




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

Cecilia Boschini¹ , Akira Takahashi² , and Mehdi Tibouchi³ 

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`

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.

1 Introduction

A multi-signature is a primitive that allows a group of signers holding individual key pairs $(\text{sk}_1, \text{pk}_1), \dots, (\text{sk}_n, \text{pk}_n)$ 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 [36].

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 ⁴ 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., [32]) 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 [39]). 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. [18] 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. [9] 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 [7], MuSig [31], MuSig-DN [37], mBCJ [18], and HBMS [6], 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., [17]), and therefore it is also another important design goal when constructing a multi-signature. Recently, Nick, Ruffing, and Seurin [36], and Alper and Burdges [4] 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 [16, 20, 21, 30]. All of these schemes follow the so-called *Fiat-Shamir with aborts (FSwA)* paradigm [26], 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, and FH follow the ideas of BN three-round Schnorr

⁴ Note in multi-signature every honest party behaves identically and thinks of themselves as “ P_1 ” [7]. Other parties P_2, \dots, P_n are called *co-signers*.

multi-signature, and the most recent scheme due to Damgård et al. [16] 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 $\mathbf{w}_2, \dots, \mathbf{w}_n$ 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 way to simulate (\mathbf{w}_1, c) for rejected instances, 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 [5]-based *trapdoor homomorphic commitment* $\text{Commit}(\mathbf{w}_1)$ to keep \mathbf{w}_1 secret until rejection sampling is successful, 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. 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 [34] 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.

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.⁵ “#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.

	Assumption	#Off	#On	Agg.	Note
BN [7]	DL	1	2	N	
MuSig [31]	DL	1	2	Y	
mBCJ [18]	DL	0	2	Y	
MuSig-DN [37]	DL & DDH	0	2	Y	
MuSig2 [36]	AOMDL	1	1	Y	
DWMS [4]	AGM	1	1	Y	
HBMS [6]	DL	0	2	Y	
ES [20]	DCK	1	2	N	Proof incomplete
MJ [30]	RSIS	1	2	Y	Proof incomplete
FH [21]	MLWE & rMLWE	1	2	N	Proof in QROM
DOTT [16]	MLWE & MSIS	0	2	N	TD Commitment
Our MuSig-L	MLWE & MSIS	1	1	Y	L must be a set ⁶

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: $\mathbf{sk}_i = \mathbf{s}_i$ and $\mathbf{pk}_i = \mathbf{t}_i = \bar{\mathbf{A}}\mathbf{s}_i$, where \mathbf{s}_i consists of small elements in a ring $R_q = \mathbb{Z}_q[X]/(F(X))$. 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

⁵ Although ES, MJ, and FH do not explicitly support offline-online paradigm, we conjecture the first round of these schemes can be securely preprocessed since they all follow the same blueprint of BN.

⁶ 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. The rationale behind this choice will be detailed in Section 4.5.

the random oracle $H_{\text{agg}} : \{0, 1\}^* \rightarrow C$, 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 [31] (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.

In the offline phase, parties exchange a bunch of “commit” messages $\mathbf{w}_i^{(1)}, \dots, \mathbf{w}_i^{(m)}$. 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 H_{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 H_{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 .

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 $\Lambda_{\tilde{\mathbf{u}}}^{\perp}(\tilde{\mathbf{A}})$, 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 $\Lambda_{\tilde{\mathbf{u}}}^{\perp}(\tilde{\mathbf{A}})$ has been rejected. This could potentially leak information about the secret key, a subtle issue avoided in [16] 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 $\mathbf{w}_1^{(1)}, \dots, \mathbf{w}_1^{(m)}$ with a trapdoor and 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 of the rejection sampling technique, that we believe of independent interest.

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 $\tilde{\mathbf{w}}_1 = \mathbf{A}\mathbf{z}_1 - c \cdot a_1 \cdot \mathbf{t}_1$, 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. [2] 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 in this work is that, in the lattice world, a simulator can secretly produce a *trapdoor* when creating the offline messages $\mathbf{W} := [\mathbf{w}_1^{(1)}, \dots, \mathbf{w}_1^{(m)}]$, using the gadget-based trapdoor generation algorithm of Micciancio and Peikert [34] with $m = O(k \log q)$. Once the corresponding trapdoor is known, the simulator can now sample $\mathbf{b} = [b^{(1)}, \dots, b^{(m)}]$ from a coset $\Lambda_{\tilde{\mathbf{w}}_1}^\perp(\mathbf{W})$ using a Gaussian preimage sampling for the SIS function $f_{\mathbf{W}}$. 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.

1.3 Other related work

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



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

There have been a number of results on threshold Schnorr-style signatures [22, 23, 38, 40]. However, to the best of our knowledge the most recent *two-round* schemes all rely on non-standard assumptions. For example, the modular approach to proving security of threshold and multi-signatures based on Schnorr signatures in [15] strongly relies on the AGM, while the threshold signature FROST [25] is proven secure in a non-standard heuristic which models the hash function (a public primitive) used for deriving the coefficients for the linear combination as a one-time VRF (a primitive with a secret key) in the security proof.

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 [8], or FHE [3, 11]. The multi-signature of [16] 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 [12] requires interactive aggregation, which again closely follows the BN Schnorr-based scheme. The recent aggregate signature by Boudgoust and Roux-Langlois [13] 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 $\{a, a + 1, \dots, b\}$. 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 \mathcal{D} is a probability distribution we write $s \leftarrow \mathcal{D}$ to indicate sampling s from \mathcal{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 $q \equiv 5 \pmod{8}$ and let $R_q := \mathbb{Z}_q[X]/(X^N + 1)$. An L^p -norm for a module element $\mathbf{v} \in R^m$ is given by the coefficient embedding: for $\mathbf{v} = (\sum_{i=0}^{N-1} v_{i,1} X^i, \dots, \sum_{i=0}^{N-1} v_{i,m} X^i)^T$, we define

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

The Euclidean norm of a vector $\mathbf{v} = (v_1, \dots, v_m)^T \in R^m$ in the canonical representation is defined as

$$\|\varphi(\mathbf{v})\|_2 := \frac{1}{\sqrt{N}} \cdot \sqrt{\sum_{i \in [n], j \in \mathbb{Z}_{2N}^*} |\varphi_j(v_i)|^2},$$

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 1 ([29, Corollary 1.2]). *Let $N \geq k > 1$ be powers of 2 and $q = 2k+1 \pmod{4k}$ 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 ([27, Lemma 2.2]). *Let $N > 1$ be a power of 2 and q a prime congruent to $5 \pmod{8}$. 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 $s_1(B)$ (resp., $s_m(B)$) be the largest (resp., least) singular value of B , i.e., $s_1(B) = \sup\{\|B\mathbf{v}\| : \mathbf{v} \in K_{\mathbb{R}}^m \wedge \|\mathbf{v}\| = 1\}$ (resp., $s_m(B) = \inf\{\|B\mathbf{v}\| : \mathbf{v} \in K_{\mathbb{R}}^m \wedge \|\mathbf{v}\| = 1\}$). For all \mathbf{v} , $s_m(B)\|\mathbf{v}\| \leq \|B\mathbf{v}\| \leq s_1(B)\|\mathbf{v}\|$. 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 $s_1(B) = \max_i \|\sigma_i\|$ and $s_m(B) = \min_i \|\sigma_i\|$ (the proof trivially follows from standard bounds, cf. [33]).

