma^2/(\gamma^2 - 1)} \cdot s_H^4 B_e^2/(s_H^2 - \eta^2)$
 16. $s_2 \leftarrow \gamma s_H^2 B_f / \sqrt{s_H^2 - \eta^2}$
 17. $B_1 \leftarrow c_n s_1 \sqrt{n}$ ▷ set by Lemma 2.5
 18. $B_2 \leftarrow c_{n(k_H+1)} s_2 \sqrt{n(k_H+1)}$ ▷ set by Lemma 2.5
Opening Parameters.
 19. Choose $d \in \mathbb{N}^\times$
 20. $B_{op} \leftarrow \xi \sqrt{n_{op} d}$ ▷ Binomial concentration bound with $\xi \approx 1.15$
 21. $\beta_{op} \leftarrow \frac{3}{4}(\sqrt{n_{op}} + \sqrt{2n_{op}d} + 6)$ ▷ Spectral norm bound of opening key
 22. Choose $\varepsilon_{op} \in (0, 1)$ ▷ typically $\varepsilon_{op} \leq 2^{-\lambda}$
 23. $\eta_{op} \leftarrow \sqrt{\ln(4n_{op}d/\varepsilon_{op})/\pi}$ ▷ $\eta_{op} \approx \eta_{\varepsilon_{op}}(\mathcal{R}_{op}^{2d})$ by Lemma 2.3
 24. Choose $\alpha > \sqrt{2}$
 25. $s_{se} \leftarrow \alpha \cdot \sqrt{2\eta_{op}}$
 26. $s_{e1} \leftarrow \frac{\alpha\sqrt{2}}{\sqrt{\alpha^2-2}} \cdot \sqrt{2\eta_{op}} \cdot \beta_{op}$
 27. $B_{se} \leftarrow c_{2n_{op}d} s_{se} \sqrt{2n_{op}d}$ ▷ set by Lemma 2.5
 28. $B_{e1} \leftarrow c_{n_{op}} s_{e1} \sqrt{n_{op}}$ ▷ set by Lemma 2.5
 29. Choose $p \in \mathbb{N}^\times$ s.t. $p > 4(B_{e1} + B_{se} B_{op})$
 30. $\text{params} \leftarrow (N, n, n_{op}, n_\pi, b_H, k_H, q_H, q_L, q, w, B_e, B_f, \varepsilon, s_L, s_H, \eta, s_1, s_2, B_1, B_2, B_{op}, \beta_{op}, \varepsilon_{op}, \eta_{op}, s_{se}, s_{e1}, B_{se}, B_{e1}, p)$
 31. $\text{seed} \leftarrow U(\{0, 1\}^{256})$
 32. $\mathbf{A}_{op} \leftarrow \mathcal{H}_1(\text{seed})$ ▷ follows $U(\mathcal{R}_{op}^{d \times d})$
 33. $u \leftarrow \mathcal{H}_2(\text{seed})$ ▷ follows $U(\mathcal{R}_q)$
 34. $a_3 \leftarrow \mathcal{H}_3(\text{seed})$ ▷ follows $U(\mathcal{R}_q)$
Output: $\text{pp} = (\text{params}, \text{seed})$

Algorithm 5.2: GKeyGen(pp)

1. $\mathbf{x}_{op,i} = [\mathbf{s}_{op,i}^T | \mathbf{e}_{op,i}^T]^T \leftarrow \mathcal{B}_1^{2d}$ s.t. $\|\mathbf{x}_{op,i}\|_2 \leq B_{op}$ and $\|\sigma(\mathbf{x}_{op,i}^* \mathbf{x}_{op,i})\|_\infty \leq \beta_{op}^2$ for $i \in \{1, 2\}$
 2. $\mathbf{b}_{op,i} = \mathbf{A}_{op}^T \mathbf{s}_{op,i} + \mathbf{e}_{op,i} \bmod p\mathcal{R}_{op}$ for $i \in \{1, 2\}$
 3. $\mathbf{f} \leftarrow \mathcal{B}_1^{k_H}$ s.t. $[\mathbf{f}]_i \neq 0 \ \forall i$, and $\max_{i \in [0, k_H-1]} \|\sigma([\mathbf{f}]_i)\|_\infty \leq B_f$
 4. $\mathbf{e} \leftarrow \mathcal{B}_1^{k_H}$ s.t. $\|\sigma(\mathbf{e}^* \mathbf{e})\|_\infty \leq B_e^2$
 5. $\mathbf{h} \leftarrow \mathbf{e} \odot \mathbf{f} \bmod q\mathcal{R}$
 6. **For** $\text{id}_i \in \mathcal{I}$ ($i = 1, \dots, N$) **do**
 7. $t_i \leftarrow \text{id}_i(x^{k_\pi}) \in \mathcal{R}$
 8. $v_{3,i} \leftarrow \mathcal{D}_{\mathcal{R}, s_2}$
 9. $[v_{1,i} | \mathbf{v}_{2,i}^T]^T \leftarrow \text{NTRU-TSampler}(\mathbf{e}, \mathbf{f}, u - a_3 v_{3,i} \bmod q\mathcal{R}, t_i, s_1, s_2, s_L, s_H)$
 10. **If** $\llbracket \|v_{1,i}\|_2 > B_1 \rrbracket \vee \llbracket \|\mathbf{v}_{2,i}^T | v_{3,i}\|_2 > B_2 \rrbracket$, **go to** 7.

Output: $\text{gpk} = (\mathbf{h}, \mathbf{b}_{op,1}, \mathbf{b}_{op,2})$, $\text{gmsk} = \mathbf{s}_{op,1}$, $\text{gsk} = (\text{id}_i, \mathbf{v}_{2,i}, v_{3,i})_{i \in [N]}$

Algorithm 5.3: GSign(gsk[i], m, gpk, pp)

1. $\mathbf{x} = [\mathbf{s}^T | \mathbf{e}^T]^T \leftarrow \mathcal{D}_{\mathcal{R}_{op}^{2d}, s_{se}}$ s.t. $\|\mathbf{x}\|_2 \leq B_{se}$
 2. $e_1, e_2 \leftarrow \mathcal{D}_{\mathcal{R}_{op}, s_{e1}}$ s.t. $\|e_1\|_2, \|e_2\|_2 \leq B_{e1}$
 3. $\text{ct}_0 \leftarrow \mathbf{A}_{op} \mathbf{s} + \mathbf{e} \bmod p\mathcal{R}_{op}$
 4. $\text{ct}_1 \leftarrow \mathbf{b}_{op,1}^T \mathbf{s} + e_1 + \lfloor p/2 \rfloor \text{id}_i(x^{k_{op}, \pi}) \bmod p\mathcal{R}_{op}$
 5. $\text{ct}_2 \leftarrow \mathbf{b}_{op,2}^T \mathbf{s} + e_2 + \lfloor p/2 \rfloor \text{id}_i(x^{k_{op}, \pi}) \bmod p\mathcal{R}_{op}$
 6. $\mathbf{v}_i \leftarrow [u - (q_L \text{id}_i(x^{k_\pi}) \mathbf{g}_H - \mathbf{h})^T \mathbf{v}_{2,i} - a_3 v_{3,i} \bmod^\pm q\mathcal{R} \mid \mathbf{v}_{2,i}^T \mid v_{3,i}]^T$
 7. **If** $\llbracket \|v_{1,i}\|_2 > B_1 \rrbracket \vee \llbracket \|\mathbf{v}_{2,i}^T | v_{3,i}\|_2 > B_2 \rrbracket \vee \llbracket \text{id} \notin \mathcal{I} \rrbracket$, **abort**
 8. $\pi \leftarrow \text{Prove}((\text{id}_i, \mathbf{v}_i, \mathbf{s}, \mathbf{e}, e_1, e_2); \text{aux} = m)$

Output: $\text{gsig} = (\text{ct}_0, \text{ct}_1, \text{ct}_2, \pi)$

Algorithm 5.4: GVerify(gpk, m, gsig, pp)

1. $b \leftarrow \llbracket \text{Verify}(\pi; \text{aux} = m) = 1 \rrbracket$

Output: b (1 if valid, 0 otherwise)

Algorithm 5.5: GOpen(gmsk, m, gsig, gpk, pp)

1. $a_1 = \text{ct}_1 - \mathbf{s}_{\text{op},1}^T \text{ct}_0 \bmod^\pm p\mathcal{R}_{\text{op}}$
2. **For** $i = 0, \dots, n_{\text{op}} - 1$ **do**
3. $t_{\text{op}} \leftarrow 0 \in \mathcal{R}_{\text{op}}$
4. $\tau_i(t_{\text{op}}) \leftarrow 0$ **if** $|\tau_i(a_1)| < |\tau_i(a_1) \pm p/2|$ **else** $\tau_i(t_{\text{op}}) \leftarrow 1$
5. **If** $\llbracket \text{GVerify}(\text{gpk}, m, \text{gsig}, \text{pp}) = 0 \rrbracket \vee \llbracket t_{\text{op}} \notin \{r(x^{k_{\text{op}},\pi}); r \in \mathcal{I}\} \rrbracket$, **return** $\perp$
6. **Else** $\text{id} \leftarrow t_{\text{op}}(x^{1/k_{\text{op}},\pi}) \in \mathcal{I}$

Output: $\text{id} \in \mathcal{I}$ or $\perp$

Notice we have $\|\llbracket \mathbf{f} \rrbracket_i\|_\infty = 1 < \min(b_H, q_L)^{1/\kappa} / \sqrt{\kappa}$. Hence, the choice of q_L, b_H is compatible with Lemma 2.1, thus ensuring each $\llbracket \mathbf{f} \rrbracket_i$ is a unit modulo q to define $\mathbf{h}$, but also modulo b_H for NTRU-TSampler. Similarly, the tags used by our sampler are of the form $t = \text{id}(x^{k_\pi})$ for $\text{id} \in \mathcal{I}$. It then holds that $\|t\|_\infty = 1 < b_H^{1/\kappa} / \sqrt{\kappa}$, ensuring t is a unit modulo b_H , also needed by NTRU-TSampler.

Zero-Knowledge Proof and Identity Reconciliation. Our scheme relies on the zero-knowledge proof framework of [LNP22] in order to prove knowledge of the certificate $(\text{id}_i, \mathbf{v}_i)$ as well as the well-formedness of the ciphertext. Specifically, we use the non-interactive version under the Fiat-Shamir heuristics and include the signed message m in the random oracle inputs used to produce the different challenges. We call (Prove, Verify) the non-interactive zero-knowledge proof system that results from it. The system proves knowledge of $(\text{id}, \mathbf{v}, \mathbf{s}, \mathbf{e}, e_1, e_2) \in \mathcal{R}_\pi \times \mathcal{R}^{2+k_H} \times \mathcal{R}_{\text{op}}^{2d+2}$ verifying

$$\begin{aligned} & [1|q_L \text{id}(x^{k_\pi}) \mathbf{g}_H^T - \mathbf{h}^T | a_3] \mathbf{v} = u \bmod q\mathcal{R} \\ & \mathbf{A}_{\text{op}} \mathbf{s} + \mathbf{e} = \text{ct}_0 \bmod p\mathcal{R}_{\text{op}} \\ & \mathbf{b}_{\text{op},1}^T \mathbf{s} + \mathbf{e}_1 + \lfloor p/2 \rfloor \text{id}(x^{k_{\text{op}},\pi}) = \text{ct}_1 \bmod p\mathcal{R}_{\text{op}} \\ & \mathbf{b}_{\text{op},2}^T \mathbf{s} + \mathbf{e}_2 + \lfloor p/2 \rfloor \text{id}(x^{k_{\text{op}},\pi}) = \text{ct}_2 \bmod p\mathcal{R}_{\text{op}} \\ & \|\mathbf{v}\|_2 \leq \sqrt{B_1^2 + B_2^2}, \quad \|\llbracket \mathbf{s}^T | \mathbf{e}^T \rrbracket^T\|_2 \leq B_{se}, \quad \|\mathbf{e}_1\|_2, \|\mathbf{e}_2\|_2 \leq B_{e1}, \quad \text{id} \in \mathcal{I} \end{aligned} \tag{1}$$

The instantiation of [LNP22] for this relation and the associated security is given in Appendix A. We note that the proof is carried in the subring $\mathcal{R}_\pi$ of degree n_π defined in GSetup, similarly to prior works [LNPS21, LNP22, JS25b, DFJS26]. We thus design the identity space to directly be a subset of $\mathcal{R}_\pi$ so that the identity represents a single element of the witness, as opposed to prior constructions like [JS25b, DFJS26]. This idea, used in [LNP22], requires us to reconcile this choice of identity space with the signature and encryption components.

One can simply embed the identity $\text{id} \in \mathcal{R}_\pi$ into the larger rings $\mathcal{R}_{\text{op}}$ and $\mathcal{R}$ by $\text{id}(x^{k_{\text{op}},\pi})$ and $\text{id}(x^{k_\pi})$ respectively. Reciprocally, following the subring embedding formalized in [LNPS21, AGJ+24] and recalled in Section 2.1, we can later embed the relations over $\mathcal{R}$ and $\mathcal{R}_{\text{op}}$ into vectors over $\mathcal{R}_\pi$. We call $\theta_{\text{op},\pi}$ the subring embedding from $\mathcal{R}_{\text{op}}$ to $\mathcal{R}_\pi^{k_{\text{op},\pi}}$, and θ_π that from $\mathcal{R}$ to $\mathcal{R}_\pi^{k_\pi}$.

To reconcile the presence of the identity id in these three rings, observe that $\theta_\pi(\text{id}(x^{k_\pi})) = [\text{id}|\mathbf{0}]^T \in \mathcal{R}_\pi^{k_\pi}$, and $\theta_{\text{op},\pi}(\text{id}(x^{k_{\text{op},\pi}})) = [\text{id}|\mathbf{0}]^T \in \mathcal{R}_{\text{op},\pi}^{k_{\text{op},\pi}}$. The ciphertext equations involving id would then become $M_{\theta_{\text{op},\pi}}(\mathbf{b}_{\text{op},i}^T)\theta_{\text{op},\pi}(\mathbf{s}) + \theta_{\text{op},\pi}(e_i) + \lfloor p/2 \rfloor \text{id}|\mathbf{0}]^T = \theta_{\text{op},\pi}(\text{ct}_i) \bmod p\mathcal{R}_\pi$. As for the signature equation over $\mathcal{R}$, the identity is involved in the term $q_L \text{id}(x^{k_\pi}) \mathbf{g}_H^T \mathbf{v}_2$, which is a sum of terms of the form $q_L b_H^i \text{id}(x^{k_\pi}) v_{2,i}$. Applying θ_π results in $q_L b_H^i M_{\theta_\pi}(v_{2,i}) [\text{id}|\mathbf{0}]^T = q_L b_H^i \text{id} \cdot \theta_\pi(v_{2,i})$. We thus managed to express everything in $\mathcal{R}_\pi$ as a function of id and the subring embeddings of the other elements, which then properly reconciles all the components.

This allows for decorrelating the different building blocks of our group signature, and choose more adequate rings for each one. It then leads to a better efficiency while avoiding overshooting the security target.

5.2 Security Analysis

In this section, we show the group signature scheme of Section 5.1 is traceable and CCA-anonymous.

