New Security Proofs and Techniques for Hash-and-Sign with Retry Signature Schemes

Benoît Cogliati[1], Pierre-Alain Fouque[2], Louis Goubin[3], Brice Minaud[4]

[1]Thales DIS France SAS [2]Université de Rennes [3]Laboratoire de Mathématiques de Versailles, UVSQ, CNRS, Université Paris-Saclay, France [4]École Normale Supérieure, PSL University, CNRS, Inria, France

Abstract. Hash-and-Sign with Retry is a popular technique to design efficient signature schemes from code-based or multivariate assumptions. Contrary to Hash-and-Sign signatures based on preimage-sampleable functions as defined by Gentry, Peikert and Vaikuntanathan (STOC 2008), trapdoor functions in code-based and multivariate schemes are not surjective. Therefore, the standard approach uses random trials. Kosuge and Xagawa (PKC 2024) coined it the Hash-and-Sign with Retry paradigm.

As many attacks have appeared on code-based and multivariate schemes, we think it is important for the ongoing NIST competition to look at the security proofs of these schemes. The original proof of Sakumoto, Shirai, and Hiwatari (PQCrypto 2011) was flawed, then corrected by Chatterjee, Das and Pandit (INDOCRYPT 2022). The fix is still not sufficient, as it only works for very large finite fields. A new proof in the Quantum ROM model was proposed by Kosuge and Xagawa (PKC 2024), but it is rather loose, even when restricted to the classical setting.

In this paper, we introduce several tools that yield tighter security bounds for Hash-and-Sign with Retry signatures in the classical setting. These include the Hellinger distance, stochastic dominance arguments, and a new combinatorial tool to transform a proof in the non-adaptative setting to the adaptative setting. Ultimately, we obtain a sharp bound for the security of Hash-and-Sign with Retry signatures, applicable to various code-based and multivariate schemes. Focusing on NIST candidates, we apply these results to the MAYO, PROV, and modified UOV signature schemes. In most cases, our bounds are tight enough to apply with the real parameters of those schemes; in some cases, smaller parameters would suffice.

1 Introduction

The provable security of post-quantum schemes is a real issue. While in the nineties, the security of RSA and ECC-based schemes have been shown to rely on well-known hard problems, the situation is different for lattice-based schemes. Although reductions to lattice problems exist in order to rule out all attacks, reductions are not used to derive parameters, as is done in the concrete security proof of RSA-OAEP and RSA-PSS [BR95,BR96] for instance. The reason is that these reductions would require very large parameters, and the schemes would not be as efficient. The preferred approach is to analyze the efficiency of practical attacks, and use it to set parameters based on the best known lattice algorithms. As a consequence, parameters need to be carefully tested, and small changes in the design can sometimes disproportionately impact its security, as illustrated by some Falcon variants [ETWY22,DEP23].

The situation is even worse when we look at the case of multivariate cryptography. Many real attacks on these schemes have been proposed, which undermine the confidence in this alternative. HFE variants and very attractive designs such as Rainbow have recently been attacked [Beu21,TPD21,BBC+22,Beu22a]. This does not mean that the underlying problem is easy, but in order to propose efficient schemes compared to lattice cryptosystems for instance, cryptographers sometimes make aggressive choices in the design. This has led to NIST launching a new call for additional digital signature proposals: contrary to

key-encapsulation mechanisms, no remaining signature candidates are considered for the fourth round. Moreover, for certain applications, such as certificate transparency, short signatures with fast verification are required. Multivariate cryptography is a good candidate to propose very short signatures, around 100B compared to more than 600B for Falcon [FHK+18]. Ten out of the 40 candidates of the new NIST call rely on multivariate problems, making it one of the most popular paradigms.

The main drawback of code-based and multivariate cryptography is the size of the public key, which is sometimes prohibitive (except for MAYO [BCC+23]). Most of these signature candidates (including [BCC+23,BCD+23,PCF+23,FIKT21]) rely on the Unbalanced Oil-and-Vinegar construction, UOV for short [KPG99]. The plain UOV problem seems to be more resistant than HFE variants or Rainbow. However, once we trust the security of the underlying hard problems, we have to look at the security of the schemes. In this article, our goal is to carefully analyze the security of these schemes and provide a rigorous and sharper analysis of their security proofs.

Related Work. The first security proof for a UOV variant has been given by Sakumoto, Shirai and Hiwatari [SSH11]. More recently, a subtle flaw has been discovered by Chatterjee, Das, and Pandit in [CDP22], who proposed another security proof to avoid the problem. The main drawback of the new proof is that the security bound depends on the size of the finite field, which needs to be super-polynomial. This is unfortunately not the case for modern UOV parameters [BCH+23], or any NIST candidate. Consequently, a new security proof is needed. In [Beu22b], Beullens proposed a proof for MAYO, but this proof falls short of giving the required security for the concrete parameters given in the MAYO NIST submission [BCC+23]. The problem is that the security proof required that the number of signatures q_{sig} times some bound B is less than 1, and this is not the case for some MAYO parameters. Finally, we can note that Kosuge and Xagawa provide a quantum proof in [KX22] as well as a new framework to analyze what they define as Hash-and-Sign with Retry, in order to capture many code-based or multivariate-based signature schemes. This new framework is a modification of the GPV framework [GPV08] when the functions are not surjective. Moreover, they present quantum security proof for these schemes.

Our Contributions. We build on the framework of [KX22] to propose a new security proof for Hash-and-Sign with Retry signatures in the classical setting. This proof provides sharper bounds than (a dequantumized variant) of the analysis of [KX22]: the dominant factor is roughly reduced from q^2/N to $\sqrt{q}/N$, where q is the total number of queries to the signing and hashing oracles, and N is the size of the salt space.

In the process, we bring several new techniques into the area of multivariate cryptography. The first is the Hellinger distance, essentially a special case of Rényi divergence. While Rényi divergences are commonly used in the analaysis of lattice-based schemes [BLL+15,Pre17,PGMP19], the Hellinger distance specifically has recently emerged as a promising technique in other cryptographic contexts [Yas21,Lee24]. We also introduce stochastic dominance arguments, which enable a game hop-like step that allows to replace a distribution in a distinguishing game with another distribution (Lemma 13)—even though it falls outside the framework of standard game-based proofs. Finally, we introduce a technique to transform a proof in the non-adaptive setting to the adaptive setting, using a new combinatorial tool. This tool ultimately yields a bound in the adaptive setting, with no loss in the reduction compared to the original non-adaptive proof (see the technical overview for more information, and Appendix E for the full details). We believe this last part to be of independent interest.

In Section 5, we show applications of our results to several post-quantum NIST candidates. In particular, our bounds are tight enough to derive practical security proofs for mutivariate schemes such as MAYO and PROV, applicable either with their real choices of parameters, or close variants.

2 Technical Overview

2.1 Overview of Hash-and-Sign with Retry

Before we explain our techniques, it is useful to recall the Hash-and-Sign with Retry paradigm. In signature schemes that follow this paradigm, the public key defines a trapdoor function. When proving the security of the scheme, the underlying hardness assumption is that, without knowing the trapdoor, it is difficult to invert the function. Typically, for the code-based and multivariate schemes we are interested in, this assumption can be decomposed into two separate assumptions: first, the function defined by the public key is hard to distinguish from a certain class of random functions; second, inverting a function sampled from that class is difficult.

Concretely, for code-based schemes, the relevant class of random functions is typically noisy encoding for a random code (or a random code with special features, such as a random cyclic code). For a UOV-based scheme, the relevant class is typically the map defined by a system of random multivariate quadratic equations. In both cases, the first assumption is that the public key is hard to distinguish from such a system. The second assumption is, respectively for code-based and for multivariate cryptosystems, that it is hard to decode a random code, and that it is hard to solve a random system of quadratic equations. In both cases, the second assumption amounts to the average-case hardness of a well-studied NP-hard problem.

If the adversary only has access to the public key, then those two assumptions are enough to establish the security of the scheme. However, in the usual EUF-CMA security model, security must be proved when the adversary has access not only to the public key, but also to a signature oracle. The crux of the proof is to show that the signature oracle does not help the adversary. Typically, this is done by showing that the signature oracle can be simulated without knowledge of the secret trapdoor.

Recall that for code-based and multivariate signatures, the trapdoor function defined by the public key is in general not surjective. Moreover, in the Hash-and-Sign with Retry paradigm, the inverse trapdoor function may be probabilistic. For a given choice of random coins for the inverse trapdoor function, and a given message hash, a preimage of the hash may not exist. At a high level, the signature algorithm proceeds by resampling the coins used in the inverse of the trapdoor function until a preimage exists. (Alternatively, the signature algorithm may also resample the message hash by using a salt, to the same effect.) The name "with retry" comes from this resampling process. The hash preimage is output as a signature for the message (together with the salt, if a salt was used).

In the ROM, a simple way to simulate the signature oracle is to choose a preimage uniformly at random, apply the public-key trapdoor function to that preimage, then program the random oracle so that the hash of the message matches the output of the trapdoor function. This approach comes with two caveats.

- First, this process may not yield the same signature distribution as a real signature. In fact, in the case of UOV signatures, the two distributions are quite far apart (it can be shown that their statistical distance is lower-bounded by an absolute constant, cf. Appendix B). This explains why, despite UOV being the dominant approach for modern multivariate signatures, standard UOV signatures do *not* have a security proof.
- Second, when programming the random oracle, there is no guarantee that the programmed value is uniformly distributed. Hence, the proof must also argue that the programmed oracle remains statistically close to uniform. Although the problem is subtle, addressing the first issue is not enough to settle the second one, as noted in [CDP22]. We defer a complete treatment to Section 5, and remain at a high level for now.

In [SSH11], Sakumoto *et al.* introduce a clever way to address the first issue by specifying a resampling process that guarantees that the two distributions are statistically close (or in fact, identical, in

the case of UOV-based systems). However, their analysis fails to account for the second issue. This was pointed out by Chatterjee *et al.* in [CDP22], who proposed a workaround. However, their solution comes at a significant cost in performance. For more details, we refer the reader to Section 5.

Here, we follow the same approach as [SSH11,KX22] to address the first issue. The core of our contributions is to address the second, remaining issue, *i.e.* we show that the modified random oracle remains statistically close to uniform.

2.2 Our Techniques

Following [SSH11,CDP22,KX22], we investigate an approach for signatures in the Hash-and-Sign with Retry framework that enables provable security. New to this work is that we aim to analyze the security of this approach in depth, in order to derive sharp bounds that are directly applicable to the real-world parameters of NIST submissions. Towards that end, we introduce several new techniques. The purpose of this section is to provide a high-level view of them.

In the provable approach of [SSH11,CDP22,KX22], the signature algorithm proceeds by first fixing the random coins used to invert the trapdoor function defined by the public key, then resampling the salted hash of the message (by changing the salt), until it falls within the range of the trapdoor function. The signature is then the preimage of the message hash. Moreover, we focus on the relevant case where the signature algorithm is deterministic (employed in most schemes based on Hash-and-Sign with Retry, such as NIST candidates UOV, MAYO, PROV). That is, for a given message, the random coins used to invert the trapdoor function are determined by the message and secret key.

In that setting, we say that a salt is *suitable* for a given message if the salted hash of the message lies within the image of the trapdoor function (when using the random coins fixed by the message). Using a sequence of standard game hops, we show that the security of the signature scheme reduces to the security of the following distinguishing game: given an oracle that takes as input a message-salt pair, and inputs 1 if the salt is suitable for the message, 0 otherwise, no adversary can distinguish the real world (where the suitability oracle follows the distribution of real signatures) from the ideal world (where the suitability oracle follows the distribution of simulated signatures). We do not elaborate on that argument here, and focus instead on the rest of the proof.

Stochastic dominance and replacement lemma. In the ideal world, for each message, one random salt is "forced" to be suitable by reprogramming the random oracle. As a result, for each message, the suitability oracle outputs 1 slightly more often in the ideal world than in the real world. The goal of the analysis is to show that the two worlds remain statistically indistinguishable. The main obstacle is that, in the real world, answers of the suitability oracle to distinct queries are not independent, which makes the analysis quite complex[1].

To circumvent that issue, we first observe that if we modify the suitability oracle to answer 1 "less often" (in a specific sense), then an optimal adversary trying to distinguish the two worlds is more likely to guess that it lives in the real world. (At a rough intuitive level, this makes sense, since the real-world suitability oracle answers 1 slightly less often.) We want to use this observation to argue that if we *replace* the real-world distribution of the suitability oracle to answer 1 "less often", but leave the ideal-world distribution untouched, then this can only increase the adversary's advantage.

To formalize that idea of "less often", we need a partial order over binary *distributions*. For that purpose, we use the notion of stochastic dominance. Given an arbitrary partial order over a set Ω, stochastic dominance defines a partial order on *distributions* over Ω. Using that notion, the above

[1] Depending on the exact definition of the suitability oracle, the answers of the oracle to distinct queries are not independent either in the real world, or in the ideal world.

intuitive idea can be expressed cleanly. This is captured in a key *replacement lemma* (Lemma 13), which is stated at a high level of generality: if a distribution Q *covers* a distribution P (Definition 15), and P stochastically dominates another distribution R, then $\Delta(P,Q) \leq \Delta(R,Q)$, where $\Delta()$ denotes the statistical distance. Hence, if our goal is to upper-bound $\Delta(P,Q)$, then we are free to "replace" P by R.

The point of this technique is that it allows us to replace the real-world distribution of the suitability oracle by any distribution that is lower with respect to the stochastic dominance order. We use this to replace it with a distribution where the oracle's answers to distinct queries are independent. In summary, we show that it is possible to modify the real-world distribution by making it answer 1 slightly less often, in such a way that the oracle's answers become pairwise independent (when defining each answer as a random variable); all the while ensuring that the adversary's advantage can only increase, thanks to the replacement lemma. After the replacement, oracle answers are independent in both the ideal and the real world.

Hellinger distance. At this point, the adversary is tasked with distinguishing between two oracles, both of which answer each query by sampling independent Bernoulli variables. (Recall that a Bernoulli variable $\mathsf{Ber}[p]$ is a binary random variable equal to 1 with probability p, and 0 with probability $1-p$.) The advantage of the adversary can then be analyzed using standard techniques for that setting, namely Rényi divergences. Rényi divergences are a family of maps that take as input two distributions, and output a measure of "how close" the two distributions are. They play an important role in lattice-based cryptography [BLL+15,Pre17,PGMP19]. The Rényi divergence of parameter 1, also called the Kullback-Leibler divergence, is the most commonly used. Here, we use the Hellinger distance instead, which corresponds (up to composition with a fixed function) to the Rényi divergence of parameter 1/2. The Hellinger distance has appeared in various cryptographic settings, especially recently [Ste12,Yas21,Lee24], and turns out to yield a sharper bound than Kullback-Leibler in our context.

To see why the Hellinger distance is helpful in our setting, consider the simplified case where the adversary is trying to distinguish between a vector of q independent Bernoulli variables $\mathsf{Ber}[p]$, and a vector of q independent Bernoulli variables $\mathsf{Ber}[p']$, where $p \neq p'$ are two constants. A naïve hybrid argument shows that the statistical distance between the two vectors is $\mathcal{O}(q)$. In reality, an analysis based on the Hellinger distance (or the Kullback-Leibler divergence) shows that it is $\mathcal{O}\left(\sqrt{q}\right)$ (which is tight). Intuitively, this is because the hybrid argument fails to take advantage of the independence between each vector entry: the hybrid argument would remain valid even if the variables were not independent. To contrast, the Hellinger distance is (sub-)additive over sequences of independent variables: it naturally accounts for independence.

In the end, the Hellinger distance allows to derive a security bound for Hash-and-Sign with Retry signatures that grows *sublinearly* with the number of hash queries, which is inherently impossible to prove without going beyond standard hybrid arguments.

Adaptive to non-adaptive reduction. The proof sketched so far only holds in the non-adaptive setting, where the adversary's queries are fixed in advance. It does not hold in the adaptive setting, where the adversary's queries may depend on the answers to past queries. The main culprit is the replacement lemma. We want to apply the replacement lemma to the vector of answers provided by the suitability oracle to the adversary's queries. Unfortunately, when we replace one distribution with another, in the adaptive setting, the adversary's queries may change. As a result, the new answers may be completely unrelated to the original ones.

Deducing adaptive security from a non-adaptive proof is notoriously thorny. Generic techniques such as complexity leveraging can incur huge reduction factors. We adopt a different approach.

In our setting, the adversary queries a suitability oracle. Suppose that the adversary issues q queries, chosen among a total of N possible distinct queries. The oracle provides binary answers. The answer to each possible query can be viewed as a Bernoulli variable (which is not, in general, independent from the variable associated with a different query). The situation can be formalized as a *box-opening game* (Definition 17): there are N closed boxes, each containing a Bernoulli variable. The vector of N Bernoulli variables contained in the boxes follows one of two possible distributions, corresponding to a real and ideal worlds. The adversary is allowed to open q boxes of her choice. She must then guess whether the variables were sampled from the real or the ideal distributions.

Suppose that the vector of N Bernoulli variables defining (say) the real-world distribution is *permutation-invariant* (Definition 16), in the sense that if we arbitrarily permute the variables, the distribution of the vector remains unchanged. If this is true for both the real and ideal worlds, then which boxes the adversary chooses to open is irrelevant, since the distribution of answers will be the same. In that case, an adaptive adversary performs no better than a non-adaptive one. This yields a case where a very simple adaptive-to-non-adaptive reduction is possible, with no loss in the reduction. Unfortunately, for the actual suitability oracle we wish to analyze, the relevant distributions are not permutation-invariant. Let us then set this idea aside for now; we will come back to it later.

We introduce the idea of *replacement game* (Definition 18). In a replacement game, each world is defined by a *pair* of vectors of N Bernoulli variables. The first vector is called the original vector, and the second vector is called the replacement vector. (The distributions of the two vectors are not in general independent.) A replacement game starts as a normal box-opening game, where the content of the boxes correspond to the original vector. Once the adversary has finished opening boxes however, the contents of the boxes are replaced by the replacement vector. Then the adversary is forced to make a guess based only on the final contents of the boxes. (This can be formalized concisely, see Definition 18.) A box-opening game is the special case of a replacement game where the original and replacement vectors are equal (with probability 1), in both worlds.

To circumvent issues that arise when we use the replacement lemma to modify a distribution, we consider replacement games, and only ever modify the distribution of replacement vectors. This means that the distribution of adversarial queries is not affected. As a result, we show that we can essentially reuse the analysis of the non-adaptive case. Like in the non-adaptive case, we reach a state where the replacement vectors are independent Bernoulli variables. Moreover, with slightly more effort, we can arrange that the replacement distributions are *identically distributed* independent Bernoulli variables (Lemma 29), in both worlds.

At that point, the replacement distributions are permutation-invariant. This means that the adversary's queries are irrelevant, exactly in the same sense as the simple permutation-invariant argument sketched earlier. Hence an adaptive adversary performs no better than a non-adaptive one. Furthermore, the distribution of original vectors is irrelevant, and the adversary's advantage is equal to the advantage of a non-adaptive adversary playing a standard box-opening game between the two replacement distributions (Lemma 25). We can then directly reuse the last part of the analysis of the non-adaptive case (involving the Hellinger distance), which concludes the analysis.

While this adaptive-to-non-adaptive reduction is not generic, the underlying ideas are somewhat general: if we decouple the distribution used by the adversary to compute queries (previously, "original distribution") from the distribution used to compute the final guess (previously, "replacement distribution"), then we can alter the latter distribution without touching the former, essentially with the same freedom as if we were in the non-adaptive case, since the adversary's queries are not affected by the changes. While this decoupling modifies the nature of the game, if at the outcome of the reasoning, the replacement distributions become permutation-independent, then the game collapses back into a normal game with a single distribution per world. Moreover, in the final game, non-adaptive and adaptive adversaries coincide.

While this approach requires some care to avoid subtle issues related to adaptivity, in the end, it allows us to derive a bound in the adaptive setting that matches the non-adaptive bound exactly. There is no loss in the reduction. Moreover, instead of having to write a new proof, we are able to reuse most of the analysis of the non-adaptive case, with a few (key) additional arguments.

3 Preliminaries

3.1 Notation

We use the notation $[a, b]_{\mathbb{N}}$ for the integer interval $\{a, \ldots, b\}$; and $[a, b]_{\mathbb{R}}$ for the real interval $\{x \in \mathbb{R} : a \leq x \leq b\}$. All distributions in this work are over finite sets. A probability distribution P over a (finite) set Ω is a map $P : \Omega \to [0, 1]_{\mathbb{R}}$ such that $\sum_{\omega \in \Omega} P(\omega) = 1$. Given an event $E \subseteq \Omega$, $P(E) = \sum_{e \in E} P(e)$ is the probability of E according to P. Note that $P(\{\omega\}) = P(\omega)$. If $X \in \Omega$ is a random variable, we write p_X for the probability distribution of X (implicitly over the ambient space Ω). If P is a distribution, we write $X \sim P$ if the distribution of X is P, i.e. $\mathsf{p}_X = P$. Given a distribution P over the reals, $\mathbb{E}[P]$ denotes the expectation of P. The statistical distance between two distributions, denoted by $\Delta(\cdot, \cdot)$, is $\Delta(A, B) = (1/2) \cdot \sum_{\omega} |A(\omega) - B(\omega)| = \max_{E \subseteq \Omega} |A(E) - B(E)| = \max_{E \subseteq \Omega} A(E) - B(E)$. Given A a probabilistic algorithm, $A(x; r)$ denotes the output of A on input x, with random tape r.

3.2 Distributions

Let $\mathsf{Ber}[p]$ denote the Bernoulli distribution over $\{0, 1\}$, defined by $\mathsf{Ber}[p](1) = p$. Let $\mathsf{Bin}[q, p]$ denote the binomial distribution with q trials, each with probability of success p. That is, the distribution of the sum of q independent random variables, each distributed according to $\mathsf{Ber}[p]$.

Operations on distributions. If P and Q are two distributions over Ω, $P+Q$ denotes the distribution of $X + Y$, where $X \sim P$ and $Y \sim Q$ are two independent random variables. The tensor product of distributions is defined as follows. If P and Q are two distributions over Ω_1 and Ω_2 respectively, $P \otimes Q$ denotes the distribution of (X, Y), where $X \sim P$ and $Y \sim Q$ are two independent random variables. Equivalently, $P \otimes Q$ is the distribution over $\Omega_1 \times \Omega_2$ defined by:

$$(P \otimes Q)(\omega_1, \omega_2) = P(\omega_1)Q(\omega_2).$$

Hypergeometric distribution. Given $n, x, y \in \mathbb{N}$ with $x, y \leq n$, the hypergeometric distribution $\mathsf{Hyp}[n, x, y]$ is defined as follows. Consider a pool of n balls, of which x are distinguished. Then $\mathsf{Hyp}[n, x, y]$ denotes the number of distinguished balls among y balls picked uniformly at random out of the pool, without replacement. By convention, let $\mathsf{Hyp}[\cdot, -1, \cdot] = -1$ (i.e. for all n, y, $\mathsf{Hyp}[n, -1, y]$ takes the value -1 with probability 1). A few properties of the hypergeometric distribution that will be useful are listed next.

- $\mathsf{Hyp}[n, x, y] = \mathsf{Hyp}[n, y, x]$. This is because the distribution can be equivalently described as picking x balls at random among n, then y balls at random among the same n, then outputing the number of balls that were picked both times.
- $\mathsf{Hyp}[n, x, y](k) = \binom{x}{k}\binom{n-x}{y-k}\binom{n}{y}^{-1} = \binom{y}{k}\binom{n-y}{x-k}\binom{n}{x}^{-1}$.
- $\mathsf{Hyp}[N, \mathsf{Bin}[N, p], q] = \mathsf{Bin}[q, p]$. This is because having $\mathsf{Bin}[N, p]$ distinguished balls among N total balls amounts to having each individual ball be distinguished with probability p, independently. Hence the number of distinguished balls among the q picked balls is $\mathsf{Bin}[q, p]$.

