On the Bit Security of Cryptographic Primitives*†

Daniele Micciancio‡ Michael Walter§

May 27, 2019

Abstract

We introduce a formal quantitative notion of "bit security" for a general type of cryptographic games (capturing both decision and search problems), aimed at capturing the intuition that a cryptographic primitive with k-bit security is as hard to break as an ideal cryptographic function requiring a brute force attack on a k-bit key space. Our new definition matches the notion of bit security commonly used by cryptographers and cryptanalysts when studying search (e.g., key recovery) problems, where the use of the traditional definition is well established. However, it produces a quantitatively different metric in the case of decision (indistinguishability) problems, where the use of (a straightforward generalization of) the traditional definition is more problematic and leads to a number of paradoxical situations or mismatches between theoretical/provable security and practical/common sense intuition. Key to our new definition is to consider adversaries that may explicitly declare failure of the attack. We support and justify the new definition by proving a number of technical results, including tight reductions between several standard cryptographic problems, a new hybrid theorem that preserves bit security, and an application to the security analysis of indistinguishability primitives making use of (approximate) floating point numbers. This is the first result showing that (standard precision) 53-bit floating point numbers can be used to achieve 100-bit security in the context of cryptographic primitives with general indistinguishability-based security definitions. Previous results of this type applied only to search problems, or special types of decision problems.

1 Introduction

It is common in cryptography to describe the level of security offered by a (concrete instantiation of a) cryptographic primitive P by saying that P provides a certain number of *bits of security*. E.g., one may say that AES offers 110-bits of security as a pseudorandom permuation [6], or that a certain lattice based digital signature scheme offers at least 160-bits of security for a given setting of the parameters. While there is no universally accepted, general, formal definition of bit security, in many cases cryptographers seem to have an intuitive (at least approximate) common understanding of what "n bits of security" means: any attacker that successfully breaks the cryptographic primitive must incur a cost[1] of at least $T > 2^n$, or, alternatively, any efficient attack achieves at most $\epsilon < 2^{-n}$ success probability, or, perhaps, a combination of these two conditions, i.e., for any attack with cost T and success probability ϵ, it must be $T/\epsilon > 2^n$. The intuition is that 2^n is the cost of running a brute force attack to retrieve an n-bit key, or the inverse success probability

*Research supported in part by the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under the SafeWare program. Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views, position or policy of the Government. The second author was also supported by the European Research Council, ERC consolidator grant (682815 - TOCNeT).

 The version published by Springer-Verlag is available at `https://doi.org/10.1007/978-3-319-78381-9_1`.

‡UC San Diego, USA. E-mail: `daniele@cs.ucsd.edu`

§IST Austria, Austria. E-mail: `michael.walter@ist.ac.at`

[1]For concreteness, the reader may think of the cost as the running time of the attack, but other cost measures are possible, and everything we say applies to any cost measure satisfying certain general closure properties, like the fact that the cost of repeating an attack k times is *at most* k times as large as the cost of a single execution.

of a trivial attack that guesses the key at random. In other words, n bits of security means "as secure as an idealized perfect cryptographic primitive with an n-bit key".

The appeal and popularity of the notion of bit security (both in theory and in practice) rests on the fact that it nicely sits in between two extreme approaches:

- The *foundations of cryptography* asymptotic approach (e.g., see [10, 9]) which identifies feasible adversaries with polynomial time computation, and successful attacks with breaking a system with non-negligible probability.
- The *concrete security* approach [3, 5], which breaks the adversarial cost into a number of different components (running time, oracle queries, etc.), and expresses, precisely, how the adversary's advantage in breaking a cryptographic primitive depends on all of them.

The foundational/asymptotic approach has the indubious advantage of simplicity, but it only offers a *qualitative* classification of cryptographic functions into secure and insecure ones. In particular, it does not provide any guidance on choosing appropriate parameters and key sizes to achieve a desired level of security in practice. On the other hand, the concrete security treatment delivers (precise, but) substantially more complex security statements, and requires carefully tracking a number of different parameters through security reductions. In this respect, bit security offers a *quantitative*, yet simple, security metric, in the form of a single number: the *bit security* or *security level* of a primitive, typically understood as the logarithm (to the base 2) of the ratio T/ϵ between the cost T and advantage ϵ of the attack, minimized over all possible adversaries.

Capturing security level with a single number is certainly convenient and useful: it allows for direct comparison of the security level of different instances of the same primitive (or even between different primitives altogether), and it provides a basis for the study of *tight reductions*, i.e., constructions and reductions that approximately preserve the security level. Not surprisingly, bit security is widely used. However, there is no formal definition of this term at this point, but rather just an intuitive common understanding of what this quantity should capture. This understanding has led to some paradoxical situations that suggest that the current "definitions" might not capture exactly what they are meant to.

It has been noted that only considering the adversary's running time is a poor measure of the cost of an attack [7, 8]. This is especially true if moving to the non-uniform setting, where an adversary may receive additional advice, and the question of identifying an appropriate cost measure has been studied before [6]. However, the paradoxical situations have not, to this day, been resolved to satisfaction, and it seems that considering only the adversary's resources is insufficient to address this issue.

In order to explain the problems with the current situation, we first distinguish between two types of primitives with respect to the type of game that defines their security (see Section 3 for a more formal definition): *search primitives* and *decision primitives*. Intuitively, the former are primitives where an adversary is trying to recover some secret information from a large search space, as in a key recovery attack. The latter are games where the adversary is trying to decide if a secret bit is 0 or 1, as in the indistinguishability games underlying the definition of pseudorandom generators or semantically secure encryption. For search games, the advantage of an adversary is usually understood to be the probability of finding said secret information, while for decision games it is usually considered to be the distinguishing advantage (which is equal to the probability that the output of the adversary is correct, over the trivial probability $\frac{1}{2}$ of a random guess).

The Peculiar Case of PRGs Informally, a PRG is a function $f : \{0,1\}^n \mapsto \{0,1\}^m$, where $m > n$, such that its output under uniform input is indistinguishable from the uniform distribution over $\{0,1\}^m$. In the complexity community it is common knowledge according to [8] that a PRG with seed length n cannot provide more than $n/2$ bits of security under the current definition of security level. This is because there are non-uniform attacks that achieve distinguishing advantage $2^{-n/2}$ in very little time against any such function. Such attacks were generalized to yield other time-space-advantage trade-offs in [7]. This is very counter-intuitive, as the best generic seed recovery attacks do not prevent n-bit security (for appropriate cost measure), and thus one would expect n bits of security in such a case to be possible.

The Peculiar Case of Approximate Samplers Many cryptographic schemes, in particular lattice based schemes, involve specific distributions that need to be sampled from during their execution. Furthermore, security reductions may assume that these distributions are sampled exactly. During the transition of such a cryptographic scheme from a theoretical construction to a practical implementation, the question arises as to how such a sampling algorithm should be implemented. In many cases, it is much more efficient or secure (against e.g. side channel attacks) or even only possible to approximate the corresponding distribution rather than generating it exactly. In such a case it is crucial to analyze how this approximation impacts the security of the scheme. Tradionally, statistical distance has been employed to quantify this trade-off between approximation and security guarantee, but it leads to the unfortunate situation where the 53-bit precision provided by floating point numbers (as implemented in hardware in commodity microprocessors) only puts a 2^{-53} bound on statistical distance, and results in a rather weak 53-bit security guarantee on the final application. Proving better security using statistical distance methods seems to require higher precision floating point numbers implemented in (substantially slower) software libraries. In recent years a number of generic results have shown improved analysis methods based on different divergences [16, 2, 17, 15] and using the conventional definition of bit security. Surprisingly, all of them apply exclusively to search primitives (with the only exception of [2], which also considers decision primitives with a specific property). This has led to the unnatural situation where it seems that decision primitives, like encryption, require higher precision sampling than search primitives. This is counter-intuitive, because in search primitives, like signature schemes, the distribution is often used to hide a specific secret and a bad approximation may leak information about it. On the other hand, it is commonly believed within the research community that for encryption schemes the distribution does not necessarily have to be followed exactly, as long as it has sufficient entropy, since none of the cryptanalytic attacks seem to be able to take advantage of a bad approximation in this case [1]. However, a corresponding proof for generic decision primitives (e.g., supporting the use of hardware floating point numbers, while still targeting 100-bit or higher levels of security) has so far eluded the attempts of the research community.

