MuSig-L: Lattice-Based Multi-Signature With Single-Round Online Phase*

Cecilia Boschini[1], Akira Takahashi[2], and Mehdi Tibouchi[3]

[1] Technion and Reichman University, Israel
cecilia.bo@cs.technion.ac.il
[2] Aarhus University, Denmark
takahashi@cs.au.dk
[3] NTT Corporation, Japan
mehdi.tibouchi.br@hco.ntt.co.jp

August 10, 2022

Abstract. Multi-signatures are protocols that allow a group of signers to jointly produce a single signature on the same message. In recent years, a number of practical multi-signature schemes have been proposed in the discrete-log setting, such as MuSig2 (CRYPTO'21) and DWMS (CRYPTO'21). The main technical challenge in constructing a multi-signature scheme is to achieve a set of several desirable properties, such as (1) security in the plain public-key (PPK) model, (2) concurrent security, (3) low online round complexity, and (4) key aggregation. However, previous lattice-based, post-quantum counterparts to Schnorr multi-signatures fail to satisfy these properties.

In this paper, we introduce MuSig-L, a lattice-based multi-signature scheme simultaneously achieving these design goals for the first time. Unlike the recent, round-efficient proposal of Damgård et al. (PKC'21), which had to rely on lattice-based trapdoor commitments, we do not require any additional primitive in the protocol, while being able to prove security from the standard module-SIS and LWE assumptions. The resulting output signature of our scheme therefore looks closer to the usual Fiat–Shamir-with-abort signatures.

* An extended abstract appeared at CRYPTO 2022. This is the full version.

Table of Contents

1 Introduction

A multi-signature is a primitive that allows a group of signers holding individual key pairs $(\mathsf{sk}_1, \mathsf{pk}_1), \ldots, (\mathsf{sk}_n, \mathsf{pk}_n)$ to jointly produce a signature on a message μ of their choice. A number of multi-signatures have been proposed in recent years, mainly motivated by several new real-world applications such as cryptocurrencies. Recent developments in the discrete log setting particularly garnered renewed attention among practitioners, since some of them even serve as a drop-in replacement for ordinary signatures already deployed in practice [NRS21].

The main technical challenge when constructing a new multi-signature scheme is to achieve a set of desirable properties, such as (1) security in the plain public-key (PPK) model, (2) concurrent security, (3) low online round complexity, and (4) key aggregation. The PPK model requires that each signer publishes its public key in the clear without any dedicated interactive key generation protocol, and that no adversaries be able to convince a verifier that an honest party P_1[1] participated in signing any messages, unless P_1 has ever agreed on it. This is essentially to prevent the well-known rogue-key attacks (e.g., [MOR01]) in a plain way (i.e., without requiring *proof of possession* wherein each party must submit a proof to prove knowledge of their secret key [RY07]). Thus proving security under the PPK model is often considered ideal in the literature.

Several round-efficient Schnorr-based proposals with proof in the PPK model appeared in the literature. However, the seminal work of Drijvers et al. [DEF+19] pointed out subtle pitfalls of many existing interactive schemes, by presenting an adversarial strategy that exploits many *concurrent sessions*. The adversary in this scenario may launch multiple instances of the signing protocol with an honest party, and forge a signature on a new message by carefully combining signature shares from different sessions. Benhamouda et al. [BLL+21] recently improved the attack and proved that those schemes can be broken even in polynomial time. Given such devastating attacks, it is crucial to prove security of the scheme in the model where concurrent sign queries are allowed.

Although some previous schemes, such as BN [BN06], MuSig [MPSW19], MuSig-DN [NRSW20], mBCJ [DEF+19], and HBMS [BD21], are indeed provably secure against concurrent attacks, they all require (at least) two rounds of interaction during the *online phase*, i.e., after parties receive the message to sign. On the other hand, it is desirable in practice to *preprocess* part of the interaction and computation without knowledge of the message to be signed, so that participants can minimize round/communication complexity later. Such an offline-online trick has become increasingly common in context of general-purpose multi-party computation (e.g., [DPSZ12]), and therefore it is also another important design goal when constructing a multi-signature. Recently, Nick, Ruffing, and Seurin [NRS21], and Alper and Burdges [AB21] concurrently proposed near-optimal Schnorr-based multi-signatures in this paradigm. One remarkable feature of these schemes – MuSig2 and DWMS– is that they only require a *single round* of interaction in the online phase while retaining security against concurrent attacks. They also support *key aggregation*, an additional optimization technique that takes a set of public keys to produce a single combined Schnorr public key. It is crucial for a multi-signature scheme to support key aggregation, because it allows verifiers to verify a signature with an ordinary Schnorr public key and thus makes the scheme interoperable with the existing verification algorithms.

State-of-the-art in the lattice setting. As Schnorr-based constructions do not withstand quantum attacks, it is an interesting question how to construct post-quantum alternatives. Indeed, several lattice-based counterparts to the aforementioned schemes exist in the literature [ES16,MJ19,FH20,BK20, DOTT21]. All of these schemes follow the so-called *Fiat-Shamir with aborts (FSwA)* paradigm [Lyu12], which shares the basic structure with Schnorr. Hence, it is well-known that many observations in the DLog setting can be reused to construct similar FSwA-based instantiations, e.g., ES, MJ, BK, and FH follow the ideas of BN three-round Schnorr multi-signature, and the most recent scheme due to Damgård et al. [DOTT21,DOTT22] closely follows the mBCJ two-round scheme. There are however several subtle issues that only arise in the lattice world. For example, one inherent issue with the Fiat-Shamir "with aborts" multi-signature is simulation of the honest sign oracle. The basic idea of these schemes is to take the sum of usual FSwA signatures produced by different parties as follows: party P_1 first starts a protocol by sending "commit" messages $\mathbf{w}_1$ of the underlying Σ-protocol, and then upon receiving $\mathbf{w}_2, \ldots, \mathbf{w}_n$ from others, P_1 locally derives challenge c by hashing $\mathbf{w} := \sum_{i=1}^{n} \mathbf{w}_i$, together with the message μ to be signed. It then performs rejection sampling on the response $\mathbf{z}_1$, and the protocol must restart as long as there exists a party who rejected their response. This means that $\mathbf{w}_1$ is always revealed, whether P_1 aborts or not. However, there is currently no known general way to simulate $(\mathbf{w}_1, c)$ for rejected

[1] Note in multi-signature every honest party behaves identically and thinks of themselves as "P_1" [BN06]. Other parties $P_2, \ldots, P_n$ are called *co-signers*.

instances,[2] and thus publicly available proofs of ES and MJ are incomplete, and FH had to rely on a non-standard assumption (which they call "rejected" LWE). Although DOTT managed to circumvent the issue by having P_1 send a [BDL+18]-based *trapdoor homomorphic commitment* $\mathsf{Commit}(\mathbf{w}_1)$ to keep $\mathbf{w}_1$ secret until rejection sampling is successful and to realize the first two-round protocol, their approach inevitably makes the scheme incompatible with preprocessing: because each $\mathbf{w}_1$ must be committed using *message-dependent commitment keys*, two rounds of interaction must always happen online. Moreover, since their scheme has to output combined commitments or randomness as part of the signature, the verifier also needs to check an aggregated commitment is opened correctly in addition to the usual FSwA verification operations. These are in fact limitations inherited from mBCJ, and thus it is an interesting open question whether lattice-based multi-signature can be securely improved while benefiting from the latest tricks in the DL setting.

1.1 Our contributions

In this paper, we introduce MuSig-L, a lattice-based multi-signature scheme simultaneously achieving the aforementioned design goals for the first time: concurrent security in the PPK model, single-round online phase, and key aggregation. In Table 1 we compare ours to previous schemes following the same paradigm. Just as MuSig2 and DWMS, our MuSig-L allows parties to preprocess the first-round "commit" messages before receiving the message to be signed. Thus all they have to communicate during the online phase is the final response value $\mathbf{z}_i$. Although the protocol must abort if there is one party that fails in rejection sampling (which is also the case with other FSwA distributed/multi-signatures), we can mitigate by executing sufficiently many parallel instances of the protocol at once. Since security against concurrent attackers is crucial in this setting, we provide detailed security proofs in a suitable model.

Our scheme does not require any additional primitive for instantiating the protocol, unlike the two-round, provably secure scheme of Damgård et al. This was made possible by our generalized rejection sampling lemma in combination with trapdoor preimage sampling of [MP12] and several technical lemmas, as we sketch below. The resulting output signature of our scheme therefore looks much closer to the usual Fiat–Shamir-with-abort signatures.

Although our MuSig-L partially follows tricks present in MuSig2 and DWMS, the resulting scheme and our new proof techniques (outlined below) are significantly different from theirs. As a consequence, we are able to prove security solely based on the standard SIS and LWE assumptions in the ring setting and in the (classical) random oracle model, while MuSig2 and DWMS are proven secure either under the "one-more" DL assumption or in the algebraic group model.

1.2 Our techniques

Scheme overview Fig. 1 describes overview of our scheme, executed by P_1. In Section 3.1 we will provide more formal algorithm specifications. In MuSig-L, a key pair is the same as in the usual FSwA: $\mathsf{sk}_i = \mathbf{s}_i$ and $\mathsf{pk}_i = \mathbf{t}_i = \bar{\mathbf{A}}\mathbf{s}_i \bmod q$, where $\mathbf{s}_i$ consists of small elements in a usual power-of-two cyclotomic ring $R = \mathbb{Z}[X]/(X^N+1)$. On receiving public keys from the other parties, P_1 derives "aggregation coefficients" by hashing a set of keys and each public key held by P_i. Here the hash function is instantiated by the random oracle $\mathsf{H}_{\mathrm{agg}} : \{0,1\}^* \to C$, where C is the same as the challenge space used by the underlying FSwA Σ-protocol. It then constructs an aggregated key $\tilde{\mathbf{t}}$ by taking the linear combination of all keys. This is similar to the key aggregation technique introduced in MuSig [MPSW19] (where they choose a_i to be *uniform in* $\mathbb{Z}_q$), but we must carefully choose the size of aggregation coefficients so that it enables security reduction to the Module-SIS assumption (a strategy similar to [MJ19]).

In the offline phase, parties exchange a bunch of "commit" messages $\mathbf{w}_i^{(1)}, \ldots, \mathbf{w}_i^{(m)}$. We then use the "random linear combination" trick similar to MuSig2 and DWMS, to aggregate the "commit" messages coming from the offline phase. That is, we force everyone to derive the "nonce" coefficients $b^{(j)}$'s through another random oracle $\mathsf{H}_{\mathrm{non}}$, and these nonces are used for computing a single aggregate commit $\tilde{\mathbf{w}}$. This operation essentially prevents malicious parties from adaptively influencing inputs to the next random oracle $\mathsf{H}_{\mathrm{sig}}$ deriving "joint challenge" $c \in C$ that all parties must agree on. Finally, P_1 locally performs rejection sampling on a potential response value $\mathbf{z}_1$, such that the distribution of revealed $\mathbf{z}_1$ is always independent of the secret $\mathbf{s}_1$.

[2] We remark that [BK20] has attempted to simulate rejected transcripts, although, to the best of our understanding, they only cover the case where the ring $R_q = \mathbb{Z}_q[X]/(X^N+1)$ happens to be a field which is not the case for most existing FSwA schemes.

Table 1: Comparison with previous DLog/FSwA-based multi-signatures with concurrent security in the plain-public key model. The column "#Off" indicates the number of rounds that can be preprocessed in the offline phase.[3] "#On" indicates the number of rounds that must occur online after receiving a signature to sign. The total number of rounds is thus given as "#Off + #On". The column "Agg." indicates whether a scheme supports key aggregation or not. Except for [FH20] proofs are provided in the classical ROM.

	Assumption	#Off	#On	Agg.	Note
BN [BN06]	DL	1	2	N	Commit and open
MuSig [MPSW19]	DL	1	2	Y	Commit and open
mBCJ [DEF+19]	DL	0	2	Y	Trapdoor commitment
MuSig-DN [NRSW20]	DL & DDH	0	2	Y	NIZK proof of PRF evaluation
MuSig2 [NRS21]	AOMDL	1	1	Y	Linear combination
DWMS [AB21]	AGM	1	1	Y	Linear combination
HBMS [BD21]	DL	0	2	Y	Trapdoor commitment
ES [ES16]	DCK	1	2	N	Commit and open; proof incomplete[4]
MJ [MJ19]	RSIS	1	3	Y	Commit and open; proof incomplete
BK [BK20]	RSIS	1	2	N	Commit and open; R_q is a field
FH [FH20]	MLWE & rMLWE	1	2	N	Commit and open; proof in QROM
DOTT [DOTT22]	MLWE & MSIS	0	2	N	Trapdoor commitment
Our MuSig-L	MLWE & MSIS	1	1	Y	Linear combination; L must be a set[5]

Generalized rejection sampling. Not relying on a commitment scheme has a major drawback: we need to deal with possible leakage, due to both sending the first messages in the clear, and with aggregating them using random coefficients.

As the $\mathbf{w}_i^{(j)}$ are sent in the clear, the adversary $\mathcal{A}$ knows *before receiving* $\mathbf{z}_i$ that the response will be sampled from the coset $\Lambda^{\perp}_{\tilde{\mathbf{u}}}(\bar{\mathbf{A}})$, where $\tilde{\mathbf{u}} := \sum_j b^{(j)}\mathbf{w}_1^{(j)} + c \cdot a_1 \cdot \mathbf{t}_1$. This information does not give $\mathcal{A}$ any advantage in case the signing protocol succeeds. However, in case of abort $\mathcal{A}$ has gained some information on $\mathbf{z}_1$, that is, it knows that some element of $\Lambda^{\perp}_{\tilde{\mathbf{u}}}(\bar{\mathbf{A}})$ has been rejected. This could potentially leak information about the secret key, a subtle issue avoided in [DOTT22] by opening the commitment to the first message only in case of a success.

The second issue is related to efficiency. Aggregating the "commit" messages using some random coefficients implies that the distribution of the response $\mathbf{z}_1$ depends on those coefficients. In particular, the distributions of $\mathbf{z}_1$ is a Gaussian with parameter Σ that changes with different choices of the $b^{(j)}$'s. This is not just a nuisance: Σ leaks information about the $b^{(j)}$'s. It is not immediate to see why this is concerning, as it only becomes an issue when simulating honest signers in the security proof. Essentially, this requires to generate $\mathbf{z}_1$ *after* generating $\mathbf{w}_1^{(1)}, \ldots, \mathbf{w}_1^{(m)}$ with a trapdoor and *before* sampling the $b^{(j)}$'s *using such a trapdoor*. Thus, the distribution of $\mathbf{z}_1$ has to be independent of the $b^{(j)}$'s.

Perhaps unsurprisingly, rejection sampling can take care of all the leakage. In particular, we show that the rejection sampling technique is secure even if: (1) $\mathcal{A}$ knows the lattice coset, (2) the secret and public Gaussian distributions have different centers, and covariance matrices (obviously, for this to make sense neither difference can be too large). In fact, we prove a more general result than what the security of MuSig-L needs, allowing not only spherical, but ellipsoidal discrete Gaussians (i.e., Gaussians whose covariance matrix Σ is not diagonal). The proof of this result required quite the effort: while we could follow the structure of the proof of the original rejection sampling theorem, the intermediate steps required to extend many existing results, either to the case of ellipsoidal Gaussians, or to sampling from lattice cosets, or both. Proofs were simplified by relying on the canonical representation of ring elements, even though the rest of the algorithms will use the coefficient representation. This is not an issue per se, as these embeddings are isometric in power-of-2 cyclotomics. The result is a rather powerful extension

[3] Although ES, MJ, BK, and FH do not explicitly support an offline-online paradigm, we conjecture the first round of these schemes can be securely preprocessed since they all follow the same blueprint of BN.

[4] We were informed by the authors of [ES16] that a complete security proof would eventually appear in a full version of their conference paper, although that full version is not available yet at the time of writing.

[5] This is because in our scheme each signer explicitly prohibits duplicate keys in the key list L so that the security proof goes through in the *offline-online* security model allowing concurrent sessions. The rationale behind this choice will be detailed in Section 4.5.

of the rejection sampling technique, that we believe of independent interest. As a direct consequence of our generalization, in Appendix B.4 we provide a complete hybrid argument allowing one to simulate rejected transcripts of the standard Fiat-Shamir with aborts protocols without commitment to $\mathbf{w}$.

Exploiting trapdoor sampling for simulation. As usual, the main technical challenge in proving security of multi-signature is to simulate the behaviors of an honest party P_1 without knowledge of the actual secret key. Although our rejection sampling lemma allows to simulate the distribution of $\mathbf{z}_1$ and thus the aggregated offline outputs $\tilde{\mathbf{w}}_1 = \bar{\mathbf{A}}\mathbf{z}_1 - c \cdot a_1 \cdot \mathbf{t}_1$, it is not immediately clear how one can make sure $\tilde{\mathbf{w}}_1$ is consistent with the offline messages $\mathbf{w}_1^{(j)}$ and nonces $b^{(j)}$. One naive approach would be to mimic the security proof for MuSig2: they essentially avoid the issue with simulation by relying on hardness of the *one-more* DL problem, a stronger assumption that solving DL is still hard even after making a limited number of queries to a DL solver oracle. Although a similar lattice-based problem was recently introduced by Agrawal et al. [AKSY21] and it might make an interesting alternative approach to proving our scheme, it is not a well-studied assumption yet and we're thus motivated to propose an entirely different proof strategy.

One crucial observation is that, in the lattice world, a simulator can secretly produce a *trapdoor* when creating the offline messages $\mathbf{W} := [\mathbf{w}_1^{(1)}, \ldots, \mathbf{w}_1^{(m)}]$, using the gadget-based trapdoor generation algorithm of Micciancio and Peikert [MP12] with $m = O(k \log q)$. Once the corresponding trapdoor is known, the simulator can now sample $\mathbf{b} = [b^{(1)}, \ldots, b^{(m)}]$ from a coset $\Lambda^{\perp}_{\tilde{\mathbf{w}}_1}(\mathbf{W})$ using a Gaussian preimage sampling for the SIS function $f_{\mathbf{W}} : \mathbf{x} \mapsto \mathbf{W} \cdot \mathbf{x} \bmod q$. In this way, our simulator can successfully output a simulated signature, offline messages, and nonces $b^{(j)}$ that are all statistically indistinguishable with actual outputs of the honest party. In Section 4.4 we realize this idea in the form of *oracle simulation lemma*, which is proven by combining the utility lemma in Section 4.2 and instantiation of the trapdoor in Section 4.3. Finally, Section 4.5 formally states CMA security of our scheme.

Supporting technical lemmas. Our analysis and the security proof of our protocol rely on a number of technical facts related to discrete Gaussian distributions over module lattices, sometimes with general covariance matrices. Most of those facts are simple extensions and generalizations of well-known results in the literature, while others are less easy to come up with. Since a number of them may be of independent interest, we have tried to state them in a relatively high level of generality, and to provide relatively self-contained proofs either way.

Caveats and future directions In this work we limited ourselves to constructing the first offline-online lattice-based multi-signature achieving several important design goals with a rigorous security proof in the classical ROM, rather than striving for concrete efficiency and proof in the QROM. That is, we showcase our new proof techniques combined with MuSig2-like tricks lead to minimal online round complexity and key aggregation, while maintaining the asymptotic signature size and the set of assumptions of [DOTT22]. Our proof invokes the double-forking technique of [NRS21], causing a quartic loss in the security reduction and making itself incompatible with the QROM. The former issue is a limitation inherited from all previous schemes within the same paradigm supporting key aggregation (except for DWMS only proven in the AGM, and mBCJ not in the PPK). In fact, without key aggregation it is rather straightforward to prove a variant of our scheme with a usual quadratic loss using the standard forking lemma, which is comparable to DOTT. We highlight lifting our proof in the QROM using, e.g., the lossy ID techniques of [AFLT16,KLS18] as an interesting direction for future work. Although concrete parameters and implementation are also left for future work, we expect a number of optimizations are applicable to our basic blueprint construction, e.g., by exploiting (Mod-)NTRU for trapdoor sampling (which will likely reduce the parameter m and thus σ_1 significantly) instead of [MP12], by relying on the "one-more" SIS assumption of [AKSY21], by applying the bit truncation tricks of Dilithium, etc. We believe our work serves as a stepping stone towards truly practical multi-signatures with quantum resiliency.

1.3 Concurrent work

Fleischhacker, Simkin and Zhang [FSZ22] also propose a new lattice-based multi-signature scheme. Both MuSig-L and their construction (denoted by FSZ below) are provably secure against rogue-key attacks in the classical ROM and under standard lattice assumptions, and support the offline-online paradigm allowing the expensive operations to be preprocessed. However, the two works contain significantly different motivations, techniques, and security models, and thus contributions are somewhat incomparable. We summarize the differences below.

– FSZ combines the one-time signature of [BK20] with their improved homomorphic Merkle tree commitment, while MuSig-L is based on the Fiat-Shamir with aborts paradigm of [Lyu12].

$P_1(\bar{\mathbf{A}} = [\mathbf{A}|\mathbb{I}_k], \mathsf{sk}_1 = \mathbf{s}_1, \mathsf{pk}_1 = \mathbf{t}_1 = \bar{\mathbf{A}} \cdot \mathbf{s}_1, \mu)$

// Key aggregation phase

$\mathbf{t}_1 \longrightarrow$

$\longleftarrow (\mathbf{t}_i)_{i\in[2,n]}$

//Derive aggregation coefficients

For $i \in [n] : a_i := \mathsf{H}_{\mathrm{agg}}((\mathbf{t}_i)_{i\in[n]}, \mathbf{t}_i)$

$\tilde{\mathbf{t}} := \sum_{i=1}^{n} a_i \mathbf{t}_i \bmod q$

// Offline phase

$\mathbf{y}_1^{(1)} \leftarrow \mathscr{D}_{\sigma_1}^{\ell+k}$

For $j \in [2, m] : \mathbf{y}_1^{(j)} \leftarrow \mathscr{D}_{\sigma_y}^{\ell+k}$

For $j \in [1, m] : \mathbf{w}_1^{(j)} := \bar{\mathbf{A}}\mathbf{y}_1^{(j)} \bmod q$

$\mathsf{com}_1 := (\mathbf{w}_1^{(j)})_{j\in[m]}$

$\mathsf{pk}_1||\mathsf{com}_1 \longrightarrow$

$\longleftarrow (\mathsf{pk}_i||\mathsf{com}_i)_{i\in[2,n]}$

// Online phase

If $\exists i \geq 2 : \mathsf{pk}_i = \mathsf{pk}_1$: Abort

$(r^{(j)})_{j\in[2,m]} := \mathsf{H}_{\mathrm{non}}((\mathsf{pk}_i||\mathsf{com}_i)_{i\in[n]}, \mu, \tilde{\mathbf{t}})$

$b^{(1)} := 1$

For $j \in [2, m]$: sample $b^{(j)} \sim \mathscr{D}_{\sigma_b}$ using randomness $r^{(j)}$

$\tilde{\mathbf{w}} := \sum_{j=1}^{m} b^{(j)} \cdot \left(\sum_{i=1}^{n} \mathbf{w}_i^{(j)}\right) \bmod q$

$\tilde{\mathbf{y}}_1 := \sum_{j=1}^{m} b^{(j)} \cdot \mathbf{y}_1^{(j)}$

$c := \mathsf{H}_{\mathrm{sig}}(\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}})$

$\mathbf{z}_1 := c \cdot a_1 \cdot \mathbf{s}_1 + \tilde{\mathbf{y}}_1$

If $\mathsf{RejSamp}(c \cdot a_1 \cdot \mathbf{s}_1, \mathbf{z}_1, (b^{(j)})_{j\in[m]}) = 0$:

  $\mathbf{z}_1 := \perp$

$\mathbf{z}_1 \longrightarrow$

If $\mathbf{z}_i = \perp$ for some i, abort

$\longleftarrow (\mathbf{z}_i)_{i\in[2,n]}$

Otherwise, compute $\tilde{\mathbf{z}} := \sum_{i=1}^{n} \mathbf{z}_i$

Output $(\tilde{\mathbf{w}}, \tilde{\mathbf{z}})$

Fig. 1: Stylized overview of our two-round lattice-based multi-signature

– The goal of FSZ is to present a concretely efficient scheme in a slightly restricted "synchronized" model where signers are only allowed to produce a single signature in each time step. They further provide a proof-of-concept implementation with benchmarks, whereas concrete efficiency estimates are outside the scope of our work due to the fact that we didn't try to optimize for parameters. In contrast, we focus on the feasibility of FSwA multi-signature with minimal online round complexity in the usual plain public-key model where signers can freely launch many concurrent signing sessions (as defined in [BN06] and [NRS21]).
– The FSZ scheme is fully non-interactive, while MuSig-L requires one round of interaction in the offline phase and potentially more in case one of the signers aborts, which however can be mitigated to some extent using parallel repetitions.
– The FSZ scheme only allows a signer to produce a bounded number of signatures for a given public key (corresponding to a Merkle root obtained by hashing many one-time public keys) whereas MuSig-L relies on a usual Dilithium-like key pair and has no limitation on the number of signatures. However, they carefully choose parameters to achieve a good balance between the maximum number of signatures and the key size, allowing instantiation sufficient for a practical scenario.

1.4 Other related work

Multi-signatures belong to a larger family of signatures that support aggregation, its closest relatives being aggregate signatures and threshold signatures.

There have been a number of results on threshold Schnorr-style signatures [GJKR07,GKMN21,KG20, NKDM03,SS01] whose techniques are somewhat analogous to multi-signature counterparts. In particular, FROST [KG20] utilizes the random linear combination trick similar to MuSig2 to realize a two-round protocol. Threshold signatures can be instantiated from lattices, but the existing t-out-of-n constructions require either to threshold secret share the signing key of GPV signature [BKP13], or FHE [BGG+18, ASY22]. The multi-signature of [DOTT21] also gives rise to the n-out-of-n threshold signature, and they in fact showed that essentially the same tricks work under both security models. We therefore highlight adapting our techniques in the threshold setting as an interesting direction for future work. The panorama of aggregate signature from lattices is similar. A three-round construction by Boneh and Kim [BK20] requires interactive aggregation, which again closely follows the BN Schnorr-based scheme. The recent aggregate signature by Boudgoust and Roux-Langlois [BRL21] requires no interaction between signers although the asymptotic signature size grows linearly in the number of signers.

2 Preliminaries

Notations For positive integers a and b such that $a < b$ we use the integer interval notation $[a, b]$ to denote $\{a, a+1, \ldots, b\}$. We also use $[b]$ as shorthand for $[1, b]$. We denote by $\mathbf{y}[j]$ the j-th component of vector $\mathbf{y}$, and by $\mathbb{I}_n$ the identity matrix of dimension n. If S is a set we write $s \leftarrow\!\!\$\ S$ to indicate sampling s from the uniform distribution defined over S; if $\mathscr{D}$ is a probability distribution we write $s \leftarrow \mathscr{D}$ to indicate sampling s from $\mathscr{D}$; if $\mathcal{A}$ is a randomized (resp. deterministic) algorithm we write $s \leftarrow \mathcal{A}$ (resp. $s := \mathcal{A}$) to indicate assigning an output from $\mathcal{A}$ to s. For a set S, $\langle S \rangle$ denotes a unique encoding of S (e.g., the sequence of strings in lexicographic order). Throughout, the security parameter is denoted by λ.

Power-of-two cyclotomics and norms We instantiate the scheme over power-of-two cyclotomics. Let N be a power of two and ζ be a primitive $2N$th root of unity. The $2N$th cyclotomic number field is denoted by $K := \mathbb{Q}(\zeta) \cong \mathbb{Q}[X]/(X^N + 1)$ and the corresponding ring of algebraic integers is $R := \mathbb{Z}[\zeta] \cong \mathbb{Z}[X]/(X^N + 1)$. Both are contained in $K_{\mathbb{R}} := K \otimes \mathbb{R} \cong \mathbb{R}[X]/(X^N + 1)$. Throughout the paper, we fix q to be a prime satisfying $q = 5 \bmod 8$ and let $R_q := R/qR \cong \mathbb{Z}_q[X]/(X^N + 1)$. An L^p-norm for a module element $\mathbf{v} \in R^m$ is given by the coefficient embedding: for $\mathbf{v} = (\sum_{i=0}^{N-1} v_{i,1} X^i, \ldots, \sum_{i=0}^{N-1} v_{i,m} X^i)^T$, we define

$$\|\mathbf{v}\|_p := \left\|(v_{0,1}, \ldots, v_{N-1,1}, \ldots, v_{0,m}, \ldots, v_{N-1,m})^T\right\|_p.$$

Elements in $R \subset K$ can be also represented through the canonical embedding $\varphi : K \to \mathbb{C}^N$ that associates to $a \in K$ its evaluation over odd powers of ζ. That is, $\varphi := (\varphi_j)_{j \in \mathbb{Z}_{2N}^*}$ where $\varphi_j : a \mapsto a(\zeta^j)$. The Euclidean norm of a vector $\mathbf{v} = (v_1, \ldots, v_m)^T \in R^m$ in the canonical representation is defined as

$$\|\varphi(\mathbf{v})\|^2 := \sum_{i \in [m], j \in \mathbb{Z}_{2N}^*} |\varphi_j(v_i)|^2/N,$$

where the scaling factor is needed to ensure that $\|\varphi(1)\| = 1$. For power-of-2 cyclotomics, this choice of norm yields that the coefficient embedding and the canonical embedding are isometric, thus we denote the L^2-norm by $\|\cdot\|$ for both representations.

We will need the following results on invertibility.

Lemma 2.1 ([LS18, Corollary 1.2]). *Let $N \geq k > 1$ be powers of 2 and $q = 2k + 1 \mod 4k$ be a prime. Then any y in R_q that satisfies either $0 < \|y\|_\infty < \frac{1}{\sqrt{k}} \cdot q^{1/k}$ or $0 < \|y\| < q^1/k$ has an inverse in R_q.*

Lemma 2.2 ([LN17, Lemma 2.2]). *Let $N > 1$ be a power of 2 and q a prime congruent to $5 \mod 8$. The ring R_q has exactly $2q^{N/2} - 1$ elements without an inverse. Moreover, every non-zero polynomial a in R_q with $\|a\|_\infty < \sqrt{q}/2$ has an inverse.*

Singular Values. Given a matrix $B \in K_{\mathbb{R}}^{n\times m}$, let $\mathrm{s}_1(B)$ (resp., $\mathrm{s}_m(B)$) be the *largest* (resp., *least*) *singular value* of B, i.e., $\mathrm{s}_1(B) = \sup\{\|B\mathbf{v}\| \ : \ \mathbf{v} \in K_{\mathbb{R}}^m \ \wedge \ \|\mathbf{v}\| = 1\}$ (resp., $\mathrm{s}_m(B) = \inf\{\|B\mathbf{v}\| \ : \ \mathbf{v} \in K_{\mathbb{R}}^m \ \wedge \ \|\mathbf{v}\| = 1\}$). For all $\mathbf{v}$, $\mathrm{s}_m(B)\|\mathbf{v}\| \leq \|B\mathbf{v}\| \leq \mathrm{s}_1(B)\|\mathbf{v}\|$. If B is a diagonal matrix, i.e., $B = \sigma_i \mathbb{I}_m$ for some $\sigma_i \in K_{\mathbb{R}}$, we have that $\mathrm{s}_1(B) = \max_i \|\sigma_i\|$ and $\mathrm{s}_m(B) \leq \min_i \|\sigma_i\|$ (the proof trivially follows from standard bounds, cf. [Mic02]).

Lemma 2.3. *Given a symmetric positive definite matrix $B \in K_{\mathbb{R}}^{m\times m}$, and a nonsingular matrix $\sqrt{B} \in K_{\mathbb{R}}^{m\times m}$ such that $B = \sqrt{B}\sqrt{B}^*$, it holds that $s_i(B) = (s_i(\sqrt{B}))^2$ for $i = 1, m$, and $s_1(B^{-1}) = (s_m(B))^{-1}$.*

Proof. Let $\sqrt{\mathbf{B}} = \mathbf{QSU}$ be the singular value decomposition of $\sqrt{\mathbf{B}}$. One can obtain the singular value decomposition of $\mathbf{B}$ from the decomposition of $\sqrt{\mathbf{B}}$:

$$\sqrt{\mathbf{B}} = \mathbf{QSU} \quad \Rightarrow \quad \mathbf{B} = \sqrt{\mathbf{B}}\sqrt{\mathbf{B}}^* = \mathbf{QSU}(\mathbf{QSU})^* = \mathbf{QS}^2\mathbf{Q} \ ,$$

thus for all $i = 1, \ldots, m$ it holds $\mathrm{s}_i(\mathbf{B}) = (\mathrm{s}_i(\sqrt{\mathbf{B}}))^2$. Analogously, from $\mathbf{B}^{-1} = \mathbf{Q}(\mathbf{S}^2)^{-1}\mathbf{Q}$ it follows that $\mathrm{s}_i(\mathbf{B}^{-1}) = \mathrm{s}_i((\mathbf{S}^2)^{-1}) = (\mathrm{s}_{m-(i-1)}(\mathbf{S}^2))^{-1} = (\mathrm{s}_{m-(i-1)}(\mathbf{B}))^{-1}$.

2.1 Discrete Gaussian Distribution

Let $\Sigma \in K_{\mathbb{R}}^{m\times m}$ be a symmetric positive definite matrix, and let $\sqrt{\Sigma} \in K_{\mathbb{R}}^{m\times m}$ be a nonsingular matrix such that $\Sigma = \sqrt{\Sigma}\sqrt{\Sigma}^*$. The discrete Gaussian distribution $\mathscr{D}_{\Sigma,\mathbf{c},\Lambda}$ over a lattice $\Lambda \subseteq R^m$ with parameters $\mathbf{c}$ and Σ is defined as

$$\rho_{\sqrt{\Sigma},\mathbf{c}}(\mathbf{z}) := \exp\left(-\pi\|\sqrt{\Sigma}^{-1}(\mathbf{z} - \mathbf{c})\|^2\right) \quad \text{and} \quad \mathscr{D}^m_{\sqrt{\Sigma},\mathbf{c},\Lambda}(\mathbf{z}) := \frac{\rho_{\sqrt{\Sigma},\mathbf{c}}(\mathbf{z})}{\sum_{\mathbf{x}\in\Lambda}\rho_{\sqrt{\Sigma}}(\mathbf{x})} \ .$$

If $\Sigma = \sigma^2 \cdot \mathbb{I}_m$ we write $\mathscr{D}^m_{\sigma,\mathbf{c},\Lambda}$. We denote by $\mathscr{D}^m_{\sqrt{\Sigma},\mathbf{c}}$ the discrete Gaussian over R^m, and omit $\mathbf{c}$ when $\mathbf{c} = 0$. In our signature scheme, signers only carry out spherical Gaussian sampling with $\sigma \in \mathbb{R}^+$ in the coefficient representation. Since the canonical and coefficient embeddings are isometric when N is a power of 2, a potential response value should be first converted to the canonical representation and then cast to our generalized rejection sampling technique whose analysis is done w.r.t. the canonical representation for technical reasons.

The *smoothing parameter* $\eta_\varepsilon(\Lambda)$ of a lattice for $\varepsilon > 0$ is the smallest $s > 0$ such that $\rho_{1/s\mathbb{I}_m}(\Lambda^* \setminus \{0\}) \leq \varepsilon$. For a positive definite matrix $\sqrt{\Sigma}$, we say that $\Sigma \geq \eta_\varepsilon(\Lambda)$ (i.e., $\mathrm{s}_m(\Sigma) \geq \eta_\varepsilon(\Lambda)$) if $\eta_\varepsilon(\sqrt{\Sigma}^{-1}\Lambda) \leq 1$, i.e., if $\rho_{\sqrt{\Sigma}^{-1}}(\Lambda) \leq \varepsilon$. Throughout the paper we assume $\varepsilon = 2^{-N}$.

We need a bound on the smoothing parameter of the Module-SIS lattice $\Lambda_q^\perp(\bar{\mathbf{A}})$ for a random matrix $\bar{\mathbf{A}}$ modulo q, generalizing [GPV08, Lemma 4.3]. Such a bound is obtained by Langlois and Stehlé [LS15, Lemma 5.1] as follows. First recall the following standard consequence of Banaszcsyk's transference.

Lemma 2.4 ([GPV08, Lemma 2.5]). *For any full-rank lattice Λ in $\mathbb{R}^n$ and any $\varepsilon > 0$, we have:*

$$\eta_\varepsilon(\Lambda) \leq \frac{\sqrt{\log\left(2n(1 + 1/\varepsilon)\right)/\pi}}{\lambda_1^\infty(\Lambda^*)},$$

where $\lambda_1^\infty(\Lambda^*)$ denotes the length of the shortest vector in the infinity norm in the dual lattice Λ^*.