Lemma 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 $s_1(B^{-1}) = (s_m(B))^{-1}$.*

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 $\mathcal{D}_{\Sigma, \mathbf{c}, \Lambda}$ over a lattice $\Lambda \subseteq R^m$ with parameters \mathbf{c} and Σ is defined as

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

We denote by $\mathcal{D}_{\Sigma, \mathbf{c}}^m$ the discrete Gaussian over R^m , and omit \mathbf{c} when $\mathbf{c} = 0$. For technical reasons, *Gaussian sampling will be always be done w.r.t. the canonical representation*, even though the rest of the algorithms will use the coefficient representation. This is not an issue per se, as the canonical and coefficient embeddings are isometric, and our generalized rejection sampling technique holds for the canonical representation. One should only be careful to use the canonical

embedding whenever sampling from a Gaussian, and to immediately convert a fresh sample to the coefficient embedding.

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., $s_m(\Sigma) \geq \eta_\varepsilon(\Lambda)$) if $\eta_\varepsilon(\sqrt{\Sigma}^{-1}\Lambda) \leq 1$, i.e., if $\rho_{\sqrt{\Sigma}^{-1}\Lambda} \leq \varepsilon$. The full version of the paper contains an upper bound on the smoothing parameter of a uniformly random lattice. Throughout the paper we assume $\varepsilon = 2^{-N}$.

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 $\mathcal{D}_{\Lambda+\mathbf{u},\sqrt{\Sigma}}(\mathbf{z}) = \rho_{\sqrt{\Sigma}}(\mathbf{z})/\rho_{\sqrt{\Sigma}}(\Lambda+\mathbf{u}) \propto \rho_{\sqrt{\Sigma}}(\mathbf{z})$.

Lemma 4 ([1, 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 $s_m(\Sigma) \geq \eta_\varepsilon(\Lambda)$,

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

The following result is a direct generalization of [35, Theorem 3.3] to the ring setting. The proof is identical, but we include it in the full version for the sake of completeness.

Lemma 5. Let $\Lambda \subset R^n$ be a full-rank module lattice, $z_1, \dots, z_m \in R$ arbitrary elements, and $\sigma_1, \dots, \sigma_m \in K_{\mathbb{R}}^{++}$ satisfying $\sigma_i \succ \sqrt{2}\eta_\varepsilon(\Lambda) \cdot \max_j \|\sqrt{z_j z_j^*}\|$ for all i . Pick $\mathbf{y}_1, \dots, \mathbf{y}_m \in K_{\mathbb{R}}^n$ independently with distributions $\mathbf{y}_i \sim \mathcal{D}_{\Lambda+\mathbf{c}_i,\sigma_i}$ 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 $\mathcal{D}_{\mathcal{I}\cdot\Lambda+\mathbf{c},\sigma}$ where \mathcal{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., $\mathcal{I} = R$), the distribution of \mathbf{y} is statistically close to $\mathcal{D}_{\Lambda+\mathbf{c},\sigma}$.

2.1 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 1 (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 MSIS $_{q,k,\ell,\beta}$ assumption holds if for any probabilistic polynomial-time algorithm \mathcal{A} , the following advantage is negligible in λ .

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

Definition 2 (MLWE_{q,k,ℓ,η} 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 MLWE_{q,k,ℓ,η} assumption holds if for any probabilistic polynomial-time algorithm \mathcal{D} , the following advantage is negligible in λ .

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

2.2 Offline-online multi-signature

Following [36], we define a two-round multi-signature scheme tailored to the offline-online paradigm. A multi-signature MS consists of a tuple of algorithms (Setup, Gen, KAgg, SignOff, SignOn, Agg, Ver).

- Setup(1^λ) outputs public parameters pp . Throughout, we assume that pp is given as implicit input to all other algorithms.
- Gen() outputs a key pair (pk, sk)
- KAgg(L) takes a set of public keys $L = \{\text{pk}_1, \dots, \text{pk}_n\}$ and deterministically outputs an aggregated public key $\tilde{\text{pk}}$.
- SignOff(sk) is an offline signing algorithm that can be run independently of the message μ to sign. It outputs an offline message off and some state information st .
- SignOn($\text{st}, \text{msgs}, \text{sk}, \mu, \{\text{pk}_2, \dots, \text{pk}_n\}$) is an online signing algorithm that takes as input the state information passed on to by SignOff, offline messages $\text{msgs} = \{\text{off}_2, \dots, \text{off}_n\}$ from cosigners, a secret key sk , a message to sign μ , and cosigner’s public keys $\{\text{pk}_2, \dots, \text{pk}_n\}$. It outputs an online message on . Following the convention introduced in [7], each signer assign indices $1, \dots, n$ 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.
- Agg($\text{on}_1, \dots, \text{on}_n$) takes online messages as input, and outputs an aggregated signature σ , which might potentially contain \perp .
- Ver($\tilde{\text{pk}}, \mu, \sigma$) takes an aggregated key $\tilde{\text{pk}}$, a message μ , and a signature σ as input. It outputs 1 or 0.

Remark 1. Nick et al. [36] 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 $\text{on} = \perp$ after running SignOn), which is inherent in the FSwA-based interactive multi-signature [16] [21] [20]. 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.

Game 1: MS-COR_{MS}(λ)

```

1: pp ← Setup(1λ)
2: for i ∈ [1, n] do
3:   (pki, ski) ← Gen()
4:   for j ∈ [1, τ] do
5:     (offi,j, sti,j) ← SignOff(ski)
6:   msgsj := (off1,j, ..., offn,j)
7:   L := {pk1, ..., pkn}
8: for j ∈ [1, τ] do
9:   for i ∈ [1, n] do
10:    oni,j ← SignOn(sti,j, msgsj \ {offi,j}, ski, μ, L \ {pki})
11:   σj ← Agg(on1,j, ..., onn,j)
12: if ∃j ∈ [1, τ] : σj ≠ ⊥ then
13:   return Ver(KAgg(L), μ, σj)
14: else
15:   return 0

```

Definition 3 (MS-COR). A two-round multi-signature scheme MS has correctness error δ if

$$\Pr [0 \leftarrow MS-COR_{MS}(\lambda, n, \tau)] \leq \delta$$

where the game $MS-COR_{MS}$ 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. [18] It is also crucial for the offline sign oracle $OSignOff$ 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 $OSignOn$.

Definition 4 (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}

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

where the game $MS-UF-CMA_{MS}$ 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*).

3 Our MuSig-L Scheme

3.1 Definition of the Scheme

See [Protocol 1](#) for detailed specifications. The basic algorithms, such as $Setup$, Gen and Ver closely follow non-optimized version of the Dilithium-G signature [19]. In the offline phase each party outputs m individual “commit” messages, followed by their own public key.