5.2.1 Traceability. Proving the traceability of our scheme starts by following the standard workflow of previous works [dPLS18,LNPS21,LNP22,JS25b]. As the zero-knowledge proof π proves knowledge of a certificate, or rather a short preimage for **NTRU-TSampler**, extracting the proof readily gives a preimage $\mathbf{v}^*$ for the tag defined by the identity id^* of an honest group member. In LWE-based constructions from previous works, the goal is to use $\mathbf{v}^*$ to solve an instance of SIS. This is done by making a guess t^+ on the tag (linked to the identity) that will be targeted by the adversary and defining the group public key as $\mathbf{B} = \mathbf{R}_1 + \mathbf{A}\mathbf{R}_2 + f(t^+)$ so that the gadget component vanishes for $t = t^+$. More precisely, with $f(t^+) = t^+ \mathbf{G}$ or $t^+ \mathbf{G}_H$ depending on the sampler, we would have $\mathbf{A}_{t^+} = [\mathbf{I}|\mathbf{A}] - \mathbf{R}_1 - \mathbf{A}\mathbf{R}_2|\mathbf{A}_3$. By also setting $\mathbf{u} = \mathbf{A}_{t^+} \mathbf{v}^+$ for some hidden preimage $\mathbf{v}^+$, the extraction then gives a SIS solution $\mathbf{v}^* - \mathbf{v}^+$ if the guess is correct, i.e., $t^+ = t^*$. Changing the public key while keeping the ability to produce user signing keys is then handled with a trapdoor switching argument. In [dPLS18,LNPS21,LNP22], this is done by embedding a *full* trapdoor in $\mathbf{A}_3$, e.g., $\mathbf{A}_3 = \mathbf{G} - (\mathbf{S}_1 + \mathbf{A}\mathbf{S}_2)$, whereas only a partial trapdoor is needed in [AGJ⁺24,JS25b,JS26]. One would thus switch (partial) trapdoors from $\mathbf{R}$ to $\mathbf{S}$, change the public key $\mathbf{B}$, and then go back to the original trapdoor, in a possibly progressive hybrid argument.

In our case, we follow the same strategy. We make a guess on the targeted identity id^* and change the public key to cancel the gadget component. We randomly guess id^+ , and define $\mathbf{h} = \mathbf{e} \oslash \mathbf{f} + q_L \text{id}^+(x^{k_\pi}) \mathbf{g}_H$ so that $\mathbf{a}_{\text{id}^+(x^{k_\pi})}^T = [1 | -(\mathbf{e} \oslash \mathbf{f})^T | a_3]$. We can do this change progressively as in [JS26] for the LWE case by resorting to our trapdoor switching result from Section 3.4 in a hybrid argument. Then, instead of reducing from a homogeneous assumption, we directly reduce from R-ISIS. Indeed, the extracted witness directly gives $\mathbf{a}_{\text{id}^*(x^{k_\pi})}^T \mathbf{v}^* = u$. If the guess is correct, i.e., $\text{id}^+ = \text{id}^*$, we get $[1 | -(\mathbf{e} \oslash \mathbf{f})^T | a_3] \mathbf{v}^* = u$ where u is

uniform. Intuitively, under NTRU, the matrix $[1 | -(\mathbf{e} \oslash \mathbf{f})^T | a_3]$ is indistinguishable from uniform which means $\mathbf{v}^*$ is a solution to R-ISIS. However, arguing this last point would require replacing $\mathbf{e} \oslash \mathbf{f}$ by some uniform $\mathbf{h}$, thus loosing the ability to sample preimages and produce user signing keys. Only for this last reduction from R-ISIS do we need a slightly stronger assumption, which captures the intuition of R-ISIS above, but also tackles the peculiarities of being able to give the signing keys of corrupt users to the adversary. At this stage, we provide two different reductions, using either of the assumptions introduced in Section 4. The simplest one uses Known-NTRU-ISIS. Giving the traceability challenger the NTRU structure still permits it to produce the individual signing keys, while finishing the argument on R-ISIS for the matrix $[1 | (\mathbf{e} \oslash \mathbf{f})^T | a_3]$. We can also conclude using the weaker assumption T-NTRU-ISIS where the signing keys would be given to the traceability challenger as part of the T-NTRU-ISIS instance, making it oblivious to the actual structure of $\mathbf{h}$. We insist that both are spiritly very close to R-ISIS over the NTRU matrix $[1 | (\mathbf{e} \oslash \mathbf{f})^T | a_3]$, especially as the preimages do not leak the underlying NTRU structure.

Theorem 5.1. *The group signature of Section 5 is traceable based on the zero-knowledge and soundness properties of (Prove, Verify), and the hardness of NTRU and T-NTRU-ISIS assumptions. More precisely, the scheme is ε_{tr} -traceable with*

$$\varepsilon_{tr} = \varepsilon_{snd} + N \left(Q_s \varepsilon_{zk} + h^{\circ k_H} (\min(\varepsilon_{\text{Known-NTRU-ISIS}}, \varepsilon_{\text{T-NTRU-ISIS}})) \right),$$

$\varepsilon_{snd}, \varepsilon_{zk}$ being the soundness error and zero-knowledge security of (Prove, Verify), $\varepsilon_{\text{Known-NTRU-ISIS}}, \varepsilon_{\text{T-NTRU-ISIS}}$ the hardness bounds of Known-NTRU-ISIS $_{n,q,k_H,B_1,B}$ and T-NTRU-ISIS $_{n,q,k_H,B_1,\mathcal{T},B}$ respectively with $\mathcal{T} = \{(\text{id} - \text{id}^+)(x^{k_\pi}); \text{id} \in \mathcal{I} \setminus \{\text{id}^+\}\}$ for some arbitrary $\text{id}^+ \in \mathcal{I}$, and $B^2 = B_1^2 + B_2^2$, and Q_s is the maximal number of emitted group signature per user key. The function $h^{\circ k_H}$ is to the k_H -th composition power of the function⁵ h defined by

$$h(x) = \varepsilon_{\text{NTRU}} + \mu \left(2\varepsilon_{\text{NTRU}} + \mu (\varepsilon_{\text{NTRU}} + x)^{1-\frac{1}{2\lambda}} \right)^{1-\frac{1}{2\lambda}},$$

with $\varepsilon_{\text{NTRU}}$ the hardness bound of NTRU $_{n,q,B_1}$ and

$$\mu = \left(1 + \frac{\lambda(2\lambda-1)\delta_{\text{gpk}}^2}{(1-\delta_{\text{gpk}})^{2\lambda+1}} \right)^{(N-1)/2\lambda}$$

$$\delta_{\text{gpk}} = \left(\frac{1 + \frac{\varepsilon}{k_H}}{1 - \frac{\varepsilon}{k_H}} \right)^{2k_H} \left(\frac{1 + \varepsilon}{1 - \varepsilon} \right)^{4(1+k_H)(n-1)+8} - 1.$$

Proof. As described in Section 5.2.1, the proof of traceability follows the structure of that of prior works, e.g., [dPLS18,LNPS21,LNP22,JS25b], which we specialize to our setting and our new sampler. In particular, the key difference

⁵ The function h has been introduced in [AGJ⁺24] in the M-LWE case. Their study from [AGJ⁺24, App. A.1] can be adapted to show that $h^{\circ k_H}$ reduces the bit security of the input by only $O(k_H)$ bits. We account for the exact loss in our security estimation.

relies on our new trapdoor switching hybrid argument and reduction to either Known-NTRU-ISIS or T-NTRU-ISIS. At a high-level, the objective is to guess the identity id^* that will be targetted by the adversary and make sure the gadget vanishes in $[1|q_L \text{id}^*(x^{k_\pi}) \mathbf{g}_H^T - \mathbf{h}^T|a_3]$, i.e., by defining $\mathbf{h} = \mathbf{e} \odot \mathbf{f} + q_L \text{id}^*(x^{k_\pi}) \mathbf{g}_H$. To do so while keeping the ability to produce all user signing keys but the targetted user's (which will not be corrupted if the guess is correct), we use a hybrid argument to progressively hide $q_L \text{id}^*(x^{k_\pi}) \mathbf{g}_H$ entry by entry. The latter relies on the trapdoor switching lemma of our new sampler (Lemma 3.5). After that, we directly reduce to Known-NTRU-ISIS, or T-NTRU-ISIS if one requires a harder, albeit q-type, assumption. We now give the formal proof. We use Game_0 to denote the traceability game of Figure 2.1, and introduce hybrid game Game_i until we can reduce traceability from the assumptions of Section 4. For clarity, we use $\text{Adv}_{tr,i}[\mathcal{A}]$ to denote the advantage of a PPT adversary $\mathcal{A}$ in Game_i . We then aim at upper-bounding $\text{Adv}_{tr,0}[\mathcal{A}] = \text{Adv}_{tr}[\mathcal{A}]$.

Game₁. We first extract the proof π from the adversarially chosen signature gsig^* . The extracted witness will then be used to construct a T-NTRU-ISIS solution. More precisely, the challenger extracts a witness $(\text{id}^*, \mathbf{v}^*, \mathbf{s}^*, \mathbf{e}^*, e_1^*, e_2^*)$ that verifies Equation (1). Lemma A.1 then gives

$$\text{Adv}_{tr,0}[\mathcal{A}] \leq \text{Adv}_{tr,1}[\mathcal{A}] + \varepsilon_{\text{snd}}$$

In the process, this discards attacks consisting on cheating in the zero-knowledge proof. In particular, at this stage, it holds that a winning group signature gsig^* satisfies $\text{GOpen}(\text{gmsk}, m^*, \text{gsig}^*, \text{gpk}, \text{pp}) = \text{id}^*$. By Equation (1), we indeed have that $\text{ct}_0 = \mathbf{A}_{\text{op}} \mathbf{s}^* + \mathbf{e}^* \bmod p\mathcal{R}_{\text{op}}$, $\text{ct}_1 = \mathbf{b}_{\text{op},1}^T \mathbf{s}^* + e_1^* + \lfloor p/2 \rfloor \text{id}^*(x^{k_{\text{op},\pi}}) \bmod p\mathcal{R}_{\text{op}}$, for the same id^* that is used in the certificate equation $\mathbf{a}_{\text{id}^*(x^{k_\pi})}^T \mathbf{v}^* = u \bmod q\mathcal{R}$. Hence, $a_1 = \text{ct}_1 - \mathbf{s}_{\text{op},1}^T \text{ct}_0 \bmod p\mathcal{R}_{\text{op}} = e_1^* + \mathbf{e}_{\text{op},1}^T \mathbf{s}^* - \mathbf{s}_{\text{op},1}^T \mathbf{e}^* + \lfloor p/2 \rfloor \text{id}^*(x^{k_{\text{op},\pi}})$. Because $\text{id}^* \in \mathcal{I}$, it means $\text{id}^*(x^{k_{\text{op},\pi}})$ has coefficients in $\{0, 1\}$, and thus its encoding has coefficient in $\{0, \lfloor p/2 \rfloor\}$. The added noise is bounded in infinity norm by $B_{e1} + B_{\text{op}} B_{se}$ by the triangle inequality and Cauchy-Schwarz. Indeed, observe that in power-of-two cyclotomic rings, we have $\|ab\|_\infty = \|M_\tau(a)\tau(b)\|_\infty = \max_i |\langle \tau(a^* x^i), \tau(b) \rangle| \leq \|a\|_2 \|b\|_2$. As a result, the decoding of a_1 gives $t_{\text{op}} = \text{id}^*(x^{k_{\text{op},\pi}})$ which belongs to the correct space as it is an embedding of an element of $\mathcal{I}$. Thus, the returned identity is $\text{id} = t_{\text{op}}(x^{1/k_{\text{op},\pi}}) = \text{id}^*$.

The adversary can then only win Game_1 by producing a signature that opens to an honest user with identity id^* , as we have shown the opening cannot output $\perp$.

Game₂. We now introduce the guess on said honest user, so that the challenger can adapt its behavior depending on which user it believes will be targetted. The challenger then samples $i^+ \leftarrow U([N])$ and defines $\text{id}^+ = \text{id}_{i^+}$. At the very end, it aborts if the output of $\text{GOpen}(\text{gmsk}, m^*, \text{gsig}^*, \text{gpk}, \text{pp})$, i.e., id^* , does not match id^+ . Because, id^+ is uniform in $\mathcal{I}$ and $\mathcal{A}$ is oblivious to the choice of id^+ , we have $\mathbb{P}[\mathcal{A} \text{ wins Game}_2] = \mathbb{P}[\mathcal{A} \text{ wins Game}_1 \wedge \text{id}^+ = \text{id}^*] = \mathbb{P}[\mathcal{A} \text{ wins Game}_1]/N$, which entails

$$\text{Adv}_{tr,1}[\mathcal{A}] = N \text{Adv}_{tr,2}[\mathcal{A}] = \binom{n_{\text{op}}}{w} \text{Adv}_{tr,2}.$$

We now assume i^+ is never queried for corruption (by querying $\mathcal{O}_{\text{Cor}}(i^+)$), as it would mean $\mathcal{A}$ lost the game. In particular, the challenger would abort when receiving the request $\mathcal{O}_{\text{Cor}}(i^+)$.

Game₃. We now wish to hide the identity guess within the group public key $\mathbf{h}$. First, doing so naively cancels the gadget which means we can no longer produce the signing key $\text{usk}[i^+]$. Although we know $\text{usk}[i^+]$ will not be given to the adversary as we have excluded queries to $\mathcal{O}_{\text{Cor}}(i^+)$, the adversary may still query group signatures from user i^+ by querying $\mathcal{O}_{\text{Sign}}(i^+, m)$. As such, before hiding the identity guess within $\mathbf{h}$, we must be able to simulate every signing query to i^+ without using its secret key. The challenger can still provide user i^+ with its identity id^+ , which means the ciphertexts can be properly constructed, while opening to id^+ (which the adversary can check due to it knowing the opening key). However, we want to avoid using $\mathbf{v}_{i^+}$. For that we simulate all the NIZK proofs in the signing queries to i^+ . Hence, when the adversary queries $\mathcal{O}_{\text{Sign}}(i^+, m)$ for some message m , the challenger uses the simulator from Lemma A.1 to generate π . The ciphertext is however generated legitimately using id^+ . Distinguishing the views of $\mathcal{A}$ between **Game₂** and **Game₃** then corresponds to distinguish between at most Q_s simulated proofs and the same number of real ones. As such, we have

$$\text{Adv}_{tr,2}[\mathcal{A}] \leq \text{Adv}_{tr,3}[\mathcal{A}] + Q_s \cdot \varepsilon_{zk},$$

where ε_{zk} is the zero-knowledge security from Lemma A.1.

Tradpoor Switching Hybrid Argument. We now no longer need to produce $\mathbf{v}_{i^+}$, which means we can proceed with our transformation of the group public key $\mathbf{h}$. Our goal is to move to a situation with $\mathbf{h} = \mathbf{e} \oslash \mathbf{f} + q_L \text{id}^+(x^{k_\pi}) \mathbf{g}_H$. Doing so brutally by replacing the k_H instance of NTRU $\mathbf{e} \oslash \mathbf{f}$ to uniform, adding the identity guess, and back to the NTRU instances would prevent us from producing *any* of the signing keys. Instead, we replace successively each entry of $\mathbf{h}$. At the j -th step, we thus move from $\mathbf{h} = \mathbf{e} \oslash \mathbf{f} + q_L \mathbf{g}_{H, \mathbf{t}_j}$ to $\mathbf{h} = \mathbf{e} \oslash \mathbf{f} + q_L \mathbf{g}_{H, \mathbf{t}_{j+1}}$, where $\mathbf{t}_k = (\text{id}^+(x^{k_\pi}) \mathbf{1}_k, \mathbf{0}_{k_H-k})$, where $\mathbf{1}_D$ is the D -dimensional all ones vector. That way, only one component needs to be turned to uniform, and we use the a_3 slot to compensate. The argument is very close to that developed in [AGJ⁺24] and improved in [JS26], except that we now rely on NTRU instances rather than M-LWE ones. One of the key aspects is to have a sampler capable of handling tags with entrywise granularity. This is because at step j , when generating the signing key of user $i \neq i^+$, we need to sample preimages with effective tag

$$\text{id}_i(x^{k_\pi}) \mathbf{1}_{k_H} - \mathbf{t}_j = ((\text{id}_i(x^{k_\pi}) - \text{id}^+(x^{k_\pi})) \mathbf{1}_j, \text{id}_i(x^{k_\pi}) \mathbf{1}_{k_H-j}).$$

This is fortunately handled by our **NTRU-TSampler** as long as all the entries are units modulo $b_H \mathcal{R}$. Let us now formalize the hybrid argument. We define intermediate hybrid games **Game_{j,0}** up to **Game_{j,9}**, for j ranging from 0 to $k_H - 1$, in a way they can properly be chained. In particular, **Game_{0,0}** corresponds to **Game₃**, and for $j \in [0, k_H - 2]$, **Game_{j,9}** = **Game_{j+1,0}**. Let $j \in [0, k_H - 1]$.