3.3 Stochastic Dominance

Definition 1 (Coupling). *Given two distributions P over Ω_1, and Q over Ω_2, a* coupling *of P and Q is a pair of random variables (X, Y) in $\Omega_1 \times \Omega_2$ such that $X \sim P$, and $Y \sim Q$.*

Note that there exist in general many couplings (X, Y) of P and Q. Typically, for "interesting" couplings, X and Y are not independent.

We restrict the following definition to finite sets to avoid issues of measurability, and because it suffices for our purpose. Given a partial order $\leq_\Omega$ on some set Ω, stochastic dominance defines a partial order $\preceq_\Omega$ on distributions over Ω.

Definition 2 (Stochastic dominance). *Let $(\Omega, \leq_\Omega)$ be a finite partially ordered set, and let P, Q be two distributions over Ω. Q is said to* stochastically dominate *P with respect to $\leq_\Omega$, written $P \preceq_\Omega Q$, if any one of the following equivalent conditions is fulfilled.*

1. *There exists a coupling (X, Y) of P and Q such that $\Pr[X \leq_\Omega Y] = 1$.*
2. *For any $S \subseteq \Omega$ that is closed upwards (with respect to $\leq_\Omega$), $Q(S) \geq P(S)$. (S is said to be* closed upwards *with respect to $\leq_\Omega$ if for all $s \in S$, for all $t \in \Omega$, $s \leq_\Omega t$ implies $t \in S$.)*
3. *For any $\leq_\Omega$-increasing map $f : \Omega \to \mathbb{R}$, $\mathbb{E}[f(P)] \leq \mathbb{E}[f(Q)]$. (We use "increasing" in the non-strict sense: f is $\leq_\Omega$-increasing if for all $x \leq_\Omega y$ in Ω, $f(x) \leq f(y)$ in $\mathbb{R}$.)*

Remark 1. We state the equivalence between the three definitions for completeness, but in this work, we will only use $1 \Rightarrow 2$, which is straightforward. The implication $3 \Rightarrow 1$ is sometimes called Strassen's theorem.

Throughout this document, $\preceq$ denotes the stochastic dominance order that arises from $\leq$ (over, say, $\mathbb{Z}$), and $\preceq_n$ denotes the stochastic dominance order that arises from the usual component-wise order $\leq_n$ on $\mathbb{Z}^n$.

Definition 3 (Stochastically increasing). *Given two partial orders $(\Omega_1, \leq_1)$ and $(\Omega_2, \leq_2)$, and a probabilistic map $F : \Omega_1 \to \Omega_2$, we say that F is* stochastically increasing *if any one of the following equivalent statements holds.*

1. *Given any two elements $x \leq_1 y$ in Ω_1, $F(x) \preceq_2 F(y)$.*
2. *Given any two distributions $P \preceq_1 Q$ over Ω_1, $F(P) \preceq_2 F(Q)$.*

Note on terminology. In the literature, the term "stochastic dominance" is sometimes used for specific choices of the underlying order $\leq_\Omega$, most commonly $\Omega = \mathbb{R}$ and $\leq_\Omega$ is the usual order over the reals. Here, we use the term "stochastic dominance" in the most general sense, where $\preceq_\Omega$ arises from an arbitrary partial order $\leq_\Omega$.

3.4 Security Notions

In this subsection, we recall standard notions of security for signature schemes, preimage-sampleable function, and hash-and-sign with retry.

Definition 4 (Digital Signature Scheme). *A Digital Signature Scheme S consists of three algorithms:*

- *S.KeyGeneration: this algorithm takes as input a security parameter λ, given as 1^λ, and outputs a public key pk and a secret key sk;*

– $S.\textsc{Sign}$: *this algorithm takes as input a secret key* sk *and a message* $\mathsf{msg} \in \mathcal{M}$*, and outputs a signature* $\mathsf{sig} = (\mathsf{salt}, s)$*, where* $\mathsf{salt} \in \mathcal{R}$*;*
– $S.\textsc{Verify}$: *this algorithm takes as input a public key* pk*, a message* msg*, and a signature* sig *and outputs a bit.*

The algorithms $S.\textsc{Sign}$ and $S.\textsc{Verify}$ have oracle access to a hash function $\mathcal{H} : \mathcal{R} \times \mathcal{M} \to \{0,1\}^{f(\lambda)}$. We define the security of signature schemes using the standard security notion of Existential Unforgeability under Chosen Message Attack. Some proofs will be in the *non-adaptive case*, i.e. where all the queries are decided before the interactions with the signing oracle.

Definition 5 (EUF-CMA security). *Let* $\mathcal{H}$ *be a random oracle, and let* $\mathcal{A}$ *be an adversary. The advantage of* $\mathcal{A}$ *against the* EUF-CMA *security of a signature scheme* $S = (S.\textsc{KeyGeneration}, S.\textsc{Sign}^{\mathcal{H}}, S.\textsc{Verify}^{\mathcal{H}})$ *is defined as*

$$\mathsf{Adv}_S^{\mathsf{EUF\text{-}CMA}}(\mathcal{A}) = \Pr[S.\textsc{Verify}(\mathsf{pk}, \mathsf{msg}, \mathsf{sig}) = \top],$$

where $(\mathsf{pk}, \mathsf{sk}) \leftarrow S.\textsc{KeyGeneration}()$*,* $(\mathsf{msg}, \mathsf{sig}) \leftarrow \mathcal{A}^{\mathcal{H}, S.\textsc{Sign}^{\mathcal{H}}(\mathsf{sk}, \cdot)}(\mathsf{pk})$*, and* $S.\textsc{Sign}^{\mathcal{H}}(\mathsf{sk}, \cdot)$ *was never queried on* msg*.*

Algorithm 1 The Hash-and-Sign with Retry paradigm based on a PSF T.

procedure $S.\textsc{KeyGeneration}(1^\lambda)$
  $(\mathrm{F}, \mathrm{I}) \leftarrow \mathsf{T}.\textsc{Gen}(1^\lambda)$
  $\mathsf{sk} \leftarrow \mathrm{I}$
  $\mathsf{pk} \leftarrow \mathrm{F}$
  return $(\mathsf{sk}, \mathsf{pk})$

procedure $S.\textsc{Verify}^{\mathcal{H}}(\mathsf{pk}, \mathsf{msg}, \mathsf{sig})$
  $(\mathsf{salt}, s) \leftarrow \mathsf{sig}$
  return $\mathrm{F}(s) = \mathcal{H}(\mathsf{msg} || \mathsf{salt})$

procedure $S.\textsc{Sign}(\mathsf{sk}, \mathsf{msg})$
  $z \leftarrow \mathrm{I}^1()$
  repeat
    $\mathsf{salt} \leftarrow_\$ \mathcal{R}$
    $s \leftarrow \mathrm{I}^2(z, \mathcal{H}(\mathsf{salt}, \mathsf{msg}))$
  until $s \neq \perp$
  return (salt, s)

Next, we define preimage-sampleable functions and their security. In algorithm 1, we give a formal description of the Hash-and-Sign with Retry paradigm, that turns a preimage-sampleable function into a signature scheme.

Remark 2. Compared to [KX22], we slightly modify the terminology around preimage-sampleable function. First, we split the inverse trapdoor I into a pair $(\mathrm{I}_1, \mathrm{I}_2)$, which is necessary to use the techniques of [SSH11]. Second, in [KX22], the term *weak preimage-sampleable* is used to describe any tuple of algorithms with the appropriate inputs and outputs, whereas *preimage-sampleable* is reserved for the case that those algorithms fulfill some correctness properties. Here, the first type of tuple is simply called *preimage-sampleable*, and we give a correctness definition separately, rather than baking it into the definition. This follows common usage in cryptography, and allows greater flexibility, by making it possible to talk about different flavors of correctness (for instance, computational or statistical correctness).

Definition 6 (Preimage-Sampleable Function (PSF)). *A PSF* T *consists of four algorithms:*

– $\textsc{Gen}$: *this algorithm takes as input a security parameter and outputs a function* $\mathrm{F} : \mathcal{X} \to \mathcal{Y}$ *with a trapdoor* I*;*

– F*: this algorithm takes as input a value* $x \in \mathcal{X}$ *and deterministically outputs* $\mathrm{F}(x)$*;*
– $\mathrm{I} = (\mathrm{I}^1, \mathrm{I}^2)$*: the first algorithm takes no input and samples a value* $z \in \mathcal{Z}$*; the second algorithm takes as input* $z \in \mathcal{Z}$*,* $y \in \mathcal{Y}$*, and outputs* $x \in \mathcal{X}$ *or* $\perp$*.*
– SampDom*: this algorithm takes as input* $\mathrm{F} : \mathcal{X} \to \mathcal{Y}$ *and outputs* $x \in \mathcal{X}$*.*

Definition 7 (Correctness of a PSF). *Let* T *be a PSF.* T *is* correct *if for any* $(\mathrm{F}, \mathrm{I}) \leftarrow \textsc{Gen}(\lambda)$*, the following conditions hold.*

1. $\mathrm{F}(x)$ *is uniform over* $\mathcal{Y}$ *for* $x \leftarrow \textsc{SampDom}(\mathrm{F})$*.*
2. *For all* $y \in \mathcal{Y}$*, given* $z \leftarrow \mathrm{I}_1()$*,* $\mathrm{I}_2(z, y)$ *outputs either* $\perp$*, or* x *satisfying* $\mathrm{F}(x) = y$*.*
3. *For all* $y \in \mathcal{Y}$*, given* $z \leftarrow \mathrm{I}_1()$*, the distribution of* $\mathrm{I}_2(z, y)$ *(over the random coins of* I_1 *and* I_2*) is equal to the distribution of* $x \leftarrow \textsc{SampDom}(\mathrm{F})$ *conditioned on* $\mathrm{F}(x) = y$*.*

Definition 8 (Security of a PSF (PS security) [KX22]). *Let* T *be a PSF. The advantage of an adversary* $\mathcal{A}$ *against the* PS *security of* T *is defined as follows:*

$$\mathsf{Adv}^{\mathsf{PS}}_{T}(\mathcal{A}) = \left|\Pr\left[\mathsf{PS}_0^{\mathcal{A}} = 1\right] - \Pr\left[\mathsf{PS}_1^{\mathcal{A}} = 1\right]\right|,$$

where PS_0 *and* PS_1 *are the games defined in Algorithm 2.*

Algorithm 2 Preimage sampling game.

procedure PS_b
  $(\mathrm{F}, \mathrm{I}) \leftarrow \textsc{Gen}(1^\lambda)$
  $b^* \leftarrow \mathcal{A}^{\mathsf{Sample}_b}(\mathrm{F})$
  return b^*

procedure Sample_1
  $x_i \leftarrow \textsc{SampDom}(\mathrm{F})$
  return x_i

procedure Sample_0
  $z_i \leftarrow \mathrm{I}^1()$
  repeat
    $y_i \leftarrow_{\$} \mathcal{Y}$
    $x_i \leftarrow \mathrm{I}^2(z_i, y_i)$
  until $x_i \neq \perp$
  return x_i

Definition 9 (INV security). *Let* $\mathcal{A}$ *be an* INV *adversary against* T*, trying to invert the public function* F*. We define its advantage as* $\mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}}(\mathcal{A}) = \Pr[\mathrm{F}(x) = y]$*, with* $(\mathrm{F}, \cdot) \leftarrow \textsc{Gen}(1^\lambda)$ *and* $y \leftarrow_{\$} \mathcal{Y}$*,* $x \leftarrow \mathcal{A}(\mathrm{F}, y)$*.*

4 New Security Proof for Hash-and-Sign with Retry

Our proof targets signatures following the Hash-and-Sign with Retry framework, as defined by Algorithm 1. Our goal is to prove the security bound given in Theorem 1.

Because we target the deterministic setting, where the random coins used to compute signatures are (pseudo-randomly) fixed by the message and secret key, it is convenient to introduce some additional notation. In the algorithms below, r_1, r_2, etc. are pseudo-random values generated from the secret key and the input message. Formally, we define $r_i = \mathcal{H}_{\mathsf{rand}}(\mathsf{sk} \| \mathsf{msg} \| i)$. The RO used to hash the message and salt is denoted by $\mathcal{H}$. The $\mathcal{H}_{\mathsf{rand}}, \mathcal{H}_{\mathsf{salt}}, \mathcal{H}_{\mathsf{s}}$ are hash functions modeled as uniformly random functions, but they do not need to be ROs. (They are only here to write the derandomized version of the signature algorithm.) It is assumed the signature algorithm is a hash-and-sign with retries, and $\mathrm{I}_1, \mathrm{I}_2, \mathrm{F}$, are the functions defining the underlying preimage-sampleable TDF.

Algorithm 3 Real and ideal signature algorithms.

```
procedure REAL-SIGN(msg)
    v ← I1(; r1)
    i ← 1
    repeat
        salt ← r2i
        s ← I2(v, Hreal(msg‖salt); r2i+1)
        i ← i + 1
    until s ≠ ⊥
    return (salt, s)

procedure Hreal(x)
    if x has already been determined then
        return previous output for x
    else
        return fresh uniform value
```

```
procedure IDEAL-SIGN(msg)
    salt ← Hsalt(sk, msg)
    s ← SAMPDOM(; Hs(sk, msg))
    Hideal(msg‖salt) ← F(s)              ▷ Program RO
    return (salt, s)

procedure Hideal(x)
    Parse x as msg‖ salt
    IDEAL-SIGN(msg)                      ▷ Discard output
    if x has already been determined then
        return previous output for x
    else
        return fresh uniform value
```

Real world. The real world is the real signature algorithm, in its derandomized variant, given in Figure 3.

Ideal world. In the ideal world, the signature algorithm is replaced by the ideal variant in Figure 3. That is, signing is done by first choosing the salt and signature $\mathbf{s}$ (pseudo-)randomly, then programming the RO so that $\mathcal{H}(\mathsf{salt}\|\mathsf{msg}) = \mathrm{F}(\mathbf{s})$.

Random oracle. In both worlds, $\mathcal{H}$ is modeled as a RO. Because the RO behaves slightly differently between the two worlds, we define $\mathcal{H}_{\mathsf{real}}$ and $\mathcal{H}_{\mathsf{ideal}}$ separately in Algorithm 3. When queried on an input, if the corresponding output has already been fixed in a previous step, $\mathcal{H}$ sends back the same output. If it has not been fixed yet, a fresh uniformly random output is sent back, and subsequent queries to the same input will yield the same output. This behavior is the same in both worlds. The only difference in the RO between the two worlds is that in the ideal world, the signature algorithm programs one value of the RO for each message. In addition, in the ideal world, any call to the random oracle on input $\mathsf{msg}\|\mathsf{salt}$ first triggers a signature call IDEAL-SIGN(msg), whose output is discarded. Even though the output is discarded, the call has an impact because IDEAL-SIGN(msg) programs the value of $\mathcal{H}_{\mathsf{ideal}}(\mathsf{msg}\|\cdot)$ on one salt input.

An important remark is that while $\mathcal{H}_{\mathsf{real}}$ is a proper random oracle (it is a uniformly random function), this is not the case of $\mathcal{H}_{\mathsf{ideal}}$. Indeed, some output values are programmed by IDEAL-SIGN, and those values are not uniformly random (when conditioned on the adversary's view): their distributions depend on key material. The core of the proof below will be to show that this is not a problem. More precisely, the main object of the proof will be to show that, even though $\mathcal{H}_{\mathsf{ideal}}$ is not *perfectly* uniformly random, it is *statistically* indistinguishable from a uniformly random function, given the adversary's view, up to a negligible quantity.

Main theorem. First, we need some notion expressing the fact that the preimage-sampleable TDF (specifically, I_2) does not return $\perp$ too often. This will help us to state the bound.

Definition 10. *We say that a preimage-sampleable TDF is (f, ε)-well-behaved for the signature scheme if it satisfies the following properties:*

- *In the main loop of the real signature algorithm I_2 returns $\boldsymbol{s} \neq \perp$ with probability $p = 1$, or some probability $p \leq 1/2$, over the randomness of the salt.*
- *Except with probability ε, over the choice of $\mathbf{v}$, i.e. the randomness of r_1, $p \geq f$.*

Intuitively, f cannot be too small, because (except with probability ε), the main loop of the real signature scheme may have to iterate $1/f$ times before finding a valid signature in rare cases. The requirement $p \notin]1/2, 1[$ is not absolutely required, but simplifies the bound, and holds true for our applications.

We present our main result in two parts. First, Proposition 1 assumes that the preimage-sampleable function is perfectly secure, and focuses on bounding the advantage of an adversary trying to distinguish the ideal world from the real world. Second, Theorem 1 gives a bound on the advantage on an adversary against the EUF-CMA-security of the Hash-and-Sign with Retry construction. Theorem 1 follows immediately from Proposition 1. Indeed, in short, once we have a bound on the adversary's ability to distinguish the two worlds, it suffices to bound the advantage of the adversary in the ideal world. Thus, Proposition 1 contains the core of our argument, while Theorem 1 is the relevant result in terms of applications.

Proposition 1. *Consider an adversary trying to distinguish the two worlds, making q_{sign} queries to the signing oracle, and $q_{\mathcal{H}}$ queries the random oracle $\mathcal{H}$. Assume the underlying preimage-sampleable TDF is correct, (f, ε)-well-behaved, and that it is perfectly secure: the advantage of any adversary in* PS *security game (Definition 8) is zero. Let $q = q_{\mathsf{sign}} + q_{\mathcal{H}}$. Let $N = 2^{\mathsf{len}_{\mathsf{salt}}}$ be the cardinality of the salt space. Then the advantage of the adversary is bounded by:*

$$\mathcal{O}\left(\frac{\log(f^{-1})}{\sqrt{f}} \cdot \frac{\sqrt{q}}{N}\right) + qe^{-\Omega(fN)} + q\varepsilon.$$

Theorem 1. *Let T be a correct, (f, ε)-well-behaved PSF, and let $\mathsf{HaS}_{\mathsf{T}}$ be the instantiation of the Hash-and-Sign with Retry construction using T as the trapdoor. Let $\mathcal{A}$ be an adversary against the* EUF-CMA*-security of $\mathsf{HaS}_{\mathsf{T}}$ that issues at most $q_{\mathcal{H}}$ random oracle queries, q_{sign} signature queries, and runs in time at most t. Then, there exists an adversary $\mathcal{B}$ against the* PS*-security of T and an adversary $\mathcal{C}$ against its* INV*-security such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{HaS}_{\mathsf{T}}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{PS}}_{\mathsf{T}}(\mathcal{B}) + q\varepsilon + q_h \mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}}(\mathcal{C}) + \frac{1}{|\mathcal{Y}|} \\ + \mathcal{O}\left(\frac{\log(f^{-1})}{\sqrt{f}} \cdot \frac{\sqrt{q}}{N}\right) + qe^{-\Omega(fN)}.\end{aligned}$$

where $q = q_{\mathcal{H}} + q_{\mathsf{sign}}$. Besides, $\mathcal{B}$ (resp. $\mathcal{C}$) runs in time $t' = t + \mathcal{O}(q_{\mathsf{sign}} + q_h)$ (resp. $t'' = t + (q_{\mathcal{H}} + q_{\mathsf{sign}} + 1)(t_{\mathsf{T}} + O(1))$ where t_{T} is an upper-bound on the running time to evaluate the T.F function). Moreover, $\mathcal{B}$ is allowed at most $q_h + q_{\mathsf{sign}}$ queries.

Theorem 1 is a corollary of Proposition 1, and follows using standard arguments, see for example [Beu22b, Proof of Lemma 8]. We provide a proof in Appendix D for completeness.

4.1 Proof of Proposition 1

For simplicity, we are going to assume that all pseudo-random-values r_i used for derandomization are perfectly random. In reality, they are generated from a seed via a PRF (or whichever other primitive

is used to generate the pseudo-random values), and we need some game hops to first replace them by truly uniform values. In the final bound, this would add a term for the distinguishing advantage against the PRF. The same goes for $\mathcal{H}_{\mathsf{salt}}$ and $\mathcal{H}_{\mathbf{s}}$: we are going to assume they are uniformly random functions. Note that we do not require those hashes to be ROs: they can be replaced by PRFs. Treating them as perfectly random avoids adding PRF terms everywhere, and does not affect the argument otherwise.

Let G_{real} denote the real-world game, where the signature algorithm is instantiated by Real-Sign in Algorithm 3. Let G_{ideal} denote the ideal-world game, where the signature algorithm is instantiated by Ideal-Sign in Algorithm 3.

Hybrid 0. Hybrid 0 is the ideal game $G_0 := G_{\mathsf{ideal}}$.

Hybrid 1. In the ideal game, the signature $\mathbf{s}$ is sampled using SampDom. The security of the preimage-sampleable TDF implies that this is the same as first sampling $\mathbf{v} \leftarrow \mathrm{I}_1$, then sampling an image $\mathbf{y} \leftarrow_{\$} \mathcal{R}$ uniformly at random in the range $\mathcal{R}$ of F, and finally sampling the signature as $\mathbf{s} \leftarrow \mathrm{I}_2(\mathbf{v}, \mathbf{y})$. (In that statement, we are treating the random tapes of all algorithms involved as uniformly random; in other words, what we are saying is that the outputs of the two processes yield identical distributions of $\mathbf{s}$ when the hash functions/r_i's are uniformly random).

Let G_1 (hybrid 1) denote the ideal game, except Ideal-Sign is replaced by Ideal-Sign$_1$: Per the

Algorithm 4 Real and ideal signatures in G_1 (differences with G_0 highlighted).

```
1: procedure Real-Sign(msg)
2:     v ← I1(; r1)
3:     i ← 1
4:     repeat
5:         salt ← r2i
6:         s ← I2(v, Hreal(msg‖salt); r2i+1)
7:         i ← i + 1
8:     until s ≠ ⊥
9:     return (salt, s)
```

```
1: procedure Hreal(x)
2:     if x has already been determined then
3:         return previous output for x
4:     else
5:         return fresh uniform value
```

```
1: procedure Ideal-Sign1(msg)
2: v ← I1(; r1)
3:     i ← 1
4:     repeat
5:         y ←$ R   using random coins r2i
6:         s ← I2(v, y; r2i+1)
7:         i ← i + 1
8:     until s ≠ ⊥
9:     salt ← r0
10:    Hideal(msg‖salt) ← y                ▷ Program RO
11:    return (salt, s)
```

```
1: procedure Hideal(x)
2:     Parse x as msg‖ salt
3:     Ideal-Sign(msg)                     ▷ Discard output
4:     if x has already been determined then
5:         return previous output for x
6:     else
7:         return fresh uniform value
```

discussion above, G_0 and G_1 are identically distributed from the adversary's perspective.

For $\mathbf{v}$ an output of I_1, let us denote by $\mathcal{R}(\mathbf{v})$ the set of $\mathbf{y}$'s such that $\mathrm{I}_2(\mathbf{v}, \mathbf{y}) \neq \perp$. (The output of I_2 depends on a random tape, but whether it outputs $\perp$ does not depend on the random tape as implied by Definition 7, so the previous sentence is well-defined.) For information, note that for a given $\mathbf{v}$, the probability p from the earlier definition of well-behaved is $|\mathcal{R}(\mathbf{v})|/|\mathcal{R}|$.

Hybrid 2. Fix a message msg (and hence a vector $\mathbf{v}$). For a salt s, define $f(s) = \mathcal{H}(\mathsf{msg}\|s)$, the hash value obtained for this choice of salt. Say that the salt s is *suitable* if $f(s) \in \mathcal{R}(\mathbf{v})$. In other words, suitable salts (for a given message) are salts for which a signature is possible (I_2 will not return $\perp$).