Game 2: MS-UF-CMA_{MS}(\mathcal{A}, λ)

<pre> 1: pp \leftarrow Setup(1^λ) 2: (pk₁, sk₁) \leftarrow Gen() 3: ctr := 0 4: S := \emptyset; Q := \emptyset 5: (L*, μ^*, σ^*) \leftarrow $\mathcal{A}^{\text{OSignOn, OSignOff, } \mathcal{H}}$(pp, pk₁) 6: if (pk₁ \notin L*) \vee ((L*, μ^*) \in Q) then 7: return 0 8: return Ver(KAgg(L*), μ^*, σ^*) </pre>	<pre> OSignOff 1: ctr := ctr + 1 2: sid := ctr; S := S \cup {sid} 3: (off, st_{sid}) \leftarrow SignOff(sk₁) 4: return off OSignOn(sid, msgs, μ, {pk₂, ..., pk_n}) 1: if sid \notin S then return \perp 2: on \leftarrow SignOn(st_{sid}, msgs, sk₁, μ, {pk₂, ..., pk_n}) 3: L := {pk₁, ..., pk_n} 4: Q := Q \cup {(L, μ)} 5: S := S \setminus {sid} 6: return on </pre>
---	--

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 m th 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 [24]. 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 \mathbf{H}_{sig} and \mathbf{H}_{agg} : $C = \{c \in R : \|c\|_\infty = 1 \wedge \|c\|_1 = \kappa\}$. In particular, a set of differences $\tilde{C} := \{c - c' : c, c' \in C \wedge c \neq c'\}$ consists of invertible elements thanks to Lemma 1.

Finally, correctness requires $q > 16\sigma_1 n$ (where n is the number of parties, cf. Theorem 1) and $\alpha\eta\kappa^2 < \sigma_1$ (cf. Lemma 6), and $2k \lceil \log_2 q \rceil + 1 > \ell + k$ is required by security (cf. Section 4.3).

Protocol 1: MuSig-L

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

Setup(1^λ)

- 1: $\mathbf{A} \leftarrow \mathcal{R}_q^{k \times \ell}$
- 2: $\tilde{\mathbf{A}} := [\mathbf{A} \parallel \mathbb{I}_k]$
- 3: $\text{pp} := \tilde{\mathbf{A}}$
- 4: **return** pp

Gen()

- 1: $\mathbf{s}_1 \leftarrow \mathcal{S}_\eta^{\ell+k}$
- 2: $\mathbf{t}_1 := \tilde{\mathbf{A}} \mathbf{s}_1 \bmod q$
- 3: $(\text{pk}, \text{sk}) := (\mathbf{t}_1, \mathbf{s}_1)$
- 4: **return** (pk, sk)

Agg($\text{on}_1, \dots, \text{on}_n$)

- 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

KAgg(L)

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

Ver(pk, σ, μ)

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

Samp(r)

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

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: $\rho \leftarrow \mathcal{U}[0, 1]$
- 3: **if** $\rho \geq \min \left(\frac{\mathcal{D}_{\Sigma}^{\ell+k}(\mathbf{z})}{M \cdot \mathcal{D}_{\Sigma}^{\ell+k}(\mathbf{z})}, 1 \right)$ **then**
- 4: **return** 0
- 5: **return** 1

SignOff(sk₁)

- 1: $\mathbf{s}_1 := \text{sk}_1$
- 2: $\mathbf{y}_1^{(1)} \leftarrow \mathcal{D}_{\sigma_1}^{\ell+k}$
- 3: **For** $j \in [2, m] : \mathbf{y}_1^{(j)} \leftarrow \mathcal{D}_{\sigma_y}^{\ell+k}$
- 4: **For** $j \in [1, m] : \mathbf{w}_1^{(j)} := \tilde{\mathbf{A}} \mathbf{y}_1^{(j)} \bmod q$
- 5: $\text{com}_1 := (\mathbf{w}_1^{(1)}, \dots, \mathbf{w}_1^{(m)})$
- 6: $\text{off}_1 := (\mathbf{t}_1, \text{com}_1)$
- 7: $\text{st}_1 := (\mathbf{y}_1^{(1)}, \dots, \mathbf{y}_1^{(m)}, \text{com}_1)$
- 8: **return** (off₁, st₁)

SignOn(st₁, msgs, sk₁, μ , (pk₂, ..., pk_n))

- 1: $(\mathbf{t}_i, \text{com}_i)_{i \in [2, n]} := \text{msgs}$
- 2: **if** $\langle (\mathbf{t}_i)_{i \in [2, n]} \rangle \neq \langle (\text{pk}_i)_{i \in [2, n]} \rangle$ **then**
- 3: **return** \perp
- 4: **if** $\exists i \geq 2 : \mathbf{t}_i = \mathbf{t}_1$ **then**
- 5: **return** \perp
- 6: $L := \{\mathbf{t}_1, \dots, \mathbf{t}_n\}$
- 7: $a_1 := H_{\text{agg}}(\langle L \rangle, \mathbf{t}_1)$
- 8: $\tilde{\mathbf{t}} := \text{KAgg}(L)$
- 9: $W := \{\mathbf{t}_i \parallel \text{com}_i\}_{i \in [n]}$
- 10: $(r^{(j)})_{j \in [2, m]} := H_{\text{non}}(\langle W \rangle, \mu, \tilde{\mathbf{t}})$
- 11: $b^{(1)} := 1$
- 12: **For** $j \in [2, m] : b^{(j)} := \text{Samp}(r^{(j)})$
- 13: **For** $j \in [1, m] : \mathbf{w}^{(j)} := \sum_{i=1}^n \mathbf{w}_i^{(j)}$
- 14: $[w_1^{(m)}, \dots, w_k^{(m)}]^T := \mathbf{w}^{(m)}$
- 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: $c := H_{\text{sig}}(\tilde{\mathbf{w}}, \mu, \tilde{\mathbf{t}})$
- 20: $\mathbf{v} := c \cdot a_1 \cdot \mathbf{s}_1$
- 21: $\mathbf{z}_1 := \mathbf{v} + \tilde{\mathbf{y}}_1$
- 22: **if** $\text{RejSamp}(\mathbf{v}, \mathbf{z}_1, (b^{(j)})_{j \in [m]}) = 0$ **then**
- 23: $\mathbf{z}_1 := \perp$
- 24: $\text{on}_1 := (\mathbf{z}_1, \tilde{\mathbf{w}})$
- 25: **return** on₁

Table 2: Parameters for our multi-signature. Further details can be found in the full version.