Game_{j,0}. We first describe the original hybrid game of step j before introducing further changes. In this game, the challenger proceeds exactly as in **Game₃** by

sampling a uniform guess i^+ , simulating the proofs in signing queries to i^+ , and extracting the proof of $\mathbf{gsig}^*$ in the end. It differs from $\mathbf{Game}_3$ in the definition of the group public key $\mathbf{h}$, and thus in the way it generates the individual signing keys. The algorithm $\mathbf{GKeyGen}$ is modified as follows.

The challenger runs steps 1 through 4 as usual, that is sampling $\mathbf{x}_{\text{op},1}, \mathbf{x}_{\text{op},2}$, defining the opening keys $\mathbf{b}_{\text{op},1}, \mathbf{b}_{\text{op},2}$, and sampling short $\mathbf{f}, \mathbf{e}$ from $\mathcal{B}_1^{k_H}$ satisfying the correct bounds. Then, it defines

$$\mathbf{h} = \mathbf{e} \odot \mathbf{f} + q_L \mathbf{g}_{H,\mathbf{t}_j} \bmod q\mathcal{R} = \begin{bmatrix} e_0 f_0^{-1} + q_L \text{id}^+(x^{k_\pi}) \\ \vdots \\ e_{j-1} f_{j-1}^{-1} + q_L b_H^{j-1} \text{id}^+(x^{k_\pi}) \\ e_j f_j^{-1} \\ \vdots \\ e_{k_H-1} f_{k_H-1}^{-1} \end{bmatrix} \bmod q\mathcal{R}.$$

Note that when $j = 0$, $\mathbf{h} = \mathbf{e} \odot \mathbf{f} \bmod q\mathcal{R}$, exactly as in $\mathbf{Game}_3$. Then, it generates the signing keys of every user but i^+ . For $i \in [N] \setminus \{i^+\}$, it defines $t_i = \text{id}_i(x^{k_\pi})$, samples $v_{3,i}$ from $\mathcal{D}_{\mathcal{R},s_2}$, and calls $\mathbf{NTRU-TSampler}$ to get

$$\begin{bmatrix} v_{1,i} \\ \mathbf{v}_{2,i} \end{bmatrix} = \mathbf{NTRU-TSampler}(\mathbf{e}, \mathbf{f}, u - a_3 v_{3,i} \bmod q\mathcal{R}, t_i \mathbf{1}_{k_H} - \mathbf{t}_j, s_1, s_2, s_L, s_H),$$

This call is properly defined as $t_i \mathbf{1}_{k_H} - \mathbf{t}_j \bmod b_H \mathcal{R} \in (\mathcal{R}_{b_H}^\times)^{k_H}$. Indeed, because $\text{id}_i \neq \text{id}^+$, we have $\|\text{id}_i(x^{k_\pi}) - \text{id}^+(x^{k_\pi})\|_\infty = \|\text{id}_i - \text{id}^+\|_\infty = 1$. We also have $\|\text{id}_i(x^{k_\pi})\|_\infty = 1$. Then, because b_H is a prime such that $b_H \bmod 4\kappa = 2\kappa + 1$ and $b_H^{1/\kappa}/\sqrt{\kappa} > 1$, the invertibility result of [LS18, Cor. 1.2] recalled in Lemma 2.1 ensures all the entries are units modulo b_H . After sampling $v_{1,i}, \mathbf{v}_{2,i}$, it checks the Euclidean norm bounds as usual. For user i^+ , the challenger simply defines the signing key as $\text{id}^+ = \text{id}_{i^+}$. Recall that because it simulates all the proofs π when answering signing queries to i^+ , the challenger does not need to produce a preimage of u . Because $\mathbf{t}_j = \mathbf{0}$ for $j = 0$, it indeed holds that $\mathbf{Game}_3$ and $\mathbf{Game}_{0,0}$ are identical. Hence

$$\text{Adv}_{tr,3}[\mathcal{A}] = \text{Adv}_{tr,0,0}[\mathcal{A}].$$

Game_{j,1}. We now proceed as in $\mathbf{Game}_{j,0}$ except that we change the generation of a_3 to add the gadget component we will need to compensate for generating the preimages. Concretely, instead of sampling a_3 uniformly, it samples $a'_3 \leftarrow U(\mathcal{R}_q)$, and defines $a_3 = q_L b_H^j - a'_3 \bmod q\mathcal{R}$. Because a'_3 is uniform and independent of $q_L b_H^j$, it holds that a_3 is still uniform. As a result, we have

$$\text{Adv}_{tr,j,0}[\mathcal{A}] = \text{Adv}_{tr,j,1}[\mathcal{A}].$$

Game_{j,2}. Let us then hide an NTRU instance within a'_3 . This creates an alternate trapdoor component for the gadget entry b_H^j . We sample e'_j, f'_j from $\mathcal{B}_1$ such that $\|\sigma(f'_j)\|_\infty \leq B_f$, and such that $\|\sigma(\mathbf{e}'^* \mathbf{e}')\|_\infty \leq B_e$, where $\mathbf{e}'$ is obtained

from $\mathbf{e}$ by replacing the entry e_j by e'_j . It then defines $a'_3 = e'_j f_j'^{-1} \bmod q\mathcal{R}$, and then $a_3 = q_L b_H^j - a'_3 = q_L b_H^j - e'_j f_j'^{-1} \bmod q\mathcal{R}$. A distinguisher between $\mathbf{Game}_{j,1}$ and $\mathbf{Game}_{j,2}$ then readily gives a distinguisher for $\mathbf{NTRU}_{n,q,\mathcal{B}_1}$. More precisely, from a distinguisher $\mathcal{D}$ between $\mathbf{Game}_{j,1}$ and $\mathbf{Game}_{j,2}$, we can construct one for $\mathbf{NTRU}_{n,q,\mathcal{B}_1}$. From the $\mathbf{NTRU}$ instance $h \in \mathcal{R}_q$, the reduction uses it to define $a'_3 = h$, and thus $a_3 = q_L b_H^j - h$. The reduction finally returns 1 ($\mathbf{NTRU}$) if $\mathcal{D}$ returns $\mathbf{Game}_{j,2}$, and 0 (uniform) if it returns $\mathbf{Game}_{j,1}$. Indeed, if h is uniform, it perfectly simulates $\mathbf{Game}_{j,1}$, whereas if $h = e' f'^{-1}$, it perfectly simulates $\mathbf{Game}_{j,2}$. The reduction is therefore correct, and we get

$$\text{Adv}_{tr,j,1}[\mathcal{A}] \leq \text{Adv}_{tr,j,2}[\mathcal{A}] + \varepsilon_{\mathbf{NTRU}},$$

$\varepsilon_{\mathbf{NTRU}}$ being the hardness bound of $\mathbf{NTRU}_{n,q,\mathcal{B}_1}$.

$\mathbf{Game}_{j,3}$. We now use our trapdoor switching result for $\mathbf{NTRU-TSampler}$. We consider $\mathbf{e}', \mathbf{f}'$ respectively obtained from $\mathbf{e}, \mathbf{f}$ by changing their $(j+1)$ -th entry e_j, f_j by e'_j, f'_j . We then change the way the challenger generates the $N-1$ preimages to define the signing keys. For $i \in [N] \setminus \{i^+\}$, it samples $[\mathbf{v}_{2,i}]_j \leftarrow \mathcal{D}_{\mathcal{R},s_2}$ and computes $u^{(i)} = u - (q_L \text{id}_i(x^{k_\pi}) b_H^j - h_j) [\mathbf{v}_{2,i}]_j \bmod q\mathcal{R}$, where h_j is the $(j+1)$ -th entry of $\mathbf{h}$. It then samples $(v_{1,i}, [\mathbf{v}_{2,i}]_0, \dots, [\mathbf{v}_{2,i}]_{j-1}, v_{3,i}, [\mathbf{v}_{2,i}]_{j+1}, \dots, [\mathbf{v}_{2,i}]_{k_H-1})$ by calling $\mathbf{NTRU-TSampler}(\mathbf{e}', \mathbf{f}', u^{(i)}, \mathbf{t}^{(i)}, s_1, s_2, s_L, s_H)$, where

$$\mathbf{t}^{(i)} = ((\text{id}_i(x^{k_\pi})) - \text{id}^+(x^{k_\pi})) \mathbf{1}_j, \quad 1, \quad \text{id}_i(x^{k_\pi}) \mathbf{1}_{k_H-j-1}.$$

It checks the bounds on $v_{1,i}$ and $[\mathbf{v}_{2,i}|v_{3,i}]^T$ as usual, and defines $\mathbf{gsk}[i] = (\text{id}_i, \mathbf{v}_{2,i}, v_{3,i})$. Because we have $B_f \geq \max(\max_k \|\sigma(f_k)\|_\infty, \|\sigma(f'_j)\|_\infty)$ and $B_e^2 \geq \|\sigma(\mathbf{e}^* \mathbf{e})\|_\infty, \|\sigma(\mathbf{e}'^* \mathbf{e}')\|_\infty$, we can directly apply our trapdoor switching lemma of Lemma 3.5. It gives that the distribution of $\mathbf{gsk}[i]$ between $\mathbf{Game}_{j,2}$ and $\mathbf{Game}_{j,3}$ are δ^2 -close, where δ is defined from Theorem 3.1 as

$$\delta = \left(\frac{1 + \varepsilon/k_H}{1 - \varepsilon/k_H} \right)^{k_H} \left(\frac{1 + \varepsilon}{1 - \varepsilon} \right)^{2(1+k_H)(n-1)+4}$$

due to having $\alpha_p = (\frac{1+\varepsilon}{1-\varepsilon})^{(1+k_H)(n-1)+1}$ and $\varepsilon_L = \varepsilon$. We can then define $\delta_{\mathbf{gpk}} = \delta^2 - 1$ and use Lemma 2.2 with $\Delta = \delta_{\mathbf{gpk}}, a = 2\lambda$ and $Q = N-1$ (as we change the sampling of $N-1$ preimages) to conclude that

$$\text{Adv}_{tr,j,2}[\mathcal{A}] \leq \underbrace{\left(1 + \frac{\lambda(2\lambda-1)\delta_{\mathbf{gpk}}^2}{(1-\delta_{\mathbf{gpk}})^{2\lambda+1}} \right)^{(N-1)/2\lambda}}_{\mu} \text{Adv}_{tr,j,3}[\mathcal{A}]^{1-1/2\lambda}.$$

$\mathbf{Game}_{j,4}$. We now simulate the $(j+1)$ -th entry of $\mathbf{h}$ as it is no longer used to generate the signing keys. We thus sample $[\mathbf{h}]_j \leftarrow U(\mathcal{R}_q)$ instead of defining it as $e_j f_j^{-1} \bmod q\mathcal{R}$. Using the same argument as for $\mathbf{Game}_{j,2}$, distinguishing $\mathbf{Game}_{j,3}$ and $\mathbf{Game}_{j,4}$ readily solves $\mathbf{NTRU}_{n,q,\mathcal{B}_1}$. Hence, we get

$$\text{Adv}_{tr,j,3}[\mathcal{A}] \leq \text{Adv}_{tr,j,4}[\mathcal{A}] + \varepsilon_{\mathbf{NTRU}},$$

Game_{j,5}. We are then ready to hide the guess id^+ within the $(j+1)$ -th entry of $\mathbf{h}$. Instead of sampling $[\mathbf{h}]_j$ uniformly, it defines it as $[\mathbf{h}]_j = h'_j + q_L b_H^j \text{id}^+(x^{k_\pi}) \bmod q\mathcal{R}$ for some uniform h'_j . As h'_j is uniform and independent of the rest, we still have a uniform entry $[\mathbf{h}]_j$, resulting in

$$\text{Adv}_{tr,j,4}[\mathcal{A}] = \text{Adv}_{tr,j,5}[\mathcal{A}].$$

Game_{j,6}. Because we have now successfully hidden the tag guess, we can undo the changes from **Game_{j,0}** to **Game_{j,4}** in reverse order. We thus reintroduce an NTRU instance within h'_j from the previous game. We then define $[\mathbf{h}]_j = e_j f_j^{-1} + q_L b_H^j \text{id}^+(x^{k_\pi}) \bmod q\mathcal{R}$. Using the same reasoning, it yields

$$\text{Adv}_{tr,j,5}[\mathcal{A}] \leq \text{Adv}_{tr,j,6}[\mathcal{A}] + \varepsilon_{\text{NTRU}},$$

Game_{j,7}. We then go back to producing the signing keys with the trapdoor underlying $\mathbf{h}$, i.e., $\mathbf{e}, \mathbf{f}$. Notice however that, as of **Game_{j,5}**, we now need to sample preimages with an effective tag of $\text{id}_i(x^{k_\pi}) \mathbf{1}_{k_H} - \mathbf{t}_{j+1}$. Concretely, for $i \neq i^+$, it samples $v_{3,i} \leftarrow \mathcal{D}_{\mathcal{R},s_2}$ and computes $u^{(i)} = u - a_3 v_{3,i} \bmod q\mathcal{R}$, and samples $(v_{1,i}, \mathbf{v}_{2,i})$ by calling **NTRU-TSampler**($\mathbf{e}, \mathbf{f}, u^{(i)}, \text{id}_i(x^{k_\pi}) \mathbf{1}_{k_H} - \mathbf{t}_{j+1}, s_1, s_2, s_L, s_H$). The rest is kept unchanged. Using Lemma 3.5 in combination with Lemma 2.2 gives

$$\text{Adv}_{tr,j,6}[\mathcal{A}] \leq \mu \cdot \text{Adv}_{tr,j,7}[\mathcal{A}]^{1-1/2\lambda}.$$

Game_{j,8}. To be able to chain with index $j+1$, we need to prepare the a_3 slot for it to welcome the next gadget component. We thus remove the NTRU instance from a'_3 . We then go back to sampling $a'_3 \leftarrow U(\mathcal{R}_q)$ and defining $a_3 = q_L b_H^j - a'_3 \bmod q\mathcal{R}$. Using the same argument as in **Game_{j,2}**, we have

$$\text{Adv}_{tr,j,7}[\mathcal{A}] \leq \text{Adv}_{tr,j,8}[\mathcal{A}] + \varepsilon_{\text{NTRU}},$$

Game_{j,9}. We finally remove the gadget term from the definition of a_3 , and simply sample $a_3 \leftarrow U(\mathcal{R}_q)$. Both distributions are identical, leading to

$$\text{Adv}_{tr,j,8}[\mathcal{A}] = \text{Adv}_{tr,j,9}[\mathcal{A}],$$

Combining the hybrid games for index j , we get

$$\begin{aligned} \text{Adv}_{tr,j,0}[\mathcal{A}] &\leq \varepsilon_{\text{NTRU}} + \mu \left(2\varepsilon_{\text{NTRU}} + \mu (\varepsilon_{\text{NTRU}} + \text{Adv}_{tr,j,9}[\mathcal{A}])^{1-\frac{1}{2\lambda}} \right)^{1-\frac{1}{2\lambda}} \\ &= h(\text{Adv}_{tr,j,9}[\mathcal{A}]) \end{aligned}$$

At this stage, we can clearly see that **Game_{j,9}** only differs from **Game_{j,0}** in the definition of $\mathbf{h}$, where it is now equal to $\mathbf{e} \odot \mathbf{f} + q_L \mathbf{g}_{H,\mathbf{t}_{j+1}}$ instead of $\mathbf{e} \odot \mathbf{f} + q_L \mathbf{g}_{H,\mathbf{t}_j}$. As a result, we directly have

$$\text{Adv}_{tr,j,9}[\mathcal{A}] = \text{Adv}_{tr,j+1,0}[\mathcal{A}],$$

We can then chain these arguments with j ranging from 0 to $k_H - 1$, and compose the function h to obtain

$$\text{Adv}_{tr,0,0}[\mathcal{A}] \leq h^{\circ k_H}(\text{Adv}_{tr,k_H-1,9}[\mathcal{A}]),$$

where the inequality still holds through compositions due to the non-decreasing behavior of h .