In the discussion that follows, we are picturing the random oracle $\mathcal{H}$ as having predefined (uniformly random) values for all inputs. So for each message, a salt is either suitable, or it is not, depending on that predefined image. When the RO is programmed on some input, we are overwriting the predefined value for that input. Importantly, we are careful to always program the oracle on a given input *before* the input has a chance to be queried. Otherwise, the behavior of the oracle could change during the game, which is not allowed. This is the role of the first two lines of $\mathcal{H}_{\mathsf{ideal}}$ in Algorithm 3: if the hash input is parsed as $\mathsf{msg}|\mathsf{salt}$, Ideal-Sign($\mathsf{msg}$) is called immediately for that message, to make sure the RO programming for msg has occurred. Note that the programming for a given message is always the same, due to the derandomized nature of the algorithm. (As an alternative, we could have equivalently called Ideal-Sign(msg) on every possible message at the beginning of the game.)

Ideal-Sign$_1$ programs the RO $\mathcal{H}$ so that $\mathcal{H}(\mathsf{msg}\|\mathsf{salt})$ is a uniform element of $\mathcal{R}(\mathbf{v})$, for one uniform salt. That is, Ideal-Sign$_1$ forces one random salt to be suitable. Afterwards, Ideal-Sign$_1$ picks that salt to sign the message. A priori, there may exist other salt values s that are suitable (for the same message). Suppose that *after* programming the RO to force the chosen value salt to be suitable, Ideal-Sign$_1$ does not pick salt for the signature, but instead picks a uniformly random salt among suitable salts (which could still be the programmed salt, but could also be another salt if other suitable salts exist).

A key observation is that this modification of Ideal-Sign$_1$ makes no difference to the output distribution of the algorithm. Indeed, after the RO is programmed, there is nothing special about the programmed salt: it is one of potentially many suitable salts, all of which follow the same distribution (for all of them, $\mathcal{H}(\mathsf{msg}\|\mathsf{salt})$ is uniform in $\mathcal{R}(\mathbf{v})$). Another key observation is that the real-world signature algorithm picks a uniformly random suitable salt to sign the message.

Combining these two observations, we arrive at this conclusion: the ideal signature can be modified as follows. First, program the RO as in the previous hybrid. Afterwards, instead of choosing the programmed salt to sign the message, call the real-world signature algorithm to choose a uniform salt among suitable salts. The previous discussion implies that this modification does not change the output distribution of the ideal signature algorithm.

Let G_2 denote the ideal game G_{ideal}, except Ideal-Sign is replaced by Ideal-Sign$_2$, which works as explained above. Pseudo-code is given in Algorithm 5.

In Algorithm 5, we have refactored the real-world signature algorithm Sign by introducing a Sign-v subroutine, but the actual algorithm is unchanged. (Implicitly, when Sign-v calls $\mathcal{H}$, it calls $\mathcal{H}_{\mathsf{real}}$ when called in the real world, and $\mathcal{H}_{\mathsf{ideal}}$ when called in the ideal world.) Meanwhile, Ideal-sign$_2$ is identical to Ideal-sign$_1$, except for the very last line: we now return a signature computed as in the real world. Per the earlier discussion, the output of Ideal-sign$_2$ is distributed identically to Ideal-sign$_1$, and hence also to the original ideal-world signature Ideal-sign.

Hybrid 3. Hybrid G_3 is the same as G_2, except the first part of Ideal-Sign$_2$ (the part that programs the RO) is moved from the Ideal-Sign routine to the RO $\mathcal{H}_{\mathsf{ideal}}$, as depicted in Algorithm 6.

Again, the games G_2 and G_3 are identical: we have simply moved the random oracle programming from Ideal-Sign to $\mathcal{H}_{\mathsf{ideal}}$, using the same computation. Note that Ideal-Sign does the programming on every call, but for a given message msg, all computed values are the same, so $\mathcal{H}_{\mathsf{ideal}}$ could equivalently do the programming only once per new msg input.

Hybrid 4. So far, all hybrids are identically distributed. We have merely rewritten the ideal game G_{ideal} in various equivalent ways. The point is that in the latest hybrid G_3, Ideal-Sign$_3$ is identical

Algorithm 5 Real and ideal signatures in G_2 (differences with G_1 highlighted).

procedure REAL-SIGN$_2$(msg)
  $\mathbf{v} \leftarrow \mathrm{I}_1(; r_1)$
  return SIGN-V($\mathbf{v}$, msg, $\mathcal{H}_{\mathsf{real}}$)

procedure SIGN-V($\mathbf{v}$, msg, $\mathcal{H}$)
  $i \leftarrow 1$
  repeat
    salt $\leftarrow r_{2i}$
    $\mathbf{s} \leftarrow \mathrm{I}_2(\mathbf{v}, \mathcal{H}(\mathsf{msg}\|\mathsf{salt}); r_{2i+1})$
    $i \leftarrow i + 1$
  until $\mathbf{s} \neq \perp$
  return (salt, $\mathbf{s}$)

procedure $\mathcal{H}_{\mathsf{real}}(x)$
  if x has already been determined **then**
    return previous output for x
  else
    return fresh uniform value

procedure IDEAL-SIGN$_2$(msg)
  $\mathbf{v} \leftarrow \mathrm{I}_1(; r_1)$
  $i \leftarrow 1$
  repeat
    $\mathbf{y} \leftarrow_{\$} \mathcal{R}$ using random coins r_{2i}
    $\mathbf{s} \leftarrow \mathrm{I}_2(\mathbf{v}, \mathbf{y}; r_{2i+1})$
    $i \leftarrow i + 1$
  until $\mathbf{s} \neq \perp$
  salt $\leftarrow r_0$
  $\mathcal{H}(\mathsf{msg}\|\mathsf{salt}) \leftarrow \mathbf{y}$ ▷ Program RO
  return SIGN-V($\mathbf{v}$, msg, $\mathcal{H}_{\mathsf{ideal}}$)

procedure $\mathcal{H}_{\mathsf{ideal}}(x)$
  Parse x as msg‖ salt
  IDEAL-SIGN(msg) ▷ Discard output
  if x has already been determined **then**
    return previous output for x
  else
    return fresh uniform value

Algorithm 6 Ideal signature in G_3 (differences with G_2 highlighted).

procedure IDEAL-SIGN$_3$(msg)
  $\mathbf{v} \leftarrow \mathrm{I}_1(; r_1)$
  return SIGN-V($\mathbf{v}$, msg, $\mathcal{H}_{\mathsf{ideal3}}$)

procedure SIGN-V($\mathbf{v}$, msg, $\mathcal{H}$)
  $i \leftarrow 1$
  repeat
    salt $\leftarrow r_{2i}$
    $\mathbf{s} \leftarrow \mathrm{I}_2(\mathbf{v}, \mathcal{H}(\mathsf{msg}\|\mathsf{salt}); r_{2i+1})$
    $i \leftarrow i + 1$
  until $\mathbf{s} \neq \perp$
  return (salt, $\mathbf{s}$)

procedure $\mathcal{H}_{\mathsf{ideal3}}(x)$
  Parse x as msg‖salt ▷ other forms of input are irrelevant
  $\mathbf{v} \leftarrow \mathrm{I}_1(; r_1)$ ▷ is the same $\mathbf{v}$ from IDEAL-SIGN because r_1 is the same
  $i \leftarrow 1$
  repeat
    $\mathbf{y} \leftarrow_{\$} \mathcal{R}$ using random coins r_{2i}
    $\mathbf{s} \leftarrow \mathrm{I}_2(\mathbf{v}, \mathbf{y}; r_{2i+1})$
    $i \leftarrow i + 1$
  until $\mathbf{s} \neq \perp$
  salt $\leftarrow r_0$
  $\mathcal{H}(\mathsf{msg}\|\mathsf{salt}) \leftarrow \mathbf{y}$ ▷ Program RO
  if x has already been determined **then**
    return previous output for x
  else
    return fresh uniform value

to the real signature algorithm. The only difference between G_3 and G_{real} is in the RO $\mathcal{H}$. In fact, the last part of $\mathcal{H}_{\mathsf{ideal}3}$ is identical to $\mathcal{H}_{\mathsf{real}}$: the specific difference lies in the random oracle programming that occurs in the first part of $\mathcal{H}_{\mathsf{ideal}3}$.

Hybrid 4 is the real game $G_4 = G_{\mathsf{real}}$. As noted, the difference between G_3 and G_{real} lies entirely in the RO. The rest of the proof is dedicated to bounding the advantage of an adversary trying to distinguish G_3 from $G_4 = G_{\mathsf{real}}$ (that is, essentially, an adversary trying to distinguish $\mathcal{H}_{\mathsf{real}}$ from $\mathcal{H}_{\mathsf{ideal}3}$).

4.2 Arguing about the Indistinguishability of Real and Ideal RO

The case of a single message. In the remainder, we focus on bounding the advantage of an adversary trying to distinguish G_3 from $G_4 = G_{\mathsf{real}}$. Towards that end, fix a message msg, and let us look at the adversary's view. For now, let us focus on queries relevant to msg: that is, signatures on msg, and hash queries of the form $\mathsf{msg}\|s$ for some salt s. "Suitable" will mean suitable for msg.

The only difference between G_3 and G_4 lies in the random oracles $\mathcal{H}_{\mathsf{real}}$ versus $\mathcal{H}_{\mathsf{ideal}3}$. Moreover, the two random oracles are identical, except for the fact that $\mathcal{H}_{\mathsf{ideal}3}$ programs one uniformly random salt value to be suitable. Beyond that, the two worlds behave identically. There are two ways for the adversary to access the hash oracle.

1. The adversary can query the signature oracle, which reveals a salt salt that is suitable.
2. The adversary can directly query the hash oracle.

We are going to assume that prior to querying the hash oracle on $\mathsf{msg}\|s$, the adversary always first queries the signature oracle on msg (increasing the number of signature queries as necessary), and learns both the chosen salt, and its hash image. Thus, the adversary always knows one suitable salt value (and its image). Moreover, there is now only one way for the adversary to access the hash oracle, which is to query it directly. Without loss of generality, we assume the adversary always queries salt values that are pairwise distinct, since querying the same value is useless. We also assume the adversary does not query the chosen salt from the signature of msg, since the hash value for that salt is already known. Call that salt the *special* salt. Summing up, from now on, the adversary only issues hash oracle queries, on salts that are pairwise distinct, and also distinct from the special salt. (Still focusing on a single message msg.)

Let $p = |\mathcal{R}(\mathbf{v})|/|\mathcal{R}|$ be the probability that a salt is suitable for msg, where $\mathbf{v}$ is determined by msg. Let us look at the distribution of the number of suitable salts observed by the adversary after performing q hash oracle queries for msg, and not counting the special salt. (We do not care whether the adversary is able to recognize which salts are suitable, we are simply counting the number of suitable salts among salts queried by the adversary).

- In the ideal world, the special salt is automatically suitable. Other salts have a probability p of being suitable, independently for each salt. Thus, the number of suitable salts observed by the adversary after q queries (not counting the special salt) is distributed according to $\mathsf{Bin}[q, p]$. (Where $\mathsf{Bin}[q, p]$ is the binomial distribution defined by the number of successful coin flips after q independent coin flips, each with a probability of success p.)
- In the ideal world, the situation is more complex. The special salt is suitable, except in the case that there exists no suitable salt at all, which happens with probability $(1-p)^N$. The *total* number of suitable salts across all N possible salts is distributed according to $\mathsf{Bin}[N, p]$. However, one suitable salt (if it exists) was reserved for the special salt, so the number of suitable salts among salts that the adversary can query is $\mathsf{Bin}[N, p] - 1$ (where the special value -1 corresponds to the case that no suitable salt exists). (The notation $\mathsf{Bin}[N, p] - 1$ means the distribution of $X - 1$, where X

is a random variable $X \sim \mathsf{Bin}[N,p]$.) Thus, the adversary issues q queries among $N-1$ possible salts, of which $\mathsf{Bin}[N,p]-1$ are suitable. Recall that the hypergeometric distribution $\mathsf{Hyp}[n,d,q]$ is the number of distinguished balls after drawing q balls from a pool of n balls, of which d are distinguished (without replacement). In conclusion, the number of suitable salts observed by the adversary in the real world is $\mathsf{Hyp}[N-1,\mathsf{Bin}[N,p]-1,q]$ (with the convention $\mathsf{Hyp}[\cdot,-1,\cdot]=-1$, corresponding to the special case where there was no suitable salt at all, even for the special salt).[2]

Suppose that the adversary observes k suitable salts among the q queried salts (more precisely, if we include the special salt, the number of suitable salts is $1+k$; the value $k=-1$ is thus reserved for the special case where even the special salt is not suitable). The key point is the following. If we condition the randomness of either game on the adversary observing exactly k suitable salts, then the distribution of the adversary's view is the same in both games. Indeed, which k salts are suitable among the q queried salts is uniformly random among the $\binom{q}{k}$ possibilities, in both worlds. Moreover, images of suitable salts are uniformly random (independently) in $\mathcal{R}(\mathbf{v})$, again in both worlds. Likewise, images of unsuitable salts are uniformly random (independently) in $\mathcal{R}\setminus\mathcal{R}(\mathbf{v})$. Hence, the adversary's view in both worlds is distributed identically when the number of suitable salts is fixed.

To put this differently, there exists a probabilistic map F that takes as input the number of suitable salts observed by the adversary, and outputs the adversary's view conditioned on having observed that number of salts; and that map F is the same in both worlds. This implies that the statistical distance between the adversary's view in both worlds is bounded by the statistical distance between the distributions of the number of observed suitable salts. Indeed, recall that the statistical distance Δ satisfies $\Delta(F(A),F(B)) \le \Delta(A,B)$ for any pair of distributions A,B, and any probabilistic map F (sometimes called the data-processing inequality). Since the adversary's view in G_3 is $F(\mathsf{Bin}[q,p])$, and the adversary's view in G_4 is $F(\mathsf{Hyp}[N-1,\mathsf{Bin}[N,p]-1,q])$, the statistical distance between the adversary's view in the two worlds is bounded by:

$$\Delta(\mathsf{Bin}[q,p],\mathsf{Hyp}[N-1,\mathsf{Bin}[N,p]-1,q]).$$

Multiple messages. So far, we have acted as if the adversary had to spend all q queries on a single message msg. In reality, the adversary may split their q queries among $m \le q$ distinct messages. For now, for simplicity, we focus on a *selective* adversary: that is, the adversary chooses in advance which message-salt pair to query, without adapting new queries to past answers. The general case is deferred to Appendix E.

Let us say that a message msg is *queried* if the adversary asks for the signature of msg, or if the adversary asks for the hash of an input of the form $\mathsf{msg}\|s$ for some salt s. Let m be the number of distinct messages queried by the adversary. Let $\mathsf{msg}_1,\ldots,\mathsf{msg}_m$ be the queried messages. As in the previous section, if a hash input $\mathsf{msg}_i\|s$ is queried, we assume the adversary first queries the signature of msg_i, and learns the associated special salt (the one chosen for the signature), and its image. In the remainder, we assume the adversary knows the signature, special salt, and hash image for all msg_i's.

As a consequence, all queries related to msg_i are queries to the hash oracle, on inputs of the form $\mathsf{msg}_i\|s$ for various salts s. Let q_i be the number of such queries for msg_i. Since we don't count signature queries (they are "free" for the $m \le q$ queried messages), we have $q=\sum_{i=1}^{m} q_i$.

If we look at the adversary's view related to a given message msg_i, everything happens as in the single-message case from the previous section. Let $\mathbf{v}_i$ denote the vector $\mathbf{v}$ associated with msg_i. Let

[2] The notation $\mathsf{Hyp}[N-1,\mathsf{Bin}[N,p]-1,q]$ is defined in the natural way: it describes the distribution obtained by first sampling $X \sim \mathsf{Bin}[N,p]$, then sampling from $\mathsf{Hyp}[N-1,X-1,q]$. Alternatively, viewing $x \mapsto \mathsf{Hyp}[N,x,q]$ as a probabilistic map, it is the image of the distribution $\mathsf{Bin}[N,p]-1$ by that map, hence the notation.

$f_i = |\mathcal{R}(\mathbf{v}_i)|/|\mathcal{R}|$ be the probability that a salt for msg_i is suitable. Then the distribution of the number of observed suitable salts for msg_i is $\mathsf{Bin}[q_i, f_i]$ in the ideal world, and $\mathsf{Hyp}[N-1, \mathsf{Bin}[N, f_i]-1, q]$ in the real world. It is worth noting that if $f_i = 1$, then all salts are suitable, and there is not difference between the real and ideal worlds for msg_i. It follows that we can focus our attention on i's such that $f_i < 1$.

The number of suitable salts for each msg_i is independent, since the vectors $\mathbf{v}_i$ are sampled independently. Hence if we look at the vector of length m recording the number of suitable salts for each msg_i, the distribution of that vector is:

$$\begin{aligned}\text{Ideal world:} \quad & \bigotimes_{i=1}^{m} \mathsf{Bin}[q_i, f_i] \\ \text{Real world:} \quad & \bigotimes_{i=1}^{m} \mathsf{Hyp}[N-1, \mathsf{Bin}[N, f_i]-1, q_i].\end{aligned}$$

As in the case of a single message, the adversary's view conditioned on a given realization of that vector is identical in both worlds. Equivalently, there exists a probabilistic function F that takes as input the vector of the number of suitable messages for each msg_i, and outputs the adversary's view conditioned on that vector. That function is identical in both worlds. To construct F, for each msg_i, and given the number of suitable salts k_i for msg_i, choose at random k_i distinct queries among the q_i queries of the adversary related to msg_i, and sample their image uniformly at random in $\mathcal{R}(\mathbf{v}_i)$; for the other $q_i - k_i$ queries, sample their image uniformly at random in $\mathcal{R} \setminus \mathcal{R}(\mathbf{v}_i)$. Regarding the special salt, F picks the special salt uniformly at random in $\mathcal{R}(\mathbf{v})$, except in the special case $k_i = -1$, where it is picked outside $\mathcal{R}(\mathbf{v})$ (and in that case, all other salts are sampled in the same way). The adversary's view in G_3 is $F(\bigotimes_{i=1}^{m} \mathsf{Bin}[q_i, f_i])$, and it is $F(\bigotimes_{i=1}^{m} \mathsf{Hyp}[N-1, \mathsf{Bin}[N, f_i]-1, q_i])$ in G_4.

We conclude that the statistical distance between G_3 and G_4 is bounded by:

$$\Delta\left(\bigotimes_{i=1}^{m} \mathsf{Bin}[q_i, f_i], \bigotimes_{i=1}^{m} \mathsf{Hyp}[N-1, \mathsf{Bin}[N, f_i]-1, q_i].\right) \tag{1}$$

The next stage of the proof is to analyze this statistical distance, using the ideas sketched in the technical overview.

4.3 Core Mathematical Results

Our goal is to upper-bound the expression (1). The bound is given in Theorem 2. Below, we use the notation $N' = N - 1$, which turns out to be a more relevant quantity in the full proof. Let:

$$\begin{aligned}\mathcal{D}_{\mathsf{ideal}} &= \bigotimes_{i=1}^{m} \mathsf{Hyp}[N', \mathsf{Bin}[N', f_i], q_i] = \bigotimes_{i=1}^{m} \mathsf{Bin}[q_i, f_i], \\ \mathcal{D}_{\mathsf{real}} &= \bigotimes_{i=1}^{m} \mathsf{Hyp}[N', \mathsf{Bin}[N'+1, f_i]-1, q_i].\end{aligned}$$

Theorem 2 (core result).

$$\Delta(\mathcal{D}_{\mathsf{ideal}}, \mathcal{D}_{\mathsf{real}}) = \mathcal{O}\left(\frac{\log(f^{-1})}{\sqrt{f}} \cdot \frac{\sqrt{q}}{N}\right) + qe^{-\Omega(fN)}.$$

The full proof is given in Appendix C in the selective setting, and in Appendix E in the adaptive setting. A high-level overview was given in the technical overview. In addition, we provide below a more precise description of the layout of the proof, in the selective setting (as explained in the technical overview, we are ultimately able to reuse the same core ideas in the adaptive setting).

We want to bound the statistical distance between $\mathcal{D}_{\mathsf{ideal}}$ and $\mathcal{D}_{\mathsf{real}}$. For now, let us pretend that $m = 1$, and look at a single component:

$$\begin{aligned}\mathcal{C}_{\mathsf{ideal}} &= \mathsf{Bin}[q_i, f_i],\\ \mathcal{C}_{\mathsf{real}} &= \mathsf{Hyp}[N', \mathsf{Bin}[N'+1, f_i] - 1, q_i].\end{aligned}$$

If we could somehow "replace" $\mathsf{Bin}[N'+1, f_i]-1$ by $\mathsf{Bin}[N', f_i - \varepsilon_i]$ in the expression of $\mathcal{C}_{\mathsf{real}}$, for some well-chosen ε_i, we would be quite happy. Indeed, we would then only have to bound the statistical distance between:

$$\begin{aligned}\mathcal{C}_{\mathsf{ideal}} &= \mathsf{Bin}[q_i, f_i],\\ \mathcal{C}'_{\mathsf{real}} &= \mathsf{Hyp}[N', \mathsf{Bin}[N', f_i - \varepsilon_i], q_i] = \mathsf{Bin}[q_i, f_i - \varepsilon_i].\end{aligned}$$

The hypergeometric distribution has disappeared, and we are left with the task of bounding the statistical distance between two binomial distributions with the same number of trials. This can be done using standard techniques, typically via the Kullback-Leibler divergence.

The crux of our argument is that such a replacement is possible. First, we introduce a notion of covering (Definition 15), then we prove a *replacement lemma* (Lemma 13) that states the following. If P, Q, R are three distributions such that Q covers P, and P stochastically dominates R, then:

$$\Delta(P, Q) \leq \Delta(R, Q).$$

Such a lemma provides what we want: if P, S, R are in the correct configuration, and we are trying to upper-bound $\Delta(P, Q)$, then we are allowed to replace P by R, since that can only increase the statistical distance.

This "replacement lemma" was designed with our goal in mind: the distributions $\mathcal{C}_{\mathsf{ideal}}, \mathcal{C}_{\mathsf{real}}, \mathcal{C}'_{\mathsf{real}}$ are in the correct configuration to apply the lemma. That is, we show:

- $\mathcal{C}_{\mathsf{ideal}}$ covers $\mathcal{C}_{\mathsf{real}}$ (Lemma 14);
- $\mathcal{C}_{\mathsf{real}}$ stochastically dominates $\mathcal{C}'_{\mathsf{real}}$ (Lemma 17).

Technically, we do not exactly show the second point: instead, we show that $\mathcal{C}_{\mathsf{real}}$ and $\mathcal{C}'_{\mathsf{real}}$ are exponentially close in statistical distance to distributions P and R such that P stochastically dominates R (the exact statement is given in Lemma 17). That is enough for the argument to go through.

That was when looking at a single component ($m = 1$). In reality, we need to bound $\Delta(\mathcal{D}_{\mathsf{ideal}}, \mathcal{D}_{\mathsf{real}})$ for an arbitrary number of components $m \geq 1$. We use the exact same proof structure for the general case. In fact, because the components are independent, much of the analysis reduces to the case of a single component (cf. Appendix C.3).

In the end, thanks to the replacement technique sketched above, we are left with bounding the statistical distance between binomials (Appendix C.4). The most standard way to do so would be to use the Kullback-Leibler divergence. In our setting, the Hellinger distance yields a slightly better bound, so we use that instead.