Now for any $\bar{\mathbf{A}} \in R_q^{n\times m}$, recall that

$$\Lambda_q^\perp(\bar{\mathbf{A}}) = \{\mathbf{x} \in R^m \ : \ \bar{\mathbf{A}}\mathbf{x} \equiv 0 \bmod q\}$$

has as its dual lattice $\frac{1}{q}\Lambda_q(\bar{\mathbf{A}})$ where

$$\Lambda_q(\bar{\mathbf{A}}) = \{\mathbf{x} \in R^m \ : \ \mathbf{x} \equiv \bar{\mathbf{A}}^*\mathbf{y} \bmod q \text{ for some } \mathbf{y} \in R^n\},$$

and $\bar{\mathbf{A}}^*$ denotes the conjugate transpose of $\bar{\mathbf{A}}$. We then have:

Lemma 2.5. *Let q be an odd integer and $\bar{\mathbf{A}}$ a uniformly random matrix in $R_q^{n\times m}$, $m > n$. Then, except with probability at most 2^{-N} on the choice of $\bar{\mathbf{A}}$, we have:*

$$\lambda_1^\infty\big(\Lambda_q(\bar{\mathbf{A}})\big) \geq \frac{1}{8\sqrt{N}} q^{1-\frac{n}{m}}. \tag{1}$$

In particular, for any $\varepsilon > 0$, with overwhelming probability on the choice of $\bar{\mathbf{A}}$, we have:

$$\eta_\varepsilon(\Lambda_q^\perp(\bar{\mathbf{A}})) \leq \frac{8}{\sqrt{\pi}} q^{\frac{n}{m}} \sqrt{N \log\big(2mN(1+1/\varepsilon)\big)}.$$

Proof. The first claim is an immediate consequence of [LS15, Lemma 5.1]. We only need to take into account the fact that, since we use the coefficient embedding as opposed to the canonical embedding in [LS15], our definition of $\Lambda_q(\bar{\mathbf{A}})$ is scaled by a factor of N compared to that of Langlois and Stehlé (which is a sublattice of $\frac{1}{N}R^m$). On the other hand, the infinity norm on R^m in our case (defined as the maximum absolute value of all coefficients of all components of a vector) can be up to N times smaller than the $\|\cdot\|_{\infty,2}$ considered in their setting. Those two factors of N compensate exactly to yield (1).

Then, (1) combined with Lemma 2.4 yields the second claim, taking into account the fact that $\Lambda_q^\perp(\bar{\mathbf{A}})$ is of full $\mathbb{Z}$-rank mN with overwhelming probability. □

The next lemma extends the classical bound on the norm of a sample from a discrete ellipsoid Gaussian over the cosets. Its proof is analogous to the original; it essentially follows observing that $\mathscr{D}_{\Lambda+\mathbf{u},\sqrt{\Sigma}}(\mathbf{z}) = \rho_{\sqrt{\Sigma}}(\mathbf{z})/\rho_{\sqrt{\Sigma}}(\Lambda+\mathbf{u}) \propto \rho_{\sqrt{\Sigma}}(\mathbf{z})$.

Lemma 2.6 ([AGHS13, Lemma 3] adapted to rings and sampling from cosets). *For any $0 < \varepsilon < 1$, lattice $\Lambda \subseteq R^m$, $\mathbf{u} \in R^m$, and symmetric positive definite matrix $\Sigma \in K_{\mathbb{R}}^{m\times m}$ such that $s_m(\Sigma) \geq \eta_\varepsilon(\Lambda)$,*

$$\Pr\left[\|\mathbf{z}\| \geq s_1(\sqrt{\Sigma})\sqrt{mN} \ : \ \mathbf{z} \leftarrow \mathscr{D}^m_{\sqrt{\Sigma},\Lambda+\mathbf{u}}\right] < \frac{1+\varepsilon}{1-\varepsilon} 2^{-mN} \ .$$

Proof. By [AGHS13, Fact2] one can sample from the distribution $\mathscr{D}^m_{\Lambda+\mathbf{u},\sqrt{\Sigma}}$ by sampling $\mathbf{t}$ from a spherical Gaussian with covariance 1 over the lattice coset $\Lambda' + \mathbf{u}' := \sqrt{\Sigma}^{-1}\Lambda + \sqrt{\Sigma}^{-1}\mathbf{u}$ and returning $\mathbf{z} = \sqrt{\Sigma}\mathbf{t}$. As $\mathrm{s}_1(\sqrt{\Sigma}^{-1}) = 1/\mathrm{s}_m(\sqrt{\Sigma})$, the lattice Λ' is obtained shrinking the vectors of Λ; thus $\mathrm{s}_m(\sqrt{\Sigma}) \geq \eta_\varepsilon(\Lambda) \geq \eta_\varepsilon(\Lambda')$. Applying [Ban93, Lemma 1.5] and [GPV08, Lemma 2.7] yields

$$\begin{aligned}
&\Pr\left[\|\mathbf{t}\| \geq s\sqrt{mN} \ : \ \mathbf{t} \leftarrow \mathscr{D}^m_{\Lambda'+\mathbf{u}'}\right] \\
&\quad = \frac{\rho((\Lambda'+\mathbf{u}') \setminus \sqrt{mN}\cdot\mathcal{B})}{\rho(\Lambda'+\mathbf{u}')} \leq \frac{\rho(\Lambda')}{\rho_{-\mathbf{u}'}(\Lambda')} 2^{-mN} \leq \frac{1+\varepsilon}{1-\varepsilon} 2^{-mN} \ ,
\end{aligned}$$

where $\mathcal{B}$ is the fundamental parallelepiped of Λ. The thesis follows observing that $\|\mathbf{z}\| = \|\sqrt{\Sigma}\mathbf{t}\| \leq \mathrm{s}_1(\sqrt{\Sigma})\|\mathbf{t}\|$. □

The following result is a direct generalization of [MP13, Theorem 3.3] to the ring setting. The proof is identical, but we include it for the sake of completeness.

Lemma 2.7. *Let $\Lambda \subset R^n$ be a full-rank module lattice, $z_1, \ldots, z_m \in R$ arbitrary elements, and $\sigma_1, \ldots, \sigma_m \in K_{\mathbb{R}}^{++}$ satisfying $\sigma_i \succ \sqrt{2}\eta_\varepsilon(\Lambda) \cdot \max_j \|\sqrt{z_j z_j^*}\|$ for all i. Pick $\mathbf{y}_1, \ldots, \mathbf{y}_m \in K_{\mathbb{R}}^n$ independently with distributions $\mathbf{y}_i \sim \mathscr{D}_{\Lambda+\mathbf{c}_i,\sigma_i}$ for some centers $\mathbf{c}_i \in K_{\mathbb{R}}^n$, and let $\mathbf{y} = \sum_i z_i \cdot \mathbf{y}_i$. Then, the distribution of $\mathbf{y}$ is statistically close to $\mathscr{D}_{\mathscr{I}\cdot\Lambda+\mathbf{c},\sigma}$ where $\mathscr{I}$ is the ideal generated by the z_i's, $\mathbf{c} = \sum_i z_i \cdot \mathbf{c}_i$ and*

$$\sigma = \sqrt{\sum_i z_i z_i^* \cdot \sigma_i^2}.$$

In particular, if the z_i's are coprime (i.e., $\mathscr{I} = R$), the distribution of $\mathbf{y}$ statistically close to $\mathscr{D}_{\Lambda+\mathbf{c},\sigma}$.

Proof. Clearly, the support of $z_i \cdot \mathbf{y}_i$ is exactly $z_i(\Lambda+\mathbf{c}_i)$, so the support of $\mathbf{y}$ is $z_1(\Lambda+\mathbf{c}_1)+\cdots+z_m(\Lambda+\mathbf{c}_m) = \mathscr{I}\cdot\Lambda+\mathbf{c}$ as required. Thus, all we have to show is that for $\mathbf{y}_0 \in \mathscr{I}\cdot\Lambda+\mathbf{c}$, $\Pr[\mathbf{y}=\mathbf{y}_0]$ is proportional to $\rho_\sigma(\mathbf{y}_0)$, up to negligible variation.

We do so by following the approach of Micciancio–Peikert [MP13, Th. 3.3], and writing $\mathbf{S} = \mathrm{diag}(\sigma_1, \ldots, \sigma_m)$, $\mathbf{S}' = \mathbf{S}\otimes\mathbb{I}_n$ and $\Lambda' = (\mathbf{S}')^{-1}\cdot\Lambda^m = \bigoplus_i \sigma_i^{-1}\Lambda \subset K_{\mathbb{R}}^{mn}$. By indepedence of the $\mathbf{y}_i$'s, the vector $\mathbf{y}' = (\mathbf{S}')^{-1}\cdot(\mathbf{y}_1,\ldots,\mathbf{y}_m)$ is distributed as $\mathscr{D}_{\Lambda'+\mathbf{c}'}$ (with parameter 1) where $\mathbf{c}' = (\mathbf{S}')^{-1}\cdot(\mathbf{c}_1,\ldots,\mathbf{c}_m)$. And we have $\mathbf{y} = \mathbf{Z}\mathbf{y}'$ where $\mathbf{Z} = ((z_1,\ldots,z_m)^T\cdot\mathbf{S})\otimes\mathbb{I}_n$. Hence, we want to prove that $\mathbf{Z}\cdot\mathscr{D}_{\Lambda'+\mathbf{c}'}$ is statistically close to $\mathscr{D}_{\mathscr{I}\cdot\Lambda+\mathbf{c},\sigma}$.

Fix $\mathbf{x}_0 \in \Lambda'+\mathbf{c}'$ and let $\mathbf{y}_0 = \mathbf{Z}\mathbf{x}_0 \in \mathscr{I}\cdot\Lambda+\mathbf{c}$. Moreover, define the sublattice $L \subsetneq \Lambda'$ given by $L = \{\mathbf{v}\in\Lambda' \ : \ \mathbf{Z}\mathbf{v}=\mathbf{0}\} = \Lambda'\cap\ker(\mathbf{Z})$. A vector $\mathbf{y}'\in\Lambda'+\mathbf{c}'$ satisfies $\mathbf{Z}\mathbf{y}' = \mathbf{y}_0$ if and only if $\mathbf{Z}(\mathbf{y}'-\mathbf{x}_0)=\mathbf{0}$, that is, $\mathbf{y}'-\mathbf{x}_0\in L$ (since the difference is in Λ'). In other words, the set of $\mathbf{y}'\in\Lambda'+\mathbf{c}'$ such that $\mathbf{Z}\mathbf{y}'=\mathbf{y}_0$ is exactly $L+\mathbf{x}_0$. Therefore:

$$\Pr[\mathbf{y}=\mathbf{y}_0] = \frac{\rho(L+\mathbf{x}_0)}{\rho(\Lambda'+\mathbf{c}')} = \rho(\mathbf{x}_0-\mathbf{x}_0^\perp)\cdot\frac{\rho(L+\mathbf{x}_0^\perp)}{\rho(\Lambda'+\mathbf{c}')},$$

where $\mathbf{x}_0^\perp$ denotes the orthogonal projection of $\mathbf{x}_0$ on $\ker(\mathbf{Z}) = \mathrm{span}(L)$. The remainder of the proof consists of two steps: showing that $\rho(\mathbf{x}_0-\mathbf{x}_0^\perp) = \rho_\sigma(\mathbf{y}_0)$ on the one hand, and showing that $\rho(L+\mathbf{x}_0^\perp)$ is negligibly close to $\rho(L)$ and hence essentially independent of $\mathbf{x}_0^\perp$ on the other hand. This will yield $\Pr[\mathbf{y}=\mathbf{y}_0]$ essentially proportional to $\rho_\sigma(\mathbf{y}_0)$ as required.

First, note that $\mathbf{x}_0-\mathbf{x}_0^\perp = \mathbf{Z}^*\mathbf{Z}\mathbf{x}_0/\sigma^2$ (where for matrices, the star denotes the conjugate transpose). Indeed, denoting the right-hand side by $\mathbf{w}$, we have:

$$\mathbf{Z}\cdot(\mathbf{x}_0-\mathbf{w}) = \mathbf{Z}\cdot\mathbf{x}_0 - \mathbf{Z}\mathbf{Z}^*\cdot\mathbf{Z}\cdot\mathbf{x}_0/\sigma^2 = \Big(1-\frac{\langle\mathbf{S}\bar{\mathbf{z}},\mathbf{S}\bar{\mathbf{z}}\rangle}{\sigma^2}\Big)\mathbf{Z}\cdot\mathbf{x}_0 = 0,$$

where we have denoted $\bar{\mathbf{z}} = (z_1^*,\ldots,z_m^*)$, and the second equality comes from the fact that $\mathbf{Z}\mathbf{Z}^* = (\bar{\mathbf{z}}^*\mathbf{S}\otimes\mathbb{I}_n)\cdot(\mathbf{S}\bar{\mathbf{z}}\otimes\mathbb{I}_n) = \langle\mathbf{S}\bar{\mathbf{z}},\mathbf{S}\bar{\mathbf{z}}\rangle\mathbb{I}_n = \sigma^2\mathbb{I}_n$. Thus, $\mathbf{x}_0-\mathbf{w}\in\ker(\mathbf{Z})$. Moreover, for any $\mathbf{u}\in\ker(\mathbf{Z})$, we have:

$$\langle\mathbf{w},\mathbf{u}\rangle = \langle\frac{1}{\sigma^2}\mathbf{Z}^*\mathbf{Z}\mathbf{x}_0,\mathbf{u}\rangle = \langle\frac{1}{\sigma^2}\mathbf{Z}\mathbf{x}_0,\mathbf{Z}\mathbf{u}\rangle = 0,$$

therefore $\mathbf{w}\in\ker(\mathbf{Z})^\perp$. This proves that the orthogonal projection $\mathbf{x}_0^\perp$ of $\mathbf{x}_0$ on $\ker(\mathbf{Z})$ is indeed equal to $\mathbf{x}_0-\mathbf{w}$ as claimed.

As a result, we have:

$$\begin{aligned}\|\mathbf{x}_0-\mathbf{x}_0^\perp\|^2 = \|\mathbf{w}\|^2 &= \langle\frac{1}{\sigma^2}\mathbf{Z}^*\mathbf{Z}\mathbf{x}_0,\frac{1}{\sigma^2}\mathbf{Z}^*\mathbf{Z}\mathbf{x}_0\rangle\\ &= \langle\frac{\mathbf{Z}\mathbf{Z}^*}{\sigma^4}\cdot\mathbf{Z}\mathbf{x}_0,\mathbf{Z}\mathbf{x}_0\rangle = \langle\frac{1}{\sigma}\mathbf{Z}\mathbf{x}_0,\frac{1}{\sigma}\mathbf{Z}\mathbf{x}_0\rangle = \Big\|\frac{\mathbf{y}_0}{\sigma}\Big\|^2.\end{aligned}$$

In particular, $\rho(\mathbf{x}_0-\mathbf{x}_0^\perp) = \rho_\sigma(\mathbf{y}_0)$ as required.

Finally, all that remains to prove is that $\rho(L+\mathbf{x}_0^\perp)$ is within negligible deviation of $\rho(L)$ (and in fact, $\rho(L+\mathbf{x})$ is within negligible deviation of $\rho(L)$ for all $\mathbf{x}\in\mathrm{span}(L)$). This results from the fact that $\eta_\varepsilon(L)\le 1$ under the assumptions of the theorem, as we will now show.

Indeed, introduce the lattice $Z\subset R^m$ of vectors $\mathbf{v}\in R^m$ orthogonal to $\bar{\mathbf{z}}$. An R-basis of Z is given by the vectors $(z_i^*,0,\ldots,0,-z_1^*,0,\ldots,0)$ for $i=2,\ldots,m$, which are all of norm $\le\sqrt{2}\cdot\max_i\|\sqrt{z_i^*z_i}\|$. Now note that $(\mathbf{S}')^{-1}(Z\otimes\Lambda) = \big((\mathbf{S})^{-1}Z\big)\otimes\Lambda$ is a sublattice of L: this is because it is contained in $\Lambda' = (\mathbf{S}')^{-1}(R^m\otimes\Lambda)$, and for all $\mathbf{z}_0\in Z$, $\mathbf{v}_0\in\Lambda$, we have:

$$\mathbf{Z}\cdot(\mathbf{S}')^{-1}(\mathbf{z}_0\otimes\mathbf{v}_0) = (\mathbf{z}^T\mathbf{S}\mathbf{S}^{-1}\mathbf{z}_0)\otimes\mathbf{v}_0 = \langle\bar{\mathbf{z}},\mathbf{z}_0\rangle\mathbf{v}_0 = \mathbf{0}.$$

In particular:

$$\eta_\varepsilon(L)\le\eta_\varepsilon((\mathbf{S}')^{-1}(Z\otimes\Lambda))\le\frac{\eta_\varepsilon(Z\otimes\Lambda)}{\min_i\|\sigma_i\|}\le\frac{\eta_\varepsilon(\Lambda)\cdot\sqrt{2}\cdot\max_i\|\sqrt{z_i^*z_i}\|}{\min_i\|\sigma_i\|}\le 1$$

where the third inequality on the smoothing of a tensor product is given by [MP13, Corollary 2.7], and the last one by assumption on the σ_i's. This concludes the proof. □

2.2 Assumptions

We restate the two lattice problems over a module that are standard in the literature: module short integer solution (MSIS) and learning with errors (MLWE). Note that the latter k elements of $\mathbf{s}$ correspond to the error term of MLWE. The set S_η is defined in Table 2.

Definition 2.8 ($\mathsf{MSIS}_{q,k,\ell,\beta}$ assumption). *Let $\lambda \in \mathbb{N}$ be a security parameter. For a prime $q(\lambda)$, a bound $\beta = \beta(\lambda) > 0$ and positive integers $k = k(\lambda)$, $\ell = \ell(\lambda)$, the $\mathsf{MSIS}_{q,k,\ell,\beta}$ assumption holds if for any probabilistic polynomial-time algorithm $\mathcal{A}$, the following advantage is negligible in λ.*

$$\mathsf{Adv}^{\mathsf{MSIS}}_{q,k,\ell,\beta}(\mathcal{A}) := \Pr\left[0 < \|\mathbf{x}\| \leq \beta \wedge [\mathbf{A}|\mathbb{I}_k] \cdot \mathbf{x} = \mathbf{0} \bmod q \ : \ \mathbf{A} \leftarrow\$ R_q^{k\times\ell}; \mathbf{x} \leftarrow \mathcal{A}(\mathbf{A})\right].$$

Definition 2.9 ($\mathsf{MLWE}_{q,k,\ell,\eta}$ assumption). *Let $\lambda \in \mathbb{N}$ be a security parameter. For a prime $q(\lambda)$, and positive integers $k = k(\lambda)$, $\ell = \ell(\lambda)$, $\eta = \eta(\lambda)$, the $\mathsf{MLWE}_{q,k,\ell,\eta}$ assumption holds if for any probabilistic polynomial-time algorithm $\mathcal{D}$, the following advantage is negligible in λ.*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{MLWE}}_{q,k,\ell,\eta}(\mathcal{D}) := |\Pr\left[b = 1 \ : \ \mathbf{A} \leftarrow\$ R_q^{k\times\ell}; \mathbf{s} \leftarrow\$ S_\eta^{\ell+k}; \mathbf{t} := [\mathbf{A}|\mathbb{I}_k] \cdot \mathbf{s} \bmod q; b \leftarrow \mathcal{D}(\mathbf{A}, \mathbf{t})\right] \\ \Pr\left[b = 1 \ : \ \mathbf{A} \leftarrow\$ R_q^{k\times\ell}; \mathbf{s} \leftarrow\$ S_\eta^{\ell+k}; \mathbf{t} \leftarrow\$ R_q^{k}; b \leftarrow \mathcal{D}(\mathbf{A}, \mathbf{t})\right]|.\end{aligned}$$

2.3 Offline-online multi-signature

Following [NRS21], we define a two-round multi-signature scheme tailored to the offline-online paradigm. A multi-signature MS consists of a tuple of algorithms $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{KAgg}, \mathsf{SignOff}, \mathsf{SignOn}, \mathsf{Agg}, \mathsf{Ver})$.

- $\mathsf{Setup}(1^\lambda)$ outputs public parameters pp. Throughout, we assume that pp is given as implicit input to all other algorithms.
- $\mathsf{Gen}()$ outputs a key pair $(\mathsf{pk}, \mathsf{sk})$
- $\mathsf{KAgg}(L)$ takes a set of public keys $L = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_n\}$ and deterministically outputs an aggregated public key $\tilde{\mathsf{pk}}$.
- $\mathsf{SignOff}(\mathsf{sk})$ is an offline signing algorithm that that can be run independently of the message μ to sign. It outputs an offline message off and some state information st.
- $\mathsf{SignOn}(\mathsf{st}, \mathsf{msgs}, \mathsf{sk}, \mu, \{\mathsf{pk}_2, \ldots, \mathsf{pk}_n\})$ is an online signing algorithm that takes as input the state information passed on to by $\mathsf{SignOff}$, offline messages $\mathsf{msgs} = \{\mathsf{off}_2, \ldots, \mathsf{off}_n\}$ from cosigners, a secret key sk, a message to sign μ, and cosigner's public keys $\{\mathsf{pk}_2, \ldots, \mathsf{pk}_n\}$. It outputs an online message on. Following the convention introduced in [BN06], each signer assign indices $1, \ldots, n$ to the signers, with itself being signer 1. In particular, these indices are merely local references to each signer and thus they are not identities.
- $\mathsf{Agg}(\mathsf{on}_1, \ldots, \mathsf{on}_n)$ takes online messages as input, and outputs an aggregated signature σ, which might potentially contain $\perp$.
- $\mathsf{Ver}(\tilde{\mathsf{pk}}, \mu, \sigma)$ takes an aggregated key $\tilde{\mathsf{pk}}$, a message μ, and a signature σ as input. It outputs 1 or 0.

Remark 2.10. Nick et al. [NRS21] additionally defines "an aggregator node" in their syntax to further optimize communication complexity of the protocol. We omit this optimization because as we shall see later, our security proof relies on each signer's ability to check individual outputs from co-signers.

In this work, we propose a scheme where cosigners may *abort* (indicated by $\mathsf{on} = \perp$ after running SignOn), which is inherent in the FSwA-based interactive multi-signature [DOTT22] [FH20] [ES16]. Hence, a single run of the protocol fails to output a valid signature with certain probability. To reduce such a correctness error, we define correctness so that it explicitly handles τ parallel repetitions of the signing protocol.

Definition 2.11 (MS-COR). *A two-round multi-signature scheme MS has correctness error δ if*

$$\Pr\left[0 \leftarrow \mathit{MS\text{-}COR}_{\mathsf{MS}}(\lambda, n, \tau)\right] \leq \delta$$

where the game $\mathit{MS\text{-}COR}_{\mathsf{MS}}$ is described in Game 1.

The following definition guarantees unforgeability of a multi-signature scheme with two rounds of interactions. Note that we explicitly allow the adversary to launch many signing sessions in parallel rather than forcing them to finish every signing attempt before starting the next one. This models real-world adversarial behaviors that exploit concurrent attacks as observed in Drijvers et al. [DEF+19] It is also crucial for the offline sign oracle $\mathsf{OSignOff}$ to not take any message as inputs, and instead a pair (μ, L) only gets included in the query set $\mathcal{Q}$ once queried to $\mathsf{OSignOn}$.

Game 1: $\mathsf{MS\text{-}COR}_{\mathsf{MS}}(\lambda)$

$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
for $i \in [1,n]$ **do**
  $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}()$
  for $j \in [1,\tau]$ **do**
    $(\mathsf{off}_{i,j}, \mathsf{st}_{i,j}) \leftarrow \mathsf{SignOff}(\mathsf{sk}_i)$
$\mathsf{msgs}_j := (\mathsf{off}_{1,j}, \ldots, \mathsf{off}_{n,j})$
$L := \{\mathsf{pk}_1, \ldots, \mathsf{pk}_n\}$
for $j \in [1,\tau]$ **do**
  for $i \in [1,n]$ **do**
    $\mathsf{on}_{i,j} \leftarrow \mathsf{SignOn}(\mathsf{st}_{i,j}, \mathsf{msgs}_j \setminus \{\mathsf{off}_{i,j}\}, \mathsf{sk}_i, \mu, L \setminus \{\mathsf{pk}_i\})$
  $\sigma_j \leftarrow \mathsf{Agg}(\mathsf{on}_{1,j}, \ldots, \mathsf{on}_{n,j})$
if $\exists j \in [1,\tau] : \sigma_j \neq \perp$ **then**
  return $\mathsf{Ver}(\mathsf{KAgg}(L), \mu, \sigma_j)$
else
  return 0

Game 2: $\mathsf{MS\text{-}UF\text{-}CMA}_{\mathsf{MS}}(\mathcal{A}, \lambda)$

$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{Gen}()$
$\mathsf{ctr} := 0$
$S := \varnothing$; $\mathcal{Q} := \varnothing$
$(L^*, \mu^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{OSignOn}, \mathsf{OSignOff}, \mathcal{H}}(\mathsf{pp}, \mathsf{pk}_1)$
if $(\mathsf{pk}_1 \notin L^*) \vee ((L^*, \mu^*) \in \mathcal{Q})$ **then**
  return 0
return $\mathsf{Ver}(\mathsf{KAgg}(L^*), \mu^*, \sigma^*)$

OSignOff
$\mathsf{ctr} := \mathsf{ctr} + 1$
$\mathsf{sid} := \mathsf{ctr}$; $S := S \cup \{\mathsf{sid}\}$
$(\mathsf{off}, \mathsf{st}_{\mathsf{sid}}) \leftarrow \mathsf{SignOff}(\mathsf{sk}_1)$
return off

OSignOn$(\mathsf{sid}, \mathsf{msgs}, \mu, \{\mathsf{pk}_2, \ldots, \mathsf{pk}_n\})$
if $\mathsf{sid} \notin S$ **then return** $\perp$
$\mathsf{on} \leftarrow \mathsf{SignOn}(\mathsf{st}_{\mathsf{sid}}, \mathsf{msgs}, \mathsf{sk}_1, \mu, \{\mathsf{pk}_2, \ldots, \mathsf{pk}_n\})$
$L := \{\mathsf{pk}_1, \ldots, \mathsf{pk}_n\}$
$\mathcal{Q} := \mathcal{Q} \cup \{(L, \mu)\}$
$S := S \setminus \{\mathsf{sid}\}$
return on

Definition 2.12 (MS-UF-CMA). *A two-round multi-signature scheme MS is said to be MS-UF-CMA secure in the random oracle model, if for any PPT adversary* $\mathcal{A}$

$$\mathsf{Adv}^{\textit{MS-UF-CMA}}_{\textit{MS}}(\mathcal{A}, \lambda) := \Pr\left[1 \leftarrow \textit{MS-UF-CMA}_{\textit{MS}}(\mathcal{A}, \lambda)\right] \leq \mathsf{negl}(\lambda)$$

where the game $\textit{MS-UF-CMA}_{\textit{MS}}$ *is described in Game 2 and* $\mathcal{H}$ *denotes the random oracle.*

As a special case, if the adversary makes no queries to the sign oracles OSignOff and OSignOn in Game 2 and its advantage is negligible, a scheme MS is said to be MS-UF-KOA (*unforgeable against key only attacks*).

2.4 General Forking Lemma

We restate the general forking lemma from [BN06].

Lemma 2.13 (General Forking Lemma). *Let* Q *be a number of queries and* C *be a set of size* $|C| > 2$. *Let* $\mathcal{B}$ *be a randomized algorithm that on input* in, $h_1, \ldots, h_Q$ *returns an index* $i \in [0, Q]$ *and a side output* out. *Let* IGen *be a randomized algorithm that we call the input generator. Let* $\mathcal{F}_{\mathcal{B}}$ *be a forking algorithm that works as in Alg. 1 given* in *as input and given black-box access to* $\mathcal{B}$. *Suppose the following probabilities.*

$$acc := \Pr[i \geq 1 : \mathsf{in} \leftarrow \mathsf{IGen}(1^\lambda); h_1, \ldots, h_Q \leftarrow\!\!\$\ C; (i, \mathsf{out}) \leftarrow \mathcal{B}(\mathsf{in}, h_1, \ldots, h_Q)]$$
$$frk := \Pr[b = 1 : \mathsf{in} \leftarrow \mathsf{IGen}(1^\lambda); (b, \mathsf{out}, \hat{\mathsf{out}}) \leftarrow \mathcal{F}_{\mathcal{B}}(\mathsf{in})]$$

Then

$$frk \geq acc \cdot \left(\frac{acc}{Q} - \frac{1}{|C|}\right).$$

Alternatively,

$$acc \leq \frac{Q}{|C|} + \sqrt{Q \cdot frk}.$$

Algorithm 1: $\mathcal{F}_{\mathcal{B}}(\mathsf{in})$

$\rho \leftarrow\!\$\ \{0,1\}^*$
$h_1, \ldots, h_Q \leftarrow\!\$\ C$
$(i, \mathsf{out}) \leftarrow \mathcal{B}(\mathsf{in}, h_1, \ldots, h_Q; \rho)$
if $i = 0$ **then**
    return $(0, \bot, \bot)$
$\hat{h}_i, \ldots, \hat{h}_Q \leftarrow\!\$\ C$
$(\hat{i}, \hat{\mathsf{out}}) \leftarrow \mathcal{B}(\mathsf{in}, h_1, \ldots, h_{i-1}, \hat{h}_i, \ldots, \hat{h}_Q; \rho)$
if $i = \hat{i} \wedge h_i \neq \hat{h}_i$ **then**
    return $(1, \mathsf{out}, \hat{\mathsf{out}})$
else
    return $(0, \bot, \bot)$

3 Our MuSig-L Scheme

3.1 Definition of the Scheme

See Protocol 1 for detailed specifications. The basic algorithms, such as $\mathsf{Setup}, \mathsf{Gen}$ and Ver closely follow non-optimized version of the Dilithium-G signature [DLL+17]. In the offline phase each party outputs m individual "commit" messages, followed by their own public key.

At the beginning of the online phase, a party P_1 performs a few sanity checks on the inputs. First, it checks that the offline messages from other parties do contain a correct set of co-signer's public keys. It then checks that its own public key $\mathbf{t}_1$ does not appear in the received messages. As we shall see in the next section, this is crucial for our security proof to go through, although we are not aware of any attacks in case duplicates are allowed. Finally, it verifies the sum of the mth commit messages $\mathbf{w}^{(m)}$ has an invertible element. This is to prevent the adversary from maliciously choosing their shares of commits so that the final sum $\tilde{\mathbf{w}} = \sum_{j=1}^{m} b^{(j)} \cdot \mathbf{w}^{(j)}$ completely cancels out.

If the inputs look reasonable, P_1 proceeds by hashing encoded offline messages to derive randomness used for sampling Gaussian nonces $b^{(j)}$'s. Since these are generated from spherical Gaussian, the algorithm Samp can be efficiently instantiated with existing samplers such as [HPRR20]. It then performs our generalized rejection sampling detailed in Section 3.2.

3.1.1 Parameters Each element of the secret signing key is chosen from $S_\eta \subseteq R$ parameterized by $\eta \geq 0$ consisting of small polynomials:

$$S_\eta = \{x \in R : \|x\|_\infty \leq \eta\}.$$

As our scheme is defined over a module of dimension $\ell + k$ every signing key belongs to $S_\eta^{\ell+k}$.

Moreover the *challenge set* $C \subseteq R$ parameterized by $\kappa \geq 0$ consists of small and sparse polynomials, which will be used as the image of random oracles $\mathsf{H}_{\mathrm{sig}}$ and $\mathsf{H}_{\mathrm{agg}}$:

$$C = \{c \in R : \|c\|_\infty = 1 \wedge \|c\|_1 = \kappa\}.$$

In particular, a set of differences $\bar{C} := \{c - c' \ : \ c, c' \in C \wedge c \neq c'\}$ consists of invertible elements thanks to Lemma 2.1.

Finally, correctness requires $q > 16\sigma_1 n$ (where n is the number of parties, cf. Theorem 3.3) and $\alpha\eta\kappa^2 < \sigma_1$ (cf. Lemma 3.1), and $2k\lceil \log_2 q \rceil + 1 > \ell + k$ is required by security (cf. Section 4.3).

3.2 Rejection Sampling

We now describe the rejection sampling algorithm used in the generation of a partial signature. For the sake of exposition, in this section *we ignore the subscript index i* indicating which signer generated a given vector or element, as we consider the view of only one signer.

Protocol 1: MuSig-L

The random oracles $\mathsf{H}_{\mathrm{agg}} : \{0,1\}^* \to C$, $\mathsf{H}_{\mathrm{sig}} : \{0,1\}^* \to C$, $\mathsf{H}_{\mathrm{non}} : \{0,1\}^* \to \{0,1\}^l$. $\langle S \rangle$ denotes unique encoding of a set S, e.g., lexicographical ordering. $||$ denotes concatenation of two strings.

$\mathsf{Setup}(1^\lambda)$

1: $\mathbf{A} \leftarrow\$ R_q^{k \times \ell}$
2: $\bar{\mathbf{A}} := [\mathbf{A} | \mathbb{I}_k]$
3: $\mathsf{pp} := \bar{\mathbf{A}}$
4: **return** pp

$\mathsf{Gen}()$

1: $\mathbf{s}_1 \leftarrow\$ S_\eta^{\ell+k}$
2: $\mathbf{t}_1 := \bar{\mathbf{A}}\mathbf{s}_1 \bmod q$
3: $(\mathsf{pk}, \mathsf{sk}) := (\mathbf{t}_1, \mathbf{s}_1)$
4: **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{Agg}(\mathsf{on}_1, \ldots, \mathsf{on}_n)$

1: **if** $\exists i \in [1,n] : \mathbf{z}_i = \perp$ **then**
2: **return** $\perp$
3: $\mathbf{z} := \sum_{i=1}^n \mathbf{z}_i$
4: $\sigma := (\tilde{\mathbf{w}}, \mathbf{z})$
5: **return** σ

$\mathsf{KAgg}(L)$

1: $\{\mathbf{t}_1, \ldots, \mathbf{t}_n\} := L$
2: **for** $i \in [1,n]$ **do**
3: $a_i := \mathsf{H}_{\mathrm{agg}}(\langle L \rangle, \mathbf{t}_i)$
4: $\tilde{\mathbf{t}} := \sum_{i=1}^n a_i \mathbf{t}_i \bmod q$
5: **return** $\tilde{\mathbf{t}}$

$\mathsf{Ver}(\mathsf{pk}, \sigma, \mu)$

1: $(\tilde{\mathbf{w}}, \mathbf{z}) := \sigma$
2: $\tilde{\mathbf{t}} := \mathsf{pk}$
3: $c := \mathsf{H}_{\mathrm{sig}}(\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}})$
4: **if** $\bar{\mathbf{A}}\mathbf{z} - c\tilde{\mathbf{t}} = \tilde{\mathbf{w}} \bmod q \wedge \|\mathbf{z}\|_2 \leq B_n$ **then**
5: **return** 1
6: **else**
7: **return** 0

$\mathsf{Samp}(r)$

1: Sample $b \sim \mathscr{D}_{\sigma_b}$ using randomness r
2: **return** b

$\mathsf{RejSamp}(\mathbf{v}, \mathbf{z}, (b^{(j)})_{j \in [m]})$

1: $\Sigma := (\sigma_1^2 + \sigma_y^2 \sum_{j=2}^m (b^{(j)})^* b^{(j)}) \cdot \mathbb{I}_{\ell+k}$
2: $\rho \leftarrow\$ [0,1]$
3: **if** $\rho \geq \min\left(\frac{\mathscr{D}^{\ell+k}(\mathbf{z})}{M \cdot \mathscr{D}^{\ell+k}_{\sqrt{\Sigma},\mathbf{v}}(\mathbf{z})}, 1 \right)$ **then**
4: **return** 0
5: **return** 1

$\mathsf{SignOff}(\mathsf{sk}_1)$

1: $\mathbf{s}_1 := \mathsf{sk}_1$
2: $\mathbf{y}_1^{(1)} \leftarrow \mathscr{D}_{\sigma_1}^{\ell+k}$
3: For $j \in [2,m] : \mathbf{y}_1^{(j)} \leftarrow \mathscr{D}_{\sigma_y}^{\ell+k}$
4: For $j \in [1,m] : \mathbf{w}_1^{(j)} := \bar{\mathbf{A}}\mathbf{y}_1^{(j)} \bmod q$
5: $\mathsf{com}_1 := (\mathbf{w}_1^{(1)}, \ldots, \mathbf{w}_1^{(m)})$
6: $\mathsf{off}_1 := (\mathbf{t}_1, \mathsf{com}_1)$
7: $\mathsf{st}_1 := (\mathbf{y}_1^{(1)}, \ldots, \mathbf{y}_1^{(m)}, \mathsf{com}_1)$
8: **return** $(\mathsf{off}_1, \mathsf{st}_1)$

$\mathsf{SignOn}(\mathsf{st}_1, \mathsf{msgs}, \mathsf{sk}_1, \mu, (\mathsf{pk}_2, \ldots, \mathsf{pk}_n))$

1: $(\mathbf{t}_i, \mathsf{com}_i)_{i \in [2,n]} := \mathsf{msgs}$
2: **if** $\langle (\mathbf{t}_i)_{i \in [2,n]} \rangle \neq \langle (\mathsf{pk}_i)_{i \in [2,n]} \rangle$ **then**
3: **return** $\perp$
4: **if** $\exists i \geq 2 : \mathbf{t}_i = \mathbf{t}_1$ **then**
5: **return** $\perp$
6: $L := \{\mathbf{t}_1, \ldots, \mathbf{t}_n\}$
7: $a_1 := \mathsf{H}_{\mathrm{agg}}(\langle L \rangle, \mathbf{t}_1)$
8: $\tilde{\mathbf{t}} := \mathsf{KAgg}(L)$
9: $W := \{\mathbf{t}_i || \mathsf{com}_i\}_{i \in [n]}$
10: $(r^{(j)})_{j \in [2,m]} := \mathsf{H}_{\mathrm{non}}(\langle W \rangle, \mu, \tilde{\mathbf{t}})$
11: $b^{(1)} := 1$
12: For $j \in [2,m] : b^{(j)} := \mathsf{Samp}(r^{(j)})$
13: For $j \in [1,m] : \mathbf{w}^{(j)} := \sum_{i=1}^n \mathbf{w}_i^{(j)} \bmod q$
14: $[w_1^{(m)}, \ldots, w_k^{(m)}]^T := \mathbf{w}^{(m)}$
15: **if** $w_1^{(m)} \notin R_q^\times$ **then**
16: **return** $\perp$
17: $\tilde{\mathbf{w}} := \sum_{j=1}^m b^{(j)} \cdot \mathbf{w}^{(j)} \bmod q$
18: $\tilde{\mathbf{y}}_1 := \sum_{j=1}^m b^{(j)} \cdot \mathbf{y}_1^{(j)}$
19: $c := \mathsf{H}_{\mathrm{sig}}(\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}})$
20: $\mathbf{v} := c \cdot a_1 \cdot \mathbf{s}_1$
21: $\mathbf{z}_1 := \mathbf{v} + \tilde{\mathbf{y}}_1$
22: **if** $\mathsf{RejSamp}(\mathbf{v}, \mathbf{z}_1, (b^{(j)})_{j \in [m]}) = 0 :$ **then**
23: $\mathbf{z}_1 := \perp$
24: $\mathsf{on}_1 := (\mathbf{z}_1, \tilde{\mathbf{w}})$
25: **return** on_1

Table 2: Parameters for our multi-signature. Further details can be found in Appendix D.

