Random Oracles and Non-Uniformity

Sandro Coretti*
New York University
corettis@nyu.edu

Yevgeniy Dodis†
New York University
dodis@cs.nyu.edu

Siyao Guo‡
Northeastern University
s.guo@neu.edu

John Steinberger
jpsteinb@gmail.com

August 8, 2022

Abstract

We revisit security proofs for various cryptographic primitives in the *auxiliary-input random-oracle model* (AI-ROM), in which an attacker $\mathcal{A}$ can compute arbitrary S bits of leakage about the random oracle $\mathcal{O}$ before attacking the system and then use additional T oracle queries to $\mathcal{O}$ during the attack. This model has natural applications in settings where traditional random-oracle proofs are not useful: (a) security against non-uniform attackers; (b) security against preprocessing. We obtain a number of new results about the AI-ROM:

- Unruh (CRYPTO '07) introduced the *pre-sampling technique*, which generically reduces security proofs in the AI-ROM to a much simpler *P-bit-fixing random-oracle model* (BF-ROM), where the attacker can arbitrarily fix the values of $\mathcal{O}$ on some P coordinates, but then the remaining coordinates are chosen at random. Unruh's security loss for this transformation is $\sqrt{ST/P}$. We improve this loss to the *optimal* value $O(ST/P)$, obtaining nearly tight bounds for a variety of indistinguishability applications in the AI-ROM.
- While the basic pre-sampling technique cannot give tight bounds for unpredictability applications, we introduce a novel "multiplicative version" of pre-sampling, which allows to dramatically reduce the size of P of the pre-sampled set to $P = O(ST)$ and yields nearly tight security bounds for a variety of unpredictability applications in the AI-ROM. Qualitatively, it validates Unruh's "polynomial pre-sampling conjecture"—disproved in general by Dodis *et al.* (EUROCRYPT '17)—for the special case of unpredictability applications.
- Using our techniques, we reprove nearly all AI-ROM bounds obtained by Dodis *et al.* (using a much more laborious compression technique), but we also apply it to many settings where the compression technique is either inapplicable (e.g., computational reductions) or appears intractable (e.g., Merkle-Damgård hashing).
- We show that for any *salted* Merkle-Damgård hash function with m-bit output there exists a collision-finding circuit of size $\Theta(2^{m/3})$ (taking salt as the input), which is significantly below the $2^{m/2}$ birthday security conjectured against uniform attackers.
- We build two compilers to generically extend the security of applications proven in the traditional ROM to the AI-ROM. One compiler simply prepends a public salt to the random oracle, showing that *salting generically provably defeats preprocessing.*

Overall, our results make it much easier to get concrete security bounds in the AI-ROM. These bounds in turn give concrete conjectures about the security of these applications (in the standard model) against *non-uniform* attackers.

*Supported by NSF grants 1314568 and 1319051.

†Partially supported by gifts from VMware Labs and Google, and NSF grants 1619158, 1319051, 1314568.

‡Supported by NSF grants CNS1314722 and CNS-1413964. Work partially done at the Simons Institute for the Theory of Computing at UC Berkeley.

1 Introduction

We start by addressing the two main themes of this work—non-uniformity and random oracles—in isolation, before connecting them to explain the main motivation for this work.

Non-uniformity. Modern cryptography (in the "standard model") usually models the attacker $\mathcal{A}$ as non-uniform, meaning that it is allowed to obtain some arbitrary (but bounded) "advice" before attacking the system. The main rationale to this modeling comes from the realization that a determined attacker will know the security parameter n of the system in advance and might be able to invest a significant amount of preprocessing to do something "special" for this fixed value of n, especially if n is not too large (for reasons of efficiency), or the attacker needs to break a lot of instances online (therefore amortizing the one-time offline cost). Perhaps the best known example of such attacks comes from *rainbow tables* ([32, 47]; see also [39, Section 5.4.3]) for inverting arbitrary functions; the idea is to use one-time preprocessing to initialize a clever data structure in order to dramatically speed up brute-force inversion attacks. Thus, restricting to uniform attackers might not accurately model realistic preprocessing attacks one would like to protect against. However, there are other, more technical, reasons why this choice is convenient:

- Adleman [2] showed that non-uniform polynomial-time attackers can be assumed to be deterministic (formally, $BPP/poly = P/poly$), which is handy for some proofs.
- While many natural reductions in cryptography are uniform, there are several important cases where the only known (or even possible!) reduction is non-uniform. Perhaps the best known example are zero-knowledge proofs [29, 28], which are *not* closed under *sequential composition* unless one allows non-uniform attackers (and simulators; intuitively, in order to use the simulator for the second zero-knowledge proof, one must use the output of the first proof's simulator as an auxiliary input to the verifier).[1] Of course, being a special case of general protocol composition, this means that any work—either using zero-knowledge as a subroutine or generally dealing with protocol composition—must use security against *non-uniform* attackers in order for the composition to work.
- The non-uniform model of computation has many applications in complexity theory, such as the famous "hardness-vs-randomness" connection (see [46, 35, 36, 34, 37]), which roughly states that *non-uniform* hardness implies non-trivial de-randomization. Thus, by defining cryptographic attackers as non-uniform machines, any lower bounds for such cryptographic applications might yield exciting de-randomization results.

Of course, despite the pragmatic, definitional, and conceptual advantages of non-uniformity, one must ensure that one does not make the attacker "too powerful," so that it can (unrealistically) solve problems which one might use in cryptographic applications. Fortunately, although non-uniform attackers can solve undecidable problems (by encoding the input in unary and outputting solutions in the non-uniform advice), the common belief is that non-uniformity cannot solve interesting "hard problems" in polynomial time. As one indirect piece of evidence, the Karp-Lipton theorem [38] shows that if NP has polynomial-size circuits, then the polynomial hierarchy collapses. And, of course, the entire field of cryptography is successfully based on the assumption that many hard problems cannot be solved even on average by polynomially sized circuits, and this belief has not been seriously challenged so far.

[1] There are some workarounds (see [27]) that permit one to define zero-knowledge under uniform attackers, but they are much harder to work with than assuming non-uniformity, and, as a result, were not adopted by the community.

Hence, by and large it is believed by the theoretical community that *non-uniformity is the right cryptographic modeling of attackers*, despite being overly conservative and including potentially unrealistic attackers.

The random-oracle model. Hash functions are ubiquitous in cryptography. They are widely used to build one-way functions (OWFs), collision-resistant hash functions (CRHFs), pseudorandom functions/generators (PRFs/PRGs), message authentication codes (MACs), etc. Moreover, they are often used together with other computational assumptions to show security of higher-level applications. Popular examples include Fiat-Shamir heuristics [24, 1] for signature schemes (e.g., Schnorr signatures [50]), full-domain-hash signatures [8], or trapdoor functions (TDFs) [8] and OAEP [9] encryption, among many others.

For each such application Q, one can wonder how to assess its security ε when instantiated with a concrete hash function H, such as SHA-3. Given our inability to prove unconditional lower bounds, the traditional approach is the following: Instead of proving an upper bound on ε for some specific H, one analyzes the security of Q assuming H is a *truly random (aka "ideal") function* $\mathcal{O}$. Since most Q are only secure against computationally bounded attackers, one gives the attacker $\mathcal{A}$ oracle access to $\mathcal{O}$ and limits the number of oracle queries that $\mathcal{A}$ can make by some parameter T. This now becomes the traditional *random-oracle model (ROM)*, popularized by the seminal paper of Bellare and Rogaway [8].

The appeal of the ROM stems from two aspects. First, it leads to very clean and intuitive security proofs for many primitives that resisted standard-model analysis under natural security assumptions (see some concrete examples below). Second, this resulting ROM analysis is *independent of the tedious specifics* of H, is done only *once* for a given hash-based application, and also provides (for non-pathological Q's) the *best possible* security one might hope to achieve with any concrete function H. In particular, we hope that a specific hash function H we use is sufficiently "well-designed" that it (essentially) matches this idealized bound. If it does, then our bound on ε was accurate anyway; and, if it does not, this usually serves as strong evidence that we should not use this particular H, rather than the indication that the idealized analysis was the wrong way to guess the exact security of Q. Ironically, in theory we know that the optimistic methodology above is false [12, 11, 45, 30, 5], and some applications secure in the ROM will be insecure for any instantiation of H, let alone maintain the idealized bound on ε. Fortunately, all counterexamples of this kind are rather artificial, and do not shed much light on the security of concrete schemes used in practice, such as the use of hash functions as OWFs, CRHFs, PRFs, PRGs, MACs, and also as parts of natural signature and encryption schemes used in practice [24, 50, 9, 8]. In other words, despite purely theoretical concerns, the following *random-oracle methodology* appears to be a good way for practitioners to assess the best possible security level of a given (natural) application Q.

> **Random-oracle methodology.** *For "natural" applications of hash functions, the concrete security proven in the random-oracle model is the right bound even in the standard model, assuming the "best possible" concrete hash function H is chosen.*

Random oracles and non-uniformity. The main motivation for this work is to examine the soundness of the above methodology, while also being consistent with the fact that attackers should be modeled as *non-uniform*. We stress that we are not addressing the conceptual question of whether non-uniform security is the "right" way to model attackers in cryptography, as this is the subject of a rather heated on-going debate between theoreticians and practitioners; see [49, 10] for some discussion on the subject. Instead, assuming we want to model attackers as non-uniform (for the

reasons stated above and to be consistent with the theoretical literature), and assuming we want to have a way of correctly assessing the *concrete*, non-asymptotic security for important uses of hash functions in applications, we ask: is the random oracle methodology a sound way to achieve this goal? Unfortunately, with the traditional modeling of the random oracle, the answer is a resounding "NO," even for the *most basic* usages of hash functions, as can be seen from the following examples.

(i) In the standard model, no single function H can be collision-resistant, as a non-uniform attacker can trivially hardwire a collision. In contrast, a single (non-salted) random oracle $\mathcal{O}$ is trivially collision-resistant in the ROM, with excellent exact security $O(T^2/M)$, where M is the range of $\mathcal{O}$. This is why in the standard model one considers a *family* of collision-resistant hash functions whose public key, which we call *salt*, is chosen *after* $\mathcal{A}$ gets its non-uniform advice. Interestingly, one of the results in this paper will show that the large gap (finding collisions in time $M^{1/2}$ vs. $M^{1/3}$) between uniform and non-uniform security exists for the popular Merkle-Damgård construction *even if salting is allowed*.

(ii) In the standard model, no PRG candidate $H(x)$ can have security better than $2^{-n/2}$ even against linear-time (in n) attackers [3, 21, 10], where n is the seed-length of x. In contrast, an expanding random oracle $\mathcal{O}(x)$ can be trivially shown to be $(T/2^n)$-secure PRG in the traditional ROM, easily surpassing the $2^{-n/2}$ barrier in the standard model (even for huge T up to $2^{n/2}$, let alone polynomial T).

(iii) The seminal paper of Hellman [32], translated to the language of non-uniform attackers, shows that a random function $H : [N] \to [N]$ can be inverted with constant probability using a non-uniform attacker of size $O\left(N^{2/3}\right)$, while Fiat and Naor [23] extended this attack to show that every (even non-random) function H can be inverted with constant probability by circuits of size at most $N^{3/4}$. In contrast, if one models H as a random oracle $\mathcal{O}$, one can trivially show that $\mathcal{O}$ is a OWF with security $O\left(T/N\right)$ in the traditional ROM. For example, setting $T = N^{2/3}$ (or even $T = N^{3/4}$), one would still get negligible security $N^{-1/3}$ (or $N^{-1/4}$), contradicting the concrete non-uniform attacks mentioned above.

To put it differently, once non-uniformity is allowed in the standard model, the separations between the random-oracle model and the standard model are *no longer* contrived and artificial but rather lead to *impossibly good* exact security of *widely deployed* applications.

Auxiliary-input ROM. The above concern regarding the random-oracle methodology is not new and was extensively studied by Unruh [52] and Dodis *et al.* [19]. Fortunately, these works offered a simple solution, by extending the traditional ROM to also allow for oracle-dependent *auxiliary input*. The resulting model, called the *auxiliary-input random-oracle model (AI-ROM)*, is parameterized by two parameters S ("space") and T ("time") and works as follows: First, as in the traditional random-oracle model, a function $\mathcal{O}$ is chosen uniformly from the space of functions with some domain and range. Second, the attacker $\mathcal{A}$ in the AI-ROM consists of two entities $\mathcal{A}_1$ and $\mathcal{A}_2$. The first-stage attacker $\mathcal{A}_1$ is computationally unbounded, gets full access to the random oracle $\mathcal{O}$, and computes some "non-uniform" advice z of size S. This advice is then passed to the second-stage attacker $\mathcal{A}_2$, who may make up to T queries to oracle $\mathcal{O}$ (and, unlike $\mathcal{A}_1$, might have additional application-specific restrictions, like bounded running time, etc.). This naturally maps to the preprocessing model discussed earlier and can also be used to analyze security against non-uniform circuits of size C by setting $S = T = C$.[2] Indeed, none of the concerns expressed in

[2]But separating S and T can also model non-uniform RAM computation with memory S and query complexity T.

examples (i)-(iii) remain valid in AI-ROM: (i) $\mathcal{O}$ itself is no longer collision-resistant since $\mathcal{A}_1$ can precompute a collision; (ii)-(iii) the generic non-uniform PRG or OWF attacks mentioned earlier can also be performed on $\mathcal{O}$ itself (by letting $\mathcal{A}_1$ treat $\mathcal{O}$ as any other function H and computing the corresponding advice for $\mathcal{A}_2$). In sum, the AI-ROM model allows us to restate the modified variant of the random oracle methodology as follows:

> **AI-Random-Oracle Methodology.** *For "natural" applications of hash functions, the concrete security proven in the AI-ROM is the right bound even in the standard model against non-uniform attackers, assuming the "best possible" concrete hash function H is chosen.*

Dealing with auxiliary information. The AI-ROM yields a clean and elegant way towards obtaining meaningful non-uniform bounds for natural applications. Unfortunately, obtaining such bounds is considerably more difficult than in the traditional ROM. In retrospect, such difficulties are expected, since we already saw several examples showing that non-uniform attackers are *very powerful* when exact security matters, which means that the security bounds obtained in the AI-ROM might often be noticeably weaker than in the traditional ROM. From a technical point, the key difficulty is this: *conditioned on the leaked value z*, which can depend on the entire function table of $\mathcal{O}$ in some non-trivial manner, many of the individual values $\mathcal{O}(x)$ are no longer random to the attacker. And this ruins many of the key techniques utilized in the traditional ROM, such as: (1) *lazy sampling*, which allows the reduction to sample the not-yet-queried values of $\mathcal{O}$ at random, as needed, without worrying that such lazy sampling will be inconsistent with the past; (2) *programmability*, which allows the reduction to dynamically define some value of $\mathcal{O}$ in a special (still random) way, as this might be inconsistent with the leakage value z it has to produce *before* knowing how and where to program $\mathcal{O}$; (3) *distinguishing-to-extraction argument*, which states that the attacker cannot distinguish the value of $\mathcal{O}$ from random without explicitly querying it (which again is false given auxiliary input). For these reasons, new techniques are required for dealing with the AI-ROM. Fortunately, two such techniques are known:

- **Pre-sampling technique.** This beautiful technique was introduced in the original, pioneering work of Unruh [52]. From our perspective, we will present Unruh's pre-sampling technique in a syntactially different (but technically equivalent) way which will be more convenient for our presentation. Specifically, Unruh implicitly introduced an intermediate oracle model, which we term the *bit-fixing random-oracle model (BF-ROM)*,[3] which can be arbitrarily fixed on some P coordinates, but then the remaining coordinates are chosen at random and independently of the fixed coordinates. Moreover, the non-uniform S-bit advice of the attacker can only depend on the P fixed points, but not on the remaining truly random points. Intuitively, dealing with the BF-ROM—at least when P is small—appears to be much easier than with the AI-ROM, as many of the traditional ROM proof techniques can be adapted provided that one avoids the "pre-sampled" set. Quite remarkably, for any value P, Unruh showed that any (S,T)-attack in the AI-ROM will have similar advantage in (appropriately chosen) P-BF-ROM, up to an additive loss of $\delta(S,T,P)$, which Unruh upper bounded by $\sqrt{ST/P}$. This yields a general recipe for dealing with the AI-ROM: (a) prove security $\varepsilon(S,T,P)$ of the given application in the P-BF-ROM;[4] (b) optimize for the right value of P by balancing $\varepsilon(S,T,P)$ and $\delta(S,T,P)$ (while also respecting the time and other constraints of the attacker).

[3] This naming in inspired by the bit-fixing source [13] from complexity theory.

[4] Observe that the parameter S is still meaningful here. $\mathcal{A}_1$ fixes $\mathcal{O}$ at P points but only passes S bits of advice to $\mathcal{A}_2$. While none of information-theoretic proofs in this paper really use this, for computational reductions S "passes

- **Compression technique.** Unfortunately, Dodis *et al.* [19] showed that the concrete security loss $\delta(S,T,P) = \sqrt{ST/P}$ proven by Unruh is not strong enough to get tight bounds for any of the basic applications of hash functions, such as building OWFs, PRGs, PRFs, (salted) CRHFs, and MACs. To remedy the situation, Dodis *et al.* [19] showed a different, less general technique for dealing with the AI-ROM, by adapting the *compression paradigm*, introduced by Gennaro and Trevisan [26, 25] in the context of black-box separations, to the AI-ROM. The main idea is to argue that if some AI-ROM attacker succeeds with high probability in breaking a given scheme, then that attacker can be used to reversibly encode (i.e., compress) a random oracle beyond what is possible from an information-theoretic point of view. Since we are considering attackers who perform preprocessing, our encoding must include the S-bit auxiliary information produced by the attacker. Thus, the main technical challenge in applying this technique is to ensure that the constructed encoding compress by (significantly) more than S bits. Dodis *et al.* [19] proceeded by successfully applying this idea to show nearly tight (and always better than what was possible by pre-sampling) bounds for a variety of natural applications, including OWFs, PRGs, PRFs, (salted) CRHFs, and MACs.

Pre-sampling or compression? The pre-sampling and compression techniques each have their pros and cons, as discussed below.

On a positive, pre-sampling is very general and seems to apply to most applications, as analyzing the security of schemes in BF-ROM is not much harder than in the traditional ROM. Moreover, as shown by Unruh, the pre-sampling technique appears at least "partially friendly" to computational applications of random oracles (specifically, Unruh applied it to OAEP encryption [9]). Indeed, if the size P of the pre-sampled set is not too large, then it can be hardwired as part of non-uniform advice to the (efficient) reduction to the computational assumption. In fact, in the asymptotic domain Unruh even showed that the resulting security remains "negligible in security parameter λ," despite not being smaller than any concrete negligible function (like the inverse Ackermann function).[5]

On a negative, the *concrete* security bounds which are currently obtainable using this technique are vastly suboptimal, largely due to the big security loss $\sqrt{ST/P}$ incurred by using Unruh's bound [52]. Moreover, for computational applications, the value of P cannot be made larger than the size of attacker for the corresponding computational assumption. Hence, for fixed ("non-asymptotic"; see Footnote 5) polynomial-size attackers, the loss $\sqrt{ST/P}$ cannot be made negligible. Motivated by this, Unruh conjectured that the security loss of pre-sampling can be improved by a tighter proof. Dodis *et al.* [19] showed that the best possible security loss is at most ST/P. For computational applications, this asymptotically disproves Unruh's conjecture, as ST/P is still non-negligible for polynomial values of P (although we will explain shortly that the situation is actually more nuanced).

Moving to the compression technique, we already mentioned that it led Dodis *et al.* [19] to establishing nearly tight AI-ROM bounds for several information-theoretic applications of random oracles. Unfortunately, each proof was noticeably more involved than the original ROM proof, or than the proof in the BF-ROM one would do if applying the more intuitive pre-sampling technique. Moreover, each primitive required a completely different set of algorithmic insights to get the re-

through" for the final non-uniform attacker against the computational assumption, and it is necessary to have $S \ll P$ in this case.