5 Application to MQ-based Signature Schemes

5.1 Overview of Provable Security for UOV-based Signature Schemes

The $\mathsf{T}_{\mathsf{UOV}}$ PSF. For the remainder of the paper, we fix a finite field $\mathbb{F}$. Let n and m be two positive integers, and let $v = n - m$. We say that a quadratic multivariate polynomial is a (n, m)-OV-polynomial

if it is of the following form:

$$q : \mathbb{F}^n \longrightarrow \mathbb{F}$$
$$(x_1, \ldots, x_n) \longmapsto \sum_{i=1}^{v} \sum_{j=i}^{n} \alpha_{i,j} x_i x_j.$$

The UOV scheme relies on a tuple $\mathcal{Q} = (q_1, \ldots, q_m)$ of (n, m)-OV-polynomials that are chosen uniformly at random (i.e. the $\alpha_{i,j}$s are sampled uniformly at random from $\mathbb{F}$). The first v coordinates of a (n, m)-OV-polynomial are called *vinegar* variables, while the remaining m coordinates are the *oil* variables. Note that, once a value is assigned to all vinegar variables, $\mathcal{Q}$ becomes linear in the oil variables. For any $\mathbf{v} = (x_1, \ldots, x_v) \in \mathbb{F}^v$, we denote $\mathcal{Q}(\mathbf{v}, \cdot)$ this linear map. In order to hide the decomposition of the input space into oil and vinegar variables, a secret invertible linear map $\mathcal{T} : \mathbb{F}^n \longleftarrow \mathbb{F}^n$ is chosen uniformly at random in $\mathsf{GL}(\mathbb{F}, n)$. The UOV trapdoor $\mathrm{I}_{\mathcal{Q},\mathcal{T}}$ then operates as follows:

- I^1 simply samples a vector $\mathbf{v} \in \mathbb{F}^v$ uniformly at random;
- $\mathrm{I}^2_{\mathcal{Q},\mathcal{T}}$ simply tries to find $\mathbf{o} \in \mathbb{F}^m$ such that $\mathcal{Q}(\mathbf{v}, \mathbf{o}) = \mathbf{y}$, given $\mathbf{v} \in \mathbb{F}^v$ and $\mathbf{y} \in \mathbb{F}^m$; if the equation admits one or more solutions, it samples one uniformly at random and then outputs $\mathcal{T}^{-1}(\mathbf{v}, \mathbf{o})$, else it outputs $\perp$.

The public one-way function F is defined as $\mathrm{F} = \mathcal{Q} \circ \mathcal{T}$, and SampDom samples a uniformly random vector $\mathbf{x}$ in $\mathbb{F}^n$. As a slight generalization, the $\mathsf{T}_{\mathsf{UOV}^-_\delta}$ PSF is defined analogously, but the secret map $\mathcal{Q}$ is a m-tuple of $(n, m + \delta)$-OV-polynomials. It is easy to see that $\mathsf{T}_{\mathsf{UOV}} = \mathsf{T}_{\mathsf{UOV}^-_0}$.

The Original UOV scheme. The original UOV scheme, as introduced in [KPG99], is specified in Algorithm 7. Its key generation algorithm corresponds to the sampling of a $\mathsf{T}_{\mathsf{UOV}}$ instance $(\mathrm{F}, \mathrm{I}_{\mathcal{Q},\mathcal{T}})$. Generating a signature boils down to:

- hashing the message msg (using an optional random salt salt);
- sampling vinegar values and using $\mathrm{I}^2_{\mathcal{Q},\mathcal{T}}$ to compute a value $\mathbf{s}$ until there is a solution.

The signature $(\mathsf{salt}, \mathbf{s})$ will then satisfy $\mathrm{F}(\mathbf{s}) = \mathcal{H}(\mathsf{msg}||\mathsf{salt})$, which corresponds to the equality that is checked by the verification algorithm. More modern descriptions [BCH+23] will sample vinegar values until the associated linear system is full-rank. It is also possible to turn UOV into a deterministic signature scheme by deriving all the random coins from the message and an additional secret key, using for example an eXtendable Output Function (XOF). The security of UOV is thus built on the hardness of inverting a UOV public key. Unfortunately, formally proving this security reduction remains an open problem to this day. As pointed out in [SSH11], the output of the signature algorithm is not uniformly distributed, as the vinegar variables are not uniformly random in $\mathbb{F}^v$. Instead, this distribution is skewed based on the secret key. This makes the usual proof strategy of reducing the EUF-CMA-security of the signature scheme to its security without a signature oracle hard to apply, as it will be difficult to sample from the distribution of signatures using only the knowledge of the public key. To solve this issue, Sakumoto *et al.* introduced different way of computing signatures, which is dubbed modified UOV (or mUOV for short) [SSH11].

The modified UOV scheme. In [SSH11], two important changes were made to the original UOV scheme:

1. the use of a random salt becomes mandatory, and its length is directly related to the security of the scheme[3];
2. the vinegar value $\mathbf{v}$ is now sampled once, uniformly at random.

Algorithm 7 The Original UOV scheme.

procedure UOV.KeyGeneration(1^λ)
  $(\mathrm{F}, \mathrm{I}_{\mathcal{Q},\mathcal{T}}) \leftarrow \mathsf{T}_{\mathsf{UOV}}.\mathrm{Gen}(1^\lambda)$
  $\mathsf{sk} \leftarrow \mathrm{I}_{\mathcal{Q},\mathcal{T}}$
  $\mathsf{pk} \leftarrow \mathrm{F}$
  return $(\mathsf{sk}, \mathsf{pk})$

procedure UOV.Verify$^{\mathcal{H}}$($\mathsf{pk}, \mathsf{msg}, \mathsf{sig}$)
  $(\mathsf{salt}, \mathbf{s}) \leftarrow \mathsf{sig}$
  return $\mathrm{F}(\mathbf{s}) = \mathcal{H}(\mathsf{msg}||\mathsf{salt})$

procedure UOV.Sign($\mathsf{sk}, \mathsf{msg}$)
  $\mathsf{salt} \leftarrow_{\$} \mathcal{R}$
  repeat
    $\mathbf{v} \leftarrow \mathrm{I}^1()$
    $\mathbf{s} \leftarrow \mathrm{I}^2_{\mathcal{Q},\mathcal{T}}(\mathbf{v}, \mathcal{H}(\mathsf{msg}||\mathsf{salt}))$
  until $\mathbf{s} \neq \perp$
  return $(\mathsf{salt}, \mathbf{s})$

Algorithm 8 The Modified UOV scheme.

procedure mUOV.KeyGeneration(1^λ)
  $(\mathrm{F}, \mathrm{I}_{\mathcal{Q},\mathcal{T}}) \leftarrow \mathsf{T}_{\mathsf{UOV}}.\mathrm{Gen}(1^\lambda)$
  $\mathsf{sk} \leftarrow \mathrm{I}_{\mathcal{Q},\mathcal{T}}$
  $\mathsf{pk} \leftarrow \mathrm{F}$
  return $(\mathsf{sk}, \mathsf{pk})$

procedure mUOV.Verify$^{\mathcal{H}}$($\mathsf{pk}, \mathsf{msg}, \mathsf{sig}$)
  $(\mathsf{salt}, s) \leftarrow \mathsf{sig}$
  return $\mathrm{F}(s) = \mathcal{H}(\mathsf{msg}||\mathsf{salt})$

procedure mUOV.Sign($\mathsf{sk}, \mathsf{msg}$)
  $\mathbf{v} \leftarrow \mathrm{I}^1()$
  repeat
    $\mathsf{salt} \leftarrow_{\$} \mathcal{R}$
    $\mathbf{s} \leftarrow \mathrm{I}^2_{\mathcal{Q},\mathcal{T}}(\mathbf{v}, \mathcal{H}(\mathsf{msg}||\mathsf{salt}))$
  until $\mathbf{s} \neq \perp$
  return $(\mathsf{salt}, \mathbf{s})$

New salt values are generated until the hash value belongs to the image of $\mathcal{Q}(\mathbf{v}, \cdot)$. A formal description of mUOV can be found in Algorithm 8. These modifications have the benefit of turning the signature generation into a preimage sampling experiment for $\mathsf{T}_{\mathsf{UOV}}$. The authors then prove the following result.

Lemma 1 ([SSH11]). *If $\mathcal{A}$ is an adversary against the* PS*-security of $\mathsf{T}_{\mathsf{UOV}}$, then*

$$\mathsf{Adv}^{\mathsf{PS}}_{\mathsf{T}_{\mathsf{UOV}}}(\mathcal{A}) = 0.$$

Using Lemma 1, they deduce the following security reduction in the Random Oracle Model (ROM).

Theorem 3 ([SSH11]). *Let $\mathcal{A}$ be an adversary against the* EUF-CMA*-security of* mUOV *that runs in time t, makes at most q_{sign} signature queries and $q_{\mathcal{H}}$ random oracle queries. There exists an adversary $\mathcal{B}$ against the* INV*-security of $\mathsf{T}_{\mathsf{UOV}}$ that satisfies*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{mUOV}}(\mathcal{A}) \leq \frac{q_{\mathcal{H}} + q_{\mathsf{sign}} + 1}{1 - (q_{\mathsf{sign}} + q_{\mathcal{H}})q_{\mathsf{sign}}/|\mathcal{R}|} \mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}_{\mathsf{UOV}}}(\mathcal{B}).$$

*Besides, $\mathcal{B}$ runs in time $t' = t + (q_{\mathcal{H}} + q_{\mathsf{sign}} + 1)(t_{\mathsf{UOV}} + O(1))$ where t_{UOV} is an upper-bound on the running time to evaluate the $\mathsf{T}_{\mathsf{UOV}}$.*F *function.*

A critical step in the proof of Theorem 3 is to build a simulator $\mathcal{S}$ for simulating the signing process without the knowledge of the mUOV secret. In a nutshell, in order to generate a signature for a message msg, $\mathcal{S}$ proceeds as follows:

– it samples a uniformly random signature $(\mathsf{salt}, \mathbf{s})$ in $\mathcal{R} \times \mathbb{F}^n$;

[3] This can still be done deterministically from the output of a XOF.

– it programs the random oracle $\mathcal{H}$ so that $\mathcal{H}(\mathsf{msg}||\mathsf{salt}) = \mathrm{F}(\mathbf{s})$.

Sakumoto *et al.* argue that, given Lemma 1, $\mathcal{S}$ is indistinguishable from a legitimate signer as long as $\mathcal{H}(\mathsf{msg}||\mathsf{salt})$ has never been queried before by the adversary. However, as pointed out by Chatterjee *et al.*, the proof of this theorem contains two potential gaps [CDP22]:

– since F is neither bijective nor known to be regular, for a fixed F, the value $\mathrm{F}(\mathbf{s})$ has no reason to be uniformly random when $\mathbf{s} \leftarrow_{\$} \mathbb{F}^n$. Because the random oracle is programmed to take this value, its distribution is no longer uniform.
– in Algorithm 8, salts are actually chosen uniformly at random among salts that yield a value that belongs to the image of $\mathcal{Q}(\mathbf{v}, \cdot)$, which creates a difference with the distribution of salts that are generated by $\mathcal{S}$.

Taking these issues into account, they prove the following result.

Theorem 4 ([CDP22]). *Let $\mathcal{A}$ be an adversary against the* EUF-CMA*-security of* mUOV *that runs in time t, makes at most q_{sign} signature queries and $q_{\mathcal{H}}$ random oracle queries. There exists an adversary $\mathcal{B}$ against the* INV*-security of* $\mathsf{T}_{\mathsf{UOV}}$ *that satisfies*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{mUOV}}(\mathcal{A}) \leq q_{\mathcal{H}}\mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}_{\mathsf{UOV}}}(\mathcal{B}) + 2\frac{q_{\mathsf{sign}}}{|\mathbb{F}|}.$$

Besides, $\mathcal{B}$ runs in time $t' = t + (q_{\mathcal{H}} + q_{\mathsf{sign}} + 1)(t_{\mathsf{UOV}} + O(1))$ where t_{UOV} is an upper-bound on the running time to evaluate the $\mathsf{T}_{\mathsf{UOV}}$.F *function.*

In a nutshell, the authors prove that, as long as the choice of $\mathbf{v}$ yields a full-rank linear map, salts and outputs of F are both uniformly random. Thus, the statistical distance between both distributions can be upper-bounded by $2\frac{q_{\mathsf{sign}}}{|\mathbb{F}|}$. Unfortunately, this results in a scheme that can only be proven secure when the size of the underlying field is superpolynomial in the security parameter. Recently, Kosuge and Xagawa proved the security of the mUOV scheme in the QROM [KX22].

Theorem 5 ([KX22], Proposition B.4). *For any quantum* EUF-CMA *adversary $\mathcal{A}$ of* mUOV *issuing at most q_s classical queries to the signing oracle and $q_{\mathcal{H}}$ quantum random oracle queries to $\mathcal{H}$, there exist an* INV *adversary $\mathcal{B}$ issuing q_s sampling queries such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{mUOV}}(\mathcal{A}) \leq & (2q_{\mathcal{H}}+1)^2\mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}}(\mathcal{B}) + \frac{3}{2}q'_{\mathsf{sign}}\sqrt{\frac{q'_{\mathsf{sign}}+q_{\mathcal{H}}+1}{|\mathcal{R}|}} \\ & + 2(q_{\mathsf{sign}}+q_{\mathcal{H}}+2)\sqrt{\frac{q'_{\mathsf{sign}}-q_{\mathsf{sign}}}{|\mathcal{R}|}},\end{aligned}$$

where q'_{sign} is a bound on the total number of queries to $\mathcal{H}$ in all the signing queries, and the running time of $\mathcal{B}$is about that of $\mathcal{A}$.

Using a similar proof in the classical setting, it would be possible to prove that, for any *classical* adversary $\mathcal{A}$ against the EUF-CMA-security of mUOV, one has[4]

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{mUOV}}(\mathcal{A}) \leq q_{\mathcal{H}}\mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}}(\mathcal{B}) + q'_{\mathsf{sign}}\frac{q_{\mathcal{H}}}{|\mathcal{R}|} + 2q_{\mathcal{H}}\frac{q'_{\mathsf{sign}}-q_{\mathsf{sign}}}{|\mathcal{R}|}.$$

[4] A recent update of [KX22] makes a similar observation.

Besides, for the *deterministic* variant of mUOV, this bound could be improved to

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{mUOV}}(\mathcal{A}) \leq q_{\mathcal{H}}\mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}}(\mathcal{B}) + \frac{q''_{\mathsf{sign}}q_{\mathcal{H}}}{|\mathcal{R}|} + 2\frac{q_{\mathcal{H}}(q'_{\mathsf{sign}} - q_{\mathsf{sign}})}{|\mathcal{R}|}, \tag{2}$$

where q''_{sign} denotes an upper-bound on the total number of random oracle queries during a single call to the signing oracle. We do not provide a sketch of this proof, as in the following section an analogous bound will be proven with a slightly different strategy. We note that the bound from Eq. (2) depends on the value of q'_{sign} and q''_{sign}. In [SSH11], the authors argue that the expected value of q''_{sign} is equal to 2, and thus the one of q'_{sign} is equal to $2q_{\mathsf{sign}}$.

On variants of UOV. Eq. (2) yields better security bounds for some variants of mUOV. Indeed, if τ denotes the probability that $\mathcal{Q}(\mathbf{v}, \cdot)$ is not full-rank when $\mathbf{v}$ is chosen uniformly at random, the expected value of $q'_{\mathsf{sign}} - q_{\mathsf{sign}}$ becomes τq_{sign}. As an example, the UOV^- and MAYO trapdoor parameters can be chosen so that $\tau q_{\mathsf{sign}} \approx 1$, which ensures security as long as $q_{\mathcal{H}} \ll |\mathcal{R}|$.

5.2 Application of Theorem 1 to mUOV and mUOV^-

Let us fix three integers n, m, and δ such that $m+\delta \leq n$. Let mUOV^-_δ denote the variant of mUOV where the underlying quadratic system $\mathcal{Q} = (q_1, \ldots, q_m)$ consists of $(n, m+\delta)$-OV-polynomials. Standard mUOV corresponds to $\delta = 0$. The NIST candidate PROV corresponds to $\delta = 8$.

In order to apply Theorem 1 to mUOV^-_δ, the critical step is to find two real numbers f and ε such that $\mathsf{T}_{\mathsf{UOV}^-_\delta}$ is (f, ε)-well-behaved. Note that, during the computation of I_2, the value of p is exactly $\frac{1}{|\mathbb{F}|^{m-r}}$ where r is the rank of the linear map $\mathcal{Q}(\mathbf{v}, \cdot)$. Let us denote $\mathbf{M}_{\mathbf{v}} \in \mathbb{F}^{m\times(m+\delta)}$ the matrix of the linear map $\mathcal{Q}(\mathbf{v}, \cdot)$ for any $\mathbf{v} \in \mathbb{F}^{n-m-\delta}$, and q_i the i-th component of $\mathcal{Q}$ for $i = 1, \ldots, m$. By definition, it is clear that the j-th coefficient of the i-th row of $\mathbf{M}_{\mathbf{v}}$ is exactly the coefficient of $x_{n-m-\delta+j}$ of $q_i(\mathbf{v}, \cdot)$. More formally, this coefficient is equal to

$$\sum_{k=1}^{n-m-\delta} \alpha^i_{k,j} v_k, \text{ where } q_i(\mathbf{x}) = \sum_{j=1}^{n-m-\delta}\sum_{k=j}^{n} \alpha^i_{j,k} x_j x_k.$$

It is clear that, if $\mathbf{v}$ is null, then the rank of $\mathbf{M}_{\mathbf{v}}$ is 0. Otherwise, $\mathbf{M}_{\mathbf{v}}$ is a uniformly random matrix since the $\alpha^i_{j,k}$ are uniformly random and independent. In order to accurately study the rank of $\mathbf{M}_{\mathbf{v}}$, we rely on the following result.

Lemma 2 ([Lan95,Lev05]). *Let n, N and r be three integers such that $1 \leq r \leq N \leq n$. One has*

$$\begin{aligned} p(|\mathbb{F}|, N, n, r) &:= \Pr[\mathsf{rank}(A) = r] \\ &= |\mathbb{F}|^{(r-N)(n-r)} \frac{\prod_{j=N-r+1}^{N}(1-|\mathbb{F}|^{-j}) \prod_{j=n-r+1}^{n}(1-|\mathbb{F}|^{-j})}{\prod_{j=1}^{r}(1-|\mathbb{F}|^{-j})}, \end{aligned}$$

where the probability is taken over the uniformly random choice of A in $\mathbb{F}^{N\times n}$.

Combining our previous remark and Lemma 2, we get

$$\Pr[\mathsf{rank}(\mathbf{M}_{\mathbf{v}}) = r] = \begin{cases} \frac{1}{|\mathbb{F}|^{n-m-\delta}} + \left(1 - \frac{1}{|\mathbb{F}|^{n-m-\delta}}\right)\frac{1}{|\mathbb{F}|^{m(m+\delta)}} & \text{if } r = 0, \\ \left(1 - \frac{1}{|\mathbb{F}|^{n-m-\delta}}\right) p(|\mathbb{F}|, m, m+\delta, r) & \text{if } r = 1, \ldots, m. \end{cases}$$

For any λ, let $r_0(\lambda)$ be the biggest r such that $2^\lambda \Pr[\mathsf{rank}(\mathbf{M}_{\mathbf{v}}) < r] \leq 1$. In that case, $\mathsf{T}_{\mathsf{UOV}^-_\delta}$ is $\left(\frac{1}{|\mathbb{F}|^{m-r_0(\lambda)}}, \Pr[\mathsf{rank}(\mathbf{M}_{\mathbf{v}}) < r_0(\lambda)]\right)$-well-behaved. Applying Theorem 1, we get the following result.

Corollary 1. *Let $\mathcal{A}$ be an adversary against the* EUF-CMA*-security of* mUOV^-_δ *that issues at most $q_{\mathcal{H}}$ random oracle queries, q_{sign} signature queries, and runs in time at most t. Then, there exists an adversary $\mathcal{B}$ against the* INV*-security of* $\mathsf{T}_{\mathsf{UOV}^-_\delta}$ *such that*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{mUOV}}(\mathcal{A}) \leq \mathcal{O}\left((m-r_0(\lambda))\log(|\mathbb{F}|)|\mathbb{F}|^{(m-r_0(\lambda))/2}\frac{\sqrt{q}}{|\mathcal{R}|}\right) + q\Pr[\mathsf{rank}(\mathbf{M_v}) < r_0(\lambda)] + q\mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}_{\mathsf{UOV}}}(\mathcal{B}) + qe^{-\Omega\left(\frac{|\mathcal{R}|}{|\mathbb{F}|^{m-r_0(\lambda)}}\right)},$$

where $q = q_{\mathcal{H}} + q_{\mathsf{sign}}$. Besides, $\mathcal{B}$ runs in time $t' = t + (q_{\mathcal{H}} + q_{\mathsf{sign}} + 1)(t_{\mathsf{UOV}} + O(1))$ where t_{UOV} is an upper-bound on the running time to evaluate the $\mathsf{T}_{\mathsf{UOV}}$.F*.*

We now evaluate the security of various instances of mUOV^- using the bound from Corollary 1. In Table 1, we apply the parameter sets from the UOV submission to the NIST PQC competition [BCD+23] to the mUOV signature scheme. Then, in Table 2, we apply Corollary 1 to the PROV submission [PCF+23], which is actually an instance of mUOV^-_8. We remark that our bounds allows us to optimize the $|\mathcal{R}|$ parameter. This is done in Table 3.

Variant	λ	$\lvert\mathbb{F}\rvert$	n	m	|salt|	|sig|	|pk|	|sk|	f	$\log(f^{-1})/\sqrt{f}$	ε	τ
mUOV-Ip	128	256	112	44	16	128	43576	48	2^{-24}	3×2^{15}	$2^{-127.99}$	1/15
mUOV-Is	128	16	160	64	16	96	66576	48	2^{-20}	5×2^{12}	2^{-143}	1/255
mUOV-III	192	256	184	72	16	200	189232	48	2^{-32}	2^{21}	2^{-199}	1/255
mUOV-V	256	256	244	96	16	260	446992	48	2^{-40}	5×2^{23}	2^{-287}	1/255

Table 1. Parameter sets and corresponding key and signature sizes for the mUOV signature scheme, in bytes, based on the parameter sets of the UOV submission. τ is an upper-bound on the probability that $\mathbf{M_v}$ is not full-rank.

Variant	λ	$\lvert\mathbb{F}\rvert$	n	m	δ	|salt|	|sig|	|pk|	|sk|	f	$\log(f^{-1})/\sqrt{f}$	ε	τ
PROV-I	128	256	136	46	8	24	160	68326	16	2^{-8}	2^{7}	2^{-159}	2^{-71}
PROV-III	192	256	200	70	8	32	232	215694	24	2^{-16}	2^{12}	2^{-263}	2^{-71}
PROV-V	256	256	264	96	8	40	304	524192	32	2^{-16}	2^{12}	2^{-263}	2^{-71}

Table 2. Parameter sets and corresponding key and signature sizes for the PROV signature scheme, in bytes. τ is an upper-bound on the probability that $\mathbf{M_v}$ is not full-rank.

Scheme	mUOV				PROV		
Variant	Ip	Is	III	V	I	III	V
Alternative |salt|	11	10	15	20	9	14	18
New |sig|	123	90	199	264	145	214	282

Table 3. Alternative salt sizes (in Bytes) for mUOV and PROV based on the application of Theorem 1. Updated corresponding signature sizes (in Bytes) are provided for reference.

Remark 3. When $r = \mathsf{rank}(\mathbf{M_v}) > 0$, the expected number of salt sampling is $|\mathbb{F}|^{m-r}$. This means that the signature algorithm may leak the value of r with a simple timing attack, depending on the probability τ that $\mathbf{M_v}$ is not full-rank. Tables 1 and 2 illustrate the fact that an adequate choice for δ can protect against such attacks, as it is unlikely that PROV will ever need to sample an additional salt as long as $q_{\mathsf{sign}} \ll 2^{71}$.