Parameter	Description
n	Number of parties
τ	Number of parallel repetitions
$N = \text{poly}(\lambda)$	A power of two defining the degree of $f(X)$
$f(X) = X^N + 1$	The $2N$ -th cyclotomic polynomial
$q = 5 \pmod{8}$	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 \mathbf{c}
$C = \{c \in R : \ c\ _\infty = 1 \wedge \ c\ _1 = \kappa\}$	Challenge space where $ C = \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 6
$\sigma_1 = \sigma_x \sigma_y \sqrt{N(2kw+1)(\ell+k)}$	Standard deviation of the Gaussian distribution
$\sigma_y = \frac{\sigma^2}{\pi^{3/2}} 2^{\frac{2}{N\kappa}} q^{\frac{2}{N\kappa}} N^2 \sqrt{(kw+1)(2+N+\log((\ell+k)N))}$	Standard deviation of the Gaussian distribution
$\sigma_b = \frac{2^{2/\kappa}}{\sqrt{\pi}} \cdot 2^{\frac{2}{N\kappa}} N^{3/2} \sqrt{kw+1}$	Standard deviation of the Gaussian distribution
$\hat{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_1)$	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 H_{non}

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.

To understand the distribution of the response \mathbf{z} , we start from analyzing the distribution of the masking vector $\hat{\mathbf{y}} = \sum_{j=1}^m b^{(j)} \cdot \mathbf{y}^{(j)}$. 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:

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

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

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

Applying Lemma 5 with $b^{(j)}$ in the place of the z_i and $\mathbf{y}^{(j)}$ of y_i yields that the masking vector $\hat{\mathbf{y}} = \mathbf{y}^{(1)} + \sum_{j=2}^m b^{(j)} \cdot \mathbf{y}^{(j)}$ is distributed according to a discrete Gaussian with parameter

$$\Sigma = s \cdot \mathbb{I}_{\ell+k} \in K_{\mathbb{R}}^{(\ell+k) \times (\ell+k)}, \quad \text{where } s = \sigma_1^2 + \sigma_y^2 \cdot \sum_{j=2}^m b^{(j)*} b^{(j)} \quad (1)$$

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 [16], 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 $\Lambda_{\bar{\mathbf{u}}}^\perp(\bar{\mathbf{A}})$, where $\bar{\mathbf{u}} := \bar{\mathbf{A}} \left(\sum_j b^{(j)} \mathbf{y}^{(j)} \right) + c \cdot a \cdot \mathbf{t}$. 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 6 summarizes the rejection sampling technique used in MuSig-L; the general result can be found in the full version. 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. Essentially, we first 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 [26, Lemma 4.7] to consider the case of the behavior of a pair of distributions over a subset of their domain. 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 6 (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 $h : V \rightarrow [0, 1]$ be the composition of three probability distributions $h := \mathcal{D}_b \times \mathcal{D}_u \times \mathcal{D}_v$, where \mathcal{D}_b returns $\{1, b^{(2)}, \dots, b^{(m)}\}$ for $b^{(j)} \stackrel{\$}{\leftarrow} \mathcal{D}_{\sigma_b}$, \mathcal{D}_u returns a vector $\mathbf{u} \in R^k$, and \mathcal{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 $\widehat{\Sigma} = \text{diag}(\sigma_1^2, \dots, \sigma_1^2)$. 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:**
- $(b^{(1)}, \dots, b^{(m)}, \mathbf{u}, \mathbf{v}) \stackrel{\$}{\leftarrow} h$
 - $\mathbf{z} \stackrel{\$}{\leftarrow} \mathcal{D}_{\sqrt{\Sigma}, \mathbf{v}, \Lambda_{\bar{\mathbf{u}}}^\perp}^{\ell+k}$
 - with probability $1 - \min \left(1, \frac{\mathcal{D}_{\widehat{\Sigma}}^{\ell+k}(\mathbf{z})}{M \cdot \mathcal{D}_{\sqrt{\Sigma}, \mathbf{v}}^{\ell+k}(\mathbf{z})} \right)$, set $\mathbf{z} := \perp$
 - output $(\mathbf{z}, b^{(1)}, \dots, b^{(m)}, \mathbf{u}, \mathbf{v})$

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

- SimRS:**
- $(b^{(1)}, \dots, b^{(m)}, \mathbf{u}, \mathbf{v}) \stackrel{\$}{\leftarrow} h$

- $\mathbf{z} \xleftarrow{\$} \mathcal{D}_{\sqrt{\Sigma}, \Lambda_{\perp}^{\perp}}^{\ell+k}$
- with probability $1 - 1/M$, set $\mathbf{z} := \perp$
- output $(\mathbf{z}, b^{(1)}, \dots, b^{(m)}, \mathbf{u}, \mathbf{v})$

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

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.

Lemma 7. *The definition of the signing algorithm of MuSig-L in Protocol 1 with the parameters in Table 2 satisfies the hypotheses of Lemma 6.*

The proof of Lemma 7 is a routine calculation, thus we defer it to the full version of the paper. Observe that the statistical distance is negligible, and the probability of returning something is larger than $1/M(1 - \text{negl}(\lambda))$ as $\epsilon = 2^{-N}$ and t is set so that $\exp(-t(\pi - 1)) = 2^{-N} = \text{negl}(\lambda)$.

3.3 Correctness and Efficiency Analysis

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

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

where the game MS-COR_{MS} is described in Game 1.

Proof. The correctness game MS-COR_{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:

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

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

2. There is a collision on the public keys:

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

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 $\Pr [\text{bad}_2] \leq \frac{n(n-1)}{|S_{\eta}^k|^2} = \frac{n(n-1)}{\eta^{kN}} \leq 2^{-\text{poly}(\lambda)}$.

3. The invertibility condition is not satisfied:

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

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 5](#) 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 [\[28, Corollary 7.5\]](#) (i.e., [Lemma 8](#)) $w_1^{(m)}$ is statistically close to uniform over the entire ring, (and the same for all the signers) and [Lemma 2](#) ensures that: $\Pr[\text{bad}_3] = \frac{2}{q^{N/2}} - \frac{1}{q^N} = 2^{-\text{poly}(\lambda)}$.

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

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

[Lemma 7](#) shows that the hypotheses of [Lemma 6](#) are satisfied, thus we have: $\Pr[\text{bad}_4] \leq 1 - [\frac{1}{M} + \frac{\varepsilon + \delta_2 - \varepsilon \delta_2}{M}]^n = 1 - \frac{1}{M^n} + \text{negl}(\lambda)$.

5. The aggregated signature does not pass verification:

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

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 6](#) \mathbf{z}_i is statistically close to a Gaussian with parameter $\hat{\Sigma} = \sigma_1 \mathbb{I}_{\ell+k}$. Hence by [Lemma 4](#) we can bound the norm of \mathbf{z}_i as: $\|\mathbf{z}_i\| \leq s_1(\sqrt{\Sigma})\sqrt{N(\ell+k)} = \sigma_1\sqrt{N(\ell+k)} =: B$. Since the sum of n independent Gaussian samples with parameter σ_1 is statistically close to Gaussian with $\sqrt{n} \cdot \sigma_1$ ([Lemma 5](#)), 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](#) of the full version. 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 $\Pr[\text{bad}_5] \leq n2^{-195}$.