Parameter	Description
n	Number of parties
τ	Number of parallel repetitions
$N = \mathsf{poly}(\lambda)$	A power of two defining the degree of $f(X)$
$f(X) = X^N + 1$	The $2N$-th cyclotomic polynomial
$q = 5 \mod 8$	Prime modulus
$w = \lceil \log_2 q \rceil$	Logarithm of the modulus
$R = \mathbb{Z}[X]/(f(X))$	Cyclotomic ring
$R_q = \mathbb{Z}_q[X]/(f(X))$	Ring
k	The height of random matrix $\mathbf{A}$
ℓ	The width of random matrix $\mathbf{A}$
$B = \sigma_1\sqrt{N(\ell+k)}$	The maximum L^2-norm of signature share $\mathbf{z}_i \in R^{\ell+k}$
$B_n = \sqrt{n}B$	The maximum L^2-norm of combined signature $\mathbf{z} \in R^{\ell+k}$
κ	The maximum L^1-norm of challenge vector c
$C = \{c \in R : \|c\|_\infty = 1 \wedge \|c\|_1 = \kappa\}$	Challenge space where $\lvert C\rvert = \binom{N}{\kappa}2^\kappa$
η	The maximum L^∞-norm of the secret $\mathbf{s}$
$S_\eta = \{\mathbf{s} \in R : \|\mathbf{s}\|_\infty \leq \eta\}$	Set of small secrets
$T = \kappa^2\eta\sqrt{N(\ell+k)}$	Chosen to satisfy the hypotheses of Lemma 3.1
$\sigma_1 = \sigma_b\sigma_y\sqrt{N(2kw+1)(\ell+k)}$	Standard deviation of the Gaussian distribution
$\sigma_y = \frac{2^9}{\pi\sqrt{\pi}} 2^{\frac{2}{Nk}} q^{\frac{k}{\ell+k}} N^2 \sqrt{(kw+1)\left(2+N+\log\left((\ell+k)N\right)\right)}$	Standard deviation of the Gaussian distribution
$\sigma_b = \frac{2^{5/2}}{\sqrt{\pi}} \cdot 2^{\frac{2}{Nk}} N^{3/2}\sqrt{kw+1}$	Standard deviation of the Gaussian distribution
$\hat{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_1)$	Covariance matrix of the target Gaussian distribution
$\alpha = \frac{\sigma_1 - 1}{T}$	Parameter defining M
$t = \sqrt{\frac{N}{(\pi-1)\log_2 e}}$	Parameter defining M
$M = e^{t/\alpha + 1/(2\alpha^2)}$	The expected number of restarts until a single party can proceed
$M_n = M^n$	The expected number of restarts until all n parties proceed simultaneously
l	Output bit lengths of the random oracle $\mathsf{H}_{\mathrm{non}}$

To understand the distribution of the response $\mathbf{z}$, we start from analyzing the distribution of the masking vector $\tilde{\mathbf{y}} = \sum_{j=1}^m b^{(j)} \cdot \mathbf{y}^{(j)}$. The vectors $\mathbf{y}^{(j)}$ and the elements $b^{(j)}$ are sampled according different Gaussian distributions:

– The vectors $\mathbf{y}^{(j)} \in R^{\ell+k}$ are sampled from two discrete Gaussians with parameters $\sigma_1 > \sigma_y > 0$ so that $\mathbf{y}^{(1)}$ has higher entropy:

$$\mathbf{y}^{(1)} \leftarrow \mathscr{D}_{\sigma_1}^{\ell+k} \quad \wedge \quad \mathbf{y}^{(j)} \leftarrow \mathscr{D}_{\sigma_y}^{\ell+k} \text{ for all } 1 < j \leq m \ .$$

– The elements $b^{(j)} \in R$, $j = 1, \ldots, m$ are all sampled from a discrete Gaussian with parameter $\sigma_b > 0$ but the first, which is constant:

$$b^{(1)} = 1, \ b^{(j)} \leftarrow \mathscr{D}_{\sigma_b} \text{ for all } 1 < j \leq m \ .$$

Applying Lemma 2.7 with $b^{(j)}$ in the place of the z_i and $\mathbf{y}^{(j)}$ of y_i yields that the masking vector $\tilde{\mathbf{y}} = \mathbf{y}^{(1)} + \sum_{j=2}^m b^{(j)} \cdot \mathbf{y}^{(j)}$ is distributed according to a discrete Gaussian with parameter

$$\Sigma = s \cdot \mathbb{I}_{\ell+k} \in K_{\mathbb{R}}^{(\ell+k)\times(\ell+k)}, \text{ where } s = \sigma_1^2 + \sigma_y^2 \cdot \sum_{j=2}^m {b^{(j)}}^* b^{(j)} \tag{2}$$

As the products ${b^{(j)}}^* b^{(j)}$ are small and $\sigma_1 \gg \sigma_y$, we have that $\Sigma \approx \sigma_1^2 \cdot \mathbb{I}_{\ell+k}$. Generalizing the rejection sampling lemma to the case of sampling from ellipsoid discrete Gaussians allows to ensure that the distribution of $\mathbf{z}$ does not depend on the $b^{(j)}$, but it is always statistically close to a spherical Gaussian with parameter σ_1. However, as the first message of the protocol is sent in the clear instead of being committed to like in [DOTT21], we also need to make sure that in case of aborts this message does not leak information about the secret. In such a case, an adversary knows that the rejected instance was sampled from the coset $\Lambda_{\tilde{\mathbf{u}}}^{\perp}(\bar{\mathbf{A}})$, where $\tilde{\mathbf{u}} := \bar{\mathbf{A}}\left(\sum_j b^{(j)}\mathbf{y}^{(j)}\right) + c \cdot a \cdot \mathbf{t}$. Thus we need to further generalize the rejection sampling technique, to the case in which the adversary always knows from which coset the response has been sampled.

Lemma 3.1 summarizes the rejection sampling technique used in MuSig-L; the general result can be found in Appendix B. Its proof is similar to the proof of the original rejection sampling lemma, but relies on a new result about the concentration of the squared norm of ellipsoidal Gaussians (cf. Appendix A).

This is used in Lemma B.4 to show that the behavior of the two distributions is not that different when restricted to Gaussian samples from cosets. Finally, we extend the original generalized rejection sampling lemma [Lyu12, Lemma 4.7] to consider the case of the behavior of a pair of distributions over a subset of their domain (cf. Lemma B.2). Observe that the latter requires that the measure of the coset does not change significantly. All results are proved w.r.t. the canonical embedding.

Lemma 3.1 (Rejection Sampling Algorithm). *Let $\Lambda \in R^{\ell+k}$ be a lattice. Let $\alpha, T, m > 0$, $\varepsilon \leq 1/2$. Define $\sigma_1, \sigma_b, \sigma_y > 0$ such that $\sigma_y > \eta_\varepsilon(\Lambda^\perp)$, $\sigma_b > \eta_\varepsilon(R)$, and $\sigma_1 \geq \max\{\alpha T, \sigma_y \sigma_b \sqrt{Nm(\ell+k)}\}$.*

Consider a set $V \subseteq R^{1\times m} \times R^k \times R^{\ell+k}$. Let $h : V \to [0,1]$ be the composition of three probability distributions $h := \mathscr{D}_b \times \mathscr{D}_u \times \mathscr{D}_v$, where $\mathscr{D}_b$ returns $\{1, b^{(2)}, \ldots, b^{(m)}\}$ for $b^{(j)} \leftarrow \mathscr{D}_{\sigma_b}$, $\mathscr{D}_u$ returns a vector $\mathbf{u} \in R^k$, and $\mathscr{D}_v$ returns a vector $\mathbf{v} \in R^{\ell+k}$ such that $\|\mathbf{v}\| \leq T$.

Let $\Sigma = (\sigma_1^2 + \sigma_y^2 \sum_{j=2}^m b^{(j)^*} b^{(j)}) \cdot \mathbb{I}_{\ell+k}$, and $\widehat{\Sigma} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_1^2)$. Then, for any $t > 0$, $M := \exp(\pi/\alpha^2 + \pi t/\alpha)$, and $\epsilon := 2(1+\varepsilon)/(1-\varepsilon)\exp(-t^2(pi-1))$ the distribution of the following algorithm

RejSamp*:*

- $(b^{(1)}, \ldots, b^{(m)}, \mathbf{u}, \mathbf{v}) \leftarrow h$
- $\mathbf{z} \leftarrow \mathscr{D}^{\ell+k}_{\sqrt{\Sigma}, \mathbf{v}, \Lambda^\perp_{\mathbf{u}}}$
- *with probability* $1 - \min\left(1, \frac{\mathscr{D}^{\ell+k}_{\sqrt{\widehat{\Sigma}}}(\mathbf{z})}{M \cdot \mathscr{D}^{\ell+k}_{\sqrt{\Sigma}, \mathbf{v}}(\mathbf{z})}\right)$, *set* $\mathbf{z} := \perp$
- *output* $(\mathbf{z}, b^{(1)}, \ldots, b^{(m)}, \mathbf{u}, \mathbf{v})$

is within statistical distance $\frac{\epsilon}{2M} + \frac{2\varepsilon}{M}$ of the distribution of:

SimRS*:*

- $(b^{(1)}, \ldots, b^{(m)}, \mathbf{u}, \mathbf{v}) \leftarrow h$
- $\mathbf{z} \leftarrow \mathscr{D}^{\ell+k}_{\sqrt{\widehat{\Sigma}}, \Lambda^\perp_{\mathbf{u}}}$
- *with probability* $1 - 1/M$, *set* $\mathbf{z} := \perp$
- *output* $(\mathbf{z}, b^{(1)}, \ldots, b^{(m)}, \mathbf{u}, \mathbf{v})$

Moreover, RejSamp *outputs something with probability larger than $\frac{1-\epsilon}{M}(1 - \frac{4\varepsilon}{(1+\varepsilon)^2})$.*

Proof. In the following we show that our choice of parameters satisfies the hypotheses of Theorem B.1. The center trivially satisfies the requirements. Regarding the Gaussian parameters Σ and $\widehat{\Sigma}$, we have that:

$$\mathrm{s}_1(\Sigma) = \sigma_1^2 \left\| \left(1 + \frac{\sigma_y^2 \sum_{j=2}^m (b^{(j)})^* b^{(j)}}{\sigma_1^2}\right) \right\|^2 \leq \sigma_1^2 \left(1 + \frac{\sigma_y^2 \sigma_b^2 Nm}{\sigma_1^2}\right) \leq \sigma_1^2 \left(1 + \frac{1}{(\ell+k)}\right)$$

where the first inequality follows applying the triangular inequality, and Lemma 2.6 to bound the norm of the $b^{(j)}$'s. Setting $\beta := 1/2$, yields that trivially $\mathrm{s}_{\ell+k}(\Sigma) > \sigma_1^2/2$ and $\mathrm{s}_1(\Sigma) < (1 + \frac{2}{\ell+k})\sigma_1^2$; observing that $\mathrm{s}_1(\widehat{\Sigma}) = \mathrm{s}_1(\widehat{\Sigma}) = \sigma_1^2$ yields the thesis. □

Observe that efficient sampling from cosets requires a trapdoor for $\mathbf{A}$, which is not compatible with a reduction from MSIS with the matrix $\mathbf{A}$. However, we only use this lemma in the security reduction to prove that honest signing can be simulated, thus this sampling does not have to be efficient and we can carry through the reduction.

Lemma 3.2. *The definition of the signing algorithm of MuSig-L in Protocol 1 with the parameters in Table 2 satisfies the hypotheses of Lemma 3.1.*

The proof of Lemma 3.2 is a routine calculation, thus we defer it to Appendix D. Observe that the statistical distance is negligible, and the probability of returning something is larger than $1/M(1-\mathsf{negl}(\lambda))$ as $\varepsilon = 2^{-N}$ and t is set so that $\exp(-t(\pi-1) = 2^{-N} = \mathsf{negl}(\lambda)$.

3.3 Correctness and Efficiency Analysis

Theorem 3.3. *MuSig-L has correctness error $\delta = \left(1 - \frac{1}{M^n}\right)^\tau (1 + \mathsf{negl}(\lambda))$ when defined with the parameters in Table 2, i.e.,*

$$\Pr\left[0 \leftarrow \textit{MS-COR}_{\textit{MS}}(\lambda, n, \tau)\right] \leq \delta$$

where the game MS-COR$_{\textit{MS}}$ is described in Game 1.

Proof. The correctness game $\mathsf{MS\text{-}COR}_{\mathsf{MS}}$ returns 0 if for every $j \in [1,\tau]$ one of the following five events occurs :

1. The public keys have not been encoded correctly:

$$\mathsf{bad}_1 := (\langle(\mathbf{t}_i)_{i\in[2,n]}\rangle \neq \langle(\mathsf{pk}_i)_{i\in[2,n]}\rangle) \ .$$

   By definition of correctness, $\Pr\left[\mathsf{bad}_1\right] = 0$.

2. There is a collision on the public keys:

$$\mathsf{bad}_2 := (\exists i_1, i_2 \in [1,n] : \mathbf{t}_{i_1} = \mathbf{t}_{1_2}) \ .$$

   The vectors $\mathbf{t}_i$ are generated as the product of the public matrix $\bar{\mathbf{A}}$ times a secret vector sampled uniformly at random in the set $S_\eta^{\ell+k}$. As $\bar{\mathbf{A}} = [\mathbf{A}|\mathbb{I}_k]$, multiplication by $\bar{\mathbf{A}}$ is injective over the last k coefficients, and by the birthday argument we obtain the bound $\Pr\left[\mathsf{bad}_2\right] \leq \frac{n(n-1)}{|S_\eta^k|^2} = \frac{n(n-1)}{\eta^{kN}} \leq 2^{-\mathsf{poly}(\lambda)}$.

3. The invertibility condition is not satisfied:

$$\mathsf{bad}_3 := (\exists i \in [1,n] : w_1^{(m)} \notin R_q^\times) \ .$$

   Again, the vector $w_1^{(m)}$ is the product of the first row of $\bar{\mathbf{A}}$ times $\bar{\mathbf{y}} := \sum_{i=1}^n \mathbf{y}_i^{(m)}$. As $\sigma_y \geq \eta_\varepsilon(R)\sqrt{2}$, Lemma 2.7 applied component-wise to $\bar{\mathbf{y}}$ guarantees that each of its components is statistically close to a Gaussian with parameter $n\sigma_y$. Thus, by [LPR13, Corollary 7.5] (i.e., Lemma 4.2) $w_1^{(m)}$ is statistically close to uniform over the entire ring, (and the same for all the signers) and Lemma 2.2 ensures that: $\Pr\left[\mathsf{bad}_3\right] = \frac{2}{q^{N/2}} - \frac{1}{q^N} = 2^{-\mathsf{poly}(\lambda)}$.

4. One of the signers aborts during the RS step:

$$\mathsf{bad}_4 := (\exists i \in [1,n] : \mathsf{RejSamp}(\mathbf{v}, \mathbf{z}_1, (b^{(j)})_{j\in[m]}) = 0) \ .$$

   Lemma 3.2 shows that the hypotheses of Lemma 3.1 are satisfied, thus we have: $\Pr\left[\mathsf{bad}_4\right] \leq 1 - [\frac{1}{M} + \frac{\epsilon+\delta_2-\epsilon\delta_2}{M}]^n = 1 - \frac{1}{M^n} + \mathsf{negl}(\lambda)$.

5. The aggregated signature does not pass verification:

$$\mathsf{bad}_5 := (\mathsf{Ver}(\mathsf{KAgg}(L), \mu, \sigma_j) = 0) \ .$$

   The verification includes two checks, the linear relation and the norm bound. The former is trivially always satisfied, as the output of the hashes is the same for all signers once the ordering of the components of the input to each hash is set (e.g., to the lexicographical ordering). Analogously, the sampling of the $b^{(j)}$'s is deterministic once the nonces are computed, thus all the signers get the same $\tilde{\mathbf{w}}$. One only needs to estimate the probability that a honestly generated $\mathbf{z}$ does not satisfy the norm bound.
   By Lemma 3.1 $\mathbf{z}_i$ is statistically close to a Gaussian with parameter $\widehat{\Sigma} = \sigma_1^2 \cdot \mathbb{I}_{\ell+k}$. Hence by Lemma 2.6 we can bound the norm of $\mathbf{z}_i$ as: $\|\mathbf{z}_i\| \leq \mathrm{s}_1(\sqrt{\Sigma})\sqrt{N(\ell+k)} = \sigma_1\sqrt{N(\ell+k)} =: B$. Since the sum of n independent Gaussian samples with parameter σ_1 is statistically close to Gaussian with $\sqrt{n}\cdot\sigma_1$ (Lemma 2.7), the norm of the aggregate signature can be bound by $B_n = \sqrt{n}\cdot B$. Finally, we need to ensure that there is no wrap around when aggregating signatures, i.e., that $q/2 > n\|\mathbf{z}\|_\infty$. The norm of $\mathbf{z}$ can be bounded as $\|\mathbf{z}\|_\infty \leq 8\sigma_1$ by substituting $m = 1$, $\mathbf{c} = 1$, and $r = 8\sigma_b$ in Lemma B.6. The bound holds with probability smaller than 2^{-195}. Hence, $q > 16n\sigma_1$ is enough to avoid the wrap around in the aggregation. The bound holds with probability greater than $1 - 2^{-195}$. Thus $\Pr\left[\mathsf{bad}_5\right] \leq n2^{-195}$.

Putting everything together we get that

$$\Pr\left[0 \leftarrow \mathsf{MS\text{-}COR}_{\mathsf{MS}}(\lambda, n, \tau)\right] = \prod_{j=1}^{\tau}\sum_{i=1}^{5}\Pr\left[\mathsf{bad}_i\right] = \left(1 - \frac{1}{M^n} + n2^{-195} + \mathsf{negl}(\lambda)\right)^\tau \ .$$

□

3.3.1 Number of Aborts, Round Complexity, and Signature Length. In its standard form, this protocol requires some repetitions to deal with possible aborts in order to produce a signature. As the probability that a single signer outputs something is essentially $1/M$ (cf. Section 3.2), successful signing requires around M^n rounds, where $M = \exp(1/(2\alpha^2) + t/(2\alpha))$. Analogously to [DOTT21], having a small M^n requires $\alpha \propto n$. However, as long as $n = o(N^{-4})$ this does not imply an increase in the norm of each signature share, as $\sigma_1 = O(N^4\sqrt{N})$. Larger values of n yield an increase of roughly[6] $O(\log(n))$ in the signature size when comparing with Dilithium-G. Standard optimizations are possible. For example, running parallel executions of the same protocol at once yields at least one instance in which no signer aborts, thus the protocol is exactly 2-rounds. To this aim $\lambda \cdot \log\left(\frac{M_n}{M_n - 1}\right)$ parallel instances suffice.

The length of the signature only depends on B_n, as a standard optimization is for signatures to be composed by $(c, \mathbf{z})$ instead of $(\tilde{\mathbf{w}}, \mathbf{z})$. Verification in this case amounts to checking $c = \mathsf{H}_{\mathrm{sig}}(\bar{\mathbf{A}}\mathbf{z} - c\tilde{\mathbf{t}}, \mu, \tilde{\mathbf{t}})$ instead of $\bar{\mathbf{A}}\mathbf{z} - c\tilde{\mathbf{t}} = \tilde{\mathbf{w}}$ in addition to the norm check. With this optimization, signatures output by our scheme are $O(N(\ell + k)\log(\sigma_1\sqrt{n}))$ bits long. Relying on a trapdoor to simulate the signing oracle in the security proof affects the length of the signature, as it yields $\sigma_y = O(N^2\sqrt{N})$ and $\sigma_b = O(N^2)$ (cf. Section 4.3). Moreover, our rejection sampling technique requires σ_1 to be larger than $\sigma_y \cdot \sigma_b$, i.e., $\sigma_1 = O(N^4\sqrt{N})$. This implies that signature length is in fact $O(N(\ell + k)\log(N\sqrt{n}))$, i.e., larger than a non-optimized, single-user version of Dilithium-G by a factor $O(\log(N\sqrt{n}))$, but equal to [DOTT21][7].

4 Security Proofs

4.1 Reduction to LWE and SIS

For simplicity, we first consider a situation where the adversary does not make any sign oracle queries, i.e., $Q_s = 0$. Our proof closely follows "the double forking technique" of [MPSW19], except that in our scheme the aggregation coefficients a_i's are picked from the challenge space C consisting of small and sparse ring elements.

Theorem 4.1. *MuSig-L is MS-UF-KOA-secure under* $\mathsf{MSIS}_{q,k,\ell+1,\beta}$ *and* $\mathsf{MLWE}_{q,k,\ell,\eta}$ *assumptions with* $\beta = 8\kappa\sqrt{B_n^2 + \kappa^3}$. *Concretely, for any PPT adversary* $\mathcal{A}$ *against MS-UF-KOA that makes at most* Q *queries to the random oracles, there exist PPT adversaries* $\mathcal{B}'$ *and* $\mathcal{D}$ *such that*

$$\mathsf{Adv}^{\textit{MS-UF-KOA}}_{\textit{MuSig-L}}(\mathcal{A}) \leq \frac{Q(2Q+3)}{|C|} + \frac{2^{k+1}}{q^{kN/2}} + \mathsf{Adv}^{\mathsf{MLWE}}_{q,k,\ell,\eta}(\mathcal{B}') + \sqrt{\frac{Q^2}{|C|} + Q\sqrt{Q \cdot \mathsf{Adv}^{\mathsf{MSIS}}_{q,k,\ell+1,\beta}(\mathcal{D})}} \tag{3}$$

Proof sketch. We first sketch the high-level ideas of proof. The complete reduction algorithms are described in Alg. 6. First, we construct a "wrapper" $\mathcal{B}$ that internally invokes $\mathcal{A}$ to obtain a forged signature. The wrapper makes sure that a crucial query to $\mathsf{H}_{\mathrm{sig}}$ with input $\tilde{\mathbf{t}}^*$ is only made *after* the corresponding query to $\mathsf{H}_{\mathrm{agg}}$, and aborts otherwise (indicated by the $\mathsf{bad}_{\mathrm{agg}}$ flag). Moreover, it guarantees that no aggregated keys collide with each other, and aborts otherwise (indicated by the $\mathsf{bad}_{\mathrm{kcol}}$ flag). By the $\mathsf{MLWE}_{q,k,\ell,\eta}$ assumption, an honestly generated public key $\mathbf{t}_1 := \mathbf{t}^* = \bar{\mathbf{A}}\mathbf{s}^* \bmod q$ is indistinguishable with a uniformly random element in R_q. Hence, one can regard the input $(\mathbf{A}, \mathbf{t}^*)$ as an instance of the $\mathsf{MSIS}_{q,k,\ell+1,\beta}$ problem.

We then invoke the general forking lemma [BN06] (Lemma 2.13) twice. The first fork happens at the return value of $\mathsf{H}_{\mathrm{agg}} : \{0,1\}^* \to C$ (handled by the algorithm $\mathcal{D}$, internally running $\mathcal{C}$); the second fork happens at the return value of $\mathsf{H}_{\mathrm{sig}} : \{0,1\}^* \to C$ (handled by $\mathcal{C}$, internally running $\mathcal{B}$). Hence, after running the wrapper $\mathcal{B}$ in total 4 times, we get four forgeries satisfying the equations

[6] Observe that to avoid rejecting valid signatures due to arithmetic overflow q has to be larger than the size of the coefficients in the aggregated signature, i.e., the size of the ring has to grow linearly with $\sqrt{n}$ too. This is inherent to additively aggregating signatures. In general, having a larger q makes MSIS harder, but MLWE easier. Compensating for it requires increasing N by a factor $O\left(1 + \frac{\log n}{\log q_0}\right)$, where q_0 is the modulus used in the single party case. However, one usually sets $q > 2^{20}$, which makes $\frac{\log n}{\log q_0}$ less than 2 even for billions of users, and allows to neglect this factor in the signature size estimates.

[7] This is not immediately evident from their analysis of the signature length. In fact, verifiability requires a signature to include the randomness used to generate the commitments. Such randomness is sampled from a discrete Gaussian of parameter s, which has to be large enough to be sampled using a trapdoor, i.e., linear in N (cf. [DOTT21, Theorem 2]) times square root of the number of parties (since the *sum* of n Gaussian randomness is output as a signature). This adds a factor $O(\log(N\sqrt{n}))$ to their signature length, making it equivalent to ours.

$$\tilde{\mathbf{w}}_1 = \bar{\mathbf{A}}\mathbf{z}_1^* - c_1^* \sum_{i\neq 1} a_i \mathbf{t}_i - c_1^* a_{1,1}\mathbf{t}^* = \bar{\mathbf{A}}\hat{\mathbf{z}}_1^* - \hat{c}_1^* \sum_{i\neq 1} a_i \mathbf{t}_i - \hat{c}_1^* a_{1,1}\mathbf{t}^* \bmod q \tag{4}$$

$$\tilde{\mathbf{w}}_2 = \bar{\mathbf{A}}\mathbf{z}_2^* - c_2^* \sum_{i\neq 1} a_i \mathbf{t}_i - c_2^* a_{2,1}\mathbf{t}^* = \bar{\mathbf{A}}\hat{\mathbf{z}}_2^* - \hat{c}_2^* \sum_{i\neq 1} a_i \mathbf{t}_i - \hat{c}_2^* a_{2,1}\mathbf{t}^* \bmod q \tag{5}$$

where, in particular, $c_1^* \neq \hat{c}_1^*$, $c_2^* \neq \hat{c}_2^*$, and $a_{1,1} \neq a_{2,1}$ thanks to the forker algorithms $\mathcal{F}_{\mathcal{B}}$ and $\mathcal{F}_{\mathcal{C}}$, respectively. Rearranging the above equations, we get that

$$\bar{\mathbf{A}}\bar{\mathbf{z}}_1 - \bar{c}_1 \sum_{i\neq 1} a_i \mathbf{t}_i - \bar{c}_1 a_{1,1}\mathbf{t}^* = \mathbf{0} \bmod q \tag{6}$$

$$\bar{\mathbf{A}}\bar{\mathbf{z}}_2 - \bar{c}_2 \sum_{i\neq 1} a_i \mathbf{t}_i - \bar{c}_2 a_{2,1}\mathbf{t}^* = \mathbf{0} \bmod q \tag{7}$$

where $\bar{\mathbf{z}}_i = \mathbf{z}_i^* - \hat{\mathbf{z}}_i^*$ and $\bar{c}_i = c_i^* - \hat{c}_i^*$ for $i = 1, 2$, respectively. By multiplying the first equation by $\bar{c}_2$ and the second by $\bar{c}_1$, the second terms cancel out. This gives us

$$\bar{\mathbf{A}}(\bar{c}_2\bar{\mathbf{z}}_1 - \bar{c}_1\bar{\mathbf{z}}_2) - \bar{c}_1\bar{c}_2\bar{a}\mathbf{t}^* = 0. \tag{8}$$

where $\bar{a} = a_{1,1} - a_{2,1}$. Since $\bar{c}_1$, $\bar{c}_2$, and $\bar{a}$ are all non-zero and none of them are zero-divisors thanks to Lemma 2.1, $\bar{c}_1\bar{c}_2\bar{a}$ is guaranteed to be non-zero. Moreover, both $\bar{c}_2\bar{\mathbf{z}}_1 - \bar{c}_1\bar{\mathbf{z}}_2$ and $\bar{c}_1\bar{c}_2\bar{a}$ have relatively small L^2-norms. Thus we obtain a valid solution to SIS w.r.t. the instance matrix $[\mathbf{A}|\mathbb{I}_k|\mathbf{t}^*]$.

4.2 Switching Lemma

Before sketching our CMA security proof, we first prove a simple yet very powerful technical lemma. Let us first recall a regularity lemma in the ring setting.

Lemma 4.2 (Corollary 7.5 of [LPR13]). *Let R be the ring of integers in the mth cyclotomic number field K of degree N, and $q \geq 2$ an integer. For positive integers $k \leq n \leq \mathsf{poly}(N)$, let $\bar{\mathbf{A}} = [\mathbf{A}|\mathbb{I}_k] \in R_q^{k\times n}$, where $\mathbb{I}_k \in R_q^{k\times k}$ is the identity matrix and $\mathbf{A} \in R_q^{k\times(n-k)}$ is uniformly random. Then with probability $1 - 2^{-\Omega(N)}$ over the choice of $\mathbf{A}$, the distribution of $\bar{\mathbf{A}}\mathbf{x} \in R_q^k$, where $\mathbf{x} \leftarrow \mathscr{D}_\sigma^n$ with parameter $\sigma > 2N \cdot q^{k/n+2/(Nn)}$, satisfies that the probability of each of the q^{Nk} possible outcomes is in the interval $(1 \pm 2^{-\Omega(N)})q^{-Nk}$. In particular, it is within statistical distance $2^{-\Omega(N)}$ of the uniform distribution over R_q^k.*

As a consequence, we obtain the following switching lemma. This will make the hybrid arguments for simulation significantly modular as we shall see soon.

Lemma 4.3 (Switching lemma). *Let R, N, q, k, n and σ be as in Lemma 4.2. Consider the following two algorithms:*

$\mathcal{A}_0$: $\mathbf{A} \hookleftarrow R_q^{k\times(n-k)}$; $\mathbf{x} \leftarrow \mathscr{D}_\sigma^n$; $\mathbf{u} = [\mathbf{A}|\mathbb{I}_k] \cdot \mathbf{x} \bmod q$; output $(\mathbf{A}, \mathbf{x}, \mathbf{u})$.

$\mathcal{A}_1$: $\mathbf{A} \hookleftarrow R_q^{k\times(n-k)}$; $\mathbf{u} \hookleftarrow R_q^k$; $\mathbf{x} \leftarrow \mathscr{D}^n_{\Lambda_{\mathbf{u}}^\perp(\bar{\mathbf{A}}),\sigma}$; output $(\mathbf{A}, \mathbf{x}, \mathbf{u})$.

Then $\Delta(\mathcal{A}_0, \mathcal{A}_1) = 2^{-\Omega(N)}$.