[5] Any AI-ROM attacker of size $t = t(\lambda)$ getting inverse polynomial advantage $\delta = 1/p(\lambda)$ for infinitely many λ's has advantage $\delta - \sqrt{ST/P}$ in the BF-ROM, which can be made to be $\delta/2$ by suitably choosing $P \approx O(t^2/\delta^2)$, which is polynomial and therefore suited for a reduction to a computational hardness assumption.

quired level of compression. And it is not entirely clear how far this can go. For example, we do not see any way to apply the compression paradigm to relatively basic applications of hash functions beyond using the hash function *by itself* as a given primitive; e.g., to show AI-ROM security of the classical Merkle-Damgård paradigm [43, 17] (whose *tight* AI-ROM security we will later establish in this work). Moreover, unlike pre-sampling, the compression paradigm cannot be applied at all to computational applications, as the compressor and the decompressor are computationally unbounded.

1.1 Our Results

We obtain a number of results about dealing with the AI-ROM, which, at a high-level, take the best features from pre-sampling (simplicity, generality) and compression (tightness).

Improving Unruh. Recall, Unruh [52] showed that one can move from the AI-ROM to the P-BF-ROM at the additive cost $\delta(S,T,P) \leq \sqrt{ST/P}$, and Dodis *et al.* [19] showed that $\delta(S,T,P) = \Omega\left(ST/P\right)$ in general. We show that the true additive error bound is indeed $\delta(S,T,P) = \Theta(ST/P)$, therefore improving Unruh's bound by a quadratic factor; see Theorem 5. Namely, the effect of S bits of auxiliary information $z = z(\mathcal{O})$ against an attacker making T adaptive random-oracle queries can be simulated to within an additive error $O(ST/P)$ by fixing the value of the random oracle on P points (which depend on the function z), and picking the other points at random and independently of the auxiliary information.

While the quadratic improvement might appear "asymptotically small," we show that it already matches the near-tight bound for all indistinguishability applications (specifically, PRGs and PRFs) proved by [19] using much more laborious compression arguments. For example, to match the $\varepsilon = O(\sqrt{ST/N} + T/N)$ bound for PRGs with seed domain N, we show using a simple argument that the random oracle is $\varepsilon' = O(P/N + T/N)$-secure in the P-BF-ROM, where the first term corresponds to the seed being chosen from the pre-sampled set, and the second term corresponds to the probability of querying the oracle on the seed in the attack stage. Setting $P = O(\sqrt{STN})$ to balance the P/N and ST/P terms, we immediately get our final bound, which matches that of [19]. For illustrative purposes, we also apply our improved bound to argue the AI-ROM security of a couple of indistinguishability applications not considered by [19]. First, we show an improved—compared to its use as a (standard) PRF—bound for the random oracle as a *weak* PRF, which is enough for chosen-plaintext secure symmetric-key encryption. Our proof is a very simple adaptation of the PRF proof in the BF-ROM, while we believe the corresponding compression proof, if possible at all, would involve noticeable changes to the PRF proof of [19] (due to the need for better compression to get the improved bound). Second, we also apply it to a typical example of a computational application, namely, the (KEM-variant of the) TDF-based public-key encryption scheme $\mathsf{Enc}_f(m;x) = (f(x), \mathcal{O}(x) \oplus m)$ from the original Bellare-Rogaway paper [8], where f is a trapdoor permutation (part of the public key, while the inverse is the secret key) and x is the randomness used for encryption. Recall that the compression technique cannot be applied to such applications.

To sum up, we conjecture that the improved security bound ST/P should be sufficient to get good bounds for most natural indistinguishability applications; these bounds are either tight, or at least they match those attainable via compression arguments (while being much simpler and more general).

Improved pre-sampling for unpredictability applications. Even with our improved bound of ST/P for pre-sampling, we will not match the nearly tight compression bounds obtained by Dodis *et al.* [19] for OWFs and MACs. In particular, finding the optimal value of P will result in "square root terms" which are not matched by any existing attacks. As our key insight, we notice that this is not due to the limitations of pre-sampling (i.e., going through the BF-ROM), but rather to the fact that achieving an *additive* error is unnecessarily restrictive for *unpredictability* applications. Instead, we show that if one is happy with a multiplicative factor of 2 in the probability of breaking the system, then one can achieve this *generically* by setting the pre-sampling set size $P \approx ST$; see Theorem 6.

This has a number of implications. First, with this multiplicative pre-sampling technique, we can easily match the compression bounds for the OWF and MAC unpredictability applications considered by Dodis *et al.* [19], but with much simpler proofs. Second, we also apply it to a natural information-theoretic application where we believe the compression technique will fail to get a good bound; namely, building a (salted) CHRF family via the Merkle-Damgård paradigm, where the salt is the initialization vector for the construction (see Theorem 12). The salient feature of this example is that the random oracle is applied in iteration, which poses little difficulties to adapting the standard-ROM proof to the BF-ROM, but seems to completely blow up the complexity of the compression arguments, as there are too many possibilities for the attacker to cause a collision for different salts when the number of blocks is greater than 1.[6] The resulting AI-ROM bound $O(ST^2/M)$ becomes vacuous for circuits of size roughly $M^{1/3}$, where M is the range of the compression function. This bound is well below the conjectured $M^{1/2}$ birthday security of CRHFs based on Merkle-Damgård against uniform attackers. Quite unexpectedly, we show that $M^{1/3}$ security we prove *is tight*: there exists a (non-uniform) collision-finding attack implementable by a circuit of size $O\left(M^{1/3}\right)$ (see Theorem 13)! This example illustrates once again the the surprising power of non-uniformity.

Implications to computational reductions. Recall that, unlike compression techniques, pre-sampling can be applied to computational reductions, by "hardwiring" the pre-sampling set of size P into the attacker breaking the computational assumption. However, this means that P cannot be made larger than the maximum allowed running time t of such an attacker. Since standard pre-sampling incurs additive cost $\Omega(ST/P)$, one cannot achieve final security better that ST/t, irrespective of the value of ε in the (t, ε)-security of the corresponding computational assumption. For example, when t is polynomial (in the security parameter) and $\varepsilon \ll 1/t$ is exponentially small, we only get inverse polynomial security (at most ST/t) when applying standard pre-sampling. In contrast, the multiplicative variant of pre-sampling sets the list size to be roughly $P \approx ST$, which is polynomial for polynomial S and T and can be made smaller than the complexity t of the standard model attacker for the computational assumption we use. Thus, when t is polynomial and ε is exponentially small, we will get negligible security using multiplicative pre-sampling. For a concrete illustrative example, see the bound in Theorem 15 when we apply our improved pre-sampling to the natural *computational unpredictability* application of Schnorr signatures [50].[7] To put it differently, while the work of Dodis *et al.* [19] showed that Unruh's "pre-sampling conjecture" is false in general—meaning that negligible security is not possible with a polynomial list size P—we show that it is qualitatively true for *unpredictability applications*, where the list size can be made

[6] The same difficulty of compression should also apply to indistinguishability applications of Merkle-Damgård, such as building PRFs [6].

[7] Interestingly, general Fiat-Shamir transform is *not* secure in AI-ROM, and thus our proof used the specifics of Schnorr's signatures.

polynomial (roughly ST).

Moreover, we show that in certain computational *indistinguishability* applications, we can still apply our improved pre-sampling technique *inside the reduction*, and get final security higher than the ST/t barrier mentioned above. We illustrate this phenomenon in our analysis of TDF encryption (cf. Theorem 16) by separating the probability of the attacker's success into 2 disjoint events: (1) the attacker, given ciphertext $f(x)$, managed to query the random oracle on the TDP preimage x ; (2) the attacker succeeds in distinguishing the value $\mathcal{O}(x)$ from random without querying $\mathcal{O}(x)$. Now, for the event (1), we can reduce to the TDP security with polynomial list size using our improved *multiplicative* pre-sampling (since is an *unpredictability* event), while for the event (2), we can prove *information-theoretic* security using standard additive pre-sampling, without the limitation of having to upper bound P by the running time of the TDP attacker. It is an interesting open question to classify precisely the type of indistinguishability applications where such "hybrid" reduction technique can be applied.

Going to the traditional ROM. So far, the general paradigm we used is to reduce the hard-to-analyze security of *any* scheme in the AI-ROM to the much simpler and proof-friendly security of the *same* scheme in the BF-ROM. However, an even simpler approach, if possible, would be to reduce the security in the AI-ROM all the way to the traditional ROM. Of course, we know that this is impossible without any modifications to the scheme, as we have plenty of examples where the AI-ROM security of the scheme is much weaker than its ROM security (or even disappears completely). Still, when a simple modification is possible without much inconvenience to the users, reducing to the ROM has a number of obvious advantages over the BF-ROM:

- While much simpler than in the AI-ROM, one must still prove a security bound in BF-ROM. It would be much easier if one could just utilize an already proven result in ROM and seamlessly "move it" to the AI-ROM at a small cost.
- Some natural schemes secure in the traditional ROM are *insecure* in the BF-ROM (and also in the AI-ROM) without any modifications. Simple example include the general Fiat-Shamir heuristic [24, 1] or the FDH signature scheme [8] (see Section C.1). Thus, to extend such schemes to the AI-ROM, we must modify them anyway, so we might as well try to generically ensure that ROM security is already enough.

As our next set of results, we show two simple compilers which build a hash function $\mathcal{O}'$ to be used in AI-ROM application out of hash function $\mathcal{O}$ used in the traditional ROM application. Both results are in the common-random-string model. This means that they utilize a public random string (which we call salt and denote a) chosen *after* the auxiliary information about $\mathcal{O}$ is computed by the attacker. The honest parties are then assumed to have reliable access to this a value. We note that in basic applications, such as encryption and authentication, the salt can simply be chosen at key generation and be made part of the public key/parameters, so this comes at a small price indeed.

The first transformation analyzed in Section 6.1 is simply *salting*; namely $\mathcal{O}'_a(x) = \mathcal{O}(a, x)$, where a is a public random string chosen from the domain of size K. This technique is widely used in practice (going back to password hashing [44]), and was analyzed by Dodis *et al.* [19] in the context of AI-ROM, by applying the compression argument to show that *salting provably defeats preprocessing* for the few natural applications they consider (OWFs, PRGs, PRFs, and MACs). What our work shows is that salting provably defeats pre-processing *generically*, as opposed to a

few concrete applications analyzed by [19].[8] Namely, by making the salt domain K large enough, one gets almost the same security in AI-ROM than in the traditional ROM. To put differently, when salting is possible, one gets the *best of both worlds: security against non-uniform attacks, but with exact security matching that in the traditional ROM.*

The basic salting technique sacrificed a relatively large factor of K from the domain of the random oracle $\mathcal{O}$ in order to build $\mathcal{O}'$ (for K large enough to bring the "salting error" down). When the domain of $\mathcal{O}$ is an expensive resource, in Section 6.2 we also design a more domain-efficient compiler, which only sacrifices a small factor $k \geq 2$ in the domain of $\mathcal{O}$, at the cost that each evaluation of $\mathcal{O}'$ takes $k \geq 2$ evaluations of $\mathcal{O}$ (and the "salting error" decays exponentially in k). This transformation is based on the adaptation of the technique of Maurer [42], originally used in the context of key-agreement with randomizers. While the basic transformation needs $O(k \log N)$ bits of public salt, we also show than one can reduce the number of random bits to $O(k + \log N)$. And since we do not envision k to be larger than $O(\log N)$ for any practical need, the total length of the salt is always $O(\log N)$.

Our main lemma. The key technical contribution of our work is Lemma 1, proved in Section 2.1, which roughly shows that a random oracle with auxiliary input is "close" to the convex combination of "P-bit-fixing sources" (see Definition 1). Moreover, we give both additive and multiplicative versions of this "closeness," so that we can later use different parameters to derive our Theorem 5 (for indistinguishability applications in the AI-ROM) and Theorem 6 (for unpredictability applications in the AI-ROM) in Section 2.2.

1.2 Other Related Work

Most of the related work was already mentioned earlier. The realization that multiplicative error is enough for unpredictability applications, and this can lead to non-trivial savings, is related to the work of Dodis *et al.* [20] in the context of improved entropy loss of key derivation schemes. Tessaro [51] generalized Unruh's presampling techniques to the random-*permutation* model, albeit without improving the tightness of the bound.

De *et al.* [18] study the effect of salting for inverting a *permutation* $\mathcal{O}$ as well as for a specific pseudorandom generator based on one-way permutations. Chung *et al.* [14] study the effects of salting in the design of collision-resistant hash functions, and used Unruh's pre-sampling technique to argue that salting defeats preprocessing in this important case. Using salting to obtain non-uniform security was also advocated by Mahmoody and Mohammed [41], who used this technique for obtaining non-uniform black-box separation results.

Finally, the extensive body of work on the *bounded storage model* [42, 4, 22, 53] is related to the special case of AI-ROM, where all T queries in the second stage are done by the challenger to derive the key (so that one tries to minimize T to ensure local computability), but the actual attacker is not allowed any such queries after S-bit preprocessing.

[8] Of course, by performing a direct analysis of the *salted* scheme (e.g., using Theorems 5 or 6), we might get better exact security bounds than by using our general result; namely, shorter salt would be enough to get the claimed amount of security. Still, for settings where obtaining the smallest possible salt value is not critical, the simplicity and generality of our compilers offer a convenient and seamless way to argue security in AI-ROM without doing a direct analsyis.

2 Dealing with Auxiliary Information

Since an attacker with oracle-dependent auxiliary input may obtain the output of arbitrary functions evaluated on a random oracle's function table, it is not obvious how the security of schemes in the *auxiliary-input random-oracle model (AI-ROM)* can be analyzed. To remedy this situation, Unruh [52] introduced the *bit-fixing random-oracle model (BF-ROM)*, in which the oracle is fixed on a subset of the coordinates and uniformly random and independent on the remaining ones, and showed that such an oracle is indistinguishable from an AI-RO.

In Section 2.1, we improve the security bounds proved by Unruh [52] in the following two ways: First, we show that a BF-RO is indistinguishable from an AI-RO up to an additive term of roughly ST/P, where P is the size of the fixed portion of the BF-RO; this improves greatly over Unruh's bound, which was in the order of $\sqrt{ST/P}$. Second, we prove that the probability that any distinguisher outputs 1 in the AI-ROM is at most twice the probability that said distinguisher outputs 1 in the BF-ROM—already when P is roughly equal to ST.

Section 2.2 contains the formalizations of the AI and BF-ROMs, attackers with oracle-dependent advice, and the notion of application. As a consequence of the connections between the two models, the security of *any* application in the BF-ROM translates to the AI-ROM at the cost of the ST/P term, and, additionally, the security of *unpredictability* applications translates at the mere cost of a multiplicative factor of 2 (as long as $P \geq ST$). The corresponding theorems and their proofs can also be found in Section 2.2.

2.1 Replacing Auxiliary Information by Bit-Fixing

In this section, we show that any random oracle about which an attacker may have a certain amount of auxiliary information can be replaced by a suitably chosen convex combination of bit-fixing sources. This substitution comes at the price of either an additive term to the distinguishing advantage or a multiplicative one to the probability that a distinguisher outputs 1. To that end, consider the following definition:

Definition 1. *An (N, M)-source is a random variable X with range $[M]^N$. A source is called*

- $(1-\delta)$-dense *if for every subset $I \subseteq [N]$,*

$$H_\infty(X_I) \;\geq\; (1-\delta) \cdot |I| \cdot \log M \;=\; (1-\delta) \cdot \log M^{|I|}.$$

- $(P, 1-\delta)$-dense *if it is fixed on at most P coordinates and is $(1-\delta)$-dense on the rest,*
- P-bit-fixing *if it is fixed on at most P coordinates and uniform on the rest.*

That is, the min-entropy of every subset of the function table of a δ-dense source is at most a fraction of δ less than what it would be for a uniformly random one.

Lemma 1. *Let X be distributed uniformly over $[M]^N$ and $Z := f(X)$, where $f : [M]^N \to \{0,1\}^S$ is an arbitrary function. For any $\gamma > 0$ and $P \in \mathbb{N}$, there exists a family $\{Y_z\}_{z \in \{0,1\}^S}$ of convex combinations Y_z of P-bit-fixing (N, M)-sources such that for any distinguisher D taking an S-bit input and querying at most $T < P$ coordinates of its oracle,*

$$\left|\mathsf{P}\left[\mathcal{D}^X(f(X)) = 1\right] - \mathsf{P}\left[\mathcal{D}^{Y_{f(X)}}(f(X)) = 1\right]\right| \;\leq\; \frac{(S + \log 1/\gamma) \cdot T}{P} + \gamma$$

and

$$\mathsf{P}\left[\mathcal{D}^X(f(X)) = 1\right] \;\leq\; 2^{(S + 2\log 1/\gamma)T/P} \cdot \mathsf{P}\left[\mathcal{D}^{Y_{f(X)}}(f(X)) = 1\right] + 2\gamma\,.$$

Lemma 1 is proved using a technique (cf. Claim 2) put forth by Göös *et al.* [31] in the area of communication complexity. The technique was also adopted in a paper by Kothari *et al.* [40], who gave a simplified argument for decomposing high-entropy sources into bit-fixing sources with constant density (cf. Definition 1). For self-containment, Section A of the appendix contains a proof of this decomposition technique. Furthermore, the proof of Claim 3 below uses the well-known H-coefficient technique by Patarin [48], while following a recent re-formulation of it due to Hoang and Tessaro [33].

Proof. Fix an arbitrary $z \in \{0,1\}^S$ and let X_z be the distribution of X conditioned on $f(X) = z$. Let $S_z = N \log M - H_\infty(X_z)$ be the min-entropy deficiency of X_z. Let $\gamma > 0$ be arbitrary.

Claim 2. *For every $\delta > 0$, X_z is γ-close to a convex combination of finitely many $(P', 1-\delta)$-dense sources for*

$$P' \;=\; \frac{S_z + \log 1/\gamma}{\delta \cdot \log M}\ .$$

The proof of Claim 2 can be found in Section A of the appendix.

Let X'_z be the convex combination of $(P', 1-\delta)$-dense sources that is γ-close to X_z for a $\delta = \delta_z$ to be determined later. For every $(P', 1-\delta)$ source X' in said convex combination, let Y' be the corresponding P'-bit-fixing source Y', i.e., X' and Y' are fixed on the same coordinates to the same values. The following claim bounds the distinguishing advantage between X' and Y' for any T-query distinguisher.

Claim 3. *For any $(P', 1-\delta)$-dense source X' and its corresponding P'-bit-fixing source Y', it holds that for any (adaptive) distinguisher $\mathcal{D}$ that queries at most T coordinates of its oracle,*

$$\left|\mathsf{P}\big[\mathcal{D}^{X'} = 1\big] - \mathsf{P}\big[\mathcal{D}^{Y'} = 1\big]\right| \;\le\; T\delta \cdot \log M,$$

and

$$\mathsf{P}\big[\mathcal{D}^{X'} = 1\big] \;\le\; M^{T\delta} \cdot \mathsf{P}\big[\mathcal{D}^{Y'} = 1\big].$$

Proof. Assume without loss of generality that $\mathcal{D}$ is deterministic and does not query any of the fixed positions. Let $T_{X'}$ and $T_{Y'}$ be the random variables corresponding to the transcripts containing the query/answer pairs resulting from $\mathcal{D}$'s interaction with X' and Y', respectively. For a fixed transcript τ, denote by $\mathsf{p}_{X'}(\tau)$ and $\mathsf{p}_{Y'}(\tau)$ the probabilities that X' and Y', respectively, produce the answers in τ if the queries in τ are asked. Observe that these probabilities depend only on X' resp. Y' and are independent of $\mathcal{D}$.