Putting everything together we get that

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

□

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 $\frac{1}{M}$ (cf. [Section 3.2](#)), successful signing requires

around M^n rounds, where $M = \exp(1/(2\alpha^2) + t/(2\alpha))$. Analogously to [16], 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 roughly⁷ $O(\log(n))$ 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 $c = H_{\text{sig}}(\mathbf{A}\mathbf{z} - c\mathbf{t}, \mu, \tilde{\mathbf{t}})$ instead of $\tilde{\mathbf{A}}\mathbf{z} - c\tilde{\mathbf{t}} = \tilde{\mathbf{w}}$ 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 [16]⁸.

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 [31], 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. Full security proof is deferred to the full version.

⁷ 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. As observed in [16], 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.

⁸ 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. [16, 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.

Theorem 2. *MuSig-L is MS-UF-KOA-secure under $\text{MSIS}_{q,k,\ell+1,\beta}$ and $\text{MLWE}_{q,k,\ell,\eta}$ 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*

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

Proof sketch. We first sketch the high-level ideas of proof. The complete reduction algorithms are described in the full version. 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 H_{sig} with input $\tilde{\mathbf{t}}^*$ is only made *after* the corresponding query to H_{agg} , and aborts otherwise (indicated by the bad_{agg} flag). Moreover, it guarantees that no aggregated keys collide with each other, and aborts otherwise (indicated by the bad_{ccol} flag). By the $\text{MLWE}_{q,k,\ell,\eta}$ assumption, an honestly generated public key $\mathbf{t}_1 := \mathbf{t}^* = \mathbf{A}\mathbf{s}^* \bmod q$ 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 $\text{MSIS}_{q,k,\ell+1,\beta}$ problem.

We then invoke the general forking lemma [7] twice. The first fork happens at the return value of $\text{H}_{\text{agg}} : \{0,1\}^* \rightarrow C$ (handled by the algorithm \mathcal{D} , internally running \mathcal{C}); the second fork happens at the return value of $\text{H}_{\text{sig}} : \{0,1\}^* \rightarrow C$ (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

$$\tilde{\mathbf{w}}_1 = \bar{\mathbf{A}}\mathbf{z}_1^* - c_1^* \sum_{i \neq 1} a_i \mathbf{t}_i - c_1^* a_{1,1} \mathbf{t}^* = \bar{\mathbf{A}}\hat{\mathbf{z}}_1^* - \hat{c}_1^* \sum_{i \neq 1} a_i \mathbf{t}_i - \hat{c}_1^* a_{1,1} \mathbf{t}^* \bmod q \quad (3)$$

$$\tilde{\mathbf{w}}_2 = \bar{\mathbf{A}}\mathbf{z}_2^* - c_2^* \sum_{i \neq 1} a_i \mathbf{t}_i - c_2^* a_{2,1} \mathbf{t}^* = \bar{\mathbf{A}}\hat{\mathbf{z}}_2^* - \hat{c}_2^* \sum_{i \neq 1} a_i \mathbf{t}_i - \hat{c}_2^* a_{2,1} \mathbf{t}^* \bmod q \quad (4)$$

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

$$\bar{\mathbf{A}}\bar{\mathbf{z}}_1 - \bar{c}_1 \sum_{i \neq 1} a_i \mathbf{t}_i - \bar{c}_1 a_{1,1} \mathbf{t}^* = \mathbf{0} \bmod q \quad (5)$$

$$\bar{\mathbf{A}}\bar{\mathbf{z}}_2 - \bar{c}_2 \sum_{i \neq 1} a_i \mathbf{t}_i - \bar{c}_2 a_{2,1} \mathbf{t}^* = \mathbf{0} \bmod q \quad (6)$$

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

$$\bar{\mathbf{A}}(\bar{c}_2 \bar{\mathbf{z}}_1 - \bar{c}_1 \bar{\mathbf{z}}_2) - \bar{c}_1 \bar{c}_2 \bar{a} \mathbf{t}^* = \mathbf{0}. \quad (7)$$

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 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 8 (Corollary 7.5 of [28]). *Let $F(X)$ be the $2N$ -th cyclotomic polynomial and let $R = \mathbb{Z}[X]/(F(X))$ and $R_q = R/qR$. For positive integers $k \leq n \leq \text{poly}(N)$, 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 $\mathbf{x} \leftarrow \mathcal{D}_\sigma^n$ 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 9 (Switching lemma). *Let R, N, q, k, n and σ be as in Lemma 8. Consider the following two algorithms:*

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

$$\mathcal{A}_1: \mathbf{A} \leftarrow \mathcal{R}_q^{k \times (n-k)}; \mathbf{u} \leftarrow \mathcal{D}_{\Lambda_{\bar{\mathbf{u}}}^\perp(\bar{\mathbf{A}}), \sigma}^n; \mathbf{x} \leftarrow \mathcal{D}_{\Lambda_{\bar{\mathbf{u}}}^\perp(\bar{\mathbf{A}}), \sigma}^n; \text{output } (\mathbf{A}, \mathbf{x}, \mathbf{u}).$$

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

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

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

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

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

$$\frac{\rho_\sigma(\Lambda_{\bar{\mathbf{u}}}^\perp)}{\rho_\sigma(R^n)} = \Pr_{\mathbf{x} \sim \mathcal{D}_\sigma^n} [\mathbf{u} = \bar{\mathbf{A}}\mathbf{x} \bmod q].$$

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 $\mathbf{A} \cdot \mathcal{D}_\sigma^n \bmod q$. To sum up, we have shown that for all $(\mathbf{A}, \mathbf{x}, \mathbf{u})$:

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

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

$$\begin{aligned} \Pr[(A_1, X_1, U_1) = (\mathbf{A}, \mathbf{x}, \mathbf{u})] &= \frac{1}{|R_q^{k \times (n-k)}|} \cdot \frac{1}{|R_q^k|} \cdot \mathcal{D}_{\Lambda_{\mathbf{u}}^\perp(\bar{\mathbf{A}}),\sigma}^n(\mathbf{x}) \\ &= \frac{1}{q^{Nk(n-k)}} \cdot \frac{1}{q^{Nk}} \cdot \mathcal{D}_{\Lambda_{\mathbf{u}}^\perp(\bar{\mathbf{A}}),\sigma}^n(\mathbf{x}), \end{aligned}$$

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

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

for $\mathcal{U}_{R_q^k}$ the uniform distribution on R_q^k . Now [Lemma 8](#) 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 $\Delta(H_{\mathbf{A},\sigma}(\mathbf{u}), \mathcal{U}_{R_q^k}) = 2^{-\Omega(N)}$. As a result:

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

as required. \square

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 singer 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:

$$\bar{\mathbf{A}}\mathbf{z}_1 - c \cdot a_1 \cdot \mathbf{t}_1 = \sum_{j=1}^m b^{(j)} \mathbf{w}_1^{(j)} \pmod{q}. \quad (8)$$

From our rejection sampling lemma (Lemma 6), we can indeed simulate c and \mathbf{z}_1 , and thus they already determine the sum $\tilde{\mathbf{w}}_1 := \sum_{j=1}^m b^{(j)} \mathbf{w}_1^{(j)} \pmod{q}$. However, since the offline commit messages $\mathbf{w}_1^{(j)}$ must be handed over to the adversary *before* the simulator sees adversary’s commitments $\mathbf{w}_i^{(j)}$, 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 `OSignOff` outputs $\mathbf{w}_1^{(j)}$, whenever the simulator receives a query to \mathbf{H}_{non} or the online oracle `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 (8), and then program the random oracles \mathbf{H}_{non} and \mathbf{H}_{sig} such that they output $b^{(j)}$ ’s and c , respectively.⁹ 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 $j = 1, \dots, m$. From the regularity result (Lemma 8), 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 $\mathbf{b} := [b^{(2)}, \dots, b^{(m)}]$, we take advantage of the gadget-based trapdoor (Ring-)SIS inversion algorithm of [34]. (Recall that $b^{(1)} = 1$ so we only need to sample $m - 1$ elements.) Let $\mathbf{W} := [\mathbf{w}^{(2)}, \dots, \mathbf{w}^{(m)}]$ 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 $\mathbf{g}^T = [1, 2, 4, \dots, 2^{w-1}]$ 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.