Bounding $\text{Adv}_{tr,k_H-1,9}[\mathcal{A}]$. We conclude the proof by giving two different reductions, one from Known-NTRU-ISIS and the other from T-NTRU-ISIS.

Reduction from Known-NTRU-ISIS. We provide a first reduction which turns an adversary that is successful against $\text{Game}_{k_H-1,9}$ into an attacker against Known-NTRU-ISIS $_{n,q,k_H,B_1,B}$ for $B = \sqrt{B_1^2 + B_2^2}$. The reduction receives the instance $(\mathbf{e}, \mathbf{f}, a_3, u)$ and is asked to produce a solution $\mathbf{v}^*$ such that $[1 | (\mathbf{e} \oslash \mathbf{f})^T | a_3] \mathbf{v}^* = u \bmod q\mathcal{R}$ and $\|\mathbf{v}^*\|_2 \leq B$. For that, it simply runs the challenger of $\text{Game}_{k_H-1,9}$, which it is able to do thanks to the knowledge of $\mathbf{e}$ and $\mathbf{f}$. Indeed, knowing $\mathbf{e}$ and $\mathbf{f}$ allows it to still produce all the signing keys for users $i \neq i^+$. In the end, the challenger extracts the proof in gsig^* to obtain $\text{id}^*, \mathbf{v}^*$ such that $[1 | q_L \text{id}^*(x^{k_\pi}) \mathbf{g}_H^T - \mathbf{h}^T | a_3] \mathbf{v}^* = u \bmod q\mathcal{R}$ with $\|\mathbf{v}^*\|_2 \leq \sqrt{B_1^2 + B_2^2}$. Because the guess id^+ must match id^* for the adversary to win (see Game_2), we have $[1 | -\mathbf{h}'^T | a_3] \mathbf{v}^* = u \bmod q\mathcal{R}$, using the fact that $q_L \text{id}^*(x^{k_\pi}) \mathbf{g}_H - \mathbf{h} = q_L(\text{id}^* - \text{id}^+)(x^{k_\pi}) \mathbf{g}_H - \mathbf{h}' = \mathbf{h}'$, and where $\mathbf{h}' = \mathbf{e} \oslash \mathbf{f}$. As a result, $[v_1^* | -v_2^* | v_3^*]$ is a direct solution to Known-NTRU-ISIS. It then results in

$$\text{Adv}_{tr,k_H-1,9}[\mathcal{A}] \leq \varepsilon_{\text{Known-NTRU-ISIS}},$$

where $\varepsilon_{\text{Known-NTRU-ISIS}}$ is the hardness bound of Known-NTRU-ISIS $_{n,q,k_H,B_1,B}$.

Reduction from T-NTRU-ISIS. We then show a reduction from T-NTRU-ISIS instead. In particular, an attacker for $\text{Game}_{k_H-1,9}$ can also be turned into a solver for T-NTRU-ISIS $_{n,q,k_H,B_1,\mathcal{T},B}$ for $B = \sqrt{B_1^2 + B_2^2}$ and $\mathcal{T} = \{t - t^+; t \in \mathcal{I}(x^{k_\pi}) \setminus \{t^+\}\}$ for some arbitrary $t^+ \in \mathcal{I}(x^{k_\pi})$. Concretely, the tag space is the set of differences between the identities and a unique one represented by t^+ . The difference with the previous reduction lies in the way the traceability challenger is simulated, which is slightly more complex in the T-NTRU-ISIS case. The reduction to $\text{Game}_{k_H-1,9}$ works as follows.

The reduction receives the instance $(\mathbf{a}, u, (t_i, \mathbf{v}_i)_{i \in [N-1]})$, which it parses as $\mathbf{a}^T = [1 | \mathbf{h}'^T | a_3]$ where $\mathbf{h}' = \mathbf{e} \oslash \mathbf{f} \bmod q\mathcal{R}$. Similarly, all the t_i are distinct and of the form $\text{id}_{j_i}(x^{k_\pi}) - \text{id}^+(x^{k_\pi})$ for some $\text{id}^+ \in \mathcal{I}$ common to all i . The challenger first recovers id^+ as

$$\text{id}^+ = \frac{1}{N} \left(\binom{n_\pi - 1}{w - 1} \mathbf{1}_{\mathcal{R}_\pi} - \sum_{i=1}^{N-1} t_i(x^{1/k_\pi}) \right),$$

where $\mathbf{1}_{\mathcal{R}_\pi}$ is the element of $\mathcal{R}_\pi$ whose coefficients are all ones. Let us now prove below why this indeed equals the desired id^+ .

First all the t_i are distinct and non-zero. Because the tags are of the form $(\text{id}_{j_i} - \text{id}^+)(x^{1/k_\pi})$, it entails that all the id_{j_i} are distinct elements of $\mathcal{I} \setminus \{\text{id}^+\}$. Yet, as $\mathcal{I}$ is of size N , it must hold that $\{\text{id}_{j_i}; i \in [N-1]\} \cup \{\text{id}^+\}$ matches the entire set of identities $\mathcal{I}$. We then have

$$\begin{aligned} S &:= \sum_{i \in [N-1]} t_i(x^{1/k_\pi}) = \sum_{i \in [N-1]} (\text{id}_{j_i} - \text{id}^+) = \sum_{i \in [N-1]} \text{id}_{j_i} - (N-1)\text{id}^+ \\ &= \sum_{\text{id} \in \mathcal{I}} \text{id} - N\text{id}^+ \end{aligned}$$

Let us now compute $S_{n_\pi, w} := \sum_{\text{id} \in \mathcal{I}} \text{id}$. For that, we look at its coefficient embedding. For any n', w' , we define $\mathcal{H}_{n', w'} = \{\mathbf{x} \in \{0, 1\}^{n'} : \|\mathbf{x}\|_1 = w'\}$. We then have $S_{n_\pi, w} = \tau^{-1}(\sum_{\mathbf{x} \in \mathcal{H}_{n_\pi, w}} \mathbf{x})$. Let us look at each entry of the latter sum. Let $j \in [n]$. We can partition $\mathcal{H}_{n_\pi, w}$ as

$$\mathcal{H}_{n_\pi, w} = \underbrace{\left\{ \begin{bmatrix} x_1 \\ \vdots \\ x_{j-1} \\ 1 \\ x_{j+1} \\ \vdots \\ x_{n_\pi} \end{bmatrix} ; \begin{bmatrix} x_1 \\ \vdots \\ x_{j-1} \\ x_{j+1} \\ \vdots \\ x_{n_\pi} \end{bmatrix} \in \mathcal{H}_{n_\pi-1, w-1} \right\}}_{A_j} \cup \underbrace{\left\{ \begin{bmatrix} x_1 \\ \vdots \\ x_{j-1} \\ 0 \\ x_{j+1} \\ \vdots \\ x_{n_\pi} \end{bmatrix} ; \begin{bmatrix} x_1 \\ \vdots \\ x_{j-1} \\ x_{j+1} \\ \vdots \\ x_{n_\pi} \end{bmatrix} \in \mathcal{H}_{n_\pi-1, w} \right\}}_{B_j}.$$

Hence, $\sum_{\mathbf{x} \in \mathcal{H}_{n_\pi, w}} \mathbf{x} = \sum_{\mathbf{x} \in A_j} \mathbf{x} + \sum_{\mathbf{x} \in B_j} \mathbf{x}$. Taking the j -th entry then gives $[\sum_{\mathbf{x} \in \mathcal{H}_{n_\pi, w}} \mathbf{x}]_j = \sum_{\mathbf{x} \in A_j} 1 + \sum_{\mathbf{x} \in B_j} 0 = |A_j| = |\mathcal{H}_{n_\pi-1, w-1}| = \binom{n_\pi-1}{w-1}$. As this is true for all j , we get $\sum_{\mathbf{x} \in \mathcal{H}_{n_\pi, w}} \mathbf{x} = \binom{n_\pi-1}{w-1} \mathbf{1}_{n_\pi}$ which entails $\sum_{\text{id} \in \mathcal{I}} \text{id} = \binom{n_\pi-1}{w-1} \mathbf{1}_{\mathcal{R}_\pi}$. We obtain $S = \binom{n_\pi-1}{w-1} \mathbf{1}_{\mathcal{R}_\pi} - N\text{id}^+$, and thus

$$\text{id}^+ = \frac{1}{N} \left(\binom{n_\pi-1}{w-1} \mathbf{1}_{\mathcal{R}_\pi} - S \right) = \frac{1}{N} \left(\binom{n_\pi-1}{w-1} \mathbf{1}_{\mathcal{R}_\pi} - \sum_{i=1}^{N-1} t_i(x^{1/k_\pi}) \right)$$

Now that the challenger recovered id^+ , it can recover all the $\text{id}_i = t_i(x^{1/k_\pi}) + \text{id}^+$ as well. It then defines $\mathbf{h} = \mathbf{h}' + q_L \text{id}^+(x^{k_\pi}) \mathbf{g}_H \bmod q\mathcal{R}$, and uses the u, a_3 from the instance to define the key materials. The signing keys are given as $(\text{id}_i, \mathbf{v}_{2,i}, v_{3,i})$ where the $\mathbf{v}_i$ are parsed as $(v_{1,i}, \mathbf{v}_{2,i}, v_{3,i})$. Note that for all i , we have

$$[1|q_L \text{id}_i(x^{k_\pi}) \mathbf{g}_H^T - \mathbf{h}^T | a_3] \mathbf{v}_i = [1|q_L t_i \mathbf{g}_H^T - \mathbf{h}'^T | a_3] \mathbf{v}_i = u \bmod q\mathcal{R}.$$

by definition of the T-NTRU-ISIS instance. Also, the $\mathbf{v}_i$ given in the instance are sampled using [NTRU-TSampler](#) and thus satisfy the correct distribution. Except for the index i^+ corresponding to id^+ where the proofs are simulated, the signing queries to user i are handled legitimately using $\text{id}_i, \mathbf{v}_i$. At the end, as

in the reduction to Known-NTRU-ISIS, the challenger extracts the proof in gsig^* to obtain $\text{id}^*, \mathbf{v}^*$ such that $[1 | -\mathbf{h}'^T | a_3] \mathbf{v}^* = u \bmod q\mathcal{R}$ with $\|\mathbf{v}^*\|_2 \leq \sqrt{B_1^2 + B_2^2}$ because the guess id^+ is correct (i.e., $\text{id}^+ = \text{id}^*$). Hence, $[v_1^* | -\mathbf{v}_2^* | v_3^*]$ is solution to T-NTRU-ISIS, giving

$$\text{Adv}_{tr, k_H-1, 9}[\mathcal{A}] \leq \varepsilon_{\text{T-NTRU-ISIS}},$$

where $\varepsilon_{\text{T-NTRU-ISIS}}$ is the hardness bound of T-NTRU-ISIS $_{n, q, k_H, \mathcal{B}_1, \mathcal{T}, B}$.

Conclusion. Combining the two reductions, we then have $\text{Adv}_{tr, k_H-1, 9}[\mathcal{A}] \leq \min(\varepsilon_{\text{Known-NTRU-ISIS}}, \varepsilon_{\text{T-NTRU-ISIS}})$, which leads to the final bound on $\text{Adv}_{tr}[\mathcal{A}]$ when propagated in all the advantage inequalities of the previous games. $\square$

5.2.2 CCA-Anonymity. We now continue by proving anonymity. The goal of our proof of anonymity is to show that a group signature does not reveal information on the user that produced it. We do so by ensuring the challenge group signature does not depend on the signing key $(\text{id}, \mathbf{v})$. This relies on the zero-knowledge property, and the IND-CCA2 security of the ciphertext component. The latter is composed of $(\text{ct}_0, \text{ct}_1, \text{ct}_2)$ but also the proof that these elements are well-formed. Its IND-CCA2 security essentially follows the Naor-Yung paradigm [NY90], but with shared randomness as in [BMV16]. Indeed, the Naor-Yung method would use two perfectly independent ciphertexts $(\text{ct}_0, \text{ct}_1)$ and $(\text{ct}'_0, \text{ct}_2)$. In our case, we reuse ct_0 which reduces the amount of randomness hidden in the NIZK. This however creates subtleties in the security proof to deal with this shared randomness. This idea was instantiated in the group signature construction of [DFJS26], and proven secure on a variant of M-LWE called Matrix-Hint M-LWE defined in Section 2.3, which they extended from [ENP24] for this specific use case. The latter admits an efficient reduction from a regular M-LWE assumption, which makes anonymity proven on standard assumptions. Our proof follows the same general ideas, although our parameters for MH-M-LWE are much smaller which results in better security and efficiency. We formalize and prove our anonymity security in Theorem 5.2 below. We call $\mathcal{D}_{\text{op}}$ the distribution over $\mathcal{R}_{\text{op}}^{2d}$ obtained by sampling $\mathcal{B}_1^{2d}$ conditioned on the concentration bounds $B_{\text{op}}, \beta_{\text{op}}$ being verified.

Theorem 5.2. *The group signature of Section 5.1 is CCA-anonymous based on the zero-knowledge and simulation soundness properties of (Prove, Verify), and the MH-M-LWE and M-LWE assumptions. More precisely, the scheme is ε_{an} -CCA-anonymous with*

$$\varepsilon_{\text{an}} = \varepsilon_{\text{snd}} + \varepsilon_{\text{zk}} + 3\varepsilon_{\text{M-LWE}}^{(\text{op})} + 2\varepsilon_{\text{MH-M-LWE}} + \varepsilon_{\text{M-LWE}}^{(\text{ct})},$$

where $\varepsilon_{\text{snd}}, \varepsilon_{\text{zk}}$ are respectively the soundness error and zero-knowledge security of (Prove, Verify), $\varepsilon_{\text{M-LWE}}^{(\text{op})}, \varepsilon_{\text{M-LWE}}^{(\text{ct})}, \varepsilon_{\text{MH-M-LWE}}$ are the respective hardness bounds of M-LWE $_{n_{\text{op}}, d, d, p, \mathcal{D}_{\text{op}}}$, M-LWE $_{n_{\text{op}}, d, d+1, p, \mathcal{D}_{\mathcal{R}_{\text{op}}^{2d}, s_{se}} \times \mathcal{D}_{\mathcal{R}_{\text{op}}, s_{e1}}}$, and the matrix hint version MH-M-LWE $_{n_{\text{op}}, d, d+1, p, \mathcal{D}_{\mathcal{R}_{\text{op}}^{2d}, s_{se}} \times \mathcal{D}_{\mathcal{R}_{\text{op}}, s_{e1}}, \mathcal{D}_{\mathcal{R}_{\text{op}}, s_{e1}}, \mathcal{D}_{\text{op}} \times \{0\}}$.

