Non-Uniform Bounds in the Random-Permutation, Ideal-Cipher, and Generic-Group Models

Sandro Coretti[1(✉)], Yevgeniy Dodis[1], and Siyao Guo[2]

[1] New York University, New York, USA
{corettis,dodis}@nyu.edu
[2] Northeastern University, Boston, USA
s.guo@neu.edu

Abstract. The random-permutation model (RPM) and the ideal-cipher model (ICM) are idealized models that offer a simple and intuitive way to assess the conjectured standard-model security of many important symmetric-key and hash-function constructions. Similarly, the generic-group model (GGM) captures generic algorithms against assumptions in cyclic groups by modeling encodings of group elements as random injections and allows to derive simple bounds on the advantage of such algorithms.

Unfortunately, both well-known attacks, e.g., based on rainbow tables (Hellman, IEEE Transactions on Information Theory '80), and more recent ones, e.g., against the discrete-logarithm problem (Corrigan-Gibbs and Kogan, EUROCRYPT '18), suggest that the concrete security bounds one obtains from such idealized proofs are often *completely inaccurate* if one considers *non-uniform* or *preprocessing* attacks in the standard model. To remedy this situation, this work

- defines the auxiliary-input (AI) RPM/ICM/GGM, which capture both non-uniform and preprocessing attacks by allowing an attacker to leak an arbitrary (bounded-output) function of the oracle's function table;
- derives the *first* non-uniform bounds for a number of important practical applications in the AI-RPM/ICM, including constructions based on the Merkle-Damgård and sponge paradigms, which underly the SHA hashing standards, and for AI-RPM/ICM applications with computational security; and
- using simpler proofs, recovers the AI-GGM security bounds obtained by Corrigan-Gibbs and Kogan against preprocessing attackers, for a number of assumptions related to cyclic groups, such as discrete

S. Coretti—Supported by NSF grants 1314568 and 1319051.

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

S. Guo—Supported by NSF grants CNS-1314722 and CNS-1413964; Part of this work done while the author was visiting the Simons Institute for the Theory of Computing at UC Berkeley.

H. Shacham and A. Boldyreva (Eds.): CRYPTO 2018, LNCS 10991, pp. 693–721, 2018.
https://doi.org/10.1007/978-3-319-96884-1_23

logarithms and Diffie-Hellman problems, and provides new bounds for two assumptions.

An important step in obtaining these results is to port the tools used in recent work by Coretti et al. (EUROCRYPT '18) from the ROM to the RPM/ICM/GGM, resulting in very powerful and easy-to-use tools for proving security bounds against non-uniform and preprocessing attacks.

1 Introduction

The random-permutation and ideal-cipher models. The random-permutation model (RPM) and the ideal-cipher model (ICM) are idealized models that offer a simple and intuitive way to prove the (conjectured) security of many important applications. This holds especially true in the realms of symmetric cryptography and hash-function design since most constructions of block ciphers and hash functions currently do not have solid theoretical foundations from the perspective of provable security. In fact, the *exact security bounds* obtained in such idealized models are often viewed as guidance for both designers and cryptanalysts in terms of the *best possible* security level that can be achieved by the corresponding construct in the standard model. By and large, this method has been quite successful in practice, as most separations between the standard model and various idealized models [3,8,10,11,28,35] are somewhat contrived and artificial and are not believed to affect the security of widely used applications. In fact, the following *RPM/ICM methodology* appears to be a good way for practitioners to assess the best possible security level of a given (natural) application.

> **RPM/ICM methodology.** *For "natural" applications of hash functions and block ciphers, the concrete security proven in the RPM/ICM is the right bound even in the standard model, assuming the "best possible" instantiation for the idealized component (permutation or block cipher) is chosen.*

Both the RPM and the ICM have numerous very important practical applications. In fact, most practical constructions in symmetric-key cryptography and hash-function design are naturally defined in the RPM/ICM. The following are a few representative examples:

- The famous AES cipher is an example of key-alternating cipher, which can be abstractly described and analyzed in the RPM [2,12], generalizing the Even-Mansour [21,22] cipher $\mathsf{EM}_{\pi,s}(x) = \pi(x \oplus s) \oplus s$, where π is a public permutation, s is the secret key, and x is the message.
- The compression function of the SHA-1/2 [38,43] and MD5 [40] hash functions, as well as the popular HMAC scheme [4], is implemented via the Davies-Meyer (DM) hash function $\mathsf{DM}_E(x, y) = E_x(y) \oplus y$, for a block cipher E. But its collision-resistance can only be analyzed in the ICM [48].
- The round permutation of SHA-3 [37]—as part of the *sponge mode of operation* [6]—can be defined in the RPM: given old n-bit state s and new r-bit

block message x (where $r < n$), the new state is $s' = \pi(s \oplus (x \| 0^{n-r}))$, where π is a public permutation. The sponge mode is useful for building CRHFs, message authentication codes (MACs), pseudorandom functions (PRFs) [7], and key derivation functions [24], among others.
- The round function of MD6 [41] can be written as $f_Q(x) = \mathsf{trunc}_r(\pi(x \| Q))$, where Q is a constant, trunc_r is the truncation to r bits, and π is a public permutation. This construction was shown indifferentiable from a random oracle in the RPM [20].
- Many other candidate collision-resistant hash functions can be described using either ideal ciphers (e.g., the large PGV family [9]) or random permutations (e.g., [6,20,42,45]).

The generic group model. Another well-known idealized model is the so-called generic-group model (GGM), which serves the purpose of proving lower bounds on the complexity of generic attacks against common computational problems in cyclic groups used in public-key cryptography, such as the discrete-logarithm problem (DL), the computational and decisional Diffie-Hellman problems (CDH and DDH), and many more. Generic attacks are algorithms that do not exploit the specific representation of the elements of a group. This property is modeled by considering generic encoding captured by a random injection $\sigma : \mathbb{Z}_N \to [M]$ and allowing the algorithm access to a group-operation oracle, which, given a pair of encodings $(\sigma(x), \sigma(y))$, returns $\sigma(x + y)$.

The justification for the GGM is rooted in the fact that there are no unconditional hardness proofs for important group-related problems, and that there are some groups based on elliptic curves for which no better algorithms than the generic ones are known. Hence, results in the GGM provide at least some indication as to how sensible particular assumptions are. There are a plethora of security bounds proven in the GGM, e.g., lower bounds on the complexity of generic algorithms against DL or CDH/DDH by Shoup [44] or the knowledge-of-exponent assumption by Abe and Fehr [1] and Dent [17].

Non-uniformity and preprocessing. Unfortunately, a closer look reveals that the rosy picture above can only be true if one considers *uniform* attacks (as explained below). In contrast, most works (at least in theoretical cryptography) consider attackers in the *non-uniform* setting, where the attacker is allowed to obtain some arbitrary (but bounded) *advice* before attacking the system. The main rationale for this modeling comes from the realization that a determined attacker will know the parameters of a target system in advance and might be able to invest a significant amount of preprocessing to do something to speed up the actual attack, or to break many instances at once (therefore amortizing the one-time preprocessing cost). Perhaps the best known example of such attacks are *rainbow tables* [30,36] (see also [32, Sect. 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.

There are also other, more technical, reasons why the choice to consider non-uniform attackers is convenient (see [14] for details), the most important of which is security under composition. A well-known example are zero-knowledge proofs [26,27], which are *not* closed under (even sequential) composition unless one allows non-uniform attackers and simulators. Of course, being a special case of general protocol composition, this means that any work that uses zero-knowledge proofs as a subroutine must consider security against *non-uniform* attackers in order for the composition to work. Hence, it is widely believed by the theoretical community that *non-uniformity is the right cryptographic modeling of attackers*, despite being overly conservative and including potentially unrealistic attackers—due to the potentially unbounded pre-computation allowed to generate the advice.

Idealized models vs. non-uniformity and preprocessing. When considering non-uniform attackers, it turns out that the RPM/ICM methodology above is blatantly false: once non-uniformity or preprocessing is allowed, the separations between the idealized models and the standard model are *no longer* contrived and artificial, but rather lead to *impossibly good* exact security of most *widely deployed* applications. To see this, consider the following examples:

- *One-way permutations:* Hellman [30] showed that there is a preprocessing attack that takes S bits of advice and makes T queries to a permutation $\pi : [N] \to N$ and inverts a random element of $[N]$ with probability roughly ST/N. Hence, a permutation cannot be one-way against attackers of size beyond $T = S = N^{1/2}$. However, in the RPM, a random permutation is easily shown to be invertible with probability at most T/N, therefore suggesting security against attackers of size up to N.
- *Even-Mansour cipher:* In a more recent publication, Fouque *et al.* [23] showed a non-uniform $N^{1/3}$ attack against the Even-Mansour cipher that succeeds with constant probability. As with OWPs, the analysis in the RPM model suggests an incorrect security level, namely, up to the birthday bound since one easily derives an upper bound of T^2/N on the distinguishing advantage of any attacker in RPM.

Similar examples also exist in the GGM:

- *Discrete logarithms:* A generic preprocessing attack by Mihalcik [34] and Bernstein and Lange [5] (and a recent variant by Corrigan-Gibbs and Kogan [15]) solves the DL problem with advantage ST^2/N in a group of order N, whereas the security of DL in the GGM is known to be T^2/N [44].
- *Square DDH:* A generic preprocessing attack by Corrigan-Gibbs and Kogan [15] breaks the so-called *square DDH (sqDDH)* problem—distinguishing (g^x, g^{x^2}) from (g^x, g^y) in a cyclic group $G = \langle g \rangle$ of order N—with advantage $\sqrt{ST^2/N}$, whereas the security of sqDDH in the GGM can be shown to be T^2/N.

Table 1. Asymptotic upper and lower bounds on the security of applications in the AI-ICM/AI-RPM and in the standard model (SM) against (S, T)-attackers.