1.1 Contribution and Techniques

We present a new notion of *bit security* associated to a general cryptographic game. Informally, we consider a game in which an adversary has to guess an n-bit secret string[2] x. This captures, in a unified setting, both decision/indistinguishability properties, when $n = 1$, and arbitrary search/unpredictability properties, for larger n. The definition of bit security is quite natural and intuitive, building on concepts from information theory, but we postpone its description to the end of this section. For now, what matters is that a distinguishing feature of our framework is to explicitly allow the adversary to output a special "don't know" symbol $\perp$, rather than a random guess. So, we can talk about the probability α that the adversary outputs something (other than $\perp$), and the (conditional) probability β that the output correctly identifies the secret. This makes little difference for search problems, but for decision problems it allows the adversary to express different degrees of confidence in its guess: admitting failure is more informative than a random guess. We proceed by specializing our notion of bit security to the two important settings of search and decision problems and show that:

- For the case of search primitives (large secret size $n = |x|$), this yields the traditional notion of bit security, as the logarithm of the ratio T/ϵ between the attack cost T, and the success probability $\epsilon = \alpha\beta$. The fact that our definition is consistent with the current one in the case of search primitives gives us confidence in its validity, since in this case the traditional definition is very intuitive and there are no paradoxes casting doubts about it.
- Surprisingly, for decision primitives (i.e., for $n = 1$), our definition yields a different formula, which, instead of being linear the distinguishing advantage $\delta = 2\beta - 1$, is quadratic in δ. In other words, the bit security is the logarithm of $T/(\alpha\delta^2)$. This is not entirely new, as a similar proposal was already put forward in [11, 14] in a more specific context, but has so far received very little attention.

[2]More generally, the adversary has to output a value satisfying a relation $R(x, a)$ which defines successful attacks. For simplicity, in this introduction, we assume R is the identity function. See Definition 5 for the actual definition.

One of the implications of our new definition is that it seemingly resolves the paradoxical situation about the bit security of pseudorandom generators (PRGs) described in [7]. (The significance of the nonuniform attacks to one-way functions described in [7] can already be addressed by an appropriate choice of cost measure.) For the PRG case, an attack achieving distinguishing advantage $\delta = 2^{-n/2}$ even in constant time does not necessarily contradict n-bit security. In fact, [7] shows that for any algorithm distinguishing the output of any function $f : \{0,1\}^n \mapsto \{0,1\}^{n+1}$ from uniform with distinguishing advantage δ must use at least $T = \Omega(\delta^2 2^n)$ resources (for a suitable definition of resources, similar to the one-way function case). So, this shows that by our definition, there exist PRGs with bit security $\log_2(T/\delta^2) = n$, as one would expect.

Of course, as definitions are abitrary, it is not clear if changing a definion is really solving any real problem, and our definition of bit security needs to be properly supported and justified. Notice that a reduction $A \leq B$ showing that if A is n-bit secure, then B is $n/2$-bit secure, may be interpreted in different ways:

- Either the construction of B from A is not optimal/tight, i.e., it incurrs an actual security degradation
- Or the construction is tight, but the reduction (i.e., the security proof) is not
- Or the definition of bit security is incorrect.

The last possibility is most delicate when reducing between different types of cryptographic primitives (e.g., from search to decision) where the definition of bit security may take different (and somehow arbitrary) forms. All these comments apply equally well to tight reductions, mapping n-bit security to n-bit security. We support and justify our definition by providing a collection of results (typically in the form of *tight reductions*[3] between different cryptographic primitives), which are the main technical contribution of this paper. For example,

- We observe that the Goldreich-Levin hard-core predicate is tight according to our definition, i.e., if $f(x)$ is an n-bit secure one-way permutation,[4] then $G(r,x) = (r, f(x), \langle r, x\rangle)$ is an n-bit secure PRG.
- There is a simple reduction showing that if G is an n-bit secure PRG, then the same G (and also f) is an n-bit secure one-way function. (Interestingly, the reduction is not completely trivial, and makes critical use of the special symbol $\perp$ in our definition. See Theorem 4.)

Notice that, while both reductions are between different types of cryptographic primitives (search and decision, with different bit security formulas), combining them together gives a search-to-search reduction which uses the same security definition on both sides. Since it would be quite counterintuitive for such a simple reduction (from PRG to OWF) to increase the level of security from $n/2$ to n bits, this provides some confidence that our definition is on target, and the Goldreich-Levin PRG is indeed as secure as the underlying one-way function.

Other technical results presented in this paper include:

- Approximate samplers: we give a proof in Section 5.3.2 that shows for the first time that the sampling precision requirement is essentially the same for search and decision primitives to maintain security. We do this by extending a result from [15] for search primitives to decision primitives using our definition of bit security.
- Hybrid argument: since our new definition of advantage no longer matches the simple notion of statistical distance, the standard proof of the hybrid argument [12] (so ubiquitously used in cryptography and complexity) is no longer valid. While the proof in our setting becomes considerably more involved, we show (Theorem 7) that hybrid arguments are still valid.

[3] In the context of this work, "tight" means that bit security is (approximately) preserved, up to small additive logarithmic terms corresponding to the polynomial running time of an attack. More specifically, a reduction is tight if it maps a primitive providing n-bit security, to another with security level $n - O(\log n)$. For simplicity, we omit all the $O(\log n)$ in this introduction.

[4] The actual reduction holds for any one-way functions. Here we focus on permutations just to emphasize the connection with PRGs. See Theorem 3.

- Additional examples about non-verifiable search problems (Theorem 5), and tight reductions for message-hiding encryption (Theorem 6), and multi-message security (Corollary 1).

Beside increasing our confidence in the validity of our new bit security notion, these reductions also start building a toolbox of techniques that can be used to fruitfully employ the new definition in the analysis of both old and new cryptographic primitives, and improve our theoretical understanding of the relation between different cryptographic primitives by means of tight reductions. Finally, they allow us to expand the use of divergence techniques [16, 2, 17, 15] to bound the floating point precision required to secure cryptographic primitives with indistinguishability security properties.

We conclude this section with an informal overview of the new bit security definition. As already mentioned, our definition is based on concepts from information theory. In a purely information theoretic setting, the advantage of an adversary A in discovering a secret X could be modeled by the mutual information $\epsilon = I(A, X)/H(X)$, normalized by the entropy of the secret $H(X)$ to ensure $\epsilon \leq 1$. Of course, this approach completely fails in the computational setting, where the output of a one-way permutation $f(X)$ is perfectly correlated with the input X, but still we do not want to consider a trivial algorithm $A(f(X)) = f(X)$ as a successful attack (with advantage $\epsilon = I(A, X)/H(X) = 1$!) to the one-way permutation input recovery problem: what the adversary knows ($f(X)$) identifies the input X information theoretically, but it does not provide knowledge of it. We adapt this definition to the computational setting by replacing A with a different random variable Y which equals (1) the secret X when A is successful (i.e., $A = X$), and (2) an independent copy X' of the secret (conditioned on $X' \neq X$) when A failed to output X. We find this definition intuitively appealing, and we consider it the main conceptual contribution of this paper. But words are of limited value when arguing about the validity of a new definition. We view the technical results described above the most important evidence to support our definition, and the main technical contribution of this work.

1.2 Related Work

While the informal concept of bit security is widely used in cryptography, not many papers directly address the problem of its formal definition. Some of the works that are perhaps most directly related to our are [6, 7, 8], which pinpoint the shortcoming of the traditional definition. The work of Bernstein and Lange [6] provides an extensive survey of relevant literature, and attempts to provide a better definition. In [6, Appendix B] the authors analyze different measures to address the underlying problems, and show how each of them can be used to make partial progress towards a more sound definition of bit security, while pointing out that none of them seem to solve the problem entirely. In contrast, the definitions and results in this paper concern the definition of adversarial advantage, which we consider to be orthogonal to any of the ideas presented in [6]. So, we see our work as complementary to [6, 7, 8].

