A Detailed Analysis of Fiat-Shamir with Aborts

Julien Devevey[1], Pouria Fallahpour[1], Alain Passelègue[1,2], and Damien Stehlé[1,3]

[1] ENS de Lyon, Lyon, France
[2] Inria, Lyon, France
[3] Institut Universitaire de France, Paris, France

Abstract. Lyubashevky's signatures are based on the Fiat-Shamir with Aborts paradigm. It transforms an interactive identification protocol that has a non-negligible probability of aborting into a signature by repeating executions until a loop iteration does not trigger an abort. Interaction is removed by replacing the challenge of the verifier by the evaluation of a hash function, modeled as a random oracle in the analysis. The access to the random oracle is classical (ROM), resp. quantum (QROM), if one is interested in security against classical, resp. quantum, adversaries. Most analyses in the literature consider a setting with a bounded number of aborts (i.e., signing fails if no signature is output within a prescribed number of loop iterations), while practical instantiations (e.g., Dilithium) run until a signature is output (i.e., loop iterations are unbounded).

In this work, we emphasize that combining random oracles with loop iterations induces numerous technicalities for analyzing correctness, runtime, and security of the resulting schemes, both in the bounded and unbounded case. As a first contribution, we put light on errors in all existing analyses. We then provide two detailed analyses in the QROM for the bounded case, adapted from Kiltz *et al.* [EUROCRYPT'18] and Grilo *et al.* [ASIACRYPT'21]. In the process, we prove the underlying Σ-protocol to achieve a stronger zero-knowledge property than usually considered for Σ-protocols with aborts, which enables a corrected analysis. A further contribution is a detailed analysis in the case of unbounded aborts, the latter inducing several additional subtleties.

Keywords: Fiat-Shamir with aborts, Lyubashevsky's signature, QROM

1 Introduction

The Fiat-Shamir heuristic [FS86] transforms a public-coin interactive proof system into a digital signature, by replacing the public coins of the verifier with hash function evaluations. In the random oracle model (ROM), the publicly available hash function is modeled as a uniform function, which the adversary is given (classical) access to. One of the most famous instances of the Fiat-Shamir heuristic is Schnorr's signature [Sch89], whose security relies on the discrete logarithm problem (heuristically, in the ROM). If considering quantum adversaries, two adaptations are required: first, the discrete logarithm hardness assumption must be replaced by another one that is conjectured to be quantum-resistant;

second, the adversary should be granted quantum access to the random oracle (QROM) as it can query the hash function in quantum superposition.

In [Lyu09, Lyu12], Lyubashevsky proposed a lattice-based signature scheme that is reminiscent of Schnorr's. A key difference is that the underlying interactive proof system has a non-negligible probability of aborting. Aborting allows to make the signature distribution independent of the signing key and is necessary to avoid attacks against the signature schemes (see [ASY22, Section 4.1]). To handle the aborts, the protocol execution is repeated within a loop, until no abort occurs in the current loop iteration. Similarly to the Fiat-Shamir heuristic, one may replace the non-final verifier steps by hash function evaluations and model the hash function as a random oracle: this technique is referred to as Fiat-Shamir with aborts.

The combination of the Fiat-Shamir heuristic and rejection sampling leads to several difficulties when analyzing the resulting signature scheme. Most analyses of Lyubashevsky's signatures consider a variant that we refer to as Fiat-Shamir with Bounded Aborts (FSwBA). In this variant, the number of loop iterations is a priori bounded by a parameter B of the scheme. If no non-aborting iteration is encountered within this bounded number B of loop iterations, the signing algorithm fails (the failure symbol $\perp$ is output). With FSwBA, the runtime analysis is trivial. In the security proof, the upper bound on the number of iterations is technically convenient as it provides a bound on how many random oracle values are being programmed by the challenger, which eases the analysis of the random oracle programming impact on the adversary's view. The most detailed security analyses are provided in [AFLT16] for the ROM, and in [KLS18] for the QROM. An alternative proof strategy in the QROM is suggested in [GHHM21], but not detailed. In concrete instantiations of Fiat-Shamir with aborts, such as the Dilithium signature scheme [DKL$^+$18], the signing algorithm typically does not enforce any upper bound on the number of loop iterations. We call this variant Fiat-Shamir with Unbounded Aborts (FSwUA). It is more difficult to analyze, as arbitrarily many hash values may be programmed by the challenger in the security proof.

Contributions. Our first set of contributions relates to FSwBA. First, we explain below that all existing security analyses of FSwBA contain a subtle common flaw, with additional errors in the QROM analysis of [KLS18]. We then provide two security analyses of FSwBA in the QROM, the first one by correcting the one from [KLS18], and the second by adapting the approach suggested in [GHHM21]. As detailed below, it turns out that these QROM analyses are incomparable. In the process, we prove that the underlying Σ-protocol achieves a stronger notion of zero-knowledgeness than usually considered for Σ-protocols with aborts, which enables the proofs. Still for FSwBA, as far as we are aware of, there is no detailed correctness analysis in the literature: this is actually not trivial, and we provide a detailed correctness analysis.

Our second set of results concerns FSwUA. On the negative side, we exhibit an interactive proof system such that applying FSwUA to it leads to a signature scheme such that:

- for all signing keys, with non-zero probability over the random oracle randomness, signing loops forever for all messages; in particular, the expected signing runtime is infinite;
- with overwhelming probability over the random oracle randomness, for all messages and all signing keys, the expected runtime of signing over its own randomness is below a fixed polynomial.

This suggests a modification of the signing efficiency requirement, in which the runtime expectation is not taken over the randomness of the random oracle, but should be bounded by a polynomial with overwhelming probability over the randomness of the random oracle. On the positive side, we give analyses of correctness, signing efficiency (with respect to the modified definition) and security for FSwUA in the QROM (with a tighter reduction in the ROM).

Finally, as a side contribution, we generalize our analysis to rely on a Σ-protocol whose simulator's quality is measured in terms of the Rényi divergence (rather than the statistical distance) for non-aborting transcripts. As pointed out in [DFPS22], in the case of Lyubashevsky's signature with Gaussians [Lyu12], when the signature is replaced with the non-aborting simulator in the security proof, the analysis based on the divergence provides security for a larger range of parameters. This notably allows to decrease the standard deviation of the distribution of the signature and hence the signature size.

2 Technical Overview

We focus on analyzing the Fiat-Shamir with aborts transform in the context of digital signatures. Our techniques also allow to transform a constant-round public-coin interactive proof system into a non-interactive one, and most of our results carry over to this setup (a notable exception being the results exploiting the Rényi divergence simulation mentioned above). We specifically consider how this technique allows the challenger to simulate replies to sign queries without knowing the signing key (which is made possible by allowing the challenger to program the random oracle). More formally, we are interested in reducing the signature unforgeability under chosen message attacks (CMA) to its unforgeability under no-message attacks (NMA). How to obtain NMA security is beyond the scope of this work, and can be handled in different ways (see, e.g., [Lyu09, Lyu12, AFLT16, DFMS19]).

In order to fix notation, we refer to the prover's first message in the underlying Σ-protocol as the commitment w, and a transcript is a triple (w, c, z) where c is a uniformly random challenge. After applying the Fiat-Shamir transform, the challenge c is then replaced by a hash value $H(w\|\mu)$, with μ being the signed message. An adversary against the CMA security of a digital signature in the random oracle model is allowed to make two types of queries: sign queries and hash queries (the latter queries being classical in the ROM and in quantum superposition in the QROM). In the security analysis, we eventually want the challenger to be able to reply to the queries without relying on the signing key. For this purpose, we can let the challenger modify the way it replies to the queries, as long as the modifications are not visible to the adversary. The fact that the signing algorithm uses the hash function, which is controlled by the challenger, is handful for simulating signatures without knowing the signing key, but induces a difficulty: the challenger must reply to hash queries and sign queries consistently. In the case of Schnorr's signature, a sign query uses the hash function exactly once. However, in the Fiat-Shamir with Aborts variant, a sign query uses the hash function several times: the hash function is evaluated once in every loop iteration.

In what follows, we first describe flaws from existing analyses in the bounded abort setting, and then how we fix them. In the process, we introduce a stronger zero-knowledge definition for Σ-protocols with aborts, which allows to fix the analysis, and prove that existing protocols achieve this definition. We finally explain how our analysis extends to the case of unbounded aborts.

2.1 Flaws in Existing Analyses of FSwBA

An unsubstantiated intuition. We start by describing a first flaw appearing in all existing analyses. These analyses start as follows: in the genuine security experiment (denoted Game 0), all (successful or not) transcripts generated during a sign query use a challenge that is computed with the hash function. Then, a first hybrid (Game 1) changes the sign algorithm by sampling a uniformly random challenge and programming the hash function consistently with the successful proof transcript *only.* All proofs immediately conclude these two games are identical: the (unsubstantiated) intuition is that the adversary does not have access to the aborted transcripts, and hence programming these transcripts does not impact the adversary's view.

F1. Assume the challenger in the genuine CMA (or even CMA_1) security game answers a sign query μ using a sequence of commitments $w_1, w_2, \ldots$. Assume that rejecting is a deterministic function of w and c (this is for example the case for Lyubashevsky's signatures with the parameters considers in [AFLT16]). Then, as soon as w_1 fails to produce a valid transcript, the hash value $H(w_1 \| \mu)$ is fixed and the sign oracle can no longer return a valid signature which uses commitment w_1. This is not the case in Game 1, since the hash value $H(w_1 \| \mu)$ is not programmed by the failed attempt, and the sign query could return a signature (w_1, c', z') for $c' \neq c$.

FSwBA has been analyzed and used numerous times (we focus here on the most detailed analyses), yet the above flaw **F1** appears in [Lyu12, Lemma 5.3], [Lyu16, Lemma 4.1], [KLS18, Theorem 3.2], and [Kat21, Lemma 4.6]. It also appears in [AFLT16] though not in Game 1 but in Game 0: in the proof of [AFLT16, Theorem 1], the authors directly start with the above Game 1 rather than with the correct Game 0. Finally, the difficulty with the hash function inconsistencies seems identified in [ABB$^+$17, Appendix B.4], but the authors do not handle the case of inconsistencies between different sign queries for the same message.

The fact that the adversary can make hash queries on superpositions of all inputs in the QROM makes it even more difficult to argue that the adversary cannot detect random oracle programmings, which induces additional errors.

Correlated challenges and implicit quantum sign queries in the QROM. To avoid the latter difficulty in the QROM, the reduction from [KLS18] is made history-free, i.e., the random oracle is never reprogrammed during the execution of the security game (see [BDF$^+$11] for a general treatment of history-free reductions in the QROM). For this purpose, the authors let the hash function call the signing algorithm, to guarantee that the hash and sign queries are handled consistently. On the downside, any subsequent signing algorithm modification in the security proof is of a quantum nature, as the adversary can make quantum queries to the hash function. This leads the analysis to two additional errors.

Consider the CMA$_1$ security analysis [KLS18, Theorem 3.2]: as above, the reduction starts with Game 0, which is the genuine security experiment. In Game 1, on a hash query $(w\|\mu)$, the oracle calls a $\mathsf{GetTrans}$ function which runs the signing algorithm on input μ and checks if the commitment w of the *non-aborting* transcript matches the hash query. If the random oracle is called on that input (possibly as part of a quantum superposition), it is programmed to reply with the challenge programmed by the signing algorithm. This guarantees consistencies of hash values defined by hash queries and by sign queries. In addition, Game 1 replaces the Σ-protocol execution in the $\mathsf{GetTrans}$ function (called in both sign and hash queries) by the simulator. The authors bound the advantage loss of that game hop by $BQ_S\varepsilon_{zk}$, with B being the maximum number of loop iterations, Q_S the number of sign queries, and ε_{zk} the zero-knowledge error of the underlying interactive protocol.

As pinpointed above, a first flaw **F1** comes from the fact *only the non-aborting* challenge is programmed by $\mathsf{GetTrans}$, but two additional flaws are induced by relying on the simulator in $\mathsf{GetTrans}$.

F2. Recall that the zero-knowledge property of the underlying Σ-protocol is for a single execution of the protocol (as opposed to correlated executions). Hence, replacing executions which rely on challenges computed as hash values by simulated transcripts requires challenges to be statistically independent. This is only possible if the hash function is evaluated on distinct inputs $(w\|\mu)$, which is not guaranteed: there might be collisions on commitments w's used within a sign query for a message μ.

F3. Since the adversary can only make classical sign queries, it could seem that transitioning from real to simulated transcripts is required only for those that are generated by the sign queries (there are at most BQ_S of them, leading to the $BQ_S\varepsilon_{zk}$ term). However, the adversary can make quantum hash queries, and for consistency of hash evaluation, these queries make calls to the $\mathsf{GetTrans}$ function. Hence this transition has to be done for all possible sign queries (not only those that are actually made). In particular, even in the ROM, the reduction loss should already be $B(Q_S+Q_H)\varepsilon_{zk}$ as each sign query and hash query induces up to B simulated transcripts. In the QROM, the loss is even larger as the adversary can make Q_H quantum hash queries.

Flaws **F2** and **F3** appear in the QROM analyses of [KLS18, Theorems 3.2 and 3.3] and [Kat21, Lemma 4.6].

2.2 Corrected Analyses of FSwBA

The security analysis in the ROM, which only suffers from **F1**, can be readily modified to handle this difficulty by bounding the probability the random oracle gets evaluated twice on a previously defined input. If the commitment has high min-entropy, this event happens with negligible probability. **F1** vanishes as the hash function is never evaluated twice on the same input. It would seem that the rest of the analysis goes through (e.g., following [AFLT16]), but this fix induces an additional problem described below.

We provide two different analyses in the QROM. The first one follows and fixes the [KLS18] analysis. The second one extends the adaptive reprogramming technique of [GHHM21] to Fiat-Shamir with aborts and achieves strong CMA security. When instantiated to the ROM, the latter analysis is arguably simpler than the one from [AFLT16], for which reason we only describe this one.

Fixing the [KLS18] analysis (up to the additional problem). We deviate from the original analysis immediately after Game 0. We let the GetTrans function program the hash values not only for the non-aborting transcript, but also for all the intermediate *aborting* transcript. We further make the GetTrans function deterministic by deriving randomness from a random function. When a hash query is made on input $(w\|\mu)$, the GetTrans function is then called to check if signing μ defines a hash value for $(w\|\mu)$. All this avoids falling into **F1**.

Then, one would like to rely on simulated transcripts so that we can simulate the game without knowing the signing key. To avoid falling into **F2**, one then needs to prove that all challenges are independent. We define a hybrid game in which GetTrans outputs a special symbol if it calls conflicting hash inputs (i.e., uses twice the same commitment w inside the loop). Applying the One-Sided O2H Lemma [AHU19] combined with the high min-entropy of commitments then allows us to bound the distinguishing advantage of a (quantum) adversary between these two games by $(Q_S + Q_H)B/\sqrt{2^\alpha}$, with α denoting the min-entropy of commitments, Q_S, Q_H the number of sign and hash queries, and B the parameter bounding the number of loop iterations. Note that GetTrans is invoked by both hash and sign queries (hence the Q_S+Q_H term). This solves **F2**: one can now replace real transcripts by simulated ones in GetTrans as all challenges are uniformly random and independent. Yet again, both hash and sign queries rely on GetTrans, which itself uses either actual or simulated transcripts. A similar argument as before, relying on Oracle-Indistinguishability [Zha12, Theorem 1.1] allows to bound the distinguishing advantage of a (quantum) adversary by $(Q_S + Q_H)^{3/2}/\sqrt{B \cdot \varepsilon_{zk}}$, handling the last error **F3**.

A security analysis in the QROM based on adaptive reprogramming. Independently, we provide a different security analysis based on the technique developed in [GHHM21]. In the latter, the authors study adaptive reprogramming in the

QROM and exploit it to analyze the (no-abort) Fiat-Shamir heuristic. They suggest that the latter analysis can be extended to the Fiat-Shamir with aborts setting, and we provide such an analysis (see Theorem 4).

Adaptive reprogramming considers a setting in which a quantum adversary has access to a random oracle, and in addition can query a reprogramming oracle $\mathcal{O}$ with inputs μ. The oracle answers to such a query by sampling w from a target distribution (the commitment space in our case) and returning it to the adversary. In addition, the oracle either leaves the random oracle unchanged, or reprograms it on input $(w\|\mu)$. In the classical setting, it is clear that an adversary cannot tell whether $\mathcal{O}$ affects the random oracle unless it has already made the hash query $(w\|\mu)$. In [GHHM21], the authors provide a bound for the distinguishing advantage of a quantum adversary.

Adaptive reprogramming allows to immediately move from Game 0 to a Game 1 in which the GetTrans function, on input μ, samples fresh uniformly random and independent challenges c and reprograms the random oracle according to c on input $w\|\mu$. This immediately solves **F1** as it programs all intermediate values (even though some values can get programmed multiple times), as well as **F2** since challenges are now set to uniformly random and independent values thanks to reprogramming. Note that hash queries do not need to run GetTrans as adaptive reprogramming guarantees the adversary cannot find inconsistencies (which would allow to distinguish Games 0 and 1). It remains to replace real transcripts by simulated ones, which is easily argued with a security loss of $BQ_S\varepsilon_{zk}$, since only the (classical) sign queries rely on running the simulator. Doing so, we circumvent **F3**. One then needs to keep consistency in the hash values, which is done by keeping track of the last values reprogrammed by the (polynomial number of classical) sign queries.

Insufficiency of the usual simulators for Σ-protocols with aborts. While the above approaches seem sound, they induce an additional subtle problem: we now run all aborting and non-aborting executions of the underlying Σ-protocol at every step of the reduction, and in particular in the game hop replacing real transcripts by simulated ones. The no-abort Honest-Verifier Zero-Knowledge (naHVZK) property usually considered for Σ-protocols with aborts is insufficient to analyze this game hop. Rather than trying to rely on the prior naHVZK notion, we choose an alternative route and exploit a stronger Honest-Verifier Zero-Knowledge (HVZK) for Σ-protocols with aborts, which requires the simulator to be able to simulate both aborting and non-aborting transcripts. Equipped with this definition, the above proofs go through immediately.

There is still one major issue to solve: this definition of strong simulation is not known to be achieved by Σ-protocols involved in Lyubashevsky's signatures (which might be the reason for the existence of the naHVZK notion). We construct a simulator for this setting, which works as follows: With probability p, it generates a non-aborting transcript (using the well-known naHVZK simulator), with p being the known probability that a protocol iteration does not trigger an abort. Else, with probability $1-p$, it returns a uniform commitment w (and $\perp$ for the z-part of the transcript). The main technicality is to show that uniform commitments are indeed indistinguishable from aborting transcript. Recall that the commitment w is of the form $\mathbf{Ay}$ for a public matrix $\mathbf{A}$ and a vector $\mathbf{y}$ sampled from a source distribution Q. If Q has high min-entropy, we use the fact that aborting does not decrease much the min-entropy of Q and use the leftover hash lemma to conclude that the protocol is *statistical* zero-knowledge. While this already handles many settings of Lyubashevsky's signature, we want the source distribution to have lower entropy in some cases. We prove that if the distribution Q is such that LWE is hard for noise distribution set to Q, the protocol is *computational* zero-knowledge, for a variant of computational zero-knowledgedness that is compatible with the Fiat-Shamir transform.

Correctness analysis of FSwBA signatures. In addition to these technical issues regarding the security analysis, it turns out that bounding the number of loop iterations and returning $\perp$ when the bound is reached makes the correctness analysis somewhat non-trivial. This is often brushed away in existing works, and we are not aware of a correct analysis. The goal is to provide a small upper bound on the probability that the signing algorithm outputs $\perp$. For this purpose, it is tempting to argue that at each loop iteration, the abort probability is the failure probability $\beta \in (0, 1)$ of the underlying proof system, and hence that the signing abort probability is β^B where B is the bound on the number of iterations. This is incorrect, as the executions of the underlying proof system are not statistically independent: all challenges are derived from the hash function. It hence seems unavoidable to assume the ROM not only for security but also for correctness, but this is not sufficient, as statistical dependencies between the loop iterations can stem from collisions between inputs of the hash function: if the hash inputs are the same in two iterations, the returned challenges are the same.