Proof. Let (A_i, X_i, U_i) be random variables corresponding to outputs of $\mathcal{A}_i$. For any fixed $\mathbf{A} \in R_q^{k\times(n-k)}$, $\mathbf{x} \in R_q^n$ and $\mathbf{u} \in R_q^k$, we have

$$\begin{aligned}\Pr\left[(A_0, X_0, U_0) = (\mathbf{A}, \mathbf{x}, \mathbf{u})\right] &= \Pr[A_0 = \mathbf{A}] \cdot \Pr[X_0 = \mathbf{x}] \cdot \left[\mathbf{u} = \bar{\mathbf{A}}\mathbf{x} \bmod q\right] \\ &= \frac{1}{|R_q^{k\times(n-k)}|} \cdot \mathscr{D}_\sigma^n(\mathbf{x}) \cdot \left[\mathbf{x} \in \Lambda_{\mathbf{u}}^\perp(\bar{\mathbf{A}})\right]\end{aligned}$$

where we have let $\bar{\mathbf{A}} = [\mathbf{A}|\mathbb{I}_k]$, and $\left[\mathbf{u} = \bar{\mathbf{A}}\mathbf{x} \bmod q\right] = \left[\mathbf{x} \in \Lambda_{\mathbf{u}}^\perp(\bar{\mathbf{A}})\right]$ is the Iverson bracket notation: it has value 1 if the condition is met and 0 otherwise. Thus, the probability is 0 if $\mathbf{x} \notin \Lambda_{\mathbf{u}}^\perp(\bar{\mathbf{A}})$, and for $\mathbf{x} \in \Lambda_{\mathbf{u}}^\perp(\bar{\mathbf{A}})$, we have:

$$\begin{aligned}\Pr\left[(A_0, X_0, U_0) = (\mathbf{A}, \mathbf{x}, \mathbf{u})\right] &= \frac{1}{|R_q^{k\times(n-k)}|} \cdot \frac{\rho_\sigma(\mathbf{x})}{\rho_\sigma(R^n)} \\ &= \frac{1}{q^{Nk(n-k)}} \cdot \frac{\rho_\sigma(\mathbf{x})}{\rho_\sigma(\Lambda_{\mathbf{u}}^\perp)} \cdot \frac{\rho_\sigma(\Lambda_{\mathbf{u}}^\perp)}{\rho_\sigma(R^n)} \\ &= \frac{1}{q^{Nk(n-k)}} \cdot \mathscr{D}_{\Lambda_{\mathbf{u}}^\perp(\bar{\mathbf{A}}),\sigma}(\mathbf{x}) \cdot \frac{\rho_\sigma(\Lambda_{\mathbf{u}}^\perp)}{\rho_\sigma(R^n)}.\end{aligned}$$

In particular, summing over all possible choices of $\mathbf{x}$ for a fixed $\mathbf{A}$, we see that:

$$\frac{\rho_\sigma(\Lambda^\perp_{\mathbf{u}})}{\rho_\sigma(R^n)} = \Pr_{\mathbf{x}\sim\mathscr{D}^n_\sigma}\left[\mathbf{u} = \bar{\mathbf{A}}\mathbf{x} \bmod q\right].$$

We denote this probability $H_{\mathbf{A},\sigma}(\mathbf{u})$. In other words, $H_{\mathbf{A},\sigma}$ is the probability distribution over R_q^k given by $\bar{\mathbf{A}} \cdot \mathscr{D}^n_\sigma \bmod q$. To sum up, we have shown that for all $(\mathbf{A}, \mathbf{x}, \mathbf{u})$:

$$\Pr\left[(A_0, X_0, U_0) = (\mathbf{A}, \mathbf{x}, \mathbf{u})\right] = \begin{cases} \mathscr{D}^n_{\Lambda^\perp_{\mathbf{u}}(\bar{\mathbf{A}}),\sigma}(\mathbf{x}) \cdot \frac{H_{\mathbf{A},\sigma}(\mathbf{u})}{q^{Nk(n-k)}} & \text{if } \mathbf{x} \in \Lambda^\perp_{\mathbf{u}}(\bar{\mathbf{A}}), \\ 0 & \text{if } \mathbf{x} \notin \Lambda^\perp_{\mathbf{u}}(\bar{\mathbf{A}}). \end{cases}$$

On the other hand, still for fixed $\mathbf{A}, \mathbf{u}, \mathbf{x}$, we have:

$$\begin{aligned} \Pr\left[(A_1, X_1, U_1) = (\mathbf{A}, \mathbf{x}, \mathbf{u})\right] &= \frac{1}{|R_q^{k\times(n-k)}|} \cdot \frac{1}{|R_q^k|} \cdot \mathscr{D}^n_{\Lambda^\perp_{\mathbf{u}}(\bar{\mathbf{A}}),\sigma}(\mathbf{x}) \\ &= \frac{1}{q^{Nk(n-k)}} \cdot \frac{1}{q^{Nk}} \cdot \mathscr{D}^n_{\Lambda^\perp_{\mathbf{u}}(\bar{\mathbf{A}}),\sigma}(\mathbf{x}), \end{aligned}$$

and in particular this probability is non zero only for vectors $\mathbf{x} \in \Lambda^\perp_{\mathbf{u}}(\bar{\mathbf{A}})$. Therefore, the statistical distance $\Delta(\mathcal{A}_0, \mathcal{A}_1)$ can be written as:

$$\begin{aligned} \Delta(\mathcal{A}_0, \mathcal{A}_1) &= \sum_{\mathbf{A},\mathbf{u},\mathbf{x}} \left| \Pr\left[(A_0, X_0, U_0) = (\mathbf{A}, \mathbf{x}, \mathbf{u})\right] - \Pr\left[(A_1, X_1, U_1) = (\mathbf{A}, \mathbf{x}, \mathbf{u})\right] \right| \\ &= \sum_{\mathbf{A}\in R_q^{k\times(n-k)}, \mathbf{u}\in R_q^k} \frac{1}{q^{Nk(n-k)}} \sum_{\mathbf{x}\in\Lambda^\perp_{\mathbf{u}}(\bar{\mathbf{A}})} \mathscr{D}^n_{\Lambda^\perp_{\mathbf{u}}(\bar{\mathbf{A}}),\sigma}(\mathbf{x}) \cdot \left| H_{\mathbf{A},\sigma}(\mathbf{u}) - \frac{1}{q^{Nk}} \right| \\ &= \sum_{\mathbf{A}\in R_q^{k\times(n-k)}} \frac{1}{q^{Nk(n-k)}} \sum_{\mathbf{u}\in R_q^k} \left| H_{\mathbf{A},\sigma}(\mathbf{u}) - \frac{1}{q^{Nk}} \right| \\ &= \sum_{\mathbf{A}\in R_q^{k\times(n-k)}} \frac{1}{q^{Nk(n-k)}} \Delta\big(H_{\mathbf{A},\sigma}, \mathscr{U}_{R_q^k}\big), \end{aligned}$$

for $\mathscr{U}_{R_q^k}$ the uniform distribution on R_q^k. Now Lemma 4.2 says that there exists a subset $S \subset R_q^{k\times(n-k)}$ of cardinality at most $2^{-\Omega(N)}|R_q^{k\times(n-k)}|$ such that for all $\mathbf{A} \notin S$, we have $\Delta\big(H_{\mathbf{A},\sigma(\mathbf{u})}, \mathscr{U}_{R_q^k}\big) = 2^{-\Omega(N)}$. As a result:

$$\begin{aligned} \Delta(\mathcal{A}_0, \mathcal{A}_1) &= \sum_{\mathbf{A}\in S} \frac{1}{q^{Nk(n-k)}} \Delta\big(H_{\mathbf{A},\sigma}, \mathscr{U}_{R_q^k}\big) + \sum_{\mathbf{A}\notin S} \frac{1}{q^{Nk(n-k)}} \Delta\big(H_{\mathbf{A},\sigma}, \mathscr{U}_{R_q^k}\big) \\ &\le \frac{|S|}{q^{Nk(n-k)}} \cdot 1 + 1 \cdot 2^{-\Omega(N)} \le 2^{-\Omega(N)} \end{aligned}$$

as required. □

4.3 Simulating Nonces via Trapdoor Sampling

As a first step towards CMA security, recall that our goal is to simulate the view of the adversary interacting with an honest signer P_1. This essentially amounts to simulating the distribution of the offline messages $(\mathbf{w}_1^{(j)})_{j\in[m]}$, nonces $(b^{(j)})_{j\in[m]}$, challenge c, and $\mathbf{z}_1$, such that they satisfy the condition:

$$\bar{\mathbf{A}}\mathbf{z}_1 - c \cdot a_1 \cdot \mathbf{t}_1 = \sum_{j=1}^m b^{(j)}\mathbf{w}_1^{(j)} \bmod q. \tag{9}$$

From our rejection sampling lemma (Lemma 3.1), we can indeed simulate c and $\mathbf{z}_1$, and thus they already determine the sum $\tilde{\mathbf{w}}_1 := \sum_{j=1}^m b^{(j)}\mathbf{w}_1^{(j)} \bmod q$. However, since the offline commit messages $\mathbf{w}_1^{(j)}$ must be handed over to the adversary *before* the simulator sees adversary's $\mathbf{w}_i^{(j)}$, we are restricted to generating $b^{(j)}$'s such that they "explain" the above constraint for already fixed $(\mathbf{w}_1^{(j)})_{j\in[m]}$ and $\tilde{\mathbf{w}}_1$.

More concretely, after $\mathsf{OSignOff}$ outputs $\mathbf{w}_1^{(j)}$, whenever the simulator receives a query to $\mathsf{H}_{\mathrm{non}}$ or the online oracle $\mathsf{OSignOn}$ with adversarially chosen $\mathbf{w}_i^{(j)}$ and μ as inputs, the simulator already has to prepare $c, \mathbf{z}_1$ as well as $b^{(j)}$ satisfying (9), and then program the random oracles $\mathsf{H}_{\mathrm{non}}$ and $\mathsf{H}_{\mathrm{sig}}$ such that they output $b^{(j)}$'s and c, respectively.[8] We overcome this technical hurdle by making use of lattice-based trapdoor sampling. For readability we will drop the party index "1" for the rest of this subsection.

Recall that the first "commit" messages are computed as $\mathbf{w}^{(j)} := \bar{\mathbf{A}}\mathbf{y}^{(j)}$ for $j = 1, \ldots, m$. From the regularity result (Lemma 4.2), they are statistically indistinguishable with matrices uniformly sampled from $R_q^{k\times m}$. Now let us define suitable trapdoor generator and sampling algorithms to perform sign oracle simulation. To sample the vector $\mathbf{b} := [b^{(2)}, \ldots, b^{(m)}]$, we take advantage of the gadget-based trapdoor (Ring-)SIS inversion algorithm of [MP12]. (Recall that $b^{(1)} = 1$ so we only need to sample $m-1$ elements.) Let $\mathbf{W} := [\mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(m)}]$ be the parity check matrix for which we would like to obtain a trapdoor. For integers $k, w = \lceil \log_2 q \rceil, m' = kw + 1$, let $m = 2kw + 1$. Let $\mathbf{g}^T = [1, 2, 4, \ldots, 2^{w-1}]$ be a gadget vector and $\mathbf{G} = \mathbb{I}_k \otimes \mathbf{g} \in R^{k\times kw}$ be the corresponding gadget matrix. Then the Micciancio-Peikert trapdoor can be directly applied as follows.

- $\mathsf{TrapGen}(1^\lambda)$: It samples a uniformly random matrix $[\mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(kw+1)}] \in R_q^{k\times kw}$. It sets $\bar{\mathbf{W}} = [\mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(kw+1)}]$ and samples the trapdoor matrix $\mathbf{R} \in R^{kw\times kw}$ following the Gaussian $\mathscr{D}_{\bar{s}}^{kw\times kw}$ with parameter $\bar{s}$. Then the parity check matrix is defined as

$$\mathbf{W} = [\bar{\mathbf{W}} | \mathbf{G} - \bar{\mathbf{W}}\mathbf{R}] \in R_q^{k\times 2kw}. \tag{10}$$

  It outputs $(\mathbf{W}, \mathbf{R})$.

- $\mathsf{TrapSamp}(\mathbf{R}, \mathbf{w}', \sigma_b)$: Given a target vector $\mathbf{w}' \in R^k$, it samples a vector $\mathbf{b} \in R^{2kw} = R^{m-1}$, whose distribution is statistically close the discrete Gaussian $\mathscr{D}^{m-1}_{\Lambda^\perp_{\mathbf{w}'}(\mathbf{W}),\sigma_b}$ supported on the lattice coset

$$\Lambda^\perp_{\mathbf{w}'}(\mathbf{W}) := \left\{\mathbf{x} \in R^{2kw} \ : \ \mathbf{W}\cdot\mathbf{x} = \mathbf{w}' \mod q\right\}. \tag{11}$$

  This can be instantiated with [MP12, Alg. 3] or its adaptation in the module setting [BEP+21]. Note that efficiency of the sampler does not matter here, since trapdoor Gaussian sampling operations are only required by simulation, and parties in the actual protocol are never asked to do so.

4.3.1 Indistinguishability of W output by TrapGen We show that m columns of the parity check matrix $\mathbf{W}$ generated as above is indistinguishable with $[\mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(m)}]$ in the actual protocol. We apply the regularity lemma twice to argue that $\mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(m)}$ are uniform both in the actual protocol and in $\mathsf{TrapGen}$, up to an negligible error.

- In the actual protocol, the distribution of $\mathbf{w}^{(2)} = \bar{\mathbf{A}}\mathbf{y}^{(2)}, \ldots, \mathbf{w}^{(m)} = \bar{\mathbf{A}}\mathbf{y}^{(m)}$ is statistically close to uniform over $R_q^{k\times 2kw}$ if

$$\sigma_y > 2N \cdot q^{k/(\ell+k)+2/(N(\ell+k))} \tag{12}$$

  as required by Lemma 4.2. Note that, since the matrix $\bar{\mathbf{A}}$ is reused, the statistical distance grows linearly in m. The same remark applies to $\bar{\mathbf{W}}\mathbf{R}$ below.

- We now check the distribution of $\mathbf{W}$ output by $\mathsf{TrapGen}$. By construction, the distribution of kw columns $\bar{\mathbf{W}} = [\mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(kw+1)}]$ are uniform over $R_q^{k\times kw}$. As Lemma 4.2 requires a matrix to contain an identity submatrix, we need to bound the probability that $\bar{\mathbf{W}}$ contains no invertible submatrix, i.e., $\bar{\mathbf{W}}$ is not full rank. As our scheme assumes $q = 5 \mod 8$, we can use Lemma 2.2 to argue this only happens with negligible probability (see Appendix C.3 for formal analysis). Hence, we can indeed apply Lemma 4.2 to guarantee the distribution of $\bar{\mathbf{W}}\cdot\mathbf{R}$ is statistically close to uniform over $R_q^{k\times kw}$ if

$$\bar{s} > 2N \cdot q^{1/w+2/(Nkw)}. \tag{13}$$

[8] Note that once $b^{(j)}$'s are simulated, finding corresponding uniform randomness $r^{(j)}$'s are easy assuming that the Samp algorithm is "sampleable" [BCI+10]. Such a property can be for example satisfied by simple CDT-based samplers [Pei10, DDLL13] allowing one to recover a uniformly random coin given the output sample. Alternatively, one could simplify the description of our scheme by assuming the random oracle $\mathsf{H}_{\mathrm{non}}$ itself directly outputs Gaussian samples as in [ABB10].

Algorithm 2: Simulation of honest signing algorithm

$\mathcal{T}(\bar{\mathbf{A}}, a, \mathbf{s}, \mathbf{t})$

// Offline

for $j \in [1, m]$ **do**
  if $j = 1$ **then**
    $\mathbf{y}^{(1)} \leftarrow \mathscr{D}_{\sigma_1}^{\ell+k}$
    $b^{(1)} := 1$
  else
    $\mathbf{y}^{(j)} \leftarrow \mathscr{D}_{\sigma_y}^{\ell+k}$
    $b^{(j)} \leftarrow \mathscr{D}_{\sigma_b}$
  $\mathbf{w}^{(j)} := \bar{\mathbf{A}}\mathbf{y}^{(j)} \bmod q$
$\tilde{\mathbf{y}} := \sum_{j=1}^{m} b^{(j)}\mathbf{y}^{(j)}$

// Online

$c \leftarrow\$ C$
$\mathbf{v} := c \cdot a \cdot \mathbf{s}$
$\mathbf{z} := \mathbf{v} + \tilde{\mathbf{y}}$
$\rho \leftarrow\$ [0, 1)$
if $\rho > \min\left(\frac{\mathscr{D}^{\ell+k}_{\sqrt{\widehat{\Sigma}}}(\mathbf{z})}{M \cdot \mathscr{D}^{\ell+k}_{\sqrt{\Sigma},\mathbf{v}}(\mathbf{z})}, 1\right)$ **then**
  $\mathbf{z} := \perp$
return $(\bar{\mathbf{A}}, a, \mathbf{t}, (\mathbf{w}^{(j)}, b^{(j)})_{j\in[m]}, c, \mathbf{z})$

$\mathcal{S}(\bar{\mathbf{A}}, a, \mathbf{t})$

$\mathbf{w}^{(1)} \leftarrow\$ R_q^k$
$([\mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(m)}], \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^\lambda)$
$\mathbf{z} \leftarrow \mathscr{D}^{\ell+k}_{\sqrt{\widehat{\Sigma}}}$
$c \leftarrow\$ C$
$\mathbf{w}' := \bar{\mathbf{A}}\mathbf{z} - c \cdot a \cdot \mathbf{t} - \mathbf{w}^{(1)}$
$b^{(1)} := 1$
$(b^{(2)}, \ldots, b^{(m)}) \leftarrow \mathsf{TrapSamp}(\mathbf{R}, \mathbf{w}', \sigma_b)$
$\rho \leftarrow\$ [0, 1)$
if $\rho > 1/M$ **then**
  $\mathbf{z} := \perp$
return $(\bar{\mathbf{A}}, a, \mathbf{t}, (\mathbf{w}^{(j)}, b^{(j)})_{j\in[m]}, c, \mathbf{z})$

4.3.2 Indistinguishability of $b^{(j)}$'s output by TrapSamp

To sample from spherical Guassian with parameter σ_b, the gadget-based $\mathsf{TrapSamp}$ algorithm requires $\sigma_b \approx \mathrm{s}_1(\mathbf{R}) \cdot \mathrm{s}_1(\sqrt{\Sigma_{\mathbf{G}}})$ [MP12, §5.4] where $\sqrt{\Sigma_{\mathbf{G}}}$ is a parameter used when performing Gaussian sampling from a coset $\Lambda^{\perp}_{\mathbf{w}'}(\mathbf{G})$. As $\Sigma_{\mathbf{G}}$ is a constant, we only need to evaluate $\mathrm{s}_1(\mathbf{R})$, which is $\bar{s} \cdot O(\sqrt{Nkw} + \sqrt{Nk\log_2 q})$ from [MP12, §5.2]. Together with the condition (13) on $\bar{s}$ required by regularity, one can bound the parameter σ_b.

4.4 Oracle simulation lemma

Now let us turn to our main goal: security against adversaries that make concurrent chosen-message queries. For our honest party oracle simulator to succeed, we need the following lemma. It can proved via standard hybrid arguments, by invoking the switching lemma multiple times, indistinguishability of $\mathsf{TrapGen}$ and $\mathsf{TrapSamp}$ as stated above, and our generalized rejection sampling lemma (Lemma 3.1). Conditions on the parameters are detailed in Appendix D.

Lemma 4.4. *Let $\sigma_1, \sigma_y, \sigma_b, \Sigma, \widehat{\Sigma}, M$ be parameters satisfying conditions in Lemma 3.1 and Section 4.3. Suppose $q = 5 \mod 8$ as in Lemma 2.2. Let $\mathbf{A} \leftarrow\$ R^{k\times\ell}$, $\bar{\mathbf{A}} := [\mathbf{A}|\mathbb{I}_k]$, $\mathbf{s} \in S_\eta^{\ell+k}$, $\mathbf{t} := \bar{\mathbf{A}}\mathbf{s}$, $a \in C$. The output distributions of $\mathcal{T}$ and $\mathcal{S}$ in Alg. 2 are statistically indistinguishable.*

Proof. We prove via standard hybrid arguments. Each hybrid is detailed in Alg. 3.

- Hyb_0 is identical to $\mathcal{T}$.
- Hyb_1 is identical to Hyb_0, except that $\mathbf{w}^{(j)}$'s are sampled uniformly and $\mathbf{y}^{(j)}$'s are sampled from Gaussian defined over a coset $\Lambda^{\perp}_{\mathbf{w}^{(j)}}(\bar{\mathbf{A}}) = \left\{\mathbf{x} \in R^{k+\ell} : \bar{\mathbf{A}}\mathbf{x} = \mathbf{w}^{(j)} \bmod q\right\}$. From Lemma 4.3, Hyb_0 and Hyb_1 are statistically close.
- Hyb_2 is identical to Hyb_1, except that $\tilde{\mathbf{y}}$, a linear combination of $\mathbf{y}^{(j)}$'s, is directly sampled from Gaussian over a coset $\Lambda^{\perp}_{\tilde{\mathbf{w}}}(\bar{\mathbf{A}})$, where $\tilde{\mathbf{w}} = \sum_j b^{(j)}\mathbf{w}^{(j)} \mod q$. From Lemma 2.7, Hyb_1 and Hyb_2 are statistically close.
- Hyb_3 is identical to Hyb_2, except that $\mathbf{z}$ is sampled from Gaussian over a coset $\Lambda^{\perp}_{\mathbf{u}}$ centered at $\mathbf{v}$, where $\mathbf{u} = \tilde{\mathbf{w}} + c \cdot a \cdot \mathbf{t}$ and $\mathbf{v} = c \cdot a \cdot \mathbf{s}$. Clearly, the output distributions of Hyb_2 and Hyb_3 are equivalent.

– Hyb_4 is identical to Hyb_3, except that $\mathbf{z}$ is sampled from Gaussian over a coset $\Lambda_{\mathbf{u}}^{\perp}$ centered at 0 and it is output with constant probability $1/M$. From Lemma 3.1, Hyb_3 and Hyb_4 are statistically close.

– Hyb_5 is identical to Hyb_4, except that $\mathbf{w}' = \tilde{\mathbf{w}} - \mathbf{w}^{(1)}$ is uniformly sampled from R_q^k and a vector $[b^{(2)}, \ldots, b^{(m)}]$ is sampled from spherical Gaussian over a coset $\Lambda_{\mathbf{w}'}^{\perp}(\mathbf{W})$, where $\mathbf{W} = [\mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(m)}]$. From Lemma 4.3, Hyb_4 and Hyb_5 are statistically close.

– Hyb_6 is identical to Hyb_5, except that $\mathbf{z}$ is sampled from Gaussian over $R^{\ell+k}$ and $\tilde{\mathbf{w}}$ is defined as $\tilde{\mathbf{w}} = \bar{\mathbf{A}}\mathbf{z} - c \cdot a \cdot \mathbf{t}$. From Lemma 4.3, Hyb_5 and Hyb_6 are statistically close.

– Hyb_7 is identical to Hyb_6, except that a matrix $[\mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(m)}]$ is generated with the corresponding trapdoor $\mathbf{R}$. From indistinguishability of the $\mathsf{TrapGen}$ algorithm, Hyb_6 and Hyb_7 are statistically close.

– Hyb_8 is identical to Hyb_7, except that a vector $[b^{(2)}, \ldots, b^{(m)}]$ is sampled using the trapdoor sampling algorithm. From indistinguishability of the $\mathsf{TrapSamp}$ algorithm, Hyb_7 and Hyb_8 are statistically close.

Note that the distribution output by Hyb_8 is identical to one by $\mathcal{S}$. This concludes the proof.

□

4.5 MS-UF-CMA security of MuSig-L

Given the oracle simulation lemma, we are finally ready to conclude with our main theorem.

Theorem 4.5. *If MuSig-L is MS-UF-KOA-secure, then it is MS-UF-CMA-secure. Concretely, for any PPT adversary $\mathcal{X}$ against MS-UF-KOA that makes at most Q_h queries to the random oracles and in total Q_s queries to* $\mathsf{OSignOff}$ *and* $\mathsf{OSignOn}$*, there exists PPT adversary $\mathcal{A}$ such that*

$$\mathsf{Adv}^{\textit{MS-UF-CMA}}_{\textit{MuSig-L}}(\mathcal{X}) \leq 2(Q_h + Q_s)^2 \cdot \left(\frac{1 + 2^{-\Omega(N)}}{q^{kN}}\right)^m + \frac{(2Q_h + Q_s)^2}{\rho_{\sigma_b}(R)} + e \cdot (Q_s + 1) \cdot \left(Q_s \cdot \epsilon_s + \mathsf{Adv}^{\textit{MS-UF-KOA}}_{\textit{MuSig-L}}(\mathcal{A})\right)$$

where ϵ_s is determined by the statistical distance of Lemma 4.4.

Proof sketch We sketch the high-level ideas. The complete reduction algorithms are described in Alg. 7. We denote by $\mathsf{H}'_{\mathrm{agg}}, \mathsf{H}'_{\mathrm{non}}, \mathsf{H}'_{\mathrm{sig}}$ (resp. $\mathsf{H}_{\mathrm{agg}}, \mathsf{H}_{\mathrm{non}}, \mathsf{H}_{\mathrm{sig}}$) the random oracles in the $\mathsf{MS\text{-}UF\text{-}CMA}$ game (resp. $\mathsf{MS\text{-}UF\text{-}KOA}$ game), respectively.

On a high-level, we simulate the adversary's view by first producing a trapdoor for the outputs of $\mathsf{OSignOff}$, and then answer every query to $\mathsf{OSignOn}$ and $\mathsf{H}_{\mathrm{non}}$ using a known trapdoor. In a bit more detail:

– $\mathsf{OSignOff}$: For every concurrent session launched by the adversary, it stores in table WT party 1's commit messages $[\mathbf{w}_1^{(j)}, \ldots, \mathbf{w}_1^{(m)}]$ with a known trapdoor $\mathbf{R}$ produced by the $\mathsf{TrapGen}$ algorithm.

– $\mathsf{H}'_{\mathrm{non}}$: Whenever it receives a query of the form $(\{\mathbf{t}_i || \mathsf{com}_i\}_{i\in[n]}, \mu, \tilde{\mathbf{t}})$, it first makes sure that (1) there is no duplicate honest keys in the input, (2) the mth sum of commit message contains an invertible element, and (3) $\mathsf{com}_1 = [\mathbf{w}_1^{(j)}, \ldots, \mathbf{w}_1^{(m)}]$ (i.e., a commit message appended to the honest party's key $\mathbf{t}_1$) has been previously produced by $\mathsf{OSignOff}$. It does (3) by looking up the table WT, and if it finds a suitable trapdoor $\mathbf{R}$ associated with the corresponding session ID, $\mathsf{H}'_{\mathrm{non}}$ internally performs simulation following the procedures of Alg. 2, and then programs outputs of the random oracles $\mathsf{H}'_{\mathrm{sig}}$ and $\mathsf{H}'_{\mathrm{non}}$ accordingly. A simulated signature is finally stored in the table ST.

– $\mathsf{OSignOn}$: When the online oracle is queried, it always invokes $\mathsf{H}'_{\mathrm{non}}$ first and checks whether a simulated signature is recorded in ST. The simulation succeeds if that is the case, and aborts otherwise. The reason for aborts is that $\mathsf{H}'_{\mathrm{non}}$ must *not* produce simulated signatures for all queries, because it might be that the adversary will later submit a forgery based on the challenge c programmed inside $\mathsf{H}'_{\mathrm{non}}$. If that happens, the output of the external RO $\mathsf{H}_{\mathrm{sig}}$ is not consistent with that of $\mathsf{H}'_{\mathrm{sig}}$ anymore, and thus the reduction cannot win the $\mathsf{MS\text{-}UF\text{-}KOA}$ game. However, this issue can be handled by having $\mathsf{H}'_{\mathrm{non}}$ perform simulation only probabilistically, a proof technique similar to [DEF+19] and [DOTT22]. Such "bad challenges" are then kept in the table CT, and we evaluate the probability that the adversary does not use bad challenge to create a forgery.

– Note that this is exactly where appended public keys come in to play, and interestingly, they are crucial for proving security in the concurrent setting. Consider a modified scheme where $\mathsf{H}'_{\mathrm{non}}$ does

Algorithm 3: Hybrids for Lemma 4.4

$\mathsf{Hyb}_0(\bar{\mathbf{A}}, a, \mathbf{s}, \mathbf{t})$

for $j \in [1, m]$ **do**
  if $j = 1$ **then**
    $\mathbf{y}^{(1)} \leftarrow \mathscr{D}^{\ell+k}_{\sigma_1}$
    $b^{(1)} := 1$
  else
    $\mathbf{y}^{(j)} \leftarrow \mathscr{D}^{\ell+k}_{\sigma_y}$
    $b^{(j)} \leftarrow \mathscr{D}_{\sigma_b}$
  $\mathbf{w}^{(j)} := \bar{\mathbf{A}}\mathbf{y}^{(j)}$
$\tilde{\mathbf{y}} := \sum_{j=1}^{m} b^{(j)}\mathbf{y}^{(j)}$
$c \leftarrow\$ C$
$\mathbf{v} := c \cdot a \cdot \mathbf{s}$
$\mathbf{z} := \mathbf{v} + \tilde{\mathbf{y}}$
$\rho \leftarrow\$ [0, 1)$
if $\rho > \min\left(\frac{\mathscr{D}^{\ell+k}_{\sqrt{\widehat{\Sigma}},\mathbf{z}}(\mathbf{z})}{M \cdot \mathscr{D}^{\ell+k}_{\sqrt{\Sigma},\mathbf{v}}(\mathbf{z})}, 1\right)$ **then**
  $\mathbf{z} := \perp$
return $(\bar{\mathbf{A}}, a, \mathbf{t}, (\mathbf{w}^{(j)}, b^{(j)})_{j\in[m]}, c, \mathbf{z})$

$\mathsf{Hyb}_1(\bar{\mathbf{A}}, a, \mathbf{s}, \mathbf{t})$

for $j \in [1, m]$ **do**
  $\mathbf{w}^{(j)} \leftarrow\$ R_q^k$
  if $j = 1$ **then**
    $\mathbf{y}^{(1)} \leftarrow \mathscr{D}^{\ell+k}_{\Lambda^{\perp}_{\mathbf{w}^{(1)}}(\bar{\mathbf{A}}),\sigma_1}$
    $b^{(1)} := 1$
  else
    $\mathbf{y}^{(j)} \leftarrow \mathscr{D}^{\ell+k}_{\Lambda^{\perp}_{\mathbf{w}^{(j)}}(\bar{\mathbf{A}}),\sigma_y}$
    $b^{(j)} \leftarrow \mathscr{D}_{\sigma_b}$
$\tilde{\mathbf{y}} := \sum_{j=1}^{m} b^{(j)}\mathbf{y}^{(j)}$
$c \leftarrow\$ C$
$\mathbf{v} := c \cdot a \cdot \mathbf{s}$
$\mathbf{z} := \mathbf{v} + \tilde{\mathbf{y}}$
$\rho \leftarrow\$ [0, 1)$
if $\rho > \min\left(\frac{\mathscr{D}^{\ell+k}_{\sqrt{\widehat{\Sigma}},\mathbf{z}}(\mathbf{z})}{M \cdot \mathscr{D}^{\ell+k}_{\sqrt{\Sigma},\mathbf{v}}(\mathbf{z})}, 1\right)$ **then**
  $\mathbf{z} := \perp$
return $(\bar{\mathbf{A}}, a, \mathbf{t}, (\mathbf{w}^{(j)}, b^{(j)})_{j\in[m]}, c, \mathbf{z})$

$\mathsf{Hyb}_2(\bar{\mathbf{A}}, a, \mathbf{s}, \mathbf{t})$

for $j \in [1, m]$ **do**
  $\mathbf{w}^{(j)} \leftarrow\$ R_q^k$
  if $j = 1$ **then**
    $b^{(1)} := 1$
  else
    $b^{(j)} \leftarrow \mathscr{D}_{\sigma_b}$
$\tilde{\mathbf{w}} := \sum_{j=1}^{m} b^{(j)}\mathbf{w}^{(j)}$
$\tilde{\mathbf{y}} \leftarrow \mathscr{D}^{\ell+k}_{\Lambda^{\perp}_{\tilde{\mathbf{w}}}(\bar{\mathbf{A}}),\sqrt{\Sigma}}$
$c \leftarrow\$ C$
$\mathbf{v} := c \cdot a \cdot \mathbf{s}$
$\mathbf{z} := \mathbf{v} + \tilde{\mathbf{y}}$
$\rho \leftarrow\$ [0, 1)$
if $\rho > \min\left(\frac{\mathscr{D}^{\ell+k}_{\sqrt{\widehat{\Sigma}}}(\mathbf{z})}{M \cdot \mathscr{D}^{\ell+k}_{\sqrt{\Sigma},\mathbf{v}}(\mathbf{z})}, 1\right)$ **then**
  $\mathbf{z} := \perp$
return $(\bar{\mathbf{A}}, a, \mathbf{t}, (\mathbf{w}^{(j)}, b^{(j)})_{j\in[m]}, c, \mathbf{z})$

$\mathsf{Hyb}_3(\bar{\mathbf{A}}, a, \mathbf{s}, \mathbf{t})$

for $j \in [1, m]$ **do**
  $\mathbf{w}^{(j)} \leftarrow\$ R_q^k$
  if $j = 1$ **then**
    $b^{(1)} := 1$
  else
    $b^{(j)} \leftarrow \mathscr{D}_{\sigma_b}$
$\tilde{\mathbf{w}} := \sum_{j=1}^{m} b^{(j)}\mathbf{w}^{(j)}$
$c \leftarrow\$ C$
$\mathbf{v} := c \cdot a \cdot \mathbf{s}$
$\mathbf{u} := \tilde{\mathbf{w}} + c \cdot a \cdot \mathbf{t}$
$\mathbf{z} \leftarrow \mathscr{D}^{\ell+k}_{\Lambda^{\perp}_{\mathbf{u}}(\bar{\mathbf{A}}),\mathbf{v},\sqrt{\Sigma}}$
$\rho \leftarrow\$ [0, 1)$
if $\rho > \min\left(\frac{\mathscr{D}^{\ell+k}_{\sqrt{\widehat{\Sigma}}}(\mathbf{z})}{M \cdot \mathscr{D}^{\ell+k}_{\sqrt{\Sigma},\mathbf{v}}(\mathbf{z})}, 1\right)$ **then**
  $\mathbf{z} := \perp$
return $(\bar{\mathbf{A}}, a, \mathbf{t}, (\mathbf{w}^{(j)}, b^{(j)})_{j\in[m]}, c, \mathbf{z})$

$\mathsf{Hyb}_4(\bar{\mathbf{A}}, a, \mathbf{s}, \mathbf{t})$

for $j \in [1, m]$ **do**
  $\mathbf{w}^{(j)} \leftarrow\$ R_q^k$
  if $j = 1$ **then**
    $b^{(1)} := 1$
  else
    $b^{(j)} \leftarrow \mathscr{D}_{\sigma_b}$
$\tilde{\mathbf{w}} := \sum_{j=1}^{m} b^{(j)}\mathbf{w}^{(j)}$
$c \leftarrow\$ C$
$\mathbf{v} := c \cdot a \cdot \mathbf{s}$
$\mathbf{u} := \tilde{\mathbf{w}} + c \cdot a \cdot \mathbf{t}$
$\mathbf{z} \leftarrow \mathscr{D}^{\ell+k}_{\Lambda^{\perp}_{\mathbf{u}}(\bar{\mathbf{A}}),\sqrt{\widehat{\Sigma}}}$
$\rho \leftarrow\$ [0, 1)$
if $\rho > 1/M$ **then**
  $\mathbf{z} := \perp$
return $(\bar{\mathbf{A}}, a, \mathbf{t}, (\mathbf{w}^{(j)}, b^{(j)})_{j\in[m]}, c, \mathbf{z})$

$\mathsf{Hyb}_5(\bar{\mathbf{A}}, a, \mathbf{s}, \mathbf{t})$

for $j \in [1, m]$ **do**
  $\mathbf{w}^{(j)} \leftarrow\$ R_q^k$
$\tilde{\mathbf{w}} \leftarrow\$ R_q^k$
$\mathbf{w}' := \tilde{\mathbf{w}} - \mathbf{w}^{(1)}$
$\mathbf{W} := [\mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(m)}]$
$b^{(1)} := 1$
$[b^{(2)}, \ldots, b^{(m)}] \leftarrow \mathscr{D}^{m-1}_{\Lambda^{\perp}_{\mathbf{w}'}(\mathbf{W}),\sigma_b}$
$c \leftarrow\$ C$
$\mathbf{v} := c \cdot a \cdot \mathbf{s}$
$\mathbf{u} := \tilde{\mathbf{w}} + c \cdot a \cdot \mathbf{t}$
$\mathbf{z} \leftarrow \mathscr{D}^{\ell+k}_{\Lambda^{\perp}_{\mathbf{u}}(\bar{\mathbf{A}}),\sqrt{\widehat{\Sigma}}}$
$\rho \leftarrow\$ [0, 1)$
if $\rho > 1/M$ **then**
  $\mathbf{z} := \perp$
return $(\bar{\mathbf{A}}, a, \mathbf{t}, (\mathbf{w}^{(j)}, b^{(j)})_{j\in[m]}, c, \mathbf{z})$

$\mathsf{Hyb}_6(\bar{\mathbf{A}}, a, \mathbf{s}, \mathbf{t})$

for $j \in [1, m]$ **do**
  $\mathbf{w}^{(j)} \leftarrow\$ R_q^k$
$\mathbf{W} := [\mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(m)}]$
$b^{(1)} := 1$
$c \leftarrow\$ C$
$\mathbf{z} \leftarrow \mathscr{D}^{\ell+k}_{\sqrt{\widehat{\Sigma}}}$
$\mathbf{v} := c \cdot a \cdot \mathbf{s}$
$\mathbf{u} := \bar{\mathbf{A}}\mathbf{z}$
$\tilde{\mathbf{w}} := \mathbf{u} - c \cdot a \cdot \mathbf{t}$
$\mathbf{w}' := \tilde{\mathbf{w}} - \mathbf{w}^{(1)}$
$[b^{(2)}, \ldots, b^{(m)}] \leftarrow \mathscr{D}^{m-1}_{\Lambda^{\perp}_{\mathbf{w}'}(\mathbf{W}),\sigma_b}$
$\rho \leftarrow\$ [0, 1)$
if $\rho > 1/M$ **then**
  $\mathbf{z} := \perp$
return $(\bar{\mathbf{A}}, a, \mathbf{t}, (\mathbf{w}^{(j)}, b^{(j)})_{j\in[m]}, c, \mathbf{z})$

$\mathsf{Hyb}_7(\bar{\mathbf{A}}, a, \mathbf{s}, \mathbf{t})$

$\mathbf{w}^{(1)} \leftarrow\$ R_q^k$
$(\mathbf{W}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^\lambda)$
$b^{(1)} := 1$
$c \leftarrow\$ C$
$\mathbf{z} \leftarrow \mathscr{D}^{\ell+k}_{\sqrt{\widehat{\Sigma}}}$
$\mathbf{v} := c \cdot a \cdot \mathbf{s}$
$\mathbf{u} := \bar{\mathbf{A}}\mathbf{z}$
$\tilde{\mathbf{w}} := \mathbf{u} - c \cdot a \cdot \mathbf{t}$
$\mathbf{w}' := \tilde{\mathbf{w}} - \mathbf{w}^{(1)}$
$[b^{(2)}, \ldots, b^{(m)}] \leftarrow \mathscr{D}^{m-1}_{\Lambda^{\perp}_{\mathbf{w}'}(\mathbf{W}),\sigma_b}$
$\rho \leftarrow\$ [0, 1)$
if $\rho > 1/M$ **then**
  $\mathbf{z} := \perp$
return $(\bar{\mathbf{A}}, a, \mathbf{t}, (\mathbf{w}^{(j)}, b^{(j)})_{j\in[m]}, c, \mathbf{z})$

$\mathsf{Hyb}_8(\bar{\mathbf{A}}, a, \mathbf{s}, \mathbf{t})$

$\mathbf{w}^{(1)} \leftarrow\$ R_q^k$
$(\mathbf{W}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^\lambda)$
$b^{(1)} := 1$
$c \leftarrow\$ C$
$\mathbf{z} \leftarrow \mathscr{D}^{\ell+k}_{\sqrt{\widehat{\Sigma}}}$
$\mathbf{v} := c \cdot a \cdot \mathbf{s}$
$\mathbf{u} := \bar{\mathbf{A}}\mathbf{z}$
$\tilde{\mathbf{w}} := \mathbf{u} - c \cdot a \cdot \mathbf{t}$
$\mathbf{w}' := \tilde{\mathbf{w}} - \mathbf{w}^{(1)}$
$[b^{(2)}, \ldots, b^{(m)}] \leftarrow \mathsf{TrapSamp}(\mathbf{R}, \mathbf{w}', \sigma_b)$
$\rho \leftarrow\$ [0, 1)$
if $\rho > 1/M$ **then**
  $\mathbf{z} := \perp$
return $(\bar{\mathbf{A}}, a, \mathbf{t}, (\mathbf{w}^{(j)}, b^{(j)})_{j\in[m]}, c, \mathbf{z})$

not take individual public keys, i.e., it simply derives randomness via $\mathsf{H}'_{\mathrm{non}}(\langle \mathsf{com}_i \rangle_{i\in[n]}, \mu, \tilde{\mathbf{t}})$. It is easy to see that the simulator would have a hard time looking up the right trapdoor to perform simulation: say $\mathsf{OSignOff}$ has produced $(\mathsf{com}_1, \mathbf{R})$ in session 1 and $(\mathsf{com}'_1, \mathbf{R}')$ in session 2, respectively. Now, if the adversary queries $\mathsf{H}'_{\mathrm{non}}$ with input $((\mathsf{com}_1, \mathsf{com}'_1), \mu, \tilde{\mathbf{t}})$ there is no way for the simulator to determine which trapdoor should be used for performing simulation to sign a queried message μ. E.g. if the simulator uses a trapdoor $\mathbf{R}$, and the adversary later queries $\mathsf{OSignOn}$ in session 2 with μ and com_1 (by malicously claiming com_1 to be adversary's offline commit), a signature previously simulated by $\mathsf{H}'_{\mathrm{non}}$ is clearly invalid. Essentially the same issue happens if $\mathbf{t}_1$ occurs multiple times in the key list L.