Observe that for every transcript τ,

$$\mathsf{p}_{X'}(\tau) \;\le\; M^{-(1-\delta)T} \qquad \text{and} \qquad \mathsf{p}_{Y'}(\tau) \;=\; M^{-T} \tag{1}$$

as X' is $(1-\delta)$-dense and Y' is uniformly distributed.

Since $\mathcal{D}$ is deterministic, $\mathsf{P}[T_{X'} = \tau] \in \{0, \mathsf{p}_{X'}(\tau)\}$, and similarly, $\mathsf{P}[T_{Y'} = \tau] \in \{0, \mathsf{p}_{Y'}(\tau)\}$. Denote by $\mathcal{T}_X$ the set of all transcripts τ for which $\mathsf{P}[T_{X'} = \tau] > 0$. For such τ, $\mathsf{P}[T_{X'} = \tau] = \mathsf{p}_{X'}(\tau)$

and also $\mathsf{P}[T_{Y'} = \tau] = \mathsf{p}_{Y'}(\tau)$. Towards proving the first part of the lemma, observe that

$$\begin{aligned}\Big|\mathsf{P}\big[\mathcal{D}^{X'} = 1\big] - \mathsf{P}\big[\mathcal{D}^{Y'} = 1\big]\Big| \;&\le\; \mathsf{SD}(T_{X'}, T_{Y'}) \\ &=\; \sum_{\tau} \max\{0, \mathsf{P}[T_{X'} = \tau] - \mathsf{P}[T_{Y'} = \tau]\} \\ &=\; \sum_{\tau \in \mathcal{T}_X} \max\{0, \mathsf{p}_{X'}(\tau) - \mathsf{p}_{Y'}(\tau)\} \\ &=\; \sum_{\tau \in \mathcal{T}_X} \mathsf{p}_{X'}(\tau) \cdot \max\left\{0, 1 - \frac{\mathsf{p}_{Y'}(\tau)}{\mathsf{p}_{X'}(\tau)}\right\} \\ &\le\; 1 - M^{-T\delta} \;\le\; T\delta \cdot \log M,\end{aligned}$$

where the first sum is over all possible transcripts and where the last inequality uses $2^{-x} \ge 1 - x$ for $x \ge 0$.

As for the second part of the lemma, observe that due to (1) and the support of $T_{X'}$ being a subset of $T_{Y'}$,

$$\mathsf{P}[T_{X'} = \tau] \;\le\; M^{T\delta} \cdot \mathsf{P}[T_{Y'} = \tau]$$

for any transcript τ. Let $\mathcal{T}_{\mathcal{D}}$ be the set of transcripts where $\mathcal{D}$ outputs 1. Then,

$$\mathsf{P}[\mathcal{D}^{X'} = 1] \;=\; \sum_{\tau \in \mathcal{T}_{\mathcal{D}}} \mathsf{P}[T_{X'} = \tau] \;\le\; M^{T\delta} \cdot \sum_{\tau \in \mathcal{T}_{\mathcal{D}}} \mathsf{P}[T_{Y'} = \tau] \;=\; M^{T\delta} \cdot \mathsf{P}[\mathcal{D}^{Y'} = 1].$$

□

Let Y'_z be obtained by replacing every X' by the corresponding Y' in X'_z. Setting $\delta_z = (S_z + \log 1/\gamma)/(P \log M)$, Claims 2 and 3 imply

$$\Big|\mathsf{P}\big[\mathcal{D}^{X_z}(z) = 1\big] - \mathsf{P}\big[\mathcal{D}^{Y'_z}(z) = 1\big]\Big| \;\le\; \frac{(S_z + \log 1/\gamma) \cdot T}{P} + \gamma \;, \tag{2}$$

as well as

$$\mathsf{P}\big[\mathcal{D}^{X_z}(z) = 1\big] \;\le\; 2^{(S_z + \log 1/\gamma)T/P} \cdot \mathsf{P}\big[\mathcal{D}^{Y'_z}(z) = 1\big] + \gamma \;. \tag{3}$$

Moreover, note that for the above choice of δ_z, $P' = P$, i.e., the sources Y' are fixed on at most P coordinates, as desired.

Claim 4. $\mathbf{E}_z[S_z] \le S$ *and* $\mathsf{P}\big[S_{f(X)} > S + \log 1/\gamma\big] \le \gamma$.

Proof. Observe that $H_\infty(X_z) = H_\infty(X|Z = z) = H(X|Z = z)$ since, conditioned on $Z = z$, X is distributed uniformly over all values x with $f(x) = z$. Therefore,

$$\begin{aligned}\mathbf{E}_z[S_z] \;=\; N \log M - \mathbf{E}_z[H_\infty(X|Z = z)] \;&=\; N \log M - \mathbf{E}_z[H(X|Z = z)] \\ &=\; N \log M - H(X|Z) \;\le\; S \;.\end{aligned}$$

Again due to the uniformity of X, $\mathsf{P}[f(X) = z] = 2^{-S_z}$. Hence,

$$\mathsf{P}\big[S_{f(X)} > S + \log 1/\gamma\big] \;=\; \sum_{z \in \{0,1\}^S \,:\, S_z > S + \log 1/\gamma} \mathsf{P}\big[f(X) = z\big] \;\le\; 2^S \cdot 2^{-(S + \log 1/\gamma)} \le \gamma \;.$$

□

The first part of the lemma now follows (using $Y_z := Y'_z$) by taking expectations over z of (2) and applying the first part of Claim 4. The second part of the lemma is proved as follows:

$$\begin{aligned}
\mathsf{P}\big[\mathcal{D}^X(f(X))=1\big] &\le \mathsf{P}\big[\mathcal{D}^X(f(X))=1, S_{f(X)} \le S+\log 1/\gamma\big] + \mathsf{P}\big[S_{f(X)} > S+\log 1/\gamma\big] \\
&\le \Big(2^{(S+2\log 1/\gamma)T/P} \cdot \mathsf{P}\big[\mathcal{D}^{Y_{f(X)}}(f(X))=1, S_{f(X)} \le S+\log 1/\gamma\big] + \gamma\Big) + \gamma \\
&\le 2^{(S+2\log 1/\gamma)T/P} \cdot \mathsf{P}\big[\mathcal{D}^{Y_{f(X)}}(f(X))=1\big] + 2\gamma \ ,
\end{aligned}$$

where the second inequality follows by taking expectations over z of (3) (together with the condition $S_z \le S + \log 1/\gamma$) and the second part of Claim 4. □

2.2 From the BF-ROM to the AI-ROM

2.2.1 Capturing the Models

Before Lemma 1 from the preceding section can be used to show how security proofs in the BF-ROM can be transferred to the AI-ROM, it is necessary to formally define the two models as well as attackers with oracle-dependent advice and the notion of an application. The high-level idea is to consider two-stage attackers $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and (single-stage) challengers C with access to an oracle $\mathcal{O}$. Oracles have two interfaces pre and main, where pre is accessible only to $\mathcal{A}_1$, which may pass auxiliary information to $\mathcal{A}_2$, and both $\mathcal{A}_2$ and C may access main.

Oracles. An oracle $\mathcal{O}$ has two interfaces $\mathcal{O}.\mathsf{pre}$ and $\mathcal{O}.\mathsf{main}$, where $\mathcal{O}.\mathsf{pre}$ is accessible only once before any calls to $\mathcal{O}.\mathsf{main}$ are made. Oracles used in this work are:

- **Random oracle** $\mathsf{RO}(N, M)$**:** Samples a random function table $F \leftarrow \mathcal{F}_{N,M}$, where $\mathcal{F}_{N,M}$ is the set of all functions from $[N]$ to $[M]$; offers no functionality at $\mathcal{O}.\mathsf{pre}$; answers queries $x \in [N]$ at $\mathcal{O}.\mathsf{main}$ by the corresponding value $F[x] \in [M]$.
- **Auxiliary-input random oracle** $\mathsf{AI\text{-}RO}(N, M)$**:** Samples a random function table $F \leftarrow \mathcal{F}_{N,M}$; outputs F at $\mathcal{O}.\mathsf{pre}$; answers queries $x \in [N]$ at $\mathcal{O}.\mathsf{main}$ by the corresponding value $F[x] \in [M]$.
- **Bit-Fixing random oracle** $\mathsf{BF\text{-}RO}(P, N, M)$**:** Samples a random function table $F \leftarrow \mathcal{F}_{N,M}$; takes a list at $\mathcal{O}.\mathsf{pre}$ of at most P query/answer pairs that override F in the corresponding positions; answers queries $x \in [N]$ at $\mathcal{O}.\mathsf{main}$ by the corresponding value $F[x] \in [M]$.
- **Standard model:** Neither interface offers any functionality.

The parameters N, M are occasionally omitted in contexts where they are of no relevance. Similarly, whenever evident from the context, explicitly specifying which interface is queried is omitted.

Attackers with oracle-dependent advice. Attackers $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ consist of a preprocessing procedure $\mathcal{A}_1$ and a main algorithm $\mathcal{A}_2$, which carries out the actual attack using the output of $\mathcal{A}_1$. Correspondingly, in the presence of an oracle $\mathcal{O}$, $\mathcal{A}_1$ interacts with $\mathcal{O}.\mathsf{pre}$ and $\mathcal{A}_2$ with $\mathcal{O}.\mathsf{main}$.

Definition 2. *An* (S, T)-attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in the $\mathcal{O}$-model *consists of two procedures*

- $\mathcal{A}_1$*, which is computationally unbounded, interacts with* $\mathcal{O}.\mathsf{pre}$*, and outputs an* S*-bit string, and*

- $\mathcal{A}_2$, *which takes an* S*-bit auxiliary input and makes at most* T *queries to* $\mathcal{O}.\mathsf{main}$.

In certain contexts, additional restrictions may be imposed on $\mathcal{A}_2$, captured by some parameters p. $\mathcal{A}$ is referred to as (S, T, p)-attacker in such cases. Examples of such parameters include time and space requirements of $\mathcal{A}_2$ or a limit on the number of queries of a particular type that $\mathcal{A}_2$ makes to a challenger it interacts with. Observe that the parameter S is meaningful also in the standard model, where it measures the length of standard non-uniform advice to the attacker. The parameter T, however, is not relevant as there is no random oracle to query in the attack stage. Consequently, standard-model attackers with resources p are referred to as $(S, *, p)$-attackers.

Applications. Let $\mathcal{O}$ be an arbitrary oracle. An application G in the $\mathcal{O}$-model is defined by specifying a challenger C, which is an oracle algorithm that has access to $\mathcal{O}.\mathsf{main}$, interacts with the main stage $\mathcal{A}_2$ of an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, and outputs a bit at the end of the interaction. The *success* of $\mathcal{A}$ on G in the $\mathcal{O}$-model is defined as

$$\mathrm{Succ}_{G,\mathcal{O}}(\mathcal{A}) \;:=\; \mathsf{P}\big[\mathcal{A}_2^{\mathcal{O}.\mathsf{main}}(\mathcal{A}_1^{\mathcal{O}.\mathsf{pre}}) \leftrightarrow \mathsf{C}^{\mathcal{O}.\mathsf{main}} = 1\big],$$

where $\mathcal{A}_2^{\mathcal{O}.\mathsf{main}}(\mathcal{A}_1^{\mathcal{O}.\mathsf{pre}}) \leftrightarrow \mathsf{C}^{\mathcal{O}.\mathsf{main}}$ denotes the bit output by C after its interaction with the attacker. This work considers two types of applications, captured by the next definition.

Definition 3. *For an* indistinguishability *application* G *in the* $\mathcal{O}$*-model, the* advantage of an attacker $\mathcal{A}$ *is defined as*

$$\mathrm{Adv}_{G,\mathcal{O}}(\mathcal{A}) \;:=\; 2\left|\mathrm{Succ}_{G,\mathcal{O}}(\mathcal{A}) - \frac{1}{2}\right|.$$

For an unpredictability *application* G*, the advantage is defined as*

$$\mathrm{Adv}_{G,\mathcal{O}}(\mathcal{A}) \;:=\; \mathrm{Succ}_{G,\mathcal{O}}(\mathcal{A})\,.$$

An application G *is said to be* $((S, T, p), \varepsilon)$-secure *in the* $\mathcal{O}$*-model if for every* (S, T, p)*-attacker* $\mathcal{A}$*,*

$$\mathrm{Adv}_{G,\mathcal{O}}(\mathcal{A}) \;\le\; \varepsilon\,.$$

Combined query complexity. In order to enlist Lemma 1 for proving Theorems 5 and 6 below, the interaction of some attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with a challenger C in the $\mathcal{O}$-model must be "merged" into a single entity $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$ that interacts with oracle $\mathcal{O}$. That is, $\mathcal{D}_1^{(\cdot)} := \mathcal{A}_1^{(\cdot)}$ and $\mathcal{D}_2^{(\cdot)}(z) := \mathcal{A}_2^{(\cdot)}(z) \leftrightarrow \mathsf{C}^{(\cdot)}$ for $z \in \{0,1\}^S$. $\mathcal{D}$ is called the *combination of* $\mathcal{A}$ *and* C, and the number of queries it makes to its oracle is referred to as *the combined query complexity of* $\mathcal{A}$ *and* C. For all applications in this work there exists an upper bound $T_G^{\mathsf{comb}} = T_G^{\mathsf{comb}}(S, T, p)$ on the combined query complexity of any attacker and the challenger.

2.2.2 Additive Error for Arbitrary Applications

Using the first part of Lemma 1, one proves the following theorem, which states that the security of any application translates from the BF-ROM to the AI-ROM at the cost of an additive term of roughly ST/P, where P is the maximum number of coordinates an attacker $\mathcal{A}_1$ is allowed to fix in the BF-ROM.

Theorem 5. *For any $P \in \mathbb{N}$ and every $\gamma > 0$, if an application G is $((S,T,p),\varepsilon')$-secure in the $\mathsf{BF\text{-}RO}(P)$-model, then it is $((S,T,p),\varepsilon)$-secure in the $\mathsf{AI\text{-}RO}$-model, for*

$$\varepsilon \;\leq\; \varepsilon' + \frac{2(S+\log\gamma^{-1})\cdot T_G^{\mathsf{comb}}}{P} + 2\gamma \;,$$

where T_G^{comb} is the combined query complexity corresponding to G.

Proof. Fix P as well as γ. Set $\mathsf{BF\text{-}RO} := \mathsf{BF\text{-}RO}(P)$ and let G be an arbitrary application and C the corresponding challenger. Moreover, fix an (S,T)-attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, and let $\{Y_z\}_{z\in\{0,1\}^S}$ be the family of distributions guaranteed to exist by Lemma 1, where the function f is defined by $\mathcal{A}_1$. Consider the following (S,T)-attacker $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ (expecting to interact with $\mathsf{BF\text{-}RO}$):

- $\mathcal{A}'_1$ internally simulates $\mathcal{A}_1$ to compute $z \leftarrow \mathcal{A}_1^{\mathsf{AI\text{-}RO.pre}}$. Then, it samples one of the P-bit-fixing sources Y' making up Y_z and presets $\mathsf{BF\text{-}RO}$ to match Y' on the at most P points where Y' is fixed. The output of $\mathcal{A}'_1$ is z.
- $\mathcal{A}'_2$ works exactly as $\mathcal{A}_2$.

Let $\mathcal{D}$ be the combination of $\mathcal{A}_2 = \mathcal{A}'_2$ and C. Hence, $\mathcal{D}$ is a distinguisher taking an S-bit input and making at most T_G^{comb} queries to its oracle. Therefore, by the first part of Lemma 1,

$$\mathrm{Succ}_{G,\mathsf{AI\text{-}RO}}(\mathcal{A}) \;\leq\; \mathrm{Succ}_{G,\mathsf{BF\text{-}RO}}(\mathcal{A}') + \frac{(S+\log\gamma^{-1})\cdot T_G^{\mathsf{comb}}}{P} + \gamma \;.$$

Since there is only an additive term between the two success probabilities, the above inequality implies

$$\mathrm{Adv}_{G,\mathsf{AI\text{-}RO}}(\mathcal{A}) \;\leq\; \mathrm{Adv}_{G,\mathsf{BF\text{-}RO}}(\mathcal{A}') + \frac{2(S+\log\gamma^{-1})\cdot T_G^{\mathsf{comb}}}{P} + 2\gamma$$

for both indistinguishability and unpredictability applications.[9] □

2.2.3 Multiplicative Error for Unpredictability Applications

Using the second part of Lemma 1, one proves the following theorem, which states that the security of any *unpredictability* application translates from the BF-ROM to the AI-ROM at the cost of a multiplicative factor of 2, provided that $\mathcal{A}_1$ is allowed to fix roughly ST coordinates in the BF-ROM.

Theorem 6. *For any $P \in \mathbb{N}$ and every $\gamma > 0$, if an* unpredictability *application G is $((S,T,p),\varepsilon')$-secure in the $\mathsf{BF\text{-}RO}(P,N,M)$-model for*

$$P \;\geq\; (S + 2\log\gamma^{-1})\cdot T_G^{\mathsf{comb}} \;,$$

then it is $((S,T,p),\varepsilon)$-secure in the $\mathsf{AI\text{-}RO}(N,M)$-model for

$$\varepsilon \;\leq\; 2\varepsilon' + 2\gamma \;,$$

where T_G^{comb} is the combined query complexity corresponding to G.

[9] The extra factor of 2 is technically only necessary for indistingusihability applications.

Proof. Using the same attacker $\mathcal{A}'$ as in the proof of Theorem 5 and applying the second part of Lemma 1, one obtains, for any $P \geq (S + 2\log\gamma^{-1}) \cdot T_G^{\mathsf{comb}}$,

$$\begin{aligned} \mathrm{Succ}_{G,\mathsf{AI\text{-}RO}}(\mathcal{A}) &\leq 2^{(S+2\log 1/\gamma)T_G^{\mathsf{comb}}/P} \cdot \mathrm{Succ}_{G,\mathsf{BF\text{-}RO}}(\mathcal{A}') + 2\gamma \\ &\leq 2 \cdot \mathrm{Succ}_{G,\mathsf{BF\text{-}RO}}(\mathcal{A}') + 2\gamma \ , \end{aligned}$$

which translates into

$$\mathrm{Adv}_{G,\mathsf{AI\text{-}RO}}(\mathcal{A}) \leq 2 \cdot \mathrm{Adv}_{G,\mathsf{BF\text{-}RO}}(\mathcal{A}') + 2\gamma$$

for unpredictability applications. □

2.2.4 The Security of Applications in the AI-ROM

The connections between the *auxiliary-input random-oracle model (AI-ROM)* and the *bit-fixing random-oracle model (BF-ROM)* established above suggest the following approach to proving the security of particular applications in the AI-ROM: first, deriving a security bound in the easy-to-analyze BF-ROM, and then, depending on whether one deals with an indistinguishability or an unpredictability application, generically inferring the security of the schemes in the AI-ROM, using Theorems 5 or 6.

The three subsequent sections deal with various applications in the AI-ROM: Section 3 is devoted to security analyses of basic primitives, where "basic" means that the oracle is directly used as the primitive; Section 4 deals with the collision resistance of hash functions built from a random compression function via the Merkle-Damgård construction (MDHFs); and, finally, Section 5 analyzes several cryptographic schemes with computational security.