We provide a detailed proof of correctness. For this, we observe that the security analyses involves a game in which the signing loop iterations are statistically independent: the β^B bound above holds in these experiments. We then argue that the failure probability in the genuine execution is close to β^B, as otherwise we would be able to distinguish the genuine security experiment from the one in which the signing loop iterations are statistically independent. Our correctness analysis for FSwBA is actually a corollary of our (runtime) analysis of FSwUA, and is described in the corresponding section.

Wrapping up on FSwBA. We obtain several complete analyses with distinct security claims for signatures based on FSwBA, both in the ROM and the QROM. We provide an overview of our results in Table 1, using the same notation as above. The "reduction loss" is a bound on the difference of success probabilities of the adversary in the CMA and NMA security experiments. We assume the circuit model for quantum computations, except when mentioned otherwise. The table assumes that $Q_H \geq B \cdot Q_S$ (this assumption is justified by the fact that hash evaluations can be made without restriction whereas sign queries require interaction with the signer). Similarly, the zero-knowledge simulation time is neglected (unless it is very large, its contributions are typically dominated by the terms in the table). We also omit constant factors. Note that the reduction in the

ROM simulates the random oracle using the lazy sampling method. However, the QROM reductions are relative to another random oracle that is accessible to the challenger (this assumption may be removed by relying on a quantum pseudorandom function [Zha12]). More detailed statements can be found in the referenced theorems.

Analysis	Hash function	Reduction loss	Reduction runtime overhead
Adaptive reprogramming (Th. 4)	ROM	$2^{-\alpha} B Q_S Q_H$ $+ \varepsilon_{zk} B Q_S$	$Q_H \log(Q_H)$
Adaptive reprogramming (Th. 4)	QROM	$2^{-\alpha/2} B Q_S Q_H^{1/2}$ $+ \varepsilon_{zk} B Q_S$	$Q_H \log(B Q_S)$ with QRACM $B Q_S Q_H$ without
History-free for CMA_1 security (Th. 3)	QROM	$2^{-\alpha/2} B Q_H$ $+ \varepsilon_{zk}^{1/2} B^{1/2} Q_H^{3/2}$	$B Q_H$
History-free for CMA security (Th. 10)	QROM	$2^{-\alpha/2} B Q_S Q_H$ $+ \varepsilon_{zk}^{1/2} B^{1/2} Q_H^{3/2}$	$B Q_S Q_H$

Table 1: Comparison of the security analyses of FSwBA.

We observe that the QROM analyses are incomparable. In particular, the adaptive reprogramming technique from [GHHM21] is tight only when assuming quantum random access classical memory (QRACM), which is a stronger assumption than the quantum circuit model of computation. The history-free technique from [KLS18] is tight only when considering adversaries that may make at most one sign query for any message (CMA_1 security). This covers the deterministic version of the resulting signature, obtained by deriving the randomness from the message via a pseudo-random function evaluation. For CMA security, the reduction is not tight (even assuming QRACM) and the reduction loss is higher than the one obtained with the adaptive reprogramming technique.

2.3 Concrete Analysis of FSwUA

On the termination of FSwUA signatures. For FSwUA, we start by exhibiting an underlying identification scheme with the following peculiar property: for any execution of the key generation algorithm of the resulting signature, there exists a hash function such that the resulting signing algorithm loops forever on every input message. Yet, with overwhelming probability over the random choice of the hash function, the expected runtime is polynomially bounded. The scheme is a variant of Lyubashevsky's [Lyu09, Lyu12], with carefully crafted source and target distributions (we refer to [DFPS22] for a description of Lyubashevsky's signature with arbitrary source and target distributions). To make sure that every loop iteration always fails, we use a source and a target distribution that are uniform over some sets X_S and X_T, respectively, with $X_T \subseteq X_S$. The choice of uniform distributions leads to a deterministic rejection test: an iteration takes a uniform $\mathbf{y} \in X_S$ and maps it to a vector $\mathbf{z}$, and an abort occurs if $\mathbf{z} \notin X_T$.

Going a little further into the details, the vector $\mathbf{z}$ is of the form $\mathbf{z} = \mathbf{y} + sk \cdot \mathbf{c}$, where the integer matrix sk is the signing key and $\mathbf{c}$ is the output of the hash function H on a function of $\mathbf{y}$ and the message. We want to design X_S and X_T such that: (1) for all sk, the probability over $\mathbf{y} \leftarrow U(X_S)$ and H that $\mathbf{z} = \mathbf{y}+sk\cdot\mathbf{c}$ belongs to X_T is at least a positive constant, and (2) for all sk, there exists an H such that for all $\mathbf{y}$ and message, the vector $\mathbf{z} = \mathbf{y} + sk \cdot \mathbf{c}$ does not belong to X_T.

The first condition forces us to set X_S not much larger than X_T. For the second condition, we design H so that any $\mathbf{y}$ is sent outside of X_T. As H depends on a function of $\mathbf{y}$, we first make sure that this function is injective, so that H is a function of $\mathbf{y}$ itself (else we would have to consider the set of predecessors and design H to jointly send them all outside of X_T). This injectivity is obtained by relying on the lossy version of Lyubashevsky's signature scheme [AFLT16]. Then for a vector $\mathbf{y}$, we design $\mathbf{c}$ so that $\mathbf{y} + sk \cdot \mathbf{c}$ is not in X_T and set $\mathbf{c}$ as the output of H on $\mathbf{y}$ and the message. For this purpose, we set X_S as a hyperball and X_t as an inner crust (a corona that almost aligns with the hyperball boundary). As hyperballs are concentrated on their surface, the volume ratio can be bounded by a positive constant even with a thin crust. Now, if $\mathbf{y} \in X_S \setminus X_T$, we set $\mathbf{c} = \mathbf{0}$ (for every message). If $\mathbf{y}$ belongs to X_T, we choose $\mathbf{y'} \in X_S \setminus X_T$ near $\mathbf{y}$ and define $\mathbf{c}$ such that $\mathbf{y} + sk \cdot \mathbf{c}$ is very close to $\mathbf{y'}$: for this purpose, it suffices to round $\mathbf{y'}$ to the lattice spanned by sk; by taking sk that is well-conditioned, we can guarantee that the rounded vector is close to $\mathbf{y'}$ and remains outside of X_T. As a result, all loop iterations of the resulting FSwUA signing algorithm fail.

The above counter-example is admittedly contrived, but illustrates the fact that specific difficulties arise when analyzing the unbounded version of Fiat-Shamir with aborts. In particular, this suggests to modify the requirement of signing runtime, so that it is authorized to take longer than desired, but only with small probability over the randomness of the random oracle (see Definition 9 for the formal requirement). We show that the signature obtained with the FSwUA transform indeed fulfills this requirement (in the random oracle model).

Security and correctness analyses of FSwUA signatures. We reduce the NMA security of FSwUA signatures to their CMA security, both in the ROM and the QROM. For this purpose, it is tempting to add a bound on the number of loop iterations, argue that the adversary cannot notice the difference, and then use the NMA security to CMA security reduction of FSwBA signatures. To prove that the adversary cannot notice the difference between the unbounded and bounded versions of the signing algorithm, one would argue that the probability of reaching that bound in at least one sign query is negligible, as the number of loop iterations follows a geometric law. But as discussed earlier, this is not true since there is a statistical dependency between different iterations of the rejection sampling. However, we show that the probability of the number of iterations until a success outcome be larger than B is small, over the randomness of the random oracle. This allows us to show the expected equivalence between FSwBA and FSwUA when the bound B is large, as an adversary likely never sees $\perp$ with the FSwBA variant. Using the same notations as before, the reduction loss of this step is bounded by $Q_S \cdot \beta^B + 2^{-\alpha/2} \cdot BQ_S \cdot \sqrt{Q_H}$ in the QROM and by $Q_S \cdot \beta^B + 2^{-\alpha} \cdot BQ_SQ_H$ in the ROM. (As above, these bounds assume that $Q_H \geq BQ_S$ and omit constant terms; we additionally assume that $\beta \in (0,1)$ is a constant.)

We provide a correctness analysis of FSwUA signatures (in the ROM), which proceeds in a similar way. Assuming that the signature outputs a transcript, this transcript follows the same distribution as a transcript from the underlying identification protocol, i.e. the challenge is uniform over the challenge space. It may not be independent from previous signatures and failed iterations, but all that matters here is its marginal distribution. This lets us bound the correctness error of the signature as a function of the correctness error of the underlying identification protocol.

2.4 Related Works

The Fiat-Shamir with aborts paradigm [Lyu09, Lyu12] has been used too extensively to attempt a complete list of works whose provable security claims are impacted by the flaws we pointed at. The list notably includes the NIST-selected Dilithium signature scheme [DKL+18], whose provable security claim [DKL+18, Section 4.2] derives from [KLS18]. Our work provides fixes to the claims.

The difficulties encountered when analyzing Fiat-Shamir with aborts can be circumvented by modifying the scheme. For example, some works replace the rejection sampling used in Lyubashevsky's signatures by statistical flooding (see, e.g., [DPSZ12, Appendix A.1] in the context of zero-knowledge proofs, or [ASY22, Section 4] in the context of signatures). In [CLMQ21], the authors instantiate the hash function so that a proof can be obtained without the random oracle model. Another approach consists in committing to w rather than sending it. This idea is discussed in [BBE+18] and attributed therein to Vadim Lyubashevsky. All these proposals incur significant losses on signature sizes.

Our strong simulation has implications to masked instantiations of Lyubashevsky's signatures. For efficiency reasons, one does not want to mask the hash function evaluation. For this purpose, a heuristic assumption has been introduced in [BBE+18, BBE+19, MGTF19]: informally, it states that revealing the commitments of the aborted transcript does not hurt the security of the scheme. This assumption removes the need for masking the hash function since commitments are the only non-public information about the hash function evaluations (the message and the hash function are public). Our simulator shows that this heuristic assumption holds unconditionally for some parameter ranges.

A concurrent and independent work [BBD+] also identifies flaw F1 in prior works on Fiat-Shamir with aborts. It fixes it while still relying on a zero-knowledge notion that considers only non-aborting transcripts. Like in our approach based on adaptive reprogramming [GHHM21], the analysis from [BBD+] uses reprogramming for both aborting and non-aborting transcripts. It differs in that it then undoes the reprogrammings for rejecting transcripts. This is not required in our case as our zero-knowledge notion captures aborting transcripts. We then show that this strengthened zero-knowledge requirement is achieved for the main application of Fiat-Shamir with aborts. We further identify and fix other difficulties with the Fiat-Shamir with aborts paradigm, notably with the history-free approach from [KLS18] and termination and correctness in the unbounded case. On the other hand, the concurrent work [BBD$^+$] additionally offers a fully mechanized security proof for Dilithium (in the ROM) using the EasyCrypt formal-verification platform.

3 Preliminaries

We use code-based games to write the proofs. We use capital letters with fraktur font (e.g., $\mathfrak{L}$) to denote the list of objects. We let $\mathsf{Coll} : \mathfrak{L} \mapsto \{0,1\}$ be the function that takes as input a list and outputs 1 if and only if at least two of the elements of the list are equal. We sometimes abuse the notation and let $\mathsf{Coll}(\mathfrak{L})$ denote the event that it returns 1. We implicitly assume that all variables are parameterized by the security parameter λ. To denote that a function f (or a database) is reprogrammed at input x to the value y we use the notation $f^{x \mapsto y}$. All our logarithms are in base 2.

We provide reminders about probabilities, Rényi divergence, digital signatures, quantum computing in Appendix A.

3.1 Σ-Protocols

We start by recalling various definitions pertaining to Σ-protocols.

Definition 1 (Σ-Protocol with Aborts). *Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite sets. A Σ-protocol for a relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ with commitment set $\mathcal{W}$, challenge set $\mathcal{C}$ and response set $\mathcal{Z}$ is a 3-round interactive proof system between a prover written as $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$ and a verifier $\mathsf{V} = (\mathsf{V}_1, \mathsf{V}_2)$ with the following specifications:*

- $\mathsf{P}_1 : (x, y) \to (w, st)$ *is a* PPT *algorithm that takes as input a pair of strings in $\mathcal{X} \times \mathcal{Y}$ and outputs a commitment $w \in \mathcal{W}$ and a state $st \in \{0,1\}^*$;*
- $\mathsf{V}_1 : (x, w) \to c$ *is a* PPT *algorithm that takes as inputs a string $x \in \mathcal{X}$ and a commitment $w \in \mathcal{W}$ and outputs a challenge $c \in \mathcal{C}$;*
- $\mathsf{P}_2 : (x, y, w, c, st) \to z$ *is a* PPT *algorithm that takes as inputs a pair of strings in $\mathcal{X} \times \mathcal{Y}$, a commitment $w \in \mathcal{W}$, a challenge $c \in \mathcal{C}$, and a state st and outputs a response $z \in \mathcal{Z} \cup \{\bot\}$ (we say that P_2 aborts if it outputs $\bot$);*
- $\mathsf{V}_2 : (x, w, c, z) \to b \in \{0,1\}$ *is a deterministic polynomial-time algorithm that takes as inputs a string $x \in \mathcal{X}$, a commitment $w \in \mathcal{W}$, a challenge $c \in \mathcal{C}$, and a response $z \in \mathcal{Z}$ and outputs a bit b which represents acceptance or rejection; in the case that $z = \bot$, it returns* 0.

A Σ-protocol is said to be public-coin if V_1 *outputs a challenge string c that is uniformly sampled from the challenge space $\mathcal{C}$, independently from its input.*

Note that the above definition (and the following ones) is implicitly parameterized by the security parameter λ, that we omit for the sake of simplicity. Given a language $\mathcal{L} = \{x \in \mathcal{X} \mid \exists y \in \mathcal{Y} : (x, y) \in R\}$ for a relation $R \subseteq \mathcal{X} \times \mathcal{Y}$, we are interested in the following properties of a Σ-protocol.

Definition 2 (Correctness). *Let $\gamma, \beta > 0$. A Σ-protocol $((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is (γ, β)-correct if for every $x \in \mathcal{L}$ and valid witness $y \in \mathcal{Y}$ the following holds.*

- *If the response of the prover is not $\perp$, the verifier accepts with probability at least γ:*

$$\Pr\left[\mathsf{V}_2(x, w, c, z) = 1 \;\middle|\; \begin{array}{l} (w, st) \leftarrow \mathsf{P}_1(x, y), \\ c \leftarrow \mathsf{V}_1(x, w), z \leftarrow \mathsf{P}_2(x, y, w, c, st), \\ z \neq \perp \end{array}\right] \geq \gamma.$$

- *The probability that the prover aborts is bounded by β:*

$$\Pr\left[z = \perp \;\middle|\; \begin{array}{l} (w, st) \leftarrow \mathsf{P}_1(x, y), \\ c \leftarrow \mathsf{V}_1(x, w), z \leftarrow \mathsf{P}_2(x, y, w, c, st) \end{array}\right] \leq \beta.$$

We also let β denote *the probability of aborting*. We are interested in the regime of parameters in which $\gamma \geq 1 - \lambda^{-\omega(1)}$ and $\beta \leq 1 - 1/\mathsf{poly}(\lambda)$. Note that by repeating the protocol $\mathsf{poly}(\lambda)$ times, the parameter β is pushed toward 0, whereas γ stays close to 1.

We refer to the following definition as the one that is usually used in the literature of Fiat-Shamir with aborts. Note that we do not use it. Later in Section 4, we discuss our modifications.

Definition 3 (No-Abort Statistical Honest-Verifier Zero-Knowledge). *Let $\varepsilon_{zk}, T \geq 0$. A Σ-protocol is (ε_{zk}, T)-naHVZK if there exists a simulator Sim with runtime at most T, that given x, outputs a transcript (w, c, z) such that the distribution of (w, c, z) has statistical distance at most ε_{zk} from a honestly generated transcript (w', c', z') produced by the interaction conditioned on $z \neq \perp$.*

If Σ is public-coin, then without loss of generality, the challenge c can be sampled uniformly from the challenge space $\mathcal{C}$ and passed over as input to the simulator Sim. In the rest of the paper, we limit ourselves to public-coin Σ-protocols.

For cryptographic purposes, one instantiates the Σ-protocol with hard samples. This notion is captured in the following definition.

Definition 4 (Identification Protocol). *An identification protocol is a Σ-protocol for an NP relation R, where the prover and verifier are dealt their statement and witness by a PPT instance generator Gen.*

A useful statistical property of a Σ-protocol is the min-entropy of the commitments. We borrow the following definition from [KLS18].

Definition 5 (Commitment Min-Entropy). *For $\alpha \geq 0$, we say that an identification scheme $((\mathsf{P}_1, \mathsf{P}_2)$, $(\mathsf{V}_1, \mathsf{V}_2))$ with instance generator Gen has commitment min-entropy α if $H_\infty[w | (w, st) \leftarrow \mathsf{P}_1(x, y)] \geq \alpha$, for all $(x, y) \leftarrow \mathsf{Gen}(1^\lambda)$.*

Note that we could accommodate our results to schemes for which the above holds only with overwhelming probability over the randomness of Gen.

3.2 Fiat-Shamir Transform

Let $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ be an identification protocol with an ε-hard instance generator Gen for a binary relation R. Further, let $H : \{0,1\}^* \to \mathcal{C}$ be a hash function where $\mathcal{C}$ is the challenge space of Σ. Then, for every positive integer B, one can construct a signature scheme $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ by applying the Fiat-Shamir transform with bounded aborts (FSwBA) as in Figure 1. We are particularly interested in applying the Fiat-Shamir transform without imposing a bound on the number of iterations in the rejection sampling as it is the case for Dilithium [DKL+18], among other schemes. One can define the unbounded version $\mathsf{SIG}_\infty = \mathsf{FS}_\infty[\Sigma, H]$ of the Fiat-Shamir transform for a Σ-protocol Σ as in Figure 1. Note that the signing algorithm of SIG_∞ may not be PPT as required in Definition 10. Ideally, it would still be expected polynomial-time.

```
$\mathsf{KeyGen}(1^\lambda)$:
  $(x, y) \leftarrow \mathsf{Gen}(1^\lambda)$
  $(vk, sk) = (x, (x, y))$
  return $(vk, sk)$
```

```
$\mathsf{Sign}(sk, \mu)$:
  $\kappa := 1$
  While $z = \perp$ and $\kappa \leq B$
    $(w, st) \leftarrow \mathsf{P}_1(sk)$
    $c = H(w \| \mu)$
    $z \leftarrow \mathsf{P}_2(sk, w, c, st)$
    $\kappa := \kappa + 1$
  if $z = \perp$ return $\perp$
  return $\sigma = (w, z)$
```

```
$\mathsf{Ver}(vk, \mu, \sigma)$:
  Parse $\sigma = (w, z)$
  $c = H(w \| \mu)$
  return $\mathsf{V}_2(vk, w, c, z)$
```

Fig. 1: Signatures $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ and $\mathsf{SIG}_\infty = \mathsf{FS}_\infty[\Sigma, H]$. The signature SIG_B uses blocks highlighted with the blue color, whereas SIG_∞ does not.

In this work we show that $\mathsf{sUF\text{-}CMA}$ security (and sometimes $\mathsf{sUF\text{-}CMA}_1$) of such signatures can be reduced to their $\mathsf{UF\text{-}NMA}$ security. Here, we briefly recall two possible ways to reduce $\mathsf{UF\text{-}NMA}$ security to the security of the underlying Σ-protocol. For more details, we refer the reader to prior works (e.g., [Lyu09, Lyu12, AFLT16, DFMS19]).

- In [AFLT16, KLS18], the authors consider *lossy identification schemes* in which there exists another instance generator function $\mathsf{Gen}_{\mathsf{ls}}$ for the protocol that only outputs an instance x_{ls} without any witness. Moreover, its output distribution is computationally indistinguishable from the one of the real instance generator Gen. Further, it is said to be $\varepsilon_{\mathsf{ls}}$-sound if no cheating prover (even unbounded) can impersonate the real prover given x_{ls} as input and make the verifier to accept with probability more than $\varepsilon_{\mathsf{ls}}$. They reduce $\mathsf{UF\text{-}NMA}$ security of a signature based on the Fiat-Shamir transform to the $\varepsilon_{\mathsf{ls}}$-soundness of the underlying identification scheme and the indistinguishability of the outputs of Gen and $\mathsf{Gen}_{\mathsf{ls}}$.
- In [DFMS19] and implicitly in [Lyu09, Lyu12], the authors reduce $\mathsf{UF\text{-}NMA}$ security of a signature based on the Fiat-Shamir transform to the *proof of*

knowledge property of the underlying Σ-protocol. Their reduction is less tight than the one of [KLS18].