Since we are not aware of any attacks breaking this modified scheme, or even against simpler hashing such as $\mathsf{H}'_{\mathrm{non}}(\sum_{i=1}^{n} \mathsf{com}_i, \mu, \tilde{\mathbf{t}})$ similar to $\mathsf{MuSig2}$, we highlight proving security of optimized nonce derivation as an interesting direction for future work. One plausible approach would be hashing the session ID sid together with $(\sum_{i=1}^{n} \mathsf{com}_i, \mu, \tilde{\mathbf{t}})$. This way, a simulator can uniquely associate each $\mathsf{H}'_{\mathrm{non}}$ query to a particular session and the corresponding trapdoor.

Acknowledgment

The authors are grateful to Claudio Orlandi for discussions in the earlier stages of this work. We thank Carsten Baum, Katharina Boudgoust, Mark Simkin, and anonymous reviewers of CRYPTO 2022 for helpful comments and discussions. Cecilia Boschini has been supported by the Università della Svizzera Italiana under the SNSF project No. 182452, and by the Postdoc.Mobility grant No. P500PT_203075. Akira Takahashi has been supported by the Carlsberg Foundation under the Semper Ardens Research Project CF18-112 (BCM); the European Research Council (ERC) under the European Unions's Horizon 2020 research and innovation programme under grant agreement No. 803096 (SPEC).

References

AB21. H. K. Alper and J. Burdges. Two-round trip schnorr multi-signatures via delinearized witnesses. In *CRYPTO 2021, Part I*, vol. 12825 of *LNCS*, pp. 157–188, Virtual Event, 2021. Springer, Heidelberg.

ABB10. S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *CRYPTO 2010*, vol. 6223 of *LNCS*, pp. 98–115. Springer, Heidelberg, 2010.

AFLT16. M. Abdalla, P.-A. Fouque, V. Lyubashevsky, and M. Tibouchi. Tightly secure signatures from lossy identification schemes. *Journal of Cryptology*, 29(3):597–631, 2016.

AGHS13. S. Agrawal, C. Gentry, S. Halevi, and A. Sahai. Discrete Gaussian leftover hash lemma over infinite domains. In *ASIACRYPT 2013, Part I*, vol. 8269 of *LNCS*, pp. 97–116. Springer, Heidelberg, 2013.

AKSY21. S. Agrawal, E. Kirshanova, D. Stehle, and A. Yadav. Can round-optimal lattice-based blind signatures be practical? Cryptology ePrint Archive, Report 2021/1565, 2021. https://eprint.iacr.org/2021/1565.

ASY22. S. Agrawal, D. Stehlé, and A. Yadav. Round-optimal lattice-based threshold signatures, revisited. In *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, vol. 229 of *LIPIcs*, pp. 8:1–8:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

Ban93. W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.

BCI+10. E. Brier, J.-S. Coron, T. Icart, D. Madore, H. Randriam, and M. Tibouchi. Efficient indifferentiable hashing into ordinary elliptic curves. In *CRYPTO 2010*, vol. 6223 of *LNCS*, pp. 237–254. Springer, Heidelberg, 2010.

BD21. M. Bellare and W. Dai. Chain reductions for multi-signatures and the HBMS scheme. In *ASIACRYPT 2021, Part IV*, vol. 13093 of *LNCS*, pp. 650–678. Springer, Heidelberg, 2021.

BDL+18. C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. In *SCN 18*, vol. 11035 of *LNCS*, pp. 368–385. Springer, Heidelberg, 2018.

BEP+21. P. Bert, G. Eberhart, L. Prabel, A. Roux-Langlois, and M. Sabt. Implementation of lattice trapdoors on modules and applications. In *Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021, Daejeon, South Korea, July 20-22, 2021, Proceedings*, vol. 12841 of *Lecture Notes in Computer Science*, pp. 195–214. Springer, 2021.

BGG+18. D. Boneh, R. Gennaro, S. Goldfeder, A. Jain, S. Kim, P. M. R. Rasmussen, and A. Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. In *CRYPTO 2018, Part I*, vol. 10991 of *LNCS*, pp. 565–596. Springer, Heidelberg, 2018.

BK20. D. Boneh and S. Kim. One-time and interactive aggregate signatures from lattices. *preprint*, 2020.

BKP13. R. Bendlin, S. Krehbiel, and C. Peikert. How to share a lattice trapdoor: Threshold protocols for signatures and (H)IBE. In *ACNS 13*, vol. 7954 of *LNCS*, pp. 218–236. Springer, Heidelberg, 2013.

BLL+21. F. Benhamouda, T. Lepoint, J. Loss, M. Orrù, and M. Raykova. On the (in)security of ROS. In *EUROCRYPT 2021, Part I*, vol. 12696 of *LNCS*, pp. 33–53. Springer, Heidelberg, 2021.

BN06. M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *ACM CCS 2006*, pp. 390–399. ACM Press, 2006.

BRL21. K. Boudgoust and A. Roux-Langlois. Non-interactive half-aggregate signatures based on module lattices - a first attempt. 2021. https://eprint.iacr.org/2021/263.

Cor00. J.-S. Coron. On the exact security of full domain hash. In *CRYPTO 2000*, vol. 1880 of *LNCS*, pp. 229–235. Springer, Heidelberg, 2000.

DDLL13. L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal Gaussians. In *CRYPTO 2013, Part I*, vol. 8042 of *LNCS*, pp. 40–56. Springer, Heidelberg, 2013.

DEF+19. M. Drijvers, K. Edalatnejad, B. Ford, E. Kiltz, J. Loss, G. Neven, and I. Stepanovs. On the security of two-round multi-signatures. In *2019 IEEE Symposium on Security and Privacy*, pp. 1084–1101. IEEE Computer Society Press, 2019.

DLL+17. L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS - dilithium: Digital signatures from module lattices. *IACR Cryptol. ePrint Arch.*, p. 633, 2017.

DOTT21. I. Damgård, C. Orlandi, A. Takahashi, and M. Tibouchi. Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices. In *PKC 2021, Part I*, vol. 12710 of *LNCS*, pp. 99–130. Springer, Heidelberg, 2021.

DOTT22. I. Damgård, C. Orlandi, A. Takahashi, and M. Tibouchi. Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices. *J. Cryptol.*, 35(2):14, 2022.

DPSZ12. I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias. Multiparty computation from somewhat homomorphic encryption. In *CRYPTO 2012*, vol. 7417 of *LNCS*, pp. 643–662. Springer, Heidelberg, 2012.

ES16. R. El Bansarkhani and J. Sturm. An efficient lattice-based multisignature scheme with applications to bitcoins. In *CANS 16*, vol. 10052 of *LNCS*, pp. 140–155. Springer, Heidelberg, 2016.

FH20. M. Fukumitsu and S. Hasegawa. A lattice-based provably secure multisignature scheme in quantum random oracle model. In *ProvSec 2020*, vol. 12505 of *LNCS*, pp. 45–64. Springer, Heidelberg, 2020.

FSZ22. N. Fleischhacker, M. Simkin, and Z. Zhang. Efficient synchronized multi-signatures from lattices. *IACR Cryptol. ePrint Arch.*, 2022.

GJKR07. R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure distributed key generation for discrete-log based cryptosystems. *Journal of Cryptology*, 20(1):51–83, 2007.

GKMN21. F. Garillot, Y. Kondi, P. Mohassel, and V. Nikolaenko. Threshold Schnorr with stateless deterministic signing from standard assumptions. In *CRYPTO 2021, Part I*, vol. 12825 of *LNCS*, pp. 127–156, Virtual Event, 2021. Springer, Heidelberg.

GPV08. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *40th ACM STOC*, pp. 197–206. ACM Press, 2008.

HPRR20. J. Howe, T. Prest, T. Ricosset, and M. Rossi. Isochronous gaussian sampling: From inception to implementation. In *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pp. 53–71. Springer, Heidelberg, 2020.

KG20. C. Komlo and I. Goldberg. FROST: flexible round-optimized schnorr threshold signatures. In *Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers*, vol. 12804 of *Lecture Notes in Computer Science*, pp. 34–65. Springer, 2020.

KLS18. E. Kiltz, V. Lyubashevsky, and C. Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In *EUROCRYPT 2018, Part III*, vol. 10822 of *LNCS*, pp. 552–586. Springer, Heidelberg, 2018.

LN17. V. Lyubashevsky and G. Neven. One-shot verifiable encryption from lattices. In *EUROCRYPT 2017, Part I*, vol. 10210 of *LNCS*, pp. 293–323. Springer, Heidelberg, 2017.

LPR13. V. Lyubashevsky, C. Peikert, and O. Regev. A toolkit for ring-LWE cryptography. In *EUROCRYPT 2013*, vol. 7881 of *LNCS*, pp. 35–54. Springer, Heidelberg, 2013.

LS15. A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3):565–599, 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, Part I*, vol. 10820 of *LNCS*, pp. 204–224. Springer, Heidelberg, 2018.

Lyu12. V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT 2012*, vol. 7237 of *LNCS*, pp. 738–755. Springer, Heidelberg, 2012.

Mic02. D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In *43rd FOCS*, pp. 356–365. IEEE Computer Society Press, 2002.

MJ19. C. Ma and M. Jiang. Practical lattice-based multisignature schemes for blockchains. *IEEE Access*, 7:179765–179778, 2019.

MOR01. S. Micali, K. Ohta, and L. Reyzin. Accountable-subgroup multisignatures: Extended abstract. In *ACM CCS 2001*, pp. 245–254. ACM Press, 2001.

MP12. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, vol. 7237 of *LNCS*, pp. 700–718. Springer, Heidelberg, 2012.

MP13. D. Micciancio and C. Peikert. Hardness of SIS and LWE with small parameters. In *CRYPTO 2013, Part I*, vol. 8042 of *LNCS*, pp. 21–39. Springer, Heidelberg, 2013.

MPSW19. G. Maxwell, A. Poelstra, Y. Seurin, and P. Wuille. Simple schnorr multi-signatures with applications to bitcoin. *Des. Codes Cryptogr.*, 87(9):2139–2164, 2019.

NKDM03. A. Nicolosi, M. N. Krohn, Y. Dodis, and D. Mazières. Proactive two-party signatures for user authentication. In *NDSS 2003*. The Internet Society, 2003.

NRS21. J. Nick, T. Ruffing, and Y. Seurin. MuSig2: Simple two-round Schnorr multi-signatures. In *CRYPTO 2021, Part I*, vol. 12825 of *LNCS*, pp. 189–221, Virtual Event, 2021. Springer, Heidelberg.

NRSW20. J. Nick, T. Ruffing, Y. Seurin, and P. Wuille. MuSig-DN: Schnorr multi-signatures with verifiably deterministic nonces. In *ACM CCS 2020*, pp. 1717–1731. ACM Press, 2020.

Pei10. C. Peikert. An efficient and parallel Gaussian sampler for lattices. In *CRYPTO 2010*, vol. 6223 of *LNCS*, pp. 80–97. Springer, Heidelberg, 2010.

Reg09. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.

RY07. T. Ristenpart and S. Yilek. The power of proofs-of-possession: Securing multiparty signatures against rogue-key attacks. In *EUROCRYPT 2007*, vol. 4515 of *LNCS*, pp. 228–245. Springer, Heidelberg, 2007.

SS01. D. R. Stinson and R. Strobl. Provably secure distributed Schnorr signatures and a (t, n) threshold scheme for implicit certificates. In *ACISP 01*, vol. 2119 of *LNCS*, pp. 417–434. Springer, Heidelberg, 2001.

A Concentration of the Squared Norm of Ellipsoidal Gaussians

Fix $\Sigma \in K_{\mathbb{R}}^{m\times m}$ positive definite (in the sense that its components in all embeddings are positive definite). One can define its relative determinant and trace $\det_{K_{\mathbb{R}}} \Sigma$ and $\mathrm{Tr}_{K_{\mathbb{R}}} \Sigma$, which are totally positive, and its absolute determinant and trace $\det \Sigma$ and $\mathrm{Tr}\, \Sigma$ which are positive real numbers. The latter are the images of the former under the norm and trace maps $K_{\mathbb{R}} \to \mathbb{R}$ that extend the usual algebraic norm and trace $N_{K/\mathbb{Q}}, \mathrm{Tr}_{K/\mathbb{Q}}$.

Now the Gaussian weight over $K_{\mathbb{R}}^m$ with parameter Σ, defined by:

$$\rho_{\sqrt{\Sigma}}(\mathbf{x}) = \exp\big(-\pi\|\sqrt{\Sigma}^{-1}\vec{x}\|\big) = \exp\big(-\pi\langle \mathbf{x}, \Sigma^{-1}\mathbf{x}\rangle\big)$$

has at its Fourier transform over $K_{\mathbb{R}}^m$ the function $\widehat{\rho_{\sqrt{\Sigma}}} = \sqrt{\det \Sigma}\rho_{\sqrt{\Sigma}^{-1}}$. From there, we can deduce the following generalization of [Reg09, Claim 3.8] with the same proof.

Lemma A.1. *For any lattice $\Lambda \subset K_{\mathbb{R}}^m$, any $\mathbf{u} \in K_{\mathbb{R}}^m$, and any $\Sigma \in K_{\mathbb{R}}^m$ symmetric definite positive with $\Sigma \succ \eta_\varepsilon(\Lambda)$, we have:*

$$\rho_{\sqrt{\Sigma}}(\Lambda + \mathbf{u}) \in \sqrt{\det \Sigma}\,\mathrm{vol}(\Lambda^*) \cdot [1-\varepsilon, 1+\varepsilon].$$

Dividing two such relations, we can also infer the following corollary.

Corollary A.2. *For any lattice $\Lambda \subset K_{\mathbb{R}}^m$, any $\mathbf{u} \in K_{\mathbb{R}}^m$, and any $\Sigma, \Sigma' \in K_{\mathbb{R}}^m$ symmetric definite positive with $\Sigma, \Sigma' \succ \eta_\varepsilon(\Lambda)$, we have:*

$$\frac{\rho_{\sqrt{\Sigma}}(\Lambda + \mathbf{u})}{\rho_{\sqrt{\Sigma'}}(\Lambda + \mathbf{u})} \in \sqrt{\frac{\det \Sigma}{\det \Sigma'}} \cdot \Big[\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\Big].$$

Using these results, we can establish the following concentration bound, which shows that the squared norm $\|\mathbf{x}\|^2$ of a Gaussian vector $\mathbf{x} \sim \mathscr{D}_{\sqrt{\Sigma},\Lambda+\mathbf{u}}$ over an arbitrary lattice coset is close to $\frac{1}{2\pi}\mathrm{Tr}\,\Sigma$ as long as $\Sigma \succ 2\eta_\varepsilon(\Lambda)$.

Theorem A.3. *Fix a lattice $\Lambda \subset K_{\mathbb{R}}^m$, a vector $\mathbf{u} \in K_{\mathbb{R}}^m$, and a symmetric definite positive $\Sigma \in K_{\mathbb{R}}^m$ with $\Sigma \succ 2\eta_\varepsilon(\Lambda)$, and consider a Gaussian vector $\mathbf{x} \sim \mathscr{D}_{\sqrt{\Sigma},\Lambda+\mathbf{u}}$. Let furthermore:*

$$\nu = \frac{mN}{2\pi^2} s_1(\Sigma)^2 \quad \textit{and} \quad \beta = \frac{s_1(\Sigma)}{\pi}.$$

Then for all $t \geq 0$, we have:

$$\Pr\Big[\|\mathbf{x}\|^2 \geq \frac{\mathrm{Tr}\,\Sigma}{2\pi} + t\Big] \leq \frac{1+\varepsilon}{1-\varepsilon} e^{-\frac{t^2}{2(\nu+\beta t)}},$$

and:

$$\Pr\Big[\|\mathbf{x}\|^2 \leq \frac{\mathrm{Tr}\,\Sigma}{2\pi} - t\Big] \leq \frac{1+\varepsilon}{1-\varepsilon} e^{-\frac{t^2}{2(\nu+\beta t)}}.$$

Proof. The idea is to establish a subexponential concentration bound similar to Bernstein's inequality. To do so, we estimate the following expectation for a parameter λ to be chosen later:

$$\begin{aligned}
\mathbb{E}\big[\exp(\lambda\|\mathbf{x}\|^2)\big] &= \frac{1}{\rho_{\sqrt{\Sigma}}(\Lambda+\mathbf{u})} \sum_{\mathbf{x}\in\Lambda+\mathbf{u}} \exp(\lambda\|\mathbf{x}\|^2)\rho_{\sqrt{\Sigma}}(\mathbf{x}) \\
&= \frac{1}{\rho_{\sqrt{\Sigma}}(\Lambda+\mathbf{u})} \sum_{\mathbf{x}\in\Lambda+\mathbf{u}} \exp\Big(\lambda\|\mathbf{x}\|^2 - \pi\|\sqrt{\Sigma}^{-1}\mathbf{x}\|\Big) \\
&= \frac{1}{\rho_{\sqrt{\Sigma}}(\Lambda+\mathbf{u})} \sum_{\mathbf{x}\in\Lambda+\mathbf{u}} \exp\Big(-\pi\langle \mathbf{x}, (\Sigma^{-1} - \frac{\lambda}{\pi}\mathbb{I}_m)\mathbf{x}\rangle\Big) \\
&= \frac{\rho_{\sqrt{\Sigma_\lambda}}(\Lambda+\mathbf{u})}{\rho_{\sqrt{\Sigma}}(\Lambda+\mathbf{u})},
\end{aligned}$$

where Σ_λ is given by $\Sigma_\lambda^{-1} = \Sigma^{-1} - \frac{\lambda}{\pi}\mathbb{I}_m$. It is well-defined and symmetric definite positive as long as $\Sigma^{-1} \succ \frac{\lambda}{\pi}\mathbb{I}_m$, namely $s_m(\Sigma^{-1}) > \lambda/\pi$, or equivalently $\lambda < 1/\beta$. Moreover, if $\lambda > -1/\beta$, we also have $-\lambda/\pi < 1/s_1(\Sigma)$, and hence:

$$s_m(\Sigma_\lambda) = \frac{1}{s_1(\Sigma_\lambda^{-1})} = \frac{1}{s_1(\Sigma^{-1}) - \lambda/\pi} > \frac{1}{\frac{1}{s_m(\Sigma)} + \frac{1}{s_1(\Sigma)}} \geq \frac{s_m(\Sigma)}{2} \geq \eta_\varepsilon(\Lambda).$$

Thus, for all $|\lambda| < 1/\beta$, Corollary A.2 gives:

$$\exists \delta \in \Big[\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\Big],\ \mathbb{E}\left[\exp(\lambda\|\mathbf{x}\|^2)\right] = \delta\sqrt{\frac{\det \Sigma_\lambda}{\det \Sigma}} = \frac{\delta}{\sqrt{\det(\Sigma_\lambda^{-1}\Sigma)}} = \frac{\delta}{\sqrt{\det(\mathbb{I}_m - \frac{\lambda}{\pi}\Sigma)}}.$$

Using a standard diagonalization argument, we have the following series expansion:

$$\log\det(\mathbb{I}_m - \frac{\lambda}{\pi}\Sigma) = -\sum_{k\geq 1}\frac{\lambda^k}{k\pi^k}\operatorname{Tr}(\Sigma^k).$$

which is absolutely convergent in our domain of interest $|\lambda| < 1/\beta$. Thus:

$$\log\mathbb{E}\left[\exp(\lambda\|\mathbf{x}\|^2)\right] = \log\delta + \frac{1}{2}\sum_{k\geq 1}\frac{\lambda^k}{k\pi^k}\operatorname{Tr}(\Sigma^k) = \frac{\operatorname{Tr}\Sigma}{2\pi} + \log\delta + \frac{1}{2}\sum_{k\geq 2}\frac{\lambda^k}{k\pi^k}\operatorname{Tr}(\Sigma^k).$$

As a result, for all $|\lambda| < 1/\beta$:

$$\begin{aligned}
\log\mathbb{E}\Big[\exp\big(\lambda(\|\mathbf{x}\|^2 - \frac{\operatorname{Tr}\Sigma}{2\pi})\big)\Big] &= \log\delta + \frac{1}{2}\sum_{k\geq 2}\frac{\lambda^k}{k\pi^k}\operatorname{Tr}(\Sigma^k)\\
\log\mathbb{E}\Big[\exp\big(\lambda(\|\mathbf{x}\|^2 - \frac{\operatorname{Tr}\Sigma}{2\pi})\big)\Big] &\leq |\log\delta| + \frac{1}{2}\sum_{k\geq 2}\frac{|\lambda|^k}{k\pi^k}\operatorname{Tr}(\Sigma^k)\\
&\leq |\log\delta| + \frac{1}{2}\sum_{k\geq 2}\frac{|\lambda|^k}{2\pi^k}mNs_1(\Sigma)^k\\
&\leq \log\frac{1+\varepsilon}{1-\varepsilon} + \frac{mN}{4}\sum_{k\geq 2}\big(\beta|\lambda|\big)^k\\
&= \log\frac{1+\varepsilon}{1-\varepsilon} + \frac{mN}{4}\frac{\beta^2\lambda^2}{1-\beta|\lambda|} = \log\frac{1+\varepsilon}{1-\varepsilon} + \frac{\nu\lambda^2/2}{1-\beta|\lambda|}\\
\mathbb{E}\Big[\exp\big(\lambda(\|\mathbf{x}\|^2 - \frac{\operatorname{Tr}\Sigma}{2\pi})\big)\Big] &\leq \frac{1+\varepsilon}{1-\varepsilon}\exp\Big(\frac{\nu\lambda^2/2}{1-\beta|\lambda|}\Big).
\end{aligned}$$

Therefore, we can apply the Markov inequality to get, for all $t \geq 0$ and $\lambda \in [0, 1/\beta)$:

$$\begin{aligned}
\Pr\Big[\|\mathbf{x}\|^2 \geq \frac{\operatorname{Tr}\Sigma}{2\pi} + t\Big] &= \Pr\Big[\exp\big(\lambda(\|\mathbf{x}\|^2 - \frac{\operatorname{Tr}\Sigma}{2\pi})\big) \leq e^{\lambda t}\Big]\\
&\leq e^{-\lambda t}\,\mathbb{E}\Big[\exp\big(\lambda(\|\mathbf{x}\|^2 - \frac{\operatorname{Tr}\Sigma}{2\pi})\big)\Big] \leq \frac{1+\varepsilon}{1-\varepsilon}\exp\Big(\frac{\nu\lambda^2/2}{1-\beta\lambda} - \lambda t\Big).
\end{aligned}$$

Choosing $\lambda = \frac{t}{\beta t+\nu}$ gives the required bound:

$$\Pr\Big[\|\mathbf{x}\|^2 \geq \frac{\operatorname{Tr}\Sigma}{2\pi} + t\Big] \leq \frac{1+\varepsilon}{1-\varepsilon}e^{-\frac{t^2}{2(\nu+\beta t)}}.$$

The probability bound for the other inequality is obtained in the same way by considering $\lambda \in (-1/\beta, 0]$.

Remark A.4. The upper bound on $\|\mathbf{x}\|^2$ is non trivial whenever the covariance matrix is not too skewed. For example, taking $t = \frac{\operatorname{Tr}\Sigma}{4\pi} \geq \frac{mNs_m(\Sigma)}{4\pi}$, we get:

$$\Pr\Big[\|\mathbf{x}\|^2 \leq \frac{\operatorname{Tr}\Sigma}{4\pi}\Big] \leq 2^{-\Omega\left(mN\frac{s_m(\Sigma)^2}{s_1(\Sigma)^2}\right)}. \tag{14}$$

For very skewed Σ, we can at least use the fact that $\Pr_{\mathbf{y}\sim\mathscr{D}_{\sqrt{\Sigma_0},\Lambda+\mathbf{u}}}[\|\mathbf{y}\|^2 \leq B]$ increases when Σ_0 decreases to obtain:

$$\Pr\Big[\|\mathbf{x}\|^2 \leq \frac{mNs_m(\Sigma)}{4\pi}\Big] \leq \Pr_{\mathbf{y}\sim\mathscr{D}_{s_m(\sqrt{\Sigma}),\Lambda+\mathbf{u}}}\Big[\|\mathbf{y}\|^2 \leq \frac{mNs_m(\Sigma)}{4\pi}\Big] \leq 2^{-\Omega(mN)},$$

where the second equality is Eq. (14) applied to the diagonal covariance matrix $s_m(\Sigma)\mathbb{I}_m$.

B Rejection Sampling for Ellipsoidal Gaussians

In this section we prove a more general result than what is needed for MuSig-L. Indeed, we show that rejection sampling can be used to hide all parameters of a ellipsoidal Gaussian, as long as they are not too far off from the public ones. Similarly to the proof of the original rejection sampling result [Lyu12, Theorem 4.6], we split the proof into two steps. First, we prove a rejection sampling lemma for general probability distributions under some assumptions (Lemma B.2), and then we show that a specific choice of ellipsoidal Gaussians defined over a coset satisfies such hypotheses (Lemma B.4). In Section 3.2 we show that this result can be applied to the specific spherical distributions used in the signing algorithm of MuSig-L. For completeness, in Appendix B.4 we apply our generalized rejection sampling lemma to simulate rejected transcripts of the standard Fiat-Shamir with aborts protocol. This implies statistical honest verifier zero knowledge of [Lyu12]-type interactive protocols (as opposed to the *no-abort* HVZK often assumed in the literature).

Theorem B.1 (Rejection Sampling for Ellipsoidal Gaussians). *Let $\Lambda \in R^{\ell+k}$ be a lattice. Let $\alpha, T, m > 0$, $\varepsilon \leq 1/2$. Let $\widehat{\Sigma} \in K_{\mathbb{R}}^{m\times m}$ be a positive definite matrix such that $\widehat{\Sigma} \succ \max\{\eta_\varepsilon(\Lambda), \alpha T\}$.*

Consider a set $V \subseteq K_{\mathbb{R}}^{m\times m} \times R^k \times R^{\ell+k}$. Let $h : V \to [0,1]$ be the composition of three probability distributions $h := \mathscr{D}_s \times \mathscr{D}_u \times \mathscr{D}_v$, where $\mathscr{D}_u$ returns a vector $\mathbf{u} \in R^k$, $\mathscr{D}_v$ returns a vector $\mathbf{v} \in R^{\ell+k}$ such that $\|\mathbf{v}\| \leq T$, and $\mathscr{D}_s$ returns a positive definite matrix $\Delta_s \in K_{\mathbb{R}}^{m\times m}$ such that:

$$\Delta_s \prec \frac{1}{m\beta}\widehat{\Sigma}$$

for some $\beta \in \left[\frac{1}{2}, \frac{1}{2} + \frac{1}{m}\right]$. Set $\Sigma := \widehat{\Sigma} + \Delta_s$. Then there exists $M > 0$, and negligible $0 < \epsilon, \delta_1, \delta_2 < 1$ such that the distribution of the following algorithm

RejSamp*:*

- $(\Sigma, \mathbf{u}, \mathbf{v}) \leftarrow h$
- $\mathbf{z} \leftarrow \mathscr{D}^{\ell+k}_{\sqrt{\Sigma},\mathbf{v},\Lambda^\perp_{\mathbf{u}}}$
- *with probability* $1 - \min\left(1, \frac{\mathscr{D}^{\ell+k}_{\sqrt{\widehat{\Sigma}}}(\mathbf{z})}{M\cdot\mathscr{D}^{\ell+k}_{\sqrt{\Sigma},\mathbf{v}}(\mathbf{z})}\right)$*, set* $\mathbf{z} := \perp$
- *output* $(\mathbf{z}, \Sigma, \mathbf{u}, \mathbf{v})$

is within statistical distance $\frac{\epsilon}{2M} + \frac{\delta_1}{M}$ of the distribution of:

SimRS*:*

- $(\Sigma, \mathbf{u}, \mathbf{v}) \leftarrow h$
- $\mathbf{z} \leftarrow \mathscr{D}^{\ell+k}_{\sqrt{\widehat{\Sigma}},\Lambda^\perp_{\mathbf{u}}}$
- *with probability* $1 - 1/M$*, set* $\mathbf{z} := \perp$
- *output* $(\mathbf{z}, \Sigma, \mathbf{u}, \mathbf{v})$

Moreover, the probability that RejSamp *outputs something is at least $\frac{1-\epsilon}{M}(1-\delta_2)$.*

The rest of the section is devoted to proving Theorem B.1.

B.1 Generalized Rejection Sampling

The following expands [Lyu12, Lemma 4.7] to restrict the original rejection sampling technique to a subset of the domain, as long as its mass does not change significantly.

Lemma B.2. *Let V be an arbitrary set, $\mathcal{P}(\mathbb{Z}^m)$ be the set of subsets of $\mathbb{Z}^m$, $P \subseteq \mathcal{P}(\mathbb{Z}^m)$, and $h : V \times P \to \mathbb{R}$ and $f : \mathbb{Z}^m \to \mathbb{R}$ be probability distributions. Let $g_v : \mathbb{Z}_m \to \mathbb{R}$ be a family of probability distributions indexed by all $v \in V$ with the property that $\exists\ M \in \mathbb{R}$ such that $\forall\ (v, S) \in V \times P$,*

$$\Pr\left[Mg_v(z) \geq f(z)\ :\ z \leftarrow f^S\right] \geq 1 - \epsilon$$

where f^S is the restriction to S of f, i.e., $f^S(z) = \frac{f(z)}{f(S)}$ if $z \in S$, and $f^S(z) = 0$ otherwise (g_v^S is defined analogously). Assume that

$$\forall\ z \in \mathbb{Z}^m,\ 1 - \nu_L \leq \frac{f(S)}{g_v(S)} \leq 1 + \nu_U$$

for some $0 \leq \nu_L < 1$ and $\nu_U \geq 0$. Then the distribution of the output of the following algorithm

RejSamp*:*

- $(v, S) \leftarrow h$
- $z \leftarrow g_v^S$
- *with probability* $1 - \min\left(1, \frac{f(z)}{M \cdot g_v(z)}\right)$*, set* $z := \perp$
- *output* (z, v, S).

is within statistical distance $\frac{\epsilon}{2M} + \frac{\max\{\nu_L, \nu_U\}}{2M}$ *of the distribution of the following algorithm:*

SimRS*:*

- $(v, S) \leftarrow h$
- $z \leftarrow f^S$
- *with probability* $1 - 1/M$*, set* $z := \perp$
- *output* (z, v, S).

Moreover, the probability that RejSamp *outputs something is at least* $(1 - \nu_L)\frac{1-\epsilon}{M}$.

Proof. The proof follows the same structure of the proof of [Lyu12, Lemma 4.7]. Fixed $v \in V$, $S \in \mathcal{P}(\mathbb{Z}^m)$, define a set $S_v^S := \{z \in S \ : \ Mg_v(z) \geq f(z)\}$. One can then bound the measure of such set and of $S \setminus S_v^S$ as: $f^S(S_v^S) = \sum_{z \in S_v^S} f(z)/f(S) \geq 1 - \epsilon$ and $f^S(S \setminus S_v^S) = \sum_{z \in S \setminus S_v^S} f(z)/f(S) \leq \epsilon$. From the definition of S_v^S follows also that

$$\forall\ z \in S_v^S,\ \Pr_r[z \leftarrow \mathsf{RejSamp}(r)] = g_v^S(z)\frac{f(z)}{M \cdot g_v(z)} = \frac{f(z)}{M \cdot g_v(S)}$$

$$\forall\ z \notin S_v^S,\ \Pr_r[z \leftarrow \mathsf{RejSamp}(r)] = g_v^S(z) = \begin{cases} \frac{g_v(z)}{g_v(S)} & \text{if } z \in S \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the probability that the rejection sampling algorithm returns something can be bounded from above by

$$\begin{aligned}
&\Pr_r[z \leftarrow \mathsf{RejSamp}(r)] \\
&= \sum_{(v,S)} h(v,S) \left(\sum_{z \in S_v^S} \Pr_r[z \leftarrow \mathsf{RejSamp}(r)] + \sum_{z \notin S_v^S} \Pr_r[z \leftarrow \mathsf{RejSamp}(r)] \right) \\
&= \sum_{(v,S)} h(v,S) \left(\sum_{z \in S_v^S} \frac{f(z)}{M \cdot g_v(S)} + \sum_{z \in S \setminus S_v^S} g_v^S(z) \right) \\
&= \sum_{(v,S)} \frac{h(v,S)}{g_v(S)} \left(\sum_{z \in S_v^S} \frac{f(z)}{M} + \sum_{z \in S \setminus S_v^S} g_v(z) \right) \\
&\leq \sum_{(v,S)} \frac{h(v,S)}{g_v(S)} \left(\sum_{z \in S_v^S} \frac{f(z)}{M} + \sum_{z \in S \setminus S_v^S} \frac{f(z)}{M} \right) \\
&\leq \frac{1}{M} \sum_{(v,S)} h(v,S) \frac{f(S)}{g_v(S)} \leq \frac{1 + \nu_U}{M}
\end{aligned}$$

and from below by

$$\begin{aligned}
&\Pr_r[z \leftarrow \mathsf{RejSamp}(r)] \\
&= \sum_{(v,S)} h(v,S) \left(\sum_{z \in S_v^S} \Pr_r[z \leftarrow \mathsf{RejSamp}(r)] + \sum_{z \notin S_v^S} \Pr_r[z \leftarrow \mathsf{RejSamp}(r)] \right) \\
&= \sum_{(v,S)} h(v,S) \left(\sum_{z \in S_v^S} \frac{f(z)}{M \cdot g_v(S)} + \sum_{z \in S \setminus S_v^S} \frac{g_v(z)}{g_v(S)} \right) \\
&\geq \sum_{(v,S)} \frac{h(v,S)}{M} \frac{f(S)}{g_v(S)} f^S(S_v^S) \geq \frac{(1-\epsilon)(1-\nu_L)}{M} \ .
\end{aligned}$$

The previous computation allows to already estimate the probability that the algorithms abort for a fixed pair (v, S), i.e., that they return $(\perp, v, S)$. Let $N_{RS} = \Pr[\mathsf{RejSamp} \text{ aborts}]$, and $N_S = \Pr[\mathsf{SimRS} \text{ aborts}]$. It follows that $1 - (1-\epsilon)(1-\nu_L)/M \geq N_{RS} \geq 1 - (1+\nu_U)/M$ and $N_S = 1 - 1/M$. Equipped with these bounds we can finally bound the statistical distance between RejSamp and SimRS:

$$
\begin{aligned}
&\Delta(\mathsf{RejSamp}, \mathsf{SimRS}) \\
&= \frac{1}{2}\left(\sum_{z,v,S}\left|h(v,S)g_v^S(z)\min\left\{1, \frac{f(z)}{Mg_v(z)}\right\} - h(v,S)\frac{f^S(z)}{M}\right| + |N_{RS} - N_S|\right) \\
&= \frac{1}{2}\sum_{v,S}h(v,S)\left(\sum_{z}\left|g_v^S(z)\min\left\{1, \frac{f(z)}{M\cdot g_v(z)}\right\} - \frac{f^S(z)}{M}\right| + |N_{RS} - N_S|\right) \\
&= \frac{1}{2}\sum_{v,S}h(v,S)\left(\sum_{z\in S_v^S}\left|\frac{f(z)}{M\cdot g_v(S)} - \frac{f^S(z)}{M}\right| + \sum_{z\in S\setminus S_v^S}\left|g_v^S(z) - \frac{f^S(z)}{M}\right| + |N_{RS} - N_S|\right) \\
&= \frac{1}{2}\sum_{v,S}h(v,S)(\underbrace{\sum_{z\in S_v^S}\frac{f(z)}{M}\left|\frac{1}{g_v(S)} - \frac{1}{f(S)}\right| + \sum_{z\in S\setminus S_v^S}\left|\frac{g_v(z)}{g_v(S)} - \frac{f(z)}{M\cdot f(S)}\right|}_{=:K_1} + \underbrace{|N_{RS} - N_S|}_{=:K_2}) \ . \qquad (15)
\end{aligned}
$$

We now analyze K_1, K_2 separately:

$$
\begin{aligned}
K_1 &= \frac{f(S_v^S)}{M}\left|\frac{1}{g_v(S)} - \frac{1}{f(S)}\right| + \sum_{z\in S\setminus S_v^S}\frac{f(z)}{M}\left|\frac{Mg_v(z)}{f(z)g_v(S)} - \frac{1}{f(S)}\right| \\
&\leq \frac{f(S_v^S)}{M}\left|\frac{1}{g_v(S)} - \frac{1}{f(S)}\right| + \sum_{z\in S\setminus S_v^S}\frac{f(z)}{M}\left|\frac{1}{g_v(S)} - \frac{1}{f(S)}\right| \\
&= \frac{f(S)}{M}\left|\frac{1}{g_v(S)} - \frac{1}{f(S)}\right| = \frac{1}{M}\left|\frac{f(S)}{g_v(S)} - 1\right| \leq \frac{1}{M}\max\{\nu_L, \nu_U\} \ ,
\end{aligned}
$$

where the first inequality follows observing that $g_v(z) \leq f(z)/M$ for all $z \in S \setminus s_v^S$. Finally,

$$
\begin{aligned}
K_3 &\leq \max\left\{\left|1 - \frac{(1-\epsilon)(1-\nu_L)}{M} - \left(1 - \frac{1}{M}\right)\right|, \left|1 - \frac{(1+\nu_U)}{M} - \left(1 - \frac{1}{M}\right)\right|\right\} \\
&= \frac{1}{M}\max\left\{\epsilon + \nu_L(1-\epsilon), \nu_U\right\} \leq \frac{\epsilon + \max\{\nu_U, \nu_L\}}{M} \ .
\end{aligned}
$$

Plugging these inequalities in Eq. (15) yields

$$
\Delta(\mathsf{RejSamp}, \mathsf{SimRS}) \leq \frac{\epsilon}{2M} + \frac{\max\{\nu_L, \nu_U\}}{2M} \ .
$$

□

B.2 Technical Lemma

Before proving the main statement we need to prove a variant of Corollary A.2, namely we show that for two parameters $\sqrt{\widehat{\Sigma}} \prec \sqrt{\Sigma}$ the Gaussian mass of a lattice changes proportionally to the ratio of their determinants.