3 Basic Applications in the AI-ROM

This section treats the AI-ROM security of one-way functions (OWFs), pseudorandom generators (PRGs), normal and weak pseudorandom functions (PRFs and wPRFs), and message-authentication codes (MACs). More specifically, the applications considered are:

- **One-way functions:** For an oracle $\mathcal{O} : [N] \to [M]$, given $y = \mathcal{O}(x)$ for a uniformly random $x \in [N]$, find a preimage x' with $\mathcal{O}(x') = y$.
- **Pseudo-random generators:** For an oracle $\mathcal{O} : [N] \to [M]$ with $M > N$, distinguish $y = \mathcal{O}(x)$ for a uniformly random $x \in [N]$ from a uniformly random element of $[M]$.
- **Pseudo-random functions:** For an oracle $\mathcal{O} : [N] \times [L] \to [M]$, distinguish oracle access to $\mathcal{O}(s, \cdot)$ for a uniformly random $s \in [N]$ from oracle access to a uniformly random function $F : [L] \to [M]$.
- **Weak pseudo-random functions:** Identical to PRFs, but the inputs to the oracle are chosen uniformly at random and independently.
- **Message-authentication codes:** For an oracle $\mathcal{O} : [N] \times [L] \to [M]$, given access to an oracle $\mathcal{O}(s, \cdot)$ for a uniformly random $s \in [N]$, find a pair (x, y) such that $\mathcal{O}(s, x) = y$ for an x on which $\mathcal{O}(s, \cdot)$ was not queried.

	AI-ROM Security	Bound in [19]	Lower Bound
OWFs	$\frac{ST}{N} + \frac{T}{N}$	same	$\min\left\{\frac{ST}{N}, \left(\frac{S^2T}{N^2}\right)^{1/3}\right\} + \frac{T}{N}$
PRGs	$\left(\frac{ST}{N}\right)^{1/2} + \frac{T}{N}$	same	$\left(\frac{S}{N}\right)^{1/2} + \frac{T}{N}$
PRFs	$\left(\frac{S(T+q_{\mathsf{prf}})}{N}\right)^{1/2} + \frac{T}{N}$	same	$\left(\frac{S}{N}\right)^{1/2} + \frac{T}{N}$
wPRFs	$\left(\frac{S(T+q_{\mathsf{prf}})q_{\mathsf{prf}}}{LN}\right)^{1/2} + \frac{T}{N}$	not analyzed	not known
MACs	$\frac{S(T+q_{\mathsf{sig}})}{N} + \frac{T}{N} + \frac{1}{M}$	$\frac{S(T+q_{\mathsf{sig}})}{N} + \frac{T}{N} + \frac{T}{M}$	$\min\left\{\frac{ST}{N}, \left(\frac{S^2T}{N^2}\right)^{1/3}\right\} + \frac{T}{N}$

Table 1: *Asymptotic upper and lower bounds on the security of basic primitives against (S,T)-attackers in the AI-ROM, where q_{prf} and q_{sig} denote PRF and signing queries, respectively, and where (for simplicity) $N = M$ for OWFs. Observe that attacks against OWFs also work against PRGs and PRFs.*

The asymptotic bounds for the applications in question are summarized in Table 1. For OWFs, PRGs, PRFs, and MACs, the resulting bounds match the corresponding bounds derived by Dodis *et al.* [19], who used (considerably) more involved compression arguments; weak PRFs have not previously been analyzed.

The precise statements and the corresponding proofs can be found in the following sections; the proofs all follow the paradigm outlined in Section 2.2.4 of first assessing the security of a particular application in the BF-ROM and then generically inferring the final bound in the AI-ROM using Theorems 5 or 6.

3.1 One-Way Functions

The application $G^{\mathsf{OWF},N,M}$ in the $\mathcal{O}(N,M)$-model is defined via the challenger $\mathsf{C}^{\mathsf{OWF},N,M}$ that picks an $x \in [N]$, passes $y := \mathcal{O}(x)$ to the attacker, and outputs 1 if and only if the attacker returns a value $x' \in [N]$ with $\mathcal{O}(x') = y$.

Theorem 7. *Application $G^{\mathsf{OWF},N,M}$ is $((S,T),\varepsilon)$-secure in the $\mathsf{AI\text{-}RO}(N,M)$-model, where*

$$\varepsilon \;=\; \tilde{O}\left(\frac{ST}{\min(N,M)} + \frac{T}{\min(N,M)}\right).$$

Proof. Observe that for the OWF application, $T^{\mathsf{comb}} = T + 1$. Let $\alpha := \min(N,M)$. It suffices to show that in the $\mathcal{O} := \mathsf{BF\text{-}RO}(P,N,M)$-model, $G := G^{\mathsf{OWF},N,M}$ is $((S,T),\hat{\varepsilon})$-secure for

$$\hat{\varepsilon} \;=\; O\left(\frac{P}{\alpha} + \frac{T}{\alpha}\right).$$

Then, by setting $\gamma := 1/\alpha$ and $P := (S + 2\log\alpha)T^{\mathsf{comb}} = \tilde{O}(ST)$ and applying Theorem 6, the desired conclusion follows.

Towards proving the bound in the BF-RO, suppose $P + T < N/2$ since otherwise the bound of $O((P+T)/N)$ holds trivially. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an (S,T)-attacker. Without loss of generality, assume $\mathcal{A}$ is deterministic, $\mathcal{A}_2$ makes distinct queries to non-prefixed coordinates only, and, at the cost of one additional query, always queries its output. Let $\mathcal{L} = \{(x'_1, y'_1), \ldots, (x'_P, y'_P)\}$ be the points of $\mathcal{O}$ fixed by $\mathcal{A}_1$, and let $\mathcal{Q} = \{(x_1, y_1), \ldots, (x_T, y_T)\}$ the queries $\mathcal{A}_2$ makes, along with the corresponding answers. In slight abuse of notation, let $x \in \mathcal{L}$ stand for $x = x'_j$ for some $j \in [P]$.

Let $\mathcal{E}'$ be the event that the challenge equals one of the prefixed images of $\mathcal{O}$, i.e., $\mathcal{E}' = \{y = y'_j\}$ for some $j \in [P]$. Moreover, for $i \in [T]$, let $\mathcal{E}_i$ be the event that the i^{th} query to $\mathcal{O}$ inverts y, i.e., $y = y_i$. Observe that

$$\mathrm{Succ}_{G,\mathsf{BF\text{-}RO}}(\mathcal{A}) \;\leq\; \mathsf{P}[\mathcal{E}'] + \mathsf{P}[\textstyle\bigcup_i \mathcal{E}_i \mid \overline{\mathcal{E}'}] \;.$$

Note that if $x \notin \mathcal{L}$, y is independent from y'_i for any $i \in [P]$. Hence,

$$\mathsf{P}[\mathcal{E}'] \;\leq\; \mathsf{P}[x \in \mathcal{L}] + \mathsf{P}[\mathcal{E}' \mid x \notin \mathcal{L}] \;\leq\; \frac{P}{N} + \frac{P}{M} \;\leq\; \frac{2P}{\alpha} \;.$$

As per the second probability, observe that

$$\mathsf{P}[\textstyle\bigcup_i \mathcal{E}_i \mid \overline{\mathcal{E}'}] \;\leq\; \displaystyle\sum_i \mathsf{P}[\mathcal{E}_i \mid \overline{\mathcal{E}}_1 \cap \ldots \cap \overline{\mathcal{E}}_{i-1} \cap \overline{\mathcal{E}'}] \;.$$

For any fixed i, conditioned on particular values $y_1 \neq y, \ldots, y_{i-1} \neq y$, the event $\mathcal{E}_i = \{y_i = y\}$ occurs if either $x_i = x$ or if $x_i \neq x$ and $y_i = y$. In the latter case, additionally conditioned on $x_i \neq x$, y_i is independent of y. Hence,

$$\mathsf{P}[\mathcal{E}_i \mid \overline{\mathcal{E}}_1 \cap \ldots \cap \overline{\mathcal{E}}_{i-1} \cap \overline{\mathcal{E}'}] \;\leq\; \frac{1}{N - P - (i-1)} + \frac{1}{M} \;\leq\; \frac{2}{N} + \frac{1}{M} \;\leq\; \frac{3}{\alpha} \;,$$

where the second inequality uses $P + T < N/2$. Thus, overall,

$$\mathrm{Succ}_{G,\mathcal{O}}(\mathcal{A}) \;\leq\; O\left(\frac{P}{\alpha} + \frac{T}{\alpha}\right).$$

□

There exists an attack using rainbow tables [32] that achieves an advantage of

$$\min\left\{\frac{ST}{N}, \left(\frac{S^2T}{N^2}\right)^{1/3}\right\} + \frac{T}{N} \;.$$

3.2 Pseudorandom Generators

The application $G^{\mathsf{PRG},N,M}$ in the $\mathcal{O}(N,M)$-model is defined via the challenger $\mathsf{C}^{\mathsf{PRG},N,M}$ that picks uniformly at random a bit b, a value $x \in [N]$, as well as a value $y_1 \in [M]$, computes $y_0 := \mathcal{O}(x)$, passes y_b to the attacker, and outputs 1 if and only if the attacker returns a bit $b' = b$.

Theorem 8. *Application $G^{\mathsf{PRG},N,M}$ is $((S,T),\varepsilon)$-secure in the $\mathsf{AI\text{-}RO}(N,M)$-model, where*

$$\varepsilon \;=\; \tilde{O}\left(\sqrt{\frac{ST}{N}} + \frac{T}{N}\right).$$

Proof. Observe that for the PRGs, $T^{\mathsf{comb}} = T + 1$. It suffices to show that in the $\mathcal{O} := \mathsf{BF\text{-}RO}(P,N,M)$-model, $G := G^{\mathsf{PRG},N,M}$ is $((S,T),\hat{\varepsilon})$-secure for

$$\hat{\varepsilon} \;=\; O\left(\frac{P}{N} + \frac{T}{N}\right).$$

Then, by setting $\gamma := 1/N$ and $P := \sqrt{STN}$ and applying Theorem 5, the desired conclusion follows.

Towards proving the bound in the BF-RO, suppose $P + T < N/2$ since otherwise the bound of $O((P+T)/N)$ holds trivially. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an (S,T)-attacker. Without loss of generality, assume $\mathcal{A}$ is deterministic, and $\mathcal{A}_2$ makes distinct queries to non-prefixed coordinates only. Let $\mathcal{L} = \{(x'_1, y'_1), \ldots, (x'_P, y'_P)\}$ be the points of $\mathcal{O}$ fixed by $\mathcal{A}_1$, and let $\mathcal{Q} = \{(x_1, y_1), \ldots, (x_T, y_T)\}$ the queries $\mathcal{A}_2$ makes, along with the corresponding answers.

Let $\mathcal{E}'$ be the event that the seed equals one of the prefixed coordinates of $\mathcal{O}$, i.e., $\mathcal{E}' = \{x = x'_j\}$ for some $j \in [P]$. Moreover, for $i \in [T]$, let $\mathcal{E}_i$ be the event that the i^{th} query to $\mathcal{O}$ equals the seed x, i.e., $x = x_i$. Observe that $\mathcal{A}$ only has non-zero advantage if either $\mathcal{E}'$ or, for some $i \in [T]$, $\mathcal{E}_i$ occurs. Clearly,

$$\mathsf{P}[\mathcal{E}'] \;\leq\; \frac{P}{N}\;.$$

Furthermore, using an argument along the lines to that of the proof of Theorem 7,

$$\mathsf{P}[\textstyle\bigcup_i \mathcal{E}_i \mid \overline{\mathcal{E}'}] \;\leq\; \dfrac{T}{N-P-T} \;\leq\; \dfrac{2T}{N}\;,$$

where the last inequality uses $P + T \leq N/2$. Overall,

$$\mathrm{Adv}_{G,\mathcal{O}}(\mathcal{A}) \;\leq\; O\left(\frac{P}{N} + \frac{T}{N}\right).$$

□

The best known attack on PRGs is by De *et al.* [18] and achieves advantage $\Omega\left(\sqrt{S/N}\right)$ for the case $T = 0$.

3.3 Pseudorandom Functions

Application $G^{\mathsf{PRF},N,L,M}$ in the $\mathcal{O}(NL, M)$-model is defined via the following challenger $\mathsf{C}^{\mathsf{PRF},N,L,M}$: It picks uniformly at random a bit b and a key $s \in [N]$. Then, if the attacker queries $x \in [L]$, the challenger answers it by $\mathcal{O}(s, x)$ if $b = 0$ or by $F(x)$ for a function table $F : [L] \to [M]$ chosen uniformly at random.

For attackers $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against PRFs, we make explicit the number q_{prf} of evaluation queries $\mathcal{A}_2$ asks from the challenger.

Theorem 9. *Application $G^{\mathsf{PRF},N,L,M}$ is $((S,T,q_{\mathsf{prf}}),\varepsilon)$-secure in the*$\mathsf{AI\text{-}RO}(NL,M)$*-model, where*

$$\varepsilon \;=\; \tilde{O}\left(\sqrt{\frac{S(T+q_{\mathsf{prf}})}{N}} + \frac{T}{N}\right).$$

Proof. Observe that for the PRFs, $T^{\mathsf{comb}} = T + q_{\mathsf{prf}}$. It suffices to show that in the $\mathcal{O} := \mathsf{BF\text{-}RO}(P, NL, M)$-model, $G := G^{\mathsf{PRF},N,L,M}$ is $((S,T,q_{\mathsf{prf}}),\hat{\varepsilon})$-secure for

$$\hat{\varepsilon} \;=\; O\left(\frac{P}{N} + \frac{T}{N}\right).$$

Then, by setting $\gamma := 1/N$ and $P := \sqrt{S(T+q_{\mathsf{prf}})N}$ and applying Theorem 5, the desired conclusion follows.

The proof proceeds similarly to the case of PRGs. In particular, one observes that $\mathcal{A}$ only has non-zero advantage if one of the prefixed coordinates or one of the queries to $\mathcal{O}$ is of the type $(s, \cdot)$, where s is the key chosen by the challenger. As for PRGs, this probability can easily be bounded by $O\left(P/N + T/N\right)$. □

De *et al.* [18] provide an attack on PRGs. It achieves advantage $\Omega\left(\sqrt{S/N}\right)$ for the case $T = 0$ and can be extended to pseudorandom functions [19].

3.4 Weak Pseudorandom Functions

Application $G^{\mathsf{wPRF},N,L,M}$ in the $\mathcal{O}(NL, M)$-model is defined via the following challenger $\mathsf{C}^{\mathsf{wPRF},N,L,M}$: It picks uniformly at random a bit b and a key $s \in [N]$. Then, whenever the attacker sends a request, the challenger chooses a random $x \in [L]$ and answers the request by $(x, \mathcal{O}(s, x))$ if $b = 0$ or by $(x, F(x))$ for a function table $F : [L] \to [M]$ chosen uniformly at random.

For attackers $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against wPRFs, we make explicit the number q_{prf} of evaluation queries $\mathcal{A}_2$ asks from the challenger.

Theorem 10. *Application* $G^{\mathsf{wPRF},N,L,M}$ *is* $((S, T, q_{\mathsf{prf}}), \varepsilon)$*-secure in the* $\mathsf{AI\text{-}RO}(NL, M)$*-model, where*

$$\varepsilon \;=\; \tilde{O}\left(\sqrt{\frac{S(T + q_{\mathsf{prf}})q_{\mathsf{prf}}}{NL}} + \frac{T}{N}\right).$$

Proof. Observe that for the wPRFs, $T^{\mathsf{comb}} = T + q_{\mathsf{prf}}$. It suffices to show that in the $\mathcal{O} := \mathsf{BF\text{-}RO}(P, NL, M)$-model, $G := G^{\mathsf{wPRF},N,L,M}$ is $((S, T, q_{\mathsf{prf}}), \hat{\varepsilon})$-secure for

$$\hat{\varepsilon} \;=\; O\left(\frac{q_{\mathsf{prf}}P}{NL} + \frac{T}{N}\right).$$

Then, by setting $\gamma := 1/N$ and $P := \sqrt{S(T + q_{\mathsf{prf}})NL/q_{\mathsf{prf}}}$ and applying Theorem 5, the desired conclusion follows.

The proof proceeds similarly to the case of PRFs. In particular, one observes that $\mathcal{A}$ only has non-zero advantage if (1) one of the random queries matches one of the prefixed coordinates or (2) one of the queries to $\mathcal{O}$ is of the type $(s, \cdot)$, where s is the key chosen by the challenger. Similarly to the proof for PRFs, this probability can easily be bounded by $O\left(q_{\mathsf{prf}}P/NL + T/N\right)$. □

3.5 Message-Authentication Codes

Application $G^{\mathsf{MAC},N,L,M}$ in the $\mathcal{O}(NL, M)$-model is defined via the following challenger $\mathsf{C}^{\mathsf{MAC},N,L,M}$: It initially chooses a key $s \in [N]$ uniformly at random and answers attacker queries $x \in L$ by returning $\mathcal{O}(s, x)$. The attacker wins if he submits a pair $(x, y) \in [L] \times [M]$ with $\mathcal{O}(s, x) = y$ for a previously unqueried x.

For attackers $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against MACs, we make explicit the number q_{sig} of signing queries $\mathcal{A}_2$ asks from the challenger.

Theorem 11. *The application* $G^{\mathsf{MAC},N,L,M}$ *is* $((S, T, q_{\mathsf{sig}}), \varepsilon)$*-secure in the* $\mathsf{AI\text{-}RO}(NL, M)$*-model, where*

$$\varepsilon \;=\; \tilde{O}\left(\frac{S(T + q_{\mathsf{sig}})}{N} + \frac{T}{N} + \frac{1}{M}\right).$$

Proof. Observe that for the MAC application, $T^{\mathsf{comb}} = T + q_{\mathsf{sig}}$. It suffices to show that in the $\mathcal{O} := \mathsf{BF\text{-}RO}(P, NL, M)$-model, $G := G^{\mathsf{MAC},N,L,M}$ is $((S, T, q_{\mathsf{sig}}), \hat{\varepsilon})$-secure for

$$\hat{\varepsilon} \;=\; O\left(\frac{P}{N} + \frac{T}{N} + \frac{1}{M}\right).$$

Then, by setting $\gamma := 1/N$ and $P := (S + 2\log N)T^{\mathsf{comb}} = \tilde{O}\left(S(T + q_{\mathsf{sig}})\right)$ and applying Theorem 6, the desired conclusion follows.

Similarly to all previous proofs, the advantage of $\mathcal{A}$ is at most $1/M$ unless one of the prefixed coordinates or queries to $\mathcal{O}$ is of the type $(s, \cdot)$, where s is the key chosen by the challenger. This event is easily upper bounded by $O\left(P/N + T/N\right)$.

□

There exists an inversion attack using rainbow tables [32] that achieves an advantage of

$$\min\left\{\frac{ST}{N}, \left(\frac{S^2T}{N^2}\right)^{1/3}\right\} + \frac{T}{N}\ .$$

4 Collision Resistance in the AI-ROM

A prominent application missing from Section 3 is that of *collision resistance*, i.e., for an oracle $\mathcal{O} : [N] \times [L] \to M$, given a uniformly random salt value $a \in [N]$, finding two distinct $x, x' \in [L]$ such that $\mathcal{O}(a, x) = \mathcal{O}(a, x')$. The reason for this omission is that in the BF-ROM, the best possible bound is easily seen to be in the order of $P/N + T^2/M$. Even applying Theorem 6 for unpredictability applications with $P \approx ST$ results in a final AI-ROM bound of roughly $ST/N + T^2/M$, which is inferior to the optimal bound of $S/N + T^2/M$ proved by Dodis *et al.* [19] using compression.

However, hash functions used in practice, most notably SHA-2, are based on the Merkle-Damgård mode of operation for a compression function $\mathcal{O} : [M] \times [L] \to [M]$, modeled as a random oracle here. Specifically, a B-block message $y = (y_1, \ldots, y_B)$ with $y_j \in [L]$ is hashed to $\mathcal{O}^B(y)$, where

$$\mathcal{O}^1(y_1) = \mathcal{O}(a, y_1) \text{ and } \mathcal{O}^j(y_1, \ldots, y_j) = \mathcal{O}(\mathcal{O}^{j-1}(y_1, \ldots, y_{j-1}), y_j) \text{ for } j > 1.$$

While—as pointed out above—Dodis *et al.* [19] provide a tight bound for the one-block case, it is not obvious at all how their compression-based proof can be extended to deal with even two-block messages. Fortunately, no such difficulties appear when we apply our technique of going through the BF-ROM model, allowing us to derive a bound in Theorem 12 below.