3.3 Adaptive Reprogramming in the QROM

We rely on the following lemma for one of our analyses in the QROM. Consider the following decision game: Assume the hash function takes inputs of the form (x_1, x_2), and an adversary (with quantum access to the hash function) has access to a reprogramming oracle which can be queried with any value x_2. On a query x_2, the oracle samples a value x_1 and either leaves the hash function unchanged or reprograms it on input (x_1, x_2) to a uniformly random value y from its range. It may also maintain a state x'. Given (x_1, x'), the adversary's goal is to decide whether the oracle reprograms the hash function or not. The following lemma proves this game to be hard even for quantum adversaries. We also remind the classical variant of this lemma (Lemma 12) in Appendix A.5.

Lemma 1 (Adaptive Reprogramming [GHHM21, Proposition 2]). *Let X_1, X_2, X' and Y be finite sets, and let D be a distribution on $X_1 \times X'$. Let $\mathcal{A}$ be a distinguisher playing in the reprogramming game in Figure 2 and making q quantum queries to the random oracle and r classical queries to the* Reprogram *function. Then*

$$\left| \Pr[1 \Leftarrow \mathsf{Reprogram}_0^{\mathcal{A}}] - \Pr[1 \Leftarrow \mathsf{Reprogram}_1^{\mathcal{A}}] \right| \leq \frac{3r}{2}\sqrt{q \cdot 2^{-\alpha}},$$

where α is the min-entropy of the first component of D.

```
Game Reprogram_b :
    $H_0 \leftarrow U(Y^{X_1 \times X_2})$
    $H_1 := H_0$
    $b' \leftarrow \mathcal{A}^{|H_b\rangle,\ \mathsf{Reprogram}(\cdot)}$
    return $b'$
```

```
Reprogram($x_2$) :
    $(x_1, x') \leftarrow D$
    $y \leftarrow U(Y)$
    $H_1 := H_1^{(x_1,x_2)\mapsto y}$
    return $(x_1, x')$
```

Fig. 2: The reprogramming game.

4 A Simulator for Lyubashevsky's Σ-Protocol

As we discussed in the introduction, Definition 3 is not sufficient for our purposes. In this section, we strengthen it in both statistical and computational settings.

We consider the following statistical HVZK definition, which benefits from a simulator even for aborting transcripts of the Σ-protocol. One can see this modification as a return to the classic definition in the literature of the zero-knowledge interactive proof systems.

Definition 6 (Statistical Honest-Verifier Zero-Knowledge). *Let $\varepsilon_{zk}, T \geq 0$. A Σ-protocol is (ε_{zk}, T)-*HVZK *if there exists a simulator* Sim *with runtime at most T, that given x, outputs a transcript (w, c, z) such that the distribution of (w, c, z) has statistical distance at most ε_{zk} from a honestly generated transcript (w', c', z') produced by the interaction. This includes aborting transcripts, i.e., those for which $z = \bot$.*

If Σ is public-coin, then without loss of generality, the challenge c can be sampled uniformly from the challenge space $\mathcal{C}$ and passed over as input to the simulator Sim.

A central application of the Fiat-Shamir with aborts paradigm is Lyubashevsky's signature scheme [Lyu09, Lyu12]. We show here that the underlying Σ-protocol satisfies the zero-knowledge property of Definition 6, i.e., admits an efficient simulator for all transcripts including the aborting ones.

Let us first recall the Σ-protocol, using the formalism from [DFPS22]. Let P and Q be two distributions over $\mathbb{Z}^m$: we refer to Q as the source distribution, and to P as the target distribution. The relation R is parametrized by a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ which we assume to be in Hermite Normal Form, i.e., $\mathbf{A} = (\mathbf{I}_n | \mathbf{B})$ for some $\mathbf{B} \in \mathbb{Z}_q^{n \times (m-n)}$. It is also parameterized by some dimension k and norm bound $\beta_{\mathsf{SIS}} > 0$. The relation R is of the form:

$$R_{m,n,k,q,\beta_{\mathsf{SIS}}}(\mathbf{A}) = \left\{ (\mathbf{S}, \mathbf{T}) \in \mathbb{Z}^{m \times k} \times \mathbb{Z}_q^{n \times k} : \mathbf{AS} = \mathbf{T} \bmod q \wedge \max_{i \in [k]} \|\mathbf{s}_i\| \leq \beta_{\mathsf{SIS}} \right\}.$$

The Σ-protocol, with repetition parameter $M \geq 1$ and norm bound β_{SIS} is given in Figure 3. We note that V_2 is not needed to discuss the zero-knowledge property of the protocol.

```
$\mathsf{P}_1(\mathbf{S})$ :
  $\mathbf{y} \leftarrow Q$
  $st := \mathbf{y}$
  $\mathbf{w} = \mathbf{Ay} \bmod q$
  return $\mathbf{w}$

$\mathsf{P}_2(\mathbf{S}, \mathbf{c}, st)$ :
  $\mathbf{z} := \mathbf{y} + \mathbf{Sc}$
  with probability $\min(P(\mathbf{z})/(M \cdot Q(\mathbf{y})), 1)$
      return $\mathbf{z}$
  else return $\bot$

$\mathsf{V}_1(\mathbf{T}, \mathbf{w})$ :
  $\mathbf{c} \leftarrow U(\mathcal{C})$
  return $\mathbf{c}$

$\mathsf{V}_2(\mathbf{T}, (\mathbf{w}, \mathbf{c}, \mathbf{z}))$ :
  if $\|\mathbf{z}\| \leq \beta_{\mathsf{SIS}}$ and $\mathbf{Az} = \mathbf{w} + \mathbf{Tc} \bmod q$
      return Accept
  return Reject
```

Fig. 3: Lyubashevsky's identification protocol.

We consider the simulator Sim described in Figure 4.

The proof that the simulation is correct in the non-aborting case is quite standard and derives from the rejection sampling. For the aborting case, our proof relies on the leftover hash lemma and requires the source distribution Q

```
$\mathsf{Sim}(\mathbf{T}, \mathbf{c})$ :
with probability $1/M$
  $\mathbf{z} \leftarrow P$
  $\mathbf{w} := \mathbf{A}\mathbf{z} - \mathbf{T}\mathbf{c}$
else
  $\mathbf{w} \leftarrow U(\mathbb{Z}_q^n)$
  $\mathbf{z} := \perp$
return $(\mathbf{w}, \mathbf{z})$
```

Fig. 4: Simulator Sim of Lyubashevsky's Σ-protocol.

to have high min-entropy. The case of low min-entropy source distributions Q is handled later on.

4.1 High Min-Entropy Source Distributions

We first consider the case where Q has high min-entropy. In that case, we obtain statistical zero-knowledgedness as per Definition 6.

Theorem 1. *Let $m \geq n$ and k be positive integers, q prime, $\varepsilon, \beta_{\mathsf{SIS}} > 0$ and $\eta \in [0, 1/2]$. Assume that*

$$H_\infty(Q) \geq n \log q + \log\Big(1 - \frac{1-\eta}{M}\Big) + 2 \log \frac{1}{\varepsilon}.$$

Let $(\mathbf{S}, \mathbf{T}) \in R_{m,n,k,q,\beta_{\mathsf{SIS}}}(\mathbf{A})$ for some $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{n \times m})$. Assume that

$$\forall \mathbf{c} \in \mathcal{C} : \Pr_{\mathbf{z} \leftarrow P}\Big[P(\mathbf{z}) \leq M \cdot Q(\mathbf{z} - \mathbf{S}\mathbf{c})\Big] \geq 1 - \eta.$$

Then the distribution of the transcript $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ generated by $\langle \mathsf{P}(\mathbf{S}), \mathsf{V}(\mathbf{T}) \rangle$ is within statistical distance $\varepsilon + \eta(1 + 1/M)$ from the distribution of the triple $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ obtained by sampling $\mathbf{c}$ uniformly in $\mathcal{C}$ and sampling $(\mathbf{w}, \mathbf{z}) \leftarrow \mathsf{Sim}(\mathbf{T}, \mathbf{c})$.

Observe that $\mathbf{c}$ is distributed uniformly in $\mathcal{C}$ in both genuine and simulated transcripts. It hence suffices to study the distribution of the rest of the transcript conditioned on the value of $\mathbf{c}$.

The first part of the following result derives from [DFPS22, Lemma 2.2], and the second part derives from the description of Sim. The claim ensures that the probabilities of the event $\mathbf{z} = \perp$ in the genuine and simulated transcripts are close-by.

Lemma 2. *For all $\mathbf{c}$ output by V_1, the probability (over the random coins of P_1 and P_2) that P_2 outputs $\perp$ belongs to $[1 - 1/M, 1 - (1-\eta)/M]$. For all $\mathbf{c}$, the probability (over its random coins) that the output component $\mathbf{z}$ of Sim is equal to $\perp$ is $1 - 1/M$.*

We now consider the transcript distribution conditioned on the event $\mathbf{z} \neq \perp$.

Lemma 3. *Conditioned on $\mathbf{z} \neq \perp$, the distribution of the transcript $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ generated by (P, V) is within statistical distance η from the simulated distribution.*

Proof. For all $\mathbf{c}$ and conditioned on $\mathbf{z} \neq \perp$, the distribution of $\mathbf{z}$ output by P_2 is within statistical distance η from P (see [DFPS22, Lemma 2.2]). The latter is exactly the distribution of $\mathbf{z}$ conditioned on $\mathbf{z} \neq \perp$.

To complete the proof of Lemma 3, we argue that when $\mathbf{z} \neq \perp$, the first coefficient of the triple is fully determined by the two others, and equal to $\mathbf{Az}-\mathbf{Tc}$ in both transcript and simulation. □

Finally, we consider the statistical distance of the distributions conditioned on $\mathbf{z} = \perp$. The following claim considers the distribution of the transcript conditioned on not outputting $\perp$.

Lemma 4. *Conditioned on $\mathbf{z} = \perp$, the distribution of the transcript $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ generated by (P, V) is within statistical distance ε from the simulated distribution.*

Proof. It suffices to prove that for all $\mathbf{c}$ and conditioned on $\mathbf{z} = \perp$, the distribution of $\mathbf{w}$ in the transcript generated by (P, V) is statistically close to uniform over $\mathbb{Z}_q^n$. Thanks to the first claim above, we have:

$$\begin{aligned} H_\infty[\mathbf{y}|\mathbf{c} \wedge \mathbf{z} = \perp] &\geq H_\infty[\mathbf{y}] - \log \Pr[\mathbf{z} = \perp|\mathbf{c}] \\ &\geq H_\infty[\mathbf{y}] - \log\Big(1 - \frac{1-\eta}{M}\Big). \end{aligned}$$

We conclude by using the leftover hash lemma (Lemma 9). □

Theorem 1 follows from the above lemmas by term collection. □

4.2 Low Min-Entropy Source Distributions

The above handles many settings of Lyubashevsky's signature, as the source distribution Q is often chosen to have high min-entropy so that the map $\mathbf{y} \mapsto \mathbf{Ay} \bmod q$ is (very) surjective. In some cases, however, it is chosen of lower entropy and the map $\mathbf{y} \mapsto \mathbf{Ay} \bmod q$ is very far from surjective. For example, this allows to avoid the forking lemma in the security proof [AFLT16], which both leads to a tight security proof and facilitates unforgeability proofs in the QROM. Our pathological construction from Section 6.1 also relies on this regime.

We explain how this can be handled, for some distributions. First, we consider computational zero-knowledgedness rather than statistical zero-knowledgedness. As one needs to be able to replace real transcripts of (many) sign queries by simulated ones in the security proof, we consider a strong notion of computational zero-knowledgeness: computational indistinguishability is required to hold even when the distinguisher is given the witness (of course, the simulator does not use the witness). This definition is compatible with our Fiat-Shamir with aborts analyses. For example, in the analysis based on adaptive reprogramming (Section 5.2), transcripts can be replaced one at a time by simulated ones using a hybrid argument, since the witness allows to generate real signatures. In particular, our definition implies the notion of computational HVZK for multiple transcripts used in [GHHM21, Definition 2], which they use to argue that all transcripts can be replaced by simulated ones in a single step. Note that in all the analyses we consider in this work, when we use the zero-knowledge property, the witness x is available to the challenger.

Definition 7 (Strong Computational HVZK). *Let $\varepsilon_{zk}, T \geq 0$ with ε_{zk} a negligible function of the security parameter. A Σ-protocol $((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ for a relation R is (ε_{zk}, T)-*sc-HVZK *if there exists a simulator* Sim *with runtime at most T such that for all polynomial-time algorithm $\mathcal{A}$ and all $(x, y) \in R$, the following advantage is $\leq \varepsilon_{zk}$:*

$$\mathrm{Adv}(\mathcal{A}) = \left| \Pr \left[\mathcal{A}((w,c,z),y) = 1 \,\middle|\, \begin{array}{c} (w, st) \leftarrow \mathsf{P}_1(x,y), \\ c \leftarrow \mathsf{V}_1(x,w), \\ z \leftarrow \mathsf{P}_2(x,y,c,w,st) \end{array} \right] - \Pr \left[\mathcal{A}((w,c,z),y) = 1 \,\middle|\, (w,c,z) \leftarrow \mathsf{Sim}(x) \right] \right|.$$

One may consider classical or quantum adversaries $\mathcal{A}$.

As in the statistical case, if the Σ-protocol is public-coin, then without loss of generality, the challenge c can be sampled uniformly from the challenge space $\mathcal{C}$ and passed over as input to the simulator Sim. In the following, we use this formalism.

The computational assumption that we rely on is the Learning With Errors problem [Reg09]. We use its knapsack form, introduced in [MM11].

Definition 8 (k-LWE). *Let $m \geq n \geq 1$, $q \geq 2$ and D a distribution over $\mathbb{Z}_q^m$. The search knapsack-LWE problem* $\text{sk-LWE}_{m,n,q,D}$ *with parameters m, n, q, D consists in recovering $\mathbf{e}$ from $(\mathbf{A}, \mathbf{Ae})$, where $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m \times n})$ and $\mathbf{e} \leftarrow D$. The decision knapsack-LWE problem* $\text{dk-LWE}_{m,n,q,D}$ *with parameters m, n, q, D consists in distinguishing between the distributions $(\mathbf{A}, \mathbf{Ae})$ and $(\mathbf{A}, \mathbf{u})$, where $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m \times n})$, $\mathbf{e} \leftarrow D$ and $\mathbf{u} \leftarrow U(\mathbb{Z}_q^n)$.*

We now argue that for some distributions Q, it is possible to prove computational zero-knowledgedness in the sense of Definition 7, with exactly the same simulator as above (Figure 4).

Theorem 2. *Let $m \geq n$ and k be positive integers, $q \leq \mathsf{poly}(m,n)$ prime and $\beta_{\mathsf{SIS}} > 0$. Assume that the distribution Q is such that the* $\text{dk-LWE}_{m,n,q,Q}$ *problem is hard. Let $(\mathbf{S}, \mathbf{T}) \in R_{m,n,k,q,\beta_{\mathsf{SIS}}}(\mathbf{A})$ for some $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{n \times m})$. Assume that*

$$\forall \mathbf{c} \in \mathcal{C} : \Pr_{\mathbf{z} \leftarrow P} \left[P(\mathbf{z}) \leq M \cdot Q(\mathbf{z} - \mathbf{Sc}) \right] \geq 1 - \eta,$$

where $1 + 1/\mathsf{poly}(m,n) \leq M \leq \mathsf{poly}(m,n)$ and $\eta \geq 0$ is negligible.

Then the distribution of the transcript $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ generated by $\langle \mathsf{P}(\mathbf{S}), \mathsf{V}(\mathbf{T}) \rangle$ is computationally indistinguishable from the distribution of the triple $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ obtained by sampling $\mathbf{c}$ uniformly in $\mathcal{C}$ and sampling $(\mathbf{w}, \mathbf{z}) \leftarrow \mathsf{Sim}(\mathbf{T}, \mathbf{c})$, even if the distinguisher is given $\mathbf{S}$.

The first two claims (Lemmas 2 and 3) of the proof of Theorem 1 still hold. It hence suffices to prove the statistical indistinguishability of the genuine and simulated transcripts $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ conditioned on $\mathbf{z} = \perp$.

We first show that the genuine distribution of $\mathbf{y}$ conditioned on $\mathbf{z}$ being rejected resembles the distribution Q of $\mathbf{y}$.

Lemma 5. *Assume that $M > 1$. Consider the execution $\langle \mathsf{P}(\mathbf{S}), \mathsf{V}(\mathbf{T}) \rangle$. Let $Q^{\perp}$ denote the distribution of $\mathbf{y}$ conditioned on $\mathbf{z} = \perp$. Then we have:*

$$R_{\infty}(Q^{\perp} \| Q) \leq \frac{M}{M-1}.$$

Proof. For all $\mathbf{y}$, we have

$$Q^{\perp}(\mathbf{y}) = \frac{\Pr[\mathbf{y} \wedge \mathbf{z} = \perp]}{\Pr[\mathbf{z} = \perp]} \leq \frac{Q(\mathbf{y})}{\Pr[\mathbf{z} = \perp]}.$$

Lemma 2 ensures that the denominator is at least $1 - 1/M$. □

The following result states that if $(\mathbf{A}, \mathbf{A}\mathbf{y})$ is pseudo-random for $\mathbf{y} \leftarrow D$, then so is it for $\mathbf{y} \leftarrow D'$ for any distribution D' such that $R_{\infty}(Q' \| Q)$ is polynomially bounded.

Lemma 6. *Let $m \geq n \geq 1$. Let $q \leq \mathsf{poly}(m,n)$ prime. Let D and D' be two distributions over $\mathbb{Z}^m$ such that $R_{\infty}(D' \| D) \leq \mathsf{poly}(m,n)$. Then* dk-LWE$_{m,n,q,D}$ *reduces to* dk-LWE$_{m,n,q,D'}$.

Proof. Note first that dk-LWE$_{m,n,q,D}$ reduces to sk-LWE$_{m,n,q,D}$. Also, as we have $R_{\infty}(D' \| D) \leq \mathsf{poly}(m,n)$, by the probability preservation property (see Lemma 10), sk-LWE$_{m,n,q,D}$ reduces to sk-LWE$_{m,n,q,D'}$. Finally, by [MM11, Theorem 3.1], sk-LWE$_{m,n,q,D'}$ reduces to dk-LWE$_{m,n,q,D'}$. The composition of these reductions leads to the above claim. □

Theorem 2 now follows from combining Lemmas 5, 6 2 and 3. □

5 ROM and QROM Analyses of FSwBA

In this section we discuss the security of the Fiat-Shamir transform with bounded aborts. We first prove the UF-CMA security of the signature in the QROM based on the flawed proof in [KLS18], and then in the sequel of the section we discuss the adaptive reprogramming techniques to prove the UF-CMA security in the QROM (and with tighter reductions in the ROM).

We further provide an analysis relying on the Rényi divergence instead of the statistical distance in Appendix B.2.

5.1 The History-Free Approach

Below, we reduce the (s)$\mathsf{UF\text{-}CMA}_1$ security to its $\mathsf{UF\text{-}NMA}$ security using the statistical zero-knowledge property of the Σ-protocol. One can see this proof as a correction of [KLS18]. Due to space limitation, we detail the proof in Appendix B.1. Moreover, we also claim that the same approach applies to $\mathsf{UF\text{-}CMA}$ security in Appendix B.1 (see Theorem 10).