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

$$\mathbf{W} = [\bar{\mathbf{W}}|\mathbf{G} - \bar{\mathbf{W}}\mathbf{R}] \in R_q^{k \times 2kw}. \quad (9)$$

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

- `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

⁹ Note that once $b^{(j)}$ ’s are simulated, finding corresponding uniform randomness $r^{(j)}$ ’s are easy assuming that the `Samp` algorithm is “sampleable” [14]. Such a property can be for example satisfied by simple CDT-based samplers.

Gaussian $\mathcal{D}_{A_{\mathbf{w}'}^\perp(\mathbf{W}), \sigma_b}^{m-1}$ supported on the lattice coset

$$A_{\mathbf{w}'}^\perp(\mathbf{W}) := \{\mathbf{x} \in R^{2kw} : \mathbf{W} \cdot \mathbf{x} = \mathbf{w}' \pmod{q}\}. \quad (10)$$

This can be instantiated with [34, Alg. 3] or its adaptation in the module setting [10]. 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 \mathbf{W} output by TrapGen We show that m columns of the parity check matrix \mathbf{W} generated as above is indistinguishable with $[\mathbf{w}^{(2)}, \dots, \mathbf{w}^{(m)}]$ in the actual protocol. We apply the regularity lemma twice to argue that $\mathbf{w}^{(2)}, \dots, \mathbf{w}^{(m)}$ are uniform both in the actual protocol and in TrapGen, up to an negligible error.

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

$$\sigma_y > 2N \cdot q^{k/(\ell+k)+2/(N(\ell+k))} \quad (11)$$

as required by Lemma 8. 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 TrapGen. By construction, the distribution of kw columns $\bar{\mathbf{W}} = [\mathbf{w}^{(2)}, \dots, \mathbf{w}^{(kw+1)}]$ are uniform over $R_q^{k \times kw}$. As Lemma 8 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 $q = 5 \pmod{8}$, we can use Lemma 2 to argue this only happens with negligible probability (see full version for formal analysis). Hence, we can indeed apply Lemma 8 to guarantee the distribution of $\bar{\mathbf{W}} \cdot \mathbf{R}$ is statistically close to uniform over $R_q^{k \times kw}$ if

$$\bar{s} > 2N \cdot q^{1/w+2/(Nkw)}. \quad (12)$$

4.3.2 Indistinguishability of $b^{(j)}$'s output by TrapSamp To sample from spherical Gaussian with parameter σ_b , the gadget-based TrapSamp algorithm requires $\sigma_b \approx s_1(\mathbf{R}) \cdot s_1(\sqrt{\Sigma_{\mathbf{G}}})$ [34, §5.4] where $\sqrt{\Sigma_{\mathbf{G}}}$ is a parameter used when performing Gaussian sampling from a coset $A_{\mathbf{w}'}^\perp(\mathbf{G})$. As $\Sigma_{\mathbf{G}}$ is a constant, we only need to evaluate $s_1(\mathbf{R})$, which is $\bar{s} \cdot O(\sqrt{Nkw} + \sqrt{Nk \log_2 q})$ from [34, §5.2]. Together with the condition (12) 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,

Algorithm 1: Simulation of honest signing algorithm

$\mathcal{T}(\bar{\mathbf{A}}, a, \mathbf{s}, \mathbf{t})$ // Offline 1: for $j \in [1, m]$ do 2: if $j = 1$ then 3: $\mathbf{y}^{(1)} \leftarrow \mathcal{D}_{\sigma_1}^{\ell+k}$ 4: $b^{(1)} := 1$ 5: else 6: $\mathbf{y}^{(j)} \leftarrow \mathcal{D}_{\sigma_y}^{\ell+k}$ 7: $b^{(j)} \leftarrow \mathcal{D}_{\sigma_b}$ 8: $\mathbf{w}^{(j)} := \bar{\mathbf{A}}\mathbf{y}^{(j)} \bmod q$ 9: $\tilde{\mathbf{y}} := \sum_{j=1}^m b^{(j)}\mathbf{y}^{(j)}$ // Online 10: $c \leftarrow \mathcal{C}$ 11: $\mathbf{v} := c \cdot a \cdot \mathbf{s}$ 12: $\mathbf{z} := \mathbf{v} + \tilde{\mathbf{y}}$ 13: $\rho \leftarrow [0, 1]$ 14: if $\rho > \min \left(\frac{\mathcal{D}_{\sqrt{\hat{\Sigma}}}^{\ell+k}(\mathbf{z})}{M \cdot \mathcal{D}_{\sqrt{\hat{\Sigma}, \mathbf{v}}}^{\ell+k}(\mathbf{z})}, 1 \right)$ then 15: $\mathbf{z} := \perp$ 16: return $(\bar{\mathbf{A}}, a, \mathbf{t}, (\mathbf{w}^{(j)}, b^{(j)})_{j \in [m]}, c, \mathbf{z})$	$\mathcal{S}(\bar{\mathbf{A}}, a, \mathbf{t})$ 1: $\mathbf{w}^{(1)} \leftarrow \mathcal{R}_q^k$ 2: $([\mathbf{w}^{(2)}, \dots, \mathbf{w}^{(m)}], \mathbf{R}) \leftarrow \text{TrapGen}(1^\lambda)$ 3: $\mathbf{z} \leftarrow \mathcal{D}_{\sqrt{\hat{\Sigma}}}^{\ell+k}$ 4: $c \leftarrow \mathcal{C}$ 5: $\mathbf{w}' := \bar{\mathbf{A}}\mathbf{z} - c \cdot a \cdot \mathbf{t} - \mathbf{w}^{(1)}$ 6: $b^{(1)} := 1$ 7: $(b^{(2)}, \dots, b^{(m)}) \leftarrow \text{TrapSamp}(\mathbf{R}, \mathbf{w}', \sigma_b)$ 8: $\rho \leftarrow [0, 1]$ 9: if $\rho > 1/M$ then 10: $\mathbf{z} := \perp$ 11: return $(\bar{\mathbf{A}}, a, \mathbf{t}, (\mathbf{w}^{(j)}, b^{(j)})_{j \in [m]}, c, \mathbf{z})$
--	--

we need the following lemma. It can be proved via standard hybrid arguments, by invoking the switching lemma multiple times, indistinguishability of `TrapGen` and `TrapSamp` as stated above, and our generalized rejection sampling lemma ([Lemma 6](#)). Conditions on the parameters are detailed in the full version.