	AI Security	SM Security	Best Attack
OWP	$\frac{ST}{N}$	$\frac{T}{N}$	$\frac{ST}{N}$ [30]
EM	$\left(\frac{ST^2}{N}\right)^{1/2} + \frac{T^2}{N}$	$\frac{T^2}{N}$	$\left(\frac{S}{N}\right)^{1/2}$ [16]
BC-IC	$\left(\frac{ST}{K}\right)^{1/2} + \frac{T}{K}$	$\frac{T}{K}$	$\left(\frac{S}{K}\right)^{1/2}$ [16]
PRF-DM	$\left(\frac{ST}{N}\right)^{1/2} + \frac{T}{N}$	$\frac{T}{N}$	$\left(\frac{S}{N}\right)^{1/2}$ [16]
CRHF-DM	$\frac{(ST)^2}{N}$	$\frac{T^2}{N}$	not known
CRHF-S	$\frac{ST^2}{2^c} + \frac{T^2}{2^r}$	$\frac{T^2}{2^c} + \frac{T^2}{2^r}$	$\frac{ST^2}{N}$ [14]
PRF-S	$\left(\frac{ST^2}{2^c}\right)^{1/2}$	$\frac{T^2}{2^c}$	$\left(\frac{S}{N}\right)^{1/2}$ [16]
MAC-S	$\frac{ST^2}{2^c} + \frac{T}{2^r}$	$\frac{T^2}{2^c} + \frac{T}{2^r}$	$\min\left\{\frac{ST}{N}, \left(\frac{S^2T}{N^2}\right)^{1/3}\right\} + \frac{T}{N}$ [30]
CRHF-MD	$\frac{ST^2}{N}$	$\frac{T^2}{N}$	$\frac{ST^2}{N}$ [14]
PRF-MD-N	$\left(\frac{ST^3}{N}\right)^{1/2} + \frac{T^3}{N}$	$\frac{T^3}{N}$	$\left(\frac{S}{N}\right)^{1/2}$ [16]
NMAC/HMAC	$\frac{ST^3}{N}$	$\frac{T^3}{N}$	$\min\left\{\frac{ST}{N}, \left(\frac{S^2T}{N^2}\right)^{1/3}\right\} + \frac{T}{N}$ [30]

1.1 Contributions: Non-Uniform Bounds in the RPM/ICM/GGM

Given the above failure of the idealized-models methodology, this paper revisits security bounds derived in the RPM, ICM, and GGM and re-analyzes a number of applications highly relevant in practice w.r.t. their security against non-uniform attackers or preprocessing. To that end, following the seminal work of Unruh [47] as well as follow-up papers by Dodis *et al.* [18] and Coretti *et al.* [14], the idealized models are replaced by weaker counterparts that adequately capture non-uniformity and preprocessing by allowing the attacker to obtain *oracle-dependent* advice. The resulting models, called the *auxiliary-input* RPM, ICM, and GGM, are parameterized by S ("space") and T ("time") and work as follows: The attacker $\mathcal{A}$ in the AI model 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 idealized primitive $\mathcal{O}$, and computes some advice z of size at most 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 $\mathcal{A}_1$, may have additional application-specific restrictions, such as bounded running time, etc.). The oracle-dependent advice naturally maps to non-uniform advice when the random oracle is instantiated, and, indeed, none of the concerns expressed in the above examples remain valid in the AI-RPM/ICM/GGM.

Symmetric primitives. In the AI-RPM and AI-ICM, this work analyzes and derives non-uniform security bounds for (cf. Table 1 and Sect. 4):

- *basic applications* such as inverting a random permutation (OWP), the Even-Mansour cipher (EM), using the ideal cipher as a block cipher directly (BC-IC), the PRF security of Davies-Meyer (PRF-DM), the collision resistance of a salted version of the Davies-Meyer compression function (CRHF-DM);

Table 2. Asymptotic upper and lower bounds on the security of applications in the generic-group model against (S, T)-attackers in the AI-ROM; new bounds are in a bold-face font. The value t for the one-more DL problem stands for the number of challenges requested by the attacker. The attack against MDL succeeds with constant probability and requires that $ST^2/t + T^2 = \Theta(tN)$.

	AI-GGM Security	GGM Security	Best Attack
DL/CDH	$\frac{ST^2}{N} + \frac{T^2}{N}$	$\frac{T^2}{N}$	$\frac{ST^2}{N}$ [5,15,34]
t-fold MDL	$\left(\frac{S(T+t)^2}{tN} + \frac{(T+t)^2}{tN}\right)^t$	$\left(\frac{(T+t)^2}{tN}\right)^t$	see caption [15]
DDH	$\left(\frac{ST^2}{N}\right)^{1/2} + \frac{T^2}{N}$	$\frac{T^2}{N}$	$\frac{ST^2}{N}$ [5,15,34]
sqDDH	$\left(\frac{ST^2}{N}\right)^{1/2} + \frac{T^2}{N}$	$\frac{T^2}{N}$	$\left(\frac{ST^2}{N}\right)^{1/2}$ [15]
OM-DL	$\left(\frac{S(T+t)^2}{N}\right) + \frac{(T+t)^2}{N}$	$\frac{T^2}{N}$	$\frac{ST^2}{N}$ [5,15,34]
KEA	$\frac{ST^2}{N}$	$\frac{T^2}{N}$	not known

- the collision-resistance, the PRF security, and the MAC security of the *sponge construction*, which underlies the SHA-3 hashing standard;
- the collision-resistance of the *Merkle-Damgård construction with Davies-Meyer (MD-DM)*, which underlies the SHA-1/2 hashing standards, and PRF/MAC security of NMAC and HMAC.

Surprisingly, except for OWPs [16], no non-uniform bounds were known for any of the above applications; not even for applications as fundamental as BC-IC, Even-Mansour, or HMAC.

The bounds derived for OWP and the collision-resistance (CR) of Sponges and MD-DM are tight, i.e., there exist matching attacks by Hellman [30] (for OWPs) and by Coretti *et al.* [14] (for CR). For the remaining primitives significant gaps remain between the derived security bounds and the best known attacks. Closing these gaps is left as an interesting (and important) open problem.

Generic groups. In the AI-GGM, the following applications are analyzed w.r.t. their security against preprocessing (cf. Table 2 and Sect. 5): the discrete-logarithm problem (DL), the multiple-discrete-logarithms problem (MDL), the computational Diffie-Hellman problem (CDH), the decisional Diffie-Hellman problem (DDH), the square decisional Diffie-Hellman problem (sqDDH), the one-more discrete-logarithm problem (OM-DL), and the knowledge-of-exponent assumption (KEA).

- For DL, MDL, CDH, DDH, and sqDDH, the derived bounds match those obtained in recent work by Corrigan-Gibbs and Kogan [15]. As highlighted below, however, the techniques used in this paper allow for much simpler proofs than the one based on incompressibility arguments in [15]. All of these bounds are tight, except those for DDH, for which closing the gap remains an open problem.
- The bounds for OM-DL and KEA are new and may be non-trivial to derive using compression techniques.

Computational security. Idealized models such as the ROM, RPM, and ICM are also often used in conjunction with computational hardness assumptions such as one-way functions, hardness of factoring, etc. Therefore, this paper also analyzes the security of public-key encryption based on trapdoor functions (cf. Sect. 6) in the AI-RPM, specifically, of a scheme put forth by Phan and Pointcheval [39]. Other schemes in the AI-RPM/ICM, e.g., [29,31], can be analyzed similarly.

1.2 Methodology: Pre-Sampling

Bit-fixing oracles and pre-sampling. Unfortunately, while solving the issue of not capturing non-uniformity and preprocessing, the AI models are considerably more difficult to analyze than the traditional idealized models. From a technical point, the key difficulty is the following: *conditioned on the leaked value* z, which can depend on the entire function table of $\mathcal{O}$, many of the individual values $\mathcal{O}(x)$ are no longer random to the attacker, which ruins many of the key techniques utilized in the traditional idealized models, such as lazy sampling programmability, etc.

One way of solving the above issues is to use incompressibility arguments, as introduced by Gennaro and Trevisan [25] and successfully applied to OWPs by De *et al.* [16], to the random-oracle model by Dodis *et al.* [14], and to the GGM by Corrigan-Gibbs and Kogan [15]. Compression-based proofs generally lead to tight bounds, but are usually quite involved and, moreover, seem inapplicable to computationally secure applications. Hence, this paper, adopts the much simpler and more powerful pre-sampling approach taken recently by Coretti *et al.* [14] and dating back to Unruh [47]. The pre-sampling technique can be viewed as a general reduction from the auxiliary-input model to the so-called *bit-fixing* (BF) model, where the oracle can be arbitrarily fixed on some P coordinates, for some parameter P, but the remaining coordinates are chosen at random and independently of the fixed coordinates. Moreover, the non-uniform S-bit advice of the attacker in this model can only depend on the P fixed points, but *not* on the remaining truly random points. This makes dealing with the BF model much easier than with the AI model, as many of the traditional proof techniques can again be used, provided that one avoids the fixed coordinates.

Bit-fixing vs. auxiliary input. In order for the BF model to be useful, this work shows that any (S, T)-attack in the AI-RPM/ICM/GGM model will have similar advantage in the P-BF-RPM/ICM/GGM model for an appropriately chosen P, up to an *additive* loss of $\delta(S, T, P) \approx ST/P$. Moreover, for the special case of unpredictability applications (e.g., CRHFs, OWFs, etc.), one can set P to be (roughly) ST, and achieve a *multiplicative* loss of 2 in the exact security. This gives a general recipe for dealing with the AI models as follows: (a) prove security $\varepsilon(S, T, P)$ of the given application in the P-BF model; (b) for unpredictability applications, set $P \approx ST$, and obtain final AI security roughly $2 \cdot \varepsilon(S, T, ST)$; (c) for general applications, choose P to minimize $\varepsilon(S, T, P) + \delta(S, T, P)$.

The proof of the above connection is based on a similar connection between the AI-ROM and BF-ROM shown by [14] (improving a weaker original bound

of Unruh [47]). While borrowing a lot of tools from [14], the key difficulty is ensuring that the P-bit-fixing cipher, which "approximates" the ideal cipher conditioned on the auxiliary input z, is actually a valid cipher: the values at fixed points cannot repeat, and the remaining values are chosen at random from the "unused" values (similar issues arise for generic groups). Indeed, the proof in this paper is more involved and the resulting bounds are slightly worse than those in [14].

Using the power of pre-sampling to analyze the applications presented above, the technical bulk consists of showing the security of these applications in the easy-to-handle BF-RPM/ICM/GGM, and then using Theorem 1 to translate the resulting bound to the AI-RPM/ICM/GGM. Most of BF proofs are remarkably straightforward extensions of the traditional proofs (without auxiliary input), which is a great advantage of the pre-sampling methodology over other approaches, such as compression-based proofs.

Computational security. Note that, unlike compression-based techniques [15,18], 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.

Fortunately, the multiplicative variant of pre-sampling for unpredictability applications 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 used. Furthermore, even though the security of public-key encryption is *not* an unpredictability application, the analysis in Sect. 6 shows a way to use multiplicative pre-sampling for the part that involves the reduction to a computational assumption.

1.3 Related Work

Tessaro [46] also adapted the presampling technique by Unruh to the random-*permutation* model; the corresponding bound is suboptimal, however. De *et al.* [16] study the effect of salting for inverting a *permutation* as well as for a specific pseudorandom generator based on one-way permutations.

Corrigan-Gibbs and Kogan [15], investigate the power of preprocessing in the GGM. Besides deriving security bounds for a number of important GGM applications, they also provide new attacks for DL (based on [5,34]), MDL, and sqDDH.

The most relevant papers in the AI-ROM are those by Unruh [47], Dodis *et al.* [18], and Coretti *et al.* [14]. Chung *et al.* [13] 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 pre-processing in this important case. However, they did not focus on the exact security and obtained suboptimal

bounds (compared to the expected "birthday" bound obtained by [18]). Using salting to obtain non-uniform security was also advocated by Mahmoody and Mohammed [33], who used this technique for obtaining non-uniform black-box separation results.