Theorem 3. *Let $\varepsilon_{zk}, \alpha, T_{\mathsf{Sim}} \geq 0$, $B \geq 0$ and H and G hash functions modeled as random oracles. Assume that $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is an $(\varepsilon_{zk}, T_{\mathsf{Sim}})$-$\mathsf{HVZK}$ public-coin identification protocol, and that the commitment message of the prover has min-entropy α. For any quantum adversary $\mathcal{A}$ against $\mathsf{UF\text{-}CMA}_1$ (or $\mathsf{sUF\text{-}CMA}_1$) security of $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ that issues at most Q_H quantum queries to the random oracle H and Q_S classical queries to the signing oracle, there exists a quantum adversary $\mathcal{B}$ against $\mathsf{UF\text{-}NMA}$ security of SIG_B with $\mathsf{Time}(B) \approx \mathsf{Time}(A) + T_{\mathsf{Sim}} \cdot B \cdot (Q_S + Q_H)$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{(s)UF\text{-}CMA}_1}_{\mathsf{SIG}_B}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) &+ 2^{\frac{-\alpha+3}{2}} \cdot B \cdot (Q_S + Q_H) \\ &+ 30\sqrt{\varepsilon_{zk} \cdot B} \cdot (Q_S + Q_H)^{\frac{3}{2}} \ .\end{aligned}$$

Our reduction relies on $\mathcal{B}$ having access to a private random oracle H' with the same domain and range as H that is not accessible by $\mathcal{A}$.

The results also hold if we replace HVZK by $\mathsf{sc\text{-}HVZK}$ and assume ε_{zk} to be negligible in the security parameter.

Note that one could adjust the proof of the above statement (as well as those of the next statements) to replace access to the private random oracle by relying on a quantum pseudorandom function in the reduction [Zha12].

5.2 The Adaptive Reprogramming Approach

We show how to reduce $\mathsf{UF\text{-}CMA}$ security and $\mathsf{sUF\text{-}CMA}$ security of the signature to $\mathsf{UF\text{-}NMA}$ security, separately in the ROM and QROM. Our separate handling of the random oracle models enables us to obtain a tighter proof in the ROM compared to the lower bound that the QROM proof imposes on any ROM proof. We use the similar frameworks for adaptive reprogramming (Lemma 12 and Lemma 1) in the ROM and the QROM. Also, we note that our proof is crucially based on our new zero-knowledge simulator.

Theorem 4. *Let $\varepsilon_{zk}, \alpha, T_{\mathsf{Sim}} \geq 0$, $B \geq 0$ and H a hash function modeled as a random oracle. Assume that $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is a $(\varepsilon_{zk}, T_{\mathsf{Sim}})$-$\mathsf{HVZK}$ public-coin identification protocol and that the commitment message of the prover has min-entropy α. Let $\mathcal{A}$ be any arbitrary adversary against $\mathsf{UF\text{-}CMA}$ security of $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ that issues at most Q_H queries to the random oracle H and Q_S classical queries to the signing oracle. Let $\mathsf{X} \in \{\mathsf{UF}, \mathsf{sUF}\}$; we define Δ_X as follows: $\Delta_{\mathsf{UF}} = 0$ and $\Delta_{\mathsf{sUF}} = BQ_S \cdot 2^{-\alpha}$.*

- *In the ROM, there exists an adversary $\mathcal{B}$ against* UF-NMA *security of* SIG_B *with runtime* $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H) \log(B \cdot Q_S + Q_H))$ *such that*

$$\mathsf{Adv}^{\mathsf{X\text{-}CMA}}_{\mathsf{SIG}_B}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) + 2^{-\alpha} \cdot B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1) \\ + \varepsilon_{zk} \cdot B \cdot Q_S + \Delta_{\mathsf{X}} \ .$$

- *In the QROM, there exists an adversary $\mathcal{B}$ against* UF-NMA *security of* SIG_B *such that*

$$\mathsf{Adv}^{\mathsf{X\text{-}CMA}}_{\mathsf{SIG}_B}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) + 2^{-\frac{\alpha}{2}} \cdot \frac{3B \cdot Q_S}{2} \cdot \sqrt{(B \cdot Q_S + Q_H + 1)} \\ + \varepsilon_{zk} \cdot B \cdot Q_S + \Delta_{\mathsf{X}} \ .$$

  Our reduction relies on $\mathcal{B}$ having access to a private random oracle H' with the same domain and range as H that is not accessible by $\mathcal{A}$. Furthermore, the runtime of $\mathcal{B}$ is $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H) \log(B \cdot Q_S))$ *with QRACM, and* $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H) \cdot (B \cdot Q_S))$ *without QRACM.*

The results also hold if we replace HVZK *by* sc-HVZK *and assume ε_{zk} to be negligible in the security parameter.*

Proof. The proof is based on a sequence of hybrid games.

Game G_0. The first game is the UF-CMA security game (Figure 5).

```
Game :
  $\mathcal{M} := \varnothing$
  $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
  $(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{H,\ \mathsf{Sign}(sk,\cdot)}(vk)$
  Parse $\sigma^* = (w^*, z^*)$
  $c^* := H(w^* \| \mu^*)$
  return $[[\mu^* \notin \mathcal{M}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$

Sign$(sk, \mu)$ :
  $\mathcal{M} := \mathcal{M} \cup \{\mu\}$
  $(w, c, z) \leftarrow \mathsf{GetTrans}(\mu)$
  if $z = \perp$ return $\perp$
  return $\sigma = (w, z)$

GetTrans$(\mu)$ :
  $\kappa := 0$
  while $z = \perp$ and $\kappa \leq B$
    $(w, st) \leftarrow \mathsf{P}_1(sk)$
    $c := H(w \| \mu)$
    $z \leftarrow \mathsf{P}_2(sk, w, c, st)$
    $\kappa := \kappa + 1$
  return $(w, c, z)$
```

Fig. 5: Game G_0

Game G_1. In this game, the challenges of the transcripts are not computed by the random oracle anymore, but sampled independently and uniformly each time. Then, the random oracle is reprogrammed according to the new challenges as in Figure 6.

To bound the distance between Game_0 and Game_1, we construct a wrapper $\mathcal{D}$ around $\mathcal{A}$ that uses $\mathcal{A}$ to solve a reprogramming game. It works as in Figure 7.

```
$\underline{\mathsf{Game}}$ :
$\mathcal{M} := \varnothing$
$(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{H,\ \mathsf{Sign}(sk,\cdot)}(vk)$
Parse $\sigma^* = (w^*, z^*)$
$c^* := H(w^* \| \mu^*)$
return $[[\mu^* \notin \mathcal{M}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$

$\underline{\mathsf{Sign}(sk, \mu)}$ :
$\mathcal{M} := \mathcal{M} \cup \{\mu\}$
$(w, c, z) \leftarrow \mathsf{GetTrans}(\mu)$
if $z = \perp$ return $\perp$
return $\sigma = (w, z)$

$\underline{\mathsf{GetTrans}(\mu)}$ :
$\kappa := 0$
while $z = \perp$ and $\kappa \leq B$
  $(w, st) \leftarrow \mathsf{P}_1(sk)$
  $c \leftarrow U(\mathcal{C})$
  $z \leftarrow \mathsf{P}_2(sk, w, c, st)$
  $H = H^{w\|\mu \mapsto c}$
  $\kappa := \kappa + 1$
return $(w, c, z)$
```

Fig. 6: Game G_1. The difference from G_0 is highlighted in blue.

```
$\underline{\mathcal{D}^{H_b, \mathsf{Reprogram}}}$ :
$\mathcal{M} := \varnothing$
$(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{H_b,\ \mathsf{Sign}(sk,\cdot)}(vk)$
Parse $\sigma^* = (w^*, z^*)$
$c^* := H_b(w^* \| \mu^*)$
return $[[\mu^* \notin \mathcal{M}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$

$\underline{\mathsf{Reprogram}(\mu, sk)}$ :
$(w, st) \leftarrow \mathsf{P}_1(sk)$
$c \leftarrow U(\mathcal{C})$
$H_1 := H_1^{(w\|\mu) \mapsto c}$
return $(w, st)$

$\underline{\mathsf{Sign}(sk, \mu)}$ :
$\mathcal{M} := \mathcal{M} \cup \{\mu\}$
$\kappa := 0$
while $z = \perp$ and $\kappa \leq B$
  $(w, st) \leftarrow \mathsf{Reprogram}(\mu, sk)$
  $c := H_b(w\|\mu)$
  $z \leftarrow \mathsf{P}_2(sk, w, c, st)$
  $\kappa := \kappa + 1$
if $z = \perp$ return $\perp$
return $\sigma = (w, z)$
```

Fig. 7: The distinguisher $\mathcal{D}$.

Note that if $b = 0$ in Figure 7, then $\mathcal{D}$ perfectly simulates G_0, and otherwise it perfectly simulates G_1. Therefore,

$$\left|\Pr[1 \Leftarrow G_0^{\mathcal{A}}] - \Pr[1 \Leftarrow G_1^{\mathcal{A}}]\right| \leq \left|\Pr[1 \Leftarrow \mathsf{Reprogram}_0^{\mathcal{D}}] - \Pr[1 \Leftarrow \mathsf{Reprogram}_1^{\mathcal{D}}]\right|.$$

During the game, distinguisher $\mathcal{D}$ makes $B \cdot Q_S$ reprogramming queries and $B \cdot Q_S + Q_H + 1$ random oracle queries. In the ROM, Lemma 12 bounds the advantage of $\mathcal{D}$ by $B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1)2^{-\alpha}$. In the QROM, using Lemma 1, it follows that the advantage of $\mathcal{D}$ is bounded by

$$\frac{3B \cdot Q_S}{2} \cdot \sqrt{(B \cdot Q_S + Q_H + 1)2^{-\alpha}}.$$

Game G_2. Let Sim be the zero-knowledge simulator for Σ. In this game we modify GetTrans such that the transcripts are now produced by Sim and without the secret key. See Figure 8.

```
Game :
  $\mathcal{M} := \varnothing$
  $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
  $(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{H,\ \mathsf{Sign}(sk,\cdot)}(vk)$
  Parse $\sigma^* = (w^*, z^*)$
  $c^* := H(w^* \| \mu^*)$
  return $[[\mu^* \notin \mathcal{M}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$

Sign$(sk, \mu)$ :
  $\mathcal{M} := \mathcal{M} \cup \{\mu\}$
  $(w, c, z) \leftarrow \mathsf{GetTrans}(\mu)$
  if $z = \bot$ return $\bot$
  return $\sigma = (w, z)$

GetTrans$(\mu)$ :
  $\kappa := 0$
  while $z = \bot$ and $\kappa \leq B$
    $c \leftarrow U(\mathcal{C})$
    $(w, z) \leftarrow \mathsf{Sim}(vk, c)$
    $H := H^{w\|\mu \mapsto c}$
    $\kappa := \kappa + 1$
  return $(w, c, z)$
```

Fig. 8: Game G_2. The difference from G_1 is highlighted in blue.

We would like to bound the distance between games G_1 and G_2 using the zero-knowledge property. First we discuss the QROM case. Suppose that we are given a random oracle H' and $B \cdot Q_S$ transcripts that are either sampled honestly or sampled by the simulator. We use them to simulate G_1 or G_2, respectively. Note that in both games, after each transcript, the random oracle is reprogrammed according to the transcript. In order to simulate the reprogrammed random oracle perfectly, we keep track of a list $\mathfrak{D}$ of the classical values in which the random oracle must be reprogrammed. We describe the details in Figure 9.

Note that $\mathcal{C}$ can perfectly simulate G_1 or G_2 with its respective transcripts. Furthermore, it is given $B \cdot Q_S$ transcripts. By the statistical HVZK property of the Σ-protocol, it follows that

$$\left|\Pr[1 \Leftarrow G_1^{\mathcal{A}}] - \Pr[1 \Leftarrow G_2^{\mathcal{A}}]\right| \leq B \cdot Q_S \cdot \varepsilon_{zk}.$$

The ROM case is similar except that instead of using the private random oracle H' to simulate H, we use the lazy sampling method. The rest of the reduction is exactly the same as in Figure 9. We obtain

$$\left|\Pr[1 \Leftarrow G_1^{\mathcal{A}}] - \Pr[1 \Leftarrow G_2^{\mathcal{A}}]\right| \leq B \cdot Q_S \cdot \varepsilon_{zk}.$$

Game G_3. In this game, we add one more statement to the winning conditions. Let $(\mu^*, (w^*, z^*))$ be the forgery. If the value $w^* \| \mu^*$ has been programmed in the random oracle H during the game, then we abort. The value $w^* \| \mu^*$ would be programmed during the game if the adversary has made a sign query with μ^*. As the winning condition in the UF-CMA game already requires a forgery for a

```
$\mathcal{C}^{|H'\rangle}(\{w_{i,\kappa}, c_{i,\kappa}, z_{i,\kappa}\}_{i\in[Q_S],\kappa\in[B]})$ :
$\mathcal{M} := \varnothing$
$i := 0$
$\mathfrak{D} := \varnothing$
$(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{|H\rangle,\ \mathsf{Sign}(sk,\cdot)}(vk)$
Parse $\sigma^* = (w^*, z^*)$
$c^* := H_b(w^* \| \mu^*)$
return $[[\mu^* \notin \mathcal{M}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$
```

```
$H(w\|\mu)$ :
if $\exists c$ such that $(w, \mu, c) \in \mathfrak{D}$
    return $c$
return $H'(w\|\mu)$
```

```
$\mathsf{Sign}(sk, \mu)$ :
$\mathcal{M} := \mathcal{M} \cup \{\mu\}$
$i := i + 1$
$\kappa := 0$
while $z = \perp$ and $\kappa \leq B$
    $(w, c, z) = (w_{i,\kappa}, c_{i,\kappa}, z_{i,\kappa})$
    if $\exists c'$ such that $(w, \mu, c') \in \mathfrak{D}$
        $\mathfrak{D} := \mathfrak{D} \setminus (w, \mu, c')$
    $\mathfrak{D} := \mathfrak{D} \cup (w, \mu, c)$
    $\kappa := \kappa + 1$
if $z = \perp$ return $\perp$
return $\sigma = (w, z)$
```

Fig. 9: The distinguisher $\mathcal{C}$ for real and simulated transcripts of Σ based on $\mathcal{A}$.

message that has not been queried before, the adversary's view is identical to that of the previous one.

It remains to reduce G_3 to $\mathsf{UF\text{-}NMA}$ security. The signing algorithm does not use the signing key anymore and uses the zero-knowldege simulator to answer the sign queries. The last remaining technicality lies in how to simulate the random oracle. In the ROM, we use the lazy sampling method. At each query to the random oracle, we return a match if there exists any in the database, otherwise we return a fresh sampled element from the range of H and add it in the database. But in the QROM, we cannot simulate the random oracle with the lazy sampling method since the access to it is quantum. Recall that the reduction also has access to another random oracle to which the adversary does not have access. The reduction then tweaks this private random oracle over the reprogrammed inputs and uses it to simulate the random oracle queries of the adversary. Therefore, using $\mathsf{UF\text{-}NMA}$ game, one can perfectly simulate G_3 for the adversary. If the adversary $\mathcal{A}$ finds a forgery (μ^*, σ^*), then the random oracle has not been reprogrammed at this value during the course of G_3 since it has not been queried before. Hence, it would be a valid signature for $\mathsf{UF\text{-}NMA}$ game.

Strong Unforgeability. For the $\mathsf{sUF\text{-}CMA}$ security, we modify the above games. Now, the challenger maintains the list $\mathcal{M}$ of message-signature pairs that were queried by the adversary via the signature oracle. Each game, at its final step, also checks whether the forgery $(\mu^*, (w^*, z^*))$ belongs to this list or not, and if it is it returns 0. With these modifications, everything remains the same up to Game G_2. The last two games G_2 and G_3 behave differently only if we have the following conditions: $(\mu^*, (w^*, z^*)) \notin \mathcal{M}$, the random oracle has been reprogrammed on input $w^*\|\mu^*$, and $\mathsf{V}_2(vk, w^*, c^*, z^*) = 1$. The input $w^*\|\mu^*$ has been reprogrammed only if the adversary has made a sign query on μ^*. The probability of w^* appearing in any given loop iteration of the rejection sampling of $\mathsf{Sign}(sk, \mu^*)$ is bounded by $2^{-\alpha}$. In total, there are at most B iterations per sign query, and the adversary makes at most Q_S queries. By the union bound, the probability that $w^*\|\mu^*$ has been reprogrammed is bounded by $BQ_S \cdot 2^{-\alpha}$. The reduction from G_3 to the $\mathsf{UF\text{-}NMA}$ game works as before.

Runtime. We discuss two cases separately.

- In the ROM: Each sign query requires to run the zero-knowledge simulator up to B times. For each hash (resp. sign) query, the reduction performs 1 (resp. up to B) programming operation. It maintains a sorted data structure $\mathfrak{D}$ in order to search and insert in $\mathcal{O}(\log(B \cdot Q_S + Q_H))$ steps. The runtime of the reduction is of order $\mathsf{Time}(\mathcal{A}) + \mathcal{O}(T_{\mathsf{Sim}} \cdot (B \cdot Q_S + Q_H) \cdot \log(B \cdot Q_S + Q_H))$.
- In the QROM: We split the runtime analysis in two different models depending on whether we have access to QRACM or not. To answer the hash and sign queries properly, the reduction maintains a database of reprogrammed input-outputs, and at each query, it searches over the database to find a match. Note that it is being carried out in superposition. The size of the database is at most $B \cdot Q_S$, and a naive exhaustive search takes $B \cdot Q_S$. Moreover, for each sign query, the reduction runs the zero-knowledge simulator at most B times. Thus, the runtime would be $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H)(B \cdot Q_S))$. With QRACM, the reduction has the advantage to maintain a sorted database and quantumly search over the database. It reduces the search time to $\log(B \cdot Q_S)$. It yields the runtime $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H)\log(B \cdot Q_S))$. □

6 Concrete Analysis of FSwUA: Negative Result

In the rest of the paper, we focus on analyzing formally signatures constructed from combining an identification protocol with the Fiat-Shamir with unbounded aborts paradigm. To the best of our knowledge, this is the first complete analysis of FSwUA.

In this first section, we exhibit a signature constructed using FS_∞ for which the signing runtime is infinite for an instantiation of the hash function H. Therefore, the expected runtime is also infinite and the standard definition of runtime must be changed. We propose minor updates to the signature definitions so that they support such pathological behaviors. Note that FSwUA is the main paradigm used in practice: there is no reason to add a bound for the number of loop iterations in the code if the algorithm never reaches it except with negligible probability, but the latter statement thus needs to be proven.

In Section 7, we prove Fiat-Shamir with unbounded aborts does yield signatures (both in the ROM and QROM with tighter reductions in the ROM) which satisfy all correctness, runtime, and security requirements. Correctness of FSwBA is also addressed in Section 7 as a corollary of our analysis.

6.1 Infinite Signing Runtime in the Worst Case of FSwUA

In this section, we aim to prove the following theorem.

Theorem 5. *There exists a parametrization of* dk-LWE$_{m,n,q,Q}$ *such that the following holds assuming the hardness of* dk-LWE$_{m,n,q,Q}$*. There exists a public-coin identification protocol Σ with instance generator* Gen *such that, with overwhelming probability over the randomness of* Gen*, there exists a hash function H_{bad} such that the signing algorithm of $\mathsf{SIG}_\infty := \mathsf{FS}_\infty[\Sigma, H_{bad}]$ on inputs the signing key and any message does not halt.*

The proof relies on constructing the appropriate identification protocol, and then identifying a specific bad instantiation for the hash function. The main idea is to instantiate Lyubashevsky's signature scheme with source distribution Q being the uniform distribution over a ball B and target distribution being the uniform distribution over a corona C, as illustrated in Figure 10. For a keypair $\mathbf{A}, \mathbf{S}$, a loop iteration samples $\mathbf{y} \leftarrow U(B)$, defines a commitment $\mathbf{w} \leftarrow \mathbf{A}\mathbf{y} \bmod q$, and returns $\mathbf{y} + \mathbf{S}\mathbf{c}$ with $\mathbf{c} \leftarrow H(\mathbf{A}\mathbf{y} \bmod q \| \mu)$, if and only if $\mathbf{y} + \mathbf{S}\mathbf{c} \in C$.