Lemma 10. *Let $\sigma_1, \sigma_y, \sigma_b, \Sigma, \hat{\Sigma}, M$ be parameters satisfying conditions in [Lemma 6](#) and [Section 4.3](#). Suppose $q = 5 \bmod 8$ as in [Lemma 2](#). Let $\mathbf{A} \leftarrow \mathcal{R}^{k \times \ell}$, $\bar{\mathbf{A}} := [\mathbf{A} | \mathbb{I}_k]$, $\mathbf{s} \in \mathcal{S}_\eta^{\ell+k}$, $\mathbf{t} := \bar{\mathbf{A}}\mathbf{s}$, $a \in \mathcal{C}$. The output distributions of \mathcal{T} and \mathcal{S} in [Alg. 1](#) are statistically indistinguishable.*

Proof. We prove via standard hybrid arguments. Each hybrid is detailed in the full version.

- 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 $\Lambda_{\mathbf{w}^{(j)}}^\perp(\bar{\mathbf{A}}) = \{\mathbf{x} \in \mathbb{R}^{k+\ell} : \bar{\mathbf{A}}\mathbf{x} = \mathbf{w}^{(j)} \bmod q\}$. From [Lemma 9](#), 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 $\Lambda_{\tilde{\mathbf{w}}}^\perp(\bar{\mathbf{A}})$, where $\tilde{\mathbf{w}} = \sum_j b^{(j)}\mathbf{w}^{(j)} \bmod q$. From [Lemma 5](#), 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 $\Lambda_{\mathbf{u}}^\perp$ 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.

- Hyb₄ is identical to Hyb₃, 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 6, Hyb₃ and Hyb₄ are statistically close.
- Hyb₅ is identical to Hyb₄, except that $\mathbf{w}' = \tilde{\mathbf{w}} - \mathbf{w}^{(1)}$ is uniformly sampled from R_q^k and a vector $[b^{(2)}, \dots, b^{(m)}]$ is sampled from spherical Gaussian over a coset $\Lambda_{\mathbf{w}'}^\perp(\mathbf{W})$, where $\mathbf{W} = [\mathbf{w}^{(2)}, \dots, \mathbf{w}^{(m)}]$. From Lemma 9, Hyb₄ and Hyb₅ are statistically close.
- Hyb₆ is identical to Hyb₅, except that \mathbf{z} is sampled from Gaussian over $R_q^{\ell+k}$ and $\tilde{\mathbf{w}}$ is defined as $\tilde{\mathbf{w}} = \bar{\mathbf{A}}\mathbf{z} - c \cdot a \cdot \mathbf{t}$. From Lemma 9, Hyb₅ and Hyb₆ are statistically close.
- Hyb₇ is identical to Hyb₆, except that a matrix $[\mathbf{w}^{(2)}, \dots, \mathbf{w}^{(m)}]$ is generated with the corresponding trapdoor \mathbf{R} . From indistinguishability of the TrapGen algorithm, Hyb₆ and Hyb₇ are statistically close.
- Hyb₈ is identical to Hyb₇, except that a vector $[b^{(2)}, \dots, b^{(m)}]$ is sampled using the trapdoor sampling algorithm. From indistinguishability of the TrapSamp algorithm, Hyb₇ and Hyb₈ are statistically close.

Note that the distribution output by Hyb₈ is identical to one by \mathcal{S} . This concludes the proof. \square

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 3. *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 OSignOff and OSignOn, there exists PPT adversary \mathcal{A} such that*

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

where ϵ_s is determined by the statistical distance of Lemma 10.

Proof sketch We sketch the high-level ideas. Full security proof is deferred to the full version. We denote by $H'_{\text{agg}}, H'_{\text{non}}, H'_{\text{sig}}$ (resp. $H_{\text{agg}}, H_{\text{non}}, H_{\text{sig}}$) the random oracles in the MS-UF-CMA game (resp. MS-UF-KOA game), respectively.

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

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

- H'_{non} : Whenever it receives a query of the form $(\{\mathbf{t}_i \mid \text{com}_i\}_{i \in [n]}, \mu, \tilde{\mathbf{t}})$, it first makes sure that (1) there is no duplicate honest keys in the input, (2) the m th sum of commit message contains an invertible element, and (3) $\text{com}_1 = [\mathbf{w}_1^{(j)}, \dots, \mathbf{w}_1^{(m)}]$ (i.e., a commit message appended to the honest party's key \mathbf{t}_1) has been previously produced by 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, H'_{non} internally performs simulation following the procedures of Alg. 1, and then programs outputs of the random oracles H'_{sig} and H'_{non} accordingly. A simulated signature is finally stored in the table ST.
- OSignOn : When the online oracle is queried, it always invokes H'_{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 H'_{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 H'_{non} . If that happens, the output of the external RO H_{sig} is not consistent with that of H'_{sig} anymore, and thus the reduction cannot win the MS-UF-KOA game. However, this issue can be handled by having H'_{non} perform simulation only probabilistically, a proof technique similar to [18] and [16]. 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 offline-online paradigm. Consider a modified scheme where H'_{non} does not take individual public keys, i.e., it simply derives randomness via $H'_{\text{non}}(\langle \text{com}_i \rangle_{i \in [n]}, \mu, \tilde{\mathbf{t}})$. It is easy to see that the simulator would have a hard time looking up the right trapdoor to perform simulation: say OSignOff has produced $(\text{com}_1, \mathbf{R})$ in session 1 and $(\text{com}'_1, \mathbf{R}')$ in session 2, respectively. Now, if the adversary queries H'_{non} with input $(\langle \text{com}_1, \text{com}'_1 \rangle, \mu, \tilde{\mathbf{t}})$ 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 OSignOn in session 2 with μ and com_1 (by maliciously claiming com_1 to be adversary's offline commit), a signature previously simulated by H'_{non} is clearly invalid. Essentially the same issue happens if \mathbf{t}_1 occurs multiple times in the key list L .