Lemma B.3. *Let $\Lambda \subseteq R^m$ be a lattice, and $\Sigma,\ \widehat{\Sigma} \in K_{\mathbb{R}}^{m\times m}$ be symmetric positive definite matrices such that $\sqrt{\widehat{\Sigma}} \prec \sqrt{\Sigma}$. Then*

$$
\frac{\rho_{\sqrt{\Sigma}}(\Lambda)}{\rho_{\sqrt{\widehat{\Sigma}}}(\Lambda)} \leq \sqrt{\frac{\det(\Sigma)}{\det(\widehat{\Sigma})}} \ .
$$

Proof. By the properties of the discrete Fourier transform we have that

$$\begin{aligned}\rho_{\sqrt{\Sigma}}(\Lambda) &= \rho(\sqrt{\Sigma}^{-1}\Lambda) = \det((\sqrt{\Sigma}^{-1}\Lambda)^*)\hat{\rho}\left((\sqrt{\Sigma}^{-1}\Lambda)^*\right) \\ &= \det(\sqrt{\Sigma}\Lambda^*)\rho_{\sqrt{\Sigma}^{-1}}(\Lambda^*) = \det(\sqrt{\Sigma}\Lambda^*)\cdot\left(\sum_{\mathbf{x}\in\Lambda^*}\exp\left(-\pi\langle\mathbf{z},\Sigma^{-1}\mathbf{z}\rangle\right)\right) \\ &\leq \det(\sqrt{\Sigma}\Lambda^*)\cdot\left(\sum_{\mathbf{x}\in\Lambda^*}\exp\left(-\pi\langle\mathbf{z},\widehat{\Sigma}^{-1}\mathbf{z}\rangle\right)\right) \leq \det(\sqrt{\Sigma}\Lambda^*)\rho_{\sqrt{\widehat{\Sigma}}^{-1}}(\Lambda^*)\end{aligned}$$

where the third equality follows from $(\sqrt{\Sigma}^{-1}\Lambda)^* = \sqrt{\Sigma}\Lambda^*$, and the first inequality $\widehat{\Sigma} \prec \Sigma$. Observing that $\rho_{\sqrt{\widehat{\Sigma}}}(\Lambda) = \det(\sqrt{\widehat{\Sigma}}\Lambda^*)\cdot\rho_{\sqrt{\widehat{\Sigma}}^{-1}}(\Lambda^*)$ yields the final bound

$$\frac{\rho_{\sqrt{\Sigma}}(\Lambda)}{\rho_{\sqrt{\widehat{\Sigma}}}(\Lambda)} \leq \frac{\det((\sqrt{\Sigma}\Lambda)^*)\rho_{\sqrt{\widehat{\Sigma}}^{-1}}(\Lambda^*)}{\det(\sqrt{\widehat{\Sigma}}\Lambda^*)\rho_{\sqrt{\widehat{\Sigma}}^{-1}}(\Lambda^*)} = \sqrt{\frac{\det(\Sigma)}{\det(\widehat{\Sigma})}}\ .$$

□

The following shows that if the parameters are not too far off two ellipsoidal discrete Gaussian distributions behave similarly when restricted to a lattice coset. The proof follow somewhat closely the original.

Lemma B.4. *Let $m > 0$, $\Lambda \subseteq R^m$ be a lattice, and let $\mathbf{u} \in R^m \setminus \{\mathbf{0}\}$. For a set $V \subseteq R^m$ let $T = \max_{\mathbf{v}\in V}\|\mathbf{v}\|$. Let $\widehat{\Sigma}, \Sigma \in K_{\mathbb{R}}^{m\times m}$ be two positive definite matrices such that:*

- *$\widehat{\Sigma} \succ \max\{\eta_\varepsilon(\Lambda), \alpha T\}$ for some $\alpha > 0$,*
- *$\frac{m\beta}{m\beta+1}\widehat{\Sigma}^{-1} \prec \Sigma^{-1} \prec \frac{1}{\beta}\widehat{\Sigma}^{-1}$, for some[9] $\beta \in \left[\frac{1}{2}, \frac{1}{2}+\frac{1}{m}\right]$.*

Then, for any $\mathbf{v} \in V$ it holds

$$\Pr\left[\frac{\mathscr{D}^m_{\sqrt{\widehat{\Sigma}}}(\mathbf{z})}{\mathscr{D}^m_{\sqrt{\Sigma},\mathbf{v}}(\mathbf{z})} \leq M \ :\ \mathbf{z} \leftarrow \mathscr{D}^m_{\sqrt{\widehat{\Sigma}},\Lambda+\mathbf{u}}\right] \geq 1-\epsilon\ ,$$

where $M = \exp\left(\frac{\pi}{\alpha^2}+\frac{\pi t}{\alpha}\right)$ and $\epsilon = 2\frac{1+\varepsilon}{1-\varepsilon}\exp\left(-t^2(\pi-1)\right)$ for every $t > 0$.

Proof. The ration between Gaussians can be split into three parts that we treat separately in Lemma B.5 and Lemma B.7:

$$\begin{aligned}\frac{\mathscr{D}^m_{\sqrt{\widehat{\Sigma}}}(\mathbf{z})}{\mathscr{D}^m_{\sqrt{\Sigma},\mathbf{v}}(\mathbf{z})} &= \frac{\mathscr{D}^m_{\sqrt{\widehat{\Sigma}}}(\mathbf{z})}{\mathscr{D}^m_{\sqrt{\Sigma}}(\mathbf{z})}\cdot\frac{\mathscr{D}^m_{\sqrt{\Sigma}}(\mathbf{z})}{\mathscr{D}^m_{\sqrt{\Sigma},\mathbf{v}}(\mathbf{z})} \\ &= \underbrace{\exp\left(-\pi(\|\sqrt{\widehat{\Sigma}}^{-1}\mathbf{z}\|^2 - \|\sqrt{\Sigma}^{-1}\mathbf{z}\|^2)\right)}_{=:K_1}\cdot\underbrace{\frac{\rho_{\sqrt{\Sigma}}(R^m)}{\rho_{\sqrt{\widehat{\Sigma}}}(R^m)}}_{=:K_2}\cdot\underbrace{\frac{\mathscr{D}^m_{\sqrt{\Sigma}}(\mathbf{z})}{\mathscr{D}^m_{\sqrt{\Sigma},\mathbf{v}}(\mathbf{z})}}_{=:K_3}\ .\end{aligned}$$

Lemma B.5. $\Pr\left[K_1\cdot K_2 \leq 1 \ :\ \mathbf{z} \leftarrow \mathscr{D}^m_{\sqrt{\widehat{\Sigma}},\Lambda^\perp_{\mathbf{u}}}\right] \geq 1-\mathsf{negl}(\lambda)$.

Proof. Let $\Sigma_d^{-1} := \widehat{\Sigma}^{-1} - \Sigma^{-1}$; Σ_d^{-1} is positive definite,

$$\Sigma_d^{-1} \succ \widehat{\Sigma}^{-1} - \Sigma^{-1} \succ \widehat{\Sigma}^{-1} - \frac{1}{\beta}\widehat{\Sigma}^{-1} = \frac{\beta-1}{\beta}\widehat{\Sigma}^{-1} \succ 0\ .$$

Combining the previous observation with Theorem A.3 yields

$$K_1 = \exp\left(-\pi\|\sqrt{\Sigma_d}^{-1}\mathbf{z}\|^2\right) \leq \exp\left(-\frac{\beta-1}{\beta}\|\widehat{\Sigma}^{-1}\mathbf{z}\|^2\right) \leq \exp\left(-\frac{\beta-1}{\beta}\right)\ . \tag{16}$$

[9] The condition $\beta \leq \frac{1}{2}+\frac{1}{m}$ excludes the cases in which the inequality is trivially false, i.e., when $\beta < \frac{m\beta}{m\beta+1}$. A tighter value of such upper bound is $\beta \leq \frac{\sqrt{4m^2+1}+1}{2m}$.

Finally, Lemma B.3 yields that

$$K_2 \le \sqrt{\frac{\det(\Sigma)}{\det(\widehat{\Sigma})}} \le \left(1+\frac{1}{m\beta}\right)^{m/2} \le \exp\left(\frac{1}{2\beta}\right) . \tag{17}$$

Putting together Eq. (16) and Eq. (17) we obtain that $K_1 \cdot K_2 = \exp\left(\frac{1-2\beta}{2\beta}\right) \le 1$ with probability $1-\epsilon$ as $\beta \ge 2$. □

Before bounding K_3 we need to prove a technical lemma.

Lemma B.6 (generalized [Lyu12, Lemma 4.3]). *Under the assumptions of Lemma B.2, for any $r > 0$ it holds*

$$\Pr\left[\pi\left|\langle \mathbf{z}, \Sigma^{-1}\mathbf{c}\rangle + \langle \mathbf{c}, \Sigma^{-1}\mathbf{z}\rangle\right| > \frac{r}{s_m(\Sigma)} \;:\; \mathbf{z} \leftarrow \mathscr{D}^m_{\sqrt{\widehat{\Sigma}},\Lambda+\mathbf{u}}\right] \le 2\frac{1+\varepsilon}{1-\varepsilon}\exp\left(-\frac{(\pi-1)r^2}{\|\mathbf{c}\|^2 s_m(\Sigma)}\right) .$$

Proof. For any $t \in \mathbb{R}$ it holds

$$\begin{aligned}
&\mathrm{E}\left[\exp\left(\pi t\cdot(\langle \mathbf{z}, \Sigma^{-1}\mathbf{c}\rangle + \langle \mathbf{c}, \Sigma^{-1}\mathbf{z}\rangle)\right)\right] \\
&= \frac{1}{\rho_{\sqrt{\widehat{\Sigma}}}(\Lambda+\mathbf{u})}\sum_{\mathbf{z}\in\Lambda+\mathbf{u}}\exp\left(-\pi(\|\sqrt{\widehat{\Sigma}}^{-1}\mathbf{z}\|^2 - \langle \mathbf{z}, \Sigma^{-1}\mathbf{c}\rangle - \langle \mathbf{c}, \Sigma^{-1}\mathbf{z}\rangle)\right) \\
&= \frac{\exp\left(\pi\|\sqrt{\Sigma}^{-1}(t\mathbf{c})\|^2\right)}{\rho_{\sqrt{\widehat{\Sigma}}}(\Lambda+\mathbf{u})}\sum_{\mathbf{z}\in\Lambda+\mathbf{u}}\exp\left(-\pi(\|\sqrt{\Sigma}^{-1}(\mathbf{z}-t\mathbf{c})\|^2 + \|\sqrt{\Sigma_d}^{-1}\mathbf{z}\|^2)\right) \\
&\le \frac{\rho_{\sqrt{\Sigma},t\mathbf{c}}(\Lambda+\mathbf{u})}{\rho_{\sqrt{\widehat{\Sigma}}}(\Lambda+\mathbf{u})}\cdot\exp\left(\pi\|\sqrt{\Sigma}^{-1}(t\mathbf{c})\|^2\right)\exp\left(-\frac{\beta-1}{\beta}\right) && \text{from Eq. (16)} \\
&\le \frac{\rho_{\sqrt{\Sigma}}(\Lambda)}{\rho_{\sqrt{\widehat{\Sigma}}}(\Lambda)}\cdot\frac{1+\epsilon}{1-\epsilon}\exp\left(\pi\|\sqrt{\Sigma}^{-1}(t\mathbf{c})\|^2\right)\exp\left(-\frac{\beta-1}{\beta}\right) && \text{from Corollary A.2} \\
&\le \sqrt{\frac{\det(\Sigma)}{\det(\widehat{\Sigma})}}\cdot\frac{1+\epsilon}{1-\epsilon}\exp\left(\pi t^2\|\mathbf{c}\|^2(\mathrm{s}_1(\sqrt{\Sigma}^{-1}))^2\right)\exp\left(-\frac{\beta-1}{\beta}\right) && \text{from Lemma B.3} \\
&\le 1\cdot\frac{1+\epsilon}{1-\epsilon}\exp\left(\frac{\pi t^2\|\mathbf{c}\|^2}{\mathrm{s}_m(\Sigma)}\right) && \text{from Eq. (17) and } \beta\ge 2 \\
&= \frac{1+\epsilon}{1-\epsilon}\exp\left(\frac{\pi t^2\|\mathbf{c}\|^2}{\mathrm{s}_m(\Sigma)}\right) && \text{from Lemma 2.3}
\end{aligned}$$

Thus, applying Markov's inequality yields

$$\begin{aligned}
&\Pr\left[\pi(\langle \mathbf{z}, \Sigma^{-1}\mathbf{c}\rangle + \langle \mathbf{c}, \Sigma^{-1}\mathbf{z}\rangle) > \frac{r}{\mathrm{s}_m(\Sigma)} \;:\; \mathbf{z} \leftarrow \mathscr{D}^m_{\sqrt{\widehat{\Sigma}},\Lambda+\mathbf{u}}\right] \\
&= \Pr\left[\exp\left(\pi t\cdot(\langle \mathbf{z}, \Sigma^{-1}\mathbf{c}\rangle + \langle \mathbf{c}, \Sigma^{-1}\mathbf{z}\rangle)\right) > \exp\left(\frac{tr}{\mathrm{s}_m(\Sigma)}\right) \;:\; \mathbf{z} \leftarrow \mathscr{D}^m_{\sqrt{\widehat{\Sigma}},\Lambda+\mathbf{u}}\right] \\
&\le \frac{\mathrm{E}\left[\exp\left(\pi t\cdot(\langle \mathbf{z}, \Sigma^{-1}\mathbf{c}\rangle + \langle \mathbf{c}, \Sigma^{-1}\mathbf{z}\rangle)\right)\right]}{\exp\left(\frac{tr}{\mathrm{s}_m(\Sigma)}\right)} \\
&\le \frac{1+\varepsilon}{1-\varepsilon}\exp\left(\frac{\pi t^2\|\mathbf{c}\|^2}{\mathrm{s}_m(\Sigma)} - \frac{tr}{\mathrm{s}_m(\Sigma)}\right) \\
&= \frac{1+\varepsilon}{1-\varepsilon}\exp\left(-\frac{(\pi-1)r^2}{\|\mathbf{c}\|^2\mathrm{s}_m(\Sigma)}\right)
\end{aligned}$$

where the last equality follows setting $t := \frac{r}{\|\mathbf{c}\|^2}$. The symmetry of the distribution of $\mathbf{z}$ implies

$$\Pr\left[\pi(\langle \mathbf{z}, \Sigma^{-1}\mathbf{c}\rangle + \langle \mathbf{c}, \Sigma^{-1}\mathbf{z}\rangle) < -\frac{r}{\mathrm{s}_m(\Sigma)} \;:\; \mathbf{z} \leftarrow \mathscr{D}^m_{\sqrt{\widehat{\Sigma}},\Lambda+\mathbf{u}}\right] \le \frac{1+\varepsilon}{1-\varepsilon}\exp\left(-\frac{(\pi-1)r^2}{\|\mathbf{c}\|^2\mathrm{s}_m(\Sigma)}\right) .$$

Applying the union bound yields the thesis. □

Lemma B.7. $\Pr\left[K_3 \le \exp\left(\frac{\pi}{\alpha^2} + \frac{\pi t}{\alpha}\right) : \mathbf{z} \leftarrow \mathscr{D}^m_{\sqrt{\widehat{\Sigma}},\Lambda+\mathbf{u}}\right] \ge 1 - 2\frac{1+\varepsilon}{1-\varepsilon}\exp\left(-t^2(\pi-1)\right)$.

Proof. Observe that $K_3 = \frac{\rho_{\sqrt{\Sigma}}(\mathbf{z})}{\rho_{\sqrt{\Sigma},\mathbf{v}}(\mathbf{z})} = \exp\left(\pi(\|\sqrt{\Sigma}^{-1}\mathbf{v}\|^2 - D)\right)$, where $D := \langle \mathbf{z}, \Sigma^{-1}\mathbf{c}\rangle + \langle \mathbf{c}, \Sigma^{-1}\mathbf{z}\rangle$. For every $t > 0$, substituting $r = t \cdot \mathrm{s}_m(\sqrt{\Sigma})\|\mathbf{v}\|$ in Lemma B.6 yields that $|D| \le \frac{t\|\mathbf{v}\|}{\mathrm{s}_m(\sqrt{\Sigma})}$ with probability greater than $1 - 2\frac{1+\varepsilon}{1-\varepsilon}\exp\left(-t^2(\pi-1)\right)$ for all $\mathbf{v} \in R^m$. From the definition of the singular value we have also that $\|\sqrt{\Sigma}^{-1}\mathbf{v}\| \le \mathrm{s}_1(\sqrt{\Sigma}^{-1})\|\mathbf{v}\| = \frac{1}{\mathrm{s}_m(\sqrt{\Sigma})}\|\mathbf{v}\|$. Hence,

$$\begin{aligned} K_3 &\le \exp\left(\frac{\pi\|\mathbf{v}\|^2}{(\mathrm{s}_m(\sqrt{\Sigma}))^2} + \frac{\pi t\|\mathbf{v}\|}{\mathrm{s}_m(\sqrt{\Sigma})}\right) \le \exp\left(\frac{\pi\|\mathbf{v}\|^2}{(\mathrm{s}_m(\sqrt{\widehat{\Sigma}}))^2} + \frac{\pi t\|\mathbf{v}\|}{\mathrm{s}_m(\sqrt{\widehat{\Sigma}})}\right) \\ &= \exp\left(\pi\frac{\|\mathbf{v}\|^2 + t\|\mathbf{v}\|\alpha T}{\alpha^2 T^2}\right) \le \exp\left(\frac{\pi}{\alpha^2} + \frac{\pi t}{\alpha}\right) , \end{aligned}$$

holds with probability greater than $1 - 2\frac{1+\varepsilon}{1-\varepsilon}\exp\left(-t^2(\pi-1)\right)$. □

B.3 Proof for Theorem B.1

We are finally ready to prove our main result.

Proof. As already mentioned, it is enough to check that the parameters satisfy the assumptions of Lemma B.4, and that the Gaussian mass of a lattice coset does not significantly change when switching from the private to the public parameters. If both conditions hold, Lemma B.2 yields the thesis.

The hypothesis $\Delta_s \prec \frac{1}{m\beta}\widehat{\Sigma}$ yields

$$\Sigma \prec \widehat{\Sigma}\left(1 + \frac{1}{m\beta}\right) \quad \Rightarrow \quad \Sigma^{-1} \succ \widehat{\Sigma}^{-1}\left(1 + \frac{1}{m\beta}\right)^{-1} = \widehat{\Sigma}^{-1}\frac{m\beta}{m\beta+1}$$

thus the conditions of Lemma B.4 are satisfied.

The bounds on the ratio of the Gaussian masses of a lattice coset follows from these inequalities obtained applying [GPV08, Lemma 2.7] and Lemma B.3:

$$\begin{aligned} \frac{\rho_{\sqrt{\widehat{\Sigma}}}(\Lambda^\perp_{\mathbf{u}})}{\rho_{\sqrt{\Sigma},\mathbf{v}}(\Lambda^\perp_{\mathbf{u}})} &\le \frac{\rho_{\sqrt{\widehat{\Sigma}}}(\Lambda^\perp)}{\rho_{\sqrt{\Sigma}}(\Lambda^\perp)} \cdot \frac{1+\varepsilon}{1-\varepsilon} \le \frac{1+\varepsilon}{1-\varepsilon}\sqrt{\frac{\det(\widehat{\Sigma})}{\det(\Sigma)}} \\ \frac{\rho_{\sqrt{\Sigma},\mathbf{v}}(R^{\ell+k})}{\rho_{\sqrt{\widehat{\Sigma}}}(R^{\ell+k})} &\le \frac{\rho_{\sqrt{\Sigma}}(R^{\ell+k})}{\rho_{\sqrt{\widehat{\Sigma}}}(R^{\ell+k})} \le \sqrt{\frac{\det(\widehat{\Sigma})}{\det(\Sigma)}} \\ \frac{\rho_{\sqrt{\widehat{\Sigma}}}(\Lambda^\perp_{\mathbf{u}})}{\rho_{\sqrt{\Sigma},\mathbf{v}}(\Lambda^\perp_{\mathbf{u}})} &\ge \frac{\rho_{\sqrt{\widehat{\Sigma}}}(\Lambda^\perp)}{\rho_{\sqrt{\Sigma}}(\Lambda^\perp)} \cdot \frac{1-\varepsilon}{1+\varepsilon} \ge \frac{1-\varepsilon}{1+\varepsilon} \\ \frac{\rho_{\sqrt{\Sigma},\mathbf{v}}(R^{\ell+k})}{\rho_{\sqrt{\widehat{\Sigma}}}(R^{\ell+k})} &\ge \frac{1-\varepsilon}{1+\varepsilon} \cdot \frac{\rho_{\sqrt{\Sigma}}(R^{\ell+k})}{\rho_{\sqrt{\widehat{\Sigma}}}(R^{\ell+k})} \ge \frac{1-\varepsilon}{1+\varepsilon} \end{aligned}$$

where the last lower bound follows observing that $\forall\ z \in \Lambda^\perp$,

$$\begin{aligned} \rho_{\sqrt{\Sigma}}(z) &\ge \exp\left(-\frac{1}{2}\mathrm{s}_1(\Sigma^{-1})\|z\|^2\right) = \exp\left(-\frac{1}{2}\frac{\|z\|^2}{\mathrm{s}_m(\Sigma)}\right) \ge \exp\left(-\frac{1}{2}\frac{\|z\|^2}{\mathrm{s}_1(\widehat{\Sigma})}\right) \\ &\ge \exp\left(-\frac{1}{2}\mathrm{s}_m(\widehat{\Sigma}^{-1})\|z\|^2\right) \ge \exp\left(-\frac{1}{2}\|\sqrt{\widehat{\Sigma}}^{-1}z\|^2\right) = \rho_{\sqrt{\widehat{\Sigma}}}(z) \ . \end{aligned}$$

Thus, we can set $\nu_L := \frac{4\varepsilon}{(1+\varepsilon)^2}$ and $\nu_U := \frac{2\varepsilon}{1-\varepsilon}$ and substitute them in the formulas from Lemma B.2 to obtain[10] $\delta_1 := 2\varepsilon \ge \varepsilon/(1-\varepsilon)$ (where the inequality follows from $\varepsilon < 1/2$) and $\delta_2 := \frac{4\varepsilon}{(1+\varepsilon)^2}$. □

[10] Remark that $\varepsilon \ne \epsilon$, as the factor ε comes from the smoothing parameter of the lattice Λ, while ϵ comes from the condition in Lemma B.4.

Algorithm 4: Simulation of an honest Fiat-Shamir aborts transcript

$\mathcal{T}(\bar{\mathbf{A}}, \mathbf{s}, \mathbf{t})$

$\mathbf{y} \leftarrow \mathscr{D}_\sigma^{\ell+k}$
$\mathbf{w} := \bar{\mathbf{A}}\mathbf{y} \bmod q$
$c \leftarrow\!\$\ C$
$\mathbf{v} := c \cdot \mathbf{s}$
$\mathbf{z} := \mathbf{v} + \mathbf{y}$
$\rho \leftarrow\!\$\ [0, 1)$
if $\rho > \min\left(\frac{\mathscr{D}_\sigma^{\ell+k}(\mathbf{z})}{M \cdot \mathscr{D}_{\sigma,\mathbf{v}}^{\ell+k}(\mathbf{z})}, 1\right)$ **then**
    $\mathbf{z} := \perp$
return $(\mathbf{A}, \mathbf{t}, \mathbf{w}, c, \mathbf{z})$

$\mathcal{S}(\bar{\mathbf{A}}, \mathbf{t})$

$\mathbf{z} \leftarrow \mathscr{D}_\sigma^{\ell+k}$
$\mathbf{u} := \bar{\mathbf{A}}\mathbf{z} \bmod q$
$c \leftarrow\!\$\ C$
$\mathbf{w} := \mathbf{u} - c \cdot \mathbf{t} \bmod q$
$\rho \leftarrow\!\$\ [0, 1)$
if $\rho > 1/M$ **then**
    $\mathbf{z} := \perp$
return $(\mathbf{A}, \mathbf{t}, \mathbf{w}, c, \mathbf{z})$

B.4 Statistical Honest Verifier Zero Knowledge of the Fiat-Shamir with Aborts Σ-Protocol

In the literature HVZK of the usual FSwA interactive proofs are often only assumed to hold for the *non-aborted* cases, although there have been attempts to simulate the aborted cases as well [ES16,BK20]. For completeness, we present an explicit hybrid argument to prove statistical HVZK of FSwA as a direct consequence of our Lemma 4.3 and Theorem B.1. On a high-level, one can replace $\bar{\mathbf{A}}\mathbf{y}$ with a uniform module element due to the regularity result of [LPR13], which however requires an intermediate hybrid to perform rejection sampling on $\mathbf{z}$ following Gaussian over the *coset* $\Lambda^{\perp}_{\mathbf{w}+c\cdot\mathbf{t}}(\bar{\mathbf{A}})$. This is where our Theorem B.1 comes into play to take care of a mismatch with the original rejection sampling theorem, which assumes $\mathbf{z}$ to be chosen from the entire space.

Lemma B.8. *Suppose R, N, q, k, ℓ, and $\sigma > \eta_\varepsilon(\Lambda^{\perp}(\bar{\mathbf{A}}))$ satisfy the conditions required by Lemma 4.3 and Theorem B.1 with $n = k + \ell$ and $\Sigma = \widehat{\Sigma} = \sigma^2 \cdot \mathbb{I}_{\ell+k}$, and let M be as in Lemma B.4. Let $\mathbf{A} \leftarrow\!\$\ R^{k\times\ell}$, $\bar{\mathbf{A}} := [\mathbf{A}|\mathbb{I}_k]$, $\mathbf{s} \in S_\eta^{\ell+k}$, and $\mathbf{t} := \bar{\mathbf{A}}\mathbf{s} \bmod q$. The output distributions of $\mathcal{T}$ and $\mathcal{S}$ in Alg. 4 are statistically indistinguishable.*

Proof. We prove the above lemma via standard hybrid arguments. Each hybrid is detailed in Alg. 5.

- Hyb_0 is identical to $\mathcal{T}$ above.
- Hyb_1 is identical to Hyb_0, except that $\mathbf{w}$ is sampled uniformly and $\mathbf{y}$ is sampled from Gaussian defined over a coset $\Lambda^{\perp}_{\mathbf{w}}(\bar{\mathbf{A}}) = \left\{\mathbf{x} \in R^{k+\ell} \ :\ \bar{\mathbf{A}}\mathbf{x} = \mathbf{w} \bmod q\right\}$. From Lemma 4.3, Hyb_0 and Hyb_1 are statistically close.
- Hyb_2 is identical to Hyb_1, except that $\mathbf{z}$ is sampled from Gaussian over a coset $\Lambda^{\perp}_{\mathbf{u}}$ centered at $\mathbf{v}$, where $\mathbf{u} = \mathbf{w} + c \cdot \mathbf{t}$. Clearly, the output distribution of Hyb_2 is identical to Hyb_1.
- Hyb_3 is identical to Hyb_2, except that $\mathbf{z}$ is sampled from Gaussian over a coset $\Lambda^{\perp}_{\mathbf{u}}$ centered at 0 and it is output with constant probability $1/M$. Applying Theorem B.1, we have that Hyb_2 and Hyb_3 are statistically close.
- Hyb_4 is identical to Hyb_3, except that $\mathbf{u}$ is uniformly sampled first and $\mathbf{w}$ is set accordingly. Clearly, the output distribution of Hyb_4 is identical to Hyb_3.
- Hyb_5 is identical to Hyb_4, except that $\mathbf{z}$ is sampled from Gaussian over the entire space $R^{\ell+k}$ and $\mathbf{w}$ is defined as $\mathbf{w} = \bar{\mathbf{A}}\mathbf{z} - c\mathbf{t}$. From Lemma 4.3, Hyb_5 and Hyb_4 are statistically close.

The distribution output by Hyb_5 is identical to one by $\mathcal{S}$. This concludes the proof.

□

C Omitted Security Proofs

C.1 Proof for MS-UF-KOA Security (Theorem 4.1)

Proof. We first construct a wrapper $\tilde{\mathcal{B}}$ around MS-UF-KOA adversary $\mathcal{A}$.

Algorithm 5: Hybrids for Lemma B.8

$\mathsf{Hyb}_0(\bar{\mathbf{A}}, \mathbf{s}, \mathbf{t})$
1: $\mathbf{y} \leftarrow \mathscr{D}_\sigma^{\ell+k}$
2: $\mathbf{w} := \bar{\mathbf{A}}\mathbf{y} \bmod q$
3: $c \leftarrow\$ C$
4: $\mathbf{v} := c \cdot \mathbf{s}$
5: $\mathbf{z} := \mathbf{v} + \mathbf{y}$
6: $\rho \leftarrow\$ [0, 1)$
7: **if** $\rho > \min\left(\frac{\mathscr{D}_\sigma^{\ell+k}(\mathbf{z})}{M \cdot \mathscr{D}_{\sigma,\mathbf{v}}^{\ell+k}(\mathbf{z})}, 1\right)$ **then**
8: $\quad \mathbf{z} := \perp$
9: **return** $(\mathbf{A}, \mathbf{t}, \mathbf{w}, c, \mathbf{z})$

$\mathsf{Hyb}_1(\bar{\mathbf{A}}, \mathbf{s}, \mathbf{t})$
1: $\mathbf{w} \leftarrow\$ R_q^k$
2: $\mathbf{y} \leftarrow \mathscr{D}_{\Lambda_{\mathbf{w}}^\perp(\bar{\mathbf{A}}),\sigma}^{\ell+k}$
3: $c \leftarrow\$ C$
4: $\mathbf{v} := c \cdot \mathbf{s}$
5: $\mathbf{z} := \mathbf{v} + \mathbf{y}$
6: $\rho \leftarrow\$ [0, 1)$
7: **if** $\rho > \min\left(\frac{\mathscr{D}_\sigma^{\ell+k}(\mathbf{z})}{M \cdot \mathscr{D}_{\sigma,\mathbf{v}}^{\ell+k}(\mathbf{z})}, 1\right)$ **then**
8: $\quad \mathbf{z} := \perp$
9: **return** $(\mathbf{A}, \mathbf{t}, \mathbf{w}, c, \mathbf{z})$

$\mathsf{Hyb}_2(\bar{\mathbf{A}}, \mathbf{s}, \mathbf{t})$
1: $\mathbf{w} \leftarrow\$ R_q^k$
2: $c \leftarrow\$ C$
3: $\mathbf{v} := c \cdot \mathbf{s}$
4: $\mathbf{u} := \mathbf{w} + c \cdot \mathbf{t}$
5: $\mathbf{z} \leftarrow \mathscr{D}_{\Lambda_{\mathbf{u}}^\perp(\bar{\mathbf{A}}),\sigma,\mathbf{v}}^{\ell+k}$
6: $\rho \leftarrow\$ [0, 1)$
7: **if** $\rho > \min\left(\frac{\mathscr{D}_\sigma^{\ell+k}(\mathbf{z})}{M \cdot \mathscr{D}_{\sigma,\mathbf{v}}^{\ell+k}(\mathbf{z})}, 1\right)$ **then**
8: $\quad \mathbf{z} := \perp$
9: **return** $(\mathbf{A}, \mathbf{t}, \mathbf{w}, c, \mathbf{z})$

$\mathsf{Hyb}_3(\bar{\mathbf{A}}, \mathbf{s}, \mathbf{t})$
1: $\mathbf{w} \leftarrow\$ R_q^k$
2: $c \leftarrow\$ C$
3: $\mathbf{v} := c \cdot \mathbf{s}$
4: $\mathbf{u} := \mathbf{w} + c \cdot \mathbf{t}$
5: $\mathbf{z} \leftarrow \mathscr{D}_{\Lambda_{\mathbf{u}}^\perp(\bar{\mathbf{A}}),\sigma}^{\ell+k}$
6: $\rho \leftarrow\$ [0, 1)$
7: **if** $\rho > 1/M$ **then**
8: $\quad \mathbf{z} := \perp$
9: **return** $(\mathbf{A}, \mathbf{t}, \mathbf{w}, c, \mathbf{z})$

$\mathsf{Hyb}_4(\bar{\mathbf{A}}, \mathbf{s}, \mathbf{t})$
1: $\mathbf{u} \leftarrow\$ R_q^k$
2: $\mathbf{z} \leftarrow \mathscr{D}_{\Lambda_{\mathbf{u}}^\perp(\bar{\mathbf{A}}),\sigma}^{\ell+k}$
3: $c \leftarrow\$ C$
4: $\mathbf{v} := c \cdot \mathbf{s}$
5: $\mathbf{w} := \mathbf{u} - c \cdot \mathbf{t}$
6: $\rho \leftarrow\$ [0, 1)$
7: **if** $\rho > 1/M$ **then**
8: $\quad \mathbf{z} := \perp$
9: **return** $(\mathbf{A}, \mathbf{t}, \mathbf{w}, c, \mathbf{z})$

$\mathsf{Hyb}_5(\bar{\mathbf{A}}, \mathbf{s}, \mathbf{t})$
1: $\mathbf{z} \leftarrow \mathscr{D}_\sigma^{\ell+k}$
2: $\mathbf{u} := \bar{\mathbf{A}}\mathbf{z} \bmod q$
3: $c \leftarrow\$ C$
4: $\mathbf{v} := c \cdot \mathbf{s}$
5: $\mathbf{w} := \mathbf{u} - c \cdot \mathbf{t} \bmod q$
6: $\rho \leftarrow\$ [0, 1)$
7: **if** $\rho > 1/M$ **then**
8: $\quad \mathbf{z} := \perp$
9: **return** $(\mathbf{A}, \mathbf{t}, \mathbf{w}, c, \mathbf{z})$

Lemma C.1. *Let $\tilde{\mathcal{B}}$ be as described in Alg. 6. Let IGen_0 be an input generator that proceeds as follows: $\mathbf{A} \leftarrow\$ R_q^{k\times\ell}$; $\mathbf{s}^* \leftarrow\$ S_\eta^{\ell+k}$; $\mathbf{t}^* := \bar{\mathbf{A}}\mathbf{s}^* \bmod q$; $h_{agg,1}, \ldots, h_{agg,Q} \leftarrow\$ C$; output $\mathsf{in} := (\mathbf{A}, \mathbf{t}^*, h_{agg,1}, \ldots, h_{agg,Q})$. Then we have*

$$acc_0(\tilde{\mathcal{B}}) = \mathsf{Adv}^{\textit{MS-UF-KOA}}_{\textit{MuSig-L}}(\mathcal{A}) \tag{18}$$

where

$$acc_0(\tilde{\mathcal{B}}) \coloneqq \Pr[i_{sig} \geq 1 : \mathsf{in} \leftarrow \mathsf{IGen}_0(1^\lambda); h_{sig,1}, \ldots, h_{sig,Q} \leftarrow\$ C; (i_{sig}, \mathsf{out}) \leftarrow \tilde{\mathcal{B}}(\mathsf{in}, h_{sig,1}, \ldots, h_{sig,Q})].$$

Proof. This follows by inspection. Notice that $\tilde{\mathcal{B}}$ perfectly simulates the view of the adversary $\mathcal{A}$ in the MS-UF-KOA game, by using predetermined random oracle responses. Hence, $\tilde{\mathcal{B}}$ outputs 1 if and only if $\mathcal{A}$ succeeds in creating a valid forgery. □

Next we modify IGen's behavior, so that it now outputs a uniformly sampled public key $\mathbf{t}^*$ instead of generating it as an honest signer would.