Proof. After having simulated the zero-knowledge proof, the proof of CCA-anonymity essentially follows the Naor-Yung proof for IND-CCA2 public key encryption, but with some subtleties due to the sharing of randomness across ct_1 and ct_2 . By relying on the MH-M-LWE (which reduces from M-LWE), we can successively simulate ct_1 and ct_2 while being able to answer opening queries. We use Game_0 to denote the CCA-anonymity game of Figure 2.2, and introduce hybrid games Game_i until we can directly show the advantage of an adversary is negligible. For clarity, we use $\text{Adv}_{an,i}[\mathcal{A}]$ to denote the advantage of a PPT adversary $\mathcal{A}$ in Game_i . We then aim at upper-bounding $\text{Adv}_{an,0}[\mathcal{A}] = \text{Adv}_{an}[\mathcal{A}]$.

Game₁. We first evacuate adversaries that would cheat on the zero-knowledge proof when submitting requests to the opening oracle. This happens if the adversary provides a valid group signature containing a falsified zero-knowledge proof, which readily attacks the soundness of (Prove, Verify). Concretely, the challenger keeps the second opening key $\mathbf{s}_{\text{op},2}$ generated with **GKeyGen** at the outset of the game. When a query to $\mathcal{O}_{\text{Open}}$ is made, the challenger proceeds as usual until step 4 (included) of **GOpen**. Then, it computes $a_2 = \text{ct}_2 - \mathbf{s}_{\text{op},2}^T \text{ct}_0 \bmod p\mathcal{R}_{\text{op}}$ and proceed with the same decoding of a_2 to obtain some $t'_{\text{op}} \in \mathcal{R}_{\text{op}}$. If $t_{\text{op}} \neq t'_{\text{op}}$, the opening returns a failure symbol $\perp$. Otherwise, if $t_{\text{op}} = t'_{\text{op}}$, the opening finishes as usual, i.e., checking that **GVerify**(gpk, m , gsig, pp) = 1 and that t_{op} is the embedding of a licit identity $\text{id} \in \mathcal{I}$. If so, it returns id , and returns $\perp$ otherwise.

The only situation where $\mathcal{O}_{\text{Open}}$ would not be correctly simulated with this added condition $t_{\text{op}} = t'_{\text{op}}$ is if the adversary $\mathcal{A}$ managed to produce a valid group signature for which ct_1 and ct_2 are formed from two different identities. In that case, the challenger is able to detect this mismatch and use $\mathcal{A}$ to construct a successful adversary against the soundness of (Prove, Verify). As a result, Lemma A.1 gives

$$|\text{Adv}_{an,0}[\mathcal{A}] - \text{Adv}_{an,1}[\mathcal{A}]| \leq \varepsilon_{\text{snd}}$$

where ε_{snd} is the soundness error defined in Lemma A.1. Moving forward, we can assume all the opening queries are well-formed. In particular, we assume the submitted ct_1, ct_2 decrypt to the same identity id .

Game₂. We then simulate the zero-knowledge proof π in gsig^* so as to remove the dependency in $\text{id}_{i_b}, \mathbf{v}_{i_b}$ but also in the ciphertext randomness. This is essential to pursue the proof and simulate the ciphertexts while providing a valid proof. Hence, when generating gsig^* in **Game₂**, the challenger replaces step 6 of **GSign** by calling the simulator from Lemma A.1. Distinguishing **Game₁** and **Game₂** can then be turned into a distinguisher between a legitimate proof and a simulated one, thus attacking the zero-knowledge security of the proof system. It then gives

$$|\text{Adv}_{an,1}[\mathcal{A}] - \text{Adv}_{an,2}[\mathcal{A}]| \leq \varepsilon_{zk},$$

where ε_{zk} is the zero-knowledge security from Lemma A.1.

Game₃. Because we ensured in **Game₁** that both ciphertexts should decrypt to the same identity in opening queries, we can thus use either one of the opening keys $\mathbf{s}_{\text{op},1}, \mathbf{s}_{\text{op},2}$. We thus decrypt using $\mathbf{s}_{\text{op},1}$ (as in the legitimate use of the scheme) and simulate the second key $\mathbf{b}_{\text{op},2}$. More precisely, in **Game₃**, the challenger samples $\mathbf{b}_{\text{op},2} \leftarrow U(\mathcal{R}_{\text{op},p}^d)$ instead of generating it as an M-LWE instance.

A distinguisher between Game_2 and Game_3 can then immediately be used to solve $\text{M-LWE}_{n_{\text{op}},d,d,p,\mathcal{D}_{\text{op}}}$, where $\mathcal{D}_{\text{op}}$ is $\mathcal{B}_1^{2d}$ conditioned on the concentration bounds B_{op} and β_{op} being verified. It would run the game as usual but using the instance $(\mathbf{A}_{\text{op}}^T, \mathbf{b}_{\text{op},2})$ to define these two elements. It then outputs 1 (M-LWE) if the distinguisher returns Game_2 and 0 (uniform) if it returns Game_3 . Regardless of $\mathbf{b}_{\text{op},2}$, it holds that $\mathbf{A}_{\text{op}}$ is uniform as desired. Then, if $\mathbf{b}_{\text{op},2}$ is uniform, the reduction perfectly simulates Game_3 , whereas if $\mathbf{b}_{\text{op},2} = \mathbf{A}_{\text{op}}^T \mathbf{s}_{\text{op},2} + \mathbf{e}_{\text{op},2}$ for some $(\mathbf{s}_{\text{op},2}, \mathbf{e}_{\text{op},2})$ following $\mathcal{D}_{\text{op}}$, the reduction correctly simulates Game_2 . As such, the reduction is correct and yields

$$|\text{Adv}_{an,2}[\mathcal{A}] - \text{Adv}_{an,3}[\mathcal{A}]| \leq \varepsilon_{\text{M-LWE}}^{(\text{op})},$$

where $\varepsilon_{\text{M-LWE}}^{(\text{op})}$ designates the hardness bound of $\text{M-LWE}_{n_{\text{op}},d,d,p,\mathcal{D}_{\text{op}}}$.

Game₄. We now start simulating ct_2 . Because it shares randomness with ct_0 and ct_1 , we cannot simulate it independently of the rest from M-LWE. We thus rely on the MH-M-LWE assumption to simplify this simulation step, but note that our parameter selection ensures it reduces from M-LWE. Formally speaking, the challenger of Game_4 changes the way it generates $\text{ct}_0, \text{ct}_1, \text{ct}_2$ in gsig^* as follows. It samples $(\mathbf{s}, \mathbf{e}, e_1, e_2)$ as usual, and then samples $\mathbf{u} \leftarrow U(\mathcal{R}_{\text{op},p}^d)$ and $u_2 \leftarrow U(\mathcal{R}_{\text{op},p})$. It then defines $\text{ct}_0 = \mathbf{u}$, $\text{ct}_2 = u_2 + \lfloor p/2 \rfloor \text{id}_{i_b}(x^{k_{\text{op}},\pi}) \bmod p\mathcal{R}_{\text{op}}$, and

$$\text{ct}_1 = \mathbf{s}_{\text{op},1}^T \mathbf{u} + e_1 + \mathbf{e}_{\text{op},1}^T \mathbf{s} - \mathbf{s}_{\text{op},1}^T \mathbf{e} + \lfloor p/2 \rfloor \text{id}_{i_b}(x^{k_{\text{op}},\pi}) \bmod p\mathcal{R}_{\text{op}}.$$

We now show that a distinguisher $\mathcal{D}$ between Game_3 and Game_4 can be turned into a distinguisher $\mathcal{B}$ for $\text{MH-M-LWE}_{n_{\text{op}},d,d+1,\mathcal{D}_{\mathcal{R}_{\text{op}}^{2d},s_{se}} \times \mathcal{D}_{\mathcal{R}_{\text{op}},s_{e1}}, \mathcal{D}_{\mathcal{R}_{\text{op}},s_{e1}}, \mathcal{D}_{\text{op}} \times \{0\}}$.

On input the instance $(\mathbf{A}, \mathbf{b}, \mathbf{m}, z) \in \mathcal{R}_{\text{op},p}^{(d+1) \times d} \times \mathcal{R}_{\text{op},p}^{d+1} \times \mathcal{R}_{\text{op}}^{2d+1} \times \mathcal{R}_{\text{op}}$, it first parses it as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{\text{op}} \\ \mathbf{b}_{\text{op},2}^T \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \mathbf{u} \\ u_2 \end{bmatrix}, \mathbf{m} = \begin{bmatrix} \mathbf{e}_{\text{op},1} \\ -\mathbf{s}_{\text{op},1} \\ 0 \end{bmatrix}.$$

It then uses $\mathbf{A}_{\text{op}}, \mathbf{b}_{\text{op},2}, \mathbf{s}_{\text{op},1}, \mathbf{e}_{\text{op},1}$ to define the key material, and uses $(\mathbf{u}, u_2, z)$ to define $\text{ct}_0 = \mathbf{u}$, $\text{ct}_2 = u_2 + \lfloor p/2 \rfloor \text{id}_{i_b}(x^{k_{\text{op}},\pi}) \bmod p\mathcal{R}_{\text{op}}$, and $\text{ct}_1 = \mathbf{s}_{\text{op},1}^T \mathbf{u} + z + \lfloor p/2 \rfloor \text{id}_{i_b}(x^{k_{\text{op}},\pi}) \bmod p\mathcal{R}_{\text{op}}$ when generating gsig^* . It then calls the distinguisher $\mathcal{D}$, and returns 1 (MH-M-LWE) if $\mathcal{D}$ returns Game_3 and 0 (uniform) otherwise. Let us now show the reduction is correct.

First, regardless of $\mathbf{b}$, it holds that $\mathbf{A}_{\text{op}}, \mathbf{b}_{\text{op},2}$ are uniform as required, and that $\mathbf{e}_{\text{op},1}, \mathbf{s}_{\text{op},1}$ is correctly distributed as $\mathcal{D}_{\text{op}}$ is invariant by permutation and negation. It also holds that $z = e_1 + \mathbf{m}^T [\mathbf{s}^T | \mathbf{e}^T | e_2]^T$ for some $e_i \leftarrow \mathcal{D}_{\mathcal{R}_{\text{op}},s_{e1}}$ and $(\mathbf{s}, \mathbf{e}) \leftarrow \mathcal{D}_{\mathcal{R}_{\text{op}}^{2d},s_{se}}$. Now, if $\mathbf{b}$ is uniform, then $\mathcal{B}$ properly simulates Game_4 as $z = e_1 + \mathbf{e}_{\text{op},1}^T \mathbf{s} - \mathbf{s}_{\text{op},1}^T \mathbf{e}$. If on the other hand $\mathbf{b} = \mathbf{A} \mathbf{s} + [\mathbf{e}^T | e_2] \bmod p\mathcal{R}_{\text{op}}$, we directly get that ct_0, ct_2 are correctly simulated. As for ct_1 , we have

$$\begin{aligned} \text{ct}_1 &= \mathbf{s}_{\text{op},1}^T (\mathbf{A}_{\text{op}} \mathbf{s} + \mathbf{e}) + e_1 + \mathbf{e}_{\text{op},1}^T \mathbf{s} - \mathbf{s}_{\text{op},1}^T \mathbf{e} + \lfloor p/2 \rfloor \text{id}_{i_b}(x^{k_{\text{op}},\pi}) \bmod p\mathcal{R}_{\text{op}} \\ &= \mathbf{s}_{\text{op},1}^T \mathbf{A}_{\text{op}} \mathbf{s} + \mathbf{e}_{\text{op},1}^T \mathbf{s} + e_1 + \lfloor p/2 \rfloor \text{id}_{i_b}(x^{k_{\text{op}},\pi}) \bmod p\mathcal{R}_{\text{op}} \end{aligned}$$

$$= \mathbf{b}_{\text{op},1}^T \mathbf{s} + e_1 + \lfloor p/2 \rfloor \text{id}_{i_b}(x^{k_{\text{op}},\pi}) \bmod p\mathcal{R}_{\text{op}},$$

as in **Game**₃. The reduction is therefore correct. It yields

$$|\text{Adv}_{an,3}[\mathcal{A}] - \text{Adv}_{an,4}[\mathcal{A}]| \leq \varepsilon_{\text{MH-M-LWE}}$$

where the parameter $\varepsilon_{\text{MH-M-LWE}}$ denotes the hardness bound of the problem $\text{MH-M-LWE}_{n_{\text{op}},d,d+1,\mathcal{D}_{\mathcal{R}_{\text{op}}^{2d},s_{se}} \times \mathcal{D}_{\mathcal{R}_{\text{op}},s_{e1}},\mathcal{D}_{\mathcal{R}_{\text{op}},s_{e1}},\mathcal{D}_{\text{op}} \times \{0\}}$.

Game₅. We then simulate ct_2 without the identity id_{i_b} , by sampling ct_2 from $U(\mathcal{R}_{\text{op},p})$ directly. Because in **Game**₄, u_2 is uniform and independent of the term $\lfloor p/2 \rfloor \text{id}_{i_b}(x^{k_{\text{op}},\pi})$, it holds that ct_2 also follows the uniform distribution. As a consequence, it holds that

$$\text{Adv}_{an,4}[\mathcal{A}] = \text{Adv}_{an,5}[\mathcal{A}]$$

Game₆. We now proceed in reverse order, while leaving ct_2 untouched as it is already independent of the user identity and the bit b . More precisely, in **Game**₆, the challenger samples $(\mathbf{s}, \mathbf{e}, e_1, e_2)$ as usual, and construct $\text{ct}_0 = \mathbf{A}_{\text{op}} \mathbf{s} + \mathbf{e} \bmod p\mathcal{R}_{\text{op}}$ and $\text{ct}_1 = \mathbf{b}_{\text{op},1}^T \mathbf{s} + e_1 + \lfloor p/2 \rfloor \text{id}_{i_b}(x^{k_{\text{op}},\pi}) \bmod p\mathcal{R}_{\text{op}}$ as before, but still samples ct_2 uniformly. Using the same reduction as in **Game**₄, it holds that

$$|\text{Adv}_{an,5}[\mathcal{A}] - \text{Adv}_{an,6}[\mathcal{A}]| \leq \varepsilon'_{\text{MH-M-LWE}} \leq \varepsilon_{\text{MH-M-LWE}}$$

where $\varepsilon'_{\text{MH-M-LWE}}$ is the hardness bound of $\text{MH-M-LWE}_{n,d,d,p,\mathcal{D}_{\mathcal{R}_{\text{op}}^{2d},s_{se}},\mathcal{D}_{\mathcal{R}_{\text{op}},s_{e1}},\mathcal{D}_{\text{op}},1}$. The inequality $\varepsilon'_{\text{MH-M-LWE}} \leq \varepsilon_{\text{MH-M-LWE}}$ is due to the trivial reduction that discards $(d+1)$ -th sample. This is possible because the hint features a 0 in front of e_2 .