The cornerstone of our proof is to show that there exists a hash function H_{bad} such that, for every message μ and every $\mathbf{y}$, the challenge $\mathbf{c} = H(\mathbf{A}\mathbf{y} \bmod q \| \mu)$ is such that $\mathbf{y} + \mathbf{S}\mathbf{c} \notin C$. This implies that the signing algorithm of $\mathsf{FS}_\infty[\Sigma, H_{bad}]$ never halts on any input message.

Proof (Theorem 5). We instantiate Lyubashevsky's signature in the low-density regime. We first construct the identification protocol, and then explain how to instantiate H_{bad} to obtain the result.

We use the following parameters:

- dimensions $n > 0$ and $m = 2n \geq 14$;
- a challenge bound $\tau > 24\sqrt{m}$;
- a good conditioning parameter $d = 300$;
- a crust width $t = d\tau$ and a corona width $t' = d\tau/3 - (d+1)\sqrt{m}$;
- a radius $r = m(t + t')$;
- a prime modulus $q \leq \mathsf{poly}(n)$ that satisfies $q \geq 16(r + \sqrt{m})^4$.

We define the following relation R:

$$R := \left\{ ((\mathbf{A}, \mathbf{A}\mathbf{S}), \mathbf{S}) \mid \mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{S} = \frac{2}{3} d\mathbf{I}_m + \mathbf{E} \in \mathbb{Z}^{m \times m}, \sigma_1(\mathbf{E}) \leq \frac{d}{3} \right\},$$

where $\sigma_1(\mathbf{E})$ denotes the largest singular value of $\mathbf{E}$ (when viewed as a real-valued matrix). Note that d is a multiple of 3 so that $\mathbf{S}$ is indeed integral.
Our choice of matrix $\mathbf{S}$ makes it so that $\sigma_1(\mathbf{S}) \leq d$ and $\mathbf{S}$ is full-rank (note that this is a real-valued matrix). We have $\mathbf{S}^{-1} = (2d/3)^{-1} \sum_{k \geq 0} (-(2d/3)^{-1}\mathbf{E})^k$, which satisfies $\sigma_1(\mathbf{S}^{-1}) \leq 3/d$. The matrix $\mathbf{S}$ is the relation witness. We now consider the challenge space $\mathcal{C}$. We set:

$$\mathcal{C} := \{\mathbf{c} \in \mathbb{Z}^m | \|\mathbf{c}\| \leq \tau\}.$$

As $t = d\tau$, we have $t \geq \|\mathbf{Sc}\|$ for all $\mathbf{c} \in \mathcal{C}$ and all $\mathbf{S} \in \mathbb{Z}^{m \times m}$ with $\sigma_1(\mathbf{S}) \leq d$.

We further define the ball B and corona C as follows.

$$B := \mathcal{B}_m(r) \text{ and } C := \mathcal{B}_m(r-t) \setminus \mathcal{B}_m(r-t-t') \ .$$

A graphical representation is given in Figure 10.

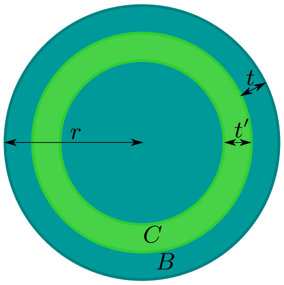

Fig. 10: The sets B and C in dimension 2.

We instantiate Lyubashevsky's signature scheme as recalled in Section 4, with the source distribution Q set as the uniform distribution over $\mathbb{Z}^m \cap B$ and the target distributions P set as the uniform distribution over $\mathbb{Z}^m \cap C$. The norm bound check of the verification algorithm is instantiated to $\|\mathbf{z}\| \leq r$, where $\mathbf{z}$ is the vector output by the prover. Finally, the rejection parameter M is set to $M = 100$.

Lemma 7. *The identification protocol Σ obtained by instantiating Figure 3 as described above is $(1, 1/M)$-correct. Under the* dk-LWE$_{m,n,q,Q}$ *hardness assumption, it is* sc-HVZK*.*

Proof. We prove each property as follows:
Correctness. The perfect correctness ($\gamma = 1$) follows from the fact that if the prover outputs something, it is by definition a rounding of an element belonging to C and satisfies the relation that the verifier checks. By design, the probability that the verifier outputs some $\mathbf{z} \neq \perp$ is $1/M$.

Zero-Knowledgedness. We now aim at using Theorem 2 to argue the zero-knowledgedness of the protocol. It suffices to show that for all $\mathbf{c} \in \mathcal{C}$ and all $\mathbf{z} \in \mathbb{Z}^m \cap C$, we have that $P(\mathbf{z}) \leq M \cdot Q(\mathbf{z} - \mathbf{Sc})$.

Note first that for the considered $\mathbf{S}$'s and $\mathbf{c}$'s, if $\mathbf{z}$ belongs to the support of P, then $\mathbf{z} - \mathbf{Sc}$ belongs to the support of Q. For such a $\mathbf{z}$, we have:

$$\begin{aligned} \frac{Q(\mathbf{z} - \mathbf{Sc})}{P(\mathbf{z})} &= \frac{|\mathbb{Z}^m \cap C|}{|\mathbb{Z}^m \cap B|} \\ &\geq \frac{\mathrm{Vol}(\mathcal{B}(r - t - \sqrt{m})) - \mathrm{Vol}(\mathcal{B}(r - t - t' + \sqrt{m}))}{\mathrm{Vol}(\mathcal{B}(r + \sqrt{m}))} \\ &= \left(1 - \frac{t + 2\sqrt{m}}{r + \sqrt{m}}\right)^m - \left(1 - \frac{t + t'}{r + \sqrt{m}}\right)^m \ . \end{aligned}$$

By expanding the difference of powers, we then obtain:

$$\begin{aligned}\frac{Q(\mathbf{z}-\mathbf{Sc})}{P(\mathbf{z})} &= \frac{t'-2\sqrt{m}}{r+\sqrt{m}} \cdot \sum_{k=0}^{m-1} \Big(1-\frac{t+2\sqrt{m}}{r+\sqrt{m}}\Big)^{m-1-k} \Big(1-\frac{t+t'}{r+\sqrt{m}}\Big)^{k} \\ &\geq \frac{t'-2\sqrt{m}}{r+\sqrt{m}} \cdot m \cdot \Big(1-\frac{t+t'}{r+\sqrt{m}}\Big)^{m-1} \\ &\geq \frac{t'-2\sqrt{m}}{t+t'+1} \cdot \Big(1-\frac{1}{2m}\Big)^{m} \ .\end{aligned}$$

In the last inequality, we use the fact that $r = m(t+t')$. Now, using the definitions of t and t', we obtain that the latter is $\geq 1/100$.

Let us now consider the probability β that some answer is output by P_2. Note that our choice of t is such that for any $\mathbf{S}$ and challenge $\mathbf{c}$, it holds that

$$C \subseteq B + \mathbf{Sc}.$$

Therefore, the probability that a uniform element from $B + \mathbf{Sc}$ belongs to C is:

$$\begin{aligned}\beta = \frac{\mathrm{Vol}(C)}{\mathrm{Vol}(B)} &= \left(1-\frac{t}{r}\right)^{m} - \left(1-\frac{t+t'}{r}\right)^{m} \\ &= \left(1-\frac{t}{r}-1+\frac{t+t'}{r}\right) \cdot \sum_{k=0}^{m-1} \left(1-\frac{t}{r}\right)^{m-1-k} \left(1-\frac{t+t'}{r}\right)^{k} \\ &\geq \frac{t'}{r} \cdot m \cdot \left(1-\frac{t+t'}{r}\right)^{m-1} \\ &\geq \frac{t'}{t+t'} \cdot \left(1-\frac{1}{m}\right)^{m} .\end{aligned}$$

For the last inequality, we used the fact that $r = m(t+t')$ and $1 - 1/m < 1$. By using the definitions of t and t', we obtain:

$$\beta \geq \frac{t'}{4(t+t')} \geq \frac{1}{4} \cdot \frac{\tau - 3(1+1/d)\sqrt{m}}{4\tau - 3(1+1/d)\sqrt{m}} \ .$$

We claim that the latter is $\geq 1/20$. Indeed, having this inequality is equivalent to $\tau \geq 12(1+1/d)\sqrt{m}$, which is satisfied when $\tau \geq 24\sqrt{m}$. □

We then show that, for any choice of $\mathbf{A}, \mathbf{S}$ such that $((\mathbf{A}, \mathbf{AS}), \mathbf{S}) \in R$, there exists a hash function H such that, using H to instantiate FWsUA, the signing algorithm of $\mathsf{FS}_\infty[\Sigma, H]$ never halts on any input message. That is, for every message μ and every $\mathbf{y}$, the challenge $\mathbf{c} = H(\mathbf{Ay} \bmod q \| \mu)$ is such that $\mathbf{y} + \mathbf{Sc} \notin C$.

Fix the matrices $\mathbf{A}$ and $\mathbf{S}$. We now show how to instantiate the hash function H so that the above holds. A first important observation is that multiplication by $\mathbf{A}$ of a short integer vector is injective. Note that $\mathbf{Ay} = \mathbf{Ay}' \bmod q$ for some $\mathbf{y} \neq \mathbf{y}' \in B$ implies that there exists an integer vector $\mathbf{x} \in \mathbb{Z}^m$

(namely $\mathbf{y}-\mathbf{y}'$) such that $\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$ and $0 < \|\mathbf{x}\| \le 2r$. Applying the following lemma with $B = 2r$, it holds that with probability at least $1 - 2^{-\Omega(n)}$ over the random choice of $\mathbf{A}$, such a vector $\mathbf{x}$ does not exist, by our choices of m and q.

Lemma 8. *Let $m, n > 0$ and q a prime. Let $B < q$. Then:*

$$\Pr_{\mathbf{A}\in\mathbb{Z}_q^{n\times m}} \left(\lambda_1(\Lambda_q^{\perp}(\mathbf{A})) < B\right) \le \mathrm{Vol}(\mathcal{B}_m(1)) \frac{(B+\sqrt{m}/2)^m}{q^n} \ .$$

Proof. The following relations follow from a union bound, the statistical independence of the rows of $\mathbf{A}$ and the fact that every short enough integer vector is non-zero modulo q.

$$\begin{aligned}\Pr_{\mathbf{A}\hookleftarrow U(\mathbb{Z}_q^{n\times m})}(\lambda_1(\Lambda_q^{\perp}(\mathbf{A})) < B) &\le \sum_{\substack{\mathbf{y}\in\mathbb{Z}^m \\ 0<\|\mathbf{y}\|\le B}} \Pr_{\mathbf{A}\hookleftarrow U(\mathbb{Z}_q^{n\times m})}(\mathbf{A}\mathbf{y} = \mathbf{0} \bmod q) \\ &= \sum_{\substack{\mathbf{y}\in\mathbb{Z}^m \\ 0<\|\mathbf{y}\|\le B}} \left(\Pr_{\mathbf{a}\hookleftarrow U(\mathbb{Z}_q^{m})}(\mathbf{a}^{\top}\mathbf{y} = \mathbf{0} \bmod q)\right)^n \\ &= \sum_{\substack{\mathbf{y}\in\mathbb{Z}^m \\ 0<\|\mathbf{y}\|\le B}} \frac{1}{q^n} \ .\end{aligned}$$

Finally, we note that the volume of the m-dimensional hyperball of center $\mathbf{0}$ and radius $B + \sqrt{m}/2$ is an upper bound on the number of summands. □

As a consequence, we can define H as a function of $\mathbf{y}$ as $\mathbf{A}\mathbf{y}$ uniquely determines $\mathbf{y}$. Based on the protocol, it suffices to find a challenge $\mathbf{c} \in \mathcal{C}$ for each $\mathbf{y} \in \mathbb{Z}^m \cap B$, it holds that $\mathbf{y} + \mathbf{S}\mathbf{c} \notin C$. We then set $H(\mathbf{A}\mathbf{y} \bmod q, \mu)$ to be this $\mathbf{c}$ for all messages μ.

First, note that if $\mathbf{y} \notin C$, then setting $\mathbf{c} := \mathbf{0}$ leads to $\mathbf{y} + \mathbf{S}\mathbf{c}$ being rejected. Thus, we focus on the other case. Let $\Lambda(\mathbf{S})$ be the full-rank lattice generated by the matrix $\mathbf{S}$ (recall that $\mathbf{S}$ is full-rank). Define the scaling $\lambda = t'/\|\mathbf{y}\|$ and note that $\|\lambda\mathbf{y}\| = t'$. Let $\mathbf{x} \in \Lambda(\mathbf{S})$ be such that $\lambda\mathbf{x} \in \mathbf{y} + \mathcal{P}(\mathbf{S})$, where $\mathcal{P}(\mathbf{S}) = \mathbf{S}\cdot[0,1]^n$ denotes the (closed) fundamental parallelepiped spanned by $\mathbf{S}$. In particular there exists a lattice point $\mathbf{e} \in \mathbf{x} + \mathcal{P}(\mathbf{S})$ such that

$$\langle \mathbf{e} - \lambda\mathbf{y}, \lambda\mathbf{y}\rangle \ge 0 \ , \tag{1}$$

since otherwise there would exist an affine hyperplane separating $\lambda\mathbf{y} \in \mathbf{x} + \mathcal{P}(\mathbf{S})$ from $\lambda\mathbf{y}$, which would contradict the definition of $\mathbf{x}$. Note that $\|\mathbf{e} - \lambda\mathbf{y}\| \le d\sqrt{m}$: indeed, when written in the basis $\mathbf{S}$, all of its coordinates belong to $[-1, 1]$, and we have $\sigma_1(\mathbf{S}) \le d$. Since $\mathbf{e} \in \Lambda(\mathbf{S})$, there exists $\mathbf{k} \in \mathbb{Z}^n$ such that $\mathbf{e} = \mathbf{S}\mathbf{k}$. We set the challenge $\mathbf{c}$ as $\mathbf{k}$. To conclude, we prove the following statements.

$$\|\mathbf{c}\| \le \tau \quad \text{and} \quad \mathbf{y} + \mathbf{e} \notin C \ .$$

The first one follows from the following (recall that $t' = d\tau/3 - (d+1)\sqrt{m}$):

$$\begin{aligned}\|\mathbf{c}\| = \|\mathbf{S}^{-1}\mathbf{e}\| &\leq \sigma_1(\mathbf{S}^{-1})[\|\mathbf{e} - \lambda\mathbf{y}\| + \|\lambda\mathbf{y}\|] \\ &\leq \frac{3}{d}(d\sqrt{m} + t' + \sqrt{m}) = \tau \ .\end{aligned}$$

By using Equation (1), we obtain the following.

$$\begin{aligned}\|\mathbf{y} + \mathbf{e}\|^2 &= \|\lambda\mathbf{y} + \mathbf{y} + (\mathbf{e} - \lambda\mathbf{y})\|^2 \\ &= \|\lambda\mathbf{y}\|^2 + \|\mathbf{y}\|^2 + \|\mathbf{e} - \lambda\mathbf{y}\|^2 \\ &\quad + 2\langle\lambda\mathbf{y}, \mathbf{y}\rangle + 2\langle\lambda\mathbf{y}, \mathbf{e} - \lambda\mathbf{y}\rangle + 2\langle\mathbf{e} - \lambda\mathbf{y}, \mathbf{y}\rangle \\ &\geq \|\lambda\mathbf{y}\|^2 + \|\mathbf{y}\|^2 + 2\langle\lambda\mathbf{y}, \mathbf{y}\rangle \\ &= ((\lambda + 1)\|\mathbf{y}\|)^2 \ .\end{aligned}$$

Using the definition of λ and the lower bound on $\|\mathbf{y}\|$, we obtain that

$$\|\mathbf{y} + \mathbf{e}\| \geq (t' + r - t - t') = r - t \ .$$

This completes the proof: instantiated with this hash function, the signing algorithm of the Fiat-Shamir transform of the above Σ-protocol never halts. □

So far, this only exhibits a single bad choice for the hash function, while signatures based on FSwUA support messages of unbounded length. Hence, there are infinitely many possible hash functions (functions with domain $\mathcal{W} \times \{0,1\}^*$ and range $\mathcal{C}$, with $\mathcal{W}$ being the commitment space). As a consequence, it is not immediate that a single bad hash function implies an infinite expected runtime for the signature scheme in the ROM, and one could think that simply considering the runtime when H is a random oracle could be sufficient to fix it.

Corollary 1. *We have* $\Pr_H[\forall w \in \mathcal{W}, H(w\|\mu) = H_{bad}(w\|\mu)] \geq |\mathcal{C}|^{-|\mathcal{W}|}$ *for any message* μ*. Therefore, the expected runtime of* $\mathsf{Sign}(sk, \mu)$ *over the choice of the random oracle* H *is infinite.*

Our result relies on the hardness of the dk-LWE problem when the weight vector is sampled from the uniform distribution over a hyperball. This is an unusual distribution for dk-LWE. However, it can be checked that for appropriate parameters, the proof of [BLR+18, Section 5] that decision LWE is hard for a noise distribution that is uniform in a hypercube carries over to the hyperball setting.

6.2 Updated Signature Definition

As shown in Section 6.1, there are instances of identification protocols that yield signature schemes with infinite expected runtime of the signing algorithm. This requires relaxing the runtime requirement in the definition to be expected polynomial time with overwhelming probability over the choice of the hash function.

Yet, there is another subtlety doing so: in the security game, an adversary might make a sign query that never halts. In the case of the above construction, the challenger, which is unbounded, can still notice it as the commitment space is bounded and the rejection step is deterministic. Once all the potential commitments have failed to produce a valid signature, the challenger knows that it cannot answer the query. This is however not the case of every signature scheme. To take such event into account, we consider that an attacker automatically wins if the challenger takes more than T' time to answer a signature query, for some parameter T'. An alternative choice could be to consider that an adversary which makes a non-terminating sign query loses, since the challenger does not answer anymore. We prefer to add this parameter T' as this makes the definition stronger by further guaranteeing that an adversary cannot find a query which forces the signer to run for a long time, which could be desirable in practice as well.

We now state our updated definition for signatures. It is highly similar to the standard Definition 10 and we only highlight the differences.

Definition 9 (Modified Digital Signature in the ROM). *Let H be a random oracle to which all algorithms have oracle access. A signature scheme is a tuple* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *of algorithms with the following specifications. Everything is as in Definition 10, except for the runtime of* Sign*, which we define below, and a minor tweak in the security game.*

- $\mathsf{Sign}^H : (sk, \mu) \to \sigma$ *is a probabilistic algorithm that takes as inputs a signing key sk and a message $\mu \in \mathcal{M}$ and outputs a signature σ. We denote with $T_{\mathsf{Sign}^H(sk,\mu)}$ the runtime of* $\mathsf{Sign}(sk, \mu)$*.*

Let $\gamma > 0, T = \mathsf{poly}(\lambda)$ and $\varepsilon = \mathsf{negl}(\lambda)$. We say that the signature scheme is γ-correct if for any pair (vk, sk) in the range of KeyGen *and μ,*

$$\Pr[\mathsf{Verify}(vk, \mu, \mathsf{Sign}(sk, \mu)) = 1 \mid \mathsf{Sign}(sk, \mu) \textit{ halts}] \geq \gamma,$$

and we say that it is (T, ε)-efficient if for any pair (vk, sk) in the range of KeyGen *and μ,*

$$\Pr_H[T_{\mathsf{Sign}^H(sk,\mu)} > T] < \varepsilon.$$

where both probabilities are taken over the random coins of the two algorithms and the random oracle.

In addition, we update the security game as follows. Let T' be another function of λ. We define T'-$\mathsf{UF}\text{-}\mathsf{CMA}$ *security exactly as* $\mathsf{UF}\text{-}\mathsf{CMA}$ *security in Definition 11, except that we further make the adversary win as soon as it makes a sign query for which the signing algorithm takes more than T' steps to halt.*

Definition 9 does not forbid the situation described in Subsection 6.1 from occurring but guarantees that it should be hard to find non-halting queries.

7 Concrete Analysis of FSwUA: Positive Results

Equipped with this updated definition, we prove that signatures constructed from applying FSwUA to an identification protocol yields a signature scheme that satisfies all three *correctness*, *runtime*, and *security* requirements. This result extends to prove that FSwBA signatures satisfy *correctness*.