Formally, the collision resistance of Merkle-Damgård hash functions (MDHFs) in the $\mathcal{O}(ML, M)$-model is captured by the application $G^{\mathsf{MDHF},M,L}$, which is defined via the following challenger $\mathsf{C}^{\mathsf{MDHF},M,L}$: It initially chooses a public initialization vector (IV) $a \in [M]$ uniformly at random and sends it to the attacker. The attacker wins if he submits $y = (y_1, \ldots, y_B)$ and $y' = (y'_1, \ldots, y'_{B'})$ such that $y \neq y'$ and $\mathcal{O}^B(y) = \mathcal{O}^{B'}(y')$.

For attackers $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in the following theorem, we make the simplifying assumption that $T > \max(B, B')$. We prove the following bound on the security of MDHFs in the AI-ROM:

Theorem 12. *Application $G^{\mathsf{MDHF},M,L}$ is $((S, T, B), \varepsilon)$-secure in the $\mathsf{AI\text{-}RO}(ML, M)$-model, where*

$$\varepsilon \ = \ \tilde{O}\left(\frac{ST^2}{M} + \frac{T^2}{M}\right).$$

Proof. Observe that for the MDHFs application, $T^{\mathsf{comb}} = T + \max(B, B') = O\left(T\right)$. It suffices to show that in the $\mathcal{O} := \mathsf{BF\text{-}RO}(P, ML, M)$-model, $G := G^{\mathsf{MDHF},M,L}$ is $((S, T), \hat{\varepsilon})$-secure for

$$\hat{\varepsilon} \ = \ O\left(\frac{PT}{M} + \frac{T^2}{M}\right).$$

Then, by setting $\gamma := 1/N$ and $P := (S + 2\log N)T^{\mathsf{comb}} = \tilde{O}(ST)$ and applying Theorem 6, the desired conclusion follows.

Towards proving the bound in the BF-RO, let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an (S,T)-attacker. Without loss of generality, assume $\mathcal{A}$ is deterministic, $\mathcal{A}_2$ makes distinct queries to non-prefixed coordinates only, and, at the cost of one additional query, always queries its output. Let $\mathcal{L} = \{((a'_1, x'_1), y'_1), \ldots, ((a'_P, x'_P), y'_P)\}$ be the points of $\mathcal{O}$ fixed by $\mathcal{A}_1$, and let $\mathcal{Q} = \{((a_1, x_1), y_1), \ldots, ((a_T, x_T), y_T)\}$ the queries $\mathcal{A}_2$ makes, along with the corresponding answers.

Call a salt value $a' \in [M]$ *dirty* if it appears in $\mathcal{L}$. Moreover, call it *reachable* if there exists a chain from the IV a chosen by the challenger, i.e., if there exist $j_1, \ldots, j_d$ such that $a_{j_1} = a$, $\mathcal{O}(a_{j_1}, x_{j_1}) = a_{j_2}, \ldots, \mathcal{O}(a_{j_d}, x_{j_d}) = a'$. Let R_i denote the set of values a' is reachable after the first i queries $\mathcal{A}_2$ makes to $\mathcal{O}$. The set R_i is called *dirty* if some $a' \in R_i$ is dirty and *clean* otherwise. Finally, for every $i \in [T]$, (a_i, x_i) is said to *form a collision* if $a_i \in R_i$ and $\mathcal{O}(a_i, x_i) \in R_i$.

Assume without loss of generality that the queries of $\mathcal{A}_2$ contain the evaluation of $\mathcal{O}^B(y)$ and $\mathcal{O}^{B'}(y')$. The success probability of $\mathcal{A}_2$ is at most

$$\begin{aligned} &\sum_{i=1}^{T} \left(\mathsf{P}[(a_i, x_i) \text{ forms collision} \mid R_i \text{ is clean}] + \mathsf{P}[R_{i+1} \text{ is dirty} \mid R_i \text{ is clean}]\right) \\ \leq\ &\sum_{i=1}^{T} \left(\frac{i}{M} + \frac{P}{M}\right) \ =\ O\left(\frac{PT}{N} + \frac{T^2}{N}\right). \end{aligned}$$

□

Observe that if S and T are taken to be the circuit size, the bound in Theorem 12 becomes vacuous for circuits of size $M^{1/3}$, i.e., it provides security only *well below* the birthday bound and may therefore seem extremely loose. Quite surprisingly, however, it is tight:

Theorem 13. *There exists an (S,T)-attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against application $G := G^{\mathsf{MDHF},M,L}$ in the $\mathcal{O} := \mathsf{AI\text{-}RO}(ML, M)$-model with advantage at least*

$$\mathrm{Adv}_{G,\mathcal{O}}(\mathcal{A}) \ =\ \tilde{\Omega}\left(\frac{ST^2}{M} + \frac{1}{M}\right),$$

assuming $ST^2 \leq M/2$ and $L \geq M$.

The attack is loosely based on rainbow tables [32] and captured by the following (S,T)-attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$:

- $\mathcal{A}_1$: Obtain the function table $F : [M] \times [L] \to [M]$ from $\mathcal{O}$. For $i = 1, \ldots, m := S/(3\lceil \log L \rceil)$, proceed as follows:
  1. Choose $a_{i,0} \in [M]$ uniformly at random.
  2. Compute $a_{i,\ell-1} \leftarrow F^{(\ell-1)}(a_{i,0}, 0)$, where $\ell := \lfloor T/2 \rfloor$.[10]
  3. Find values $x_i \neq x'_i$ such that $a_{i,\ell} := F(a_{i,\ell-1}, x_i) = F(a_{i,\ell-1}, x'_i)$; abort if no such values exist.

  Output the triples $(a_{i,\ell-1}, x_i, x'_i)$ for $i = 1, \ldots, m$.

[10] $F^{(k)}$ stands for the k-fold application of F, and, for the sake of concreteness, let $[L] = \{0, \ldots, L-1\}$.

- $\mathcal{A}_2$: Obtain the public initialization vector a from $\mathsf{C}^{\mathsf{MDHF},M,L}$ and the m triples output by $\mathcal{A}_1$. Proceed as follows:
  1. If $a = a_{i,\ell-1}$ for some i, return (x_i, x_i').
  2. Otherwise, set $\tilde{a} \leftarrow a$ and for $j = 1, \ldots, T$, proceed as follows:
     (a) Query $\tilde{a} \leftarrow \mathcal{O}(\tilde{a}, 0)$.
     (b) If $\tilde{a} = a_{i,\ell-1}$ for some i, return $(0^j \| x_i, 0^j \| x_i')$; otherwise return $(0, 1)$.

Lemma 14. *The advantage of the (S,T)-attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against $G = G^{\mathsf{MDHF},M,L}$ in the $\mathcal{O} = \mathsf{AI\text{-}RO}(ML, M)$-model is*

$$\mathrm{Adv}_{G,\mathcal{O}}(\mathcal{A}) \;\geq\; \frac{ST^2}{50M \log L} + \frac{1}{M} \;,$$

assuming $ST^2 \leq M/2$.

Proof. First, observe that the probability that for one of the values $a_{i,\ell-1}$, there is no collision (x_i, x_i') is at most

$$m \cdot \frac{M!}{M^M} \;\leq\; m \cdot e^{-(M-1)/2} \;\leq\; S \cdot e^{-M/2}$$

if $L = M$ and zero if $L > M$.

Let $A := \{a_{i,j} \mid i = 1, \ldots, m,\ j = 0, \ldots, \ell - 1\}$ be the set of values encountered while building the m chains during preprocessing. Observe that the attack always succeeds if $\tilde{a} \in A$ within the first ℓ queries $\mathcal{A}_2$ makes to $\mathcal{O}$, as in such a case $a_{i,\ell-1}$ (for the appropriate i) can be reached in the remaining ℓ queries.

For $j = 0, \ldots, \ell$, let $\mathcal{E}_j$ be the event that $\tilde{a} \in A$ *for the first time* after the j^{th} query to $\mathcal{O}$. Then, conditioned on a particular set A,

$$\mathsf{P}[\mathcal{E}_j] \;=\; \left(1 - \frac{|A|}{M}\right)^j \cdot \frac{|A|}{M} \;\geq\; \left(1 - \frac{j|A|}{M}\right) \cdot \frac{|A|}{M} \;\geq\; \frac{|A|}{2M} \;,$$

using Bernoulli's inequality as well as the facts that $j \leq T$, $|A| \leq m\ell$, and $m\ell T \leq ST^2 \leq M/2$. Since the events $\mathcal{E}_j$ are disjoint, the probability that $\tilde{a} \in A$ within the first ℓ queries is at least

$$\frac{\ell|A|}{2M} \;\geq\; \frac{T|A|}{4M} \;.$$

Putting the above together, one obtains

$$\mathrm{Adv}_{G,\mathcal{O}}(\mathcal{A}) \;\geq\; \mathbf{E}\left[\frac{T|A|}{4M}\right] - S \cdot e^{-M/2} \;,$$

and, hence, it only remains to compute the expected size of $|A|$. Towards this, following [32], let $\mathcal{E}_{i,j}$ be the event that $a_{i,j}$ is new when discovered during preprocessing. Note that

$$\begin{aligned}
\mathsf{P}[\mathcal{E}_{i,j}] \;&\geq\; \mathsf{P}[\mathcal{E}_{i,0} \cap \ldots \cap \mathcal{E}_{i,j}] \\
&\geq\; \prod_{k=0}^{j} \mathsf{P}[\mathcal{E}_{i,k} \mid \mathcal{E}_{i,0} \cap \ldots \cap \mathcal{E}_{i,k-1}] \\
&\geq\; \left(\frac{M - i\ell}{M}\right)^{j+1} \;\geq\; 1 - \frac{2m\ell^2}{M} \;\geq\; 1 - \frac{ST^2}{M} \;\geq\; \frac{1}{2}
\end{aligned}$$

since at most $i\ell$ values are *not* new when $a_{i,k}$ is chosen. Therefore,

$$\mathbf{E}[|A|] \;\geq\; \sum_{i=1}^{m}\sum_{j=0}^{\ell-1}\mathsf{P}[\mathcal{E}_{i,j}] \;\geq\; \frac{m\ell}{2} \;\geq\; \frac{ST}{12\lceil \log L\rceil}\;.$$

Combining all of the above yields

$$\mathrm{Adv}_{G,\mathcal{O}}(\mathcal{A}) \;\geq\; \frac{ST^2}{49M\log L} - S\cdot e^{-M/2} \;\geq\; \frac{ST^2}{50M\log L}\;.$$

The additional term $1/M$ is the probability that the attack's output $(0,1)$ in case of failure is a collision. □

It should be noted that in practice hash functions use a fixed IV a, and, therefore—in contrast to, e.g., function inversion, where usually the cost of a single preprocessing stage can be amortized over many inversion challenges—the rather sizeable amount of preprocessing required by the attack to just find a collision may not be justified. However, in some cases, the hash function used in a particular application (relying on collision-resistance) is salted by prepending a random salt value to the input. Such salting essentially corresponds to the random-IV setting considered here, and, therefore, the attack becomes relevant again as one might be able to break many instances of the application using a single preprocessing phase.

5 Computationally Secure Applications in the AI-ROM

This section illustrates the bit-fixing methodology on two typical computationally secure applications: (1) Schnorr signatures [50], where Theorem 6 can be applied since forging signatures is an unpredictability application, and (2) trapdoor-function (TDF) key-encapsulation (KEM) [8], where an approach slightly more involved than merely analyzing security in the BF-ROM and applying Theorem 5 is required in order to get a tighter security reduction; see below.

(Please refer to Section B of the appendix for the definitions of digital signatures, KEMs, TDFs, and other standard concepts used in this section.)

Fiat-Shamir with Schnorr. Let G be a cyclic group of prime order $|G| = N$. The Schnorr signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Vfy})$ in the $\mathcal{O}(N^2, N)$-model works as follows:

- **Key generation:** Choose $x \in \mathbb{Z}_N$ uniformly at random, compute $y \leftarrow g^x$, and output $\mathsf{sk} := x$ and $\mathsf{vk} := y$.
- **Signing:** To sign a message $m \in [N]$ with key $\mathsf{sk} = x$, pick $r \in \mathbb{Z}_N$ uniformly at random, compute $a \leftarrow g^r$, query $c \leftarrow \mathcal{O}(a, m)$, set $z \leftarrow r + cx$, and output $\sigma := (a, z)$.
- **Verification:** To verify a signature $\sigma = (a, z)$ for a message m with key $\mathsf{vk} = y$, query $c \leftarrow \mathcal{O}(a, m)$, and check whether $g^z \stackrel{?}{=} ay^c$. If the check succeeds and $c \neq 0$, accept the signature, and reject it otherwise.

For attackers $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in Theorem 15, which assesses the security of Fiat-Shamir with Schnorr in the AI-ROM, we make the running time t and space complexity s of $\mathcal{A}_2$ explicit. Moreover, if $\mathcal{A}$ is an attacker against $G^{\mathsf{DS},\Sigma}$, there is an additional parameter q_{sig} that restricts $\mathcal{A}_2$ to making at most q_{sig} signing queries.

Theorem 15. *Assume $G^{\mathsf{DL},G}$ for a prime $|G| = N$ is $((S', *, t', s'), \varepsilon')$-secure, and let $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Vfy})$ be the Schnorr scheme. Then, for any $T, q_{\mathsf{sig}} \in \mathbb{N}$, $G^{\mathsf{DS},\Sigma}$ is $((S, T, t, s, q_{\mathsf{sig}}), \varepsilon)$-secure in the $\mathsf{AI\text{-}RO}(N^2, N)$-model for*

$$\varepsilon \;=\; \tilde{O}\left(\sqrt{T\varepsilon'} + \frac{Sq_{\mathsf{sig}}(q_{\mathsf{sig}} + T)}{N}\right),$$

any $S \leq S'/\tilde{O}\left(T + q_{\mathsf{sig}}\right)$, $t \leq t' - \tilde{O}\left(S(T + q_{\mathsf{sig}})\right)$, and $s \leq s' - \tilde{O}\left(S(T + q_{\mathsf{sig}})\right)$.

Proof. Let $P \in \mathbb{N}$ be arbitrary, and set $\mathsf{BF\text{-}RO} := \mathsf{BF\text{-}RO}(P, N^2, N)$ and $\mathsf{AI\text{-}RO} := \mathsf{AI\text{-}RO}(N^2, N)$. One first shows that $G^{\mathsf{DS},\Sigma}$ is ε-secure in the $\mathsf{BF\text{-}RO}$-model for

$$\varepsilon \;\leq\; 2 \cdot \max\left(\varepsilon' + E, \frac{T}{N} + \sqrt{T\varepsilon'}\right)$$

for

$$E \;\leq\; \frac{q_{\mathsf{sig}}(q_{\mathsf{sig}} + T + P + 1)}{N}\,.$$

Then, by observing that $T^{\mathsf{comb}}_{G^{\mathsf{DS},\Sigma}} = q_{\mathsf{sig}} + T + 1$ for digital signatures, setting $\gamma := 1/N$ as well as $P := (S + 2\log N)T^{\mathsf{comb}}_{G^{\mathsf{DS},\Sigma}}$, and applying Theorem 6 to the above, one gets a final security bound of

$$\tilde{O}\left(\sqrt{T\varepsilon'} + \frac{Sq_{\mathsf{sig}}(q_{\mathsf{sig}} + T)}{N}\right)$$

Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an $(S, T, t, s, q_{\mathsf{sig}})$-attacker against $G^{\mathsf{DS},\Sigma}$ in the $\mathsf{BF\text{-}RO}$-model. Consider attacker $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ against $G^{\mathsf{DL},G}$:

- $\mathcal{A}'_1$: Run $\mathcal{A}_1$ internally, to get the list $\mathcal{L}$ of coordinates and values $\mathcal{A}_1$ would fix its oracle to as well as the auxiliary information z that $\mathcal{A}_1$ would pass to $\mathcal{A}_2$. For every entry $((a, m), c) \in \mathcal{L}$, compute the discrete logarithm r of a and store $((a, m), c, r)$ in enhanced list $\mathcal{L}'$. Output $(z, \mathcal{L}')$.

- $\mathcal{A}'_2$: First, consider the following algorithm $\mathsf{A}(y, h_1, \ldots, h_T)$, which internally runs $\mathcal{A}_2$:

  1. Run $\mathcal{A}_2(z)$ and answer oracle queries made by $\mathcal{A}_2$ using the values $h_1, \ldots, h_T$. When $\mathcal{A}_2$ asks to see a signature of m, generate a simulated triple (a, c, z) (by choosing z and c uniformly at random and setting $a \leftarrow g^z y^{-c}$) and return $\sigma = (a, z)$. If $\mathcal{A}_2$ has made an oracle query (a, m) or if there exists a tuple $((a, m), \cdot, \cdot) \in \mathcal{L}'$, halt and output $(0, 0)$. Otherwise, answer oracle queries for (a, m) by c.
  2. When $\mathcal{A}_2$ terminates and outputs a valid forgery (m^*, σ^*) for $\sigma^* = (a^*, z^*)$ proceed as follows:
     (a) If $((a^*, m^*), c^*, r^*) \in \mathcal{L}'$, for some c^* and r^*, compute $x \leftarrow (z^* - r^*)/c^*$ and output $(-1, x)$.
     (b) If (a^*, m^*) was the J^{th} oracle query by $\mathcal{A}_2$, output $(J, (\sigma^*, m^*))$.

     If no valid forgery is output, output $(0, 0)$.

  $\mathcal{A}'_2$, on input y and auxiliary input $(z, \mathcal{L}')$, proceeds as follows: It picks uniformly random values $h_1, \ldots, h_T$, and runs $\mathsf{A}(y, h_1, \ldots, h_T; \rho)$, where ρ are the uniformly random coins for use by A. Then:

  – If A outputs $(-1, x)$, $\mathcal{A}'_2$ outputs x.

– If A outputs $(0,0)$, $\mathcal{A}_2'$ aborts.
– If A outputs (J,σ), $\mathcal{A}_2'$ chooses fresh $(h_J',\ldots,h_T')$ and runs $\mathsf{A}(y,h_1,\ldots,h_{J-1},h_J',\ldots,h_T';\rho)$. If the output is (J,σ') and $h_J \neq h_J'$, $\mathcal{A}_2'$ extracts $x := (z-z')/(h_J-h_J')$ from $\sigma=(a,z)$ and $\sigma'=(a,z')$ and outputs it. Otherwise, $\mathcal{A}_2'$ aborts.

Observe that for the above choice of P, $\mathcal{A}'$ is an (S',T',t',s')-attacker.

Consider the first execution of algorithm $\mathsf{A}(y,h_1,\ldots,h_T;\rho)$. The probability that the output has $J \neq 0$ is at least $\varepsilon' - E$, where

$$E \;\leq\; \frac{q_{\mathsf{sig}}(q_{\mathsf{sig}}+T+P+1)}{N}.$$

Consider the following two cases:

- The probability that $J=-1$ is at least $\varepsilon/2 - E$. In that case, $\mathcal{A}_2'$ has success probability $\varepsilon' \geq \varepsilon/2 - E$, or, equivalently, $\varepsilon \leq 2(\varepsilon'+E)$.
- The probability that $J>0$ is at least $\varepsilon/2$. Following the forking lemma in [7, Lemma 1], the probability that $\mathcal{A}_2'$ succeeds is at least

$$\varepsilon' \;\geq\; \frac{\varepsilon}{2}\left(\frac{\varepsilon}{2T}-\frac{1}{N}\right),$$

which implies

$$\varepsilon \;\leq\; 2\cdot\left(\frac{T}{N}+\sqrt{T\varepsilon'}\right).$$

□

For comparison, note that the security of Schnorr signatures in the standard ROM is

$$O\left(\sqrt{T\varepsilon'}+\frac{q_{\mathsf{sig}}(q_{\mathsf{sig}}+T)}{N}\right),$$

i.e., in the AI-ROM the second term worsens by a factor of S.