Game₇. We then re-insert an opening key $\mathbf{s}_{\text{op},2}$ within $\mathbf{b}_{\text{op},2}$ by reverting the change from **Game**₃. The challenger then samples $(\mathbf{s}_{\text{op},2}, \mathbf{e}_{\text{op},2})$ from $\mathcal{D}_{\text{op}}$ and defines $\mathbf{b}_{\text{op},2} = \mathbf{A}_{\text{op}}^T \mathbf{s}_{\text{op},2} + \mathbf{e}_{\text{op},2} \bmod p\mathcal{R}_{\text{op}}$. Using the exact same argument as in **Game**₃ gives

$$|\text{Adv}_{an,6}[\mathcal{A}] - \text{Adv}_{an,7}[\mathcal{A}]| \leq \varepsilon_{\text{M-LWE}}^{(\text{op})}.$$

Game₈. To proceed with simulating ct_1 as we did ct_2 , we need to remove the use of $\mathbf{s}_{\text{op},1}$. However, because it is used to perform the opening queries, we first need to switch keys to $\mathbf{s}_{\text{op},2}$ when decrypting the ciphertexts. We then change the opening queries by instead computing $a_2 = \text{ct}_2 - \mathbf{s}_{\text{op},2}^T \text{ct}_0 \bmod p\mathcal{R}_{\text{op}}$, and decoding a_2 in the usual way to obtain some t_{op} . The rest remains unchanged. This correctly simulates the opening queries because we ensured at the beginning of the proof that all the opening queries are well-formed. As a result, decrypting either ciphertexts yield the same result and thus the same view for the adversary. Hence,

$$\text{Adv}_{an,7}[\mathcal{A}] = \text{Adv}_{an,8}[\mathcal{A}].$$

Game₉. We then simulate the first opening key $\mathbf{b}_{\text{op},1}$, as we did in **Game**₃. More precisely, $\mathbf{b}_{\text{op},1}$ is now sampled uniformly in $\mathcal{R}_{\text{op},p}^d$. Using the reduction from M-LWE of **Game**₃, we get

$$|\text{Adv}_{an,8}[\mathcal{A}] - \text{Adv}_{an,9}[\mathcal{A}]| \leq \varepsilon_{\text{M-LWE}}^{(\text{op})}.$$

Game₁₀. We now simulate the first ciphertext ct_1 , as we did the second in **Game₄**. When generating the signature gsig^* , the challenger proceeds as in the previous game except it no longer samples $(\mathbf{s}, \mathbf{e}, e_1, e_2)$ and samples $\mathbf{u} \leftarrow U(\mathcal{R}_{\text{op},p}^d)$ and $u_1 \leftarrow U(\mathcal{R}_{\text{op},p})$ instead. It then defines $\text{ct}_0 = \mathbf{u}$, $\text{ct}_1 = u_1 + \lfloor p/2 \rfloor \text{id}_{i_b}(x^{k_{\text{op},\pi}}) \bmod p\mathcal{R}_{\text{op}}$, and still samples ct_2 uniformly. At first glance, we seem to rely on the same argument as in **Game₄**. Though, notice a small difference is that ct_1 and ct_2 are now uncorrelated due to ct_2 being perfectly uniform and independent of ct_1 . It means we no longer need to account for the correlated randomness we had before. We can thus argue the games are indistinguishable under regular M-LWE. More precisely, from a distinguisher $\mathcal{D}$ between **Game₉** and **Game₁₀**, we can construct one for M-LWE $_{n_{\text{op}},d,d+1,p,\mathcal{D}_{\mathcal{R}_{\text{op}}^{2d},s_{se}} \times \mathcal{D}_{\mathcal{R}_{\text{op}},s_{e1}}}$. From the M-LWE instance $(\mathbf{A}, \mathbf{b}) \in \mathcal{R}_{\text{op},p}^{(d+1) \times d} \times \mathcal{R}_{\text{op},p}^{d+1}$, the reduction parses it as $\mathbf{A} = [\mathbf{A}_{\text{op}}^T | \mathbf{b}_{\text{op},1}]^T$ and $\mathbf{b} = [\mathbf{u}^T | u_1]^T$, and uses $\mathbf{A}_{\text{op}}, \mathbf{b}_{\text{op},1}$ to define the key materials. It then uses $\mathbf{u}, u_1$ to generate gsig^* as described above. The reduction finally returns 1 (M-LWE) if $\mathcal{D}$ returns **Game₉**, and 0 (uniform) if it returns **Game₁₀**.

Regardless of $\mathbf{b}$, the elements $\mathbf{A}_{\text{op}}, \mathbf{b}_{\text{op},1}$ are uniformly distributed as expected. Then, when $\mathbf{b}$ is uniform, the reduction perfectly simulates **Game₁₀**. On the other hand, when $\mathbf{b} = \mathbf{A}\mathbf{s} + [\mathbf{e}^T | e_1]^T \bmod p\mathcal{R}_{\text{op}}$, it holds that $\text{ct}_0 = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod p\mathcal{R}_{\text{op}}$, and $\text{ct}_1 = \mathbf{b}_{\text{op},1}^T \mathbf{s} + e_1 + \lfloor p/2 \rfloor \text{id}_{i_b}(x^{k_{\text{op},\pi}}) \bmod p\mathcal{R}_{\text{op}}$, exactly as in **Game₉**. The reduction thus properly simulates both games depending on the distribution of $\mathbf{b}$ and is thus correct. It entails

$$|\text{Adv}_{an,9}[\mathcal{A}] - \text{Adv}_{an,10}[\mathcal{A}]| \leq \varepsilon_{\text{M-LWE}}^{(\text{ct})},$$

where $\varepsilon_{\text{M-LWE}}^{(\text{ct})}$ is the hardness bound of M-LWE $_{n_{\text{op}},d,d+1,p,\mathcal{D}_{\mathcal{R}_{\text{op}}^{2d},s_{se}} \times \mathcal{D}_{\mathcal{R}_{\text{op}},s_{e1}}}$.

Game₁₁. We perform the final change by simply generating ct_1 uniformly with $\text{ct}_1 \leftarrow U(\mathcal{R}_{\text{op},p})$. As in **Game₅**, the situation in **Game₁₀** uses a perfectly random independent mask u_1 to hide $\text{id}_{i_b}(x^{k_{\text{op},\pi}})$, meaning ct_1 was already perfectly uniform in **Game₁₀**. Thence,

$$\text{Adv}_{an,10}[\mathcal{A}] = \text{Adv}_{an,11}[\mathcal{A}]$$

Bounding $\text{Adv}_{an,11}[\mathcal{A}]$. In **Game₁₁**, the user index i_b and thus the bit b are perfectly hidden to the adversary because the challenge signature gsig^* is not constructed from b . We then have $\text{Adv}_{an,11}[\mathcal{A}] = 0$, which then leads to

$$\text{Adv}_{an}[\mathcal{A}] = \text{Adv}_{an,0}[\mathcal{A}] \leq \varepsilon_{\text{snd}} + \varepsilon_{zk} + 3\varepsilon_{\text{M-LWE}}^{(\text{op})} + 2\varepsilon_{\text{MH-M-LWE}} + \varepsilon_{\text{M-LWE}}^{(\text{ct})},$$

as desired. $\square$

6 Performance

We finally showcase the performance of our group signature scheme described in Section 5, in terms of both size and runtime. We in particular give an implementation of **NTRU-TSampler** to show its concrete efficiency compared to **TSampler** from [JS26], and also provide an optimization of the arithmetic backend that results in a very substantial runtime improvement in preimage sampling.

6.1 Parameter Selection

Let us first describe our parameter selection methodology. We aim at reaching $\lambda = 128$ bits of classical security for a group size of $N \geq 2^{32}$ for a fairer comparison with prior constructions. We then select parameters as in [GSetup](#), while choosing $n, n_{\text{op}}, n_{\pi}, b_H, q_L, k_H, \varepsilon, \gamma, \varepsilon_{\text{op}}, \alpha$ (as well as all the parameters of the proof system of [LNP22](#)) so as to reach sufficient hardness bounds for all the identified assumptions and reach (close to) λ bits of security for ε_{tr} and ε_{an} . We aim for the smallest parameters to reach better compactness. A complete example parameter set is given in [Appendix B](#) for completeness.

Hardness Bounds. The hardness of all our assumptions are estimated using the lattice estimator [\[APS15\]](#), using a sieving cost model for the hardness bounds of $2^{-\log_2(\sqrt{3/2}\beta+16.4)} \approx 2^{-0.292\beta-16.4}$, where β is the minimal BKZ blocksize. Several works have adopted this cost model over the Core-SVP one, judging it better reflected the complexity of lattice reduction. Notice that ≈ 112 bits of Core-SVP (which is the usually targetted security) equates to $\lambda = 128$ in this more realistic cost model. The several M-LWE assumptions are then estimated as described, and so is the M-SIS assumption needed for the soundness of [\(Prove, Verify\)](#).

We still zoom in on four of our assumptions, namely NTRU, MH-M-LWE, Known-NTRU-ISIS, and T-NTRU-ISIS. First, we instantiate NTRU with a distribution producing very small elements so as to reach a compact gadget trapdoor. In particular, $\mathbf{e}$ and $\mathbf{f}$ follow a centered binomial distribution and are therefore ternary. In addition, our modulus q is in practice larger than in schemes like Falcon, which then places us in the overstretched regime of NTRU. We indeed have $q > n^{2.484}/\exp(6)$, which then means DSD (Dense Sublattice Discovery) attacks [\[DvW21\]](#) are likely to be more efficient than uSVP attacks. We evaluate the hardness of NTRU in this setting using the lattice estimator, which accounts for both kinds of attacks. It highlights that DSD attacks are indeed more efficient, but our parameters still provide a comfortable security margin.

For the second assumption, we specify that our parameters are compatible with the reduction of [Lemma 2.6](#) from M-LWE to MH-M-LWE. We thus bound $\varepsilon_{\text{MH-M-LWE}}$ from above by $\varepsilon_{\text{M-LWE}} + \frac{\varepsilon_{\text{op}}}{1-\varepsilon_{\text{op}}}$, where $\varepsilon_{\text{M-LWE}}$ is the hardness bound of $\text{M-LWE}_{n_{\text{op}}, d, d, p, \mathcal{D}_{\mathcal{R}_{\text{op}}^{2d}, \sqrt{2}\eta_{\text{op}}}}$. The latter is then assessed using the lattice estimator just like all the other M-LWE assumptions.

Then, the hardness of Known-NTRU-ISIS is estimated using $\text{R-ISIS}_{n, 2+k_H, q, B}$, which states that given $(\mathbf{a}^T = [1|\mathbf{h}^T|a_3], u) \in \mathcal{R}_q^{2+k_H} \times \mathcal{R}_q$ with $\mathbf{h}, a_3, u$ uniform in $\mathcal{R}_q$, one has to find some $\mathbf{v}^* \in \mathcal{R}^{2+k_H}$ such that $\mathbf{a}^T \mathbf{v}^* = u \bmod q\mathcal{R}$ and $\|\mathbf{v}^*\|_2 \leq B$. In our case, we design our scheme so that $B < q$ so as to thwart the trivial attacks on Known-NTRU-ISIS consisting on completing the NTRU trapdoor to do preimage sampling as in Falcon, as discussed in [Section 4.2](#). As described in the latter, the NTRU structure does not seem to give an advantage in solving the R-ISIS instance as it is incompatible with preimage sampling. In a similar way, the hardness of T-NTRU-ISIS is assessed with that of $\text{R-ISIS}_{n, 2+k_H, q, B}$ as well. The preimages indeed do not seem to significantly simplify the R-ISIS

instance, as opposed to the NTRU-ISIS_f assumption where one can use them to perform lattice reduction over and hope to find a short enough solution for NTRU-ISIS_f. In addition, ignoring the preimages would lead to an assumption that straightforwardly reduces from NTRU _{n,q,B_1} , and R-ISIS _{$n,2+k_H,q,B$} . However, we discussed in Section 4.2 that the NTRU structure itself does not readily solve the problem as opposed to NTRU-ISIS_f. We would thus not even need to consider the hardness of NTRU. Nevertheless, because our NTRU assumption still needs to be hard for our trapdoor switching argument as described above, we conservatively estimate $\varepsilon_{\text{Known-NTRU-ISIS}}$ and $\varepsilon_{\text{T-NTRU-ISIS}}$ by $k_H \varepsilon_{\text{NTRU}} + \varepsilon_{\text{R-ISIS}}$, which corresponds to simulating $\mathbf{h}$ uniformly under NTRU and then relying on R-ISIS.

6.2 Sampler Implementation and Runtime

We provide two C implementations⁶ of the sampler component of our group signature to allow a comparison with [DFJS26] which relies on the sampler of [JS26]. This provides an example to benchmark our sampler against theirs. The first implementation uses the same libraries and arithmetic backend, and already outperforms that of [JS26]. The second implementation significantly improves the first using a different handling of floating-point polynomials needed in the key generation and the sampler itself.

Issuer Key Generation. The issuer key generation is very different from that of [JS26]. In their case, the secret key is $(\mathbf{R}_1, \mathbf{R}_2)$ sampled from a binomial distribution, and their spectral norm must be bounded by some value B_R . This last point is done in their implementation by approximating the norm using an iterated power method as

$$\|\mathbf{R}_j\|_2^2 \approx \max_{i \in [n/2]} \frac{\sigma_i([\mathbf{R}_j \mathbf{R}_j^*]^{33})_{1,1}}{\sigma_i([\mathbf{R}_j \mathbf{R}_j^*]^{32})_{1,1}},$$

where σ_i is the i -th component of the canonical embedding σ . In addition to the FFTs needed to compute the σ_i , this method first requires computing $(\mathbf{R}_j \mathbf{R}_j^*)^k$ but without reduction modulo q , that is in $\mathcal{R}$, which results in very large integer coefficients of up to 420 bits [AGJ⁺24, App. B]. Computations over high-precision integers then significantly hinder performance of the norm check. Overall, as one needs to repeat it for both matrices, their key generation involves computing two powers of $d \times d$ matrices over $\mathcal{R}$ in high-precision integers, and 4 half FFTs over these high-precision integers.

In our case, our secret key $(\mathbf{f}, \mathbf{e})$ is also taken from a binomial distribution, but the spectral norm computation is much simpler. For $\mathbf{f}$, we simply need to compute $\max_j \|\sigma([\mathbf{f}]_j)\|_\infty = \max_j \max_{i \in [n/2]} |\sigma_i([\mathbf{f}]_j)|$, which requires only $k_H = 2$ half FFTs over ternary integers. For $\mathbf{e}$, we need to compute the spectral norm of $\mathbf{e}^* \mathbf{e}$. Because $\mathbf{e}$ is ternary, the infinity norm of $\mathbf{e}^* \mathbf{e}$, which is equal to $\|\mathbf{e}\|_2^2$, remains bounded by a small value $nk_H < q/2$. As such, we can compute $\mathbf{e}^* \mathbf{e}$ in $\mathcal{R}_q$ directly while ensuring no wraparound occurs. After that, we need to compute one extra

⁶ <https://github.com/ntrutsampler-gsig/ntrutsampler-gsig>

half FFT to compute $\|\sigma(\mathbf{e}^*\mathbf{e})\|_\infty = \max_{i \in [n/2]} |\sigma_i(\mathbf{e}^*\mathbf{e})|$. Actually, because $\mathbf{e}^*\mathbf{e}$ is in $\mathcal{K}_{\mathbb{R}}^{++}$, the $\sigma_i(\mathbf{e}^*\mathbf{e})$ are positive reals and we then only need to compute their real parts. This is however negligible in the runtime. In the end, we only need to compute $k_H + 1 = 3$ half FFTs over very small integers, and we in particular avoid the need for high-precision integers.

The key generation also involves the precomputations for the perturbation sampling which consist in floating-point polynomial operations including multiplications, conjugates and inverses. Our first implementation, using Flint [FLI23] for the FFTs and such floating-point operations, improves the issuer key generation runtime upon [JS26] by a factor of 7 to 30. Switching the floating-point backend for the FFTs and polynomial operations gains an extra factor of roughly 6. Looking at the absolute runtimes, we improved the issuer key generation time from 190 ms to 2.8 ms on average, and with much less variation (the maximum runtime of [JS26] can shoot up to 700 ms, while ours remains around 3 ms).