The realization that multiplicative error is enough for unpredictability applications and can lead to non-trivial savings, is related to the work of Dodis *et al.* [19] in the context of improved entropy loss of key derivation schemes.

2 Capturing the Models

This section explains how the various idealized models considered in this paper—the ideal-cipher-model (ICM), the random-permutation model (RPM), and the generic-group model (GGM)—are captured. Attackers in these models are modeled as two-stage attackers $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, and applications as (single-stage) challengers C. Both $\mathcal{A}$ and C are given access to an oracle $\mathcal{O}$. Oracles $\mathcal{O}$ 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. In certain scenarios it is also useful to consider an additional interface $\mathsf{main\text{-}c}$ that is only available to the challenger C.

Notation. Throughout this paper, P, K, N, and M are natural numbers and $[x] = \{0, \ldots, x-1\}$ for $x \in \mathbb{N}$. For applications in the generic-group model, $[N]$ is identified with the cyclic group $\mathbb{Z}_N$. Furthermore, denote by $\mathcal{P}_N$ the set of permutations $\pi : [N] \to [N]$ and by $\mathcal{I}_{N,M}$ the set of injections $f : [N] \to [M]$.

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. Some oracles may also have an additional interface $\mathcal{O}.\mathsf{main\text{-}c}$. Oracles used in this work are:

- *Auxiliary-input ideal cipher* $\mathsf{AI\text{-}IC}(K, N)$*:* Samples a random permutation $\pi_k \leftarrow \mathcal{P}_N$ for each $k \in [K]$; outputs all π_k at $\mathcal{O}.\mathsf{pre}$; answers both forward and backward queries $(k, x) \in [N]$ at $\mathcal{O}.\mathsf{main}$ by the corresponding value $\pi_k(x) \in [N]$ or $\pi_k^{-1}(x) \in [N]$, respectively.
- *Bit-fixing ideal cipher* $\mathsf{BF\text{-}IC}(P, K, N)$*:* Takes a list at $\mathcal{O}.\mathsf{pre}$ of at most P query/answer pairs (without collisions for each k); samples a random permutation $\pi_k \leftarrow \mathcal{P}_N$ consistent with said list for each k; the other interfaces behave as with $\mathsf{AI\text{-}IC}$.
- *Auxiliary-input random permutation* $\mathsf{AI\text{-}RP}(N)$*:* Special case of an auxiliary-input ideal cipher with $K = 1$.
- *Bit-fixing random permutation* $\mathsf{BF\text{-}RP}(P, N)$*:* Special case of a bit-fixing ideal cipher with $K = 1$.
- *Auxiliary-input generic group* $\mathsf{AI\text{-}GG}(N, M)$*:* Samples a random injection $\sigma \leftarrow \mathcal{I}_{N,M}$; outputs all of σ at $\mathcal{O}.\mathsf{pre}$; answers *forward* queries $x \in [N]$ at $\mathcal{O}.\mathsf{main}$ by the corresponding value $\sigma(x) \in [N]$; answers *group-operation* queries (s, s') at $\mathcal{O}.\mathsf{main}$ as follows: if $s = \sigma(x)$ and $s' = \sigma(y)$ for some x, y, the oracle replies by $\sigma(x+y)$ and by $\perp$ otherwise; answers *inverse* queries s at interface $\mathcal{O}.\mathsf{main\text{-}c}$ by returning $\sigma^{-1}(s)$ if s is in the range of F and by $\perp$ otherwise.

- *Bit-fixing generic group* $\mathsf{BF\text{-}GG}(P, N, M)$*:* Samples a random size-N subset $\mathcal{Y}$ of $[M]$ and outputs $\mathcal{Y}$ at $\mathcal{O}.\mathsf{pre}$; takes a list at $\mathcal{O}.\mathsf{pre}$ of at most P query/answer pairs without collisions and all answers in $\mathcal{Y}$; samples a random injection $\sigma \leftarrow \mathcal{I}_{N,M}$ with range $\mathcal{Y}$ and consistent with said list; the other interfaces behave as with $\mathsf{AI\text{-}GG}$.
- *Standard model:* None of the interfaces offer any functionality.

The parameters P, K, N, and 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. Note that the non-auxiliary-input versions of the above oracles can be defined by not offering any functionality at $\mathcal{O}.\mathsf{pre}$. However, they are not used in this paper.

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 the preprocessing. 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 1. *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, if additional restrictions, captured by some parameters p, are imposed on $\mathcal{A}_2$ (e.g., 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), $\mathcal{A}$ is referred to as (S, T, p)-attacker.

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 $\mathcal{O}.\mathsf{main}$ as well as possibly to $\mathcal{O}.\mathsf{main\text{-}c}$, interacts with 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

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

where $\mathcal{A}_2^{\mathcal{O}.\mathsf{main}}(\mathcal{A}_1^{\mathcal{O}.\mathsf{pre}}) \leftrightarrow \mathsf{C}^{\mathcal{O}.\mathsf{main},\mathcal{O}.\mathsf{main\text{-}c}}$ 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 2. *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 state and prove Theorem 1 in Sect. 3, 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.

3 Auxiliary Input vs. Bit Fixing

Since dealing with idealized models with *auxiliary input (AI)* directly is difficult, this section establishes useful connections between AI models and their *bit-fixing (BF)* counterparts, which are much less cumbersome to analyze. Specifically, for ideal ciphers, random permutations (as special cases of ideal ciphers), and generic groups, Theorem 1 below relates the advantage of attackers in a BF model to that in the corresponding AI model, allowing to translate the security of (1) *any* application at an *additive* security loss and of (2) *unpredictability* applications at a *multiplicative* security loss from the BF setting to the AI setting.

Theorem 1. *Let $P, K, N, M \in \mathbb{N}$, $N \ge 16$, and $\gamma > 0$. Moreover, let* $(\mathsf{AI}, \mathsf{BF}) \in \{(\mathsf{AI\text{-}IC}(K,N), \mathsf{BF\text{-}IC}(P,K,N)), (\mathsf{AI\text{-}GG}(N,M), \mathsf{BF\text{-}GG}(P,N,M))\}$. *Then,*

1. *if an application G is $((S,T,p),\varepsilon')$-secure in the* BF*-model, it is $((S,T,p),\varepsilon)$-secure in the* AI*-model, where*

$$\varepsilon \;\le\; \varepsilon' + \frac{6(S + \log \gamma^{-1}) \cdot T_G^{\mathsf{comb}}}{P} + \gamma\,;$$

2. *if an* unpredictability *application G is $((S,T,p),\varepsilon')$-secure in the* BF*-model for*

$$P \;\ge\; 6(S + \log \gamma^{-1}) \cdot T_G^{\mathsf{comb}},$$

it is $((S,T,p),\varepsilon)$-secure in the AI*-model for*

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

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

Proof Outline

This section contains a brief outline of the proof of Theorem 1. The full proof of Theorem 1 is provided in the full version of this paper; it follows the high-level structure of the proof in [14], where a similar theorem is shown for the random-oracle model.

1. *Leaky sources vs. dense sources:* A (K, N)-cipher source X is the random variable corresponding to the function table of a cipher $F : [K] \times [N] \to [M]$. It turns out that if X has min-entropy $H_\infty(X) = K \log N! - S$ for some S, it can be replaced by a convex combination of so-called *dense* sources, which are fixed on a subset of the coordinates and have almost full min-entropy everywhere else:

Definition 3. *A (K, N)-cipher source X is called $(\bar{P}, 1-\delta)$-*dense *for $\bar{P} = (P_1, \ldots, P_K) \in [N]^K$ if it is fixed on at most P_k coordinates $(k, \cdot)$ for each $k \in [K]$ and if for all families $I = \{I_k\}_{k\in[K]}$ of subsets I_k of* non-fixed *coordinates $(k, \cdot)$,*

$$H_\infty(X_I) \;\geq\; (1-\delta) \sum_{k=1}^{K} \log(N - P_k)^{\underline{|I_k|}},$$

where $a^{\underline{b}} := a!/(a-b)!$ and X_I is X restricted to the coordinates in I. X is called $(1-\delta)$-dense *if it is $(0, 1-\delta)$-dense, and $\bar{P}$-*fixed *if it is $(\bar{P}, 1)$-dense.*

More concretely, one can prove that a cipher source X as above is close to a convex combination of finitely many $(\bar{P}', 1-\delta)$-dense sources for some $\bar{P} = (P_1, \ldots, P_K)$ satisfying $\sum_{k=1}^{K} P_k \approx \frac{S}{\delta}$. The proof is an adaptation of the proof of the corresponding lemma for random functions in [14], the difference being that the version here handles cipher sources.

2. *Dense sources vs. bit-fixing sources:* Any dense source has a corresponding bit-fixing source, which is simply a function table chosen uniformly at random from all those that agree with the P fixed positions. It turns out that a T-query distinguisher's
   - *advantage* at telling a dense source and its corresponding bit-fixing source apart can be upper bounded by approximately $T\delta$, and that its
   - *probability of outputting 1* is at most a factor of approximately $2^{T\delta}$ larger when interacting with the bit-fixing as compared to the dense source.

   Compared to the case of random functions [14], some additional care is needed to properly handle *inverse* queries. Given the above, by setting $\delta \approx S/P$, one obtains additive and multiplicative errors of roughly ST/P and $2^{ST/P}$, respectively.
3. *From bit fixing to auxiliary input:* The above almost immediately implies that an application that is $((S, T), \varepsilon)$-secure in the BF-ICM is $((S, T), \varepsilon')$-secure in the AI-ICM for

$$\varepsilon' \;\approx\; \varepsilon + \frac{ST}{P}$$

and even

$$\varepsilon' \approx 2\varepsilon$$

if it is an unpredictability application, by setting $P \approx ST$. Observe that for the additive case, the final security bound in the AI-ICM is obtained by choosing P in a way that minimizes $\varepsilon(P) + ST/P$.

For the generic-group model, the proof proceeds similarly, with two important observations:

(a) once the range is fixed, a random injection behaves like a random permutation, which is covered by ideal ciphers as a special case;
(b) the group-operation oracle can be implemented by three (two inverse and one forward) calls to the injection.

4 Non-Uniform Bounds for Hash Functions and Symmetric Primitives

This section derives non-uniform security bounds for a number of primitives commonly analyzed in either the random-permutation model (RPM) or the ideal-cipher model (ICM). The primitives in question can be grouped into *basic*, *sponge-based*, and *Merkle-Damgård-based* applications.

In the following, for primitives in the RPM, $\pi, \pi^{-1} : [N] \to [N]$ denote the permutation and its inverse to which $\mathsf{AI\text{-}RP}(N)$ and $\mathsf{BF\text{-}RP}(P, N)$ offer access at interface main. Similarly, for primitives in the ICM $E, E^{-1} : [K] \times [N] \to [N]$ denote the ideal cipher and its inverse to which $\mathsf{AI\text{-}IC}(K, N)$ and $\mathsf{BF\text{-}IC}(P, K, N)$ offer access at interface main (cf. Sect. 2).