Lemma C.2. *Let $\tilde{\mathcal{B}}$ be as described in Alg. 6. Let IGen_1 be an input generator that proceeds as follows: $\mathbf{A} \leftarrow\$ R_q^{k\times\ell}$; $\mathbf{t}^* \leftarrow\$ R_q^k$; $h_{agg,1}, \ldots, h_{agg,Q} \leftarrow\$ C$; output $\mathsf{in} := (\mathbf{A}, \mathbf{t}^*, h_{agg,1}, \ldots, h_{agg,Q})$. Then there exists a PPT algorithm $\mathcal{B}'$ such that*

$$|acc_1(\tilde{\mathcal{B}}) - acc_0(\tilde{\mathcal{B}})| \leq \mathsf{Adv}^{\mathsf{MLWE}}_{q,k,\ell,\eta}(\mathcal{B}') \tag{19}$$

where

$$acc_1(\tilde{\mathcal{B}}) \coloneqq \Pr[i_{sig} \geq 1 : \mathsf{in} \leftarrow \mathsf{IGen}_1(1^\lambda); h_{sig,1}, \ldots, h_{sig,Q} \leftarrow\$ C; (i_{sig}, \mathsf{out}) \leftarrow \tilde{\mathcal{B}}(\mathsf{in}, h_{sig,1}, \ldots, h_{sig,Q})].$$

Proof. We construct a distinguisher $\mathcal{B}'$ against the $\mathsf{MLWE}_{q,k,\ell,\eta}$ problem. Given an $\mathsf{MLWE}_{q,k,\ell,\eta}$ instance $(\mathbf{A}, \mathbf{t}^*) \in R_q^{k\times\ell} \times R_q^k$ as input, $\mathcal{B}'$ samples $h_{\mathrm{sig},1}, \ldots, h_{\mathrm{sig},Q}, h_{\mathrm{agg},1}, \ldots, h_{\mathrm{agg},Q} \in C$ as in $\mathrm{acc}_0()$ and $\mathrm{acc}_1()$. Then $\mathcal{B}'$ constructs inputs to $\tilde{\mathcal{B}}$ and outputs 1 if the return value of $\tilde{\mathcal{B}}$ satisfies $i_{\mathrm{sig}} \geq 1$ and 0 otherwise. By construction, we get the bound in the statement. □

Next we construct a slightly modified wrapper $\mathcal{B}$. Unlike $\tilde{\mathcal{B}}$, it aborts whenever either of the flags $\mathsf{bad}_{\mathrm{agg}}$ or $\mathsf{bad}_{\mathrm{kcol}}$ is set to true.

Lemma C.3. *Let $\mathcal{B}$ be as described in Alg. 6. Let IGen_1 and $acc_1()$ be as described in Lemma C.2. Then we have*

$$|acc_1(\mathcal{B}) - acc_1(\tilde{\mathcal{B}})| \leq \Pr[\mathsf{bad}_{agg}] + \Pr[\mathsf{bad}_{\mathrm{kcol}}]. \tag{20}$$

Proof. By construction, the view of the adversary $\mathcal{A}$ in an execution of $\mathcal{B}$ and $\tilde{\mathcal{B}}$ is identical unless either $\mathsf{bad}_{\mathrm{agg}}$ or $\mathsf{bad}_{\mathrm{kcol}}$ is set. By the union bound, we obtain the statement. □

Let us bound the probability that each bad event occurs.

Lemma C.4. $\Pr[\mathsf{bad}_{agg}] \leq \frac{Q(Q+1)}{|C|} + \left(\frac{2}{q^{N/2}}\right)^k$

Proof. We consider two cases: (1) for all $1 \leq i \leq k$, $t_i^* \notin R_q^\times$, and (2) there exists $1 \leq i \leq k$ such that $t_i^* \in R_q^\times$, where $\mathbf{t}^* := [t_1^*, \ldots, t_k^*]^T$ is uniformly generated honest party's key. From Lemma 2.2, the probability that k uniform ring elements are simultaneously non-invertible is at most $(2/q^{N/2})^k$. Hence, the $\Pr[\mathsf{bad}_{\mathrm{agg}} \wedge \forall i \in [1,k] : t_i^* \notin R_q^\times]$ is bounded by $(2/q^{N/2})^k$.

Let us consider the latter case. As the flag $\mathsf{bad}_{\mathrm{agg}}$ is set only if the adversary has queried $\mathsf{H}_{\mathrm{sig}}$ with $\tilde{\mathbf{t}}$ before querying $\mathsf{H}_{\mathrm{agg}}$, it amounts to bounding the following probability for a given uniform key $\mathbf{t}_1 = \mathbf{t}^*$.

$$\begin{aligned}
&\max_{\tilde{\mathbf{t}},\mathbf{t}_2,\ldots,\mathbf{t}_n} \Pr\left[\tilde{\mathbf{t}} = \sum_{i=1}^{n} a_i \mathbf{t}_i \bmod q \;:\; a_1,\ldots,a_n \leftarrow\!\$\, C\right] \\
=&\max_{\tilde{\mathbf{t}},\mathbf{t}_2,\ldots,\mathbf{t}_n} \Pr\left[a_1\mathbf{t}_1 = \tilde{\mathbf{t}} - \sum_{i=2}^{n} a_i \mathbf{t}_i \bmod q \;:\; a_1,\ldots,a_n \leftarrow\!\$\, C\right] \\
\leq&\max_{\tilde{\mathbf{t}},\mathbf{t}_2,\ldots,\mathbf{t}_n,a_2,\ldots,a_n} \Pr\left[a_1\mathbf{t}_1 = \tilde{\mathbf{t}} - \sum_{i=2}^{n} a_i \mathbf{t}_i \bmod q \;:\; a_1 \leftarrow\!\$\, C\right]
\end{aligned}$$

Now, given that the honest public key $\mathbf{t}_1 = \mathbf{t}^*$ has at least one invertible element, the above probability can be bounded by the probability that uniform a_1 hits a specific ring element, which is at most $1/|C|$. The adversary $\mathcal{A}$ queries $\mathsf{H}_{\mathrm{agg}}$ at most Q times and one of the additional calls to $\mathsf{H}_{\mathrm{agg}}$ at the end made by $\mathcal{B}$ may potentially set $\mathsf{bad}_{\mathrm{agg}}$ to true. Since the bad event occurs if the aggregated public key derived inside $\mathsf{H}_{\mathrm{agg}}$ hits any of at most Q entries already present in $\mathrm{HT}_{\mathrm{sig}}$, by the union bound we get $\Pr[\mathsf{bad}_{\mathrm{agg}} \wedge \exists i \in [1,k] : t_i^* \notin R_q^\times] \leq \frac{Q(Q+1)}{|C|}$. □

Lemma C.5. $\Pr[\mathsf{bad}_{\mathrm{kcol}}] \leq \frac{Q(Q+1)}{|C|} + \left(\frac{2}{q^{N/2}}\right)^k$

Proof. The proof is analogous to one for $\mathsf{bad}_{\mathrm{agg}}$. The probability that all elements in $\mathbf{t}^*$ are simultaneously non-invertible is again at most $(2/q^{N/2})^k$ thanks to Lemma 2.2. Assuming there exists some invertible entry in $\mathbf{t}^*$, it amounts to finding the probability that the aggregated public key derived inside $\mathsf{H}_{\mathrm{agg}}$ hits any of at most Q entries already present in KT. By the same argument as above, and by the union bound, we get $\Pr[\mathsf{bad}_{\mathrm{kcol}} \wedge \exists i \in [1,k] : t_i^* \notin R_q^\times] \leq \frac{Q(Q+1)}{|C|}$. □

Now we construct another wrapper $\mathcal{C}$, which internally invokes the forker algorithm $\mathcal{F}_{\mathcal{B}}$ (Alg. 1) from Lemma 2.13. $\mathcal{F}_{\mathcal{B}}$ takes care of rewinding $\mathcal{B}$ and outputs two forgeries with distinct challenges.

Lemma C.6. *Let $\mathcal{C}$ be as described in Alg. 6. Let IGen_2 be an input generator that works as follows: $\mathbf{A} \leftarrow\!\$\, R_q^{k\times\ell}$; $\mathbf{t}^* \leftarrow\!\$\, R^k$; output $\mathsf{in}_{\mathcal{C}} := (\mathbf{A}, \mathbf{t}^*)$. Then we have*

$$acc_1(\mathcal{B}) \leq \frac{Q}{|C|} + \sqrt{Q \cdot acc_2(\mathcal{C})} \tag{21}$$

where

$$acc_2(\mathcal{C}) := \Pr[i_{agg} \geq 1 : \mathsf{in}_{\mathcal{C}} \leftarrow \mathsf{IGen}_2(1^\lambda); h_{agg,1},\ldots,h_{agg,Q} \leftarrow\!\$\, C; (i_{agg}, \mathsf{out}_{\mathcal{C}}) \leftarrow \mathcal{C}(\mathsf{in}_{\mathcal{C}}, h_{agg,1},\ldots,h_{agg,Q})].$$

Moreover, if $\mathcal{C}$ halts with output $i_{agg} \geq 1$ and $\mathsf{out}_{\mathcal{C}} = (L^*, \vec{a}, c^*, \mathbf{z}^*, \hat{c}^*, \hat{\mathbf{z}}^*)$, where $L = \{\mathbf{t}_1 = \mathbf{t}^*, \mathbf{t}_2, \ldots, \mathbf{t}_{n^*}\}$, it holds that

$$\bar{\mathbf{A}}\mathbf{z}^* - c^* \sum_{i\neq 1} a_i \mathbf{t}_i - c^* a_1 \mathbf{t}^* = \bar{\mathbf{A}}\hat{\mathbf{z}}^* - \hat{c}^* \sum_{i\neq 1} a_i \mathbf{t}_i - \hat{c}^* a_1 \mathbf{t}^* \bmod q \;\wedge\; \|\mathbf{z}^*\|_2 \leq B_n \;\wedge\; \|\hat{\mathbf{z}}^*\|_2 \leq B_n \tag{22}$$

Proof. Until the forking point, i.e., when the adversary receives the i_{sig}th challenge from $\mathsf{H}_{\mathrm{sig}}$, the forker $\mathcal{F}_{\mathcal{B}}$ uses the same random coin and the same random oracle responses $h_{\mathrm{sig},1},\ldots,h_{\mathrm{sig},i_{\mathrm{sig}}-1}$ to answer the queries made by $\mathcal{B}$. Therefore, the view of the adversary $\mathcal{B}$ in both runs is identical, implying that the inputs to the i_{sig}th query are identical in both runs: $\tilde{\mathbf{w}}^* = \hat{\tilde{\mathbf{w}}}^*$, $\mu^* = \hat{\mu}^*$, and $\tilde{\mathbf{t}}^* = \hat{\tilde{\mathbf{t}}}^*$.

We argue that, as long as neither of $\mathsf{bad}_{\mathrm{agg}}$ nor $\mathsf{bad}_{\mathrm{kcol}}$ is set, it holds that $i_{\mathrm{agg}} = \hat{i}_{\mathrm{agg}}$, $L^* = \hat{L}^*$, and $\vec{a} = \vec{\hat{a}}$. First, it must be that $L^* = \hat{L}^*$ because otherwise two different public key lists would lead to the same aggregated key $\tilde{\mathbf{t}}^*$, contradicting the assumption that $\mathsf{bad}_{\mathrm{kcol}}$ was not set. If $\mathsf{bad}_{\mathrm{agg}}$ is not set, then a query to $\mathsf{H}_{\mathrm{sig}}$ with input $(\tilde{\mathbf{w}}^*, \mu, \tilde{\mathbf{t}}^*)$ must have been made *after* the corresponding query to $\mathsf{H}_{\mathrm{agg}}$ with input $(L^*, \mathbf{t}^*)$ leading to $\tilde{\mathbf{t}}^*$. Since all the value assigned before the forking point are identical in both runs, we have that $i_{\mathrm{agg}} = \hat{i}_{\mathrm{agg}}$ and $\vec{a} = \vec{\hat{a}}$. Because outputs from both runs satisfy the verification conditions, we get (22).

Finally, by the general forking lemma (Lemma 2.13), we find the upper bound for $\mathrm{acc}_1(\mathcal{B})$ as stated in the lemma. □

Finally, we construct yet another wrapper $\mathcal{D}$, which internally invokes the forker algorithm $\mathcal{F}_{\mathcal{C}}$ (Alg. 1) from Lemma 2.13. $\mathcal{F}_{\mathcal{C}}$ takes care of rewinding $\mathcal{C}$ and outputs two four forgeries with distinct challenges and distinct aggregation coefficients for the honest public key.

Lemma C.7. *Let $\mathcal{D}$ as described in Alg. 6 and define*

$$acc_3(\mathcal{D}) := \Pr\left[\mathsf{out}_{\mathcal{D}} \neq \bot \ :\ \mathbf{A} \leftarrow\!\$\ R_q^{k\times\ell}; \mathbf{t}^* \leftarrow\!\$\ R_q^k; \mathsf{out}_{\mathcal{D}} \leftarrow \mathcal{D}(\mathbf{A}||\mathbf{t}^*)\right].$$

Then we have

$$acc_2(\mathcal{C}) \leq \frac{Q}{|C|} + \sqrt{Q \cdot acc_3(\mathcal{D})} \tag{23}$$

Moreover, if $\mathcal{D}$ halts with output $\mathsf{out}_{\mathcal{D}} = \mathbf{x} \in R^{\ell+k+1}$, it holds that

$$[\mathbf{A}|\mathbb{I}_k|\mathbf{t}^*]\mathbf{x} = \mathbf{0} \bmod q \ \wedge\ \mathbf{x} \neq \mathbf{0} \ \wedge\ \|\mathbf{x}\|_2 \leq 8\kappa\sqrt{B_n^2 + \kappa^3}. \tag{24}$$

Proof. Until the forking point, i.e., when $\mathsf{H}_{\mathrm{agg}}$ uses the i_{agg}th aggregation coefficient $h_{\mathrm{agg},i_{\mathrm{agg}}}$ to define the entry $\mathrm{HT}_{\mathrm{agg}}[L^*, \mathbf{t}^*]$, the view of the adversary is identical in 4 runs. Hence, we have $L^* := L_1^* = L_2^*$. Since the coefficients corresponding to $\mathbf{t}_i \in L^*$ for $i \geq 2$ are assigned before the one for $\mathbf{t}_1$, we also have $a_2 := a_{1,2} = a_{2,2}, \ldots, a_{n^*} := a_{1,n^*} = a_{2,n^*}$. Hence, from Lemma C.6, the values in $\mathsf{out}_{\mathcal{C}}$ and $\hat{\mathsf{out}}_{\mathcal{C}}$ satisfy the following equations

$$\bar{\mathbf{A}}\mathbf{z}_1^* - c_1^* \sum_{i\neq 1} a_i\mathbf{t}_i - c_1^* a_{1,1}\mathbf{t}^* = \bar{\mathbf{A}}\hat{\mathbf{z}}_1^* - \hat{c}_1^* \sum_{i\neq 1} a_i\mathbf{t}_i - \hat{c}_1^* a_{1,1}\mathbf{t}^* \bmod q \tag{25}$$

$$\bar{\mathbf{A}}\mathbf{z}_2^* - c_2^* \sum_{i\neq 1} a_i\mathbf{t}_i - c_2^* a_{2,1}\mathbf{t}^* = \bar{\mathbf{A}}\hat{\mathbf{z}}_2^* - \hat{c}_2^* \sum_{i\neq 1} a_i\mathbf{t}_i - \hat{c}_2^* a_{2,1}\mathbf{t}^* \bmod q \tag{26}$$

where, in particular, $c_1^* \neq \hat{c}_1^*$, $c_2^* \neq \hat{c}_2^*$, and $a_{1,1} \neq a_{2,1}$ thanks to the forker algorithms $\mathcal{F}_{\mathcal{B}}$ and $\mathcal{F}_{\mathcal{C}}$, respectively. Rearranging the above equations, we get that

$$\bar{\mathbf{A}}\bar{\mathbf{z}}_1 - \bar{c}_1 \sum_{i\neq 1} a_i\mathbf{t}_i - \bar{c}_1 a_{1,1}\mathbf{t}^* = \mathbf{0} \bmod q \tag{27}$$

$$\bar{\mathbf{A}}\bar{\mathbf{z}}_2 - \bar{c}_2 \sum_{i\neq 1} a_i\mathbf{t}_i - \bar{c}_2 a_{2,1}\mathbf{t}^* = \mathbf{0} \bmod q \tag{28}$$

where $\bar{\mathbf{z}}_i = \mathbf{z}_i^* - \hat{\mathbf{z}}_i^*$ and $\bar{c}_i = c_i^* - \hat{c}_i^*$ for $i = 1, 2$. By multiplying the former by $\bar{c}_2$ and the latter by $\bar{c}_1$, respectively, we eventually have

$$\bar{\mathbf{A}}(\bar{c}_2\bar{\mathbf{z}}_1 - \bar{c}_1\bar{\mathbf{z}}_2) - \bar{c}_1\bar{c}_2\bar{a}\mathbf{t}^* = \mathbf{0} \bmod q \tag{29}$$

$$\tag{30}$$

where $\bar{a} = a_{1,1} - a_{2,1}$. Since $\bar{c}_1$, $\bar{c}_2$, and $\bar{a}$ are all non-zero and none of them are zero-divisors due to Lemma 2.1, the solution $\mathbf{x}$ output by $\mathcal{D}$ is guaranteed to be non-zero.

Let us bound L^2-norm of $\mathbf{x}$. By the verification conditions, we have $\|\bar{\mathbf{z}}_i\|_2 \leq 2B_n$ for $i = 1, 2$. Since $\bar{c}_i \in \bar{C}$, it also holds that $\|\bar{c}_i\|_1 \leq 2\kappa$. Therefore, we obtain $\|\bar{c}_2\bar{\mathbf{z}}_1\|_2 \leq \|\bar{c}_2\|_1\|\bar{\mathbf{z}}_1\|_2 \leq 4B_n\kappa$. As $\|\bar{c}_1\bar{\mathbf{z}}_2\|_2$ has the same bound, the difference $\bar{c}_2\bar{\mathbf{z}}_1 - \bar{c}_1\bar{\mathbf{z}}_2$ has L^2-norm bounded by $8B_n\kappa$. Since $\bar{a} \in \bar{C}$, we also

have $\|\bar{a}\|_2 \leq 2\sqrt{\kappa}$ and thus $\|\bar{c}_1\bar{c}_2\bar{a}\|_2 \leq \|\bar{c}_1\|_1\|\bar{c}_2\|_1\|\bar{a}\|_2 \leq 8\kappa^2\sqrt{\kappa}$. Putting them together, we obtain $\|\mathbf{x}\|_2 \leq 8\kappa\sqrt{B_n^2+\kappa^3}$.

Finally, by the general forking lemma (Lemma 2.13), we find the upper bound for $\mathrm{acc}_2(\mathcal{C})$ as stated in the lemma. □

Now we are ready to derive the statement of Theorem 4.1. From Lemmas C.1 to C.3, we get

$$\mathsf{Adv}^{\mathsf{MS\text{-}UF\text{-}KOA}}_{\mathsf{MuSig\text{-}L}}(\mathcal{A}) \leq \mathrm{acc}_1(\mathcal{B}) + \frac{2Q(Q+1)}{|C|} + \frac{2^{k+1}}{q^{kN/2}} + \mathsf{Adv}^{\mathsf{MLWE}}_{q,k,\ell,\eta}(\mathcal{B}').$$

By plugging the inequalities of Lemma C.6 and Lemma C.7 into the above,

$$\mathsf{Adv}^{\mathsf{MS\text{-}UF\text{-}KOA}}_{\mathsf{MuSig\text{-}L}}(\mathcal{A}) \leq \frac{Q(2Q+3)}{|C|} + \frac{2^{k+1}}{q^{kN/2}} + \mathsf{Adv}^{\mathsf{MLWE}}_{q,k,\ell,\eta}(\mathcal{B}') + \sqrt{\frac{Q^2}{|C|} + Q\sqrt{Q\cdot\mathrm{acc}_3(\mathcal{D})}}.$$

The statement of Lemma C.7 tells that whenever $\mathcal{D}$ halts with some output $\mathbf{x} \neq \perp$, it is a valid solution to the $\mathsf{MSIS}_{q,k,\ell,\beta}$ problem with norm bound $\beta = 8\kappa\sqrt{B_n^2+\kappa^3}$. Therefore $\mathrm{acc}_3(\mathcal{D}) = \mathsf{Adv}^{\mathsf{MSIS}}_{q,k,\ell+1,\beta}(\mathcal{D})$. Putting together, we get the bound in the statement. □

C.2 Proof for MS-UF-CMA Security (Theorem 4.5)

Proof. We prove via several game hops. We denote by $\Pr[\mathbf{G}_i(\mathcal{X})]$ the probability that $\mathbf{G}_i(\mathcal{X})$ halts with output 1. The random oracles in the MS-UF-CMA game (resp. MS-UF-KOA game) are denoted by $\mathsf{H}'_{\mathrm{agg}}, \mathsf{H}'_{\mathrm{non}}, \mathsf{H}'_{\mathrm{sig}}$ (resp. $\mathsf{H}_{\mathrm{agg}}, \mathsf{H}_{\mathrm{non}}, \mathsf{H}_{\mathrm{sig}}$), respectively.

$\mathbf{G}_0$ This game is identical to the MS-UF-CMA game:

$$\Pr[\mathbf{G}_0(\mathcal{X})] = \mathsf{Adv}^{\mathsf{MS\text{-}UF\text{-}CMA}}_{\mathsf{MuSig\text{-}L}}(\mathcal{X}). \tag{31}$$

$\mathbf{G}_1$ This game is identical to $\mathbf{G}_0$, except that $\mathsf{OSignOff}$ keeps all the commit messages and their preimages in the table WT for each session ID and aborts if WT has identical entries associated to different session IDs: it performs $\mathrm{WT}[\mathsf{sid}] := ((\mathbf{w}_1^{(j)})_{j\in[m]}, (\mathbf{y}_1^{(j)})_{j\in[m]})$ after generating $\mathbf{w}_1^{(j)}$'s, and sets $\mathsf{bad}_{\mathrm{wcol}}$ to true if there exists $\mathsf{sid}' \neq \mathsf{sid}$ such that $\mathrm{WT}[\mathsf{sid}'] = \mathrm{WT}[\mathsf{sid}]$.

Since the adversary $\mathcal{X}$ makes at most Q_s queries to $\mathsf{OSignOff}$ and each $\mathbf{w}_1^{(j)}$ is statistically close to $\mathscr{U}(R_q^k)$ from Lemma 4.2, by the union bound, we get

$$|\Pr[\mathbf{G}_1(\mathcal{X})] - \Pr[\mathbf{G}_0(\mathcal{X})]| \leq \Pr[\mathsf{bad}_{\mathrm{wcol}}] \leq Q_s^2 \cdot \left(\frac{1+2^{-\Omega(N)}}{q^{kN}}\right)^m. \tag{32}$$

$\mathbf{G}_2$ This game is identical to $\mathbf{G}_1$, except that $\mathsf{OSignOff}$ aborts if $\mathcal{X}$ has previously queried $\mathsf{H}'_{\mathrm{non}}$ with input containing $(\mathbf{w}_1^{(j)})_{j\in[m]}$: it sets $\mathsf{bad}_{\mathrm{non}}$ to true if there exists $(\{\mathbf{t}_i||\mathsf{com}_i\}_{i\in[n]}, \mu, \tilde{\mathbf{t}})$ such that $\mathrm{HT}'_{\mathrm{non}}[\langle\{\mathbf{t}_i||\mathsf{com}_i\}_{i\in[n]}\rangle, \mu, \tilde{\mathbf{t}}] \neq \perp$ and $\mathsf{com}_1 = (\mathbf{w}_1^{(j)})_{j\in[m]}$.

Since the adversary $\mathcal{X}$ makes at most Q_h queries to $\mathsf{H}'_{\mathrm{non}}$ and Q_s queries to $\mathsf{OSignOn}$ (which internally invokes $\mathsf{H}'_{\mathrm{non}}$), by the union bound, we get

$$|\Pr[\mathbf{G}_2(\mathcal{X})] - \Pr[\mathbf{G}_1(\mathcal{X})] \leq \Pr[\mathsf{bad}_{\mathrm{non}}] \leq (Q_h+Q_s)^2 \cdot \left(\frac{1+2^{-\Omega(N)}}{q^{kN}}\right)^m. \tag{33}$$

$\mathbf{G}_3$ This game is identical to $\mathbf{G}_2$, except $\mathsf{H}'_{\mathrm{non}}$ takes care of generating the online output $(\mathbf{z}_1, \tilde{\mathbf{w}})$ and keeps it in the table ST, as soon as it receives a valid input for which $\mathsf{OSignOn}$ may potentially generate a signature later. Then $\mathsf{OSignOn}$ simply looks up the table ST associated with a set of $(\mathbf{t}_i, \mathsf{com}_i)$, μ, and $\tilde{\mathbf{t}}$ to answer queries. Concretely, it proceeds as in Alg. 7. As this is equivalent to the previous game, we have

$$\Pr[\mathbf{G}_3(\mathcal{X})] = \Pr[\mathbf{G}_2(\mathcal{X})] \tag{34}$$

$\mathbf{G}_4$ This game is identical to $\mathbf{G}_3$, except that, if $\mathsf{H}'_{\mathrm{non}}$ receives a valid input for which $\mathsf{OSignOn}$ may potentially generate a signature later, it samples challenge c and tries to program $\mathsf{H}'_{\mathrm{sig}}$ such that $\mathsf{H}'_{\mathrm{sig}}(\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}) =$

Algorithm 6: Reduction to $\mathsf{MSIS}_{q,k,\ell+1,\beta}$ and $\mathsf{MLWE}_{q,k,\ell,\eta}$

$\tilde{\mathcal{B}}(\mathsf{in}, h_{\mathrm{sig},1}, \ldots, h_{\mathrm{sig},Q}; \rho) / \mathcal{B}(\mathsf{in}, h_{\mathrm{sig},1}, \ldots, h_{\mathrm{sig},Q}; \rho)$

// Lines highlighted in orange are only executed in $\mathcal{B}$

$(\mathbf{A}, \mathbf{t}^*, h_{\mathrm{agg},1}, \ldots, h_{\mathrm{agg},Q}) := \mathsf{in}$
$\mathsf{ctr}_{\mathrm{agg}} := 0$; $\mathsf{ctr}_{\mathrm{sig}} := 0$;
$\mathsf{pp} := \bar{\mathbf{A}} = [\mathbf{A} | \mathbb{I}_k]$; $\mathsf{pk}_1 := \mathbf{t}^*$
$(L^*, \mu^*, (\tilde{\mathbf{w}}^*, \mathbf{z}^*)) \leftarrow \mathcal{A}^{\mathsf{H}_{\mathrm{agg}}, \mathsf{H}_{\mathrm{non}}, \mathsf{H}_{\mathrm{sig}}}(\mathsf{pp}, \mathsf{pk}_1)$
if $\mathbf{t}^* \notin L^*$ **then**
    return $(0, \perp)$
$n^* := |L^*|$
$\{\mathbf{t}_1 = \mathbf{t}^*, \mathbf{t}_2, \ldots, \mathbf{t}_{n^*}\} := L^*$
for $i \in [1, n^*]$ **do**
    $a_i := \mathsf{H}_{\mathrm{agg}}(L^*, \mathbf{t}_i)$
$\vec{a} := (a_1, \ldots, a_{n^*})$
$\tilde{\mathbf{t}}^* := \sum_{i=1}^{n^*} a_i \mathbf{t}_i \bmod q$
$c^* := \mathsf{H}_{\mathrm{sig}}(\tilde{\mathbf{w}}^*, \mu^*, \tilde{\mathbf{t}}^*)$
if $\bar{\mathbf{A}}\mathbf{z}^* - c^* \tilde{\mathbf{t}}^* \bmod q \neq \tilde{\mathbf{w}} \vee \|\mathbf{z}^*\|_2 > B_n$ **then**
    return $(0, \perp)$
Find i_{agg} such that $\mathrm{HT}_{\mathrm{agg}}[L^*, \mathbf{t}^*] = h_{\mathrm{agg},i_{\mathrm{agg}}}$
Find i_{sig} such that $\mathrm{HT}_{\mathrm{sig}}[\tilde{\mathbf{w}}^*, \mu^*, \tilde{\mathbf{t}}^*] = h_{\mathrm{sig},i_{\mathrm{sig}}}$
$\mathsf{out} := (i_{\mathrm{agg}}, \tilde{\mathbf{w}}^*, \mu^*, L^*, \vec{a}, c^*, \mathbf{z}^*)$
return $(i_{\mathrm{sig}}, \mathsf{out})$

$\mathcal{C}(\mathsf{in}_{\mathcal{C}}, h_{\mathrm{agg},1}, \ldots, h_{\mathrm{agg},Q}; \rho_{\mathcal{C}})$

$(\mathbf{A}, \mathbf{t}^*) := \mathsf{in}_{\mathcal{C}}$
$\mathsf{in}_{\mathcal{B}} := (\mathbf{A}, \mathbf{t}^*, h_{\mathrm{agg},1}, \ldots, h_{\mathrm{agg},Q})$
$(b, \mathsf{out}_{\mathcal{B}}, \hat{\mathsf{out}}_{\mathcal{B}}) \leftarrow \mathcal{F}_{\mathcal{B}}(\mathsf{in}_{\mathcal{B}})$
$(i_{\mathrm{agg}}, \tilde{\mathbf{w}}^*, \mu^*, L^*, \vec{a}, c^*, \mathbf{z}^*) := \mathsf{out}_{\mathcal{B}}$
$(\hat{i}_{\mathrm{agg}}, \hat{\tilde{\mathbf{w}}}^*, \hat{\mu}^*, \hat{L}^*, \vec{\hat{a}}, \hat{c}^*, \hat{\mathbf{z}}^*) := \hat{\mathsf{out}}_{\mathcal{B}}$
if $b = 0$ **then**
    return $(0, \perp)$
$\mathsf{out}_{\mathcal{C}} := (L^*, \vec{a}, c^*, \mathbf{z}^*, \hat{c}^*, \hat{\mathbf{z}}^*)$
return $(i_{\mathrm{agg}}, \mathsf{out}_{\mathcal{C}})$

$\mathcal{D}(\mathbf{A}')$

$[\mathbf{A} | \mathbf{t}^*] := \mathbf{A}'$
$\mathsf{in}_{\mathcal{C}} := (\mathbf{A}, \mathbf{t}^*)$
$(b, \mathsf{out}_{\mathcal{C}}, \hat{\mathsf{out}}_{\mathcal{C}}) \leftarrow \mathcal{F}_{\mathcal{C}}(\mathsf{in}_{\mathcal{C}})$
$(L_1^*, \vec{a}_1, c_1^*, \mathbf{z}_1^*, \hat{c}_1^*, \hat{\mathbf{z}}_1^*) := \mathsf{out}_{\mathcal{C}}$
$(L_2^*, \vec{a}_2, c_2^*, \mathbf{z}_2^*, \hat{c}_2^*, \hat{\mathbf{z}}_2^*) := \hat{\mathsf{out}}_{\mathcal{C}}$
$\{\mathbf{t}_1 := \mathbf{t}^*, \mathbf{t}_2, \ldots, \mathbf{t}_{n^*}\} := L_1^*$
$(a_{1,1}, a_{1,2}, \ldots, a_{1,n^*}) := \vec{a}_1$
$(a_{2,1}, a_{2,2}, \ldots, a_{2,n^*}) := \vec{a}_2$
if $b = 0$ **then**
    return $\perp$
$\bar{c}_1 := c_1^* - \hat{c}_1^*$; $\bar{c}_2 := c_2^* - \hat{c}_2^*$
$\bar{\mathbf{z}}_1 := \mathbf{z}_1^* - \hat{\mathbf{z}}_1^*$; $\bar{\mathbf{z}}_2 := \mathbf{z}_2^* - \hat{\mathbf{z}}_2^*$
$\bar{a} := a_{1,1} - a_{2,1}$
$\mathbf{x} := \begin{bmatrix} \bar{c}_2 \bar{\mathbf{z}}_1 - \bar{c}_1 \bar{\mathbf{z}}_2 \\ -\bar{c}_1 \bar{c}_2 \bar{a} \end{bmatrix}$
return $\mathbf{x}$

$\mathcal{B}'(\mathbf{A}, \mathbf{t}^*)$

$h_{\mathrm{sig},1}, \ldots, h_{\mathrm{sig},Q}, h_{\mathrm{agg},1}, \ldots, h_{\mathrm{agg},Q} \leftarrow\$ C$
$\mathsf{in} := (\mathbf{A}, \mathbf{t}^*, h_{\mathrm{agg},1}, \ldots, h_{\mathrm{agg},Q})$
$(i_{\mathrm{sig}}, \mathsf{out}) \leftarrow \tilde{\mathcal{B}}(\mathsf{in}, h_{\mathrm{sig},1}, \ldots, h_{\mathrm{sig},Q})$
if $i_{\mathrm{sig}} \geq 1$ **then**
    return 1
else
    return 0

$\mathsf{H}_{\mathrm{agg}}(L, \mathbf{t})$

if $\mathrm{HT}_{\mathrm{agg}}[L, \mathbf{t}] \neq \perp$ **then**
    return $\mathrm{HT}_{\mathrm{agg}}[L, \mathbf{t}]$
if $\mathbf{t}^* \notin L \vee \mathbf{t} \notin L$ **then**
    $\mathrm{HT}_{\mathrm{agg}}[L, \mathbf{t}] \leftarrow\$ C$
    return $\mathrm{HT}_{\mathrm{agg}}[L, \mathbf{t}]$
for $\mathbf{t}' \in L \setminus \{\mathbf{t}^*\}$ **do**
    $\mathrm{HT}_{\mathrm{agg}}[L, \mathbf{t}'] \leftarrow\$ C$
$\mathsf{ctr}_{\mathrm{agg}} ++$
$\mathrm{HT}_{\mathrm{agg}}[L, \mathbf{t}^*] := h_{\mathrm{agg},\mathsf{ctr}_{\mathrm{agg}}}$
$\{\mathbf{t}_1 = \mathbf{t}^*, \mathbf{t}_2, \ldots, \mathbf{t}_n\} := L$
for $i \in [1, n]$ **do**
    $a_i := \mathrm{HT}_{\mathrm{agg}}[L, \mathbf{t}_i]$
$\tilde{\mathbf{t}} := \sum_{i=1}^{n} a_i \mathbf{t}_i \bmod q$
$\mathrm{KT}[L] := \tilde{\mathbf{t}}$
if $\exists (\tilde{\mathbf{w}}, \mu) : \mathrm{HT}_{\mathrm{sig}}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}] \neq \perp$ **then**
    $\mathsf{bad}_{\mathrm{agg}} := \mathsf{true}$
if $\exists L' : L' \neq L \wedge \mathrm{KT}[L'] = \tilde{\mathbf{t}}$ **then**
    $\mathsf{bad}_{\mathrm{kcol}} := \mathsf{true}$
return $\mathrm{HT}_{\mathrm{agg}}[L, \mathbf{t}]$

$\mathsf{H}_{\mathrm{sig}}(\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}})$

if $\mathrm{HT}_{\mathrm{sig}}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}] = \perp$ **then**
    $\mathsf{ctr}_{\mathrm{sig}} ++$
    $\mathrm{HT}_{\mathrm{sig}}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}] := h_{\mathrm{sig},\mathsf{ctr}_{\mathrm{sig}}}$
return $\mathrm{HT}_{\mathrm{sig}}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}]$

$\mathsf{H}_{\mathrm{non}}(W, \mu, \tilde{\mathbf{t}})$

if $\mathrm{HT}_{\mathrm{non}}[W, \mu, \tilde{\mathbf{t}}] = \perp$ **then**
    $\mathrm{HT}_{\mathrm{non}}[W, \mu, \tilde{\mathbf{t}}] \leftarrow\$ \{0,1\}^l$
return $\mathrm{HT}_{\mathrm{non}}[W, \mu, \tilde{\mathbf{t}}]$

c. It then aborts with $\mathsf{bad}_{\mathrm{sig}} = \mathsf{true}$ if $\tilde{\mathbf{w}}$ has been previously queried to $\mathsf{H}'_{\mathrm{sig}}$ by the adversary. The additional operations are highlighted in green in Alg. 7.