Theorem 6 (Runtime). *Let $\gamma > 0, \beta \in (0,1)$ and H a hash function modeled as a random oracle. Let $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ be an identification protocol that is (γ, β)-correct and has commitment min-entropy α. Let $\mathsf{SIG}_\infty = \mathsf{FS}_\infty[\Sigma, H]$. Let $\mathcal{M}$ be the message space and $I_{\mathsf{Sign}^H}(sk, \mu)$ denote the random variable counting the number of iterations of the signing algorithm on input (sk, μ) using a random oracle H where $\mu \in \mathcal{M}$. It holds that for any $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$, any message $\mu \in \mathcal{M}$, and any integer i:*

$$\Pr_H(I_{\mathsf{Sign}^H}(sk, \mu) > i) \leq \beta^i + \frac{2^{-\alpha}}{(1-\beta)^3}.$$

Proof. Let us start by introducing the random variables $(w_i, c_i, z_i, \mathtt{acc}_i)_{i \geq 1}$. It denotes an infinite sequence of transcripts, where $\mathtt{acc}_i$ is the random variable denoting whether the transcript is accepted or not. It takes value in $\{0,1\}$, where 0 denotes rejection and 1 acceptance. For the sake of the proof, let the sequence continue regardless of whether a prior transcript was accepted or not. Let $N = I_{\mathsf{Sign}^H(sk,\mu)}$. It denotes the index of the first accepting transcript, i.e., $N = \mathrm{argmin}_i(\{\mathtt{acc}_i = 1\})$. Let us denote by M the index of the first collision, i.e., $M = \min\{i | \exists j < i, w_j = w_i\}$. Note that once H is fixed, a transcript is a deterministic function of w_i.

Let $i \geq 1$. Let us decompose:

$$\begin{aligned}\Pr_H(N > i) &= \Pr_H(N < M) \cdot \Pr_H(N > i | N < M) \\ &\quad + \Pr_H(N \geq M) \cdot \Pr_H(N > i | N \geq M) \\ &\leq 1 \cdot \Pr_H(N > i | N < M) + \Pr_H(N \geq M) \cdot 1.\end{aligned}$$

We now focus on studying each of these probabilities. The second one can be rewritten as

$$\Pr_H(N \geq M) = \sum_{k=2}^{\infty} \Pr_H(M = k) \cdot \Pr_H(N \geq M | M = k).$$

Let us first focus on $\Pr_H(M = k)$. The random variable M only depends on the w_i's, which are i.i.d.: we can bound the collision probability using Lemma 11. Hence $\Pr_H(M = k) \leq k^2 \cdot 2^{-\alpha-1}$. Next, as long as no collision occurred, all c_i's can be seen as "fresh" randomness, i.e., all c_i's are uniform over the challenge space and most importantly, they are independent. Hence conditioned on $M = k$, we know that the probability of rejecting the first $k-1$ samples is β^{k-1}. Then

$$\Pr_H(N \geq M) \leq \sum_{k=2}^{\infty} k^2 \cdot 2^{-\alpha-1} \cdot \beta^{k-1} = 2^{-\alpha-1} \cdot \frac{\beta+1-(1-\beta)^3}{(1-\beta)^3}$$
$$\leq 2^{-\alpha} \cdot \frac{1}{(1-\beta)^3} \ ,$$

where the equality comes from the fact that $\sum_{k\geq 1} k^2 \cdot \beta^{k-1} = (\beta+1)/(1-\beta)^3$. Now, as we previously stated, conditioned on $N < M$, the distribution of N is geometric with parameter $1-\beta$. Hence, we have $\Pr_H(N > i | N < M) = \beta^i$. Plugging everything together, we obtain

$$\Pr_H(N > i) \leq \beta^i + \frac{2^{-\alpha}}{(1-\beta)^3} \ . \qquad \square$$

Assume that $\alpha = \omega(\log(\lambda))$. Setting $i = \omega(\log(\lambda)/\log(1/\beta))$ ensures that with overwhelming probability over the choice of H, signing runs in polynomial time. We note that this bound does not contradict the previous (negative) result. Indeed, it does not imply any statement on the finiteness of the expected value of T_{Sign^H}, which is infinite in the previous section.

We move on to checking that FSwUA satisfies the new γ-correctness property, assuming that the underlying identification protocol is (γ, β)-correct.

Theorem 7. *Let $\gamma > 0, \beta \in (0,1)$ and let H denote a hash function modeled as a random oracle. Let $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ be an identification protocol that is (γ, β)-correct. Let T denote the runtime of one interaction in the worst-case. Let $\alpha > 0$ be its commitment min-entropy. Let $\mathsf{SIG}_\infty = \mathsf{FS}_\infty[\Sigma, H]$. Then for any $i = \omega(\log(\lambda)/\log(1/\beta))$, it is γ-correct as well as $(iT, \beta^i + 2^{-\alpha}/(1-\beta)^3)$-efficient.*

Proof. Let $(sk, vk) \leftarrow \mathsf{KeyGen}$ and $\mu \in \mathcal{M}$. Conditioned on $\mathsf{Sign}(sk, \mu)$ halting, the output transcript follows the same distribution as a transcript from the identification protocol conditioned on not being $\perp$. In particular, the challenge is uniform over $\mathcal{C}$, as it is a hash that comes from the random oracle. Only its marginal distribution is important here, as well as the fact that it is independent from the first and last message of the prover. Hence, this transcript is accepted with probability γ over the random coins of Sign and the random oracle. □

With FSwBA, the problem is reversed: bounding the runtime becomes easy, whereas proving the correctness becomes mildly more tedious, as one needs to check that $\perp$ is not output too often.

Theorem 8. *Let $\gamma > 0, \beta \in (0,1)$ and $B > 0$. Let H be a hash function modeled a random oracle. Let $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ be an identification protocol that is (γ, β)-correct and has commitment min-entropy α. Let $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$. Then, for any $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and any message $\mu \in \mathcal{M}$, we have*

$$\Pr[\mathsf{Verify}(vk, \mu, \mathsf{Sign}(sk, \mu)) = 1] \geq \gamma \cdot \left(1 - \beta^B - \frac{2^{-\alpha}}{(1-\beta)^3}\right),$$

where the randomness is taken over H as well as the coins of Sign.

Proof. The result follows from Theorem 6. Indeed, assuming that Sign did not output $\perp$, then the final challenge that it outputs is uniform over the challenge space $\mathcal{C}$. It may not be independent from previous executions of the identification protocol, but nonetheless its marginal distribution is uniform over $\mathcal{C}$. Hence, assuming that Sign did not output $\perp$, it outputs a signature that is accepted by Verify with probability at least γ, by correctness of the identification protocol. In the case where Sign outputs $\perp$, this signature is of course rejected by Verify. Hence, by the law of total probabilities we have

$$\Pr[\mathsf{Verify}(vk, \mu, \mathsf{Sign}(sk, \mu)) = 1] \geq \gamma \cdot \left(1 - \beta^B - \frac{2^{-\alpha}}{(1-\beta)^3}\right). \qquad \square$$

We finally prove the security of the unbounded version of the Fiat-Shamir transform in both ROM and QROM. We note that our proof in the ROM is tighter. We reduce the T'-$\mathsf{UF\text{-}CMA}$ security of the unbounded signature scheme to the $\mathsf{UF\text{-}CMA}$ security of the bounded one in the QROM.

Theorem 9. *Let $\alpha \geq 0, \beta \in (0,1)$, and let H be a hash function modeled as a random oracle. Assume that $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is a (γ, β)-correct identification protocol, and that the commitment message of P_1 has min-entropy α. Let T denote the runtime of one iteration of the protocol with the hash function. Let $T' > BT$. For any arbitrary adversary $\mathcal{A}$ against T'-$\mathsf{UF\text{-}CMA}$ security of $\mathsf{SIG}_\infty = \mathsf{FS}_\infty[\Sigma, H]$ that issues at most Q_H queries to the random oracle H and Q_S classical queries to the signing oracle and for any fixed integer B, the same adversary $\mathcal{A}$ against $\mathsf{UF\text{-}CMA}$ security of $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ is such that $|\mathsf{Adv}^{T'\text{-}\mathsf{UF\text{-}CMA}}_{\mathsf{SIG}_\infty}(\mathcal{A}) - \mathsf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathsf{SIG}_B}(\mathcal{A})|$ is bounded as*

$$Q_S \cdot \beta^B + \frac{\beta^B \cdot 2^{-\alpha}}{(1-\beta)^3} + \begin{cases} 2^{-\alpha} \cdot B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1) & \textit{in the ROM,} \\ 2^{-\frac{\alpha}{2}} \cdot \frac{3B \cdot Q_S}{2} \cdot \sqrt{(B \cdot Q_S + Q_H + 1)} & \textit{in the QROM.} \end{cases}$$

This also holds replacing $\mathsf{UF\text{-}CMA}$ with $\mathsf{UF\text{-}CMA}_1$ or $\mathsf{sUF\text{-}CMA}$ security.

Proof. We proceed with three hybrid games.

Game G_0. We define Game G_0 as the $\mathsf{UF\text{-}CMA}$ security of SIG_B.

Game G_1. Let Game G_1 be game T'-$\mathsf{UF\text{-}CMA}$ in which the adversary is promised to not make any sign query that takes more than T' steps to halt. In the ROM, if the advantage of the adversary $\mathcal{A}$ to distinguish these games is non-zero, then $\mathcal{A}$ must have queried a message μ such that $\mathsf{Sign}(sk, \mu) = \perp$ in Game G_0. The similar statement holds in the QROM. Note that we cannot assume $\mathcal{A}$ is a purified quantum circuit since the queries to the signing oracle must be classical and cannot be purified. Nevertheless, we can purify $\mathcal{A}$ between the sign queries (the random oracle queries are quantum and would cause no problem for purification). This is equivalent to saying that after the i-th sign query μ_i, and receiving σ_i as the outcome, the adversary applies U_i, where U_i comes from a distribution derived from $\{\sigma_j\}_{j \leq i}$, and then measures one of its registers to obtain μ_{i+1}. It repeats this process Q_S times. By doing so, we can prove the above statement. As long as $\mathsf{Sign}(sk, \mu_i) \neq \perp$, the distributions of σ_i and thus U_i are identical. It follows that the mixed state of the adversary remains identical in both games.

Let $\mathcal{R}^{G_0,\mathcal{A}}$ be an algorithm that runs G_0 with $\mathcal{A}$ as a subroutine, records the sign queries of $\mathcal{A}$, and wins if one of them is answered by $\perp$. We have

$$\left|\Pr[1 \Leftarrow G_1^{\mathcal{A}}] - \Pr[1 \Leftarrow G_0^{\mathcal{A}}]\right| \leq \Pr[\mathsf{win}(\mathcal{R}^{G_0,\mathcal{A}})].$$

We aim at bounding the winning probability of $\mathcal{R}$. Remember G_1 from Figure 6, which we rename G_0' in this proof. In Theorem 4, we proved that

$$\left|\Pr[1 \Leftarrow G_0^{\mathcal{A}}] - \Pr[1 \Leftarrow G_0'^{\mathcal{A}}]\right| \leq \frac{3B \cdot Q_S}{2} \cdot \sqrt{(B \cdot Q_S + Q_H + 1)2^{-\alpha}},$$

in the QROM, and

$$\left|\Pr[1 \Leftarrow G_0^{\mathcal{A}}] - \Pr[1 \Leftarrow G_0'^{\mathcal{A}}]\right| \leq 2^{-\alpha} \cdot B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1),$$

in the ROM. It follows that we can replace Game G_0 in $\Pr[\mathsf{win}(\mathcal{R}^{G_0,\mathcal{A}})]$ with G_0' and only lose the above terms in their corresponding random oracle models.

Finally, using the union bound and the β-correctness of the identification protocol, the winning probability of the algorithm $\mathcal{R}$ relative to G_0' is bounded by $Q_S \cdot \beta^B$.

Game G_2. This is the genuine T'-UF-CMA game. The distinguishing advantage of $\mathcal{A}$ is bounded by the probability that $\mathcal{A}$ makes a sign query that takes more than T' steps to halt. Theorem 6 implies that this probability is bounded by $\beta^{T'/T} + 2^{-\alpha}/(1-\beta)^3 \geq \beta^B + 2^{-\alpha}/(1-\beta)^3$. This completes the proof. □

Acknowledgments. The authors thank Léo Ducas, Serge Fehr, Andreas Hülsing, Christian Majenz and Thomas Prest for helpful discussions. The authors were supported by the AMIRAL ANR grant (ANR-21-ASTR-0016), the PEPR quantique France 2030 programme (ANR-22-PETQ-0008) and the PEPR Cyber France 2030 programme (ANR-22-PECY-0003).

References

ABB$^+$17. E. Alkim, N. Bindel, J. Buchmann, Ö. Dagdelen, E. Eaton, G. Gutoski, J. Krämer, and F. Pawlega. Revisiting TESLA in the quantum random oracle model. In *PQCrypto*, 2017.

AFLT16. M. Abdalla, P.-A. Fouque, V. Lyubashevsky, and M. Tibouchi. Tightly secure signatures from lossy identification schemes. *J. Cryptol.*, 2016.

AHU19. A. Ambainis, M. Hamburg, and D. Unruh. Quantum security proofs using semi-classical oracles. In *CRYPTO*, 2019.

ASY22. S. Agrawal, D. Stehlé, and A. Yadav. Round-optimal lattice-based threshold signatures, revisited. In *ICALP*, 2022.

BBD$^+$. M. Barbosa, G. Barthe, C. Doczkal, J. Don, S. Fehr, B. Grégoire, Y.-H. Huang, A. Hülsing, Y. Lee, and X. Wu. Fixing and mechanizing the security proof of Fiat-Shamir with aborts and Dilithium. Submitted to CRYPTO 2023.

BBE$^+$18. G. Barthe, S. Belaïd, T. Espitau, P.-A. Fouque, B. Grégoire, M. Rossi, and M. Tibouchi. Masking the GLP lattice-based signature scheme at any order. In *EUROCRYPT*, 2018.

BBE$^+$19. G. Barthe, S. Belaïd, T. Espitau, P.-A. Fouque, M. Rossi, and M. Tibouchi. GALACTICS: gaussian sampling for lattice-based constant- time implementation of cryptographic signatures, revisited. In *CCS*, 2019.

BDF$^+$11. D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. In *ASIACRYPT*, 2011.

BLR$^+$18. S. Bai, T. Lepoint, A. Roux-Langlois, A. Sakzad, D. Stehlé, and R. Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. *J. Cryptol.*, 2018.

CLMQ21. Y. Chen, A. Lombardi, F. Ma, and W. Quach. Does Fiat-Shamir require a cryptographic hash function? In *CRYPTO*, 2021.

DFMS19. J. Don, S. Fehr, C. Majenz, and C. Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In *CRYPTO*, 2019.

DFPS22. J. Devevey, O. Fawzi, A. Passelègue, and D. Stehlé. On rejection sampling in Lyubashevsky's signature scheme. In *ASIACRYPT*, 2022.

DKL$^+$18. L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *TCHES*, 2018.

DPSZ12. I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias. Multiparty computation from somewhat homomorphic encryption. In *CRYPTO*, 2012.

FS86. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, 1986.

GHHM21. A. B. Grilo, K. Hövelmanns, A. Hülsing, and C. Majenz. Tight adaptive reprogramming in the QROM. In *ASIACRYPT*, 2021.

JS19. S. Jaques and J. M. Schanck. Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE. In *CRYPTO*, 2019.

Kat21. S. Katsumata. A new simple technique to bootstrap various lattice zero-knowledge proofs to QROM secure NIZKs. In *CRYPTO*, 2021.

KLS18. E. Kiltz, V. Lyubashevsky, and C. Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In *EUROCRYPT*, 2018.

LSS14. A. Langlois, D. Stehlé, and R. Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In *EUROCRYPT*, 2014.

Lyu09. V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, 2009.

Lyu12. V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, 2012.

Lyu16. V. Lyubashevsky. Digital signatures based on the hardness of ideal lattice problems in all rings. In *ASIACRYPT*, 2016.

MGTF19. V. Migliore, B. Gérard, M. Tibouchi, and P.-A. Fouque. Masking dilithium - efficient implementation and side-channel evaluation. In *ACNS*, 2019.

MM11. D. Micciancio and P. Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In *CRYPTO*, 2011.

NC11. M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2011.

Reg09. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 2009.

Sch89. C.-P. Schnorr. Efficient identification and signatures for smart cards (abstract). In *EUROCRYPT*, 1989.

vEH14. T. van Erven and P. Harremoës. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 2014.

Zha12. M. Zhandry. How to construct quantum random functions. In *FOCS*, 2012.

A Additional Preliminaries

A.1 Probabilities

For a finite set C, we let $U(C)$ denote the uniform distribution over C. Let X and Y be two random variables over some finite space Ω. We denote the statistical distance between them as

$$\Delta(X,Y) := \frac{1}{2}\sum_{\omega\in\Omega} \big|\Pr_X[\omega] - \Pr_Y[\omega]\big|.$$

The min-entropy of X is

$$H_\infty(X) := -\log \max_{\omega\in\Omega} \Pr_X[\omega].$$

Lemma 9 (Leftover Hash Lemma). *Let $\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y}\}$ be a 2-universal hash function family. Then for any random variable X over $\mathcal{X}$ and $\varepsilon > 0$ such that $H_\infty(X) \geq \log|\mathcal{Y}| + 2\log(1/\varepsilon)$, the distributions $(h, h(X))$ and $(h, U(\mathcal{Y}))$ are within statistical distance ε.*

Further, the family $\{\mathbf{A} \in \mathbb{Z}_q^{n\times m} : \mathbf{y} \mapsto \mathbf{Ay}\}$ is 2-universal for any prime q.

Assuming now that $\mathsf{Supp}(X) \subseteq \mathsf{Supp}(Y)$, the Rényi divergence of infinite order is defined as follows:

$$R_\infty(X\|Y) := \max_{x\in\mathsf{Supp}(X)} \frac{\Pr_X(x)}{\Pr_Y(x)}.$$

We use the same notations if X, Y are probability distributions. In the following, for the sake of simplicity, we restrict ourselves to discrete distributions. The definition above and our results involving the Rényi divergence carry over to continuous ones. The same holds for their applicability to Lyubashevsky's signature, as argued in [DFPS22]. Some background on the Rényi divergence are reminded below.

A.2 Properties of the Rényi Divergence

The following lemma borrowed from [LSS14] lists a few properties of the Rényi divergence. Proofs can be found in [vEH14].

Lemma 10. *Let P and Q be two discrete probability distributions such that we have $\mathsf{Supp}(P) \subseteq \mathsf{Supp}(Q)$. The following properties hold.*

- ***Log. Positivity:*** $R_\infty(P\|Q) \geq R_\infty(P\|P) = 1$.

- ***Data Processing Inequality:*** *$R_\infty(P^f \| Q^f) \leq R_\infty(P\|Q)$ for any probabilistic function f, where X^f denotes the distribution of $f(x)$ where $x \hookleftarrow X$.*
- ***Multiplicativity:*** *Let P and Q be two distributions of a pair of random variables X_1 and X_2 and P_i and Q_i denote the marginal distribution of X_i under P and Q, respectively. We have*

$$R_\infty(P\|Q) \leq R_\infty(P_1\|Q_1) \cdot \max_{x_1 \in \mathsf{Supp}(P_1)} R_\infty((P_2|x_1)\|(Q_2|x_1)).$$

- ***Probability Preservation:*** *Let $E \subseteq \mathsf{Supp}(Q)$ be an arbitrary event. Then we have*

$$P(E) \leq Q(E) \cdot R_\infty(P\|Q).$$

A.3 Signatures

Here we briefly recall the formalism of digital signatures.