Basic applications. The security of the following basic applications in the RPM resp. ICM is considered:

- *One-way permutation inversion (OWP):* Given $\pi(x)$ for an $x \in [N]$ chosen uniformly at random, an attacker has to find x.
- *Even-Mansour cipher (EM):* The PRF security of the Even-Mansour cipher

$$\mathsf{EM}_{\pi,s}(m) := \pi(m \oplus s_2) \oplus s_1$$

  with key $s = (s_1, s_2)$.
- *Ideal cipher as block cipher (ICM):* The PRF security of the ideal cipher used as a block cipher directly.
- *PRF security of Davies-Meyer (PRF-DM):* The PRF security of the Davies-Meyer (DM) compression function DM_E

$$\mathsf{DM}_E(h, m) := E(m, h) \oplus h$$

  when h is used as the key.
- *A collision-resistant variant of Davies-Meyer (CRHF-DM):* The collision-resistance of a salted variant

$$\mathsf{DM}_{E,a,b}(h, m) := E(m, h) + am + bh$$

  of the DM compression function, where the first-stage attacker $\mathcal{A}_1$ is unaware of the public random salt value (a, b).

Sponge-based constructions. The sponge construction is a popular hash-function design paradigm and underlies the SHA-3 hash-function standard. For $N = 2^n$, $r \leq n$, $c = n - r$, it hashes a message $m = m_1 \cdots m_\ell$ consisting of r-bit blocks m_i to $y := \mathsf{Sponge}_{\pi,\mathsf{IV}}(m)$ as follows, where $\mathsf{IV} \in \{0,1\}^c$ is a c-bit initialization vector (IV):[1]

1. Set $s_0 \leftarrow 0^r \| \mathsf{IV}$.
2. For $i = 1, \ldots, \ell$: set $s_i \leftarrow \pi(m_i \oplus s_{i-1}^{(1)} \| s_{i-1}^{(2)})$, where $s_{i-1}^{(1)}$ denotes the first r bits of s_{i-1} and $s_{i-1}^{(2)}$ the remaining c bits.
3. Output $y := s_\ell^{(1)}$.

This work considers the following applications based on the sponge paradigm:

- *Collision-resistance:* The collision resistance of the sponge construction for a randomly chosen public IV unknown to the first-stage attacker $\mathcal{A}_1$.
- *PRF security:* The PRF security of the sponge construction with the IV serving as the key.
- *MAC security:* The MAC security of the sponge construction with the IV serving as the key.

Merkle-Damgård constructions with Davies-Meyer: Another widely used approach to the design of hash functions is the well-known Merkle-Damgård paradigm. For a compression function $f : [N] \times [K] \to [N]$ and an IV $\mathsf{IV} \in [N]$, a message $\overline{m} = m_1 \cdots m_\ell$ consisting of ℓ blocks $m_i \in [K]$, is hashed to $y := \mathsf{MD}_{f,\mathsf{IV}}(\overline{m})$ as follows:[2]

1. Set $h_0 \leftarrow \mathsf{IV}$.
2. For $i = 1, \ldots, \ell$: set $h_i \leftarrow f(h_{i-1}, m_i)$.
3. Output $y := h_\ell$.

This work considers the Merkle-Damgård construction with f instantiated by the Davies-Meyer compression function

$$\mathsf{DM}_E(h, m) \;:=\; E(m, h) \oplus h\,,$$

resulting in the *Merkle-Damgård-with-Davies-Meyer* function (MD-DM)

$$\mathsf{MD\text{-}DM}_{E,\mathsf{IV}}(\overline{m}) \;:=\; \mathsf{MD}_{\mathsf{DM}_E,\mathsf{IV}}(\overline{m})\,,$$

which underlies the SHA-2 hashing standard. This work considers the following applications based on the MD-DM hash function:

- *Collision-resistance:* The collision resistance of the MD-DM construction for a randomly chosen public IV unknown to the first-stage attacker $\mathcal{A}_1$.
- *PRF security:* The PRF security of the NMAC/HMAC variants

$$\mathsf{NMAC}_{E,k}(\overline{m}) \;:=\; \mathsf{DM}_E(k_1, \mathsf{MD\text{-}DM}_{E,k_2}(\overline{m}))$$

  of the MD-DM construction with key $k = (k_1, k_2)$.
- *MAC security:* The MAC security of the NMAC/HMAC variant of the MD-DM construction.

[1] To keep things simple, no padding is considered here.

[2] As with the sponge construction, for simplicity no padding is considered here.

Discussion. The asymptotic security bounds derived for the applications listed above are summarized in Table 1. No non-uniform bounds were previously known for any of these primitives, except for OWPs, for which the same bound was derived by De *et al.* [16] using an involved, compression-based proof.

As can be seen from Table 1, a matching attack, derived by Hellman *et al.* [30], is known for OWPs. Moreover, for CRHFs based on sponges and Merkle-Damgård with Davies-Meyer, a variant of a recent attack by Coretti *et al.* [14] closely matches the derived bounds.[3] For the remaining applications, significant gaps remain: For indistinguishability applications such as BI-IC and PRFs, adapting an attack on PRGs by De *et al.* [16] results in an advantage of roughly $\sqrt{S/N}$. For the MAC applications, the best attacks are based on rainbow tables for inverting functions [30].

All security bounds are derived by following the bit-fixing approach: the security of a particular application is assessed in the *bit-fixing (BF)* RPM/ICM, and then Theorem 1 is invoked to obtain a corresponding bound in the *auxiliary-input (AI)* RPM/ICM and similarly for the random-permutation model. Deriving security bounds in the BF-ICM/RPM turns out to be quite straightforward, and all of the proofs closely follow the corresponding proofs in the ICM/RPM without auxiliary input; intuitively, the only difference is that one needs to take the list $\mathcal{L}$ of the at most P input/output pairs where $\mathcal{A}_1$ fixes the random permutation or the ideal cipher.

The security proofs for one-way permutations, the ideal cipher as block cipher, the collision-resistant variant of Davies-Meyer, collision-resistance of the sponge construction, and the PRF and MAC security of NMAC/HMAC with Davies-Meyer are provided after the brief overview below. The precise definitions of the remaining applications as well as the corresponding theorems and proofs can be found in the full version of this paper.

4.1 One-Way Permutations

The one-way-permutation inversion application G^{OWP} is defined via the challenger $\mathsf{C}^{\mathsf{OWP}}$ that randomly and uniformly picks an $x \in N$, passes $y := \pi(x)$ to the attacker, and outputs 1 if and only if the attacker returns x.

Theorem 2 below provides an upper bound on the success probability of any attacker in inverting π in the $\mathsf{AI\text{-}RP}'$-model, which is defined as the $\mathsf{AI\text{-}RP}$-model, except that no queries to π^{-1} are allowed. The bound matches known attacks (up to logarithmic factors) and are also shown by De *et al.* [16] via a more involved compression argument.

Theorem 2. *The application* G^{OWP} *is* $\left((S,T),\ \tilde{O}\left(\frac{ST}{N}\right)\right)$*-secure in the* $\mathsf{AI\text{-}RP}'(N)$*-model for* $N \geq 16$.

Proof. It suffices to show that G^{OWP} is $\left((S,T), O\left(\frac{P+T}{N}\right)\right)$-secure in the $\mathsf{BF\text{-}RP}'(P,N)$-model. Then, by observing that $T^{\mathsf{comb}}_{G^{\mathsf{OWP}}} = T+1$, setting $\gamma := 1/N$

[3] The original attack by [14] was devised for Merkle-Damgård with a random compression function.

and $P = 2(S + \log N)(T + 1) = \tilde{O}(ST)$, and applying Theorem 1, the desired conclusion follows.

Assume $P + T < N/2$ 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 and always queries its output. Let $\mathcal{L} = \{(x'_1, y'_1), \ldots, (x'_P, y'_P)\}$ be the list submitted by $\mathcal{A}_1$. Recall that the challenger uniformly and randomly picks an x from $[N]$ and outputs $y := \pi(x)$. Let $x_1, \ldots, x_T$ denote the queries made by $\mathcal{A}_2$ and let $y_i := \pi(x_i)$ for $i \in [T]$ be the corresponding answers. Let $\mathcal{E}$ be the event that y appears in $\mathcal{L}$ namely $x = x'_i$ for some $i \in [P]$. Note that

$$\begin{aligned} \mathrm{Succ}_{G,\mathsf{BF\text{-}RP}}(\mathcal{A}) &\leq \mathsf{P}[\mathcal{E}] + \mathsf{P}[\exists i \in [T], x_i = x | \neg\mathcal{E}] \\ &\leq \mathsf{P}[\mathcal{E}] + \sum_{i=1}^{T} \mathsf{P}[x_i = x | \neg\mathcal{E}, x_1 \neq x, \ldots, x_{i-1} \neq x] \ . \end{aligned}$$

Observe that $\mathsf{P}[\mathcal{E}] \leq P/N$. Moreover, conditioned on $y \notin \mathcal{L}$ and any fixed choice of $(x_1, y_1), \ldots, (x_{i-1}, y_{i-1})$, x_i is a deterministic value while x is uniformly distributed over $[N] \setminus \{x_1, \ldots, x_{i-1}, x'_1, \ldots, x'_P\}$. Thus,

$$\mathsf{P}[x_i = x | \neg\mathcal{E}, x_1 \neq x, \ldots, x_{i-1} \neq x] \ \leq \ 1/(N - P - T) \ \leq \ 2/N \ ,$$

where the second inequality uses $P + T < N/2$. Therefore, $\mathrm{Succ}_{G,\mathsf{BF\text{-}RP}}(\mathcal{A}) \leq \frac{P}{N} + \frac{2T}{N} = O(\frac{P+T}{N})$. □

4.2 The Ideal Cipher as a Block Cipher

The ideal cipher can be directly used as a block cipher even in the presence of leakage. The corresponding application $G^{\mathsf{BC\text{-}IC}}$ is defined via the following challenger $\mathsf{C}^{\mathsf{BC\text{-}IC}}$: it initially chooses random bit $b \leftarrow \{0,1\}$; if $b = 0$, it picks a key $k^* \leftarrow [K]$ uniformly at random, and answers forward queries $m \in [N]$ made by $\mathcal{A}_2$ by the value $E(k^*, m)$ and inverse queries $c \in [N]$ by $E^{-1}(k^*, c)$; if $b = 1$, forward queries m are answered by $f(m)$ and inverse queries c by $f^{-1}(c)$, where f is an independently chosen uniform random permutation; the attacker wins if and only if he correctly guesses b.

Theorem 3. *Application* $G^{\mathsf{BC\text{-}IC}}$ *is* $\left((S,T,q), \tilde{O}\left(\frac{T}{K} + \sqrt{S(T+q)/K}\right)\right)$*-secure in the* $\mathsf{AI\text{-}IC}(K,N)$*-model for* $N \geq 16$.