To the best of our knowledge there are only two works proposing an alternative definition of adversarial advantage for decision problems: the aforementioned works of Goldreich and Levin [11, 14] and the infamous HILL paper [13]. The latter primarily works with the tradional definition of adversarial advantage, but presents the advantage function δ^2 (note the lack of α) as an alternative, observing that many of their reductions are much tighter in this case. Our work can be considered as a generalization of them, and supporting the definitional choices made in [11, 14]. In the last years, bit security has been the focus on a body of work [16, 2, 17, 15] aimed at optimizing the parameters and floating point precision requirements of lattice cryptography. Our work resolves the main problem left open in [17, 15] of extending definitions and techniques from search to decision problems, and support the secure use of standard precision floating point numbers in the implementation of cryptographic primitives (like encryption) with indistinguishability security properties.

2 Preliminaries

Notation. We denote the integers by $\mathbb{Z}$ and the reals by $\mathbb{R}$. Roman and Greek letters can denote elements from either set, while bold letters denote vectors over them. Occasionally, we construct vectors on the fly

using the notation $(\cdot)_{i\in S}$ for some set S (or in short $(\cdot)_i$ if the set S is clear from context), where $\cdot$ is a function of i. For a set S, we denote its complement by $\bar{S}$. We denote the logarithm to the base 2 by log and the one to the base e by ln.

Calligraphic letters are reserved for probability distributions and $x \leftarrow \mathcal{P}$ means that x is sampled from the distribution $\mathcal{P}$. For any x in the support of $\mathcal{P}$, we denote its probability under $\mathcal{P}$ by $\mathcal{P}(x)$. All distributions in this work are discrete, and $\mathcal{U}(S)$ is the uniform distribution over the support S. If S is clear from context, we simply write $\mathcal{U}$ instead of $\mathcal{U}(S)$. A probability ensemble $\{\mathcal{P}_\theta\}_\theta$ is a family of distributions indexed by a parameter θ (which may be a string or a vector). We extend any divergence δ between distributions to probability ensembles by $\delta(\{\mathcal{P}_\theta\}_\theta, \{\mathcal{Q}_\theta\}_\theta) = \max_\theta \delta(\mathcal{P}_\theta, \mathcal{Q}_\theta)$. For notational simplicity, we do not make a distinction between random variables, probability distributions, and probabilistic algorithms generating them.

Definition 1 *The* statistical distance *between two distributions $\mathcal{P}$ and $\mathcal{Q}$ over S is defined as* $\Delta_{SD}(\mathcal{P}, \mathcal{Q}) = \frac{1}{2}\sum_{x\in S}|\mathcal{P}(x) - \mathcal{Q}(x)|$.

2.1 Information Theory

For our definition, we need a few concepts from information theory.

Definition 2 *The* Shannon entropy *of a random variable X is given by*

$$H(X) = \mathbb{E}_X\left[\log \frac{1}{\Pr\{X\}}\right] = -\sum_x \Pr[X = x] \log \Pr[X = x].$$

Definition 3 *For two random variables X and Y, the* conditional entropy *of X given Y is*

$$H(X|Y) = \mathbb{E}_Y[H(X|Y)] = \sum_{x,y} \Pr[X = x, Y = y] \log \frac{\Pr[Y = y]}{\Pr[X = x, Y = y]}.$$

Definition 4 *The* mutual information *between two random variables X and Y is*

$$I(X;Y) = H(X) - H(X|Y).$$

3 Security Games

In this section we formally define the bit security of cryptographic primitives in a way that captures practical intuition and is theoretically sound. As the security of cryptographic primitives is commonly defined using games, we start by defining a general class of security games.

Definition 5 *An n-bit* security game *is played by an adversary A interacting with a challenger X. At the beginning of the game, the challenger chooses a secret x, represented by the random variable $X \in \{0,1\}^n$, from some distribution $\mathcal{D}_X$. At the end of the game, A outputs some value, which is represented by the random variable A. The goal of the adversary is to output a value a such that $R(x, a)$, where R is some relation. A may output a special symbol $\perp$ such that $R(x, \perp)$ and $\bar{R}(x, \perp)$ are both false.*

This definition is very general and covers a lot of standard games from the literature. Some illustrative examples are given in Table 1. But for the cryptographic primitives explicitly studied in this paper, it will be enough to consider the simplest version of the definition where $R = \{(x, x)|x \in X\}$ is the identity relation, i.e., the goal of the adversary is to guess the secret x. We formally define the indistinguishability game for two distributions because we refer to it extensively throughout this work.

Definition 6 *Let $\{\mathcal{D}^0_\theta\}_\theta$, $\{\mathcal{D}^1_\theta\}_\theta$ be two distribution ensembles. The* indistinguishability game *is defined as follows: the challenger C chooses $b \leftarrow \mathcal{U}(\{0,1\})$. At any time after that the adversary A may (adaptively) request samples by sending θ_i to C, upon which C draws samples $c_i \leftarrow \mathcal{D}^b_{\theta_i}$ and sends c_i to A. The goal of the adversary is to output $b' = b$.*

Table 1: Typical instantiations of security games covered by Definition 5. The security parameter is denoted by κ. In the definition of digital signatures, the list Q of the adversary's queries are regarded as part of its output.

Game	R	n	$\mathcal{D}_X$
Uninvertibility of one-way permutations	$\{(x,y) \mid x = y\}$	$O(\kappa)$	$\mathcal{U}$
Uninvertibility of one-way functions f	$\{(x,y) \mid f(x) = f(y)\}$	$O(\kappa)$	$\mathcal{U}$
2nd pre-image resistance for hash functions h	$\{(x,y) \mid x \neq y, h(x) = h(y)\}$	$O(\kappa)$	$\mathcal{U}$
Indistinguishability of two distributions	$\{(x,y) \mid x = y\}$	1	$\mathcal{U}$
Unforgeability of signature scheme (K,S,V)	$\{(x,(m,\sigma,Q)) \mid (pk,sk) \leftarrow K(x), V(pk,m,\sigma) = 1, m \notin Q\}$	$O(\kappa)$	$K(\mathcal{U})$

We loosely classify primitives into two categories according to their associated security games: we call primitives, where the associated security game is a 1-bit game ($O(\kappa)$-bit game), *decision primitives* (*search primitive*, respectively).

Note that we allow the adversary to always output $\perp$, which roughly means "I don't know", even for decision primitives. This is a crucial difference from previous definitions that force the distinguisher to always output a bit. The reason this is important is that in games, where the distinguisher is not able to check if it produced the correct result, it is more informative to admit defeat rather than guessing at random. In many cases this will allow for much tighter reductions (cf. Section 5.2). Such a definition in the context of indistinguishability games is not entirely new, as Goldreich and Levin [11, 14] also allowed this type of flexibility for the distinguisher. To the best of our knowledge, this is the only place this has previously appeared in the cryptographic literature.

Now we are ready to define the advantage. The definition is trying to capture the amount of information that the adversary is able to learn about the secret. The reasoning is that the inverse of this advantage provides a lower bound on the number of times this adversary needs to be run in order to extract the entire secret. We use tools from information theory to quantify exactly this information, in particular the Shannon entropy. Other notions of entropy might be worth considering, but we focus on Shannon entropy as the most natural definition that captures information. A straight-forward definition could try to measure the mutual information between the random variables X (modeling the secret) and A (modeling the adversary output, cf. Definition 5). Unfortunately, the variable A might reveal X completely in an information theoretical sense, yet not anything in a computational sense. To break any computationally hidden connection between X and A, we introduce another random variable Y, which indicates, when A actually achieves its goal and otherwise does not reveal anything about the secret.

Definition 7 *For any security game with corresponding random variable X and $A(X)$, the adverdary's* advantage *is*

$$\mathrm{adv}^A = \frac{I(X;Y)}{H(X)} = 1 - \frac{H(X|Y)}{H(X)}$$

where $I(\cdot;\cdot)$ is the mutual information, $H(\cdot)$ is the Shannon entropy, and $Y(X,A)$ is the random variable with marginal distributions $Y_{x,a} = \{Y \mid X = x, A = a\}$ defined as

1. *$Y_{x,\perp} = \perp$, for all x.*
2. *$Y_{x,a} = x$, for all $(x,a) \in R$.*
3. *$Y_{x,a} = \{x' \leftarrow \mathcal{D}_X \mid x' \neq x\}$, for all $(x,a) \in \bar{R}$.*

At first glance, the definition of Y might not be obviously intuitive, except for case 1. For case 2, note that x completely determines the set $R(x,\cdot)$ and if the adversary finds an element in it, then it wins the

game. Therefore, one can think of $R(x, \cdot)$ as a secret set, and finding any element in it as completely breaking the scheme. Finally, the third case defines Y to follow the distribution of the secret, but is conditioned on the event that it is incorrect. The intuition here is that if an adversary outputs something, then his goal is to bias the secret distribution towards the correct one, i.e. it will allow us to quantify how much better A performs than random guessing.