Definition 10 (Digital Signature). *A signature scheme is a tuple of* PPT *algorithms* (KeyGen*,* Sign, Verify) *with the following specifications:*

- KeyGen : $1^\lambda \to (vk, sk)$ *outputs a verification key vk and a signing key sk;*
- Sign : $(sk, \mu) \to \sigma$ *takes as inputs a signing key sk and a message μ and outputs a signature σ;*
- Verify : $(vk, \mu, \sigma) \to b \in \{0,1\}$ *is a deterministic algorithm that takes as inputs a verification key vk, a message μ, and a signature σ and outputs a bit $b \in \{0,1\}$.*

Let $\gamma > 0$. We say that it is γ-correct if for any pair (vk, sk) in the range of KeyGen *and μ,*

$$\Pr[\mathsf{Verify}(vk, \mu, \mathsf{Sign}(sk, \mu)) = 1] \geq \gamma,$$

where the probability is taken over the random coins of the signing algorithm. We say that it is correct in the (Q)ROM if the above holds when the probability is also taken over the randomness of the random oracle modeling the hash function used in the scheme.

We also remind the definition of existential unforgeability against chosen message attacks (UF-CMA).

Definition 11 (Security). *Let $T, \delta \geq 0$. A signature scheme* SIG = (KeyGen, Sign, Verify) *is said to be (T, δ)-*UF-CMA *secure in the ROM if for any quantum adversary $\mathcal{A}$ with runtime $\leq T$ given (classical) access to the signing oracle and (quantum) access to a random oracle H, it holds that*

$$\Pr_{(vk,sk) \leftarrow \mathsf{KeyGen}(1^\lambda)}[\mathsf{Verify}(vk, \mu^*, \sigma^*) = 1 | (\mu^*, \sigma^*) \leftarrow \mathcal{A}^{H,\mathsf{Sign}}(vk)] \leq \delta,$$

where the randomness is also taken over the random coins of $\mathcal{A}$. The adversary should also not have issued a sign query for m^*. The above probability of forging *a signature is called the advantage of $\mathcal{A}$ and denoted by* $\mathsf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathsf{SIG}}(\mathcal{A})$*. If $\mathcal{A}$ does not output anything, then it automatically fails.*

*If we allow the adversary to forge a new signature for a previously queried message, the security is called strong existential unforgeability against chosen message attack (*sUF-CMA*). Existential unforgeability against one-per-message (resp. no-message) chosen message attack, denoted by* $\mathsf{UF\text{-}CMA}_1$ *(resp.* UF-NMA*) is defined similarly except that the adversary is allowed to query at most one (resp. not allowed to query any) signature per message. Further, one can similarly define* $\mathsf{sUF\text{-}CMA}_1$ *by taking the conjunction of* sUF-CMA *and* $\mathsf{UF\text{-}CMA}_1$*.*

Note that for deterministic signatures, the $\mathsf{UF\text{-}CMA}_1$ and UF-CMA security notions coincide.

A.4 Quantum Computations

A quantum state $|\psi\rangle$ of a system is a unit vector in the Hilbert space $\mathbb{C}^d$. Each step of a quantum algorithm is either a unitary transformation or a quantum measurement over the states. A unitary transformation over the space $\mathbb{C}^d$ is a $d \times d$ matrix $\mathbf{U}$ such that $\mathbf{U}\mathbf{U}^* = \mathbf{I}_d$ where $\mathbf{U}^*$ is the conjugate-transpose of $\mathbf{U}$. Let $\{|b_i\rangle\}_{i\in[d]}$ be an orthonormal basis for $\mathbb{C}^d$. Measuring a state $|\psi\rangle$ with this basis returns a value i with probability $|\langle b_i|\psi\rangle|^2$, and the post-measurement state is $|b_i\rangle$.

Let $f : \{0,1\}^n \to \{0,1\}^m$ be an arbitrary function. Then the quantum oracle $|f(\cdot)\rangle$ is a unitary transformation, acting on the computational basis $\{|x\rangle|y\rangle : x \in \{0,1\}^n, y \in \{0,1\}^m\}$ as $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$, and extended by linearity. An oracle-aided quantum machine $\mathcal{A}^{\mathcal{O}}$ is allowed to use an oracle $\mathcal{O}$ as a black box by querying $\mathcal{O}$ in some quantum state.

Consider a scenario in which we have an array (a data structure) of N classical strings $x_1, \ldots, x_N$. Quantum Random Access Classical Memory (QRACM) allows us to load this data to a quantum register in superposition. More precisely, a QRACM operation is defined by

$$U_{QRACM} : |i\rangle|y_i\rangle \mapsto |i\rangle|y_i \oplus x_i\rangle$$

for all i and y_i, and is extended by linearity. The efficiency of quantum random access gates has been a point of debate [JS19]. In this work, we single out the results using QRACM because of its difference from the quantum circuit model.

For more details on quantum computations, we refer the reader to [NC11].

A.5 Useful Lemmas

We recall an upper bound on the collision probability of i.i.d. random variables.

Lemma 11. *Let $\mathfrak{L}$ be a list of i.i.d. random variables $\{X_i\}_i$ over a finite set, each of which has min-entropy α. We have*

$$\Pr[\mathsf{Coll}(\mathfrak{L})] \leq |\mathfrak{L}|^2 \cdot 2^{-\alpha-1}.$$

Proof. Let ℓ denote the size of $\mathfrak{L}$. We bound this probability recursively:

$$\begin{aligned}
\Pr[\mathsf{Coll}(\mathfrak{L}) = 1] &= \Pr[\mathsf{Coll}(\{w_i\}_{i\in[\ell]}) = 1] \\
&\leq \Pr[\mathsf{Coll}(\{w_i\}_{i\in[\ell-1]}) = 1] \\
&\quad + \Pr[\mathsf{Coll}(\{w_i\}_{i\in[\ell-1]}) = 0 \wedge \mathsf{Coll}(\{w_i\}_{i\in[\ell]}) = 1] \\
&= \Pr[\mathsf{Coll}(\{w_i\}_{i\in[\ell-1]}) = 1] + (\ell - 1) \cdot 2^{-\alpha} \\
&\qquad\qquad\qquad\qquad\vdots \\
&\leq (\ell-1) \cdot 2^{-\alpha} + (\ell-2) \cdot 2^{-\alpha} + \cdots + 2^{-\alpha} \\
&\leq |\ell|^2 \cdot 2^{-\alpha-1}.
\end{aligned}$$

□

We finally state the classical variant of Lemma 1.

Lemma 12 (Classical Adaptive Reprogramming). *Let X_1, X_2, X' and Y be finite sets, and let D be a distribution on $X_1 \times X'$. Let $\mathcal{A}$ be a distinguisher playing in the reprogramming game in Figure 11 and making q classical queries to the random oracle and r classical queries to the* Reprogram *function. Then*

$$\left| \Pr[1 \Leftarrow \mathsf{Reprogram}_0^{\mathcal{A}}] - \Pr[1 \Leftarrow \mathsf{Reprogram}_1^{\mathcal{A}}] \right| \leq rq \cdot 2^{-\alpha},$$

where α is the min-entropy of the first component of D.

```
Game Reprogram_b :
$H_0 \leftarrow U(Y^{X_1 \times X_2})$
$H_1 := H_0$
$b' \leftarrow \mathcal{A}^{H_b,\ \mathsf{Reprogram}(\cdot)}$
return $b'$
```

```
Reprogram($x_2$) :
$(x_1, x') \leftarrow D$
$y \leftarrow U(Y)$
$H_1 := H_1^{(x_1,x_2)\mapsto y}$
return $(x_1, x')$
```

Fig. 11: The reprogramming game.

Proof. Note that the adversary makes q random oracle queries, implying that at most q input-output pairs of the random oracle are being revealed. If a reprogramming query does not coincide with these values, then the view of the adversary is identical for $b = 0$ and $b = 1$. For each reprogramming query, the probability of having a collision with the known random oracle values is at most $q \cdot 2^{-\alpha}$ since the input min-entropy of each reprogramming call is α. One can complete the proof by using the union bound. □

B Deferred Material of Section 5

B.1 History-Free Analysis: Proof of Theorem 3

In this section, we provide the corrected security analysis of Theorem 3, that is based on [KLS18], in the QROM. We need the following lemmas. The first one is the one-sided O2H lemma.

Lemma 13 (One-Sided O2H [AHU19, Theorem 3], adapted). *Let X, Y, S be three finite sets with $S \subseteq X$. Let $H, G : X \to Y$ be two functions such that $H(x) \neq G(x)$ if and only if $x \in S$. Let $\mathcal{A}$ be a quantum algorithm that distinguishes quantum oracles $|G\rangle$ and $|H\rangle$ with q queries and success probability $\varepsilon_{\mathcal{A}}$. Then, there exists a quantum algorithm $\mathcal{B}$ that, given access to the oracle $|H\rangle$ and $\mathcal{A}$, finds an element in S with success probability $\geq \varepsilon_{\mathcal{A}}^2/(4q^2)$.*

The next lemma links two notions of indistinguishability.

Lemma 14 (Oracle-Indistinguishability [Zha12, Theorem 1.1]). *Let D_1 and D_2 be efficiently samplable distributions with supports contained in a finite set Y. Let X be an arbitrary finite set. Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be two functions from X to Y such that, on each input $x \in X$, they output an independent sample in Y from D_1 and D_2, respectively. Let $\mathcal{A}$ be a quantum adversary that distinguishes two quantum oracles $|\mathcal{O}_1\rangle$ and $|\mathcal{O}_2\rangle$ with advantage ε by making q quantum queries. Then there exists a quantum algorithm $\mathcal{B}$ that distinguishes D_1 and D_2 with advantage $\geq (6q)^{-3}\varepsilon^2$.*

The proof of Theorem 3 is based on a sequence of hybrid games. Recall that we assumed the reduction has access to another random oracle H' to which the adversary does not have access to, which serves to simulate the random oracle.

Game G_0. This is the genuine $\mathsf{UF\text{-}CMA}_1$ game, as described in Figure 12.

```
Game :
  $\mathcal{M} := \varnothing$
  $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
  $(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{|H\rangle,\ \mathsf{Sign}(sk,\cdot)}(vk)$
  Parse $\sigma^* = (w^*, z^*)$
  $c^* := H(w^* \| \mu^*)$
  return $[[\mu^* \notin \mathcal{M}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$

Sign$(sk, \mu)$ :
  if $\mu \in \mathcal{M}$ return $\bot$
  $\mathcal{M} := \mathcal{M} \cup \{\mu\}$
  $(w, c, z) \leftarrow \mathsf{GetTrans}(\mu)$
  if $z = \bot$ return $\bot$
  return $\sigma = (w, z)$
```

```
$H(w\|\mu)$ :
  return $H'(w\|\mu)$

GetTrans$(\mu)$ :
  $\kappa := 0$
  while $z = \bot$ and $\kappa \leq B$
    $(w, st) \leftarrow \mathsf{P}_1(sk)$
    $c := H'(w\|\mu)$
    $z \leftarrow \mathsf{P}_2(sk, w, c, st)$
    $\kappa := \kappa + 1$
  return $(w, c, z)$
```

Fig. 12: Game G_0

Game G_1. In this game, described in Figure 13, we record all the transcripts produced during GetTrans and return them as its output. The function Sign runs GetTrans on its input μ. Hence, we modify it to single out the last transcript of the recording and continue with it as before. Nothing else changes in this game.

```
$\mathsf{Sign}(sk, \mu)$ :
  if $\mu \in \mathcal{M}$ return $\perp$
  $\mathcal{M} := \mathcal{M} \cup \{\mu\}$
  $\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i \in [\kappa]} \leftarrow \mathsf{GetTrans}(\mu)$
  if $z^{(\kappa)} = \perp$ return $\perp$
  return $\sigma = (w^{(\kappa)}, z^{(\kappa)})$
```

```
$\mathsf{GetTrans}(\mu)$ :
  $\kappa := 1,\ z^{(0)} := \perp$
  while $z^{(\kappa-1)} = \perp$ and $\kappa \leq B$
    $(w^{(\kappa)}, st^{(\kappa)}) \leftarrow \mathsf{P}_1(sk)$
    $c^{(\kappa)} := H'(w^{(\kappa)} \| \mu)$
    $z^{(\kappa)} \leftarrow \mathsf{P}_2(sk, w^{(\kappa)}, c^{(\kappa)}, st^{(\kappa)})$
    $\kappa := \kappa + 1$
  return $\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i \in [\kappa]}$
```

Fig. 13: Game G_1

This change is only internal to the oracles and the adversary's view remains identical to that of G_0.

Game G_2. Its only difference with Game G_1 is that we replace the randomness of the prover in GetTrans with a uniform function $RF : \{0,1\} \times \mathcal{M} \times [B] \rightarrow \mathcal{R}$ which is hidden from adversary's view, to derandomize GetTrans. Note that it depends on the message μ and number of the round in the rejection sampling to ensure uniqueness of the random coin with respect to them. It only changes the GetTrans subroutine. Further, the function GetTrans becomes a deterministic function with respect to the message μ. We use subscripts to emphasize on this fact in Figure 14. Although the signatures become deterministic, since we are only interested in UF-CMA$_1$ security, the adversary's view remains unchanged. The changes are depicted in Figure 14.

```
$\mathsf{GetTrans}(\mu)$ :
  $\kappa := 1,\ z_\mu^{(0)} := \perp$
  while $z_\mu^{(\kappa-1)} = \perp$ and $\kappa \leq B$
    $(w_\mu^{(\kappa)}, st_\mu^{(\kappa)}) := \mathsf{P}_1(sk; RF(0\|\mu\|\kappa))$
    $c_\mu^{(\kappa)} := H'(w_\mu^{(\kappa)} \| \mu)$
    $z_\mu^{(\kappa)} := \mathsf{P}_2(sk, w_\mu^{(\kappa)}, c_\mu^{(\kappa)}, st_\mu^{(\kappa)}; RF(1\|\mu\|\kappa))$
    $\kappa = \kappa + 1$
  return $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i \in [\kappa]}$
```

Fig. 14: Game G_2

Game G_3. In this game, described in Figure 15, we change the way that the random oracle queries are answered. Upon receiving an input $w\|\mu$, the oracle H queries the GetTrans function on input μ to receive a sequence of transcripts. Then if w is equal to one of the commitments in the transcripts, it returns its corresponding challenge. This is just a syntactic change and the adversary's view remains identical. The modifications can be seen in Figure 15.

```
$H(w\|\mu)$ :
    $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]} := \mathsf{GetTrans}(\mu)$
    if $\exists i : w = w_\mu^{(i)}$ return $c_\mu^{(i)}$
    return $H'(w\|\mu)$
```

Fig. 15: Game G_3

Game G_4. Let $\mathfrak{L}_\mu$ be the list of commitments generated for the message μ in the $\mathsf{GetTrans}(\mu)$ function. In this game, we modify $\mathsf{GetTrans}(\mu)$ such that if $\mathsf{Coll}(\mathfrak{L}_\mu)$ occurs, then it returns a special symbol Υ. We also change both Sign and H to return Υ if their call to GetTrans returns Υ. All these changes are reflected in Figure 16.

```
$\mathsf{Sign}(sk, \mu)$ :
    if $\mu \in \mathcal{M}$ return $\bot$
    $\mathcal{M} := \mathcal{M} \cup \{\mu\}$
    if $\mathsf{GetTrans}(\mu) = \Upsilon$ return $\Upsilon$
    $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]} := \mathsf{GetTrans}(\mu)$
    if $z_\mu^{(\kappa)} = \bot$ return $\bot$
    return $\sigma_\mu = (w_\mu^{(\kappa)}, z_\mu^{(\kappa)})$

$H(w\|\mu)$ :
    if $\mathsf{GetTrans}(\mu) = \Upsilon$ return $\Upsilon$
    $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]} := \mathsf{GetTrans}(\mu)$
    if $\exists i : w = w_\mu^{(i)}$ return $c_\mu^{(i)}$
    return $H'(w\|\mu)$

$\mathsf{GetTrans}(\mu)$ :
    $\kappa := 1$, $z_\mu^{(0)} := \bot$
    while $z_\mu^{(\kappa-1)} = \bot$ and $\kappa \le B$
        $(w_\mu^{(\kappa)}, st_\mu^{(\kappa)}) := \mathsf{P}_1(sk; RF(0\|\mu\|\kappa))$
        $c_\mu^{(\kappa)} := H'(w_\mu^{(\kappa)}\|\mu)$
        $z_\mu^{(\kappa)} := \mathsf{P}_2(sk, w_\mu^{(\kappa)}, c_\mu^{(\kappa)}, st_\mu^{(\kappa)}; RF(1\|\mu\|\kappa))$
        $\kappa := \kappa + 1$
    $\mathfrak{L}_\mu := \{w_\mu^{(i)}\}_{i\in[\kappa]}$
    if $\mathsf{Coll}(\mathfrak{L}_\mu)$ return $\Upsilon$
    return $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]}$
```

Fig. 16: Game G_4. The differences from Game G_3 are depicted in blue.

Let C be the concatenation $H\|\mathsf{GetTrans}$ that sends $w\|\mu$ to the bit-string $H(w\|\mu)\|\mathsf{GetTrans}(\mu)$. The queries of the adversary (both sign queries and random oracle queries) can be answered by using quantum queries to C. Therefore, without loss of generality, we assume that the adversary makes Q_S+Q_H quantum queries directly to the concatenation function. Let C_3 and C_4 be the concatenation functions in Game G_3 and Game G_4, respectively. If an adversary distinguishes G_3 from G_4, one can construct a wrapper around $\mathcal{A}$ distinguishing C_3 from C_4 since all the queries in the game can be simulated by the concatenation function as described above. They behave differently only on the inputs including a message that triggers Υ. Building on that, we use Lemma 13 to construct an algorithm $\mathcal{B}$ based on $\mathcal{A}$ which extracts a message μ triggering Υ as follows

$$\begin{aligned}&\left|\Pr[1 \leftarrow \mathcal{A}^{|\mathsf{C}_3\rangle}] - \Pr[1 \leftarrow \mathcal{A}^{|\mathsf{C}_4\rangle}]\right| \\ &\leq 2(Q_S + Q_H)\sqrt{\Pr[\mu \text{ triggers } \Upsilon \mid \mu \leftarrow \mathcal{B}^{|\mathsf{C}_3\rangle}]}.\end{aligned}$$

Now, note that C_3 never outputs Υ. In fact, for every $w\|\mu$, the value of $\mathsf{C}_3(w\|\mu)$ is independent from $\mathsf{Coll}(\mathfrak{L}_\mu)$. Therefore, algorithm $\mathcal{B}$ can do nothing except a totally random guess. For each message μ, the probability of $\mathsf{Coll}(\mathfrak{L}_\mu)$ can be bounded by Lemma 11. Hence we have

$$\left|\Pr[1 \Leftarrow G_4^{\mathcal{A}}] - \Pr[1 \Leftarrow G_3^{\mathcal{A}}]\right| \leq 2(Q_S + Q_H) \cdot B \cdot 2^{\frac{-\alpha-1}{2}}.$$

$\mathsf{Game}\ G_5$. In this game, we let the challenges $c_\mu^{(i)}$'s in the $\mathsf{GetTrans}$ function be produced as in the Σ-protocol without using the random oracle and sampled from the uniform distribution. To make $\mathsf{GetTrans}$ deterministic, we use a uniform function $RF' : \mathcal{M} \times [B] \to \mathcal{C}$ as a function to sample the challenges. The domain $\mathcal{M} \times [B]$ of the function suffices for our purpose since within the $\mathsf{UF\text{-}CMA}_1$ security the adversary is not allowed to query one message twice. Replacing the verifier V_1 with RF' is sufficient, since the identification protocol is public-coin. Note that both Sign and H change accordingly. Nevertheless, the distribution of $\mathsf{GetTrans}$, and consequently those of Sign and H, remains identical to that of the previous game thanks to the handling $\mathsf{Coll}(\mathfrak{L}_\mu)$. In this game the rounds of the rejection sampling are finally independent and each one has the same distribution as the real transcript in the Σ-protocol. All these changes are reflected in Figure 17.

```
GetTrans($\mu$) :
$\kappa := 1,\ z_\mu^{(0)} := \perp$
while $z_\mu^{(\kappa-1)} = \perp$ and $\kappa \leq B$
    $(w_\mu^{(\kappa)}, st_\mu^{(\kappa)}) := \mathsf{P}_1(sk; RF(0\|\mu\|\kappa))$
    $c_\mu^{(\kappa)} := RF'(\mu\|\kappa)$
    $z_\mu^{(\kappa)} := \mathsf{P}_2(sk, w_\mu^{(\kappa)}, c_\mu^{(\kappa)}, st_\mu^{(\kappa)}; RF(1\|\mu\|\kappa))$
    $\kappa := \kappa + 1$
$\mathfrak{L}_\mu := \{w_\mu^{(i)}\}_{i \in [\kappa]}$
if $\mathsf{Coll}(\mathfrak{L}_\mu)$ return $\Upsilon$
return $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i \in [\kappa]}$
```

Fig. 17: $\mathsf{Game}\ G_5$. The difference from $\mathsf{Game}\ G_4$ is depicted in blue.