Perturbation Sampling. Our sampler also slightly outperforms that of [JS26] in the perturbation sampling and signature generation. The improvement however remains quite small, which is explained by the `SampleFz` routine from the perturbation sampler. Theirs makes $2d = 8$ calls to this ring sampler, over a ring of degree $n = 256$ (in addition to $nd(k_H + 1) = 3072$ spherical zero-centered sampling over $\mathbb{Z}$). Ours instead makes $k_H + 1 = 3$ calls to the ring sampler but for a degree of $n = 1024$. It means we have more recursive steps in total, all involving splitting/merging of polynomials but most importantly multiplication and inversion of polynomials with floating-points coefficients. Hence, although we do not require the extra 3072 $\mathbb{Z}$ -sampling, the extra call to `SampleFz` seems to dilute the benefits of having a smaller dimension. Profiling however shows that most of the computation time is dedicated to these high-precision arithmetic operations. In particular, the current arithmetic backend, for floating-point polynomial operations in [JS26] and our first implementation, is Flint [FLI23], which seems to suffer from stability issues in those computations. Indeed, even though [AGJ⁺24] showed that the IEEE standard double precision was theoretically sufficient, using such precision in Flint for the `SampleFz` routine seems to cause precision errors. These errors no longer occur when increasing the precision to roughly 200 bits, which then requires intensive operations and the use of “arbitrary precision reals”. Improving the floating-point polynomial arithmetic to reduce the precision would then likely lead to dramatic improvements in the perturbation sampling, which is currently the bottleneck of the sampler.

Our second implementation aims at fixing this very issue, without any further optimization. We overhaul the `poly_real` backend of our first implementation (inherited from [JS26]) to instead work with the native C type `double`, providing standard precision floating-points. Not only does this confirm the theoretical claim of [AGJ⁺24] that standard double precision is sufficient for `SampleFz`, but it also confirms our intuition that a better handling of floating-points should drastically improve the runtimes. We indeed observe that our perturbation sampling is 65 times faster, and the overall sampler 26 times faster (which includes the gadget sampling that is almost not affected by this new implementation).

The verification of the preimages constituting the user keys in our group signature is also improved by a factor of 9 due to the reduced dimension and the fact that we work over an NTRU structure rather than an M-LWE one.

Procedure		Time (ms)			
		mean	med	min	max
Issuer Key Generation	[JS26, Alg. 4.2]	189.804	133.694	127.496	692.575
	❶ GKeyGen st. 3-5	18.554	18.050	17.827	22.609
	❷ GKeyGen st. 3-5	2.854	2.832	2.799	3.162
Perturbation Sampling	[JS26, Alg. 3.3]	51.937	51.826	50.309	52.884
	❶ PerturbationSampler	53.041	52.942	52.350	55.048
	❷ PerturbationSampler	0.810	0.812	0.783	0.870
Gadget Sampling	[JS26, Alg. 3.2, st. 3-14]	1.669	1.651	1.597	1.870
	❶ NTRU-TSampler st. 3-10	1.181	1.176	1.144	1.442
	❷ NTRU-TSampler st. 3-10	1.122	1.124	1.077	1.161
Full Sampler	[JS26, Alg. 3.2]	55.254	55.247	54.286	55.606
	❶ NTRU-TSampler	53.750	53.545	53.184	55.117
	❷ NTRU-TSampler	2.070	2.061	2.020	2.241
Issuer Signing	[JS26, Alg. 4.3]	55.308	55.322	54.426	56.088
	❶ GKeyGen st. 7-10	54.337	54.346	53.905	55.376
	❷ GKeyGen st. 7-10	2.547	2.546	2.442	2.628
User Verification	[JS26, Alg. 4.4]	0.563	0.564	0.746	0.595
	❶ GSign st. 6-7	0.067	0.066	0.062	0.073
	❷ GSign st. 6-7	0.062	0.061	0.059	0.068

Table 6.1. Benchmark results in milliseconds (ms) over 1000 executions. We take the timings of [JS26], but other parameterizations of the sampler, e.g., in [DFJS26], may result in larger timings. ❶ corresponds to our first implementation, and ❷ to our second implementation with C native floating-points instead of Flint’s arbitrary precision reals.

6.3 Size Performance and Comparison with Prior Work

We give the size estimates of our group signature scheme and the reached security in Table 6.2. We also give these estimates for a CPA-anonymous variant. In the latter, the anonymity proof does not need to rely on the MH-M-LWE assumption and can thus use a ciphertext noise that does not follow a Gaussian distribution. In particular, we can choose a better parameterization of the encryption part to use a PKE close to Kyber.

Our scheme reaches small key and group signature sizes, while having very efficient key generation, signing, verification, and opening procedures. In addition, our scheme is directly proven in the CCA-anonymous model with a rather small overhead of less than 3 KB on the signature size. We compare our scheme to the state-of-the-art lattice group signatures of [LNPS21,LNP22,LN22,JS25b,DFJS26] for a similar N in Table 6.3.

Compared to prior constructions based on the same paradigm as ours of proving knowledge of a certificate while providing an encryption of the identity, i.e., [LNPS21,LNP22,JS25b], our scheme obtains the smallest sizes on all

	Key Sizes		Group Signature Size			Security	
	$ \text{gpk} $	$ \text{gsk}[i] $	$ \text{ct} $	$ \pi $	Total	λ_{an}	λ_{tr}
CPA	5.50 KB	4.68 KB	1.63 KB	35.04 KB	36.67 KB	160	127
CCA	7.28 KB	4.68 KB	2.50 KB	37.11 KB	39.61 KB	126	127

Table 6.2. Key and signature size estimates in kilobytes (KB) for our group signature scheme. We give the bit security for anonymity (λ_{an}) and traceability (λ_{tr}).

metrics while retaining an efficient key generation, signing, and verification procedures. The last scheme based on this paradigm, due to Lyubashevsky and Nguyen [LN22], achieves sizes that are roughly twice as compact in the CPA-anonymous construction. We note however this is at the expense of a computationally intensive signature generation. This is mostly due to the fact that their system is based on proving knowledge of a Falcon signature (s_1, s_2) that verifies $s_1 + hs_2 = \mathcal{H}(\text{id})$. To avoid proving the extended output function evaluation $\mathcal{H}(\text{id})$, they instead prove knowledge of some (s_1, s_2) such that $s_1 + hs_2 \in \{\mathcal{H}(\text{id}_i); i \in [N]\}$, where the latter set is then public. They prove this statement using a compact OR-proof, but said proof requires performing a number of rather complex operations which grows linearly with the number of users. As a contrast, our signing time is constant in the number of users N , which makes our construction much more scalable and runtime efficient than theirs.

In the opposite direction, the recent construction of [DFJS26] aims at reaching small sizes by complexifying the key generation instead of the signing. Although they showcase more compact signature sizes than [LNPS21, LNP22, JS25b], their technique relies on delegating lattice bases to each users which has several shortcomings. First, trapdoor delegation naturally deteriorates the quality of the original one, which in turn requires (much) larger parameters. This increase then leads to a significantly larger group public key size. Second, because the user key now consists of a full basis and not just a small certificate or preimage, the size of said key is much larger than in the BMW approach. On these two aspects alone, our scheme is $29\times$ and $79\times$ more compact, while also reaching group signatures roughly 30% smaller. The third aspect is with regard to the users' key generation complexity. Each user key consists of a full-rank *matrix* of preimages generated with the sampler of [JS26]. Producing this matrix alone multiplies the timings reported in Table 6.1 of [JS26] by 36 for their parameters (and this is not accounting for the fact that the sampler parameters in [DFJS26] are actually larger). The most concerning actually is that this matrix is not a basis of the desired lattice, and should be transformed into one. This operation unfortunately requires computing an HNF in high dimension, which is extremely runtime-intensive, albeit polynomial-time. Each user key generation is then likely to be in the order of (at least) several seconds, while ours only requires sampling a single preimage, which takes 2.5 ms on average with our second implementation, as reported in Table 6.1.

	KeyGen	Sign	gpk	gsk[i]	gsig
[LNPS21]	Easy	Easy	96.00 KB (+ ?)	90.65 KB	203.00 KB (+ ?)
[LNP22]	Easy	Easy	48.25 KB (+0.75)	26.98 KB	98.02 KB (+6.47)
[LN22]	Easy	Complex	3.17 KB (+ ?)	2.50 KB	19.40 KB (+ ?)
[JS25b]	Easy	Easy	29.72 KB (+1.22)	11.82 KB	75.71 KB (+6.47)
[DFJS26]	Complex	Easy	159.72 KB (+3.50)	369.84 KB	52.49 KB (+1.38)
Ours	Easy	Easy	5.50 KB (+1.78)	4.68 KB	36.67 KB (+2.94)

Table 6.3. Comparison of lattice-based static group signatures. The KeyGen and Sign columns refer to the complexity of the key generation and user signing respectively. The added terms in parenthesis corresponds to the overhead incurred by the CCA-anonymous version. When no such version is documented, we leave it unspecified (?).

Acknowledgments

We thank Olivier Sanders for interesting insights on the T-NTRU-ISIS and the Known-NTRU-ISIS assumptions.

References

- AGJ⁺24. S. Argo, T. Güneysu, C. Jeudy, G. Land, A. Roux-Langlois, and O. Sanders. Practical Post-Quantum Signatures for Privacy. In *CCS*, 2024.
- APS15. M. R. Albrecht, R. Player, and S. Scott. On the Concrete Hardness of Learning With Errors. *J. Math. Cryptol.*, 2015.
- Ban93. W. Banaszczyk. New Bounds in Some Transference Theorems in the Geometry of Numbers. *Math. Ann.*, 1993.
- BBS04. D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. In *CRYPTO*, 2004.
- BCC04. E. F. Brickell, J. Camenisch, and L. Chen. Direct Anonymous Attestation. In *CCS*, 2004.
- BDK⁺18. J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In *EuroS&P*, 2018.
- BJRW23. K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen. On the Hardness of Module Learning with Errors with Short Distributions. *J. Cryptol.*, 2023.
- BL07. E. Brickell and J. Li. Enhanced Privacy ID: a Direct Anonymous Attestation Scheme with Enhanced Revocation Capabilities. In *WPES*, 2007.
- BLNS23. J. Bootle, V. Lyubashevsky, N. K. Nguyen, and A. Sorniotti. A Framework for Practical Anonymous Credentials from Lattices. In *CRYPTO*, 2023.
- BMV16. S. Biagioni, D. Masny, and D. Venturi. Naor-Yung Paradigm with Shared Randomness and Applications. In *SCN*, 2016.
- BMW03. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In *EUROCRYPT*, 2003.
- BS04. D. Boneh and H. Shacham. Group Signatures with Verifier-Local Revocation. In *CCS*, 2004.
- CGM19. Y. Chen, N. Genise, and P. Mukherjee. Approximate Trapdoors for Lattices and Smaller Hash-and-Sign Signatures. In *ASIACRYPT*, 2019.

- CL01. J. Camenisch and A. Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In *EUROCRYPT*, 2001.
- CvH91. D. Chaum and E. van Heyst. Group Signatures. In *EUROCRYPT*, 1991.
- DFJS26. P. Delhom, P.-A. Fouque, C. Jeudy, and O. Sanders. Lattice Group Signatures, Revisited. *IACR Cryptol. ePrint Arch.*, page 847, 2026.
- DLP14. L. Ducas, V. Lyubashevsky, and T. Prest. Efficient Identity-Based Encryption over NTRU Lattices. In *ASIACRYPT*, 2014.
- dPLS18. R. del Pino, V. Lyubashevsky, and G. Seiler. Lattice-Based Group Signatures and Zero-Knowledge Proofs of Automorphism Stability. In *CCS*, 2018.
- DvW21. L. Ducas and W. P. J. van Woerden. NTRU Fatigue: How Stretched is Overstretched? In *ASIACRYPT*, 2021.
- ENP24. T. Espitau, G. Niot, and T. Prest. Flood and Submerse: Distributed Key Generation and Robust Threshold Signature from Lattices. In *CRYPTO*, 2024.
- EWY23. T. Espitau, A. Wallet, and Y. Yu. On Gaussian Sampling, Smoothing Parameter and Application to Signatures. In *ASIACRYPT*, 2023.
- FLI23. FLINT Team. FLINT: Fast Library for Number Theory, 2023. Version 3.0.0, <https://flintlib.org>.
- GKV10. S. Dov Gordon, J. Katz, and V. Vaikuntanathan. A Group Signature Scheme from Lattice Assumptions. In *ASIACRYPT*, 2010.
- GL20. N. Genise and B. Li. Gadget-Based iNTRU Lattice Trapdoors. In *INDOCRYPT*, 2020.
- GM18. Nicholas Genise and Daniele Micciancio. Faster Gaussian Sampling for Trapdoor Lattices with Arbitrary Modulus. In *EUROCRYPT*, 2018.
- HPS98. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. In *ANTS*, 1998.
- Int16. Intel. A Cost-Effective Foundation for End-to-End IoT Security, White Paper. <https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/intel-epid-iot-security-white-paper.pdf>, 2016.
- ISO13. ISO/IEC. ISO/IEC 20008-2:2013 Information Technology — Security Techniques — Anonymous Digital Signatures — Part 2: Mechanisms using a Group Public Key. <https://www.iso.org/standard/56916.html>, 2013.
- JRS24. C. Jeudy, A. Roux-Langlois, and O. Sanders. Phoenix: Hash-and-Sign with Aborts from Lattice Gadgets. In *PQCrypto*, 2024.
- JS25a. C. Jeudy and O. Sanders. Improved Lattice Blind Signatures from Recycled Entropy. In *CRYPTO*, 2025.
- JS25b. C. Jeudy and O. Sanders. Worst-Case Lattice Sampler with Truncated Gadgets and Applications. In *ASIACRYPT*, 2025.
- JS26. C. Jeudy and O. Sanders. Tag-Friendly Lattice Sampler and Applications. *IACR Cryptol. ePrint Arch.*, page 083, 2026.
- Kle00. P. N. Klein. Finding the closest lattice vector when it’s unusually close. In *SODA*, 2000.
- LN22. V. Lyubashevsky and N. K. Nguyen. BLOOM: Bimodal Lattice One-Out-of-Many Proofs and Applications. *ASIACRYPT*, 2022.
- LNP22. V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General. *CRYPTO*, 2022.

- LNPS21. V. Lyubashevsky, N. K. Nguyen, M. Plançon, and G. Seiler. Shorter Lattice-Based Group Signatures via "Almost Free" Encryption and Other Optimizations. In *ASIACRYPT*, 2021.
- LS15. A. Langlois and D. Stehlé. Worst-case to Average-case Reductions for Module Lattices. *DCC*, 2015.
- LS18. V. Lyubashevsky and G. Seiler. Short, Invertible Elements in Partially Splitting Cyclotomic Rings and Applications to Lattice-Based Zero-Knowledge Proofs. In *EUROCRYPT*, 2018.
- MG02. D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: A Cryptographic Perspective*. Kluwer Academic Publishers, 2002.
- MP12. D. Micciancio and C. Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *EUROCRYPT*, 2012.
- MR07. D. Micciancio and O. Regev. Worst-Case to Average-Case Reductions Based on Gaussian Measures. *SIAM J. Comput.*, 2007.
- NY90. M. Naor and M. Yung. Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *STOC*, 1990.
- Pei10. C. Peikert. An Efficient and Parallel Gaussian Sampler for Lattices. In *CRYPTO*, 2010.
- Pei16. C. Peikert. A Decade of Lattice Cryptography. *Found. Trends Theor. Comput. Sci.*, 2016.
- PFH⁺20. T. Prest, P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. *FALCON. Tech. rep.*, 2020. Available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>.
- Pre17. T. Prest. Sharper Bounds in Lattice-Based Cryptography Using the Rényi Divergence. In *ASIACRYPT*, 2017.
- PS21. A. Pellet-Mary and D. Stehlé. On the Hardness of the NTRU Problem. In *ASIACRYPT*, 2021.
- ST21. O. Sanders and J. Traoré. EPID with Malicious Revocation. In *CT-RSA*, 2021.
- TCG15. TCG. <https://trustedcomputinggroup.org/authentication/>, 2015.
- YJW23. Y. Yu, H. Jia, and X. Wang. Compact Lattice Gadget and Its Applications to Hash-and-Sign Signatures. In *CRYPTO*, 2023.