With the definition of the advantage in place, the definition of bit security follows quite naturally.

Definition 8 *Let $T : \{A \mid A \text{ is any algorithm}\} \mapsto \mathbb{Z}_+$ be a measure of resources that is linear under repetition, i.e. $T(kA) = kT(A)$, where kA is the k time repetition of A. For any primitive, we define its* bit security *as* $\min_A \log \frac{T(A)}{adv^A}$.

For convenience we will often write $T(A)$ as T^A or simply T if A is clear from context. Note that we leave out a concrete definition of the resources on purpose, since we focus on the advantage. Our definition can be used with many different measures, for example running time, space, advice, etc., or combinations of them.

4 The Adversary's Advantage

While the advantage as defined in the previous section captures the intuition about how well an adversary performs, it seems too complex to be handled in actual proofs or to be used in practice. A simple definition in terms of simple quantities related to the adversary would be much more desirable. We begin by defining the quantities of an adversary that we are interested in.

Definition 9 *For any adversary A playing a security game, we define its* output probability *as $\alpha^A = \Pr[A \neq \perp]$ and its* conditional success probability *as $\beta^A = \Pr[R(X, A)|A \neq \perp]$, where the probabilities are taken over the randomness of the entire security game (including the internal randomness of A). Finally, in the context of decision primitives, we also define A's* conditional distinguishing advantage *as $\delta^A = 2\beta^A - 1$. With all of these quantities, when the adversary A is clear from context, we drop the corresponding superscript.*

The goal of this section is to distill a simple definition of advantage in terms of α_A and β^A by considering a broad and natural class of adversaries and games.

Theorem 1 *For any n-bit security game with uniform secret distribution, let A be an adversary that for any secret $x \in \{0,1\}^n$ outputs $\perp$ with probability $1 - \alpha$, some value a such that $R(x, a)$ with probability $\beta\alpha$, and some value a such that $\bar{R}(x, a)$ with probability $(1 - \beta)\alpha$. Then*

$$\mathrm{adv}^A = \alpha \left(1 - \frac{(1-\beta)\log(2^n - 1) + H(\mathcal{B}_\beta)}{n}\right) \tag{1}$$

where $\mathcal{B}_\beta$ denotes the Bernoulli distribution with parameter β.

We defer the proof to Appendix A. Note that for large n we get $\mathrm{adv}^A \approx \alpha^A\beta^A$, which is exactly A's success probability. Plugging this into Definition 8 matches the well-known definition of bit security for search primitives. On the other hand, for $n = 1$ this yields $\mathrm{adv}^A = \alpha^A(1-H(\mathcal{B}_{\beta^A})) = \alpha^A(\delta^A)^2/(2\ln 2)+O(\alpha^A(\delta^A)^4)$ by Taylor approximation, which, for our purposes, can be approximated by $\alpha^A(\delta^A)^2$. This matches the definition of Levin [14], who proposed this definition since it yields the inverse sample complexity of noticing the correlation between the adversary output and the secret. The fact that it can be derived from Definition 7 suggests that this is the "right" definition of the adversary's advantage.

We now redefine the adversary's advantage according to above observations, which, combined with Definition 8 yields the definition of bit security we actually put forward and will use throughout the rest of this work.

Definition 10 *For a search game, the advantage of the adversary A is*

$$\mathrm{adv}^A = \alpha^A \beta^A$$

and for a decision game, it is

$$\mathrm{adv}^A = \alpha^A (\delta^A)^2.$$

Note that assuming that Definition 10 is equivalent to 7 for all adversaries is quite a leap as we only proved it for a subclass of them, and in fact, it is not true at least for decision games. However, the following theorem shows that when used in the context of bit security (Definition 8) for decision games, Definition 10 and 7 are in fact equivalent, since we are quantifying over all adversaries.

Theorem 2 *For any distinguisher D playing a decision game with $\mathrm{adv}^D = \zeta$ according to Definition 7, there is a distinguisher D' such that $T^D = T^{D'}$ and $\alpha^{D'}(\delta^{D'})^2 \geq \zeta/9$ for the same game.*

Before we prove Theorem 2, we observe that the distinguisher D' that we construct from D will run D and decide on its output depending on the result. As such, D' is essentially a distinguisher for the indistinguishability game (restricted to one query) against the two distributions induced by the secret on D. We start with a simple lemma that analyzes how well such a simple distinguisher does in this game.

Lemma 1 *Let $\mathcal{D}_x$ for $x \in \{0,1\}$ be two distributions over the same support $\{a,b,c\}$ and denote their probabilities by $z_x = \mathcal{D}_x(z)$ for $z \in \{a,b,c\}$. Let D_z be a distinguisher for the indistinguishability game instantiated with $\mathcal{D}_x$ that on input z returns $\arg\max_x(z_x)$ and $\perp$ otherwise. Then,*

$$\alpha^{D_z}(\delta^{D_z})^2 = \frac{1}{2}\frac{(z_1 - z_0)^2}{z_1 + z_0}.$$

We now prove Theorem 2 by showing that for any distinguisher D there is an event $z \in \{\perp, 0, 1\}$ such that $\alpha^{D_z}(\delta^{D_z})^2 \approx \mathrm{adv}^D$.

Proof [of Theorem 2] Since adv^D is independent of the support/domain of D (as long as it has size exactly 3), we identify $\{\perp, 0, 1\}$ with a, b, c to highlight this genericity.

With the same notation as in Lemma 1, we note that the conditional entropy of the secret X given Y is

$$H(X|Y) = \frac{1}{2}\left(H_1(a_0, a_1) + H_1(b_0, b_1) + H_1(c_0, c_1)\right)$$

where

$$\begin{aligned} H_1(z_0, z_1) &= z_0 \log \frac{z_0 + z_1}{z_0} + z_1 \log \frac{z_0 + z_1}{z_1} \\ &= ((z_0 + z_1)\log((z_0 + z_1) - z_0 \log z_0 - z_1 \log z_1. \end{aligned}$$

Setting $\bar{z} = z_1 - z_0$, H_1 can be rewritten as

$$H_1(z_0, \bar{z}) = (2z_0 + \bar{z})\log(2z_0 + \bar{z}) + z_0 \log z_0 + (z_0 + \bar{z})\log(z_0 + \bar{z}).$$

We use the following bound on H_1:

$$H_1(z_0, \bar{z}) \geq 2z_0 \qquad \text{for } \bar{z} \geq 0 \tag{2}$$

$$H_1(z_0, \bar{z}) \geq 2z_0 + \bar{z} - \frac{\bar{z}^2}{z_0} \qquad \text{for } |\bar{z}| \leq z_0 \tag{3}$$

where (2) follows from monotonicity in $\bar{z}$ and (3) from Taylor approximation of order 2 in $\bar{z}$ at $\bar{z} = 0$. Since $\bar{z} > z_0$ implies that (2) is larger than (3), these bounds imply

$$H_1(z_0, \bar{z}) \geq \max\left(2z_0, 2z_0 + \bar{z} - \frac{\bar{z}^2}{z_0}\right) \tag{4}$$

for all $\bar{z} \in [-z_0, 1-z_0]$. In the following, we will apply the bound (3) for $\bar{z} \in [-z_0, 0]$ and (4) for $\bar{z} \in [0, 1-z_0]$.