Proof. It suffices to show that $G^{\mathsf{BC\text{-}IC}}$ is $((S,T,q), O((T+P)/K))$-secure in the $\mathsf{BF\text{-}IC}(P,K,N)$-model since then the theorem follows by observing that $T^{\mathsf{comb}}_{G^{\mathsf{BC\text{-}IC}}} = T + q$, setting $\gamma := 1/N$ and

$$P \ := \ \sqrt{(S + \log N)(T+q)K} \ = \tilde{\Theta}\left(\sqrt{S(T+q)K}\right),$$

and applying Theorem 1.

Clearly, $\mathcal{A}_2$ only has non-zero advantage in guessing bit b if it makes a (forward or inverse) query involving the key k^* chosen by the challenger or if k^* appears in one of the prefixed query/answer pairs. The latter occurs with probability at most P/K, whereas the former occurs with probability at most $T/(K-(T+P)) \leq 2T/K$, using that $T+P \leq K/2$, an assumption one can always make since, otherwise, $G^{\mathsf{BC\text{-}IC}}$ is trivially $O((T+P)/K)$-secure. □

4.3 A Collision-Resistant Variant of Davies-Meyer

The plain Davies-Meyer (DM) compression function cannot be collision-resistant against non-uniform attackers, which begs the question of if and how it can be salted to withstand non-uniform attacks. To that end, let $N = K = 2^\kappa$ for some $\kappa \in \mathbb{N}$ and interpret $[N]$ as a finite field of size N. For two values $a, b \in [N]$, let

$$\mathsf{DM}_{E,a,b}(h,m) \;:=\; E(m,h) + am + bh \;.$$

Note that for $a = 0$ and $b = 1$, $\mathsf{DM}_{E,a,b}$ is the usual DM compression function.

The application $G^{\mathsf{CRHF\text{-}DM}}$ of collision-resistance of the salted DM function is defined via the following challenger $\mathsf{C}^{\mathsf{CRHF\text{-}DM}}$: it picks two random values $a, b \in [N]$ and passes them to the attacker; the attacker wins if and only if it returns two pairs $(h,m) \neq (h',m')$ such that $\mathsf{DM}_{E,a,b}(h,m) = \mathsf{DM}_{E,a,b}(h',m')$.

Theorem 4. *$G^{\mathsf{CRHF\text{-}DM}}$ is $\left((S,T), \tilde{O}\left(\frac{(ST)^2}{N}\right)\right)$-secure in the $\mathsf{AI\text{-}IC}(N,N)$-model for $N \geq 16$.*

Proof. At the cost of at most 2 additional queries to E, assume that the pairs (h,m) and (h',m') output by $\mathcal{A}_2$ are such that $\mathcal{A}_2$ has queried its oracle E on all points $\mathsf{DM}_{E,a,b}$ would query E when evaluated on (h,m) and (h',m'). It suffices to show that $G^{\mathsf{CRHF\text{-}DM}}$ is $\left((S,T), O\left(\frac{T^2}{N} + \frac{P(P+T)}{N}\right)\right)$-secure in the $\mathsf{BF\text{-}IC}(P,N,N)$-model. Then, by observing that $T^{\mathsf{comb}}_{G^{\mathsf{CRHF\text{-}DM}}} = T+2$, setting $\gamma := 1/N$ and $P = 2(S+\log N)(T+2) = \tilde{O}(ST)$, and applying Theorem 1, the desired conclusion follows.

Set $T' := T+2$ and consider an interaction of $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and $\mathsf{C}^{\mathsf{CRHF\text{-}DM}}$ in the $\mathsf{BF\text{-}IC}(P,N,N)$-model. Denote by $((k'_i, x'_i), y'_i)$ for $i = 1, \ldots, P$ the query/answer pairs prefixed by $\mathcal{A}_1$ and by $((k_i, x_i), y_i)$ for $i = 1, \ldots, T'$ the queries $\mathcal{A}_2$ makes to E. Let $\mathcal{E}$ be the event that there exists *no* collision among the prefixed values, i.e., there exist no $i \neq j$ such that

$$E(k'_i, x_i) + ak'_i + bh'_j \;=\; E(k'_j, x_j) + ak'_j + bh'_j \tag{1}$$

and that $b \neq 0$. For any fixed $i \neq j$, consider two cases:

1. $k_i \neq k_j$: in this case, the two pairs cause a collision if and only if

$$a \;=\; \frac{(y'_j - y'_i) - b(x'_i - x'_j)}{k'_i - k'_j} \;,$$

which happens with probability at most $1/N$.

2. $k_i = k_j$: in this case, $x'_i \neq x'_j$, and the two pairs cause a collision if and only if $b = (y'_j - y'_i)/(x'_i - x'_j)$, which happens with probability at most $1/N$ as well.

Summarizing, $\mathsf{P}[\neg\mathcal{E}] \leq (P^2+1)/N = O\left(P^2/N\right)$.

Moving to queries made by $\mathcal{A}_2$, let $\mathcal{E}'_i$ be the event that after the i^{th} query made by $\mathcal{A}_2$, there exists no collision between any query pair and a prefixed pair or among the query pairs themselves; the corresponding conditions are analogous to (1). Consider the probability $\mathsf{P}[\neg\mathcal{E}'_i|\mathcal{E}'_{i-1},\mathcal{E}]$. If the i^{th} query is a forward query, then a collision occurs only if $y_i = a(k_i - k_j) + b(x_i - x_j) + y_j$ for some $j < i$ or if the analogous condition holds for a collision with a prefixed pair and some $j \in \{1,\ldots,P\}$; if the i^{th} query is a backward query, then a collision occurs only if

$$x_i \;=\; \frac{a(k_i - k_j) - (y_j - y_i)}{b}$$

for some $j < i$ or if the analogous condition holds for a collision with a prefixed pair and some $j \in \{1,\ldots,P\}$ (using that $b \neq 0$). In either case,

$$\mathsf{P}[\neg\mathcal{E}'_i|\mathcal{E}'_{i-1},\mathcal{E}] \;\leq\; \frac{(i-1)+P}{N-(T'+P)} \;\leq\; \frac{2((i-1)+P)}{N}\;,$$

using that $T' + P \leq N/2$, an assumption on may always make since, otherwise, the desired bound holds trivially. Summarizing, setting $\mathcal{E}' := \mathcal{E}'_{T'}$,

$$\begin{aligned}\mathsf{P}[\neg\mathcal{E}'|\mathcal{E}] \;=\; \mathsf{P}[\neg\mathcal{E}'_{T'}|\mathcal{E}] &\leq \sum_{i=1}^{T'} \mathsf{P}[\neg\mathcal{E}'_i|\mathcal{E}'_{i-1},\mathcal{E}] \\ &\leq \sum_{i=1}^{T'} \frac{2((i-1)+P)}{N} \\ &= O\left(\frac{T^2}{N} + \frac{TP}{N}\right)\;.\end{aligned}$$

Clearly, $\mathcal{A}_2$ only wins if $\mathcal{E}$ or $\mathcal{E}'$ occurs, and hence the overall security in the $\mathsf{BF\text{-}IC}(P,N,N)$-model is $O\left(\frac{T^2}{N} + \frac{P(P+T)}{N}\right)$. □

4.4 CRHFs from Unkeyed Sponges

The application $G^{\mathsf{CRHF\text{-}S}}$ of collision resistance for the sponge construction is defined via the following challenger $\mathsf{C}^{\mathsf{CRHF\text{-}S}}$: it picks an initialization vector $\mathsf{IV} \leftarrow \{0,1\}^c$ uniformly at random, passes it to the attacker, and outputs 1 if and only if the attacker returns two messages $m \neq m'$ such that $\mathsf{Sponge}_{\pi,\mathsf{IV}}(m) = \mathsf{Sponge}_{\pi,\mathsf{IV}}(m')$.

The following theorem provides an upper bound on the probability that an (S,T,ℓ)-attacker finds a collision of the sponge construction in the AI-RPM, where ℓ is an upper bound on the lengths of the messages m and m' the attacker submits to the challenger. The proof follows the approach by Bertoni *et al.* [6].

Theorem 5. *Application* $G^{\mathsf{CRHF\text{-}S}}$ *is* $\left((S,T,\ell),\tilde{O}\left(\frac{S(T+\ell)^2}{2^c} + \frac{(T+\ell)^2}{2^r}\right)\right)$*-secure in the* $\mathsf{AI\text{-}RP}(N)$*-model, for* $N = 2^n = 2^{r+c} \geq 16$.

Node graphs. A useful formalism for security proofs of sponge-based constructions is that of node and supernode graphs, as introduced by Bertoni *et al.* [6]. For a permutation $\pi : \{0,1\}^n \to \{0,1\}^n$, consider the following (directed) *node graph* $G_\pi = (V, E)$ with $V = \{0,1\}^r \times \{0,1\}^c = \{0,1\}^n$ and $E = \{(s,t) \mid \pi(s) = t\}$. Moreover, let $G'_\pi = (V', E')$ be the (directed) *supernode graph*, with $V' = \{0,1\}^c$ and $(s^{(2)}, t^{(2)}) \in E'$ iff $((s^{(1)}, s^{(2)}), (t^{(1)}, t^{(2)})) \in E$ for some $s^{(1)}, t^{(1)} \in \{0,1\}^r$. Observe that the value of $\mathsf{Sponge}_{\pi,\mathsf{IV}}(m)$ for an ℓ-block message $m = m_1 \cdots m_\ell$ is obtained by starting at $s_0 := (0^r, \mathsf{IV}) \in \{0,1\}^n$ in G_π, moving to $s_i \leftarrow \pi(m_i \oplus s_{i-1}^{(1)} \| s_{i-1}^{(2)})$ for $i = 1, \ldots, \ell$, and outputting $s_\ell^{(1)}$. In other words, in the supernode graph, m corresponds to a path of length ℓ starting at node IV and ending at $s_\ell^{(2)}$, and $s_1^{(1)}, \ldots, s_\ell^{(1)} \in \{0,1\}^r$ are the values that *appear* on that path.

Proof. At the cost of at most 2ℓ additional queries to π, assume that the messages m and m' output by $\mathcal{A}_2$ are such that $\mathcal{A}_2$ has queried its oracle π on all points $\mathsf{Sponge}_{\pi,\mathsf{IV}}(\cdot)$ would query π when evaluated on m and m'.

It suffices to show that $G^{\mathsf{CRHF\text{-}S}}$ is $\left((S,T,\ell), O\left(\frac{(T+\ell)^2}{2^r} + \frac{(T+\ell)^2+(T+\ell)P)}{2^c}\right)\right)$-secure in the $\mathsf{BF\text{-}RP}(P,N)$-model. Then, by observing that $T^{\mathsf{comb}}_{G^{\mathsf{CRHF\text{-}S}}} = T + 2\ell$, setting $\gamma := 1/N$ and $P := 2(S + \log N)(T + \ell) = \tilde{O}\left(S(T+\ell)\right)$, and applying Theorem 1, the desired conclusion follows.