TDF Key Encapsulation. Let F be a trapdoor family (TDF) generator. TDF encryption is a key-encapsulation mechanism $\Pi = (\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ that works as follows:

- **Key generation:** Run the TDF generator to obtain $(f,f^{-1}) \leftarrow F$, where $f,f^{-1}:[N]\to[N]$. Set the public key $\mathsf{pk} := f$ and the secret key $\mathsf{sk} := f^{-1}$.
- **Encapsulation:** To encapsulate a key with public key $\mathsf{pk}=f$, choose $x \in [N]$, query $k \leftarrow \mathcal{O}(x)$, compute $y \leftarrow f(x)$, and output $(c,k) \leftarrow (y,k)$.
- **Decapsulation:** To decapsulate a ciphertext $c=y$ with secret key $\mathsf{sk}=f^{-1}$, output $k \leftarrow \mathcal{O}(f^{-1}(y))$.

Theorem 16 deals with the security of TDF key encapsulation in the AI-ROM. Once again, for attackers $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$, the running time t and space complexity s of $\mathcal{A}_2$ is made explicit.

Theorem 16. *Let Π be TDF encapsulation. If $G^{\mathsf{TDF},F}$ is $((S',*,t',s'),\varepsilon')$-secure, then, for any $T \in \mathbb{N}$, $G^{\mathsf{KEM\text{-}CPA},\Pi}$ is $((S,T,t,s),\varepsilon)$-secure in the $\mathsf{AI\text{-}RO}(N,N)$-model, where*

$$\varepsilon \;=\; \tilde{O}\left(\varepsilon' + \sqrt{\frac{ST}{N}}\right)$$

and $S = S' - \tilde{O}(ST)$, $t = t' - \tilde{O}(t_{\mathsf{tdf}} \cdot T)$, and $s = s' - \tilde{O}(ST)$, where t_{tdf} is the time required to evaluate the TDF.

Moreover, $G^{\mathsf{KEM\text{-}CCA},\Pi}$ is $((S,T,t,s),\varepsilon)$-secure with the same parameters, except that $t = t' - \tilde{O}(t_{\mathsf{tdf}} \cdot ST)$.

Observe that the above security bound corresponds simply to the sum of the security of the TDF and the security of $\mathcal{O}$ as a PRG (cf. Section 3); in the standard random-oracle model, the security of TDF encryption is simply upper bounded by $O(\varepsilon')$ (cf. Section B.2).

An important point about the proof of Theorem 16 is that it does not follow the usual paradigm of deriving the security of TDF encryption in the BF-ROM and thereafter applying Theorem 5 (for CPA/CCA security is an indistinguishability application). Doing so—as Unruh does for RSA-OAEP [52] (but in an "asymptotic sense," as explained in Footnote 5)—would immediately incur an additive error of $ST/P \leq ST/t'$, since the size of the list P is upper bounded by the TDF attacker size t'. So the naive application Theorem 5 would result in poor exact security.

Instead, our tighter proof of Theorem 16 considers two hybrid experiments (one of which is the original CPA/CCA security game in the AI-ROM). The power of the BF-ROM is used twice—with different list sizes: (1) to argue the indistinguishability of the two experiments and (2) to upper bound the advantage of the attacker in the second hybrid. Crucially, a reduction to TDF security is only required for (1), which has an unpredictability flavor and can therefore get by with a list size of roughly $P \approx ST$; observe that this is polynomial for efficient (S,T)-attackers. The list size for (2) is obtained via the usual balancing between ST/P and the security bound in the BF-ROM.[11]

Proof. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an (S,T,t,s)-attacker against $G^{\mathsf{KEM\text{-}CPA},\Pi}$. Denote by H_0 the experiment in which $\mathcal{A}$ interacts with the challenger $\mathsf{C}^{\mathsf{KEM\text{-}CPA},\Pi}$ (and oracle $\mathsf{AI\text{-}RO}$). Consider the following hybrid H_1: It behaves as H_0, but when $\mathcal{A}_2$ queries x to $\mathcal{O}$, the oracle answers by $\perp$. (The key is still generated as $\mathcal{O}(x)$.) Observe that H_0 and H_1 behave identically unless $\mathcal{A}_2$ queries x; denote this event by $\mathcal{E}$.

Towards bounding $\mathsf{P}[\mathcal{E}]$, consider the following distinguisher $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$ with oracle access to an (N,N)-source:

- $\mathcal{D}_1$ works exactly as $\mathcal{A}_1$.
- $\mathcal{D}_2$, on input z, simulates the interaction between $\mathcal{A}_2(z)$ and $\mathsf{C}^{\mathsf{KEM\text{-}CPA},\Pi}$, answering oracle queries by either of them using its own oracle. If $\mathcal{A}_2$ queries x at some point, $\mathcal{D}_2$ outputs 1; if $\mathcal{A}_2$ terminates without querying x, $\mathcal{D}_2$ outputs 0.

Let X be a uniform (N,N)-source. Observe that

$$\mathsf{P}[\mathcal{E}] \;=\; \mathsf{P}[\mathcal{D}_2^X(\mathcal{D}_1(X)) = 1]$$

and that $\mathcal{D}_2$ makes at most $T+1$ queries to its oracle.

[11] A similar approach also works to improve the security bounds of [52] for RSA-OAEP in the AI-ROM.

Let $P := \lceil (S+2\log N)(T+1) \rceil$. By the second part of Lemma 1, there exists a family $\{Y_z\}_{z\in\{0,1\}^S}$ of convex combinations Y_z of P-bit-fixing (N,N)-sources such that

$$\mathsf{P}\big[\mathcal{D}_2^X(\mathcal{D}_1(X))=1\big] \;\le\; 2\cdot \mathsf{P}\left[\mathcal{D}_1^{Y_{\mathcal{D}_2(X)}}(\mathcal{D}_2(X))=1\right] + 2N^{-1} \;.$$

Consider the following attacker against the TDF security of F:

- $\mathcal{B}_1$ internally runs $z \leftarrow \mathcal{A}_1(X)$ on a uniform (N,N)-source X. Thereafter, it samples a P-bit-fixing source Y' from the convex combination Y_z corresponding to z and outputs $(z,\mathcal{L})$, where $\mathcal{L}$ is the list of the at most P input/output paris at which Y' is fixed.
- $\mathcal{B}_2$, on input $(z,\mathcal{L})$, obtains a pair (f,y) from its challenger and internally runs $\mathcal{A}_2(z)$. It passes the public key $\mathsf{pk} := f$ to $\mathcal{A}_2$ and the challenge ciphertext (y,r) to $\mathcal{A}_2$, where $r \in [N]$ is chosen uniformly at random.

  When $\mathcal{A}_2$ makes an oracle query x, $\mathcal{B}_2$ answers it using lazy sampling but consistent with the list $\mathcal{L}$. Each time, $\mathcal{B}_2$ computes $f(x)$ and if the result equals y, $\mathcal{B}_2$ outputs x to its challenger.

Note that unless x is in the pre-fixed list $\mathcal{L}$, $\mathcal{B}$ perfectly simulates the experiment $\mathcal{D}_1^{Y_{\mathcal{D}_2(X)}}(\mathcal{D}_2(X))$ to $\mathcal{A}$. Hence,

$$\mathsf{P}\left[\mathcal{D}_1^{Y_{\mathcal{D}_2(X)}}(\mathcal{D}_2(X))=1\right] \;\le\; \mathrm{Succ}_{\mathcal{B}}(G^{\mathsf{TDF},F}) + \frac{P}{N} \;\le\; \varepsilon' + \frac{P}{N} \;.$$

Summarizing,

$$\mathsf{P}[\mathcal{E}] \;\le\; 2\left(\varepsilon' + \frac{(S+\log N)(T+2)}{N}\right) \;. \tag{4}$$

It remains to analyze the advantage of $\mathcal{A}$ in the hybrid experiment H_1. To that end, consider the following distinguisher $\mathcal{D} = (\mathcal{D}_1,\mathcal{D}_2)$ with oracle access to an (N,N)-source:

- $\mathcal{D}_1$ works exactly as $\mathcal{A}_1$.
- $\mathcal{D}_2$, on input z, simulates the interaction between $\mathcal{A}_2(z)$ and $\mathsf{C}^{\mathsf{KEM\text{-}CPA},\Pi}$, answering oracle queries by either of them using its own oracle, except that whenever $\mathcal{A}_2$ queries x, $\mathcal{D}_2$ provides $\perp$ as the answer. At the end, $\mathcal{D}_2$ outputs whatever bit the challenger outputs.

Let $P \in \mathbb{N}$ be an arbitrary integer. By the first part of Lemma 1, there exists a family $\{Y_z\}_{z\in\{0,1\}^S}$ of convex combinations Y_z of P'-bit-fixing (N,N)-sources such that

$$\left|\mathsf{P}\big[\mathcal{D}_2^X(\mathcal{D}_1(X))=1\big] - \mathsf{P}\left[\mathcal{D}_1^{Y_{\mathcal{D}_2(X)}}(\mathcal{D}_2(X))=1\right]\right| \;\le\; \frac{(S+\log N)\cdot T}{P} + N^{-1} \;.$$

Consider now the experiment $\mathcal{D}_1^{Y_{\mathcal{D}_2(X)}}(\mathcal{D}_2(X))$. Unless x is among the at most P positions at which $Y_{\mathcal{D}_2(X)}$ is fixed, the view of $\mathcal{A}_2$ is independent of the challenge bit b, and, hence,

$$\mathsf{P}\left[\mathcal{D}_1^{Y_{\mathcal{D}_2(X)}}(\mathcal{D}_2(X))=1\right] \;\le\; \frac{1}{2} + \frac{P}{N} \;.$$

Choosing a list size $P = \Theta\left(\sqrt{STN}\right)$ allows to bound the advantage of $\mathcal{A}$ in H_1 by

$$\tilde{O}\left(\sqrt{\frac{ST}{N}}\right),$$

which dominates the second term in (4). The theorem follows.

The proof for CCA security is similar, except that the second stage $\mathcal{B}_2$ of the attacker against the TDF security of F proceeds as follows:

- $\mathcal{B}_1$ internally runs $z \leftarrow \mathcal{A}_1(X)$ on a uniform (N, N)-source X. Thereafter, it samples a P-bit-fixing source Y' from the convex combination Y_z corresponding to z and outputs $(z, \mathcal{L})$, where $\mathcal{L}$ is the list of the at most P input/output paris at which Y' is fixed.

- $\mathcal{B}_2$, on input $(z, \mathcal{L})$, obtains a pair (f, y) from its challenger. For every pair $(x, \mathcal{O}(x))$ appearing in list $\mathcal{L}$, $\mathcal{B}_2$ computes $y \leftarrow f(x)$ and records the pair $(y, \mathcal{O}(x))$.

  $\mathcal{B}_2$ then internally runs $\mathcal{A}_2(z)$. It passes the public key $\mathsf{pk} := f$ to $\mathcal{A}_2$ and the challenge ciphertext (y, r) to $\mathcal{A}_2$, where $r \in [N]$ is chosen uniformly at random.

  Decryption queries y' by $\mathcal{A}_2$ are answered as follows: If (y', r') has been recorded for some r', the answer is r'; otherwise, $\mathcal{B}_2$ chooses a random value r', records (y', r'), and returns r'.

  When $\mathcal{A}_2$ makes an oracle query x, $\mathcal{B}_2$ answers it using lazy sampling but consistent with the list $\mathcal{L}$. Each time, $\mathcal{B}_2$ computes $f(x)$ and if the result equals y, $\mathcal{B}_2$ outputs x to its challenger.

□

6 Salting Defeats Auxiliary Information

There exist schemes that are secure in the standard ROM but not so in the AI-ROM. A simple example is if the random oracle itself is directly used as a collision-resistant hash function $\mathcal{O} : [N] \to [M]$ for some N and M: in the ROM, $\mathcal{O}$ is easily seen to be collision-resistant, while in the AI-ROM, the first phase $\mathcal{A}_1$ of an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ (cf. Section 2.2) can simply leak a collision to $\mathcal{A}_2$, which then outputs it, thereby breaking the collision-resistance property.

Section C.1 in the appendix briefly highlights two schemes with computational security where the above phenomenon can be observed as well. The first one is a generic transformation of an identification scheme into a signature scheme using the so-called *Fiat-Shamir transform*, and the second one is the well-known *full-domain hash*.[12]

To remedy the situation with schemes such as those mentioned above, in this section we prove that the security of any *standard* ROM scheme can be carried over to the BF-ROM by sacrificing part of the domain of the BF-RO for *salting*. First, in Section 6.1, we analyze the standard way of salting a random oracle by prefixing a randomly chosen (public) value to every oracle query. Second, in Section 6.2, we also show how to adapt a technique by Maurer [42], originally used in the context of key-agreement with randomizers, to obtain a more domain-efficient salting technique, albeit with a longer salt value; the salt length can be reduced by standard derandomization techniques based on random walks on expander graphs.

The salting method has the advantage of being applicable to every possible application that can be proven secure in the *standard* ROM. Moreover, for most of the applications presented in the preceding sections, salting with values from a sufficiently large space allows to recover the bounds proved by analyzing the security of the (unsalted) applications in the ROM directly. Thus, in this sense, *salting provably defeats preprocessing*. However, this comes at the price of assuming a much larger domain of the random oracle (so that S is now a much tinier fraction of the random oracle domain). Moreover, in many concrete cases we observe that by analyzing the *salted* scheme in the BF-ROM directly and using Theorems 5 or 6, we might get considerably better security bounds for

[12] By virtue of Theorem 6, the existence of attacks in the AI-ROM against the above schemes obviously implies that these schemes cannot be secure in the BF-ROM either. It is also relatively straight-forward to devise direct attacks in the BF-ROM.

salted applications than by using our general theorems.[13] Nevertheless, for settings where obtaining the smallest possible salt value is not critical, the simplicity and generality of our compilers offer a convenient and seamless way to argue security in AI-ROM.

6.1 Standard Salting

The standard way of salting a scheme is to simply prepend a public salt value to every oracle query: Consider an arbitrary application G with the corresponding challenger C. Let $\mathsf{C}_{\mathsf{salt}}$ be the challenger that is identical to C except that it initially chooses a uniformly random value $a \in [K]$, outputs a to $\mathcal{A}_2$, and prepends a to every oracle query. Denote the corresponding application by G_{salt}. Observe that the salt value a is chosen after the first stage $\mathcal{A}_1$ of the attack, and, hence, as long as the first stage $\mathcal{A}_1$ of the attacker in the BF-ROM does not prefix a position starting with a, it is as if the scheme were executed in the standard ROM. Moreover, note that the time and space complexities s and t, respectively, of $\mathcal{A}_2$ increase roughly by P due to the security reduction used in the proof.

Theorem 17. *For any $P \in \mathbb{N}$, if an application G is $((S', T', t', s'), \varepsilon')$-secure in the $\mathsf{RO}(N, M)$-model, then G_{salt} is $((S, T, t, s), \varepsilon)$-secure in the $\mathsf{BF\text{-}RO}(P, NK, M)$-model for*

$$\varepsilon \;=\; \varepsilon' + \frac{P}{K}\;,$$

$S = S' - \tilde{O}(P)$, $T = T'$, $t = t' - \tilde{O}(P)$, and $s = s' - \tilde{O}(P)$.

Proof. Fix N, M, K, as well as P. Set $\mathsf{BF\text{-}RO} := \mathsf{BF\text{-}RO}(P, N, M)$ and $\mathsf{RO} := \mathsf{RO}(N, M)$, and let G be an arbitrary application and C be the corresponding challenger.

Fix an (S, T, t, s)-attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against G_{salt}, and consider the following (S', T', t', s')-attacker $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ against G (expecting to interact with RO):

- $\mathcal{A}'_1$ internally runs $\mathcal{A}_1$ to obtain the list $\mathcal{L}$ of preset values and auxiliary information z and outputs $(z, \mathcal{L})$.
- $\mathcal{A}'_2$, on input $(z, \mathcal{L})$, chooses a uniformly random value $a \in [K]$. If there exists a query $((a, x), y) \in \mathcal{L}$ for some $x \in [N]$ and $y \in [M]$, $\mathcal{A}'_2$ halts. Otherwise, it internally simulates $\mathcal{A}_2$ on z as follows: $\mathcal{A}'_2$ forwards all messages between $\mathcal{A}_2$ and the challenger; oracle queries (a', x) by $\mathcal{A}_2$ are answered as follows: if $a' = a$, query x is asked to RO and the answer is passed to $\mathcal{A}_2$; else if $(a', x, y) \in \mathcal{L}$ for some $y \in [M]$, y is passed to $\mathcal{A}_2$; otherwise, the query is answered from a function table chosen uniformly at random.

It is easily seen that the view of $\mathcal{A}_2$ is the same in both the experiment where it interacts directly as well as via $\mathsf{C}_{\mathsf{salt}}$ with $\mathsf{BF\text{-}RO}$ and the experiment where it interacts via $\mathcal{A}'_2$ as well as via C with RO, unless the salt chosen by $\mathcal{A}'_2$ happens to fall into the preset list $\mathcal{L}$, which happens with probability at most P/K. □

Combining Theorem 17 with Theorems 5 and 6 from Section 2.2 yields the following corollaries:

Corollary 18. *For any $P \in \mathbb{N}$ and every $\gamma > 0$, if an arbitrary application G is $((S', T', t', s'), \varepsilon')$-secure in the $\mathsf{RO}(N, M)$-model, then G_{salt} is $((S, T, t, s), \varepsilon)$-secure in the $\mathsf{AI\text{-}RO}(NK, M)$-model for*

$$\varepsilon \;=\; \varepsilon' + \frac{P}{K} + \frac{2(S + \log \gamma^{-1}) \cdot T^{\mathsf{comb}}_{G_{\mathsf{salt}}}}{P} + 2\gamma$$

[13] This, of course, is not surprising, since our general analysis anyway goes through the BF-ROM model, so one would expect that direct analysis might be even better.

and any $S = S' - \tilde{O}(P)$, $T = T'$, $t = t' - \tilde{O}(P)$, *and* $s = s' - \tilde{O}(P)$, *where* $T^{\mathsf{comb}}_{G_{\mathsf{salt}}}$ *is the combined query complexity corresponding to* G_{salt}.

Corollary 19. *For every* $\gamma > 0$, *if an unpredictability application* G *is* $((S', T', t', s'), \varepsilon')$*-secure in the* $\mathsf{RO}(N, M)$*-model, then* G_{salt} *is* $((S, T, t, s), \varepsilon)$*-secure in the* $\mathsf{AI\text{-}RO}(NK, M)$*-model for*

$$\varepsilon \;=\; 2\varepsilon + \frac{2(S + 2\log\gamma^{-1}) \cdot T^{\mathsf{comb}}_{G_{\mathsf{salt}}}}{K} + 2\gamma$$

and any $S = S'/\tilde{O}(T^{\mathsf{comb}}_{G_{\mathsf{salt}}})$, $T = T'$, $t' = t - \tilde{O}(P)$, *and* $s' = s - \tilde{O}(P)$, *where* $P = (S + 2\log\gamma^{-1})T^{\mathsf{comb}}_{G_{\mathsf{salt}}}$ *and where* $T^{\mathsf{comb}}_{G_{\mathsf{salt}}}$ *is the combined query complexity corresponding to* G_{salt}.