W.l.o.g. assume $\bar{a} \geq 0$, $\bar{b} \leq 0$ and $\bar{c} \leq 0$ (note that $\sum_{z \in \{a,b,c\}} \bar{z} = 0$). Using (3) and (4)

$$\begin{aligned} H(X|Y) &\geq \frac{1}{2}\left[\max\left(2a_0, 2a_0 + \bar{a} - \frac{\bar{a}^2}{a_0}\right) + 2b_0 + \bar{b} - \frac{\bar{b}^2}{b_0} + 2c_0 + \bar{c} - \frac{\bar{c}^2}{c_0}\right] \\ &= 1 + \frac{1}{2}\left[\max\left(-\bar{a}, -\frac{\bar{a}^2}{a_0}\right) - \frac{\bar{b}^2}{b_0} - \frac{\bar{c}^2}{c_0}\right] \end{aligned}$$

which shows that

$$\begin{aligned} \mathrm{adv}^D &\leq \frac{1}{2}\left[-\max\left(-\bar{a}, -\frac{\bar{a}^2}{a_0}\right) + \frac{\bar{b}^2}{b_0} + \frac{\bar{c}^2}{c_0}\right] \\ &= \frac{1}{2}\left[\min\left(\bar{a}, \frac{\bar{a}^2}{a_0}\right) + \frac{\bar{b}^2}{b_0} + \frac{\bar{c}^2}{c_0}\right] \\ &\leq \frac{3}{2}\max\left[\min\left(\bar{a}, \frac{\bar{a}^2}{a_0}\right), \frac{\bar{b}^2}{b_0}, \frac{\bar{c}^2}{c_0}\right]. \end{aligned}$$

Note that if the maximum is attained by one of the latter two terms, since $\bar{b}$ and $\bar{c}$ are negative, we have $\alpha^{D_b}(\delta^{D_b})^2 \geq \frac{\bar{b}^2}{4b_0}$ by Lemma 1 (and similarly for c). So $\mathrm{adv}^D \leq 6\alpha^{D_z}(\delta^{D_z})^2$ for one of $z \in \{b, c\}$.

Now assume the maximum is $\min(\bar{a}, \frac{\bar{a}^2}{a_0})$. If $\frac{\bar{a}^2}{a_0} \leq \bar{a}$, then $\bar{a} \leq a_0$ and so $a_0 + a_1 \leq 3a_0$. Again by Lemma 1, $\alpha^{D_a}(\delta^{D_a})^2 \geq \frac{\bar{a}^2}{6a_0}$. Finally, if $\bar{a} \leq \frac{\bar{a}^2}{a_0}$ then $a_0 \leq \bar{a}$, which means $a_0 + a_1 \leq 3\bar{a}$ and so by Lemma 1, $\alpha^{D_a}(\delta^{D_a})^2 \geq \frac{\bar{a}}{6}$. In both cases we have $\mathrm{adv}^D \leq 9\alpha^{D_a}(\delta^{D_a})^2$. □

5 Security Reductions

To argue that our definition is useful in a theoretical sense, we apply it to several natural reductions, which arise when constructing cryptographic primitives from other ones. As the novelty of our definition lies mostly with decision games, we will focus on decision primitives that are built from search primitives (cf. Section 5.1), search primitives that are built from decision primitives (cf. Section 5.2), and finally decision primitves that are built from other decision primitives (cf. 5.3).

Throughout this section we will refer to two distribution ensembles $\{\mathcal{D}^0_\theta\}_\theta$ and $\{\mathcal{D}^1_\theta\}_\theta$ as κ-bit indistinguishable, if the indistinguishability game from Definition 6 instantiated with $\{\mathcal{D}^0_\theta\}_\theta$ and $\{\mathcal{D}^1_\theta\}_\theta$ is κ-bit secure.

5.1 Search to Decision

A classical way to turn a search primitive into a decision primitive is the Goldreich-Levin hardcore bit[11].

Definition 11 *Let $f : X \mapsto Y$ be a function and $b : X \mapsto \{0,1\}$ be a predicate. The predicate b is a κ-bit secure* hardcore bit *for f, if the distributions $(f(x), b(x))$ and $(f(x), \mathcal{U}(\{0,1\}))$, where $x \leftarrow \mathcal{U}(X)$, are κ-bit indistinguishable.*

Goldreich and Levin showed a way to construct a function with a hardcore bit from any one-way function. In this setting, one would hope that if the one-way function is κ-bit secure then also the hardcore bit is close to κ bit secure. The next theorem due to Levin [14] establishes exactly such a connection.

Theorem 3 (adapted from [14]) *Let $f : \{0,1\}^n \mapsto \{0,1\}^k$ be a κ-bit secure one-way function. Then $b(x,r) = \langle x, r\rangle \mod 2$ is a $(\kappa - O(\log n))$-bit secure hardcore bit for $g(x,r) = (f(x), r)$.*

This theorem was proven in [14], and all we did was to adapt the statement from [14] to our notation/framework. So, we refer the reader to [14] for the proof details, and move on to make some general observations. The proof for this theorem assumes a distinguisher D for b and constructs from it an inverter A for f, where $\mathrm{adv}^D = \mathrm{adv}^A$ (and the running time is polynomially related). Such security preserving reductions are information theoretically only possible with a definition of advantage that is proportional to $(\delta^D)^2$ for decision primitives, if it is proportional to $\alpha^A\beta^A$ for search primitives. This is because any inverter querying a distinguisher with advantage δ^D and attempting to learn an $(\alpha^A\beta^A)$-fraction of a uniformly chosen n-bit secret, must make at least $\Omega(n\alpha^A\beta^A/(\delta^D)^2)$ queries. Denote the resources of D by T^D and note that $T^A \geq \Omega(\alpha^A\beta^A/(\delta^D)^2)T^D$ is a lower bound on the resources of A. The goal of the proof is to find an upper bound on $T^A/\mathrm{adv}^A = T^A/\alpha^A\beta^A \geq \Omega(T^D/(\delta^D)^2)$. This is only possible by assuming an upper bound on $T^D/(\delta^D)^2$. If only a bound on T^D/δ^D is assumed, then the upper bound on T^A/adv^A must contain a linear factor in $1/\delta^D$, which may be as large as $O(2^n)$ and thus result in a dramatic loss in (nominal) security.

5.2 Decision to Search

In the following subsections we show constructions and the corresponding reductions in the other direction. The first is just a straightforward converse to the Goldreich-Levin theorem, showing that any PRG is also a OWF for the same bit security. The second construction is presented as a very natural and straight-forward way of turning a decision primitive into a search primitive. The third reduction is one that naturally arises in cryptographic applications, for example identification protocols.

5.2.1 PRGs are one-way functions

While the following theorem is intuitively trivial (and technically simple), as explained in the introduction it serves to justify our definition of bit security. The proof also illustrates the subtle difference between an adversary that outputs $\perp$ and one that outputs a random guess.

Theorem 4 *If g is a PRG with κ-bit security, then it is also a $(\kappa - 4)$-bit secure one-way function.*

Proof Assume A is an attack to g as a one-way function with cost T, output probability α^A, and conditional success probability β^A. We turn A into an adversary D to g as a PRG by letting $D(y)$ output 1 if $g(A(y)) = y$ and $\perp$ otherwise. Assume that A has conditional success probability $\beta^A = 1$. This is without loss of generality because one-way function inversion is a verifiable search problem, and A can be modified (without affecting its advantage) to output $\perp$ when its answer is incorrect. So, A has advantage α^A, equal to its output probability. Notice that D is successful only when the indistinguishability game chooses the secret bit 1, and then A correctly inverts the PRG. So, the success probability of D is precisely $\alpha^D\beta^D = \alpha^A/2$. The output probability of D can be a bit higher, to take into account the possibility that on secret bit 0, the challenger picks a random string that belongs (by chance) to the image of the PRG, and A correctly inverts it. Assume g has stretch 1 (which corresponds to the worst case). In that case, $\alpha^D = 3/4 \cdot \alpha^A$. It follows that $\alpha^D = \alpha^A/2$ and $\beta^D = (\alpha^A/2)/\alpha^D = 2/3$. So, D has advantage $\alpha^D(\delta^D)^2 = \alpha^D(2\beta^D - 1)^2 = \alpha^A/12$. Since the two algorithms have essentially the same cost, they achieve the same level of bit security, up to a small constant additive term $\log 12 < 4$. $\square$

We remark that our proof differs from the standard text-book reduction that pseudorandom generators are one-way functions in a simple, but crucial way: when $A(y)$ fails to invert g, instead of outputting 0 as a "best guess" at the decision problem, it outputs $\perp$ to explicitly declare failure. The reader can easily check that the standard reduction has output probability $\alpha^D = 1$ and (conditional) success probability $\beta^D \leq (\alpha^A + 1)/2$. So, the advantage of the distiguisher in the standard proof is $\alpha^D(2\beta^D - 1)^2 = (\alpha^A)^2$, resulting in a substantial drop $(\log \alpha^A)$ in the bit security proved by the reduction.