5.3 Application of Theorem 1 to MAYO

Alternative description of $\mathsf{T}_{\mathsf{UOV}}$. In this section we present an alternative way to describe the UOV trapdoor due to Beullens [Beu21]. Let F be a UOV public map and let $(\mathcal{Q}, \mathcal{T})$ be its corresponding secret key. Let us also denote $V = \mathcal{T}^{-1}(\mathbb{F}^{n-m} \times \{0\}^m)$ and

$$O = \mathcal{T}^{-1}(\{0\}^{n-m} \times \mathbb{F}^m).$$

It is clear that $\mathbb{F}^n = V + O$. Moreover, for any $\mathbf{o} \in O$, one has $\mathrm{F}(\mathbf{o}) = 0$. Hence, an alternative way of describing the secret key associated to F is the description of V and O. Besides, as stated in [Beu21, Theorem 1], to any UOV public map, we can associate a bilinear form F' defined as

$$\begin{aligned} \mathbb{F}^n \times \mathbb{F}^n &\longrightarrow \mathbb{F}^m \\ (\mathbf{x}, \mathbf{y}) &\longmapsto \mathrm{F}(\mathbf{x}+\mathbf{y}) - \mathrm{F}(\mathbf{x}) - \mathrm{F}(\mathbf{y}). \end{aligned}$$

Let us fix any $\mathbf{t} \in \mathbb{F}^m$. This gives us an alternative way of solving the equation $\mathrm{F}(\mathbf{s}) = \mathbf{t}$ using the knowledge of V and O. Indeed, let us fix any $\mathbf{v} \in V$. Then, for any $\mathbf{o} \in O$, one has

$$\mathrm{F}(\mathbf{v}+\mathbf{o}) = \mathrm{F}(\mathbf{v}) + \mathrm{F}(\mathbf{o}) + \mathrm{F}'(\mathbf{v}, \mathbf{o}),$$

with $\mathrm{F}(\mathbf{o}) = 0$. Hence, solving $\mathrm{F}(\mathbf{s}) = \mathbf{t}$ is equivalent to find $\mathbf{o} \in O$ such that

$$\mathrm{F}'(\mathbf{v}, \mathbf{o}) = \mathbf{t} - \mathrm{F}(\mathbf{v}),$$

which is linear in $\mathbf{o}$.

Finally, in order to shorten the secret key and to simplify the computations, one can generate O such that $\mathbb{F}^n = \mathbb{F}^{n-m} \times \{0\}^m + O$.

The $\mathsf{T}_{\mathsf{MAYO}}$ PSF. Let us $\mathsf{T}_{\mathsf{MAYO}}$ be a trapdoor function used in MAYO [Beu22b]. The function $\mathsf{T}_{\mathsf{MAYO}}$.Gen first generates a subspace $O \subset \mathbb{F}^n$ of dimension o, along with a quadratic map $\mathrm{F} : \mathbb{F}^n \longrightarrow \mathbb{F}^m$ such that $\mathrm{F}(O) = \{0\}$, as described at the start of this section. Moreover, for $1 \leq i, j \leq k$, let $\mathbf{E}_{i,j}$ be a matrix such that

$$\mathbf{E} = \begin{pmatrix} \mathbf{E}_{1,1} & \cdots & \mathbf{E}_{1,k} \\ \vdots & \ddots & \vdots \\ \mathbf{E}_{k,1} & \cdots & \mathbf{E}_{k,k} \end{pmatrix}$$

is non-singular. Then, it generates the quadratic map F^* as follows:

$$\begin{aligned} (\mathbb{F}^n)^k &\longrightarrow \mathbb{F}^m \\ (\mathbf{x}_1, \ldots, \mathbf{x}_k) &\longmapsto \sum_{i=1}^{k} \mathbf{E}_{i,i}\mathrm{F}(\mathbf{x}_i) + \sum_{1 \leq i < j \leq k} \mathbf{E}_{i,j}\mathrm{F}'(\mathbf{x}_i, \mathbf{x}_j). \end{aligned}$$

The function $\mathsf{T}_{\mathsf{MAYO}}$.F will be the quadratic map F^*, while $\mathsf{T}_{\mathsf{MAYO}}$.I will work as follows:

– $\mathsf{T}_{\mathsf{MAYO}}.\mathrm{I}_1$ simply samples $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_k) \in (\mathbb{F}^{n-o} \times \{0\}^o)^k$ uniformly at random;
– $\mathsf{T}_{\mathsf{MAYO}}.\mathrm{I}_2$ tries to solve in $\mathbf{o} = (\mathbf{o}_1, \ldots, \mathbf{o}_k) \in O^k$ the linear system of equations

$$\mathrm{F}^*(\mathbf{v}_1 + \mathbf{o}_1, \ldots, \mathbf{v}_k + \mathbf{o}_k) = \mathbf{y}, \tag{3}$$

where $\mathbf{y}$ is the target that was provided as input, and outputs $\mathbf{v} + \mathbf{o}$.

Finally, $\mathsf{T}_{\mathsf{MAYO}}.\textsc{SampDom}$ simply samples a value uniformly at random in $(\mathbb{F}^n)^k$. Note that, for Eq.(3) to admit solutions with a high probability, it is required that $m \leq ko$.

The MAYO Signature Scheme. In [Beu22b], Beullens introduces the MAYO signature scheme as a variant of the original UOV, as specified in Algorithm 9. Note that, in that case, the signature algorithm will continue to sample vinegar values until the corresponding linear system of equations is full-rank. Just like in the original UOV scheme, it is possible to prove the security of MAYO since signatures are uniformly random when the system is full-rank. In particular, the following theorem is proven.

Theorem 6 ([Beu22b], Theorem 6). *Let $\mathcal{A}$ be an adversary against the* EUF-CMA*-security of* MAYO *that runs in time t, makes q_{sign} signing queries and $q_{\mathcal{H}}$ random oracle queries. Let $\tau = \frac{|\mathbb{F}|^{k-n+o}}{q-1} + \frac{|\mathbb{F}|^{m-ko}}{|\mathbb{F}|-1}$ be a bound on the restarting probability and suppose $q_{\mathsf{sign}}\tau < 1$, then there exists an adversary $\mathcal{B}$ against the* INV*-security of $\mathsf{T}_{\mathsf{MAYO}}$ that runs in time $t + \tilde{\mathcal{O}}(q_{\mathsf{sign}} + q_{\mathcal{H}})$ such that*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{T}_{\mathsf{MAYO}}}(\mathcal{A}) \leq \left(\mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}_{\mathsf{MAYO}}}(\mathcal{B}) + \frac{1}{|\mathbb{F}|^m}\right)(1 - q_{\mathsf{sign}}\tau)^{-1} + \frac{(q_{\mathcal{H}} + q_{\mathsf{sign}})q_{\mathsf{sign}}}{|\mathcal{R}|}.$$

Algorithm 9 The Original MAYO scheme.

procedure $\mathsf{MAYO}.\textsc{KeyGeneration}(1^\lambda)$
  $(\mathrm{F}, \mathrm{I}_O) \leftarrow \mathsf{T}_{\mathsf{MAYO}}.\textsc{Gen}(1^\lambda)$
  $\mathsf{sk} \leftarrow \mathrm{I}_O$
  $\mathsf{pk} \leftarrow \mathrm{F}$
  return $(\mathsf{sk}, \mathsf{pk})$

procedure $\mathsf{MAYO}.\textsc{Verify}^{\mathcal{H}}(\mathsf{pk}, \mathsf{msg}, \mathsf{sig})$
  $(\mathsf{salt}, s) \leftarrow \mathsf{sig}$
  return $\mathrm{F}(s) = \mathcal{H}(\mathsf{msg}||\mathsf{salt})$

procedure $\mathsf{MAYO}.\textsc{Sign}(\mathsf{sk}, \mathsf{msg})$
  $\mathsf{salt} \leftarrow_\$ \mathcal{R}$
  repeat
    $\mathbf{v} \leftarrow \mathrm{I}^1()$
  until $\mathrm{F}^*(\mathbf{v} + \mathbf{o})$ is full-rank
  $\mathbf{s} \leftarrow \mathrm{I}^2_O(\mathbf{v}, \mathcal{H}(\mathsf{msg}||\mathsf{salt}))$
  return $(\mathsf{salt}, \mathbf{s})$

Remark 4. In [Beu22b], Beullens actually breaks down the INV-security of $\mathsf{T}_{\mathsf{MAYO}}$ in two problems:

– distinguishing the public map F from a uniformly random quadratic map F_0;
– solving the so-called Whipped MQ problem, which consists in solving the equation $\mathrm{F}^*_0(\mathbf{x}) = \mathbf{y}$ for a uniformly random $\mathbf{y}$ in $\mathbb{F}^m$.

The full description of these problems is not needed for our discussion, so that we do not formally introduce them.

Remark 5. MAYO has also been submitted to the NIST PQC competition for digital signature schemes [BCC+23]. As in the case of UOV, we will now introduce a modified MAYO algorithm, dubbed mMAYO, and we will study its security based on the submitted parameter sets for the original scheme.

The modified MAYO Signature Scheme. As in Section 5.2, we introduce the mMAYO signature scheme that is an instantiation of the probabilistic hash and sign with retry construction, using $\mathsf{T}_{\mathsf{MAYO}}$ as the trapdoor. A formal description is given in Algorithm 10. In order to apply Theorem 1, we need to study the PS-security of $\mathsf{T}_{\mathsf{MAYO}}$ and to determine appropriate values for f and ε. First, we note that one has the following result.

Lemma 3 ([Beu22b]). *Let $\mathcal{A}$ be an adversary against the PS-security of $\mathsf{T}_{\mathsf{MAYO}}$. One has*

$$\mathsf{Adv}^{\mathsf{PS}}_{\mathsf{T}_{\mathsf{MAYO}}}(\mathcal{A}) = 0.$$

While [Beu22b] does not rely on the PSF formalism, this result can be seen as a byproduct of the proof of [Beu22b, Lemma 7]. For the sake of completeness, we briefly sketch the proof of Lemma 3.

Proof. Recall that $\mathbf{v}$ is chosen uniformly at random in $\left(\mathbb{F}^{n-o} \times \{0\}^o\right)^k$ by I_1. Once $\mathbf{v}$ is fixed, one gets a linear map of rank $r \leq m$. The Sample_0 game will then sample uniformly at random a target $\mathbf{y}$ such that the system admits solutions. Then, I_2 samples such a solution $\mathbf{o} \in O^k$ uniformly at random. Since there are exactly $|\mathbb{F}|^{ko-r}$ possible solutions for each of the $|\mathbb{F}|^r$ suitable $\mathbf{y}$, $\mathbf{o}$ will be uniformly random in O^k. Due to the fact that

$$\left(\mathbb{F}^n\right)^k = \left(\mathbb{F}^{n-o} \times \{0\}^o\right)^k + O^k,$$

the output of I_2 will indeed be uniformly random in $\left(\mathbb{F}^n\right)^k$. □ □

Second, as a consequence of the proof of [Beu22b, Lemma 2], one has the following two cases:

- either $\mathbf{v}_1, \ldots, \mathbf{v}_k$ are linearly dependent,
- or the linear part of the system $\mathrm{F}^*(\mathbf{v} + \mathbf{o})$ is equivalent to $\mathbf{M}_{\mathbf{v}}\mathbf{o}$, where $\mathbf{M}_{\mathbf{v}}$ is a uniformly random matrix in $|\mathbb{F}|^{m \times ko}$.

Using [Beu22b, Lemma 9], the first case occurs with a probability that is bounded by $\frac{|\mathbb{F}|^{k-n+o}}{|\mathbb{F}|-1}$, which is negligible in all MAYO parameter sets. We can thus use the following approximation:

$$\Pr[\mathsf{rank}(\mathbf{M}_{\mathbf{v}}) = r] \simeq p(|\mathbb{F}|, ko, m, r)$$

if $r = 1, \ldots, m$, and $\Pr[\mathsf{rank}(\mathbf{M}_{\mathbf{v}}) = 0]$ is negligible. Using a similar strategy as in Section 5.2, let $r_0(\lambda)$ be the maximum r such that $2^\lambda \Pr[\mathsf{rank}(\mathbf{M}_{\mathbf{v}}) < r] \lesssim 1$. Then $\mathsf{T}_{\mathsf{MAYO}}$ is (f, ε)-well-behaved, with

$$(f, \varepsilon) = \left(\frac{1}{|\mathbb{F}|^{m-r_0(\lambda)}}, \Pr[\mathsf{rank}(\mathbf{M}_{\mathbf{v}}) < r_0(\lambda)]\right).$$

Applying Theorem 1, we get the following result.

Corollary 2. *Let $\mathcal{A}$ be an adversary against the EUF-CMA-security of mMAYO that issues at most $q_{\mathcal{H}}$ random oracle queries, q_{sign} signature queries, and runs in time at most t. Then, there exists an adversary $\mathcal{B}$ against the INV-security of $\mathsf{T}_{\mathsf{MAYO}}$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{mMAYO}}(\mathcal{A}) \leq \mathcal{O}\left((m - r_0(\lambda)) \log(|\mathbb{F}|) |\mathbb{F}|^{(m-r_0(\lambda))/2} \frac{\sqrt{q}}{|\mathcal{R}|}\right) \\ + q \Pr[\mathsf{rank}(\mathbf{M}_{\mathbf{v}}) < r_0(\lambda)] + q\mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}_{\mathsf{MAYO}}}(\mathcal{B}) + q e^{-\Omega\left(\frac{|\mathcal{R}|}{|\mathbb{F}|^{m-r_0(\lambda)}}\right)},\end{aligned}$$

where $q = q_{\mathcal{H}} + q_{\mathsf{sign}}$. Besides, $\mathcal{B}$ runs in time $t' = t + (q_{\mathcal{H}} + q_{\mathsf{sign}} + 1)(t_{\mathsf{MAYO}} + O(1))$ where t_{MAYO} is an upper-bound on the running time to evaluate the $\mathsf{T}_{\mathsf{MAYO}}$.F function.

Algorithm 10 The Modified UOV scheme.

procedure mMAYO.KeyGeneration(1^λ)
  $(\mathrm{F}, \mathrm{I}_O) \leftarrow \mathsf{T}_{\mathsf{MAYO}}.\mathrm{Gen}(1^\lambda)$
  $\mathsf{sk} \leftarrow \mathrm{I}_O$
  $\mathsf{pk} \leftarrow \mathrm{F}$
  return $(\mathsf{sk}, \mathsf{pk})$

procedure mMAYO.Verify$^{\mathcal{H}}$($\mathsf{pk}, \mathsf{msg}, \mathsf{sig}$)
  $(\mathsf{salt}, s) \leftarrow \mathsf{sig}$
  return $\mathrm{F}(s) = \mathcal{H}(\mathsf{msg}||\mathsf{salt})$

procedure mMAYO.Sign($\mathsf{sk}, \mathsf{msg}$)
  $\mathbf{v} \leftarrow \mathrm{I}^1()$
  repeat
    $\mathsf{salt} \leftarrow_\$ \mathcal{R}$
    $\mathbf{s} \leftarrow \mathrm{I}_O^2(\mathbf{v}, \mathcal{H}(\mathsf{msg}||\mathsf{salt}))$
  until $\mathbf{s} \neq \perp$
  return $(\mathsf{salt}, \mathbf{s})$

We now evaluate the security of various instances of mMAYO using the bound from Corollary 2. In Table 4, we apply the parameter sets from the MAYO submission to the NIST PQC competition [BCC+23] to the mMAYO signature scheme, while Table 5 presents optimized salt sizes based on Corollary 2.

Variant	λ	$\lvert\mathbb{F}\rvert$	n	m	o	k	|salt|	|sig|	|pk|	|sk|	f	$\log(f^{-1})/\sqrt{f}$	ε	τ
mMAYO_1	128	16	66	64	8	9	24	321	1168	24	2^{-8}	2^7	2^{-131}	2^{-36}
mMAYO_2	128	16	78	64	18	4	24	180	5488	24	2^{-8}	2^7	2^{-131}	2^{-36}
mMAYO_3	192	16	99	96	10	11	32	577	2656	32	2^{-8}	2^7	2^{-203}	2^{-60}
mMAYO_5	256	16	133	128	12	12	40	838	5008	40	2^{-12}	3×2^8	2^{-319}	2^{-68}

Table 4. Parameter sets and corresponding key and signature sizes for the mMAYO signature scheme, in bytes, based on the parameter sets of the MAYO submission. τ is an upper-bound on the probability that $\mathbf{M}_\mathbf{v}$ is not full-rank [BCC+23].

Scheme	mMAYO			
Variant	1	2	3	5
Alternative |salt|	9	9	13	18
New |sig|	306	165	558	816

Table 5. Alternative salt sizes (in Bytes) for mMAYO based on the application of Theorem 1. Updated corresponding signature sizes (in Bytes) are provided for reference.

We also note that, as long as the vinegar variables that are sampled during a signing query lead to a full-rank system, MAYO and mMAYO have exactly the same behavior. Hence, the bound from Corollary 2 also applies to MAYO with an additional $q_{\mathsf{sign}}\tau$ term. This yields the following result.

Corollary 3. *Let $\mathcal{A}$ be an adversary against the* EUF-CMA*-security of* MAYO *that issues at most $q_\mathcal{H}$ random oracle queries, q_{sign} signature queries, and runs in time at most t. Then, there exists an*

adversary $\mathcal{B}$ *against the* INV*-security of* $\mathsf{T}_{\mathsf{MAYO}}$ *such that*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{MAYO}}(\mathcal{A}) \leq \mathcal{O}\left((m-r_0(\lambda))\log(|\mathbb{F}|)|\mathbb{F}|^{(m-r_0(\lambda))/2}\frac{\sqrt{q}}{|\mathcal{R}|}\right) + q_{\mathsf{sign}}\tau + q\Pr[\mathsf{rank}(\mathbf{M}_{\mathbf{v}}) < r_0(\lambda)] + q\mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}_{\mathsf{MAYO}}}(\mathcal{B}) + qe^{-\Omega\left(\frac{|\mathcal{R}|}{|\mathbb{F}|^{m-r_0(\lambda)}}\right)},$$

where $q = q_{\mathcal{H}} + q_{\mathsf{sign}}$ *and* τ *is an upper-bound on the probability that* $\mathbf{M}_{\mathbf{v}}$ *is not full-rank. Besides,* $\mathcal{B}$ *runs in time* $t' = t + (q_{\mathcal{H}} + q_{\mathsf{sign}} + 1)(t_{\mathsf{MAYO}} + O(1))$ *where* t_{MAYO} *is an upper-bound on the running time to evaluate the* $\mathsf{T}_{\mathsf{MAYO}}.\mathrm{F}$ *function.*

Remark 6. We note that neither Theorem 6 nor Corollary 3 give meaningful bounds for the MAYO_1, MAYO_2 and MAYO_3 parameter sets. However, both results do apply for MAYO_5, and our bound would allow the salt length to be safely reduced from 40 bytes to 18 bytes.

6 Conclusion

In this work, we prove a new generic result on the security of the hash-and-sign with retry construction when it is based on a (f, ε)-well-behaved preimage-sampleable function. In particular, this proof relies on a new combinatorial tool that allows us to get adaptive security from a non-adaptive argument. Moreover, our bound is tighter than previous bounds, since its security degradation is of the form $\frac{\log(f^{-1})\sqrt{q}}{\sqrt{f}|\mathcal{R}|} + q\varepsilon$, where $\mathcal{R}$ is the salt space of the signature scheme. To show the applicability of our result, we use it to revisit the security proof of the PROV submission to the NIST PQC competition. Moreover, we provide variants of the UOV and MAYO submissions, using the same parameter sets, whose security can actually be proven using our result.

As a first open problem, it will be nice to look at the quantum security proof of Kosuge and Xagawa [KX22] and see if we can gain the $\sqrt{q}$ factor. Another problem will be to study if our new technique from adaptive-to-non-adaptive proof can have other applications.

References

BBC+22. John Baena, Pierre Briaud, Daniel Cabarcas, Ray A. Perlner, Daniel Smith-Tone, and Javier A. Verbel. Improving support-minors rank attacks: Applications to GeMSS and rainbow. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 376–405. Springer, Heidelberg, August 2022.

BCC+23. Ward Beullens, Fabio Campos, Sofía Celi, Basil Hess, and Matthias J. Kannwischer. MAYO. Version 1.0 – 05/30/2023. *Round 1 Additional Signatures, Post-Quantum Cryptography: Digital Signature Schemes*, 2023.

BCD+23. Ward Beullens, Ming-Shing Chen, Jintai Ding, Boru Gong, Matthias J. Kannwischer, Jacques Patarin, Bo-Yuan Peng, Dieter Schmidt, Cheng-Jhih Shih, Chengdong Tao, and Bo-Yin Yang. UOV: Unbalanced Oil and Vinegar. Algorithm Specifications and Supporting Documentation Version 1.0 – 05/30/2023. *Round 1 Additional Signatures, Post-Quantum Cryptography: Digital Signature Schemes*, 2023.

BCH+23. Ward Beullens, Ming-Shing Chen, Shih-Hao Hung, Matthias J. Kannwischer, Bo-Yuan Peng, Cheng-Jhih Shih, and Bo-Yin Yang. Oil and vinegar: Modern parameters and implementations. *IACR TCHES*, 2023(3):321–365, 2023.

Beu21. Ward Beullens. Improved cryptanalysis of UOV and Rainbow. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 348–373. Springer, Heidelberg, October 2021.

Beu22a. Ward Beullens. Breaking rainbow takes a weekend on a laptop. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 464–479. Springer, Heidelberg, August 2022.

Beu22b. Ward Beullens. MAYO: Practical post-quantum signatures from oil-and-vinegar maps. In Riham AlTawy and Andreas Hülsing, editors, *SAC 2021*, volume 13203 of *LNCS*, pages 355–376. Springer, Heidelberg, September / October 2022.

BLL+15. Shi Bai, Adeline Langlois, Tancrède Lepoint, Damien Stehlé, and Ron Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 3–24. Springer, Heidelberg, November / December 2015.

BR95. Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *EUROCRYPT'94*, volume 950 of *LNCS*, pages 92–111. Springer, Heidelberg, May 1995.

BR96. Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 399–416. Springer, Heidelberg, May 1996.

CDP22. Sanjit Chatterjee, M. Prem Laxman Das, and Tapas Pandit. Revisiting the security of salted UOV signature. In Takanori Isobe and Santanu Sarkar, editors, *Progress in Cryptology – INDOCRYPT 2022*, pages 697–719, Cham, 2022. Springer International Publishing.

DEP23. Léo Ducas, Thomas Espitau, and Eamonn W. Postlethwaite. Finding short integer solutions when the modulus is small. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 150–176. Springer, Heidelberg, August 2023.

ETWY22. Thomas Espitau, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Shorter hash-and-sign lattice-based signatures. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 245–275. Springer, Heidelberg, August 2022.

FHK+18. Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang, et al. Falcon: Fast-fourier lattice-based compact signatures over ntru. *Submission to the NIST's post-quantum cryptography standardization process*, 36(5):1–75, 2018.

FIKT21. Hiroki Furue, Yasuhiko Ikematsu, Yutaro Kiyomura, and Tsuyoshi Takagi. A new variant of unbalanced Oil and Vinegar using quotient ring: QR-UOV. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 187–217. Springer, Heidelberg, December 2021.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

KPG99. Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar signature schemes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 206–222. Springer, Heidelberg, May 1999.

KX22. Haruhisa Kosuge and Keita Xagawa. Probabilistic hash-and-sign with retry in the quantum random oracle model. Cryptology ePrint Archive, Report 2022/1359, 2022. https://eprint.iacr.org/2022/1359.