Consider now an interaction of $\mathcal{A}_2$ with $\mathsf{C}^{\mathsf{CRHF\text{-}S}}$ and incrementally build the node and supernode graphs (as defined above), adding edges when $\mathcal{A}_2$ makes the corresponding (forward or inverse) query to π, and starting with the edges that correspond to the at most P prefixed query/answer pairs.

Let $\mathcal{E}_{\mathsf{coll}}$ be the event that a (valid) collision occurs. Clearly, this happens if and only if there exists a value $s^{(1)} \in \{0,1\}^r$ that appears as the last value on two different paths from IV. Let $\mathcal{E}_{\mathsf{path,i}}$ be the event that after the i^{th} query to π, there is a unique path from IV to any node in the supernode graph and that no prefixed supernode is reachable from IV.

Observe that when $\mathcal{E}_{\mathsf{path}} := \mathcal{E}_{\mathsf{path},\mathsf{T}+2\ell}$ occurs, the values that appear on these paths are uniformly random and independent since every node inside a supernode has the same probability of being chosen. Hence,

$$\mathsf{P}[\mathcal{E}_{\mathsf{coll}} | \mathcal{E}_{\mathsf{path}}] \;\le\; \binom{T+2\ell}{2} \cdot 2^{-r} \;=\; O\left(\frac{(T+\ell)^2}{2^r}\right).$$

Moreover,

$$\mathsf{P}[\neg\mathcal{E}_{\mathsf{path,i}} | \mathcal{E}_{\mathsf{path,i-1}}] \;\le\; \frac{(i+P)\cdot 2^r}{2^{r+c} - (i-1+P)} \;\le\; \frac{i+P}{2^c - (T+2\ell+P)/2^r} \;\le\; \frac{i+P}{2^{c-1}}$$

if the i^{th} query is a forward query, and

$$\mathsf{P}[\neg\mathcal{E}_{\mathsf{path,i}} | \mathcal{E}_{\mathsf{path,i-1}}] \;\le\; \frac{i\cdot 2^r}{2^{r+c} - (i-1+P)} \;\le\; \frac{i}{2^{c-1}}$$

if the i^{th} query is an inverse query, using that $T + 2\ell + P \leq N/2$, an assumption one may always make since, otherwise, the lemma holds trivially. Letting $T' := T + 2\ell$,

$$\begin{aligned}\mathsf{P}[\neg\mathcal{E}_{\mathsf{path}}] \;=\; \mathsf{P}[\neg\mathcal{E}_{\mathsf{path},\mathsf{T}'}] &\leq \mathsf{P}[\neg\mathcal{E}_{\mathsf{path},\mathsf{T}'}|\mathcal{E}_{\mathsf{path},\mathsf{T}'-1}] + \mathsf{P}[\neg\mathcal{E}_{\mathsf{path},\mathsf{T}'-1}] \\ &\leq \sum_{i=1}^{T'} \mathsf{P}[\neg\mathcal{E}_{\mathsf{path},\mathsf{i}}|\mathcal{E}_{\mathsf{path},\mathsf{i}-1}] + \mathsf{P}[\mathcal{E}_{\mathsf{path},0}] \\ &\leq \sum_{i=0}^{T'} \frac{(i+P)}{2^{c-1}} \;=\; O\left(\frac{(T+\ell)(T+\ell+P)}{2^{-c}}\right),\end{aligned}$$

observing that $\mathsf{P}[\mathcal{E}_{\mathsf{path},0}] \leq \frac{P}{2^c}$, the probability that a node inside supernode IV is prefixed. □

4.5 PRFs via NMAC with Davies-Meyer

For simplicity, let $K = N$. Recall that the NMAC construction using the Davies-Meyer compression function is defined as

$$\mathsf{NMAC}_{E,k}(\overline{m}) \;:=\; \mathsf{DM}_E(k_1, \mathsf{MD\text{-}DM}_{E,k_2}(\overline{m}))$$

where $k = (k_1, k_2)$.

The application $G^{\mathsf{PRF\text{-}MD\text{-}N}}$ of PRF security for NMAC is defined via the following challenger $\mathsf{C}^{\mathsf{PRF\text{-}MD\text{-}N}}$: it picks a random bit $b \leftarrow \{0,1\}$ and a key $k \leftarrow [N]$; when the attacker queries a message $m = m_1 \cdots m_\ell$ consisting of blocks m_i, if $b = 0$, the challenger answers by $\mathsf{NMAC}_{E,k}(\overline{m})$, and, if $b = 1$, the challenger answers by a value chosen uniformly at random for each $\overline{m}$. The attacker wins, if and only if he correctly guesses the bit b.

The following theorem provides an upper bound on the advantage of an (S, T, q, ℓ)-attacker in distinguishing the sponge construction from a random function in the AI-ICM, where q is an upper bound on the number of messages $\overline{m}$ the attacker submits to the challenger and ℓ is an upper bound on the length of those messages.

Theorem 6. *$G^{\mathsf{PRF\text{-}MD\text{-}N}}$ is $\left((S, T, q, \ell), \tilde{O}\left(\frac{Tq^2\ell}{N} + \sqrt{\frac{S(T+\ell q)q^2\ell}{N}}\right)\right)$-secure in the $\mathsf{AI\text{-}IC}(N, N)$-model, for $N \geq 16$.*

Proof. Follows from Lemma 1 by observing that $T^{\mathsf{comb}}_{G^{\mathsf{PRF\text{-}MD\text{-}N}}} = T + q\ell$, setting $\gamma := 1/N$ and $P := \sqrt{\frac{S(T+q\ell)N}{q^2\ell}}$, and applying Theorem 1. □

Lemma 1. *For any $P, N \in \mathbb{N}$, $G^{\mathsf{PRF\text{-}MD\text{-}N}}$ is $\left((S, T, q, \ell), O\left(q^2\ell\frac{T+P}{N}\right)\right)$-secure in the $\mathsf{BF\text{-}IC}(P, N, N)$-model.*

The proof of Lemma 1 uses the fact the Merkle-Damgård construction with the DM function is *almost-universal* in the BF-ICM; this property is captured by

the application $G^{\mathsf{AU\text{-}MD}}$ defined by the following challenger $\mathsf{C}^{\mathsf{AU\text{-}MD}}$: It expects $\mathcal{A}_2$ to submit two messages $\overline{m}$ and $\overline{m}'$. Then, it picks a random key k. The attacker wins if $\mathsf{MD\text{-}DM}_{E,k}(\overline{m}) = \mathsf{MD\text{-}DM}_{E,k}(\overline{m}')$.

The proof of almost-universality uses the fact that the DM function is a PRF when keyed by h (cf. full version of this paper).

Lemma 2. *For any $P, N \in \mathbb{N}$, $G^{\mathsf{AU\text{-}MD}}$ is $\left((S,T,q,\ell), O\left(\ell\frac{T+P}{N}\right)\right)$-secure in the $\mathsf{BF\text{-}IC}(P,N,N)$-model.*

Proof (sketch). Consider a sequence of ℓ hybrid experiments, where in the i^{th} hybrid, instead of evaluating $\mathsf{MD\text{-}DM}_{E,k}(\overline{m})$ for $\overline{m} = m_1 \cdots m_\ell$, the challenger computes $\mathsf{MD\text{-}DM}_{E,k'}(m_{i+1} \cdots m_\ell)$, where $k' \leftarrow f(m_1 \cdots m_i)$ for a uniformly random function $f : [N]^i \to N$. By the PRF security of the Davies-Meyer function, the distance between successive hybrids is at most $8(T+P)/N$. Moreover, in the last hybrid, the success probability of $\mathcal{A}_2$ is at most $1/N$. □

Proof (of Lemma 1, sketch). Using the PRF security of the Davies-Meyer (DM) function, it suffices to show security in the hybrid experiment in which the outer DM evaluation is replaced by a uniform random function f. In this hybrid experiment, $\mathcal{A}_2$ only has non-zero advantage in guessing bit b if two of its q queries to the challenger cause a collision right before f. Let ε be the probability that this event occurs.

Consider the following attacker $\mathcal{A}' := (\mathcal{A}_1, \mathcal{A}_2')$ against the $\mathsf{C}^{\mathsf{AU\text{-}MD}}$: $\mathcal{A}_2'$ runs $\mathcal{A}_2$ internally, forwarding its oracle queries to and back from its own oracle, and answering every query $\mathcal{A}_2$ would make to its challenger by a fresh uniformly random value. Once $\mathcal{A}_2$ terminates, $\mathcal{A}_2'$ picks a pair of queries made by $\mathcal{A}_2$ uniformly at random and submits it to its own challenger. It is easily seen that the advantage of $\mathcal{A}_2'$ is at least ε/q^2. Therefore, the final PRF security of NMAC is $q^2\ell(T+P)/N$. □

4.6 MACs via NMAC with Davies-Meyer

The application $G^{\mathsf{MAC\text{-}MD\text{-}N}}$ of MAC security of the NMAC construction is defined via the following challenger $\mathsf{C}^{\mathsf{MAC\text{-}MD\text{-}N}}$: it initially picks a random key $k \leftarrow [N]$; when the attacker queries a message $\overline{m} = m_1 \cdots m_\ell$ consisting of blocks m_i, the challenger answers by $\mathsf{MD\text{-}DM}_{\mathcal{O},k}(\overline{m})$. The attacker wins if he submits a pair $(\overline{m}, y)$ with $\mathsf{MD\text{-}DM}_{\mathcal{O},k}(\overline{m}) = y$ for a previously unqueried $\overline{m}$.

Theorem 7. *$G^{\mathsf{MAC\text{-}MD\text{-}N}}$ is $\left((S,T,q,\ell), \tilde{O}\left(q^2\ell\frac{S(T+q\ell)}{N}\right)\right)$-secure in the $\mathsf{AI\text{-}IC}(N)$-model, for $N \geq 16$.*

Proof. It suffices to show that $G^{\mathsf{MAC\text{-}MD\text{-}N}}$ is $\left((S,T,q,\ell), O\left(q^2\ell\frac{T+P}{N}\right)\right)$-secure in the $\mathsf{BF\text{-}IC}(P,N)$-model. Then, by observing that $T^{\mathsf{comb}}_{G^{\mathsf{MAC\text{-}MD\text{-}N}}} = T + q\ell$, setting $\gamma := 1/N$ and $P = 2(S + \log N)(T + q\ell) = \tilde{O}\left(S(T+q\ell)\right)$ and applying Theorem 1, the desired conclusion follows.