5.2.2 Secret Recovery

We proceed by giving a construction of a search primitive from two distributions. We are not aware of any immediate applications, but this simple example is supposed to serve as evidence that our definitions

for search and decision primitives behave nicely under composition. It also provides an example of "non verifiable" search problem, i.e., a cryptographic problem with exponentially large secret space defined by a game at the end of which A cannot efficiently determine if the secret has been found. Differently from Theorem 4, this time one *cannot* assume without loss of generality that the (hypothetical) attacker to the search problem has conditional success probability $\beta = 1$.

Definition 12 *Let $\mathcal{D}_0, \mathcal{D}_1$ be two distributions. We define the n-bit* secret recovery game *as the following n-bit security game: the challenger X chooses an n-bit secret $x \leftarrow \mathcal{U}(\{0,1\}^n)$ and sends the vector $\mathbf{c} = (c_i \leftarrow D_{x_i})_{i \leq n}$ to A. The adversary A attempts to guess x, i.e. R is the equality relation.*

The next theorem shows that when instantiating the game with two indistinguishable distributions, the secret recovery game enjoys essentially the same bit security.

Theorem 5 *If the κ-bit secret recovery game is instantiated with two κ-bit secure indistinguishable distributions $\mathcal{D}_0$ and $\mathcal{D}_1$, and $\mathcal{D}_0$ is publicly sampleable, then it is $(\kappa - 1)$-bit secure.*

Proof Let A be an adversary against the secret recovery game that recovers x from the vector $\mathbf{c}$ with advantage $\mathrm{adv}^A = \alpha^A \beta^A$. We build a distinguisher D against the indistinguishability of $\mathcal{D}_0$ and $\mathcal{D}_1$ with essentially the same resources and advantage: D chooses a secret $x \in \{0,1\}^\kappa$ uniformly at random, which is non-zero with high probability (otherwise output $\perp$) and constructs the vector $\mathbf{c}$ by sampling $\mathcal{D}_0$ itself for every zero bit in x and querying its orcale for every 1 bit in x (which will return either samples from $\mathcal{D}_0$ or from $\mathcal{D}_1$). It sends $\mathbf{c}$ to A and returns 1 iff A returns x, otherwise it outputs $\perp$.

The resources of D are essentially the same as those of A, so we analyze its advantage $\mathrm{adv}^D = \alpha^D (\delta^D)^2$. The output probability of D, conditioned on $x \neq 0$, is almost exactly A's success probability, but note that A is only presented with the correct input distribution if D's challenger returns samples from $\mathcal{D}_1$, which is the case with probability $\frac{1}{2}$. So $\alpha^D \geq \frac{1-2^{-\kappa}}{2} \alpha^A \beta^A$. Furthermore, D's conditional distinguishing advantage is $\delta^D \geq 1 - 2^{-\kappa+1}$, because it only outputs the incorrect value if A returned x even though $\mathbf{c}$ consisted of samples only from $\mathcal{D}_0$. Note that in this case A has no information about x, which was chosen uniformly at random and thus the probability of this event is at most $2^{-\kappa}$. Accordingly, $\mathrm{adv}^D = \alpha_D (\delta^D)^2 \geq \frac{(1-2^{-\kappa+1})^2}{2} \alpha^A \beta^A \approx \mathrm{adv}_A / 2$. $\square$

5.2.3 Indistinguishability implies Message-Hiding

In our last example for this section we show that IND-CCA secure encryption schemes enjoy a message hiding property, which we first formally define.

Definition 13 *A private or public key encryption scheme is κ-bit* message hiding*, if the following security game is κ-bit secure: the challenger chooses a message $m \in \{0,1\}^n$ uniformly at random and sends its encryption to A. The adversary A attempts to guess m, while C provides it with encryption (in case of private key schemes) and decryption oracles.*

This property naturally arises in the context of constructions of identification protocols from encryption schemes (see e.g. [4]), where a random message is encrypted and identification relies on the fact that only the correct entity can decrypt it. While it seems intuitively obvious that breaking message hiding is no easier than distinguishing encrypted messages, showing that this is true in a quantifiable sense for specific definitions of bit security is not as obvious. The next theorem establishes this connection.

Theorem 6 *If a scheme with message space larger than 2^κ is κ-bit IND-CCA secure, it is κ-bit* message hiding.

Proof Let A be an adversary that is able to extract a random message from an encryption scheme with advantage $\mathrm{adv}^A = \alpha^A \beta^A$. We construct a IND-CCA distinguisher D against the scheme with essentially the same resources and advantage: D generates two messages $m_0, m_1 \leftarrow \{0,1\}^m$ uniformly at random, which are

distinct with overwhelming probability (if not, output $\perp$). It sends them to the challenger, which encrypts one of them. Upon receiving the challenge cipher text c_b, D forwards it to A. Any queries to the encryption (in case of private key encryption) or decryption oracle are simply forwarded to D's own oracles. If A returns a message in $\{m_0, m_1\}$, D returns the corresponding bit. Otherwise, it outputs $\perp$.

The resources of D are essentially the same as for A, so we focus on its advantage. Note that conditioned on the event that $m_0 \neq m_1$, D's output probability α^D is at least as large as the success probability of A, so $\alpha^D \geq (1 - 2^{-\kappa})\alpha^A\beta^A$. The conditional distinguishing advantage of D is $\delta^D \geq 1 - 2^{-\kappa+1}$, since the only way D will guess incorrectly is when A somehow outputs the wrong message $m_{\bar{b}}$. Since A has no information about this message (which was chosen uniformly at random), the probability of this happening is at most $2^{-\kappa}$. This shows that D's advantage in the indistinguishability game is $\mathrm{adv}^D = \alpha^D(\delta^D)^2 \geq (1-2^{-\kappa})\alpha^A\beta^A(1-2^{-\kappa+1})^2 \approx \alpha^A\beta^A = \mathrm{adv}^A$, where the latter is A's advantage in the message hiding game. $\square$

5.3 Decision to Decision

Finally, we turn to reductions between decision primitives. The results in this section are very generic. The first establishes the validity of hybrid arguments when using our definition of advantage for decision primitives. Our second result extends a previous result for approximate sampling to any decision primitive fitting our definition.

5.3.1 The Hybrid Argument

This section is devoted to proving a general hybrid argument for indistinguishability games using our definition of advantage. Formally, we prove the following lemma.

Lemma 2 *Let $\mathcal{H}_i$ be k distributions and $G_{i,j}$ be the indistinguishability game instantiated with $\mathcal{H}_i$ and $\mathcal{H}_j$. Further, let $\epsilon_{i,j} = \max_A \mathrm{adv}^A$ over all T-bounded adversaries A against $G_{i,j}$. Then $\epsilon_{1,k} \leq 3k \sum_{i=1}^{k-1} \epsilon_{i,i+1}$.*

Applying the lemma to our definition of bit security, we immediately get the following theorem.

Theorem 7 *Let $\mathcal{H}_i$ be k distributions. If $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ are κ-bit indistinguishable for all i, then $\mathcal{H}_1$ and $\mathcal{H}_k$ are $(\kappa - 2(\log k + 1))$-bit indistinguishable.*

Proof Let A be any adversary with resources T^A (when attacking $\mathcal{H}_1$ and $\mathcal{H}_k$). By assumption, $\epsilon_{i,i+1} \leq T^A/2^\kappa$ (where $\epsilon_{i,j}$ is defined as in Lemma 2) for all T^A-bounded adversaries against $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$. By Lemma 2, $\epsilon_{i,k} \leq 3k^2T^A/2^\kappa$ for all T^A-bounded adversaries, in particular A. $\square$

As a simple application, we get the following corollary.

Corollary 1 *If a public key encryption scheme is κ-bit IND-CCA secure, then it is $(\kappa - 2(\log k + 1))$-bit IND-CCA secure in the k message setting.*

In contrast to the standard hybrid argument, which simply exploits the triangle inequality of statistical distance, we lose an additional factor of $3k$ in the advantage in Lemma 2. In particular, consider the case where the bounds $\epsilon_{i,i+1} = \epsilon$ are the same for all i. This means that $\epsilon_{1,k} \leq 3k^2\epsilon$. Note that this additional factor has only a minor impact on bit security. (See below for details.) Still, one may wonder if this additional factor is an artifact of a non-tight proof or if it is indeed necessary. Consider a distinguisher D that never outputs $\perp$ (i.e. $\alpha^D = 1$). Its distinguishing advantage $\delta^D_{i,j}$ in game $G_{i,j}$ is exactly the statistical distance between $D(\mathcal{H}_i)$ and $D(\mathcal{H}_j)$. Assume $\delta^D_{i,i+1} = \epsilon$ for all i, so D's advantage in the game $G_{i,j}$ according to Definition 10 is ϵ^2. The standard hybrid argument, or equivalently triangle inequality for statistical distance, implies that $\delta^D_{1,k}$ cannot be larger than – but may be as large as – $k\epsilon$. So, D's advantage in $G_{1,k}$ may be as large as $k^2\epsilon^2$, which is k^2 times as large as D's advantage against the individual hybrids. This seems to

suggest that our argument is tight (up to the constant factor 3). Either way, as Theorem 7 and Corollary 1 demonstrate, this additional factor only affects the constant in front of the log term in the number of hybrids, so, we believe, it is only of secondary importance and we leave it as an open problem.