Let us bound $\Pr[\mathsf{bad}_{\mathrm{sig}}]$. It amounts to bounding the following probability.

$$\begin{aligned}
&\max_{\tilde{\mathbf{w}},\mathbf{w}^{(1)},\ldots,\mathbf{w}^{(m)}} \Pr\left[\tilde{\mathbf{w}} = \sum_{j=1}^{m} b^{(j)}\mathbf{w}^{(j)} \bmod q \ :\ b^{(2)},\ldots,b^{(m)} \leftarrow \mathscr{D}_{\sigma_b}; b^{(1)} = 1\right] \\
\leq &\max_{\tilde{\mathbf{w}},\mathbf{w}^{(1)},\ldots,\mathbf{w}^{(m)}} \Pr\left[b^{(m)}\mathbf{w}^{(m)} = \tilde{\mathbf{w}} - \sum_{j=1}^{m-1} b^{(j)}\mathbf{w}^{(j)} \bmod q \ :\ b^{(2)},\ldots,b^{(m)} \leftarrow \mathscr{D}_{\sigma_b}; b^{(1)} = 1\right] \\
\leq &\max_{\tilde{\mathbf{w}},\mathbf{w}^{(1)},\ldots,\mathbf{w}^{(m)},b^{(2)},\ldots,b^{(m-1)}} \Pr\left[b^{(m)}\mathbf{w}^{(m)} = \tilde{\mathbf{w}} - \sum_{j=1}^{m-1} b^{(j)}\mathbf{w}^{(j)} \bmod q \ :\ b^{(m)} \leftarrow \mathscr{D}_{\sigma_b}; b^{(1)} = 1\right]
\end{aligned}$$

Since the first element of $\mathbf{w}^{(m)}$ is guaranteed to be invertible in R_q, the above probability is at most

$$\begin{aligned}
&\max_{b' \in R} \Pr\left[b^{(m)} = b' \ :\ b^{(m)} \leftarrow \mathscr{D}_{\sigma_b}\right] \\
=&\Pr\left[b^{(m)} = 0 \ :\ b^{(m)} \leftarrow \mathscr{D}_{\sigma_b}\right] \\
=&\frac{1}{\rho_{\sigma_b}(R)}.
\end{aligned}$$

The adversary $\mathcal{X}$ makes at most Q_h queries to $\mathsf{H}'_{\mathrm{sig}}$, Q_h queries to $\mathsf{H}'_{\mathrm{non}}$, and Q_s queries to $\mathsf{OSignOn}$ (which internally calls $\mathsf{H}'_{\mathrm{non}}$ and thus $\mathsf{H}'_{\mathrm{sig}}$ as well). Since for every query to $\mathsf{H}'_{\mathrm{non}}$ and $\mathsf{OSignOn}$, there is a chance of setting the $\mathsf{bad}_{\mathrm{sig}}$ flag, by the union bound we get

$$|\mathbf{G}_4(\mathcal{X}) - \mathbf{G}_3(\mathcal{X})| \leq \Pr[\mathsf{bad}_{\mathrm{sig}}] \leq \frac{(2Q_h + Q_s)^2}{\rho_{\sigma_b}(R)} \tag{35}$$

$\mathbf{G}_5$ This game is identical to $\mathbf{G}_4$, except that $\mathsf{H}'_{\mathrm{non}}$ internally tosses a biased coin that comes out heads with probability ϖ and tails with $1 - \varpi$[11]. The additional operations are highlighted in orange in Alg. 7. If the coin comes out heads, $\mathsf{H}'_{\mathrm{non}}$ performs generation of the online output as in the previous game and then stores the challenge c in the table $\mathrm{CT}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}]$; if the coin comes out tails, $\mathsf{H}'_{\mathrm{non}}$ merely samples randomness $(r^{(j)})_{j\in[2,m]}$ and skips generation of the online output entirely. Note that, in the latter case, if the adversary $\mathcal{X}$ later makes a query to $\mathsf{OSignOn}$ with the same input, $\mathsf{OSignOn}$ fails to return an online output $(\mathbf{z}_1, \tilde{\mathbf{w}})$. If this happens, the oracle sets $\mathsf{bad}_{\mathrm{sim}}$ to true. Moreover, when the adversary returns a forgery $(\tilde{\mathbf{w}}^*, \tilde{\mathbf{z}}^*)$ on μ^* that satisfies verification conditions w.r.t. an aggregated key $\tilde{\mathbf{t}}^*$, the game aborts with $\mathsf{bad}_{\mathrm{chal}} = \mathsf{true}$ if the table entry $\mathrm{CT}[\tilde{\mathbf{w}}^*, \mu^*, \tilde{\mathbf{t}}^*]$ is defined.

Observe that the adversary $\mathcal{X}$ defines the table CT at most Q_s times, because the relevant branch inside $\mathsf{H}'_{\mathrm{non}}$ is only executed if the corresponding offline messages have been already generated by $\mathsf{OSignOff}$. Moreover, $\mathcal{X}$ makes at most Q_s queries to the online oracle $\mathsf{OSignOn}$ in which $\mathsf{bad}_{\mathrm{sim}}$ gets potentially set. Hence, $\mathbf{G}_5$ outputs 1 whenever the coin comes out heads for at most Q_s queries to $\mathsf{H}'_{\mathrm{non}}$ used by $\mathsf{OSignOn}$, *and* it comes out tail for one crucial query associated with the forgery $(\tilde{\mathbf{w}}^*, \mu^*, \tilde{\mathbf{t}}^*)$, i.e.,

$$\varpi^{Q_s} \cdot (1 - \varpi) \cdot \Pr[\mathbf{G}_4(\mathcal{X})] \leq \Pr[\mathbf{G}_5(\mathcal{X})] \tag{36}$$

By setting $\varpi = Q_s/(Q_s + 1)$, since $(1/(1 + 1/Q_s))^{Q_s} \geq 1/e$ for $Q_s \geq 0$, we get

$$\frac{1}{e(Q_s + 1)} \cdot \Pr[\mathbf{G}_4(\mathcal{X})] \leq \Pr[\mathbf{G}_5(\mathcal{X})]. \tag{37}$$

$\mathbf{G}_6$ This game is identical to $\mathbf{G}_5$, except that $\mathsf{H}'_{\mathrm{non}}$ internally proceed as in $\mathcal{S}$ of Lemma 4.4 to generate the online messages, and $\mathsf{OSignOn}$ later obtains a signature share $\mathbf{z}_1$ simulated inside $\mathsf{H}'_{\mathrm{non}}$. The additional operations are highlighted in purple in Alg. 7. Note that $\mathsf{RevSamp}$ indicates a suitable "reverse sampling algorithm" corresponding to the original Gaussian sampler Samp. In fact, once $b^{(j)}$'s are simulated, finding corresponding uniform randomness $r^{(j)}$'s are easy assuming that the Samp algorithm is "sampleable" [BCI+10]. Such a property can be for example satisfied by simple CDT-based samplers.

[11] This game hop closely follows proofs for the RSA-FDH [Cor00], mBCJ [DEF+19], and DOTT [DOTT22] schemes.

Since the adversary $\mathcal{X}$ makes at most Q_s queries to $\mathsf{OSignOff}$ and $\mathsf{OSignOn}$, and $\mathcal{X}$ observes at most Q_s simulated $(r^{(j)})_{j\in[2,m]}$'s through queries to H'_{non} (recall that H'_{non} only performs simulation if it finds the corresponding trapdoor recorded in WT), from the oracle simulation lemma (Lemma 4.4), the view of $\mathcal{X}$ in $\mathbf{G}_5$ and $\mathbf{G}_6$ is statistically indistinguishable. That is,

$$|\Pr[\mathbf{G}_6(\mathcal{X})] - \Pr[\mathbf{G}_5(\mathcal{X})]| \leq Q_s \cdot \epsilon_s \tag{38}$$

Reduction to the MS-UF-KOA game We are now ready to construct another adversary $\mathcal{A}$ breaking the MS-UF-KOA game. $\mathcal{A}$ runs $\mathcal{X}$ by simulating $\mathbf{G}_6$, but by querying the external random oracles $\mathsf{H}_{\text{agg}}, \mathsf{H}_{\text{non}}, \mathsf{H}_{\text{sig}}$ in the MS-UF-KOA game, when responding to queries to $\mathsf{H}'_{\text{agg}}, \mathsf{H}'_{\text{non}}, \mathsf{H}'_{\text{sig}}$ in $\mathbf{G}_6$, respectively. The complete reduction algorithms are described in Alg. 7 with additional operations being highlighted in blue. Note that, unless the table of simulated challenges $\mathrm{CT}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}]$ is defined, the output of $\mathsf{H}'_{\text{sig}}(\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}})$ is consistent with the external oracle $\mathsf{H}_{\text{sig}}(\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}})$. Therefore, as long as $\mathcal{X}$ wins the game $\mathbf{G}_6$, $\mathcal{A}$ can also output a forgery that is valid in the MS-UF-KOA game, i.e.,

$$\mathsf{Adv}^{\mathsf{MS\text{-}UF\text{-}KOA}}_{\mathsf{MuSig\text{-}L}}(\mathcal{A}) = \Pr[\mathbf{G}_6(\mathcal{X})]. \tag{39}$$

Putting the bounds together, we obtain the theorem statement. □

C.3 Probability that uniform $\mathbf{M} \in R_q^{k\times n}$ is not full rank

Suppose $k \leq n$ and let $\mathbf{M} \in R_q^{k\times n}$ be a uniform random matrix. We want to prove that, except with negligible probability, it is what we will call "full rank", namely, equivalent over R_q to a matrix of the form $[\mathbf{A}|\mathbb{I}_k]$ for some $\mathbf{A} \in R_q^{k\times(n-k)}$: in other words, there exists invertible matrices $\mathbf{P} \in \mathrm{GL}_k(R_q)$, $\mathbf{Q} \in \mathrm{GL}_k(R_q)$ and a matrix $\mathbf{A}$ such that $\mathbf{PMQ} = [\mathbf{A}|\mathbb{I}_k]$.

Since $q \equiv 5 \bmod 8$, the ring $R_q = R/qR$ is isomorphic to $\mathbb{F}_{q^{N/2}} \times \mathbb{F}_{q^{N/2}}$, and clearly, a matrix $\mathbf{P} \in R_q^{k\times k}$ is invertible if and only if its two components under this decomposition are invertible matrices over the field $\mathbb{F}_{q^{N/2}}$. Therefore, the matrix $\mathbf{M}$ is full rank if and only if both of its components $\mathbf{M}_0, \mathbf{M}_1 \in \mathbb{F}_{q^{N/2}}^{k\times n}$ are full rank. In particular, by the union bound, the probability that it is *not* full rank is bounded by $2p_{k,n}(q^{N/2})$, where $p_{k,n}(q^{N/2})$ is the probability that a uniformly random matrix in $\mathbb{F}_{q^{N/2}}^{k\times n}$ is not full rank. It therefore suffices to prove that this probability is negligible.

Write $Q = q^{N/2}$. Since $\mathbb{F}_Q$ is a field, a matrix $\mathbf{M}_0 \in \mathbb{F}_Q^{k\times n}$ is not full rank if and only if there exists $\mathbf{x} \in \mathbb{F}_Q^k$ a non zero vector such that $\mathbf{x}^t\mathbf{M}_0 = \mathbf{0}$. Now fix X an arbitrary subset of $\mathbb{F}_Q^k \setminus \{\mathbf{0}\}$ containing exactly one vector in each subspace of dimension 1 (i.e., one element in each equivalence class of $\mathbb{F}_Q^k \setminus \{\mathbf{0}\}$ for the relation of colinearity). Then it also holds that $\mathbf{M}_0 \in \mathbb{F}_Q^{k\times n}$ is not full rank if and only if there exists $\mathbf{x} \in X$ with $\mathbf{x}^t\mathbf{M}_0 = \mathbf{0}$. Indeed, if such an $\mathbf{x}$ exists, $\mathbf{M}_0$ is not full rank, and conversely, if $\mathbf{M}_0$ is not full rank, there exists $\mathbf{y}$ with $\mathbf{y}^t\mathbf{M}_0 = \mathbf{0}$, and we can find $\mathbf{x} \in X$ with $\mathbf{x} = \lambda\mathbf{y}$ for some $\lambda \neq 0$, which also yields $\mathbf{x}^t\mathbf{M}_0 = \mathbf{0}$.

Now, for a fixed $\mathbf{x} \in X$ and a uniformly random $\mathbf{M}_0 \in \mathbb{F}_Q^{k\times n}$, $\mathbf{x}^t\mathbf{M}_0$ is uniformly distributed in $\mathbb{F}_Q^n$. In particular, it is zero with probability exactly $1/Q^n$. Taking a union bound over all X, we get:

$$p_{k,n}(Q) \leq \sum_{\mathbf{x}\in X} \Pr_{\mathbf{M}_0}\left[\mathbf{x}^t\mathbf{M}_0 = \mathbf{0}\right] = \frac{|X|}{Q^n} = \frac{(Q^k-1)/(Q-1)}{Q^n} = \frac{1+O(1/Q)}{Q^{n-k+1}} = O(1/Q)$$

since $n - k + 1 \geq 1$. Now $1/Q = q^{-N/2}$ is negligible, so this concludes the proof.

D Correctness and Parameters

This section aims to clarify the choice of the parameters in Table 2, and serves as proof of Lemma 3.2.

Parameters are determined by the requirements of the two main techniques in this paper: generalized rejection sampling (cf. Section 3.2) and the trapdoor construction used in the simulation (cf. Section 4.3). The most complicated to compute are σ_1, σ_b, and σ_y, as the core idea of the construction relies on combining samples from the three discrete Gaussians having them as parameters. In particular, the simulation algorithm requires σ_y to be large enough to hide the trapdoor (cf. (12)), and σ_b to be large enough to allow sampling using a trapdoor (cf. Section 4.3.2) and prevent Adv from guessing the $b^{(j)}$'s

Algorithm 7: Reduction to MS-UF-KOA

$\mathbf{G}_3$ / $\mathbf{G}_4$/ $\mathbf{G}_5$ / $\mathbf{G}_6$ / $\mathcal{A}^{\mathsf{H}_{\mathrm{agg}},\mathsf{H}_{\mathrm{non}},\mathsf{H}_{\mathrm{sig}}}(\mathsf{pp},\mathbf{t}_1)$

// Procedures highlighted in green, orange, purple, blue are only executed in $\mathbf{G}_4$, $\mathbf{G}_5$, $\mathbf{G}_6$, $\mathcal{A}$ onwards, respectively.

$\mathsf{ctr} := 0$
$S := \varnothing$; $\mathcal{Q} := \varnothing$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ // Not executed in $\mathcal{A}$
$(\mathbf{t}_1, \mathbf{s}_1) \leftarrow \mathsf{Gen}()$ // Not executed in $\mathcal{A}$
$(L^*, \mu^*, (\tilde{\mathbf{w}}^*, \mathbf{z}^*)) \leftarrow \mathcal{X}^{\mathsf{OSignOff},\mathsf{OSignOn},\mathsf{H}'_{\mathrm{agg}},\mathsf{H}'_{\mathrm{non}},\mathsf{H}'_{\mathrm{sig}}}(\mathsf{pp}, \mathbf{t}_1)$
if $\mathbf{t}_1 \notin L^* \vee (L^*, \mu^*) \in \mathcal{Q}$ **then**
  return 0
$n^* := |L^*|$
$\{\mathbf{t}_1, \ldots, \mathbf{t}_{n^*}\} := L^*$
for $i \in [1, n^*]$ **do**
  $a_i := \mathsf{H}'_{\mathrm{agg}}(\langle L^* \rangle, \mathbf{t}_i)$
$\tilde{\mathbf{t}}^* := \sum_{i=1}^{n^*} a_i \mathbf{t}_i \bmod q$
$c^* := \mathsf{H}'_{\mathrm{sig}}(\tilde{\mathbf{w}}^*, \mu^*, \tilde{\mathbf{t}}^*)$
if $\bar{\mathbf{A}}\mathbf{z}^* - c^*\tilde{\mathbf{t}}^* \bmod q \neq \tilde{\mathbf{w}} \vee \|\mathbf{z}^*\|_2 > B_n$ **then**
  return 0
if $\mathrm{CT}[\tilde{\mathbf{w}}^*, \mu^*, \tilde{\mathbf{t}}^*] = c$ **then**
  $\mathsf{bad}_{\mathrm{chal}} := \mathsf{true}$
return $(L^*, \mu^*, (\tilde{\mathbf{w}}^*, \mathbf{z}^*))$
return 1

$\mathsf{OSignOff}(\mathbf{s}_1)$

$\mathsf{ctr} := \mathsf{ctr} + 1$
$\mathsf{sid} := \mathsf{ctr}$; $S := S \cup \{\mathsf{sid}\}$
$\mathbf{y}_1^{(1)} \leftarrow\!\$\ \mathscr{D}_{\sigma_1}^{\ell+k}$
For $j \in [2, m]$: $\mathbf{y}_1^{(j)} \leftarrow\!\$\ \mathscr{D}_{\sigma_y}^{\ell+k}$
For $j \in [1, m]$: $\mathbf{w}_1^{(j)} := \mathbf{A}\mathbf{y}_1^{(j)}$
$\mathbf{w}_1^{(1)} \leftarrow\!\$\ R_q^k$
$(\mathbf{w}_1^{(2)}, \ldots, \mathbf{w}_1^{(m)}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^\lambda)$
if $\exists \mathsf{sid}' \neq \mathsf{sid} : \mathrm{WT}[\mathsf{sid}'] = ((\mathbf{w}_1^{(j)})_{j\in[m]}, *)$ **then**
  $\mathsf{bad}_{\mathrm{wcol}} := \mathsf{true}$
$\mathrm{WT}[\mathsf{sid}] := ((\mathbf{w}_1^{(j)})_{j\in[m]}, (\mathbf{y}_1^{(j)})_{\in[m]})$
$\mathrm{WT}[\mathsf{sid}] := ((\mathbf{w}_1^{(j)})_{j\in[m]}, \mathbf{R})$
$\mathsf{com}_1 := (\mathbf{w}_1^{(j)})_{j\in[m]}$
if $\exists((\mathbf{t}_i, \mathsf{com}_i)_{i\neq 1}, \mu, \tilde{\mathbf{t}}) : \mathrm{HT}'_{\mathrm{non}}[\langle\{\mathbf{t}_i||\mathsf{com}_i\}_{i\in[n]}\rangle, \mu, \tilde{\mathbf{t}}] \neq \perp$ **then**
  $\mathsf{bad}_{\mathrm{non}} := \mathsf{true}$
return $(\mathbf{t}_1, \mathsf{com}_1)$

$\mathsf{OSignOn}(\mathsf{sid}, \mathsf{msgs}, \mu, (\mathsf{pk}_2, \ldots, \mathsf{pk}_n))$

if $\mathsf{sid} \notin S$ **then**
  return $\perp$
$(\mathbf{t}_i, \mathsf{com}_i)_{i\in[2,n]} := \mathsf{msgs}$
if $\langle(\mathbf{t}_i)_{i\in[2,n]}\rangle \neq \langle(\mathsf{pk}_i)_{i\in[2,n]}\rangle$ **then**
  return $\perp$
if $\exists i \geq 2 : \mathsf{pk}_i = \mathbf{t}_1$ **then**
  return $\perp$
$(\mathsf{com}_1 = (\mathbf{w}_1^{(1)}, \mathbf{w}_1^{(2)}, \ldots, \mathbf{w}_1^{(m)}), (\mathbf{y}_1^{(j)})_{j\in[m]}) := \mathrm{WT}[\mathsf{sid}]$
$(\mathsf{com}_1 = (\mathbf{w}_1^{(1)}, \mathbf{w}_1^{(2)}, \ldots, \mathbf{w}_1^{(m)}), \mathbf{R}) := \mathrm{WT}[\mathsf{sid}]$
$[w_1^{(m)}, \ldots, w_k^{(m)}]^T = \mathbf{w}^{(m)} := \sum_{i=1}^n \mathbf{w}_i^{(m)}$
if $w_1^{(m)} \notin R_q^\times$ **then**
  return $\perp$
$L := \{\mathbf{t}_1, \ldots, \mathbf{t}_n\}$
for $i \in [n]$ **do**
  $a_i := \mathsf{H}'_{\mathrm{agg}}(\langle L \rangle, \mathbf{t}_i)$
$\tilde{\mathbf{t}} := \sum_{i=1}^n a_i \mathbf{t}_i \bmod q$
$W := \{\mathbf{t}_i||\mathsf{com}_i\}_{i\in[n]}$
Call $\mathsf{H}'_{\mathrm{non}}(\langle W \rangle, \mu, \tilde{\mathbf{t}})$ with $\rho_{\mathrm{non}} = 0$
if $\mathrm{ST}[\langle W \rangle, \mu, \tilde{\mathbf{t}}] = \perp$ **then**
  $\mathsf{bad}_{\mathrm{sim}} := \mathsf{true}$
$(\mathbf{z}_1, \tilde{\mathbf{w}}) := \mathrm{ST}[\langle W \rangle, \mu, \tilde{\mathbf{t}}]$
$\mathcal{Q} := \mathcal{Q} \cup \{(L, \mu)\}$
$S := S \setminus \{\mathsf{sid}\}$
return $(\mathbf{z}_1, \tilde{\mathbf{w}})$

$\mathsf{H}'_{\mathrm{agg}}(\langle L \rangle, \mathbf{t})$

if $\mathrm{HT}'_{\mathrm{agg}}[L, \mathbf{t}] = \perp$ **then**
  $a \leftarrow\!\$\ C$
  $a := \mathsf{H}_{\mathrm{agg}}(\langle L \rangle, \mathbf{t})$
  $\mathrm{HT}'_{\mathrm{agg}}[L, \mathbf{t}] := a$
return $\mathrm{HT}'_{\mathrm{agg}}[L, \mathbf{t}]$

$\mathsf{H}'_{\mathrm{sig}}(\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}})$

if $\mathrm{HT}'_{\mathrm{sig}}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}] = \perp$ **then**
  $c \leftarrow\!\$\ C$
  $c := \mathsf{H}_{\mathrm{sig}}(\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}})$
  $\mathrm{HT}'_{\mathrm{sig}}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}] := c$
return $\mathrm{HT}'_{\mathrm{sig}}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}]$

$\mathsf{H}'_{\mathrm{non}}(\langle W \rangle, \mu, \tilde{\mathbf{t}})$

// Wlog assume the input contains honest signer's key $\mathbf{t}_1$

if $\mathrm{HT}'_{\mathrm{non}}[\langle W \rangle, \mu, \tilde{\mathbf{t}}] \neq \perp$ **then**
  return $\mathrm{HT}'_{\mathrm{non}}[\langle W \rangle, \mu, \tilde{\mathbf{t}}]$
$\{\mathbf{t}_i||\mathsf{com}_i\}_{i\in[n]} := W$
for $i \in [n]$ **do**
  $(\mathbf{w}_i^{(1)}, \ldots, \mathbf{w}_i^{(m)}) := \mathsf{com}_i$
for $j \in [m]$ **do**
  $\mathbf{w}^{(j)} := \sum_{i=1}^n \mathbf{w}_i^{(j)}$
$[w_1^{(m)}, \ldots, w_k^{(m)}]^T := \mathbf{w}^{(m)}$
$L := \{\mathbf{t}_1, \ldots, \mathbf{t}_n\}$
$a_1 := \mathsf{H}'_{\mathrm{agg}}(\langle L \rangle, \mathbf{t}_1)$
$\rho_{\mathrm{non}} \leftarrow\!\$\ [0, 1)$
if $(\nexists i \geq 2 : \mathbf{t}_i = \mathbf{t}_1) \wedge (\exists \mathsf{sid} : \mathrm{WT}[\mathsf{sid}] = (\mathsf{com}_1, *)) \wedge (w_1^{(m)} \in R_q^\times) \wedge (\mathsf{KAgg}(L) = \tilde{\mathbf{t}}) \wedge (\rho_{\mathrm{non}} \leq \varpi)$ **then**
  $(\mathbf{z}_1, \tilde{\mathbf{w}}, (r^{(j)})_{j\in[2,m]}) \leftarrow \mathsf{GenSig}(\mathsf{sid}, \mathbf{s}_1, a_1, \mu, \tilde{\mathbf{t}}, (\mathbf{w}^{(j)})_{j\in[m]})$ // Not executed in $\mathbf{G}_6$
  $(\mathbf{z}_1, \tilde{\mathbf{w}}, (r^{(j)})_{j\in[2,m]}) \leftarrow \mathsf{Sim}(\mathsf{sid}, \mathbf{t}_1, a_1, \mu, \tilde{\mathbf{t}}, (\mathbf{w}^{(j)})_{j\in[m]})$
else
  $(r^{(j)})_{j\in[2,m]} \leftarrow\!\$\ \{0,1\}^l$
  $(r^{(j)})_{j\in[2,m]} := \mathsf{H}_{\mathrm{non}}(\langle W \rangle, \mu, \tilde{\mathbf{t}})$
$\mathrm{ST}[\langle W \rangle, \mu, \tilde{\mathbf{t}}] := (\mathbf{z}_1, \tilde{\mathbf{w}})$
$\mathrm{HT}'_{\mathrm{non}}[\langle W \rangle, \mu, \tilde{\mathbf{t}}] := (r^{(j)})_{j\in[2,m]}$
return $(r^{(j)})_{j\in[2,m]}$

$\mathsf{GenSig}(\mathsf{sid}, \mathbf{s}_1, a_1, \mu, \tilde{\mathbf{t}}, (\mathbf{w}^{(j)})_{j\in[m]})$

$((\mathbf{w}_1^{(j)})_{j\in[m]}, (\mathbf{y}_1^{(j)})_{j\in[m]}) := \mathrm{WT}[\mathsf{sid}]$
$(r^{(j)})_{j\in[2,m]} \leftarrow\!\$\ \{0,1\}^l$
for $j \in [2, m]$ **do**
  $b^{(j)} := \mathsf{Samp}(r^{(j)})$
$\tilde{\mathbf{w}} := \sum_{j=1}^m b^{(j)} \mathbf{w}^{(j)} \bmod q$
$c \leftarrow\!\$\ C$
if $\mathrm{HT}'_{\mathrm{sig}}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}] \neq \perp$ **then**
  $\mathsf{bad}_{\mathrm{sig}} := \mathsf{true}$
$\mathrm{HT}'_{\mathrm{sig}}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}] := c$
$c := \mathsf{H}'_{\mathrm{sig}}(\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}})$
$\mathrm{CT}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}] := c$
$\tilde{\mathbf{y}}_1 := \sum_{j=1}^m b^{(j)} \mathbf{y}_1^{(j)}$
$\mathbf{v} := c \cdot a_1 \cdot \mathbf{s}_1$
$\mathbf{z}_1 := \mathbf{v} + \tilde{\mathbf{y}}_1$
if $\mathsf{RejSamp}(\mathbf{v}, \mathbf{z}_1, (b^{(j)})_{j\in[m]}) = 0$ **then**
  $\mathbf{z}_1 := \perp$
return $(\mathbf{z}_1, \tilde{\mathbf{w}}, (r^{(j)})_{j\in[2,m]})$

$\mathsf{Sim}(\mathsf{sid}, \mathbf{t}_1, a_1, \mu, \tilde{\mathbf{t}}, (\mathbf{w}^{(j)})_{j\in[m]})$

$((\mathbf{w}_1^{(j)})_{j\in[m]}, \mathbf{R}) := \mathrm{WT}[\mathsf{sid}]$
$c \leftarrow\!\$\ C$
$\mathbf{z}_1 \leftarrow \mathscr{D}_{\sqrt{\Sigma}}^{\ell+k}$
$\mathbf{w}'_1 := \bar{\mathbf{A}}\mathbf{z}_1 - c \cdot a_1 \cdot \mathbf{t}_1 - \mathbf{w}_1^{(1)} \bmod q$
$(b^{(2)}, \ldots, b^{(m)}) \leftarrow \mathsf{TrapSamp}(\mathbf{R}, \mathbf{w}'_1, \sigma_b)$
$(r^{(2)}, \ldots, r^{(m)}) \leftarrow \mathsf{RevSamp}(b^{(2)}, \ldots, b^{(m)})$
$\rho \leftarrow\!\$\ [0, 1]$
if $\rho > 1/M$ **then**
  $\mathbf{z}_1 := \perp$.
$\tilde{\mathbf{w}} := \sum_{j=1}^m b^{(j)} \mathbf{w}^{(j)} \bmod q$
if $\mathrm{HT}'_{\mathrm{sig}}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}] \neq \perp$ **then**
  $\mathsf{bad}_{\mathrm{sig}} := \mathsf{true}$
$\mathrm{HT}'_{\mathrm{sig}}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}] := c$
$\mathrm{CT}[\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}}] := c$
return $(\mathbf{z}_1, \tilde{\mathbf{w}}, (r^{(j)})_{j\in[2,m]})$

in advance. Applying rejection sampling to hide the secret *and the* $b^{(j)}$*'s* requires then that σ_1 is larger than σ_b, σ_y, and, of course, than the norm of the secret (cf. Lemma 3.1).

Bounding σ_b. As we already mentioned, requirements on σ_b come mostly from the security proof, as σ_b is pivotal to have only 2 rounds of interaction: it should have enough entropy to prevent an adversary to guess the final $\tilde{\mathbf{y}}$, and be large enough to allow sampling using a trapdoor. In particular, Theorem 4.5 requires σ_b to be such that $\frac{(2q_h+q_s+1)^2}{\rho_{\sigma_b}(R)} = \mathsf{negl}(\lambda)$. By [Reg09, Claim 3.8] we know that $\sigma_b \geq \eta_\varepsilon(R)$ implies $\rho_{\sigma_b}(R) = \sigma_b^N(1+\delta)$, where $\delta \leq \varepsilon$. As $N = \mathsf{poly}(\lambda)$ and $\sigma_b > 1$, it trivially holds that $\frac{(2q_h+q_s+1)^2}{\sigma_b^N(1+\delta)} = \mathsf{negl}(\lambda)$. Then, the simulation (cf. Lemma 4.4) requires that

$$\sigma_b \geq 2N \cdot q^{\frac{1}{w}+\frac{2}{Nkw}} \cdot O(\sqrt{N(kw+1)} + \sqrt{kN\log_2 q}) \ ,$$

for $w = \lceil \log_2 q \rceil$ and a parameter $m > 2k\log_2 q$ that depends on the choice of the trapdoor. As $q^{1/w} \leq 2$ and the constant hidden by the $O(\cdot)$ expression is $\approx 1/\sqrt{2\pi}$ (cf. [MP12, Lemma 2.9]), it is enough to set

$$\sigma_b := \frac{2^{5/2}}{\sqrt{\pi}} \cdot 2^{\frac{2}{Nk}} N^{3/2}\sqrt{(kw+1)} \ .$$

As Lemma 2.4 guarantees that $\eta_\varepsilon(R) \leq \sqrt{\log(2N(1+1/\varepsilon))/\pi}$, we have that $\sigma_b \geq \eta_\varepsilon(R)$. Finally, Lemma 4.4 requires that $\sigma_b > 2N \cdot q^{k/(m-1)+2/(N(m-1))}$. Observe that $q^{k/(m-1)+2/(N(m-1))} \leq 2^{1/(Nk)}\sqrt{2}$, and $2^{1/(Nk)} = 1+\nu$ for some negligible $\nu > 0$ when $N = \mathsf{poly}(\lambda)$. Hence the inequality is satisfied.

Bounding σ_y. Bounding the last two parameters requires to estimate the distribution of $\mathbf{y}$. In particular, to apply Lemma 2.7 *in the simulation* we need $\sigma_y \succ \sqrt{2} \cdot \eta_\varepsilon(\Lambda^\perp(\bar{\mathbf{A}})) \cdot \max_j \|\sqrt{b^{(j)}(b^{(j)})^*}\|$ for a random $\bar{\mathbf{A}} \in R^{(\ell+k)\times m}$; this is enough to ensure that the inequality holds for σ_1 too, as $\sigma_1 > \sigma_y$ by construction. Observe that for all j it holds that

$$\left\|\sqrt{b^{(j)}(b^{(j)})^*}\right\| = \frac{1}{\sqrt{N}}\sqrt{\sum_{i=1}^{N}\phi_i(b^{(j)})(\phi_i(b^{(j)}))^*} \leq \frac{1}{\sqrt{N}}\sqrt{N\cdot(8\sigma_b)^2} = 8\sigma_b$$

where the inequality follows from Lemma B.6 by substituting $m = 1$, $\mathbf{c} = 1$, and $r = 8\sigma_b$, and it holds with probability smaller than 2^{-195}. Set $T_b := 8\sigma_b = \frac{2^{11/2}}{\sqrt{\pi}} \cdot 2^{\frac{2}{Nk}} N^{3/2}\sqrt{(kw+1)}$. Lemma 2.5 ensures that $\eta_\varepsilon(\Lambda^\perp(\bar{\mathbf{A}})) \leq \frac{8}{\pi} q^{\frac{k}{\ell+k}}\sqrt{N\log(2(\ell+k)N(1+1/\varepsilon))}$ holds with probability $1-2^{-N}$. Now, recall that we set $\epsilon = 2^{-N}$. Hence, $\log(2(\ell+k)N(1+1/\varepsilon)) =\leq 2+N+\log((\ell+k)N)$. Thus setting

$$\sigma_y := \sqrt{2}T_b \cdot \frac{8}{\pi} q^{\frac{k}{\ell+k}}\sqrt{N\left(2+N+\log\left((\ell+k)N\right)\right)} = \frac{2^9}{\pi\sqrt{\pi}} 2^{\frac{2}{Nk}} q^{\frac{k}{\ell+k}} N^2\sqrt{(kw+1)\left(2+N+\log\left((\ell+k)N\right)\right)}$$

allows to finally apply Lemma 2.7 to show that the distribution of each $\tilde{\mathbf{y}}_i$ is statistically close to a Gaussian with covariance matrix Σ as in Eq. (2). Remark that σ_y trivially satisfies the bound $\sigma_1 > \sigma_y > 2N \cdot q^{k/(\ell+k)+2/(N(\ell+k))}$ required in Lemma 4.4.

Bounding σ_1. Now that everything else is taken care of, we can finally bound σ_1. To ensure that σ_1 is large enough (i.e., it satisfies the bound in Lemma 3.1) we need $\sigma_1 \geq \max\{\alpha T, \sigma_y\sigma_b\sqrt{Nm(\ell+k)}\}$. The norm of $c \cdot a_1 \cdot \mathbf{s}_1$ can be easily bounded as:

$$\max_{a,c\in C,\ \mathbf{s}\in S_\eta} \|ac\mathbf{s}\| \leq \sqrt{(\ell+k)N \max_{a,c\in C,\ i\in[\ell+k]} \|a\|_1^2\|c\|_1^2\|s_i\|_\infty^2} \leq \eta\kappa^2\sqrt{(\ell+k)N} =: T \ .$$

Thus $\alpha T = \alpha\eta\kappa^2\sqrt{(\ell+k)N}$.

From $\frac{\alpha T}{\sigma_y\sigma_b\sqrt{Nm(\ell+k)}} = \frac{\alpha\eta\kappa^2}{\sigma_y\sigma_b\sqrt{2kw+1}}$ it is clear that ensuring that $\sigma_1 \geq \sigma_y\sigma_b\sqrt{Nm(\ell+k)}$ is enough to satisfy the requirements of Lemma 3.1, as η, α, and κ are much smaller than the parameters. Hence we set

$$\sigma_1 := \sigma_y\sigma_b\sqrt{Nm(\ell+k)} = \frac{2^{23/2}}{\pi^2} \cdot 2^{\frac{4}{Nk}} N^4 q^{\frac{k}{\ell+k}}(kw+1)\sqrt{(2kw+1)(\ell+k)\left(2+N+\log(N(\ell+k))\right)}$$

Again, the inequality $\sigma_1 > 2N \cdot q^{k/(\ell+k)+2/(N(\ell+k))}$ needed in Lemma 4.4 is trivially satisfied.