The following paragraphs briefly discuss (in asymptotic terms and omitting logarithmic factors) how salting affects the security of the applications presented in the preceding sections. We also provide examples to illustrate that directly analyzing a salted scheme in the BF-ROM can lead to much better bounds than combining a standard-ROM security bound with one of the above corollaries.

- For most of the basic applications from Section 3, large enough salt would yield AI-ROM bounds comparable to those derived by the direct AI-ROM analysis of the corresponding *unsalted* applications, and using even larger salt allows to match the much better bounds derived in the standard ROM, justifying our claim that salt defeats preprocessing. As an example, consider the one-way function application. It is easily seen that in the standard ROM, the security of the application is $T/\min(N, M)$. Combined with Corollary 19, one obtains a final security bound of

  $$O\left(\frac{ST}{K} + \frac{T}{\min(N, M)}\right).$$

  Setting $K := \min(N, M)$, we recover the AI-ROM bound in Theorem 7, while setting $K = S\min(N, M)$, we even get the same security as in the ROM (of course, on a much larger domain). However, by inspecting the proof of Theorem 7, one can easily see that for the *salted* OWF application (i.e., given a randomly chosen a, finding a preimage under $\mathcal{O}(a, \cdot)$ of $y = \mathcal{O}(a, x)$ for a randomly chosen x), one can easily prove a considerably better bound of

  $$O\left(\frac{ST}{KN} + \frac{T}{\min(N, M)}\right)$$

  via a direct analysis in the BF-ROM and then applying Theorem 6.[14] Hence, it now suffices to set $K = S$ in order to get the same bound as in the traditional ROM. Similar phenomena can be observed for PRGs, PRFs, wPRFs, and MACs.

  For Merkle-Damgard hash functions (MDHFs), using salting, one gets a final bound in the order of $\frac{T^2}{M} + \frac{ST}{K}$. Setting $K = M$ to match the salt length already used in this application, we get a final bound on the order of

  $$\frac{T^2}{M} + \frac{ST}{M}\,.$$

  This seems to improve the bound $T^2/M + ST^2/M$ resulting from the direct analysis in the BF-ROM (see Theorem 12). Note, however, that with this value of K we now use a random oracle from M^3 to M instead of M^2 to M, while still using B evaluations of this more

[14] Indeed, this bound for salted OWFs was already obtained Dodis *et al.* [19] using the compression technique.

compressing oracle to process a B-block message (as the salt is now appended to each evaluation, instead of used as initialization vector at the beginning). In particular, for such a more compressing oracle from M^3 to M, the traditional MDHF chaining would only use $B/2$ evaluations, increasing the speed by a factor of 2. Thus, it is not immediately clear if tighter (provable) exact security is worth this efficiency slowdown.

- For the computationally secure applications from Section 5, the situation is as follows:

  (1) For Schnorr signatures, combining the standard ROM bound (cf. Theorem 27 in Section B.1 of the appendix) with Corollary 19 and using $K := N$ yields a final security bound of

  $$\sqrt{T\varepsilon'} + \frac{(q_{\mathsf{sig}} + S)(q_{\mathsf{sig}} + T)}{N} ,$$

  which actually improves over the bound obtained by the direct analysis (cf. Theorem 15), again, however, at the cost of requiring a larger random-oracle domain. As with the basic applications, using even larger salt $K := SN$, one can recover the standard-ROM bound. By analyzing salted Schnorr signatures in the BF-ROM directly, one can prove a bound of

  $$\frac{T + q_{\mathsf{sig}}^2}{N} + \sqrt{T\varepsilon'} + \frac{q_{\mathsf{sig}} S(q_{\mathsf{sig}} + T)}{KN} ,$$

  and, therefore, setting $K := S$ is sufficient to match the bound in the standard ROM.

  (2) For TDF encryption, the indirect approach using the standard ROM bound (cf. Theorem 28 in Section B.2 of the appendix), Corollary 18, and, once again, $K := N$ results in the same bounds as the direct analysis. Choosing the salt value from a larger domain or analyzing the salted application in the BF-ROM directly will not yield further improvements, as the error term ST/P dominates the security bound.

- The applications mentioned in Section C.1, i.e., applications insecure in the AI-ROM, can be endowed with salt to obtain security bounds:

  (1) For collision-resistant hashing, by combining the standard-ROM bound of T^2/M with Corollary 19 and setting $K := M$, one obtains a final security bound of

  $$\frac{T^2}{M} + \frac{ST}{M}$$

  in the AI-ROM. Increasing the size of the salt space to $K := SM$ results in the standard ROM bound. A direct analysis will not yield improved bounds since $\mathcal{A}_1$ can always leak collisions for $\Omega(P)$ salt values, and, hence the above bound is optimal.

  (2) Combining the standard-ROM bound for signature schemes based on ID schemes [1] of (roughly)

  $$T\varepsilon' + \frac{q_{\mathsf{sig}}(q_{\mathsf{sig}} + T)}{N} ,$$

  where ε' refers to the security of the underlying ID scheme, with Corollary 19, one obtains final security

  $$T\varepsilon' + \frac{(q_{\mathsf{sig}} + S)(q_{\mathsf{sig}} + T)}{N} ,$$

  using salt space $[N]$. As with Schnorr signatures, by setting $K := SN$, one recovers the standard-ROM bound. *Unlike* Schnorr signatures, however, for the general transform there

is no security improvement to be achieved by directly analyzing the salted application in the BF-ROM since $\mathcal{A}_1$ can launch the attack described in Section C.1 for $\Omega(P)$ salt values.

(3) Combining the standard-ROM bound for full-domain hash signatures [16] of (roughly)

$$q_{\mathsf{sig}} \cdot \varepsilon',$$

where ε' refers to the TDF-security of RSA (cf. Section B.2 of the appendix), with Corollary 19, one obtains final security

$$q_{\mathsf{sig}} \cdot \varepsilon' + \frac{S(q_{\mathsf{sig}} + T)}{N}$$

using salt space $[N]$. For similar reasons as above, a direct analysis of the salted scheme in the BF-ROM will not yield improved security bounds.

6.2 Improved Salting

One way to think of salting is to view the function table of $\mathsf{BF\text{-}RO}(KN, M)$ as a $(K \times N)$-matrix and let the challenger in the salted application randomly pick and announce the row to be used for oracle queries. However, K has to be around the same size as N to obtain meaningful bounds. In this section, based on a technique by Maurer [42], we provide a more domain-efficient means of salting, where the security will decay exponentially (as opposed to inverse linearly) with the domain expansion factor K, at the cost that each evaluation of the derived random oracle will cost K evaluations (as opposed to 1 evaluation) of the original random oracle.

Consider an arbitrary application G with corresponding challenger C. Let $\mathsf{C}_{\mathsf{salt}'}$ be the challenger works as follows: It initially chooses a uniformly random value $a = (a_1, \ldots, a_K) \in [N]^K$ and outputs a to $\mathcal{A}_2$. Then, it internally runs C, forwards all messages between the attacker and C, but answers the queries $x \in [N]$ that C makes to the oracle by

$$\sum_{i=1}^{K} \mathsf{BF\text{-}RO.main}(i, x + a_i) \ ,$$

where addition is in $\mathbb{Z}_N$ and $\mathbb{Z}_M$, respectively. In other words, the function table of $\mathsf{BF\text{-}RO}$ is arranged as a $K \times N$ matrix, the i^{th} row is shifted by a_i, and queries x are answered by computing the sum modulo M of all the values in the x^{th} column of the shifted matrix, denoted F_a. Denote the corresponding application by $G_{\mathsf{salt}'}$.

Theorem 20. *For any $P \in \mathbb{N}$, if an application G is $((S', T', t', s'), \varepsilon')$-secure in the $\mathsf{RO}(N, M)$-model, then $G_{\mathsf{salt}'}$ is $((S, T, t, s), \varepsilon)$-secure in the $\mathsf{BF\text{-}RO}(P, NK, M)$-model for*

$$\varepsilon' \ = \ \varepsilon + N \cdot \left(\frac{P}{KN}\right)^K \ ,$$

$S = S' - \tilde{O}(P)$, $T = T'$, $t = t' - \tilde{O}(P)$, and $s = s' - \tilde{O}(P)$.

In particular, assuming $P \leq KN/2$, setting $K = O(\log N)$ will result in additive error $N(P/NK)^K = o(\frac{1}{N})$ and domain size $O(N \log N)$. But if $P \leq N^{1-\Omega(1)}$, setting $K = O(1)$ will result in the same additive error $o(\frac{1}{N})$ in the original domain of near-optimal size $O(N)$. Hence, for most practical purposes, the efficiency slowdown K (in both the domain size and the complexity of oracle evaluation) is at most $O(\log N)$ and possibly constant.

Proof. Fix N, M, K, as well as P. Set $\mathsf{BF\text{-}RO} := \mathsf{BF\text{-}RO}(P, NK, M)$ and $\mathsf{RO} := (N, M)$, and let G be an arbitrary application and C be the corresponding challenger.

Fix an (S, T, t, s)-attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against $G_{\mathsf{salt}'}$. Consider the following (S, T, t, s)-attacker $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ against G (expecting to interact with RO):

- $\mathcal{A}'_1$ internally runs $\mathcal{A}_1$ to obtain the list $\mathcal{L}$ of preset values and auxiliary information z, and outputs $(z, \mathcal{L})$.
- $\mathcal{A}'_2$, on input $(z, \mathcal{L})$, chooses a uniformly random value $a \in [N]^K$. If $(1, x + a_1), (2, x + a_2), \ldots, (K, x + a_K)$ are all prefixed coordinates in $\mathcal{L}$ for some x, then $\mathcal{A}'_2$ halts. Otherwise, $\mathcal{A}'_2$ simulates $\mathcal{A}_2$ on z as follows: $\mathcal{A}'_2$ forwards all messages between $\mathcal{A}_2$ and the challenger. Moreover, it maintains a partial shifted function table F_a, that initially contains the points in $\mathcal{L}$.

  Whenever $\mathcal{A}_2$ makes a query $(i, x - a_i)$, $\mathcal{A}'_2$ proceeds as follows:

  - If i is the only row j for which coordinate (j, x) is undefined in F_a, $\mathcal{A}'_2$ queries $y \leftarrow \mathsf{RO.main}(x)$ and sets $F_a[(i, x)] \leftarrow y - \sum_{j \neq i} F_a[(j, x)]$.
  - Otherwise, $F_a[(i, x)]$ is set to a uniformly random value.

  In either case, $\mathcal{A}'_2$ answers the query by $F_a[(i, x)]$.

The view of $\mathcal{A}_2$ is the same in both the experiment where it interacts directly as well as via $\mathsf{C}_{\mathsf{salt}'}$ with $\mathsf{BF\text{-}RO}$ and the experiment where it interacts via $\mathcal{A}'_2$ as well as via C with RO, unless the salt vector chosen by $\mathcal{A}_2$ happens to lead to a column whose entries are all covered in $\mathcal{L}$. Using a simple union bound over all columns, this happens with probability at most

$$N\left(\frac{P_1 \cdots P_K}{N^K}\right) \;\leq\; N\left(\frac{P}{KN}\right)^K,$$

where the inequality follows from the relationship between the geometric and arithmetic means, and P_i is the number of prefixed positions in the i^{th} row of $\mathsf{BF\text{-}RO}$'s function table, and, in particular, $\sum_{i=1}^{K} P_i = P$. □

Combining the above results with those in Section 2.2 yields the following corollaries:

Corollary 21. *For any $P \in \mathbb{N}$ and every $\gamma > 0$, if an application G is $((S', T', t', s'), \varepsilon')$-secure in the $\mathsf{RO}(N, M)$-model, then $G_{\mathsf{salt}'}$ is $((S, T, t, s), \varepsilon)$-secure in the $\mathsf{AI\text{-}RO}(NK, M)$-model for*

$$\varepsilon \;=\; \varepsilon' + N \cdot \left(\frac{P}{KN}\right)^K + 2\frac{(S + \log \gamma^{-1}) \cdot T^{\mathsf{comb}}_{G_{\mathsf{salt}'}}}{P} + 2\gamma$$

and any $S = S' - \tilde{O}(P)$, $T = T'$, $t = t' - \tilde{O}(P)$, and $s = s' - \tilde{O}(P)$, where $T^{\mathsf{comb}}_{G_{\mathsf{salt}'}}$ is the combined query complexity corresponding to $G_{\mathsf{salt}'}$.

Corollary 22. *For every $\gamma > 0$, if an application G is $((S', T', t', s'), \varepsilon')$-secure in the $\mathsf{RO}(N, M)$-model, then $G_{\mathsf{salt}'}$ is $((S, T, t, s), \varepsilon)$-secure in the $\mathsf{AI\text{-}RO}(NK, M)$-model for*

$$\varepsilon \;=\; 2\varepsilon + 2N \cdot \left(\frac{(S + 2\log \gamma^{-1}) T^{\mathsf{comb}}_{G_{\mathsf{salt}'}}}{KN}\right)^K + 2\gamma$$

and any $S = S'/\tilde{O}(T^{\mathsf{comb}}_{G_{\mathsf{salt}'}})$, $T = T'$, $t' = t - \tilde{O}(P)$, and $s' = s - \tilde{O}(P)$, where $P = (S + 2\log \gamma^{-1}) T^{\mathsf{comb}}_{G_{\mathsf{salt}'}}$ and where $T^{\mathsf{comb}}_{G_{\mathsf{salt}'}}$ is the combined query complexity corresponding to $G_{\mathsf{salt}'}$.

Applications. Concerning the applications from Sections 3 to 5, using the improved corollaries above, one can prove bounds similar to those obtained at the end of Section 6.1. The advantage of the domain-efficient salting is that one can get by with a smaller domain. However, the number of queries any application makes increases by a factor of K. As an example, consider MDHFs: The domain requirements are now lower as with standard salting, while the number of evaluations of the underlying random oracles increases significantly, which is undesirable in practical applications of the Merkle-Damgård construction.

More randomness-efficient salting. Note that $\mathsf{C}_{\mathsf{salt}'}$ requires $O(K \log N)$ random bits to generate a. Although we never envision the value to K to be super-logarithmic, when $K = \log N$, it would take $O(\log^2 N)$ random bits to generate a. We observe that the number of required random bits can be reduced to $\log N + O(K)$ via standard derandomization techniques, which brings the total randomness complexity to $O(\log N)$ even when $K = \log N$. Specifically, instead of drawing $a_1, \ldots, a_K$ uniformly and randomly, we generate them via a random walk over an expander graph. We provide the details below.

Let A be the adjacency matrix of a d-regular graph over $[N]$ vertices where d is a constant. Let $d = \lambda_1 \geq \cdots \geq \lambda_N$ be the eigenvalues of A and suppose $|\lambda_i| \leq d \cdot c$ for $2 \leq i \leq N$, where $c < 1$ is a constant. Given A, the modified $\mathsf{C}_{\mathsf{salt}'}$ is the same as before except instead of choosing a uniformly over $[N]^K$, we pick a_1 uniformly over $[N]$ then perform a $K-1$-step random walk over A and let a_i be the i-th node on this walk.

Theorem 23. *For any $P \in \mathbb{N}$, if an application G is $((S', T', t', s'), \varepsilon')$-secure in the $\mathsf{RO}(N, M)$-model, then $G'_{\mathsf{salt}'}$ is $((S, T, t, s), \varepsilon)$-secure in the $\mathsf{BF\text{-}RO}(P, NK, M)$-model for*

$$\varepsilon \;=\; \varepsilon' + N \cdot \left(\sqrt{\frac{P}{KN}} + c \right)^K ,$$

$S = S' - \tilde{O}(P)$, $T = T'$, $t = t' - \tilde{O}(P)$, and $s = s' - \tilde{O}(P)$.

For $P \leq KN/2$, $K = O(\log N)$ and sufficiently small c, the additive error term $N \cdot (\sqrt{P/(KN)} + c)^K = o(\frac{1}{N})$, and the public seed required by the challenger is now only $O(\log N)$.

Proof. The proof is similar as Theorem 20. In particular, the view of $\mathcal{A}_2$ is the same in both the experiment where it interacts directly with $\mathsf{BF\text{-}RO}$ and the experiment where it interacts via $\mathcal{A}'_2$ with RO, unless the salt chosen by $\mathcal{A}_2$ happens to lead to a column whose entries are all covered in $\mathcal{L}$. For an arbitrary fixed column, let BAD_i be the set of shifts such that its i-th entry will be covered by $\mathcal{L}$ under those shifts. Note that $\sum_{i=1}^{K} |\mathrm{BAD}_i| \leq P$. We claim:

Claim 24. $\mathsf{P}[\forall i \in [K], a_i \in \mathrm{BAD}_i] \leq (\sqrt{P/(KN)} + c)^K$.

In other words, for any fixed column, the probability that all of its entries are all covered in $\mathcal{L}$ is at most $(\sqrt{P/(KN)} + c)^K$. By a union bound, we can conclude $\mathcal{A}_2$ halts with probability at most $N(\sqrt{P/(KN)} + c)^K$ and obtain the desired conclusion. The proof of Claim 24 is standard and included in Appendix C.2 for completeness. □

Note that by combining Theorem 23 with Theorems 5 or 6, one can obtain corollaries similar to Corollaries 21 and 22, but with improved randomness complexity for generating salt.

Acknowledgments

The authors thank Mika Göös for pointing out the decomposition lemma for high-entropy sources in [31], Andrej Bogdanov for discussions about derandomization using random walks, Daniel Wichs for suggestions on proving the security of computationally secure schemes in the AI-ROM, and Patrick Harasser for pointing out bugs in earlier versions of the proofs in Section 2.

Sandro Coretti is supported by NSF grants 1314568 and 1319051. Yevgeniy Dodis is partially supported by gifts from VMware Labs and Google, and NSF grants 1619158, 1319051, 1314568. Siyao Guo is supported by NSF grants CNS1314722 and CNS-1413964; this work was done partially while the author was visiting the Simons Institute for the Theory of Computing at UC Berkeley.

References

[1] Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, pages 418–433, 2002.

[2] Leonard M. Adleman. Two theorems on random polynomial time. In *19th Annual Symposium on Foundations of Computer Science, Ann Arbor, Michigan, USA, 16-18 October 1978*, pages 75–83, 1978.

[3] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple construction of almost k-wise independent random variables. *Random Struct. Algorithms*, 3(3):289–304, 1992.

[4] Yonatan Aumann, Yan Zong Ding, and Michael O. Rabin. Everlasting security in the bounded storage model. *IEEE Trans. Information Theory*, 48(6):1668–1680, 2002.

[5] Mihir Bellare, Alexandra Boldyreva, and Adriana Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 171–188, 2004.

[6] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 514–523, 1996.

[7] Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pages 390–399, 2006.

[8] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993.*, pages 62–73, 1993.

[9] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In *Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, pages 92–111, 1994.

[10] Daniel J. Bernstein and Tanja Lange. Non-uniform cracks in the concrete: The power of free precomputation. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II*, pages 321–340, 2013.

[11] Ran Canetti, Oded Goldreich, and Shai Halevi. On the random-oracle methodology as applied to length-restricted signature schemes. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, pages 40–57, 2004.

[12] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.

[13] Benny Chor, Oded Goldreich, Johan Håstad, Joel Friedman, Steven Rudich, and Roman Smolensky. The bit extraction problem of t-resilient functions (preliminary version). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 396–407, 1985.

[14] Kai-Min Chung, Huijia Lin, Mohammad Mahmoody, and Rafael Pass. On the power of nonuniformity in proofs of security. In *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 389–400, 2013.

[15] Sandro Coretti, Yevgeniy Dodis, Siyao Guo, and John Steinberger. Random oracles and nonuniformity (full version of this paper). Cryptology ePrint Archive, Report 2017/937, 2017. `https://eprint.iacr.org/2017/937`.

[16] Jean-Sébastien Coron. On the exact security of full domain hash. In *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, pages 229–235, 2000.

[17] Ivan Damgård. A design principle for hash functions. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 416–427, 1989.