Lan95. G. Landsberg. Über eine Anzahlbestimmung und eine damit Zusammenhangende Reihe. *J. Reine Angew. Math.*, III:87–88, 1895.

Lee24. Keewoo Lee. Bit security as cost to demonstrate advantage. *IACR Communications in Cryptology*, 1(1), 2024.

Lev05. A.A Levitskaya. Systems of random equations over finite algebraic structures. *Cybern Syst Anal*, 2005(41):67–93, 2005.

PCF+23. Jacques Patarin, Benoît Cogliati, Jean-Charles Faugère, Pierre-Alain Fouque, Louis Goubin, Robin Larrieu, Gilles Macario-Rat, and Brice Minaud. PROV: PRovable unbalanced Oil and Vinegar. Specification v1.0 – 06/01/2023. *Round 1 Additional Signatures, Post-Quantum Cryptography: Digital Signature Schemes*, 2023.

PGMP19. Thomas Prest, Dahmun Goudarzi, Ange Martinelli, and Alain Passelègue. Unifying leakage models on a Rényi day. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 683–712. Springer, Heidelberg, August 2019.

Pre17. Thomas Prest. Sharper bounds in lattice-based cryptography using the Rényi divergence. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 347–374. Springer, Heidelberg, December 2017.

SSH11. Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. On provable security of UOV and HFE signature schemes against chosen-message attack. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 68–82. Springer, Heidelberg, November / December 2011.

Ste12. John Steinberger. Improved security bounds for key-alternating ciphers via hellinger distance. *Cryptology ePrint Archive*, 2012.

TPD21. Chengdong Tao, Albrecht Petzoldt, and Jintai Ding. Efficient key recovery for all HFE signature variants. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 70–93, Virtual Event, August 2021. Springer, Heidelberg.

Yas21. Kenji Yasunaga. Replacing probability distributions in security games via Hellinger distance. In *2nd Conference on Information-Theoretic Cryptography (ITC 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

A Additional Proofs for Section 2

A.1 Probabilistic Lemmas

Lemma 4. *Let P be a distribution on Ω, and let $E \subseteq \Omega$ such that $P(E) > 0$. Let Q be the distribution on E induced by P (i.e. $Q(\omega) = P(\omega)/P(E)$). Then:*

$$\Delta(P, Q) \leq 2P(E).$$

Definition 11 (Stochastically increasing). *Given two partial orders $(\Omega_1, \leq_1)$, $(\Omega_2, \leq_2)$, and a probabilistic map $F : \Omega_1 \to \Omega_2$, we say that F is* stochastically increasing *if any one of the following equivalent statements holds.*

1. *Given any two elements $x \leq_1 y$ in Ω_1, $F(x) \preceq_2 F(y)$.*
2. *Given any two distributions $P \preceq_1 Q$ over Ω_1, $F(P) \preceq_2 F(Q)$.*

Proof of equivalence. $2 \Rightarrow 1$. Let $\overline{x}$ (resp. $\overline{y}$) denote the distribution on Ω_1 where x (resp. y) has probability 1. Applying condition 2 to $P = \overline{x}$ and $Q = \overline{y}$ yields condition 1.

$1 \Rightarrow 2$. Consider a coupling (X, Y) witnessing $P \preceq_1 Q$. For each $x \leq_1 y$ in Ω_1, let $(X_{x,y}, Y_{x,y})$ be a coupling witnessing $F(x) \preceq_2 F(y)$. Let (X', Y') be sampled as follows. First, sample x, y from (the distribution of) (X, Y). Then sample x', y' from (the distribution of) $(X_{x,y}, Y_{x,y})$. This is well-defined because $x \leq_1 y$, by construction of (X, Y). We have $X' \sim F(X)$, $Y' \sim F(Y)$, and $x' \leq_2 y'$. Hence (X', Y') is a coupling witnessing $F(P) \preceq_2 F(Q)$. □

A.2 Hellinger distance

Definition 12 (Hellinger distance). *Given two probability distributions P and Q over a finite set Ω, the* Hellinger distance *between P and Q is:*

$$\mathsf{Hel}(P, Q) = \sqrt{\frac{1}{2} \sum_{\omega \in \Omega} \left(\sqrt{P(\omega)} - \sqrt{Q(\omega)} \right)^2}.$$

Given two distributions P and Q over Ω, if we denote by $\sqrt{P}$ the vector in $\mathbb{R}^\Omega$ defined by $\sqrt{P}(\omega) = \sqrt{P(\omega)}$, then the Hellinger distance is simply the (normalized) 2-norm of $\sqrt{P} - \sqrt{Q}$:

$$\mathsf{Hel}(P,Q) = \frac{1}{\sqrt{2}}\left\|\sqrt{P} - \sqrt{Q}\right\|_2.$$

Up to composition with a fixed function, the Hellinger distance is equal to the Rényi divergence of parameter 1/2. This relates the Hellinger distance to the Kullback-Leibler divergence, which is the Rényi divergence of parameter 1, and which shares similar features. Of interest to us are the fact that both quantities are well-behaved with respect to tensor products (Lemma 10), and can be used to bound the statistical distance (Lemma 11).

In this work, we will only make use of a few properties of the Hellinger distance, which are stated next and proved in Appendix A.

Lemma 5. *Let P, Q be two probability distributions over a finite set Ω.*

$$\mathsf{Hel}(P,Q)^2 = 1 - \sum_{\omega\in\Omega}\sqrt{P(\omega)Q(\omega)}.$$

The next lemma states that the squared Hellinger distance is sub-additive over tensor products.

Lemma 6 (Sub-additivity). *Let P_1, Q_1 (resp. P_2, Q_2) be two probability distributions over a finite set Ω_1 (resp. Ω_2).*

$$\mathsf{Hel}(P_1\otimes P_2, Q_1\otimes Q_2)^2 \le \mathsf{Hel}(P_1,Q_1)^2 + \mathsf{Hel}(P_2,Q_2)^2.$$

The next lemma relates the Hellinger distance to the statistical distance, playing a role similar to Pinsker's inequality for the Kullback-Leibler divergence.

Lemma 7. *Let P, Q be two probability distributions over a finite set Ω.*

$$\Delta(P,Q) \le \sqrt{2}\mathsf{Hel}(P,Q).$$

Lemma 8 (Data-processing inequality). *Let P, Q be two probability distributions over Ω. Let $F : \Omega \to \Omega'$ be a probabilistic map.*

$$\Delta(F(P),F(Q)) \le \Delta(P,Q),$$
$$\mathsf{Hel}(F(P),F(Q)) \le \mathsf{Hel}(P,Q).$$

Remark 7. The data-processing inequality is true for all f-divergences, of which the statistical distance and Hellinger distance are special cases.

Lemma 9. *Let P, Q be two probability distributions over a finite set Ω.*

$$\mathsf{Hel}(P,Q)^2 = 1 - \sum_{\omega\in\Omega}\sqrt{P(\omega)Q(\omega)}.$$

Proof.

$$\begin{aligned}\mathsf{Hel}(P,Q)^2 &= \frac{1}{2}\sum_{\omega\in\Omega}\left(\sqrt{P(\omega)} - \sqrt{Q(\omega)}\right)^2\\ &= \frac{1}{2}\left(\sum_{\omega\in\Omega}P(\omega) + \sum_{\omega\in\Omega}Q(\omega)\right) - \sum_{\omega\in\Omega}\sqrt{P(\omega)Q(\omega)}\\ &= 1 - \sum_{\omega\in\Omega}\sqrt{P(\omega)Q(\omega)}.\square\end{aligned}$$

Remark 8. The quantity $\sum_{\omega\in\Omega}\sqrt{P(\omega)Q(\omega)}$ is sometimes called the Hellinger affinity, or Bhattacharyya coefficient.

Remark 9. A corollary of Lemma 9 is that the Hellinger distance ranges over $[0,1]_{\mathbb{R}}$ (a consequence of the normalization factor $1/\sqrt{2}$ in Definition 12).

Lemma 10 (Sub-additivity). *Let P_1, Q_1 (resp. P_2, Q_2) be two probability distributions over a finite set Ω_1 (resp. Ω_2).*

$$\mathsf{Hel}(P_1\otimes P_2, Q_1\otimes Q_2)^2 \leq \mathsf{Hel}(P_1,Q_1)^2 + \mathsf{Hel}(P_2,Q_2)^2.$$

Proof. By Lemma 9:

$$\begin{aligned}
\mathsf{Hel}(P_1\otimes P_2, Q_1\otimes Q_2)^2 &= 1 - \sum_{(\omega_1,\omega_2)\in\Omega_1\times\Omega_2}\sqrt{P_1(\omega_1)Q_1(\omega_1)P_2(\omega_2)Q_2(\omega_2)}\\
&= 1 - \left(\sum_{\omega_1\in\Omega_1}\sqrt{P(\omega_1)Q(\omega_1)}\right)\left(\sum_{\omega_2\in\Omega_2}\sqrt{P(\omega_2)Q(\omega_2)}\right)\\
&= 1 - \Big(1-\mathsf{Hel}(P_1,Q_1)^2\Big)\Big(1-\mathsf{Hel}(P_2,Q_2)^2\Big)\\
&= \mathsf{Hel}(P_1,Q_1)^2 + \mathsf{Hel}(P_2,Q_2)^2 - \mathsf{Hel}(P_1,Q_1)^2\mathsf{Hel}(P_2,Q_2)^2\\
&\leq \mathsf{Hel}(P_1,Q_1)^2 + \mathsf{Hel}(P_2,Q_2)^2.\square
\end{aligned}$$

Lemma 11. *Let P, Q be two probability distributions over a finite set Ω.*

$$\Delta(P,Q) \leq \sqrt{2}\mathsf{Hel}(P,Q).$$

Proof. Using the Cauchy-Schwartz inequality:

$$\begin{aligned}
\Delta(P,Q) &= \frac{1}{2}\sum_{\omega\in\Omega}|P(\omega)-Q(\omega)|\\
&= \frac{1}{2}\sum_{\omega\in\Omega}\left|\sqrt{P(\omega)}-\sqrt{Q(\omega)}\right|\left(\sqrt{P(\omega)}+\sqrt{Q(\omega)}\right)\\
&\leq \frac{1}{2}\sqrt{\sum_{\omega\in\Omega}\left(\sqrt{P(\omega)}-\sqrt{Q(\omega)}\right)^2}\cdot\sqrt{\sum_{\omega\in\Omega}\left(\sqrt{P(\omega)}+\sqrt{Q(\omega)}\right)^2}\\
&= \frac{1}{\sqrt{2}}\mathsf{Hel}(P,Q)\cdot\sqrt{2+2\sum_{\omega\in\Omega}\sqrt{P(\omega)Q(\omega)}}\\
&\leq \sqrt{2}\mathsf{Hel}(P,Q)
\end{aligned}$$

where the last line uses $\sum_{\omega\in\Omega}\sqrt{P(\omega)Q(\omega)}\leq 1$, which follows from Lemma 9. □

B Lower Bound on the Statistical Distance between the Real and the Ideal Distributions in UOV

Let $F:\mathbb{F}_q^n\to\mathbb{F}_q^m$ be the UOV mapping (where m is the number of oil variables, n is the number of oil + vinegar variables. By an abuse of notation, F^{-1} is the the function that associates to $y\in\mathbb{F}_q^m$ its

preimage, as computed by a honest UOV signature algorithm. Let F^{-1} is a probabilistic map: it first samples the vinegar variables uniformly at random, then computes valid oil variables if they exist. For the purpose of this note, we do not care exactly how F^{-1} chooses among possible preimages if there are multiple solutions. The only thing that matters is that as long as a solution exists, F^{-1} always outputs a solution. It never samples new vinegar variables, unless there is no solution for the current choice of vinegar variables. In other words, there is no "rejection sampling".

Real distribution. The *real* distribution is defined to be the distribution seen by the adversary when observing honest signatures. It is:

$$\mathcal{D}_R = \{(F^{-1}(y), y) : y \leftarrow_\$ \mathbb{F}_q^m\}$$

Proof distribution. The *proof* distribution is the distribution used in the proof we discussed last time, where the challenger first chooses $x \in \mathbb{F}_q^n$ uniformly at random, programs the random oracle to output $F(x)$ on a certain message, then outputs x as signature for that message. It is:

$$\mathcal{D}_P = \{(x, F(x)) : x \leftarrow_\$ \mathbb{F}_q^n\}$$

Proposition 2. *The statistical distance between $\mathcal{D}_R$ and $\mathcal{D}_P$ is non-negligible. In fact, it is lower-bounded by a non-zero constant.*

Both distributions $\mathcal{D}_R$ and $\mathcal{D}_P$ are defined over pairs (x, y) where $y = F(x)$. In particular, y is entirely determined by x. As a consequence, we can restrict our attention to the distribution of the first element of the pair for both distributions. Consequently, in the remainder, by a small abuse of notation, we view $\mathcal{D}_R$ and $\mathcal{D}_P$ as distributions over $\mathbb{F}_q^n$.

Definition 13. *Let $O \subset \mathbb{F}_q^n$ denote the oil subspace. Two elements $x, x' \in \mathbb{F}_q^n$ are said to be* twins*, written $x\ T\ x'$, if $F(x) = F(x')$ and $x' - x \in O$.*

The twin relation T is an equivalence relation. From now on, the *twins* of x means the elements of the equivalence class of x for T. The point of the definition is that the twins of $x \in \mathbb{F}_q^n$ are exactly the elements $x' \in \mathbb{F}_q^n$ that have the same image and the same vinegar variables as x. The argument we will make is that in the real UOV distribution $\mathcal{D}_R$, the probability that an element x is output by F^{-1} is constrained by its number of twins.

For $i \in \mathbb{N}^*$, let A_i denote the elements of $\mathbb{F}_q^n$ that have exactly i twins (including themselves).

Lemma 12. *The probability that a uniformly random y is such that $F^{-1}(y) \in A_2$ is lower-bounded by some constant[5] $C > 0$ (except with negligible probability over the choice of F).*

For each pair of twins in A_2, pick whichever twin has a lower probability for $\mathcal{D}_R$ (if both twins have the same probability, pick one arbitrarily). Let $A_2' \subset A_2$ be the set of those "lower probability" twins. Note that $|A_2'| = |A_2|/2$. In the remainder, the notation $\mathcal{D}_*(x)$ means "the probability of sampling x according to distribution $\mathcal{D}_*$". The final computation below only uses of the following three facts.

1. For $x \in A_2'$, $\mathcal{D}_P(x) = q^{-n}$ since $\mathcal{D}_P$ is uniform.
2. For $x \in A_2'$, $\mathcal{D}_R(x) \leq (1 - \varepsilon)q^{-n}$ for some constant $\varepsilon > 0$ (except with negligible probability over the choices of F and x). This is because the probability of choosing x for $\mathcal{D}_R$ is exactly the probability of drawing $y = F(x)$ when sampling uniformly over $\mathbb{F}_q^m$, and that F^{-1} chooses the vinegar variables to be x mod O, and that F^{-1} chooses x among the possible twins. The first event has probability q^{-m}; the second event has probability $C'q^{-(n-m)}$ for some $1 < C' < 2$ (except

[5] The probability is something like $1/(2e)$, by standard concentration bounds it won't deviate much from that value.

with negligible probability over the choices of F and x), because F^{-1} samples the vinegar variables uniformly among those that have a solution, which is a fraction $1 - 1/e$ or so; and the last event has probability less than one half by definition of A'_2. (We don't compute all this precisely, but it's worth noting that we can "beat out" any constant corrective factor by using A_i for some constant $i > 2$ instead of A_2, which means exact constants don't really matter.)

3. By Lemma 12, the cardinality of A_2 is $\Omega(q^n)$. Indeed, as already observed, there are $\Omega(q^m)$ valid choices for pairs $(F(x), x \bmod O)$, and each pair maps to a distinct T-equivalence class. By Lemma 12, the probability that a pair maps to A_2 is $\Omega(1)$. Hence there are $\Omega(q^m)$ distinct T-equivalence classes in A_2. A fortiori the cardinality of A_2 is $\Omega(q^n)$. The same holds for A'_2 since it is half the size of A_2.

At this point, we've found a set of constant probability (namely A'_2) where $\mathcal{D}_P$ and $\mathcal{D}_R$ disagree by a constant factor > 1. This is enough to imply that the statistical distance between the two distributions is $\Omega(1)$, as shown in the following computation.

$$\begin{aligned}\Delta(\mathcal{D}_P, \mathcal{D}_R) &= \frac{1}{2}\sum_{x \in \mathbb{F}_q^n} |\mathcal{D}_P(x) - \mathcal{D}_R(x)| \geq \frac{1}{2}\sum_{x \in A'_2} |\mathcal{D}_P(x) - \mathcal{D}_R(x)| \\ &\geq |A'_2| q^{-n} \varepsilon / 2 = \Omega(1).\end{aligned}$$

C Proof of Theorem 2

We use the same notation as in Section 4.

C.1 Covering Relationship

Definition 14 (upwards closure). *Let $(\Omega, \leq_\Omega)$ be a partial order, and let $S \subseteq \Omega$. The* upwards closure *of S (with respect to $\leq_\Omega$), written $\mathsf{up}_{\leq_\Omega}(S)$, is defined by:*

$$\mathsf{up}_{\leq_\Omega}(S) = \{y \in \Omega : \exists x \in S, x \leq_\Omega y\}.$$

If the relevant partial order $\leq_\Omega$ is clear from context, we may simply write $\mathsf{up}(S)$. The set S is said to be closed upwards *(with respect to $\leq_\Omega$) if $\mathsf{up}_{\leq_\Omega}(S) = S$.*

The next definition is not standard (and new to our knowledge), but will play a central role in the analysis. Recall that $\mathsf{up}_{\leq_\Omega}(S)$ denotes the upwards closure of S (Definition 14).

Definition 15 (covering relationship). *Given a partial order $(\Omega, \leq_\Omega)$, and two distributions P and Q over Ω, we say that Q* covers *P with respect to $\leq_\Omega$ if:*

$$\mathsf{up}_{\leq_\Omega}(\{\omega \in \Omega : P(\omega) < Q(\omega)\}) \subseteq \{\omega \in \Omega : P(\omega) \leq Q(\omega)\}.$$

The next lemma is the reason we introduce the notion of cover. It says that if Q covers P, then when upper-bounding $\Delta(P, Q)$, we are free to replace P with any distribution R that is stochastically dominated by P.

Lemma 13 (Replacement lemma). *Given a partial order $(\Omega, \leq_\Omega)$, let P, Q, R, R' be distributions over Ω. If Q covers P (with respect to $\leq_\Omega$), and P stochastically dominates R (again with respect to $\leq_\Omega$), then:*

$$\Delta(P, Q) \leq \Delta(R, Q).$$

More generally, if Q covers P, and P stochastically dominates R', then:

$$\Delta(P, Q) \leq \Delta(R, Q) + \Delta(R, R').$$

Proof. Let us prove the second, more general statement. Let:

$$S = \mathsf{up}_{\leq_\Omega}(\{\omega \in \Omega : P(\omega) <_\Omega Q(\omega)\}).$$

Since Q covers P, we have:

$$\{\omega \in \Omega : D_1(\omega) <_\Omega D_2(\omega)\} \subseteq S \subseteq \{\omega \in \Omega : D_1(\omega) \leq_\Omega D_2(\omega)\}.$$

Recall that the statistical distance between P and Q can be defined as:

$$\begin{aligned}\Delta(P,Q) &= \max_{E \subseteq \Omega}(Q(E) - P(E)) \\ &= \max_{E \subseteq \Omega} \sum_{e \in E}(Q(e) - P(e)).\end{aligned}$$

The maximum is reached for $E = S$, since S contains all e's such that $Q(e) - P(e) > 0$, and none of the e's such that $D_2(e) - D_1(e) < 0$. This implies $\Delta(P,Q) = Q(S) - P(S)$.

On the other hand, since S is closed upwards and P stochastically dominates R', $R'(S) \leq P(S)$. This implies $R(S) \leq P(S) + R(S) - R'(S) \leq P(S) + \max_{E \subseteq \Omega}(R(E) - R'(E)) = P(S) + \Delta(R, R')$.

Putting everything together, we get:

$$\begin{aligned}\Delta(P,Q) &= Q(S) - P(S) \\ &\leq Q(S) - R(S) + \Delta(R,R') \\ &\leq \max_{E \subseteq \Omega}(Q(E) - R(E)) + \Delta(R,R') \\ &= \Delta(R,Q) + \Delta(R,R').\Box\end{aligned}$$

C.2 The Case of a Single Component

In this section, we first consider the case $m = 1$, so that $q = q_1$. Let $p = f_1$. Let:

$$\begin{aligned}\mathcal{C}_{\mathsf{ideal}} &= \mathsf{Bin}[q,p] \\ \mathcal{C}_{\mathsf{real}} &= \mathsf{Hyp}[N', \mathsf{Bin}[N'+1,p] - 1, q].\end{aligned}$$

That is, $\mathcal{C}_{\mathsf{ideal}}$ and $\mathcal{C}_{\mathsf{real}}$ are the distributions $\mathcal{D}_{\mathsf{ideal}}$ and $\mathcal{D}_{\mathsf{real}}$ when $m = 1$.

Lemma 14. *For all $q \leq n$ in $\mathbb{N}^*$, and for all $p \in [0,1]_\mathbb{R}$, $\mathsf{Bin}[q,p]$ covers $\mathsf{Hyp}[n, \mathsf{Bin}[n+1,p] - 1, q]$.*

Proof. Let $X \sim \mathsf{Bin}[n,p]$ and let h be the probabilistic map defined by $h(x) \sim \mathsf{Hyp}[n,x,q]$, for $x \in [-1,n]$. (Recall the convention $\mathsf{Hyp}[\cdot,-1,\cdot] = -1$, which means that h is well-defined on $[-1,n]_\mathbb{N}$.) We have $h(X-1) \sim \mathsf{Hyp}[n, \mathsf{Bin}[n,p] - 1, q]$. Let us compute for $k \in [0,n]$:

$$\begin{aligned}\mathsf{Hyp}[n, \mathsf{Bin}[n,p] - 1, q](k) &= \Pr[h(X-1) = k] \\ &= \sum_{i=k+1}^{n-q+k+1} \Pr[X = i] \Pr[h(X-1) = k \mid X = i] \\ &= \sum_{i=k+1}^{n-q+k+1} \mathsf{Bin}[n,p](i)\mathsf{Hyp}[n,i-1,q](k),\end{aligned}$$

where the sum is taken over $i \in [k+1, n-q+k+1]$ because $\mathsf{Hyp}[n, i-1, q](k) = 0$ outside of that interval. Continuing:

$$\begin{aligned}
\mathsf{Hyp}[n, \mathsf{Bin}[n,p]-1, q](k) &= \sum_{i=k+1}^{n-q+k+1} p^i (1-p)^{n-i} \binom{n}{i}\binom{q}{k}\binom{n-q}{i-1-k}\binom{n}{i-1}^{-1} \\
&= \sum_{i=k+1}^{n-q+k+1} p^i (1-p)^{n-i} \binom{q}{k}\binom{n-q}{i-1-k}\frac{n-i+1}{i} \\
&= p^{k+1}(1-p)^{q-k-1}\binom{q}{k}\sum_{i=0}^{n-q} p^i (1-p)^{n-q-i}\binom{n-q}{i}\frac{n-i-k}{i+k+1} \\
&= \mathsf{Bin}[q,p](k)\frac{p}{1-p}\sum_{i=0}^{n-q}\mathsf{Bin}[n-q,p](i)\frac{n-i-k}{i+k+1}.
\end{aligned}$$

As a consequence:

$$\frac{\mathsf{Hyp}[n, \mathsf{Bin}[n,p]-1, q](k)}{\mathsf{Bin}[q,p](k)} = \frac{p}{1-p}\sum_{i=0}^{n-q}\mathsf{Bin}[n-q,p](i)\frac{n-i-k}{i+k+1}.$$

In the sum on the right-hand side, every summand is decreasing as a function of k, hence the sum is decreasing.