The rest of the subsection proves Lemma 2, where we make use of the following notation. For some distinguisher D, let $\alpha^D_{\mathcal{P},\mathcal{Q}}$ be its output probability, $\beta^D_{\mathcal{P},\mathcal{Q}}$ its conditional success probability, $\delta^D_{\mathcal{P},\mathcal{Q}}$ its conditional distinguishing advantage, and $\mathrm{adv}^D_{\mathcal{P},\mathcal{Q}} = \alpha^D_{\mathcal{P},\mathcal{Q}}(\delta^D_{\mathcal{P},\mathcal{Q}})^2$ its advantage against the distributions $\mathcal{P}, \mathcal{Q}$. Furthermore, let $\alpha^D_{\mathcal{P}} = \Pr[D(\mathcal{P}) \neq \perp]$ and $\gamma^D_{\mathcal{P}} = \Pr[D(\mathcal{P}) = 1]$ for any distribution $\mathcal{P}$. We can express the advantage of D against $\mathcal{P}$ and $\mathcal{Q}$ in terms of $\alpha^D_{\mathcal{P}}$, $\alpha^D_{\mathcal{Q}}$, $\gamma^D_{\mathcal{P}}$, $\gamma^D_{\mathcal{Q}}$:

$$\begin{aligned}
\alpha^D_{\mathcal{P},\mathcal{Q}} &= \frac{1}{2}(\alpha^D_{\mathcal{P}} + \alpha^D_{\mathcal{Q}}) \\
\beta^D_{\mathcal{P},\mathcal{Q}} &= \frac{\gamma^D_{\mathcal{P}} - \gamma^D_{\mathcal{Q}} + \alpha^D_{\mathcal{Q}}}{\alpha^D_{\mathcal{P}} + \alpha^D_{\mathcal{Q}}} \\
\delta^D_{\mathcal{P},\mathcal{Q}} &= 2\beta^D_{\mathcal{P},\mathcal{Q}} - 1 = \frac{2(\gamma^D_{\mathcal{P}} - \gamma^D_{\mathcal{Q}}) + \alpha^D_{\mathcal{Q}} - \alpha^D_{\mathcal{P}}}{\alpha^D_{\mathcal{P}} + \alpha^D_{\mathcal{Q}}} \\
\mathrm{adv}^D_{\mathcal{P},\mathcal{Q}} &= \frac{(2(\gamma^D_{\mathcal{P}} - \gamma^D_{\mathcal{Q}}) + \alpha^D_{\mathcal{Q}} - \alpha^D_{\mathcal{P}})^2}{2(\alpha^D_{\mathcal{P}} + \alpha^D_{\mathcal{Q}})}.
\end{aligned} \tag{5}$$

We begin with the observation that for computationally indistuingishable distributions the output probabilities of any bounded distinguisher D cannot vary too much under the two distributions.

Lemma 3 *Let $\mathcal{P}, \mathcal{Q}$ be two distributions. If $\mathrm{adv}^D_{\mathcal{P},\mathcal{Q}} \leq \epsilon$ for all T-bounded distinguishers, then we have $\alpha^D_{\mathcal{P}} \leq 2\alpha^D_{\mathcal{Q}} + 3\epsilon$ and $\alpha^D_{\mathcal{Q}} \leq 2\alpha^D_{\mathcal{P}} + 3\epsilon$ for any T bounded distinguisher.*

Proof We prove the first claim. (The proof of the second claim is symmetrical.) Fix any distinguisher D. Assume $\alpha^D_{\mathcal{P}} \geq 2\alpha^D_{\mathcal{Q}}$, since otherwise we are done. Consider an alternative distinguisher D', which runs D and in the event that $D \neq \perp$, outputs 1 and otherwise $\perp$. Obviously, D' is also T-bounded, and (setting $\gamma^{D'}_{\mathcal{P}} = \alpha^{D'}_{\mathcal{P}}$, $\gamma^{D'}_{\mathcal{Q}} = \alpha^{D'}_{\mathcal{Q}}$ in (5)) we get

$$\begin{aligned}
\mathrm{adv}^{D'}_{\mathcal{P},\mathcal{Q}} &= \frac{(\alpha^D_{\mathcal{P}} - \alpha^D_{\mathcal{Q}})^2}{2(\alpha^D_{\mathcal{P}} + \alpha^D_{\mathcal{Q}})} \\
&\geq \frac{(\alpha^D_{\mathcal{P}} - \alpha^D_{\mathcal{Q}})^2}{3\alpha^D_{\mathcal{P}}} \\
&= \frac{1}{3}\left(\alpha^D_{\mathcal{P}} - 2\alpha^D_{\mathcal{Q}} + \frac{(\alpha^D_{\mathcal{Q}})^2}{\alpha^D_{\mathcal{P}}}\right) \\
&\geq \frac{1}{3}\left(\alpha^D_{\mathcal{P}} - 2\alpha^D_{\mathcal{Q}}\right).
\end{aligned}$$

The first claim now follows from $\epsilon \geq \mathrm{adv}^{D'}_{\mathcal{P},\mathcal{Q}}$. □

Proof [of Lemma 2] We fix any distinguisher D and drop the superfix of α, γ, δ and adv for the rest of the proof. Furthermore, we will abbreviate $\mathcal{H}_i$ by i in the subfixes of α, γ, δ, and adv.

Using induction, one can prove

$$\sum_{i=1}^{k} \mathrm{adv}_{i,i+1} \geq \frac{\alpha_1 + \alpha_k}{\alpha_1 + 2\sum_{i=2}^{k-1}\alpha_i + \alpha_k}\mathrm{adv}_{1,k}$$

The proof proceeds by substituting in the definition of $\mathrm{adv}_{i,i+1}$ from (5), applying the induction hypothesis to the first $k-1$ terms of the sum, and then minimizing over γ_{k-1}. Details can be found in Appendix B.

It remains to show that

$$\frac{\alpha_1 + \alpha_k}{\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k} \geq \frac{1}{3k}.$$

We again proceed by induction and can thus assume that $\mathrm{adv}_{1,i} \leq 3i \sum_{j=1}^{i-1} \epsilon_{j,j+1}$ for all $i < k$ and symmetrically $\mathrm{adv}_{i,k} \leq 3(k-i) \sum_{j=i}^{k-1} \epsilon_{j,j+1}$ for all $i > 1$. By Lemma 3, this means that $\alpha_i \leq 2\alpha_1 + 9i \sum_{j=1}^{i-1} \epsilon_{j,j+1}$ for all $i < k$ and again $\alpha_i \leq 2\alpha_k + 9(k-i) \sum_{j=i}^{k-1} \epsilon_{j,j+1}$ for all $i > 1$. We note that

$$\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k = \alpha_1 + 2 \sum_{i=2}^{\lfloor (k-1)/2 \rfloor} \alpha_i + 2 \sum_{\lfloor (k-1)/2 \rfloor + 1}^{k-1} \alpha_i + \alpha_k$$

and using the above inequalities, the two sums are bounded by

$$2 \sum_{i=2}^{\lfloor (k-1)/2 \rfloor} \alpha_i \leq 2(k-3)\alpha_1 + 3k^2 \sum_{i=1}^{\lfloor (k-1)/2 \rfloor} \epsilon_{i,i+1}$$

and

$$2 \sum_{\lfloor (k-1)/2 \rfloor + 1}^{k-1} \alpha_i \leq 2(k-3)\alpha_k + 3k^2 \sum_{\lfloor (k-1)/2 \rfloor + 1}^{k-1} \epsilon_{i,i+1}$$

respectively. This bounds the entire sum:

$$\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k \leq 2k(\alpha_1 + \alpha_k) + 3k^2 \sum_{i=1}^{k-1} \epsilon_{i,i+1}$$