[18] Anindya De, Luca Trevisan, and Madhur Tulsiani. Time space tradeoffs for attacks against one-way functions and prgs. In *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, pages 649–665, 2010.

[19] Yevgeniy Dodis, Siyao Guo, and Jonathan Katz. Fixing cracks in the concrete: Random oracles with auxiliary input revisited. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2017.

[20] Yevgeniy Dodis, Krzysztof Pietrzak, and Daniel Wichs. Key derivation without entropy waste. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 93–110, 2014.

[21] Yevgeniy Dodis and John P. Steinberger. Message authentication codes from unpredictable block ciphers. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 267–285, 2009.

[22] Stefan Dziembowski and Ueli M. Maurer. Tight security proofs for the bounded-storage model. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 341–350, 2002.

[23] Amos Fiat and Moni Naor. Rigorous time/space trade-offs for inverting functions. *SIAM J. Comput.*, 29(3):790–803, 1999.

[24] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, pages 186–194, 1986.

[25] Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM J. Comput.*, 35(1):217–246, 2005.

[26] Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 305–313, 2000.

[27] Oded Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *J. Cryptology*, 6(1):21–53, 1993.

[28] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM J. Comput.*, 25(1):169–192, 1996.

[29] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *J. Cryptology*, 7(1):1–32, 1994.

[30] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 102–113, 2003.

[31] Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM J. Comput.*, 45(5):1835–1869, 2016.

[32] Martin E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Trans. Information Theory*, 26(4):401–406, 1980.

[33] Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, pages 3–32, 2016.

[34] Russell Impagliazzo. Hardness as randomness: a survey of universal derandomization. *CoRR*, cs.CC/0304040, 2003.

[35] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 356–364, 1994.

[36] Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 220–229, 1997.

[37] Valentine Kabanets. Derandomization: a brief overview. *Bulletin of the EATCS*, 76:88–103, 2002.

[38] Richard M. Karp and Richard J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing, April 28-30, 1980, Los Angeles, California, USA*, pages 302–309, 1980.

[39] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007.

[40] Pravesh Kothari, Raghu Meka, and Prasad Raghavendra. Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of csps. *STOC*, 2017.

[41] Mohammad Mahmoody and Ameer Mohammed. On the power of hierarchical identity-based encryption. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 243–272, 2016.

[42] Ueli M. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *J. Cryptology*, 5(1):53–66, 1992.

[43] Ralph C. Merkle. A certified digital signature. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 218–238, 1989.

[44] Robert Morris and Ken Thompson. Password security - A case history. *Commun. ACM*, 22(11):594–597, 1979.

[45] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, pages 111–126, 2002.

[46] Noam Nisan and Avi Wigderson. Hardness vs. randomness (extended abstract). In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, pages 2–11, 1988.

[47] Philippe Oechslin. Making a faster cryptanalytic time-memory trade-off. In *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, pages 617–630, 2003.

[48] Jacques Patarin. The "coefficients h" technique. In *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, pages 328–345, 2008.

[49] Phillip Rogaway. Formalizing human ignorance. In *Progressin Cryptology - VIETCRYPT 2006, First International Conferenceon Cryptology in Vietnam, Hanoi, Vietnam, September 25-28, 2006, Revised Selected Papers*, pages 211–228, 2006.

[50] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 239–252, 1989.

[51] Stefano Tessaro. Security amplification for the cascade of arbitrarily weak prps: Tight bounds via the interactive hardcore lemma. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, pages 37–54, 2011.

[52] Dominique Unruh. Random oracles and auxiliary input. In *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, pages 205–223, 2007.

[53] Salil P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *J. Cryptology*, 17(1):43–77, 2004.

A Decomposing High-Entropy Sources

Let X be distributed uniformly over $[M]^N$ and $Z := f(X)$, where $f : [M]^N \to \{0,1\}^S$ is an arbitrary function. Fix an arbitrary $z \in \{0,1\}^S$ and let X_z be the distribution of X conditioned on $f(X) = z$. Let $S_z = N \log M - H_\infty(X_z)$ be the min-entropy deficiency of X_z. Let $\gamma > 0$ be arbitrary.

Claim 2. *For every $\delta > 0$, X_z is γ-close to a convex combination of finitely many $(P', 1-\delta)$-dense sources for*

$$P' \;=\; \frac{S_z + \log 1/\gamma}{\delta \cdot \log M}\ .$$

Proof. For ease of notation, let $S := S_z$ and $X := X_z$. Suppose X is not $(1-\delta)$-dense, as otherwise there is nothing to show. Let $Y := X$ and I be the largest subset such that there exists a y_I,

$$\mathsf{P}[Y_I = y_I] \;>\; 2^{-(1-\delta)\cdot|I|\cdot\log M}. \tag{5}$$

Let Y' be the distribution of Y conditioned on $Y_I = y_I$.

Claim 25. *$Y'_{\overline{I}}$ is $(1-\delta)$-dense.*

Proof. Suppose $Y'_{\overline{I}}$ is not $(1-\delta)$-dense. Then, there exists a non-empty set $J \subseteq \overline{I}$ and y_J such that

$$\mathsf{P}[Y'_J = y_J] \;=\; \mathsf{P}[Y_J = y_J \mid Y_I = y_I] \;>\; 2^{-(1-\delta)\cdot|J|\cdot\log M}.$$

The set $I \cup J$ now forms a subset for which

$$\begin{aligned}
\mathsf{P}[Y_{I\cup J} = y_{I\cup J}] \;&=\; \mathsf{P}[Y_I = y_I \wedge Y_J = y_J] \\
&=\; \mathsf{P}[Y_I = y_I] \cdot \mathsf{P}[Y_J = y_J \mid Y_I = y_I] \\
&>\; 2^{-(1-\delta)|I|\cdot\log M} \cdot 2^{-(1-\delta)|J|\cdot\log M} \\
&=\; 2^{-(1-\delta)|I\cup J|\cdot\log M},
\end{aligned}$$

since I and J are disjoint. This, however, contradicts the maximality of I. □

Claim 26. *$|I| \le S/(\delta \cdot \log M)$.*

Proof. On the one hand, $H_\infty(Y) \geq N \log M - S$ implies that for any y_I,

$$\begin{aligned} \mathsf{P}[Y_I = y_I] &= \sum_{y_{\bar{I}} \in [M]^{N-|I|}} \mathsf{P}[Y_I = y_I \wedge Y_{\bar{I}} = y_{\bar{I}}] \\ &\leq 2^{(N-|I|)\cdot \log M} \cdot 2^{-(N \log M - S)} \\ &= 2^{-(|I| \cdot \log M - S)}, \end{aligned}$$

and, hence, $H_\infty(Y_I) \geq |I| \cdot \log M - S$. On the other hand, because Y_I is not $(1-\delta)$-dense, $H_\infty(Y_I) < (1-\delta) \cdot |I| \cdot \log M$. Combining the above two inequalities, one obtains the desired conclusion. □

Hence, Y' is an $(S/(\delta \log M), 1-\delta)$-dense source. Set Y now to be Y conditioned on $Y_I \neq y_I$ and recursively decompose Y as long as

$$\mathsf{P}[X \in \mathrm{supp}(Y)] > \gamma \,. \tag{6}$$

Observe that $H_\infty(Y) \geq N \log M - (S + \log 1/\gamma)$ at any point in this decomposition process since

$$\begin{aligned} \mathsf{P}[Y = y] &= \mathsf{P}[X = y \mid X \in \mathrm{supp}(Y)] \\ &\leq \frac{\mathsf{P}[X = y]}{\mathsf{P}[X \in \mathrm{supp}(Y)]} \\ &\leq \frac{2^{-(N \log M - S)}}{\gamma} = 2^{-(N \log M - (S + \log 1/\gamma))} \,. \end{aligned}$$

Note that $|\mathrm{supp}(Y)|$ decreases in every step, and since $\mathrm{supp}(X)$ is finite, after finitely many steps, this process ends with a Y_{final} with $\mathsf{P}[X \in \mathrm{supp}(Y_{\mathsf{final}})] \leq \gamma$. Hence, X is a convex combination of finitely many $((S + \log 1/\gamma)/(\delta \log M), 1-\delta)$-dense sources and Y_{final}.[15] This implies that X is γ-*close* to a convex combination of $((S + \log 1/\gamma)/(\delta \log M), 1-\delta)$-dense sources (e.g., the convex combination obtained by replacing Y_{final} by the uniform distribution). □

B Standard-ROM Definitions and Security

B.1 Fiat-Shamir with Schnorr

Digital signature schemes. A digital signature scheme is a triple of algorithms $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Vfy})$, where Gen generates a signing key sk and a verification key vk, Sig takes a signing key sk and a message m and outputs a signature σ, and Vfy takes a verification key vk, a message m, and a signature σ and outputs a single bit, indicating whether σ is valid. In the $\mathcal{O}$-oracle model, all three algorithms may make calls to $\mathcal{O}.\mathsf{main}$.

The application of digital signatures $G^{\mathsf{DS},\Sigma}$ is defined via the following challenger $\mathsf{C}^{\mathsf{DS},\Sigma}$, which captures the (standard) EUF-CMA security of a digital signature scheme: Initially, $\mathsf{C}^{\mathsf{DS},\Sigma}$ generates a key pair $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}$ and passes vk to the attacker. Then, the attacker may repeatedly submit signature queries m to the challenger, who answers them by the corresponding signature $\sigma \leftarrow \mathsf{Sig}_\sigma(m)$. In the end, the challenger outputs 1 if and only if the attacker submits a pair (m^*, σ^*) with $\mathsf{Vfy}_{\mathsf{vk}}(m^*, \sigma^*) = 1$ and such that no signature query was asked for m^*.

[15] The bound on $|I|$ is easily adapted to account for entropy deficiency $S + \log 1/\gamma$ instead of S.

The discrete-logarithm problem. The discrete-logarithm problem in a group $G = \langle g \rangle$ can be phrased as an application $G^{\mathsf{DL},G}$, defined via the challenger $\mathsf{C}^{\mathsf{DL},G}$ that picks a uniformly random $x \in \mathbb{Z}_{|G|}$, passes $y := g^x$ to the attacker, and outputs 1 if and only if the attacker finds x. Observe that $G^{\mathsf{DL},G}$ is a *standard-model* application.

Schnorr signatures in the standard ROM. In the standard ROM, using the forking lemma as stated by Bellare and Neven [7], one can show the following security bound for Schnorr signatures.

Theorem 27. *Assume $G^{\mathsf{DL},G}$ for $|G| = N$ is $((S, *, t', s'), \varepsilon')$-secure, and let $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Vfy})$ be the Schnorr scheme. Then, $G^{\mathsf{DS},\Sigma}$ is $((S, T, t, s, q_{\mathsf{sig}}), \varepsilon)$-secure in the $\mathsf{RO}(N^2, N)$-model for*

$$\varepsilon \;=\; O\left(\sqrt{T\varepsilon'} + \frac{q_{\mathsf{sig}}(q_{\mathsf{sig}} + T)}{N}\right),$$

where $t = \Omega(t')$ and $s = \Omega(s')$.

B.2 TDF Encryption

Key-encapsulation mechanisms. A key-encapsulation mechanism (KEM) is a triple of algorithms $\Pi = (K, E, D)$, where K generates a public key pk and a secret key sk, E takes a public key pk and outputs a ciphertext c and a key k, and D takes a secret key sk and a ciphertext c and outputs a key k. In the $\mathcal{O}$-oracle model, all three algorithms may make calls to $\mathcal{O}.\mathsf{main}$.

The application corresponding to CPA security for KEMs $G^{\mathsf{KEM\text{-}CPA},\Pi}$ is defined via the following challenger $\mathsf{C}^{\mathsf{KEM\text{-}CPA},\Pi}$, which captures the (standard) CCA security of a KEM scheme: Initially, $\mathsf{C}^{\mathsf{KEM\text{-}CPA},\Pi}$ generates a key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow K$ and passes pk to the attacker. Then, the challenger chooses a random bit b as well as a random key k_1, computes $(c, k_0) \leftarrow E_{\mathsf{pk}}$, and returns the challenge (c, k_b). In the end, the challenger outputs 1 if and only if the attacker submits a bit b' with $b' = b$.

To capture CCA security, one consideres the application $\mathsf{C}^{\mathsf{KEM\text{-}CCA},\Pi}$ defined by the challenger $\mathsf{C}^{\mathsf{KEM\text{-}CCA},\Pi}$ that proceeds as $\mathsf{C}^{\mathsf{KEM\text{-}CPA},\Pi}$, except that the attacker gets to ask decryption queries c', which the challenger answers with $k' \leftarrow D_{\mathsf{sk}}(c')$, provided $c' \neq c$.

Trapdoor functions. The inversion problem for a trapdoor function generator F can be phrased as an application $G^{\mathsf{TDF},F}$, defined via the challenger $\mathsf{C}^{\mathsf{TDF},F}$ that generates $(f, f^{-1}) \leftarrow F$, picks a random x, passes $y := f(x)$ to the attacker, and outputs 1 if and only if the attacker finds x. Observe that $G^{\mathsf{TDF},F}$ is a *standard-model* application.

The security of TDF key encapsulation in the standard ROM. In the standard ROM, one can show the following security bound for TDF encryption.

Theorem 28. *Let Π be TDF key encapsulation. If $G^{\mathsf{TDF},F}$ is $((S', *, t', s'), \varepsilon')$-secure, then $G^{\mathsf{KEM\text{-}CPA},\Pi}$ is $((S, T, t, s), \varepsilon)$-secure in the $\mathsf{RO}(N, N)$, where*

$$\varepsilon \;=\; O\left(\varepsilon'\right)$$

and $S = S'$, $t = \Omega(t')$, and $s = \Omega(s')$.

C Salting: Deferred Material

C.1 Schemes Insecure in the AI-ROM

Fiat-Shamir from identification schemes. Abdalla *et al.* [1] showed how to—using the Fiat-Shamir paradigm—generically build signature schemes in the ROM from identification schemes (ID schemes). In a nutshell, ID schemes set up public and private keys, and the party holding the private key, the prover, identifies themselves to the party holding the public key, the verifier, by executing a commit-challenge-response protocol, where first the prover sends a "commitment" a and then answers a subsequent challenge c form the verifier by a response z, which the verifier either accepts or rejects. The main idea behind Fiat-Shamir signatures is the following: The singing and verification keys of the signature scheme are the private and public keys, respectively, of the ID scheme. To sign a message m, generate a according to the ID scheme, query the random oracle $\mathcal{O}$ to generate the challenge $c \leftarrow \mathcal{O}(a, m)$, and compute the corresponding response z. The signature for m will be $\sigma = (a, z)$. Verification works in the obvious way.

A simple attack against the generic transformation is the following: An ID scheme can be modified to always accept $(a, c, z) = (a^*, 0^N, z^*)$ for some (arbitrary) a^* and z^* while remaining secure, since it is unlikely that the verifier asks challenge 0^N. However, the signature scheme resulting from applying the above transformation to the modified ID scheme is insecure in the AI-ROM since $\mathcal{A}_1$ can (with high probability) find a message m^* such that $\mathcal{O}(a^*, m^*) = 0^N$ and therefore forge a signature $\sigma = (a^*, z^*)$ for m^*. The attack trivially extends to P-BF-ROM (for $P = 1$) by simply setting $\mathcal{O}(a^*, m^*) = 0^N$.

Full-domain hash. The full-domain hash is essentially the concatenation of the random oracle $\mathcal{O}$ with a trapdoor permutation (f, f^{-1}): $\mathsf{Sig}(m) = f^{-1}(\mathcal{O}(m))$. That is, in order to sign a message m, the signer first computes $d \leftarrow \mathcal{O}(m)$ and then inverts d with a trapdoor permutation using hte secret key f^{-1} (verification is obvious using public key f). Similarly to the application of collision-resistant hashing, full-domain hash is insecure in the AI-ROM (for any f) because one can leak two messages m and m' with $\mathcal{O}(m) = \mathcal{O}(m')$ to the attacker, who then gets a signature for one of them, which is obviously also a signature for the other. The attack trivially extends to P-BF-ROM (for $P = 2$) by simply making setting $\mathcal{O}(m) = \mathcal{O}(m')$.

C.2 More Randomness-Efficient Salting

Claim 24. $\mathsf{P}[\forall i \in [K], a_i \in \mathrm{BAD}_i] \leq (\sqrt{P/(KN)} + c)^K$.

Proof. Let $W = \frac{1}{d} \cdot A$ be the transition matrix. Let $p_1 = \mathbf{1}/N \in R^N$ denote our initial distribution of our random walk. For BAD_i, we define diagnole matrix B_i whose (j, j)-entry ($j \in [N]$) is 1 if $j \in \mathrm{BAD}_i$, otherwise 0. Note that,

$$\mathsf{P}[\forall i \in [K], a_i \in \mathrm{BAD}_i] \;=\; \mathbf{1}^T B_K W B_{K-1} W \dots B_1 p_1 \;.$$

For a matrix M, let $||M||_2 = \max_{v \neq 0} \frac{||Mv||_2}{||v||_2}$ denote the matrix norm of M. We claim that for $i \in [K]$,

$$||B_i W||_2 \;\leq\; \sqrt{\frac{|\mathrm{BAD}_i|}{N}} + c \;. \tag{7}$$

Given (7), we can finish the proof. As $p_1 = Wp_1$ and $||M_1M_2||_2 \le ||M_1||_2||M_2||_2$, we have that

$$\begin{aligned}
\mathbf{1}^T B_K W B_{K-1} W \dots B_1 p_1 &= \mathbf{1}^T (B_K W)(B_{K-1} W) \dots (B_1 W) p_1 \\
&\le ||\mathbf{1}^T||_2 \cdot \Pi_{i\in[K]} ||B_i W||_2 \cdot ||p_1|| \\
&\le \Pi_{i\in[K]} \left(\sqrt{\frac{|\mathrm{BAD}_i|}{N}} + c \right) \\
&\le \left(\sqrt{\frac{P}{KN}} + c \right)^K ,
\end{aligned}$$

where the last inequality is because $f(y) = \ln(\sqrt{y/N} + c)$ is concave for $y \ge 0$.

Now we prove (7). Given any non-zero vecotr $v = c'\mathbf{1} + v'$ where $\mathbf{1}^T v' = 0$. As $W\mathbf{1} = \mathbf{1}$,

$$||B_i W v||_2 = ||c' B_i W \mathbf{1} + B_i W y|| = ||c' B_i \mathbf{1} + B_i W y||_2 \le ||c' B_i \mathbf{1}||_2 + ||B_i W y||_2 .$$

Note that $||v||_2 = \sqrt{(c'\sqrt{N})^2 + ||v'||_2^2}$. Let $\psi_1 = \mathbf{1}, \psi_2, \dots, \psi_N$ be W's eigenvectors and $w_1, \dots, w_N$ be correspdoning eigenvalues. Note that $w_i = \lambda_i/d \le c$ for $2 \le i \le N$.

$$||Wy||_2 = \sqrt{\sum_{2\le i\le N} (\psi_i^T y)^2 w_i^2} \le c \cdot \sqrt{\sum_{i\ge 2} (\psi_i^T y)^2} = c \cdot ||y||_2 .$$

Therefore $||B_i W y||_2 \le ||B_i||_2 ||Wy||_2 \le c||y||_2 \le c||v||_2$. Moreover, $||c' B_i \mathbf{1}||_2 = c'\sqrt{|\mathrm{BAD}_i|} \le \sqrt{\frac{\mathrm{BAD}_i}{N}} ||v||_2$. We conclude

$$||B_i W v||_2 \le \left(\sqrt{\frac{|\mathrm{BAD}_i|}{N}} + c \right) ||v||_2 ,$$

and the matrix norm of $B_i W v$ is at most $\sqrt{\frac{|\mathrm{BAD}_i|}{N}} + c$. □