The bound in the $\mathsf{BF\text{-}IC}(P,N)$-model follows immediately from Lemma 1 and the fact that with a truly random function, the adversary's success probability at breaking the MAC is at most q/N. □

4.7 Extensions to HMAC

Recall that, for simplicity, $K = N$. The HMAC construction using the Davies-Meyer compression function is defined as

$$\mathsf{HMAC}_{E,k}(\overline{m}) \;:=\; \mathsf{MD\text{-}DM}_{E,\mathsf{IV}}(k \oplus \mathsf{opad}, \mathsf{MD\text{-}DM}_{E,\mathsf{IV}}(k \oplus \mathsf{ipad}, \overline{m}))\ ,$$

where $\mathsf{IV} \in [N]$ is some fixed initialization vector. As usual, results for NMAC carry over to HMAC, even in the presence of leakage about the ideal cipher. More precisely, the HMAC construction can be seen as a special case of the NMAC by observing that

$$\mathsf{HMAC}_{E,k}(\overline{m}) \;=\; \mathsf{NMAC}_{E,k_1,k_2}(\overline{m})$$

for $k_1 = E(k \oplus \mathsf{ipad}, \mathsf{IV}) \oplus \mathsf{IV}$ and $k_2 = E(k \oplus \mathsf{opad}, \mathsf{IV}) \oplus \mathsf{IV}$. Hence, in the BF-IC-model, unless $(k \oplus \mathsf{opad}, \mathsf{IV})$ or $(k \oplus \mathsf{ipad}, \mathsf{IV})$ are prefixed by $\mathcal{A}_1$ or queried by $\mathcal{A}_2$, which happens with probability $O\left((T+P)/N\right)$, the NMAC analysis applies.

5 The Generic-Group Model with Preprocessing

This section analyzes the hardness of various problems in the generic-group model (GGM) with preprocessing. Specifically, the following applications are considered, where $N \in \mathbb{N}$ is an arbitrary prime and σ the random injection used in the GGM:

- *Discrete-logarithm problem (DL):* Given $\sigma(x)$ for a uniformly random $x \in [N]$, find x.
- *Multiple-discrete-logarithms problem (MDL):* Given $(\sigma(x_1), \ldots, \sigma(x_t))$ for uniformly random and independent $x_i \in [N]$, find $(x_1, \ldots, x_t)$.
- *Computational Diffie-Hellman problem (CDH):* Given $(\sigma(x), \sigma(y))$ for uniformly random and independent $x, y \in [N]$, find xy.
- *Decisional Diffie-Hellman problem (DDH):* Distinguish $(\sigma(x), \sigma(y), \sigma(xy))$ from $(\sigma(x), \sigma(y), \sigma(z))$ for uniformly random and independent $x, y, z \in [N]$.
- *Square decisional Diffie-Hellman problem (sqDDH):* Distinguish $(\sigma(x), \sigma(x^2))$ from $(\sigma(x), \sigma(y))$ for uniformly random and independent $x, y \in [N]$.
- *One-more-discrete-logarithm problem (OM-DL):* Given access to an oracle creating DL challenges $\sigma(x_i)$, for uniformly random and independent $x_i \in [N]$, as well as a DL oracle, make t queries to the challenge oracle and at most $t-1$ queries to the DL oracle, and solve *all* t challenges, i.e., find $(x_1, \ldots, x_t)$.
- *Knowledge-of-exponent assumption (KEA):* The KEA assumption states that if an attacker $\mathcal{A}$ is given $\sigma(x)$, for $x \in [N]$ chosen uniformly at random, and outputs A and $\hat{A}$ with $A = \sigma(a)$ and $\hat{A} = \sigma(ax)$, then it must know discrete logarithm a of A. This is formalized by requiring that for every $\mathcal{A}$ there exist an *extractor* $\mathcal{X}_{\mathcal{A}}$ that is run on the same random coins as $\mathcal{A}$ and must output the value a.

The asymptotic security bounds derived for the above applications are summarized in Table 2. The bounds for DL, MDL, CDH, DDH, and sqDDH match previously known bounds from [5,15,34]; they are tight in that there is a matching attack, except for the DDH problem, for which, remarkably, closing the gap remains an open problem. The bounds for OM-DL and KEA are new.

Note that all bounds with preprocessing are considerably worse than those without. For example, in the classical GGM, DL is secure up to roughly $N^{1/2}$ queries, whereas it becomes insecure for $S = T = N^{1/3}$ in the AI-GGM.

All security bounds are derived by following the bit-fixing approach: the security of a particular application is assessed in the *bit-fixing (BF)* GGM, and then Theorem 1 is invoked to obtain a corresponding bound in the *auxiliary-input (AI)* GGM. This approach features great simplicity since deriving security bounds in the BF-GGM turns out to be remarkably straightforward, and all of the proofs closely follow the original proofs in the classical GGM without preprocessing; the only difference is that one needs to take the list $\mathcal{L}$ of the at most P input/output pairs where $\mathcal{A}_1$ fixes σ into account.

Besides simplicity, another advantage of the bit-fixing methodology is applicability: using bit-fixing, in addition to recovering all of the bounds obtained in [15] via much more involved compression-based proofs, one also easily derives bounds for applications that may be challenging to derive using compression-based proofs, such as, e.g., the knowledge-of-exponent assumption.

As representative examples, the proofs for the DL problem and the KEA are provided below. Readers familiar with the original proofs by Shoup [44] for DL and by Abe and Fehr [1] and Dent [17] for the KEA may immediately observe the similarity. The precise definitions of the remaining applications as well as the corresponding theorems and proofs can be found in the full version of this paper.

5.1 Discrete Logarithms

The discrete-logarithm application G^{DL} is defined via the challenger C^{DL} that randomly and uniformly picks an $x \in [N]$, passes $\sigma(x)$ to the attacker, and outputs 1 if and only if the attacker returns x.

Theorem 8 below provides an upper bound on the success probability of any attacker at computing discrete logarithms in the AI-GGM. The bound is matched by the attack of Mihalcik [34] and Bernstein and Lange [5]; a variation of said attack has recently also been presented by Corrigan-Gibbs and Kogan [15].

Theorem 8. *G^{DL} is $((S,T),\varepsilon)$-secure in the $\mathsf{AI\text{-}GG}(N,M)$-model for any prime $N \geq 16$ and*

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

Proof. It suffices to show that the application G^{DL} is $\left((S,T), O\left(\frac{TP+T^2}{N}\right)\right)$-secure in the $\mathsf{BF\text{-}GG}(P,N,M)$-model. Then, by observing that $T^{\mathsf{comb}}_{G^{\mathsf{DL}}} = T+1$,

setting $\gamma := 1/N$ and $P = 6(S + \log N)(T + 1) = \tilde{O}(ST)$, and applying the second part of Theorem 1, the desired conclusion follows.

Consider now the interaction of $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with C^{DL} in the BF-GG-model. Recall that the BF-GG-oracle outputs the range $\mathcal{Y}$ of the underlying random injection σ to $\mathcal{A}_1$ via interface pre. Condition on a particular realization of this set for the remainder of the proof.

Define the following hybrid experiment involving $\mathcal{A}_1$ and $\mathcal{A}_2$:

- For each of the at most P query/answer pairs (a', s') where $\mathcal{A}_1$ fixes σ, define a (constant) polynomial $v(X) := a'$ and store the pair (v, s').
- To create the challenge, choose a value s^* uniformly at random from all unused values in $\mathcal{Y}$, define the polynomial $u^*(X) := X$, and store (u^*, s^*).
- A forward query a by $\mathcal{A}_2$ to BF-GG is answered as follows: define the (constant) polynomial $u(X) := a$, choose a value s uniformly at random from all unused values in $\mathcal{Y}$, store the pair (u, s), and return s.
- A group-operation query (s_1, s_2) by $\mathcal{A}_2$ is answered as follows:
  - If s_1 or s_2 is not in $\mathcal{Y}$, return $\perp$.
  - If s_1 has not been recorded, choose a random unused $a \in [N]$, define the (constant) polynomial $u(X) := a$, and store the pair (u, a). Proceed similarly if s_2 has not been recorded. Go to the next item.
  - Let u_1 and u_2 be the polynomials recorded with s_1 and s_2, respectively. If, for $u' := u_1 + u_2$, a pair (u', s') has been recorded, return s'. Otherwise, choose a value s' uniformly at random from all unused values in $\mathcal{Y}$, store the pair (u', s'), and return s'.
- When $\mathcal{A}_2$ outputs a value x', pick a value $x \in [N]$ uniformly at random and output 1 if and only if $x' = x$.

Observe that the hybrid experiment only differs from the original one if for a group-operation query (s_1, s_2), $u'(x) = v(x)$ for some recorded v or $u'(x) = u(x)$ for some recorded u—and similarly for the polynomial u^* corresponding to the challenge. Since in the hybrid experiment, x is chosen uniformly at random *at the end* of the execution, the probability of this event is at most $((T + 1)P + (T + 1)^2)/N$ by the Schwartz-Zippel Lemma and a union bound. Moreover, in the hybrid experiment, the probability that $x' = x$ is $1/N$. The theorem follows. □

5.2 Knowledge-of-Exponent Assumption

Informally, the knowledge-of-exponent assumption (KEA) states that if an attacker $\mathcal{A}$ is given (h, h^x), for a generator h of a cyclic group of order N and $x \in [N]$ chosen uniformly at random, and outputs group elements A and $\hat{A}$ with $\hat{A} = A^x$, then it must know discrete logarithm a of A. This is formalized by requiring that for every $\mathcal{A}$ there exist an *extractor* $\mathcal{X}_{\mathcal{A}}$ that is run on the same random coins as $\mathcal{A}$ and must output the value a.

The above is captured in the GGM by considering the following experiment $\mathrm{Exp}^{\mathcal{O}}_{\mathcal{A},\mathcal{X}_{\mathcal{A}}}$ parameterized by an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, an extractor $\mathcal{X}_{\mathcal{A}}$, and an oracle $\mathcal{O} \in \{\mathsf{AI\text{-}GG}(N, M), \mathsf{BF\text{-}GG}(N, M)\}$:

1. Run $\mathcal{A}_1$ to obtain $z \leftarrow \mathcal{A}_1^{\mathcal{O}}$.
2. Choose $x \in [N]$ uniformly at random, let $y \leftarrow \sigma(x)$, pick random coins ρ, and run
   (a) $\mathcal{A}_2$ to get $(A, \hat{A}) \leftarrow \mathcal{A}_2(z, y; \rho)$, and
   (b) $\mathcal{X}_{\mathcal{A}}$ to get $a \leftarrow \mathcal{X}_{\mathcal{A}}(z, y; \rho)$.
3. Output 1 if and only if $A = \sigma(a')$ and $\hat{A} = \sigma(a'x)$ for some a', but $a \neq a'$.

The KEA says that for every attacker $\mathcal{A}$ there exists an extractor $\mathcal{X}_{\mathcal{A}}$ such that the probability of the above experiment outputting 1 is negligible. The following theorem is equivalent to saying that the KEA holds in the AI-GGM.