Acknowledgment The authors are grateful to Claudio Orlandi for discussions in the earlier stages of this work. We thank Carsten Baum, Katharina Boudgoust, and Mark Simkin for helpful comments and discussions. Cecilia Boschini has been supported by the Università della Svizzera Italiana under the SNSF project number 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

1. Agrawal, S., Gentry, C., Halevi, S., Sahai, A.: Discrete Gaussian leftover hash lemma over infinite domains. In: ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 97–116. Springer, Heidelberg
2. Agrawal, S., Kirshanova, E., Stehlé, D., Yadav, A.: Can round-optimal lattice-based blind signatures be practical? IACR Cryptol. ePrint Arch. p. 1565
3. Agrawal, S., Stehle, D., Yadav, A.: Round-optimal lattice-based threshold signatures, revisited. Cryptology ePrint Archive, Paper 2022/634
4. Alper, H.K., Burdges, J.: Two-round trip schnorr multi-signatures via delinearized witnesses. In: CRYPTO 2021, Part I. LNCS, vol. 12825, pp. 157–188. Springer, Heidelberg, Virtual Event
5. Baum, C., Damgård, I., Lyubashevsky, V., Oechsner, S., Peikert, C.: More efficient commitments from structured lattice assumptions. In: SCN 18. LNCS, vol. 11035, pp. 368–385. Springer, Heidelberg
6. Bellare, M., Dai, W.: Chain reductions for multi-signatures and the HBMS scheme. In: ASIACRYPT 2021, Part IV. LNCS, vol. 13093, pp. 650–678. Springer, Heidelberg
7. Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: ACM CCS 2006. pp. 390–399. ACM Press
8. Bendlin, R., Krehbiel, S., Peikert, C.: How to share a lattice trapdoor: Threshold protocols for signatures and (H)IBE. In: ACNS 13. LNCS, vol. 7954, pp. 218–236. Springer, Heidelberg
9. Benhamouda, F., Lepoint, T., Loss, J., Orrù, M., Raykova, M.: On the (in)security of ROS. In: EUROCRYPT 2021, Part I. LNCS, vol. 12696, pp. 33–53. Springer, Heidelberg
10. Bert, P., Eberhart, G., Prabel, L., Roux-Langlois, A., Sabt, M.: 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. Lecture Notes in Computer Science, vol. 12841, pp. 195–214. Springer
11. Boneh, D., Gennaro, R., Goldfeder, S., Jain, A., Kim, S., Rasmussen, P.M.R., Sahai, A.: Threshold cryptosystems from threshold fully homomorphic encryption. In: CRYPTO 2018, Part I. LNCS, vol. 10991, pp. 565–596. Springer, Heidelberg
12. Boneh, D., Kim, S.: One-time and interactive aggregate signatures from lattices. preprint
13. Boudgoust, K., Roux-Langlois, A.: Compressed linear aggregate signatures based on module lattices. IACR Cryptol. ePrint Arch. p. 263
14. Brier, E., Coron, J.S., Icart, T., Madore, D., Randriam, H., Tibouchi, M.: Efficient indifferentiable hashing into ordinary elliptic curves. In: CRYPTO 2010. LNCS, vol. 6223, pp. 237–254. Springer, Heidelberg
15. Crites, E.C., Komlo, C., Maller, M.: How to prove schnorr assuming schnorr: Security of multi- and threshold signatures. IACR Cryptol. ePrint Arch. p. 1375
16. Damgård, I., Orlandi, C., Takahashi, A., Tibouchi, M.: Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices. In: PKC 2021, Part I. LNCS, vol. 12710, pp. 99–130. Springer, Heidelberg
17. Damgård, I., Pastro, V., Smart, N.P., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: CRYPTO 2012. LNCS, vol. 7417, pp. 643–662. Springer, Heidelberg

18. Drijvers, M., Edalatnejad, K., Ford, B., Kiltz, E., Loss, J., Neven, G., Stepanovs, I.: On the security of two-round multi-signatures. In: 2019 IEEE Symposium on Security and Privacy. pp. 1084–1101. IEEE Computer Society Press
19. Ducas, L., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS - dilithium: Digital signatures from module lattices. *IACR Cryptol. ePrint Arch.* p. 633
20. El Bansarkhani, R., Sturm, J.: An efficient lattice-based multisignature scheme with applications to bitcoins. In: CANS 16. LNCS, vol. 10052, pp. 140–155. Springer, Heidelberg
21. Fukumitsu, M., Hasegawa, S.: A lattice-based provably secure multisignature scheme in quantum random oracle model. In: ProvSec 2020. LNCS, vol. 12505, pp. 45–64. Springer, Heidelberg
22. Garillot, F., Kondi, Y., Mohassel, P., Nikolaenko, V.: Threshold Schnorr with stateless deterministic signing from standard assumptions. In: CRYPTO 2021, Part I. LNCS, vol. 12825, pp. 127–156. Springer, Heidelberg, Virtual Event
23. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Secure distributed key generation for discrete-log based cryptosystems. *Journal of Cryptology* **20**(1), 51–83
24. Howe, J., Prest, T., Ricosset, T., Rossi, M.: Isochronous gaussian sampling: From inception to implementation. In: Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020. pp. 53–71. Springer, Heidelberg
25. Komlo, C., Goldberg, I.: 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. Lecture Notes in Computer Science, vol. 12804, pp. 34–65. Springer
26. Lyubashevsky, V.: Lattice signatures without trapdoors. In: EUROCRYPT 2012. LNCS, vol. 7237, pp. 738–755. Springer, Heidelberg
27. Lyubashevsky, V., Neven, G.: One-shot verifiable encryption from lattices. In: EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 293–323. Springer, Heidelberg
28. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-LWE cryptography. In: EUROCRYPT 2013. LNCS, vol. 7881, pp. 35–54. Springer, Heidelberg
29. Lyubashevsky, V., Seiler, G.: Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In: EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 204–224. Springer, Heidelberg
30. Ma, C., Jiang, M.: Practical lattice-based multisignature schemes for blockchains. *IEEE Access* **7**, 179765–179778
31. Maxwell, G., Poelstra, A., Seurin, Y., Wuille, P.: Simple schnorr multi-signatures with applications to bitcoin. *Des. Codes Cryptogr.* **87**(9), 2139–2164
32. Micali, S., Ohta, K., Reyzin, L.: Accountable-subgroup multisignatures: Extended abstract. In: ACM CCS 2001. pp. 245–254. ACM Press
33. Micciancio, D.: 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
34. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg
35. Micciancio, D., Peikert, C.: Hardness of SIS and LWE with small parameters. In: CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 21–39. Springer, Heidelberg
36. Nick, J., Ruffing, T., Seurin, Y.: MuSig2: Simple two-round Schnorr multi-signatures. In: CRYPTO 2021, Part I. LNCS, vol. 12825, pp. 189–221. Springer, Heidelberg, Virtual Event

37. Nick, J., Ruffing, T., Seurin, Y., Wuille, P.: MuSig-DN: Schnorr multi-signatures with verifiably deterministic nonces. In: ACM CCS 2020. pp. 1717–1731. ACM Press
38. Nicolosi, A., Krohn, M.N., Dodis, Y., Mazières, D.: Proactive two-party signatures for user authentication. In: NDSS 2003. The Internet Society
39. Ristenpart, T., Yilek, S.: The power of proofs-of-possession: Securing multiparty signatures against rogue-key attacks. In: EUROCRYPT 2007. LNCS, vol. 4515, pp. 228–245. Springer, Heidelberg
40. Stinson, D.R., Strobl, R.: Provably secure distributed Schnorr signatures and a (t, n) threshold scheme for implicit certificates. In: ACISP 01. LNCS, vol. 2119, pp. 417–434. Springer, Heidelberg