A Instantiating the Zero-Knowledge Proof

The zero-knowledge proof relation described in Section 5.1 can be instantiated in the framework of [LNP22, LN22]. The proof is carried in the ring $\mathcal{R}_\pi$ as described in the latter section. We plug our relation in their framework, with standard optimizations (compression of commitments, packing of garbage terms, bimodal rejection sampling, exact norm proofs with four-square decompositions) and more recent ones from [JS25a, DFJS26] using elliptic rejection sampling for inhomogeneous witnesses. We then only state the security result below. For more details on the proof protocol itself, we refer to previous instantiation of the framework, e.g., [LNP22, BLNS23, AGJ⁺24, JS25a]. Ours is extremely similar, albeit adapted to our relation.

Lemma A.1 (Security of (Prove, Verify)). Let $m_2, \rho, \tilde{\eta}$ be in $\mathbb{N}^\times$, and define the challenge space $\mathcal{C}(\rho, \tilde{\eta})$ as

$$\mathcal{C}(\rho, \tilde{\eta}) = \left\{ c \in \mathcal{R}_\pi : c^* = c \quad \wedge \quad \|c\|_\infty \leq \rho \quad \wedge \quad \|c^{64}\|_1^{1/64} \leq \tilde{\eta} \right\}.$$

Let $M_{1,1}, M_{1,2}, M_2, M_3$ be in $(1, \infty)$ be rejection rates, and let $\alpha_x = \sqrt{\pi / \ln M_x}$ be the associated rejection sampling slacks. We divide the witness into the two subwitnesses $\mathbf{s}_{1,1} = (\text{id}, \mathbf{s}, \mathbf{e}, \mathbf{b}, a_{se})$ and $\mathbf{s}_{1,2} = (\mathbf{v}, e_1, e_2, a_v, a_{e1}, a_{e2})$, of respective dimensions $m_{1,1} = 2dk_{\text{op},\pi} + 3$ and $m_{1,2} = (2 + k_H)k_\pi + 2k_{\text{op},\pi} + 3$ when embedded in $\mathcal{R}_\pi$. The elements a_x are ring elements of four-square decompositions to prove ℓ^2 norms exactly, and $\mathbf{b}$ is the bimodal sign committed alongside the witness. Then, define $B_{s11} = \sqrt{w + B_{se}^2 + 1}$, $B_{s12} = \sqrt{B_1^2 + B_2^2 + 2B_{e1}^2}$ to be bounds on $\mathbf{s}_{1,1}$ and $\mathbf{s}_{1,2}$ respectively. We also define the following lifting bounds

$$\begin{aligned} B_{j,v} &= \frac{B_1}{q} + \frac{1}{2}n\sqrt{1 + k_H}B_2 + w\sqrt{\frac{1 - 1/q_H^2}{b_H^2 - 1}}B_2 + \frac{1}{2}\sqrt{n} \\ B_{j,0} &= \sqrt{n_{\text{op}}d}\sqrt{\frac{n_{\text{op}}d}{4} + \frac{1}{p^2}}B_{se} + \frac{1}{2}\sqrt{n_{\text{op}}d} \\ B_{j,1} &= \frac{B_{e1}}{p} + \frac{1}{2}n_{\text{op}}\sqrt{d}B_{se} + \frac{\lfloor p/2 \rfloor}{p}\sqrt{w} + \frac{1}{2}\sqrt{n_{\text{op}}}, \end{aligned}$$

necessary when lifting the mod- q and mod- p equations in $\mathcal{R}_\pi$. We then define the Gaussian widths of the masked openings by $\sigma_{1,1} = \alpha_{1,1}\tilde{\eta}B_{s11}$, $\sigma_{1,2} = \alpha_{1,2}\tilde{\eta}B_{s12}$, $\sigma_2 = \alpha_2\tilde{\eta}\sqrt{n_\pi m_2}$, and $\sigma_3 = \alpha_3\sqrt{337}\sqrt{B_{s11}^2 + B_{s12}^2 + B_{j,v}^2 + B_{j,0}^2 + 2B_{j,1}^2}$ for the approximate range proof component. We also define approximate range proof bound $B_{arp} = \frac{2}{\sqrt{26}}c_{256}\sigma_3\sqrt{256}$. Then, let d_π, ℓ, q_π be in $\mathbb{N}^\times$ and satisfying the following conditions.

$$\begin{aligned} q_\pi &\text{ prime s.t. } q_\pi = 5 \bmod 8, \\ q_\pi &> \max(B_1^2 + B_2^2, B_{se}^2, B_{e1}^2, w) \\ q_\pi &> B_{arp}^2 - \min(B_1^2 + B_2^2, B_{se}^2, B_{e1}^2, w) \\ q_\pi &> \max(41n_\pi(m_{1,1} + m_{1,2} + k_\pi + k_{\text{op},\pi}(d + 2))B_{arp}, w + \sqrt{n_\pi w}) \\ q_\pi &> \left(\sqrt{w}q_L\sqrt{\frac{q_H^2 - 1}{b_H^2 - 1}} + \sqrt{1 + \frac{n(1 + k_H)q^2}{4}} \right) \sqrt{B_1^2 + B_2^2} + q \left(\frac{1}{2} + B_{arp} \right) \\ q_\pi &> \sqrt{1 + \frac{n_{\text{op}}dp^2}{4}}B_{se} + p \left(\frac{1}{2} + B_{arp} \right) \\ q_\pi &> B_{e1} + \lfloor p/2 \rfloor + p \left(\frac{1}{2}\sqrt{n_{\text{op}}d}B_{se} + \frac{1}{2} + B_{arp} \right). \end{aligned}$$

For the commitment compression, we also let γ be an even divisor of $q_\pi - 1$, and D be the largest integer such that $2^{D-1}pn_\pi < \gamma$. Then, (Prove, Verify) is knowledge

sound with an expected-polynomial-time extractor with soundness error

$$\varepsilon_{snd} = \frac{2}{|\mathcal{C}(\rho, \tilde{\eta})|} + q_\pi^{-n_\pi/2} + q_\pi^{-2\ell} + 2^{-\lambda} + \varepsilon_{\text{M-SIS}}^{(snd)}$$

and zero-knowledge with loss $\varepsilon_{zk} = \varepsilon_{\text{M-LWE}}^{(zk)} + \text{negl}(\lambda)$. The loss $\varepsilon_{\text{M-SIS}}^{(snd)}$ is the hardness bound of $\text{M-SIS}_{n_\pi, d_\pi, m_1+m_2, q_\pi, \beta_\pi}$ where β_π is defined as

$$4\tilde{\eta}\sqrt{4c_{n_\pi m_1,1}^2\sigma_{1,1}^2n_\pi m_{1,1} + 4c_{n_\pi m_1,2}^2\sigma_{1,2}^2n_\pi m_{1,2} + \left(2c_{n_\pi m_2}\sigma_2\sqrt{n_\pi m_2} + (2^D\tilde{\eta} + \gamma)\sqrt{n_\pi d_\pi}\right)^2},$$

while $\varepsilon_{\text{M-LWE}}^{(zk)}$ is the hardness bound of $\text{M-LWE}_{n_\pi, m_2-d_\pi-\lfloor 256/n_\pi \rfloor - \ell - 1, m_2, q_\pi, \mathcal{B}_1, \mathcal{B}_1}$.

Following the framework of [LNP22], and using the rANS encoding for discrete Gaussian vectors, the size of non-interactive zero-knowledge proof π from **GSign** can be estimated by the following formula. The different parameters in the formula ($d_\pi, q_\pi, D, \ell, \rho, m_{1,1}, m_{1,2}, m_2, \sigma_{1,1}, \sigma_{1,2}, \sigma_2, \sigma_3$) are defined in Lemma A.1. An example parameter set satisfying all the conditions of said lemma is provided in Table B.2, along with the proof size $|\pi|$ and the associated security $\varepsilon_{snd}, \varepsilon_{zk}$ in Table B.3.

$$\begin{aligned} |\pi| &= n_\pi d_\pi (\lceil \log_2 q_\pi \rceil - D + 2.25) + n_\pi \left(\frac{256}{n_\pi} + 2\ell + 1 \right) \lceil \log_2 q_\pi \rceil \\ &+ n_\pi \lceil \log_2(2\rho + 1) \rceil + n_\pi m_{1,1} \left(\frac{1}{2} + \log_2 \sigma_{1,1} \right) + n_\pi m_{1,2} \left(\frac{1}{2} + \log_2 \sigma_{1,2} \right) \\ &+ n_\pi (m_2 - d_\pi) \left(\frac{1}{2} + \log_2 \sigma_2 \right) + 256 \left(\frac{1}{2} + \log_2 \sigma_3 \right). \end{aligned}$$

B Parameter Set

Symbol	Description	Value
λ	Security parameter	128
n	Sampler ring degree ($\mathcal{R}$)	1024
n_{op}	Encryption ring degree ($\mathcal{R}_{\text{op}}$)	256
n_{π}	Zero-knowledge proof ring degree ($\mathcal{R}_{\pi}$)	64
Parameters of NTRU-TSampler		
q	Modulus	$123877 \approx 2^{16.92}$
k_H	Gadget dimension	2
b_H	Gadget base	13
q_L, q_H	Low and High Moduli	733, 169
ε	Smoothing loss for NTRU-TSampler	2^{-40}
s_L	Gadget sampling widths	2458.821
s_H		44.033
γ	Gaussian width slack	$\sqrt{2}$
s_1	Gaussian widths in NTRU-TSampler	5415.680
s_2		3997.432
w	Hamming weight of tags	8
N	Group size	$4426165368 \approx 2^{32.04}$
B_1	User key bounds in GKeyGen	77331.846
B_2		94392.431
Opening Parameters		
d	Encryption module rank	3
p	Encryption modulus	$57347 \approx 2^{15.81}$
B_{op}	Euclidean norm of opening keys	31.859
β_{op}	Spectral norm of opening keys	45.894
ε_{op}	Smoothing loss for MH-M-LWE reduction	2^{-131}
α	Gaussian width slack for MH-M-LWE reduction	1.88
s_{se}	Gaussian width for $(\mathbf{s}, \mathbf{e})$	14.912
s_{e1}	Gaussian width for e_1, e_2	781.354
B_{se}	Tail bound on $(\mathbf{s}, \mathbf{e})$	255.648
B_{se}	Tail bound on e_1, e_2	6189.751

Table B.1. Example parameter set for the group signature of Section 5.

Symbol	Description	Value
λ	Security parameter	128
n_π	Proof system ring degree	64
k_π	Subring embedding dimension ($\mathcal{R} \rightarrow \mathcal{R}_\pi$)	16
$k_{\text{op},\pi}$	Subring embedding dimension ($\mathcal{R}_{\text{op}} \rightarrow \mathcal{R}_\pi$)	4
d_π	Module rank	16
q_π	Proof system modulus	363721328254934123189
ℓ	Soundness amplification dimension	1
$m_{1,1}$	Subwitness dimensions	27
$m_{1,2}$		75
m_2	Dimension of ABDLOP randomness	78
ρ	Infinity norm of challenges	8
$\tilde{\eta}$	Manhattan-like norm of challenges	93
$ \mathcal{C}(\rho, \tilde{\eta}) $	Size of challenge space	$\approx 2^{129}$
M_x	Rejection sampling repetition rates	$\sqrt{2}$
α_x	Rejection sampling slacks	3.011
$\sigma_{1,1}$	First subwitness rejection sampling width	71586.773
$\sigma_{1,2}$	Second subwitness rejection sampling width	34255019.341
σ_2	Second rejection sampling width	19783.241
σ_3	Third rejection sampling width	4632247388.970
γ	Mask commitment compression parameter	74677186
D	Witness commitment compression parameter	18
B_{s11}	Subwitness bounds	255.666
B_{s12}		122338.757
$B_{j,v}$	Lifting term bounds	83766429.011
$B_{j,0}$		2720550.416
$B_{j,1}$		56687.367
B_{arp}	Proven approximate range proof bound	19071479445.888

Table B.2. Example parameter set for (Prove, Verify).

Symbol	Reference	BKZ- β	Value
Hardness Bounds			
$\varepsilon_{\text{M-LWE}}^{(\text{op})}$	Theorem 5.2	384	$2^{-128.71}$
$\varepsilon_{\text{M-LWE}}^{(\text{ct})}$	Theorem 5.2	657	$2^{-208.56}$
$\varepsilon_{\text{M-LWE}}$	Lemma 2.6 (for MH-M-LWE)	564	$2^{-181.36}$
$\varepsilon_{\text{MH-M-LWE}}$	Theorem 5.2	-	$2^{-131.00}$
$\varepsilon_{\text{NTRU}}$	Theorem 5.1	525	$2^{-170.00}$
$\varepsilon_{\text{R-ISIS}}$	Section 6.1	636	$2^{-202.42}$
$\varepsilon_{\text{Known-NTRU-ISIS}}^{\text{T-NTRU-ISIS}}$	Theorem 5.1 ($k_H \varepsilon_{\text{NTRU}} + \varepsilon_{\text{R-ISIS}}$)	-	$2^{-170.00}$
$\varepsilon_{\text{M-LWE}}^{(zk)}$	Lemma A.1	492	$2^{-160.30}$
$\varepsilon_{\text{M-SIS}}^{(snd)}$	Lemma A.1	390	$2^{-130.47}$
Security			
ε_{zk}	Lemma A.1	-	$2^{-160.30}$
ε_{snd}	Lemma A.1	-	$2^{-128.40}$
ε_{tr}	Theorem 5.1	-	$2^{-127.34}$
ε_{an}	Theorem 5.2	-	$2^{-126.49}$

Symbol	Description	Size
Key Sizes		
$ \text{ipk} $	Issuer public key $\text{ipk} = (\text{seed}, \mathbf{h})$	4384 B
$ \text{isk} $	Issuer secret key $\text{isk} = (\mathbf{e}, \mathbf{f})^\dagger$	1024 B
$ \text{opk} $	Opener public key $\text{opk} = (\text{seed}, \mathbf{b}_{\text{op},1}, \mathbf{b}_{\text{op},2})$	3104 B
$ \text{osk} $	Opener secret key $\text{osk} = \mathbf{s}_{\text{op},1}$	192 B
$ \text{gpk} $	Group public key $\text{gpk} = (\text{seed}, \mathbf{h}, \mathbf{b}_{\text{op},1}, \mathbf{b}_{\text{op},2})$	7456 B
$ \text{gmsk} $	Group manager secret key $\text{gmsk} = \mathbf{s}_{\text{op},1}$	192 B
$ \text{gsk}[i] $	User secret key $\text{gsk}[i] = (\text{id}, \mathbf{v}_2, v_3)$	4795 B
Group Signature Sizes		
$ \text{ct} $	Ciphertext $\text{ct} = (\text{ct}_0, \text{ct}_1, \text{ct}_2)$	2560 B
$ \pi $	Zero-Knowledge proof π	38001 B
$ \text{gsig} $	Group signature $\text{gsig} = (\text{ct}, \pi)$	40561 B

Table B.3. Security and efficiency estimates for the parameters of Tables B.1 and B.2.

† Does not include the precomputed sampling material for [NTRU-TSampler](#).