Observe:

$$\begin{aligned}
\mathsf{Hyp}[n, \mathsf{Bin}[n+1,p]-1, q](k) &= \mathsf{Hyp}[n, \mathsf{Bin}[n,p]+\mathsf{Ber}[p]-1, q](k) \\
&= \mathsf{Hyp}[n, \mathsf{Bin}[n,p]-\mathsf{Ber}[1-p], q](k) \\
&= p\mathsf{Hyp}[n, \mathsf{Bin}[n,p], q](k) + (1-p)\mathsf{Hyp}[n, \mathsf{Bin}[n,p]-1, q](k) \\
&= p\mathsf{Bin}[q,p](k) + (1-p)\mathsf{Hyp}[n, \mathsf{Bin}[n,p]-1, q](k).
\end{aligned}$$

Hence the quotient:

$$\frac{\mathsf{Hyp}[n, \mathsf{Bin}[n+1,p]-1, q](k)}{\mathsf{Bin}[q,p](k)} = p + (1-p)\frac{\mathsf{Hyp}[n, \mathsf{Bin}[n,p]-1, q](k)}{\mathsf{Bin}[q,p](k)}$$

is also decreasing as a function of k.

It follows that if for some $k \geq 0$, we have:

$$\mathsf{Bin}[q,p](k) \geq \mathsf{Hyp}[n, \mathsf{Bin}[n+1,p]-1, q](k),$$

then the inequality remains true for larger k's. The previous statement only considers $k \geq 0$, disregarding the special case $k = -1$. However, note that:

$$\mathsf{Hyp}[n, \mathsf{Bin}[n+1,p]-1, q](-1) > 0 = \mathsf{Bin}[q,p](-1),$$

so the statement is still true for all $k \geq -1$. Hence $\mathsf{Bin}[q,p]$ covers $\mathsf{Hyp}[n, \mathsf{Bin}[n+1,p]-1, q]$. □

Recall that $\preceq$ denotes the stochastic dominance order arising from $\leq$ (over the integers).

Lemma 15. *There exist two distributions P and Q over $\mathbb{N}$, such that $\Delta\big(P, \mathsf{Bin}\big[N', p - 2\log(p^{-1})/N'\big]\big) = e^{-\Omega(pN')}$, $\Delta(Q, \mathsf{Bin}[N'+1, p]-1) = e^{-\Omega(pN')}$, and $P \preceq Q$.*

Proof. Let $\eta = 2\log(p^{-1})/(pN')$. Let $X_1, \dots, X'_N$ be independent identically distributed variables $X_k \sim \mathsf{Ber}[p]$. Let $Y_1, \dots, Y'_N$ be independent identically distributed variables $Y_k \sim \mathsf{Ber}[1-\eta]$. Let $Z_1, \dots, Z'_N$ be defined by $Z_k = \min(X_k, Y_k)$. Observe that $Z_k \sim \mathsf{Ber}\left[p - 2\log(p^{-1})/N'\right]$.

Let $S = \{i \in [1, N']_{\mathbb{N}} : X_i = 1\}$. Let E denote the event $|S| \geq pN'/2$. Let S' denote the first $pN'/2$ elements of S (for $\leq$, or for any arbitrary fixed order), if E holds, and $S' = \varnothing$ otherwise. Let F denote the event: $\exists i \in S', Y_i = 0$. Let $\mathbb{1}_F \in \{0, 1\}$ be the random variable equal to 1 if the event F occurs, 0 otherwise.

Conditioned on E, the probability of F is exactly the probability that at least one variable Y_i is equal to 0 among $pN'/2$ fixed variables, which is to say:

$$\begin{aligned}\Pr[F|E] &= 1 - (1-\eta)^{pN'/2} \\ &= 1 - e^{\log(1-2\log(p^{-1})/(pN'))pN'/2} \\ &\geq 1 - p,\end{aligned}$$

where the last inequality is because $\log(1 - 2\log(p^{-1})/(pN'))pN'/2$ is increasing as a function of N', so it is upper-bounded by its limit $\log(p)$. Let $W \sim \mathsf{Ber}[1-p]$ be a new independent random variable.

Define P to be the distribution of $\sum_{i=1}^{N'} Z_i$, conditioned on the event E. Define Q to be the distribution of $\left(\sum_{i=1}^{N'} X_i\right) - W$, conditioned on the event E. Define Q' to be the distribution of $\left(\sum_{i=1}^{N'} X_i\right) - \mathbb{1}_F$, conditioned again on the event E. We claim that this choice of P and Q satisfies the statement of the lemma.

By a standard Chernoff bound, the probability of E is $e^{-\Omega(pN')}$. We start by showing that $\Delta\left(P, \mathsf{Bin}\left[N', p + 2\log p^{-1}/N'\right]\right) = 2e^{-\Omega(pN')} = e^{-\Omega(pN')}$. This follows from Lemma 4, because $\sum_{i=1}^{N'} Z_i \sim \mathsf{Bin}\left[N', p + 2\log p^{-1}/N'\right]$, and P is the same distribution conditioned on E. Regarding Q, observe that $\left(\sum_{i=1}^{N'} X_i\right) - W \sim \mathsf{Bin}[N', p] - \mathsf{Ber}[1-p] = \mathsf{Bin}[N', p] + \mathsf{Ber}[p] - 1 = \mathsf{Bin}[N'+1, p] - 1$. Hence we have $\left(\sum_{i=1}^{N'} X_i\right) - W \sim \mathsf{Bin}[N'+1, p] - 1$, and Q is the same distribution conditioned on E. It follows that $\Delta(Q, \mathsf{Bin}[N'+1, p] - 1) = e^{-\Omega(pN')}$.

We now show $P \preceq Q$. Since $\Pr[F|\ E] \geq 1 - p$, Q trivially stochastically dominates Q'. Moreover:

$$\Pr\left[\left(\sum_{i=1}^{N'} X_i\right) - \mathbb{1}_F \geq \sum_{i=1}^{N'} Z_i\right] = 1$$

since $X_i \geq Z_i$ for all i, and $\mathbb{1}_F = 1$ only if $X_i > Z_i$ for some $i \in S$. This means that Q' stochastically dominates P. By transitivity, Q stochastically dominates P. □

Lemma 16. *The map $h : x \mapsto \mathsf{Hyp}[N', x, q]$ is stochastically increasing.*

Proof. Choose $x \leq y$. We need to show $\mathsf{Hyp}[N', x, q] \preceq \mathsf{Hyp}[N', y, q]$. Let Z be sampled uniformly at random among subsets of $[1, N']_{\mathbb{N}}$ of cardinality q. Let $X = Z \cap [1, x]_{\mathbb{N}}$. Let $Y = Z \cap [1, y]_{\mathbb{N}}$. Observe that $|X| \sim \mathsf{Hyp}[N', x, q]$, and $|Y| \sim \mathsf{Hyp}[N', y, q]$. Moreover, $\Pr[|X| \leq |Y|] = 1$ since $X \subseteq Y$. Hence by Definition 2.1, the coupling $(|X|, |Y|)$ witnesses $\mathsf{Hyp}[N', x, q] \preceq \mathsf{Hyp}[N', y, q]$. □

Lemma 17. *There exist two distributions P and Q over $\mathbb{N}$, such that $\Delta\left(P, \mathsf{Bin}\left[q, p - 2\log(p^{-1})/N'\right]\right) = e^{-\Omega(pN')}$, $\Delta(Q, \mathsf{Hyp}[N', \mathsf{Bin}[N'+1, p] - 1, q]) = e^{-\Omega(pN')}$, and $P \preceq Q$.*

Proof. By Lemma 15, there exist P' and Q' such that $\Delta\big(P', \mathsf{Bin}\big[N', p - 2\log(p^{-1})/N'\big]\big) = e^{-\Omega(pN')}$, $\Delta(Q', \mathsf{Bin}[N'+1, p] - 1) = e^{-\Omega(pN')}$, and $P' \preceq Q'$. Let $P = \mathsf{Hyp}[N', P', q]$ and $Q = \mathsf{Hyp}[N', Q', q]$. Using the data-processing inequality for the statistical distance (Lemma 8):

$$\begin{aligned}
&\Delta\big(P, \mathsf{Bin}\big[q, p - 2\log(p^{-1})/N'\big]\big) \\
&= \Delta\big(\mathsf{Hyp}[N', P', q], \mathsf{Hyp}\big[N', \mathsf{Bin}\big[N', p - 2\log(p^{-1})/N'\big], q\big]\big) \\
&\leq \Delta\big(P', \mathsf{Bin}\big[N', p - 2\log(p^{-1})/N'\big]\big) \\
&= e^{-\Omega(pN')}.
\end{aligned}$$

Likewise for $\Delta(Q, \mathsf{Bin}[q, p]) = e^{-\Omega(pN')}$. Furthermore, because $P' \preceq Q'$ and $\mathsf{Hyp}[N', \cdot, q]$ is stochastically increasing (Lemma 16), $P \preceq Q$. □

Let $\varepsilon = 2\log(p^{-1})/N'$. Putting all lemmas we have seen in this section together, we have that $\mathcal{C}_{\mathsf{ideal}}$ covers $\mathcal{C}_{\mathsf{real}}$, and $\mathcal{C}_{\mathsf{real}}$ stochastically dominates $\mathsf{Bin}[q, p - \varepsilon]$ up to some negligible quantity $e^{\Omega(-pN')}$ ("up to" in the sense of Lemma 17). With these two facts, we can apply the replacement lemma (Lemma 13) to deduce:

$$\Delta(\mathcal{C}_{\mathsf{ideal}}, \mathcal{C}_{\mathsf{real}}) \leq \Delta(\mathsf{Bin}[q, p], \mathsf{Bin}[q, p - \varepsilon]) + e^{-pN'}.$$

That is what we want, because $\Delta(\mathsf{Bin}[q, p], \mathsf{Bin}[q, p - \varepsilon])$ can be bounded rather tightly using the Hellinger distance. In the next section, we will realize the same steps in the general case $m \geq 1$.

C.3 The General Case : Multiple Components

Recall that our goal is to bound the statistical distance between:

$$\begin{aligned}
\mathcal{D}_{\mathsf{ideal}} &= \bigotimes_{i=1}^{m} \mathsf{Bin}[q_i, f_i]; \\
\mathcal{D}_{\mathsf{real}} &= \bigotimes_{i=1}^{m} \mathsf{Hyp}[N', \mathsf{Bin}[N'+1, f_i] - 1, q_i].
\end{aligned}$$

Define:

$$\mathcal{D}'_{\mathsf{real}} = \bigotimes_{i=1}^{m} \mathsf{Bin}\big[q_i, f_i - 2\log(f_i^{-1})/N'\big].$$

Recall that $\leq_n$ denotes the usual component-wise order over $\mathbb{Z}^n$. Here, we also extend $\leq$ (resp. $\leq_n$) to be defined over $\mathbb{Z} \cup \{+\infty\}$ (resp. $(\mathbb{Z} \cup \{+\infty\})^n$), in the natural way. Recall that $\preceq_n$ denotes the stochastic dominance order arising from $\leq_n$.

Lemma 18. *$\mathcal{D}_{\mathsf{ideal}}$ covers $\mathcal{D}_{\mathsf{real}}$ with respect to $\leq_m$.*

Proof. In the proof of Lemma 14, we have shown that the quotient:

$$\frac{\mathsf{Hyp}[n, \mathsf{Bin}[n+1, p] - 1, q](k)}{\mathsf{Bin}[q, p](k)}$$

is decreasing as a function of k, for $k \in [0, n]$, and even for $k \in [-1, n]$ if we define the quotient $x/0$ to be $+\infty$ for $x > 0$. It follows that the quotient:

$$\begin{aligned}
&\frac{\mathcal{D}_{\mathsf{real}}(k_1, \ldots, k_m)}{\mathcal{D}_{\mathsf{ideal}}(k_1, \ldots, k_m)} \\
&= \frac{(\bigotimes_{i=1}^m \mathsf{Hyp}[N', \mathsf{Bin}[N'+1, f_i] - 1, q_i])(k_1, \ldots, k_m)}{(\bigotimes_{i=1}^m \mathsf{Bin}[q, f_i])(k_1, \ldots, k_m)} \\
&= \prod_{i=1}^m \frac{\mathsf{Hyp}[N', \mathsf{Bin}[N'+1, f_i] - 1, q_i](k_i)}{\mathsf{Bin}[q, f_i](k_i)}
\end{aligned}$$

is $\leq_m$-decreasing as a function of $(k_1, \ldots, k_m)$. Indeed, every factor in the product is $\leq$-decreasing as function of k_i. We conclude that $\mathcal{D}_{\mathsf{ideal}}$ covers $\mathcal{D}_{\mathsf{real}}$ with respect to $\leq_m$. □

Lemma 19. *Let $P_1, \ldots, P_n$, $Q_1, \ldots, Q_n$ be distributions over a set Ω, equipped with a partial order $\leq_\Omega$. Let $\leq_{\Omega^n}$ denote the component-wise order induced by $\leq_\Omega$: $(\omega_1, \ldots, \omega_n) \leq_{\Omega^n} (\omega'_1, \ldots, \omega'_n)$ iff for all i, $\omega_i \leq_\Omega \omega'_i$. Let $\preceq_\Omega$ (resp. $\preceq_{\Omega^n}$) be the stochastic dominance orders arising from $\leq_\Omega$ (resp. $\leq_{\Omega^n}$). If for all i, $P_i \preceq_\Omega Q_i$, then $\bigotimes_{i=1}^n P_i \preceq_{\Omega^n} \bigotimes_{i=1}^n P_i$.*

Proof. Let D_i denote the distribution of a coupling (A_i, B_i) witnessing $P_i \preceq_\Omega Q_i$. Sample the random variable $((X_1, Y_1), \ldots, (X_n, Y_n))$ from $D_1 \otimes \ldots \otimes D_n$. Let $\boldsymbol{X} = (X_1, \ldots, X_n)$ and $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$. We claim that the coupling $(\boldsymbol{X}, \boldsymbol{Y})$ witnesses $\bigotimes_{i=1}^n P_i \preceq_n \bigotimes_{i=1}^n Q_i$. Indeed, $\boldsymbol{X} \sim \bigotimes_{i=1}^n P_i$, and $\boldsymbol{Y} \sim \bigotimes_{i=1}^n Q_i$. Moreover, for all i, $X_i \leq_\Omega Y_i$ holds with probability 1, so $\boldsymbol{X} \leq_{\Omega^n} \boldsymbol{Y}$ holds with probability 1. □

Lemma 20. *There exist two distributions P and Q over $\mathbb{N}^m$, such that $\Delta(P, \mathcal{D}'_{\mathsf{real}}) = qe^{-\Omega(pN')}$, $\Delta(Q, \mathcal{D}_{\mathsf{real}}) = qe^{-\Omega(pN')}$, and $P \preceq_m Q$.*

Proof. This is a consequence of Lemmas 17 and 19. Indeed, let $\varepsilon_i = 2\log(f_i^{-1})/N'$. by Lemma 17, for each $i \leq m$, there exist distributions P_i, Q_i such that

$$\begin{aligned}
\Delta(P_i, \mathsf{Bin}[q, f_i - \varepsilon_i]) &= e^{-\Omega(f_i N')} = e^{-\Omega(f N')}, \\
\Delta(Q_i, \mathsf{Hyp}[N', \mathsf{Bin}[N'+1, f_i] - 1, q]) &= e^{-\Omega(f_i N')} = e^{-\Omega(f N')},
\end{aligned}$$

and $P_i \preceq Q_i$. Set $P = \bigotimes_{i=1}^m P_i$ and $Q = \bigotimes_{i=1}^m Q_i$. By a standard hybrid argument, we have:

$$\begin{aligned}
\Delta\left(P, \bigotimes_{i=1}^m \mathsf{Bin}[q, f_i - \varepsilon_i]\right) &= me^{-\Omega(fN')} \\
\Delta\left(Q, \bigotimes_{i=1}^m \mathsf{Hyp}[N', \mathsf{Bin}[N'+1, f_i] - 1, q]\right) &= me^{-\Omega(fN')}.
\end{aligned}$$

Moreover, by Lemma 19, $P \preceq_m Q$. □

Lemma 21.

$$\Delta(\mathcal{D}_{\mathsf{ideal}}, \mathcal{D}_{\mathsf{real}}) \leq \Delta(\mathcal{D}_{\mathsf{ideal}}, \mathcal{D}'_{\mathsf{real}}) + qe^{-\Omega(pN')}.$$

Proof. This is a direct consequence of Lemmas 13, 18 and 20. Indeed, Lemmas 18 and 20 show that the premises of Lemma 13 are satisfied by $\mathcal{D}_{\mathsf{ideal}}, \mathcal{D}_{\mathsf{real}}, \mathcal{D}'_{\mathsf{real}}$, and the conclusion of Lemma 13 yields the desired result. □

C.4 Moving to the Hellinger Distance

By Lemma 21, we have reduced our original problem to upper-bounding the distance between:

$$\mathcal{D}_{\mathsf{ideal}} = \bigotimes_{i=1}^{m} \mathsf{Bin}[q_i, f_i];$$
$$\mathcal{D}'_{\mathsf{real}} = \bigotimes_{i=1}^{m} \mathsf{Bin}\left[q_i, f_i - 2\log(f_i^{-1})/N'\right].$$

Lemma 22. *Let* $0 \leq \varepsilon \leq p \leq 1$.

$$\mathsf{Hel}(\mathsf{Ber}[p], \mathsf{Ber}[p-\varepsilon]) \leq \frac{\varepsilon}{\sqrt{p(1-p)}}.$$

Proof.

$$\begin{aligned}
2\mathsf{Hel}(\mathsf{Ber}[p], \mathsf{Ber}[p-\varepsilon])^2 &= \left(\sqrt{p} - \sqrt{p-\varepsilon}\right)^2 + \left(\sqrt{1-p+\varepsilon} - \sqrt{1-p}\right)^2 \\
&= p\left(1 - \sqrt{1-\varepsilon/p}\right)^2 + (1-p)\left(\sqrt{1+\varepsilon/(1-p)} - 1\right)^2 \\
&\leq p\frac{\varepsilon^2}{p^2} + (1-p)\frac{\varepsilon^2}{(1-p)^2} \\
&= \frac{\varepsilon^2}{p(1-p)},
\end{aligned}$$

where the first inequality uses $\sqrt{1-\varepsilon/p} \geq 1 - \varepsilon/p$ and $\sqrt{1+\varepsilon/(1-p)} \leq 1 + \varepsilon/(1-p)$. □

Lemma 23.

$$\mathsf{Hel}(\mathcal{D}_{\mathsf{ideal}}, \mathcal{D}'_{\mathsf{real}}) \leq \frac{2\log(f^{-1})}{\sqrt{f}} \cdot \frac{\sqrt{q}}{N'}.$$

Proof. This follows essentially from Lemmas 10 and 22. Indeed, let $\mathsf{Sum}(x_1, \ldots, x_k) = \sum_{i=1}^{k} x_i$ (we use the same notation for all k's). Let $\varepsilon_i = 2\log(f_i^{-1})/N'$. Using Lemma 10, the data-processing inequality

for the Hellinger distance (Lemma 8), and Lemma 22, we have:

$$\begin{aligned}
\mathsf{Hel}(\mathcal{D}_{\mathsf{ideal}}, \mathcal{D}'_{\mathsf{real}})^2 &= \mathsf{Hel}\left(\bigotimes_{i=1}^{m} \mathsf{Bin}[q_i, f_i], \bigotimes_{i=1}^{m} \mathsf{Bin}[q_i, f_i - \varepsilon_i]\right)^2 \\
&\leq \sum_{i=1}^{m} \mathsf{Hel}(\mathsf{Bin}[q_i, f_i], \mathsf{Bin}[q_i, f_i - \varepsilon_i])^2 \\
&= \sum_{i=1}^{m} \mathsf{Hel}\left(\mathsf{Sum}\left(\bigotimes_{i=1}^{q_i} \mathsf{Ber}[f_i]\right), \mathsf{Sum}\left(\bigotimes_{i=1}^{q_i} \mathsf{Ber}[f_i - \varepsilon_i]\right)\right)^2 \\
&\leq \sum_{i=1}^{m} \mathsf{Hel}\left(\bigotimes_{i=1}^{q_i} \mathsf{Ber}[f_i], \bigotimes_{i=1}^{q_i} \mathsf{Ber}[f_i - \varepsilon_i]\right)^2 \\
&\leq \sum_{i=1}^{m} \sum_{i=1}^{q_i} \mathsf{Hel}(\mathsf{Ber}[f_i], \mathsf{Ber}[f_i - \varepsilon_i])^2 \\
&\leq \sum_{i=1}^{m} \sum_{i=1}^{q_i} \frac{\varepsilon_i^2}{f_i(1 - f_i)} \\
&\leq \sum_{i=1}^{m} \sum_{i=1}^{q_i} \frac{4 \log^2(f^{-1})}{N'^2 f(1 - f)} \\
&\leq \frac{8 \log^2(f^{-1})}{f} \frac{q}{N'^2},
\end{aligned}$$

where the last inequality uses $1 - f \geq 1/2$. □

Corollary 4.

$$\Delta(\mathcal{D}_{\mathsf{ideal}}, \mathcal{D}'_{\mathsf{real}}) \leq \frac{4 \log(f^{-1})}{\sqrt{f}} \cdot \frac{\sqrt{q}}{N'}.$$

Proof. This is a direct consequence of Lemmas 11 and 23. □

Theorem 2 follows immediately from Lemma 21 and Corollary 4, which concludes the proof.

D Proof of Theorem 1

Let $\mathcal{A}$ be an adversary against the EUF-CMA-security of $\mathsf{HaS}_{\mathsf{T}}$. By definition, one has

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{HaS}_{\mathsf{T}}}(\mathcal{A}) = \Pr[\mathsf{HaS}_{\mathsf{T}}.\textsc{Verify}(\mathsf{pk}, \mathsf{msg}, \mathsf{sig}) = \top],$$

where $(\mathsf{pk}, \mathsf{sk}) \leftarrow S.\textsc{KeyGeneration}()$, $(\mathsf{msg}, \mathsf{sig}) \leftarrow \mathcal{A}^{\mathcal{H}, S.\textsc{Sign}^{\mathcal{H}}(\mathsf{sk}, \cdot)}(\mathsf{pk})$, and $S.\textsc{Sign}^{\mathcal{H}}(\mathsf{sk}, \cdot)$ was never queried on msg. Using Proposition 1, it is easy to see that, for some adversary $\mathcal{B}$ against the PS-security of T, one has

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{HaS}_{\mathsf{T}}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{PS}}_{\mathsf{T}}(\mathcal{B}) + \mathcal{O}\left(\frac{\log(f^{-1})}{\sqrt{f}} \cdot \frac{\sqrt{q}}{N}\right) + qe^{-\Omega(fN)} + q\varepsilon + \Pr[\mathsf{HaS}_{\mathsf{T}}.\textsc{Verify}(\mathsf{pk}, \mathsf{msg}, \mathsf{sig}) = \top],$$

where $(\mathsf{pk}, \mathsf{sk}) \leftarrow S.\textsc{KeyGeneration}()$, $(\mathsf{msg}, \mathsf{sig}) \leftarrow \mathcal{A}^{\mathcal{H}, S.\textsc{Ideal-Sign}^{\mathcal{H}}(\mathsf{sk}, \cdot)}(\mathsf{pk})$, and $S.\textsc{Ideal-Sign}^{\mathcal{H}}(\mathsf{sk}, \cdot)$ was never queried on msg. First note that the Ideal-Sign only requires access to the public key T.F

of $\mathsf{HaS}_{\mathsf{T}}$. Hence, it can be turned into an adversary $\mathcal{C}$ against the INV-security of T as follows. $\mathcal{C}$ starts the game by sampling an index $i \in \{1, \ldots, q_h\}$ uniformly at random. It will then run $\mathcal{A}$ and answer its queries by perfectly simulating the ideal-world signing oracle, except on its i-th random oracle query. In that case, it will answer the target value y that it has been given. When $\mathcal{A}$ outputs $(\mathsf{msg}, (\mathsf{salt}, \mathbf{s}))$, $\mathcal{C}$ checks if it corresponds to the i-th random oracle query. If that is the case, it outputs $\mathbf{s}$, and otherwise it aborts. The probability that $\mathcal{A}$ succeeds is exactly $\Pr[\mathsf{HaS}_{\mathsf{T}}.\textsc{Verify}(\mathsf{pk}, \mathsf{msg}, \mathsf{sig}) = \top]$, as the target value y was sampled uniformly at random. The probability that $\mathcal{A}$ succeeds, while it has never queried the random oracle on the input $\mathsf{msg}||\mathsf{salt}$ is at most $1/|\mathcal{Y}|$. Moreover, since no information on i is revealed to $\mathcal{A}$, the probability that it outputs a forgery for the i-th random oracle query is $1/q_h$. Hence, one has

$$\mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}}(\mathcal{C}) \geq \frac{\Pr[\mathsf{HaS}_{\mathsf{T}}.\textsc{Verify}(\mathsf{pk}, \mathsf{msg}, \mathsf{sig}) = \top] - |\mathcal{Y}|^{-1}}{q_h}.$$

E Proof in the adaptive setting

We use the same notation as the previous section.