Theorem 9. *For every attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists an extractor $\mathcal{X}_{\mathcal{A}}$ such that*

$$\mathsf{P}[\mathrm{Exp}^{\mathcal{O}}_{\mathcal{A},\mathcal{X}_{\mathcal{A}}} = 1] \;\leq\; \tilde{O}\left(\frac{ST^2}{N}\right).$$

Proof (Sketch). The extractor $\mathcal{X}_{\mathcal{A}}$ internally runs $\mathcal{A}_2$ on the inputs received and keeps track of $\mathcal{A}_2$'s oracle queries using polynomials as in the proof of Theorem 8. If at the end the polynomials u_A and $u_{\hat{A}}$ corresponding to $\mathcal{A}_2$'s outputs $(A, \hat{A})$ have the form $u_A(X) = a$ and $u_{\hat{A}}(X) = aX$, then $\mathcal{X}_{\mathcal{A}}$ outputs a and otherwise $\perp$.

Observe that if the experiment outputs 1, then

- $u_{\hat{A}} \neq X \cdot u_A$ since $\mathcal{A}_2$ only creates polynomials of degree at most 1, but
- $u_{\hat{A}}(x) = x \cdot u_A(x)$ for the challenge x.

Hence, the extractor only fails if at least two of the polynomials involved (including $u_{\hat{A}}$ and $X \cdot u_A$) collide on x, which is already analyzed in the proof of Theorem 8.

The experiment $\mathrm{Exp}^{\mathcal{O}}_{\mathcal{A},\mathcal{X}_{\mathcal{A}}}$ defining KEA does not exactly match the syntax of challenger and attacker to which Theorem 1 caters, but it is easily checked that the corresponding proof can be adapted to fit $\mathrm{Exp}^{\mathcal{O}}_{\mathcal{A},\mathcal{X}_{\mathcal{A}}}$. □

6 Computationally Secure Applications

A main advantage of the pre-sampling methodology over other approaches (such as compression) to dealing with auxiliary-input in idealized models is that it also applies to applications that rely on computational hardness assumptions. To illustrate this fact, this section considers a public-key encryption scheme based on trapdoor functions by Phan and Pointcheval [39] in the auxiliary-input random-permutation model (AI-RPM). Other schemes in the AI-RPM/ICM, e.g., [29,31], can be analyzed similarly.

FDP encryption. Let F be a trapdoor family (TDF) generator. Full-domain permutation (FDP) encryption in the random-permutation model with oracle $\mathcal{O}$ is defined 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 $\mathsf{sk} := f^{-1}$.

– *Encryption:* To encrypt a message m with randomness r and public key $\mathsf{pk} = f$, compute $\tilde{y} \leftarrow f(y)$ for $y \leftarrow \mathcal{O}(m\|r)$) and output $c = \tilde{y}$.
– *Decryption:* To decrypt a ciphertext $c = y$ with secret key $\mathsf{sk} = f^{-1}$, compute $m\|r \leftarrow \mathcal{O}^{-1}(f^{-1}(y))$ and output m.

The following theorem relates to the CPA security of FDP encryption in the AI-RPM.

Theorem 10. *Let Π be FDP encryption with F. If $G^{\mathsf{TDF},F}$ is $((S', *, t', s'), \varepsilon')$-secure, then, for any $T \in \mathbb{N}$, $G^{\mathsf{PKE},\Pi}$ is $((S, T, t, s), \varepsilon)$-secure in the $\mathsf{AI\text{-}RP}(N, N)$-model, where*

$$\varepsilon \;=\; \tilde{O}\left(\varepsilon' + \sqrt{\frac{ST}{2^{\rho}}}\right)$$

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.

The straightforward approach to proving the security of FDP encryption in the AI-RPM would be to analyze the scheme in the BF-RPM with list size P and then use the general part of Theorem 1 to obtain a bound in the AI-RPM. However, such an approach, due to the additive error in the order of ST/P would require a very large list and therefore make the reduction to TDF security extremely loose.

Instead, the actual proof, which is sketched in the full version of this paper, follows the same high-level structure as that of TDF encryption in the AI-ROM, analyzed in [14]:

1. It first considers a hybrid experiment that is only distinguishable from the original CPA experiment if the attacker queries a particular value to the random permutation. To bound the probability of this event occurring, the proof moves to the BF-RPM and the analysis there—which involves the reduction to TDF security—is carried back to the AI-RPM via the *unpredictability* part of Theorem 1. This allows the list size to remain a moderate $P' \approx ST$ and hence for a tight reduction.
2. To analyze the advantage of the attacker in the hybrid experiment, the BF-RPM is used again, but using the general part of Theorem 1, which requires a larger list size P. However, since this second step involves no reduction to TDF security and is purely information-theoretic, this does not pose a problem.

Acknowledgments. The authors thank Dan Boneh, Henry Corrigan-Gibbs, and Dmitry Kogan for valuable discussions on pre-processing in generic-group models.

References

1. Abe, M., Fehr, S.: Perfect NIZK with adaptive soundness. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 118–136. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70936-7_7
2. Andreeva, E., Bogdanov, A., Dodis, Y., Mennink, B., Steinberger, J.P.: On the indifferentiability of key-alternating ciphers. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 531–550. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_29
3. Bellare, M., Boldyreva, A., Palacio, A.: An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 171–188. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_11
4. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68697-5_1
5. Bernstein, D.J., Lange, T.: Non-uniform cracks in the concrete: the power of free precomputation. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 321–340. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42045-0_17
6. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the indifferentiability of the sponge construction. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 181–197. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_11
7. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the security of the keyed sponge construction. In: Symmetric Key Encryption Workshop (SKEW) (2011)
8. Black, J.: The ideal-cipher model, revisited: an uninstantiable blockcipher-based hash function. In: Robshaw, M. (ed.) FSE 2006. LNCS, vol. 4047, pp. 328–340. Springer, Heidelberg (2006). https://doi.org/10.1007/11799313_21
9. Black, J., Rogaway, P., Shrimpton, T.: Black-box analysis of the block-cipher-based hash-function constructions from PGV. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 320–335. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_21
10. Canetti, R., Goldreich, O., Halevi, S.: On the random-oracle methodology as applied to length-restricted signature schemes. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 40–57. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24638-1_3
11. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. ACM **51**(4), 557–594 (2004)
12. Chen, S., Steinberger, J.: Tight security bounds for key-alternating ciphers. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 327–350. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_19
13. Chung, K.-M., Lin, H., Mahmoody, M., Pass, R.: On the power of nonuniformity in proofs of security. In: Innovations in Theoretical Computer Science, ITCS 2013, Berkeley, CA, USA, 9–12 January 2013, pp. 389–400 (2013)
14. Coretti, S., Dodis, Y., Guo, S., Steinberger, J.: Random oracles and non-uniformity. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10820, pp. 227–258. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_9
15. Corrigan-Gibbs, H., Kogan, D.: The discrete-logarithm problem with preprocessing. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10821, pp. 415–447. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78375-8_14

16. De, A., Trevisan, L., Tulsiani, M.: Time space tradeoffs for attacks against one-way functions and PRGs. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 649–665. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_35
17. Dent, A.W.: The hardness of the DHK problem in the generic group model. Cryptology ePrint Archive, Report 2006/156 (2006). https://eprint.iacr.org/2006/156
18. Dodis, Y., Guo, S., Katz, J.: Fixing cracks in the concrete: random oracles with auxiliary input, revisited. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10211, pp. 473–495. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56614-6_16
19. Dodis, Y., Pietrzak, K., Wichs, D.: Key derivation without entropy waste. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 93–110. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_6
20. Dodis, Y., Reyzin, L., Rivest, R.L., Shen, E.: Indifferentiability of permutation-based compression functions and tree-based modes of operation, with applications to MD6. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 104–121. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03317-9_7
21. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) ASIACRYPT 1991. LNCS, vol. 739, pp. 210–224. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57332-1_17
22. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. J. Cryptol. **10**(3), 151–162 (1997)
23. Fouque, P.-A., Joux, A., Mavromati, C.: Multi-user collisions: applications to discrete logarithm, even-mansour and PRINCE. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. LNCS, vol. 8873, pp. 420–438. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45611-8_22
24. Gaži, P., Tessaro, S.: Provably robust sponge-based PRNGs and KDFs. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 87–116. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_4
25. Gennaro, R., Trevisan, L.: 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, pp. 305–313 (2000)
26. Goldreich, O., Krawczyk, H.: On the composition of zero-knowledge proof systems. SIAM J. Comput. **25**(1), 169–192 (1996)
27. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. J. Cryptol. **7**(1), 1–32 (1994)
28. Goldwasser, S., Kalai, Y.T.: On the (in)security of the Fiat-Shamir paradigm. In: Proceedings of the 44th Symposium on Foundations of Computer Science (FOCS 2003), 11–14 October 2003, Cambridge, MA, USA, pp. 102–113 (2003)
29. Granboulan, L.: Short signatures in the random oracle model. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 364–378. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36178-2_23
30. Hellman, M.E.: A cryptanalytic time-memory trade-off. IEEE Trans. Inf. Theory **26**(4), 401–406 (1980)
31. Jonsson, J.: An OAEP variant with a tight security proof. IACR Cryptology ePrint Archive, 2002:34 (2002)
32. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. Chapman and Hall/CRC Press, Boca Raton (2007)

33. Mahmoody, M., Mohammed, A.: On the power of hierarchical identity-based encryption. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 243–272. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_9
34. Mihalcik, J.P.: An analysis of algorithms for solving discrete logarithms in fixed groups. Master's thesis, Naval Postgraduate School, Monterey, California (2010)
35. Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: the non-committing encryption case. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 111–126. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_8
36. Oechslin, P.: Making a faster cryptanalytic time-memory trade-off. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 617–630. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_36
37. National Institute of Standards and Technology (NIST): FIPS 202. SHA-3 standard: permutation-based hash and extendable-output functions. Technical report, US Department of Commerce, April 2014
38. National Institute of Standards and Technology (NIST): FIPS 180-4. Secure hash standard. Technical report, US Department of Commerce, August 2015
39. Phan, D.H., Pointcheval, D.: Chosen-ciphertext security without redundancy. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 1–18. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40061-5_1
40. Rivest, R.L.: The MD5 Message-Digest algorithm (RFC 1321). http://www.ietf.org/rfc/rfc1321.txt?number=1321
41. Rivest, R.L., et al.: The MD6 hash function: a proposal to NIST for SHA-3 (2008)
42. Rogaway, P., Steinberger, J.: Constructing cryptographic hash functions from fixed-key blockciphers. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 433–450. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_24
43. National Technical Information Service: FIPS 180-1. Secure hash standard. Technical report, US Department of Commerce, April 1995
44. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-69053-0_18
45. Shrimpton, T., Stam, M.: Building a collision-resistant compression function from non-compressing primitives. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 643–654. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70583-3_52
46. Tessaro, S.: Security amplification for the cascade of arbitrarily weak PRPs: tight bounds via the interactive hardcore lemma. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 37–54. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_3
47. Unruh, D.: Random oracles and auxiliary input. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 205–223. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74143-5_12
48. Winternitz, R.S.: A secure one-way hash function built from DES. In: Proceedings of the 1984 IEEE Symposium on Security and Privacy, Oakland, California, USA, 29 April–2 May 1984, pp. 88–90 (1984)