$\mathsf{Game}\ G_6$. In this game, we replace the transcripts with the simulated ones in each round of $\mathsf{GetTrans}$. Let Sim be the zero-knowledge simulator of Σ. We use a new uniform function $RF'' : \mathcal{M} \times [B] \to \mathcal{R}$ as the randomness generator of Sim.

```
GetTrans($\mu$) :
$\kappa := 1$, $z_\mu^{(0)} := \perp$
while $z_\mu^{(\kappa-1)} = \perp$ and $\kappa \le B$
    $c_\mu^{(\kappa)} \leftarrow U(\mathcal{C})$
    $(w_\mu^{(\kappa)}, z_\mu^{(\kappa)}) := \mathsf{Sim}(vk, c_\mu^{(\kappa)}; RF''(\mu\|\kappa))$
    $\kappa := \kappa + 1$
$\mathfrak{L}_\mu := \{w_\mu^{(i)}\}_{i\in[\kappa]}$
if $\mathsf{Coll}(\mathfrak{L}_\mu)$ return $\Upsilon$
return $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]}$
```

Fig. 18: Game G_6. The difference from Game G_5 is depicted in blue.

Note that RF'' is not accessible by the adversary. Figure 18 updates GetTrans accordingly.

Let C_5 and C_6 be concatenation functions of $H\|\mathsf{GetTrans}$ in games G_5 and G_6. Without loss of generality, we allow the adversary to make $Q_S + Q_H$ direct quantum queries to them and is tasked to distinguish C_5 and C_6. The distribution of the outcomes of C_5 and C_6 are statistically (or computationally in the case of sc-HVZK) $B \cdot \varepsilon_{zk}$-far from each other. Plugging C_5 and C_6 into Lemma 14 implies

$$\left|\Pr[1 \Leftarrow G_6^{\mathcal{A}}] - \Pr[1 \Leftarrow G_5^{\mathcal{A}}]\right| \le (6Q_S + 6Q_H)^{\frac{3}{2}} \sqrt{B \cdot \varepsilon_{zk}}.$$

In the case of sc-HVZK, note that the distributions with which Lemma 14 is instantiated are indeed efficiently samplable, as the sc-HVZK definition lets the witness be known to the distinguisher.

Game G_7. In this game, we add one more condition for a valid signature in Line 6 of the game as shown in Figure 19. This step simplifies the reduction from the UF-NMA game.

An adversary $\mathcal{A}$ distinguishes G_7 from G_6 only if it can find $(m^*, (w^*, z^*))$ such that

$$H(w^*\|\mu^*) \neq H'(w^*\|\mu^*) \wedge [[\mu^* \notin \mathcal{M}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*).$$

In the $\mathsf{sUF\text{-}CMA}_1$ game, it changes to

$$H(w^*\|\mu^*) \neq H'(w^*\|\mu^*) \wedge [[(\mu^*, (w^*, z^*)) \notin \mathcal{MS}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*),$$

where $\mathcal{MS}$ denotes the message-signature pairs revealed to the adversary during the game. Let $\mathcal{R}$ be an algorithm that runs G_6 together with the adversary, observes its output $(\mu^*, (w^*, z^*))$, and returns (μ^*, w^*). We say that $\mathcal{R}$ wins if the above conditions hold. Note that the distinguishing advantage between G_6 and G_7 of the adversary is $\le \Pr[\mathsf{win}(\mathcal{R}^{G_6,\mathcal{A}})]$. We observe that

$$\left|\Pr[\mathsf{win}(\mathcal{R}^{G_6,\mathcal{A}})] - \Pr[\mathsf{win}(\mathcal{R}^{G_2,\mathcal{A}})]\right| \le \left|\Pr[1 \Leftarrow G_6^{\mathcal{A}}] - \Pr[1 \Leftarrow G_2^{\mathcal{A}}]\right| \ ,$$

from which it follows that

$$\Pr[\mathsf{win}(\mathcal{R}^{G_6,\mathcal{A}})] \le \left|\Pr[1 \Leftarrow G_6^{\mathcal{A}}] - \Pr[1 \Leftarrow G_2^{\mathcal{A}}]\right| + \Pr[\mathsf{win}(\mathcal{R}^{G_2,\mathcal{A}})] \ .$$

```
Game :
1: $\mathcal{M} := \varnothing$
2: $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
3: $(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{|H\rangle,\ \mathsf{Sign}(sk,\cdot)}(vk)$
4: Parse $\sigma^* = (w^*, z^*)$
5: $c^* := H(w^* \| \mu^*)$
6: if $c^* \neq H'(w^* \| \mu^*)$ return $0$
7: return $[[\mu^* \notin \mathcal{M}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$
```

```
Sign(sk, μ) :
1: if $\mu \in \mathcal{M}$ return $\perp$
2: $\mathcal{M} := \mathcal{M} \cup \{\mu\}$
3: if $\mathsf{GetTrans}(\mu) = \Upsilon$ return $\Upsilon$
4: $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i \in [\kappa]} := \mathsf{GetTrans}(\mu)$
5: if $z_\mu^{(\kappa)} = \perp$ return $\perp$
6: return $\sigma_\mu = (w_\mu^{(\kappa)}, z_\mu^{(\kappa)})$
```

```
H(w‖μ) :
1: if $\mathsf{GetTrans}(\mu) = \Upsilon$ return $\Upsilon$
2: $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i \in [\kappa]} := \mathsf{GetTrans}(\mu)$
3: if $\exists i (w = w_\mu^{(i)})$ return $c_\mu^{(i)}$
4: return $H'(w \| \mu)$
```

```
GetTrans(μ) :
1: $\kappa := 1$, $z_\mu^{(0)} := \perp$
2: while $z_\mu^{(\kappa-1)} = \perp$ and $\kappa \leq B$
3:     $c_\mu^{(\kappa)} \leftarrow U(\mathcal{C})$
4:     $(w_\mu^{(\kappa)}, z_\mu^{(\kappa)}) :=$
       $\mathsf{Sim}(vk, c_\mu^{(\kappa)}; RF''(\mu \| \kappa))$
5:     $\kappa := \kappa + 1$
6: $\mathfrak{L}_\mu := \{w_\mu^{(i)}\}_{i \in [\kappa]}$
7: if $\mathsf{Coll}(\mathfrak{L}_\mu)$ return $\Upsilon$
8: return $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i \in [\kappa]}$
```

Fig. 19: Game G_7. The difference from Game G_6 is depicted in blue.

In Game G_2 we always have $H(w^* \| \mu^*) = H'(w^* \| \mu^*)$. So, whatever the other conditions are, we have $\Pr[\mathsf{win}(\mathcal{R}^{G_2,\mathcal{A}})] = 0$. In total,

$$\left| \Pr[1 \Leftarrow G_7^{\mathcal{A}}] - \Pr[1 \Leftarrow G_6^{\mathcal{A}}] \right| \leq \left| \Pr[1 \Leftarrow G_6^{\mathcal{A}}] - \Pr[1 \Leftarrow G_2^{\mathcal{A}}] \right|.$$

It remains to reduce Game G_7 to the UF-NMA security of SIG_B. The adversary $\mathcal{B}^{|H'\rangle}$ can perfectly simulate the signing oracle and H for $\mathcal{A}$, and the random function RF' using another hash function G that is modeled as a random oracle. Whenever $\mathcal{A}$ outputs a forgery (μ^*, σ^*), it would also be a valid forgery for $\mathcal{B}$ and pass the verification thanks to Line 6 in Game G_7.

Runtime. For each of the signing or random oracle query, the reduction runs the HVZK simulator B times. To simulate the random function RF', one can use the private random oracle H' that is accessible to the reduction (It is also possible to replace H' with a quantum pseudorandom function). Therefore, the runtime of the reduction is essentially $\mathsf{Time}(\mathcal{A}) + T_{\mathsf{Sim}} \cdot B \cdot (Q_S + Q_H)$. □

To conclude this section, we claim that the above approach also extends to (s)UF-CMA security, as detailed in the following statement.

Theorem 10. *Let $\varepsilon_{zk}, \alpha \geq 0$, $B \geq 0$ and H and G hash functions modeled as random oracles. Assume that $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is an $(\varepsilon_{zk}, T_{\mathsf{Sim}})$-*HVZK *public-coin identification protocol, and that the commitment message of the prover has min-entropy α. For any quantum adversary $\mathcal{A}$ against* UF-CMA *security of $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ that issues at most Q_H quantum queries to the random oracle H and Q_S classical queries to the signing oracle, there exists another quantum adversary $\mathcal{B}$ against* UF-NMA *security of SIG_B with $\mathsf{Time}(B) \approx$*

$\mathsf{Time}(A) + T_{\mathsf{Sim}} \cdot B \cdot Q_S \cdot Q_H$ *such that*

$$\mathsf{Adv}_{\mathsf{SIG}_B}^{\mathsf{(s)UF\text{-}CMA}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathsf{SIG}_B}^{\mathsf{UF\text{-}NMA}}(\mathcal{B}) + 2^{\frac{-\alpha+3}{2}} \cdot B \cdot Q_S \cdot (Q_S + Q_H) \\ + 30\sqrt{\varepsilon_{zk} \cdot B} \cdot (Q_S + Q_H)^{\frac{3}{2}} \ .$$

Our reduction relies on $\mathcal{B}$ having access to a private random oracle H' with the same domain and range as H that is not accessible by $\mathcal{A}$.

Sketch of the proof. Since most of the proof is similar to that of Theorem 3, we just give a sketch. Consider an imaginary two-dimensional data structure (for example a table) that has $|\mathcal{M}|$ columns each one indexed by one message μ, that contains all the transcripts generated during the rejection sampling process in $\mathsf{Sign}(sk, \mu)$. Part of this table contains the view of the adversary. In the proof of Theorem 3, in the first two hybrid games we derandomized each cell of the data structure using a random function which takes as input the coordinate of the cell; its message and the row number (the iteration number). In the $\mathsf{UF\text{-}CMA}_1$ game, the adversary is supposed to choose Q_S columns (messages) and receive some information of each column (the signatures) and output a forgery. As long as the adversary is not allowed to query a message twice, this derandomization does not change the view of the adversary. This is not the case in the $\mathsf{UF\text{-}CMA}$ game. Moreover, we do not know the messages on which the adversary will query the signing oracle, and so we cannot assign appropriate randomness to the queries a priori. Instead, we consider a three-dimensional data structure such that each cell is uniquely determined by a message, an iteration number in $[B]$, and a query number in $[Q_S]$. One can see this three-dimensional table as the previous table that each column has expanded to Q_S columns. This new table contains the view of the adversary in the $\mathsf{UF\text{-}CMA}$ game and if we derandomize it with a random function that takes as input the coordinate of the cell, it does not change the view of the adversary. Now, the whole proof of Theorem 3 can be similarly repeated here with a small modification that each time we look into the two-dimensional table in the $\mathsf{UF\text{-}CMA}_1$ proof, we replace it with the three-dimensional one. We mention further details for the sake of completeness.

In the $\mathsf{UF\text{-}CMA}_1$ game, to consistently answer the random oracle query on input $w\|\mu$, we output some uniform element from the range of the function, unless the column indexed by μ contains a transcript with the commitment w in which case we output its corresponding challenge in the transcript. In the $\mathsf{UF\text{-}CMA}$ game, we search over the whole section of the message μ which contains roughly $B \cdot Q_S$ cells. This lookup in the table costs roughly $B \cdot Q_S$ operations.

In order to replace the real transcripts with the simulated ones, we take care of the collisions in the outputs of the random oracle (the challenges of the transcripts) in the table. This issue stems from the fact that in the simulated transcripts, all the challenges are replaced by fresh random elements. If there is any collision, they have to be updated accordingly. Recall that each challenge is evaluated as $H(w\|\mu)$. In the $\mathsf{UF\text{-}CMA}_1$ game, since there is no repeating message, the possible collisions only appear in the same column which has size at most B.

This probability of collision was captured in the fourth hybrid game in the proof of Theorem 3. In the $\mathsf{UF\text{-}CMA}$ game, the possible collisions are spread over the whole section of the message μ. One can update the fourth hybrid game accordingly and compute the probability of success similarly.

After handling the collisions, we change the real transcripts with simulated ones. The only issue that requires to be taken care of is that the forged signature $(\mu^*, (w^*, z^*))$ by the adversary must not intersect with the reprogrammed ones. The proof is similar to that of Theorem 3 in the last hybrid up to replacing the list $\mathfrak{L}_{\mu^*}$ which is the column indexed by μ^* with the whole section of the message μ^* in the three-dimensional table. □

B.2 FSwBA Security Analysis with the Rényi Divergence

As [DFPS22] mentions, defining a version of HVZK that relies on the Rényi divergence instead of the statistical distance allows to prove the security of a larger class of Fiat-Shamir signatures. In some cases, this allows to achieve smaller signature sizes. For the sake of simplicity, we restrict ourselves to the case of Rényi divergence of infinite order. We need the following definition.

Definition 12 (Decomposable Simulator). *Let $p \in [0, 1]$. Let Sim be a zero-knowledge simulator for a Σ-protocol. We say that Sim admits a p-decomposition if there exist two algorithms $\mathsf{Sim}_\perp$ and $\mathsf{Sim}_{\not\perp}$ such that the former only outputs transcripts with $z = \perp$, the latter only outputs transcripts with $z \neq \perp$, and Sim can be defined as in Figure 20*

```
Sim(x) :
with probability $p$
  $(w, c, z) \leftarrow \mathsf{Sim}_\perp(x)$
with probability $1 - p$
  $(w, c, z) \leftarrow \mathsf{Sim}_{\not\perp}(x)$
return $(w, c, z)$
```

Fig. 20: Simulator decomposition.

One can verify that our simulator in Secion 4 for Lyubashevsky's Σ-protocol satisfies the above definition. With this formalism, we are able to extend the HVZK definition to the Rényi divergence.

Definition 13 (Decomposable Divergence HVZK). *Let $R_{zk} \geq 1, \varepsilon_{zk} > 0, p \in [0, 1]$ and $T_\perp, T_{\not\perp} \geq 0$. A Σ-protocol is said to be $(\varepsilon_{zk}, T_\perp, R_{zk}, T_{\not\perp})$-$\mathsf{DDHVZK}$ if there exists a p-decomposable simulator $\mathsf{Sim} = (\mathsf{Sim}_\perp, \mathsf{Sim}_{\not\perp})$ such that*

- *algorithm $\mathsf{Sim}_\perp$ is $(\varepsilon_{zk}, T_\perp)$-$\mathsf{HVZK}$ (or $\mathsf{sc\text{-}HVZK}$) simulator for the Σ-protocol transcript (w', c', z') conditioned on $z' = \perp$,*

- *algorithm* $\mathsf{Sim}_{\not\perp}$ *has runtime* $T_{\not\perp}$*, and given* x *outputs a transcript* (w, c, z) *such that its distribution and the one of a transcript* (w', c', z') *of the* Σ*-protocol conditioned on* $z' \neq \perp$ *are such that*

$$R_\infty\Big((w, c, z) \| (w', c', z')\Big) \leq R_{zk} \enspace .$$

Note that p can possibly differ from β, but we are interested in the case where their difference is negligible (as in the following theorem). We adapt Theorem 4 and its proof to this new setting.

Theorem 11. *Let* $R_{zk} \geq 1, \varepsilon_{zk}, T_\perp, T_{\not\perp} \geq 0, p \in [0, 1]$ *and* H *a hash function modeled as a random oracle. Assume that* $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ *is an* $(\varepsilon_{zk}, T_\perp, R_{zk}, T_{\not\perp})$-$\mathsf{DDHVZK}$ *public-coin identification protocol with a p-decomposable simulator and probability of aborting* β*, then we have the following updates on Theorem 4.*

- *In the ROM, there exists an adversary* $\mathcal{B}$ *against* $\mathsf{UF\text{-}NMA}$ *security of* SIG_B *with runtime* $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_\perp(B-1)Q_S + T_{\not\perp}Q_S)\log(B \cdot Q_S + Q_H))$ *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{X\text{-}CMA}}_{\mathsf{SIG}_B}(\mathcal{A}) \leq R_{zk}^{Q_S} \cdot (\mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) + (\varepsilon_{zk} + |p - \beta|) \cdot B \cdot Q_S + \varDelta_{\mathsf{X}}) \\ + 2^{-\alpha} \cdot B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1) \enspace .\end{aligned}$$

- *In the QROM, there exists an adversary* $\mathcal{B}$ *against* $\mathsf{UF\text{-}NMA}$ *security of* SIG_B *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{X\text{-}CMA}}_{\mathsf{SIG}_B}(\mathcal{A}) \leq R_{zk}^{Q_S} \cdot (\mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) + (\varepsilon_{zk} + |p - \beta|) \cdot B \cdot Q_S + \varDelta_{\mathsf{X}}) \\ + 2^{-\frac{\alpha}{2}} \cdot \frac{3B \cdot Q_S}{2} \cdot \sqrt{(B \cdot Q_S + Q_H + 1)} \enspace .\end{aligned}$$

  Our reduction relies on $\mathcal{B}$ *having access to a private random oracle* H' *with the same domain and range as* H *that is not accessible by* $\mathcal{A}$*. Furthermore, the runtime of* $\mathcal{B}$ *is* $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_\perp(B-1)Q_S + T_{\not\perp}Q_S)\log(B \cdot Q_S))$ *with QRACM, and* $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_\perp(B-1)Q_S + T_{\not\perp}Q_S) \cdot (B \cdot Q_S))$ *without QRACM.*

Proof. The proof is almost identical to the one of Theorem 4, where we replace $\mathsf{Game}\ G_2$ with three different games $G_{2.1}, G_{2.2}$ and $G_{2.3}$. The other changes between games remain similar. Let $\mathsf{Sim} = (\mathsf{Sim}_\perp, \mathsf{Sim}_{\not\perp})$ be the decomposition of the zero-knowledge simulator. We proceed as follows.

$\mathsf{Game}\ G_1$. It is the same as in the proof of Theorem 4.

$\mathsf{Game}\ G_{2.1}$ In this game, we change the signing algorithm. As soon as a transcript (w, c, z) with $z \neq \perp$ is being sampled during the rejection sampling loop, we discard it and replace it with a transcript generated by $\mathsf{Sim}_{\not\perp}$. The multiplicativity of the Rényi divergence implies that

$$\Pr[1 \Leftarrow G_1^{\mathcal{A}}] \leq (1 + \varepsilon_{zk})^{Q_S} \cdot \Pr[1 \Leftarrow G_{2.1}^{\mathcal{A}}].$$

Game $G_{2.2}$. We modify the signing algorithm one step further. Let $\mathsf{Bernoulli}(\beta)$ denote the Bernoulli distribution with parameter β (i.e., the probability of sampling 1 is β). We replace the honestly generated transcripts with the following distribution. Sample $b \leftarrow \mathsf{Bernouli}(\beta)$ and $c \leftarrow U(\mathcal{C})$. If $b = 1$ run $(w, z) \leftarrow \mathsf{Sim}_{\perp}(pk, c)$, and if $b = 0$ run $(w, z) \leftarrow \mathsf{Sim}_{\not\perp}(pk, c)$. Since the transcripts are being sampled independently from each other in both games $G_{2.1}$ and $G_{2.2}$, one can bound the advantage of the distinguisher by $\varepsilon_{zk} \cdot (B - 1) \cdot Q_S$.

Game $G_{2.3}$. We replace $\mathsf{Bernouli}(\beta)$ with $\mathsf{Bernouli}(p)$. The distinguishing advantage of the adversary between $G_{2.2}$ and $G_{2.3}$ would be less than $|p - \beta| \cdot (B - 1) \cdot Q_S$.

The rest of the proof is similar to that of Theorem 4. □