E.1 Preliminaries

Definition 16 (Permutation invariance). *$\boldsymbol{X} = (X_i)_{i=1}^n$ be a sequence of n binary random variables. Let σ denote an arbitrary permutation of $\{1, \ldots, n\}$, and let $\sigma(\boldsymbol{X}) = (X_{\sigma(i)})_{i=1}^n$. We say that $\boldsymbol{X}$ is* permutation-invariant *if for all σ, $\boldsymbol{X}$ and $\sigma(\boldsymbol{X})$ follow the same distribution.*

E.2 Box-opening and Replacement Games

Definition 17 (Box-opening game). *Let n, q be two integers, representing respectively the number of boxes, and the numbers of queries. Let $\boldsymbol{X^0} = (X_i^0)_{i=1}^n$ and $\boldsymbol{X^1} = (X_i^1)_{i=1}^n$ be two sequences of n binary random variables. The* box-opening game $\mathsf{BoxG}_q\big(\boldsymbol{X^0}, \boldsymbol{X^1}\big)$ *is a two-player game between a challenger and an adversary. The game proceeds as follows.*

1. *The challenger picks $b \leftarrow_{\$} \{0,1\}$, and samples a vector $\boldsymbol{x} = (x_i)_{i=1}^n$ from the distribution $\boldsymbol{X^b}$.*
2. *The adversary asks q box-opening queries to the challenger. For the k-th box-opening query, the adversary adaptively chooses an index i_k in $[1, n]$, sends it to the challenger, and receives x_{i_k}.*
3. *After asking q box-opening queries, the adversary must pick a bit b'. The adversary wins if $b' = b$.*

Definition 18 (Replacement game). *Let n, q, $\boldsymbol{X^0}$, $\boldsymbol{X^1}$ be as in the box-opening game (Definition 17). Let $\boldsymbol{X'^0} = (X_i'^0)_{i=1}^n$ and $\boldsymbol{X'^1} = (X_i'^1)_{i=1}^n$ be two additional sequences of n binary random variables. Note that $\boldsymbol{X^0}$ (resp. $\boldsymbol{X^1}$) need not be independent from $\boldsymbol{X'^0}$ (resp. $\boldsymbol{X'^1}$). The* replacement game $\mathsf{RepG}_q\big((\boldsymbol{X^0}, \boldsymbol{X'^0}), (\boldsymbol{X^1}, \boldsymbol{X'^1})\big)$ *is a two-player game between a challenger and an adversary. The game proceeds as follows.*

1. *The challenger picks $b \leftarrow_{\$} \{0,1\}$, and samples a vector $\boldsymbol{x} = (x_i)_{i=1}^n$ from the distribution $\boldsymbol{X^b}$.*
2. *The adversary asks q box-opening queries to the challenger. For the k-th box-opening query, the adversary adaptively chooses an index i_k in $[1, n]$, sends it to the challenger, and receives x_{i_k}.*
3. *After asking q box-opening queries, the adversary picks $b' = 1$ if $\Pr[(X_{i_1}'^0, \ldots, X_{i_k}'^0) = (x_{i_1}, \ldots, x_{i_k})] \leq \Pr[(X_{i_1}'^1, \ldots, X_{i_k}'^1) = (x_{i_1}, \ldots, x_{i_k})]$, $b' = 0$ otherwise. The adversary wins if $b' = b$.*

Definition 19 (Selective adversaries). *Given a box-opening (resp. replacement) games G, we write G^{Sel} for the same game with the added constraint that the adversary must be selective. That is, all adversarial queries must be sampled at the start, independently of the challenger's answers. We use the term* adaptive *for the original game, where this added constraint does not exist.*

Definition 20 (Optimal advantage). *Given a box-opening (resp. replacement) games G, we write* $\mathsf{opt}(G)$ *for the advantage of the best adversary playing the game.*

The proof of the following two lemmas is immediate.

Lemma 24. *Let $q \leq n$ in $\mathbb{N}$, and let $\boldsymbol{X^0} = (X_i^0)_{i=1}^n$ and $\boldsymbol{X^1} = (X_i^1)_{i=1}^n$ be two sequences of n binary random variables. Then:*

$$\mathsf{opt}\big(\mathsf{BoxG}_q\big(\boldsymbol{X^0}, \boldsymbol{X^1}\big)\big) = \mathsf{opt}\big(\mathsf{RepG}_q\big((\boldsymbol{X^0}, \boldsymbol{X^0}), (\boldsymbol{X^1}, \boldsymbol{X^1})\big)\big).$$

Lemma 25. *Let $q \leq n$ in $\mathbb{N}$, and let $\boldsymbol{X^0}$, $\boldsymbol{X'^0}$, $\boldsymbol{X^1}$, $\boldsymbol{X'^1}$ be four sequences of n binary random variables. Suppose that $\boldsymbol{X'^0}$ and $\boldsymbol{X'^1}$ are both permutation-invariant. Then:*

$$\mathsf{opt}\big(\mathsf{RepG}_q\big((\boldsymbol{X^0}, \boldsymbol{X'^0}), (\boldsymbol{X^1}, \boldsymbol{X'^1})\big)\big) = \mathsf{opt}\Big(\mathsf{BoxG}_q^{\mathsf{Sel}}\big(\boldsymbol{X'^0}, \boldsymbol{X'^1}\big)\Big).$$

E.3 Main Argument

Setup. The main proof in the Section 4 is divided into two parts. The first part is a series of game hops. The second part is an analysis of the difference between the real and ideal worlds in the penultimate game. The only difference between the two worlds lies in the random oracle. Relevant queries to the random oracle are of the form "$\mathsf{msg}\|\mathsf{salt}$" (queries not of that form behave identically in both worlds, and have no impact on the proof).

Let us use the same notation as in the original selective proof (Appendix C). Since the adversary is limited to q queries, at most q distinct messages can be queried. Let us fix arbitrarily q pairwise distinct messages $(\mathsf{msg}_i)_{i=1}^q$. For each $\mathsf{msg}\|\mathsf{salt}$ query, the only relevant information about the message is whether it is equal to an already queried message. If it is distinct from all previously queried messages, then the exact choice of message has no bearing on the distribution. As a consequence, we can assume without loss of generality that the i-th *distinct* message queried by the adversary is msg_i. Let f_i denotes the probability that a salt for msg_i is suitable, for $i \in [1, q]_{\mathbb{N}}$.

As noted in the selective proof, for a given query $\mathsf{msg}\|\ \mathsf{salt}$, the distribution of the corresponding hash is determined by whether salt is suitable for msg or not. More precisely, in both the real and ideal worlds, the observed hash value can be expressed as a probabilistic function of a single bit equal to 1 if the pair is suitable, 0 otherwise. This function is the same in both worlds. As a consequence, for each query $\mathsf{msg}\|\ \mathsf{salt}$, instead of revealing the hash value to the adversary, we can reveal whether the pair is suitable, and this can only increase the probability of success of the adversary.

For $i \in [1, q]_{\mathbb{N}}$, $j \in [1, N]_{\mathbb{N}}$, let $X_{i,j}^{\mathsf{real}}$ (resp. $X_{i,j}^{\mathsf{ideal}}$) be the random variable equal to 1 if the j-th salt for msg_i is suitable in the real world (resp. in the ideal world), 0 otherwise. Let $\mathsf{w} \in \{\mathsf{real}, \mathsf{ideal}\}$. For fixed i, the vector $(X_{i,j}^{\mathsf{w}})_{j=1}^N$ is permutation-invariant, since all salts play a symmetrical role. As a consequence, the distribution of that vector is entirely determined by the random variable $S_i^{\mathsf{w}} = \sum_{j=1}^N X_{i,j}^{\mathsf{w}}$. Indeed, in both worlds, given S_i^{w}, the distribution of $(X_{i,j}^{\mathsf{w}})_{j=1}^N$ is uniformly random among binary vectors with Hamming weight exactly S_i^{w}.

Let $\boldsymbol{S^{\mathsf{w}}}$ denote the vector of random variables $(S_1^{\mathsf{w}}, \ldots, S_q^{\mathsf{w}})$. Given $\boldsymbol{S^{\mathsf{w}}}$, define $\mathsf{M}(\boldsymbol{S^{\mathsf{w}}})$ to be the distribution over $q \times N$ binary matrices defined by i-th row being sampled uniformly at random among binary vectors with Hamming weight S_i^{w}. The distribution of the variables $X_{i,j}^{\mathsf{w}}$ is $\mathsf{M}(\boldsymbol{S^{\mathsf{w}}})$. As a consequence, an adversary trying to distinguish the ideal world from the real world is exactly playing the box-opening game $\mathsf{BoxG}_q\big(\mathsf{M}(\boldsymbol{S^{\mathsf{ideal}}}), \mathsf{M}(\boldsymbol{S^{\mathsf{real}}})\big)$.

In conclusion, in order to upper-bound the probability of distinguishing the two worlds in the adaptive case, it suffices to upper-bound:

$$\mathsf{opt}\big(\mathsf{BoxG}_q\big(\mathsf{M}(\boldsymbol{S^{\mathsf{ideal}}}), \mathsf{M}(\boldsymbol{S^{\mathsf{real}}})\big)\big).$$

To fully describe those games, it remains to express $\boldsymbol{S}^{\mathsf{ideal}}$ and $\boldsymbol{S}^{\mathsf{real}}$. Following the analysis of the selective case, for $i \in \{1, \ldots, q\}$, we have:

$$S_i^{\mathsf{ideal}} \sim \mathsf{Bin}[N', f_i], \qquad S_i^{\mathsf{real}} \sim \mathsf{Bin}[N'+1, f_i] - 1.$$

Core result. The rest of this section is to prove the following theorem. Define $f = \min f_i$, as in the selective case.

Theorem 7 (main result, adaptive case).

$$\mathsf{opt}\big(\mathsf{BoxG}_q\big(\mathsf{M}(\boldsymbol{S}^{\mathsf{ideal}}), \mathsf{M}(\boldsymbol{S}^{\mathsf{real}})\big)\big) = \mathcal{O}\left(\frac{\log(f^{-1})}{\sqrt{f}} \cdot \frac{\sqrt{q}}{N'}\right) + qe^{-\Omega(fN')}.$$

E.4 Proof of Theorem 7

Notation. If $\boldsymbol{v} = (v_1, \ldots, v_n)$ is a vector, and I is a subset of $[1, n]_{\mathbb{N}}$, then v_I denotes the vector:

$$v_I = (v_{i_1}, \ldots, v_{i_{|I|}}),$$

where $i_1 < \cdots < i_{|I|}$ are the elements of I. Likewise, if $X = (X_1, \ldots, X_n)$ is a random variable, then X_I denotes $(X_{i_1}, \ldots, X_{i_{|I|}})$.

Lemma 26 (Replacement lemma, adaptive version). *Let $q \leq n \in \mathbb{N}$. Let P, Q, R, R' be random variables ranging over $\{0,1\}^n$. If Q_I covers P_I (with respect to $\leq_{|I|}$) for all $I \subseteq [1,n]_{\mathbb{N}}$, and $\Pr[P \geq_n R] = 1$, then:*

$$\mathsf{opt}(\mathsf{BoxG}_q(P, Q)) \leq \mathsf{opt}\big(\mathsf{RepG}_q((P, R), (Q, Q))\big).$$

More generally, if Q_I covers P_I for all I, and $\Pr[P \geq_\Omega R'] = 1$, then:

$$\mathsf{opt}(\mathsf{BoxG}_q(P, Q)) \leq \mathsf{opt}\big(\mathsf{RepG}_q((P, R), (Q, Q))\big) + \Delta(R, R').$$

Proof. Let us start with the first inequality. Our goal is to show that an optimal adversary has a higher advantage in the game $\mathsf{opt}\big(\mathsf{RepG}_q((P, R), (Q, Q))\big)$ than in the game $\mathsf{opt}(\mathsf{BoxG}_q(P, Q))$. In both games, the adversary is trying to distinguish between two worlds. In the case $b = 1$, those two worlds are the same in both games (since the original and replacement distributions is (Q, Q) on both sides). The difference occurs in the case $b = 0$. In that case, in the first game (resp. second game), the original and replacement distributions are (P, P) (resp. (P, R)). To prove the inequality, we show that for each possible realization of P, if the adversary guesses correctly in the first game, then she always guesses correctly in the second game.

Let $(\boldsymbol{p}, \boldsymbol{r})$ be an arbitrary realization of (P, R). Since $\Pr[P \geq_n R] = 1$, we have $\boldsymbol{p} \geq_n \boldsymbol{r}$. Without loss of generality, the adversary is deterministic. Hence the adversary's queries $i_1, \ldots, i_q$ are fixed by $\boldsymbol{p}$. Recall we are trying to show that if the adversary guesses correctly in the first game, then she also guesses correctly in the second game. Also recall that we are in the case $b = 0$. Assume that the adversary guesses correctly in the first game. We need to show that she also guesses correctly in the second game.

Let $I = \{i_1, \ldots, i_q\}$. Because the adversary guesses correctly in the first game, we have $\Pr[P_I = p_I] \geq \Pr[Q_I = p_I]$. Because Q_I covers P_I, this inequality remains true if we replace p_I by any vector $v \leq_q p_I$. In particular, we have $\Pr[P_I = r_I] \geq \Pr[Q_I = r_I]$, which means that the adversary guesses correctly in the second game.

The second inequality follows immediately by applying the data-processing inequality for the statistical distance to the map $v \mapsto W(\mathsf{RepG}_q((P,v),(Q,Q)))$ (assimilating v on the right-hand side with a constant distribution) on inputs R and R', where $W(G)$ is the random variable equal to 1 if the adversary wins the game G, 0 otherwise. □

The proofs of the next two lemmas are straightforward adaptions of the corresponding lemmas in the selective case (namely, Lemma 18 and Lemma 20), since the original distribution (and hence the distribution of adversarial queries) remains unchanged.

Lemma 27. *For all $I \subseteq [1,q]_{\mathbb{N}} \times [1,N']_{\mathbb{N}}$, $\mathsf{M}((\mathsf{Bin}[N', f_i])_{i=1}^q)_I$ covers $\mathsf{M}((\mathsf{Bin}[N'+1, f_i]-1)_{i=1}^q)_I$ with respect to $\preceq_{|I|}$.*

Define:

$$S_i'^{\mathsf{real}} = \mathsf{Bin}\big[N', f_i - 2\log(f_i^{-1})/N'\big].$$

Lemma 28. *There exist two distributions P and Q over $q \times N'$ binary matrices, such that $\Delta\big(P, S'^{\mathsf{real}}\big) = qe^{-\Omega(pN')}$, $\Delta\big(Q, S^{\mathsf{real}}\big) = qe^{-\Omega(pN')}$, and $P \preceq_{qN'} Q$.*

Putting the previous lemmas together, we get:

$$\begin{aligned}
&\mathsf{opt}\big(\mathsf{BoxG}_q\big(\mathsf{M}(\boldsymbol{S}^{\mathsf{ideal}}), \mathsf{M}(\boldsymbol{S}^{\mathsf{real}})\big)\big) \\
&= \mathsf{opt}\big(\mathsf{RepG}_q\big((\mathsf{M}(\boldsymbol{S}^{\mathsf{ideal}}), \mathsf{M}(\boldsymbol{S}^{\mathsf{ideal}})), (\mathsf{M}(\boldsymbol{S}^{\mathsf{real}}), \mathsf{M}(\boldsymbol{S}^{\mathsf{real}}))\big)\big) \\
&\leq \mathsf{opt}\big(\mathsf{RepG}_q\big((\mathsf{M}(\boldsymbol{S}^{\mathsf{ideal}}), \mathsf{M}(\boldsymbol{S}^{\mathsf{ideal}})), (\mathsf{M}(\boldsymbol{S}^{\mathsf{real}}), \mathsf{M}(\boldsymbol{S}'^{\mathsf{real}}))\big)\big) + qe^{-\Omega(pN')}.
\end{aligned}$$

Let:

$$\begin{aligned}
S_i''^{\mathsf{ideal}} &= \mathsf{Bin}[N', f] \\
S_i''^{\mathsf{real}} &= \mathsf{Bin}\big[N', f - 2\log(f^{-1})/N'\big].
\end{aligned}$$

Lemma 29. *It holds that:*

$$\begin{aligned}
&\mathsf{opt}\big(\mathsf{RepG}_q\big((\mathsf{M}(\boldsymbol{S}^{\mathsf{ideal}}), \mathsf{M}(\boldsymbol{S}^{\mathsf{ideal}})), (\mathsf{M}(\boldsymbol{S}^{\mathsf{real}}), \mathsf{M}(\boldsymbol{S}'^{\mathsf{real}}))\big)\big) \\
&\leq \mathsf{opt}\big(\mathsf{RepG}_q\big((\mathsf{M}(\boldsymbol{S}^{\mathsf{ideal}}), \mathsf{M}(\boldsymbol{S}''^{\mathsf{ideal}})), (\mathsf{M}(\boldsymbol{S}^{\mathsf{real}}), \mathsf{M}(\boldsymbol{S}''^{\mathsf{real}}))\big)\big)
\end{aligned}$$

Proof. We proceed by building a direct reduction between the two games. Let $\mathcal{A}$ be an adversary against the first game. We want to build an adversary $\mathcal{B}$ against the second game, such that the advantage of $\mathcal{B}$ (in the second game) is at least as high as the advantage of $\mathcal{A}$ (in the first game). If this is possible for an arbitrary $\mathcal{A}$, then by applying this construction to an optimal $\mathcal{A}$, we obtain the desired result.

Let $\mathcal{A}$ be an arbitrary adversary playing the first game. We now build an adversary $\mathcal{B}$ playing the second game. $\mathcal{B}$ begins by calling $\mathcal{A}$, and forwarding queries and answers back and forth between $\mathcal{A}$ and its own game instance, until all q queries $i_1, \ldots, i_q$ are issued. (It is worth noting that the distribution of query answers is the same in both games, whether we are in the ideal or real world.)

$\mathcal{B}$ then receives the replaced query answers $x_1, \ldots, x_q$. Let:

$$p_i = \frac{1 - f_{i_k}}{1 - f}.$$

For each $k \in [1,q]_{\mathbb{N}}$, $\mathcal{B}$ samples y_k from the distribution $x_k \vee \mathsf{Ber}[p_i]$, where $\vee$ represents the usual or operator on $\{0,1\}$. (Equivalently, if $x_k = 1$, then $y_k = 1$, otherwise y_k is sampled from $\mathsf{Ber}[p_i]$.)

By construction, each x_k was sampled according to $\mathsf{Ber}[f]$ in the ideal world, and $\mathsf{Ber}\big[f - 2\log(f^{-1})/N'\big]$ in the real world. The point of the y_k's is that, by choice of p_i, y_k is distributed according to $\mathsf{Ber}[f_{i_k}]$. That is, in the ideal world, $y_1, \ldots, y_q$ is sampled exactly as in the replacement distribution of the first game. On the other hand, in the real world, the distribution of y_k follows a Bernoulli distribution of parameter:

$$f_{i_k} - \frac{2\log(f^{-1})(1-p_i)}{N'}.$$

The replacement distribution in the first game would yield a parameter:

$$f_{i_k} - \frac{2\log(f_{i_k}^{-1})}{N'}.$$

Simple functional analysis shows that the second expression is always higher (this follows from the fact that $x \mapsto \log(1/x)/(1-x)$ is decreasing over the open interval $]0,1[$). Hence, given y_k, $\mathcal{B}$ can build a sample from $\mathsf{Ber}\left[f_{i_k} - \frac{2\log(f_{i_k}^{-1})}{N'}\right]$ by doing a mixture of y_k and $\mathsf{Ber}[f_{i_k}]$. Such as mixture leaves the ideal-world distribution $\mathsf{Ber}[f_{i_k}]$ of y_k unchanged.

In the end, this transformation maps the sample $y_1, \ldots, y_q$ from the replacement distribution of the second game to the replacement distribution of the first game, whether we are in the ideal world or the real world. The newly created sample can then be fed to the adversary $\mathcal{A}$, and $\mathcal{B}$ mirrors the final output of $\mathcal{A}$. Because all inputs of $\mathcal{A}$ have the correct distribution, the probability of success of $\mathcal{B}$ is exactly the same as $\mathcal{A}$. □

Observe that $\mathsf{M}(\boldsymbol{S}''^{\mathsf{ideal}})$ and $\mathsf{M}(\boldsymbol{S}''^{\mathsf{real}})$ are permutation-invariant. Hence we can apply Lemma 25 to get:

$$\begin{aligned}
&\mathsf{opt}\big(\mathsf{RepG}_q\big((\mathsf{M}(\boldsymbol{S}^{\mathsf{ideal}}), \mathsf{M}(\boldsymbol{S}''^{\mathsf{ideal}})), (\mathsf{M}(\boldsymbol{S}^{\mathsf{real}}), \mathsf{M}(\boldsymbol{S}''^{\mathsf{real}}))\big)\big) \\
&= \mathsf{opt}\Big(\mathsf{BoxG}_q^{\mathsf{Sel}}\big(\mathsf{M}(\boldsymbol{S}''^{\mathsf{ideal}}), \mathsf{M}(\boldsymbol{S}''^{\mathsf{real}})\big)\Big).
\end{aligned}$$

Thus, we reduce to the selective case. Whence the exact same analysis as Corollary 4 yields:

$$\begin{aligned}
&\mathsf{opt}\Big(\mathsf{BoxG}_q^{\mathsf{Sel}}\big(\mathsf{M}(\boldsymbol{S}''^{\mathsf{ideal}}), \mathsf{M}(\boldsymbol{S}''^{\mathsf{real}})\big)\Big) \\
&= \Delta\big(\mathsf{Bin}[q, f], \mathsf{Bin}\big[q, f - 2\log(f^{-1})/N'\big]\big) \\
&\leq \frac{4\log(f^{-1})}{\sqrt{f}} \cdot \frac{\sqrt{q}}{N'},
\end{aligned}$$

which concludes the proof of Theorem 7.