This in turn leads to the lower bound

$$\frac{\alpha_1 + \alpha_k}{\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k} \geq \frac{1}{2k + \frac{3k^2 \sum_{i=1}^{k-1} \epsilon_{i,i+1}}{\alpha_1 + \alpha_k}}$$

The last step is noticing that we can assume that $(\alpha_1 + \alpha_k) \geq 6k \sum_{i=1}^{k-1} \epsilon_{i,i+1}$, because $(\alpha_1 + \alpha_k)/2 \geq \epsilon_{1,k}$ and otherwise we would be done. Using this assumption we have

$$\frac{\alpha_1 + \alpha_k}{\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k} \geq \frac{1}{2k + \frac{3k^2}{6k}} \geq \frac{1}{3k}$$

as desired. □

5.3.2 Approximate Samplers

In this section we bridge the gap between search and decision primitives making use of approximate samplers, for the first time by extending a result from [15] to arbitrary decision primitives. It might be possible to extend other results from the literature [16, 2, 17] to decision primitives using our definition, but we leave that for future work. Our main result is given in Theorem 8. Combining it with results from [15] it implies that approximating a distribution with relative error bounded by $2^{-\kappa/2}$ (e.g., as provided by floating point numbers with $\kappa/2$-bit mantissa) allows to preserve almost all of κ bits of security.

Before introducing the result formally, we first need to cover some preliminaries from [15].

Background Using the same terminology as [15], let $\delta(\mathcal{P}, \mathcal{Q})$ be some divergence on probability distributions. A λ-*efficient* divergence satisfies three properties:

1. *Sub-additivity for joint distributions:* if $(X_i)_i$ and $(Y_i)_i$ are two lists of discrete random variables over the support $\prod_i S_i$, then

$$\delta((X_i)_i, (Y_i)_i) \leq \sum_i \max_a \delta([X_i \mid X_{<i} = a], [Y_i \mid Y_{<i} = a]),$$

   where $X_{<i} = (X_1, \ldots, X_{i-1})$ (and similarly for $Y_{<i}$), and the maximum is taken over $a \in \prod_{j<i} S_j$.

2. *Data processing inequality:* $\delta(f(\mathcal{P}), f(\mathcal{Q})) \leq \delta(\mathcal{P}, \mathcal{Q})$ for any two distributions $\mathcal{P}$ and $\mathcal{Q}$ and (possibly randomized) algorithm $f(\cdot)$, i.e., the measure does not increase under function application.

3. *Pythagorean probability preservation* with parameter $\lambda \in \mathbb{R}$: if $(X_i)_i$ and $(Y_i)_i$ are two lists of discrete random variables over the support $\prod_i S_i$ and

$$\delta((X_i \mid X_{<i} = a_i), (Y_i \mid Y_{<i} = a_i)) \leq \lambda$$

   for all i and $a_i \in \prod_{j<i} S_j$, then

$$\Delta_{SD}((X_i)_i, (Y_i)_i) \leq \left\| \left(\max_{a_i} \delta((X_i \mid X_{<i} = a_i), (Y_i \mid Y_{<i} = a_i)) \right)_i \right\|_2.$$

As an example, the max-log distance $\Delta_{ML}(\mathcal{P}, \mathcal{Q}) = \max|\log \mathcal{P}(x) - \log \mathcal{Q}(x)|$ is λ-efficient for any $\lambda \leq \frac{1}{3}$ [15].

Main Result for Approximate Samplers The next theorem states the main result of this section. It shows that it suffices to approximate a distribution $\mathcal{P}$ up to distance $\delta(\mathcal{P}, \mathcal{Q}) \leq 2^{-\kappa/2}$ for an efficient divergence δ in order to maintain almost κ bits of security.

Theorem 8 *Let $S^{\mathcal{P}}$ be a 1-bit secrecy game with black-box access to a probability ensemble $(\mathcal{P}_\theta)_\theta$, and δ be a λ-*efficient *measure for any $\lambda \leq \frac{1}{4}$. If $S^{\mathcal{P}}$ is κ-bit secure and $\delta(\mathcal{P}_\theta, \mathcal{Q}_\theta) \leq 2^{-\kappa/2}$, then $S^{\mathcal{Q}}$ is $(\kappa - 8)$-bit secure.*

The remainder of this section is devoted to proving Theorem 8. We first rewrite a lemma from [15], which we will use in our proof.

Lemma 4 (adapted from [15]) *Let $S^{\mathcal{P}}$ be any security game with black-box access to a probability distribution ensemble $\mathcal{P}_\theta$. For any adversary A with resources T that plays $S^{\mathcal{P}}$ and event E over its output, denote $\gamma_{\mathcal{P}} = \Pr[A \in E]$. For the same event, denote by $\gamma_{\mathcal{Q}}$ the probability of E when A is playing $S^{\mathcal{Q}}$. If $\frac{T}{\gamma_{\mathcal{P}}} \geq 2^k$ and $\delta(\mathcal{P}_\theta, \mathcal{Q}_\theta) \leq 2^{-k/2}$ for any $2^{-k/2}$-efficient δ, then $\frac{T}{\gamma_{\mathcal{Q}}} \geq 2^{k-3}$.*

From Lemma 4 we can derive a bound on the output probability of an adversary when switching the distribution of the scheme.

Corollary 2 *For any adversary A with resources T attacking $S^{\mathcal{P}}$ and any event E over A's output, denote the probability of E by $\gamma_{\mathcal{P}}$. Denote the probability of E over A's output when attacking $S^{\mathcal{Q}}$ by $\gamma_{\mathcal{Q}}$. If δ is $\sqrt{\gamma_{\mathcal{Q}}/16T}$-efficient and $\delta(\mathcal{P}_\theta, \mathcal{Q}_\theta) \leq \sqrt{\gamma_{\mathcal{Q}}/16T}$, then $16\gamma_{\mathcal{P}} \geq \gamma_{\mathcal{Q}}$.*

Proof We use Lemma 4 and set k such that $2^{k-4} = \frac{T}{\gamma_{\mathcal{Q}}}$. This implies that $\frac{T}{\gamma_{\mathcal{Q}}} \geq 2^{k-3}$ is false. Assuming towards a contradiction that $16\gamma_{\mathcal{P}} < \gamma_{\mathcal{Q}}$, we see that

$$2^{k-4} = \frac{T}{\gamma_{\mathcal{Q}}} \leq \frac{T}{16\gamma_{\mathcal{P}}}$$

contradicting Lemma 4. □

With this bound in place, we are ready for the main proof.

Proof [of Theorem 8] Fix any T^A-bounded adversary A against $S^{\mathcal{P}}$, output probability $\alpha^A_{\mathcal{P}}$ and conditional success probability $\beta^A_{\mathcal{P}}$. By assumption we have $\alpha^A_{\mathcal{P}}(2\beta^A_{\mathcal{P}}-1)^2 \leq T^A/2^\kappa$. Denote the output and conditional success probability of A against $S^{\mathcal{Q}}$ by $\alpha^A_{\mathcal{Q}}$ and $\beta^A_{\mathcal{Q}}$. Assume towards contradiction that $\alpha^A_{\mathcal{Q}}(2\beta^A_{\mathcal{Q}}-1)^2 > T^A/2^{\kappa-8}$.

First we apply Corollary 2 to obtain $\alpha^A_{\mathcal{P}} \geq 2^{-4}\alpha^A_{\mathcal{Q}}$. Note that by assumption $\sqrt{\alpha^A_{\mathcal{Q}}/16T} > 2^{(-\kappa+4)/2} > 2^{-\kappa/2} \geq \delta(\mathcal{P}_\theta, \mathcal{Q}_\theta)$ and that trivially $\sqrt{\alpha^A_{\mathcal{Q}}/16T} \leq \frac{1}{4}$.

We now consider the hypothetical modified games $\hat{S}^{\mathcal{P}}$ and $\hat{S}^{\mathcal{Q}}$, which are the same as $S^{\mathcal{P}}$ and $S^{\mathcal{Q}}$ with the only difference that the adversary has the ability to restart the game with fresh randomness at any time. Consider the adversary B against $\hat{S}$ that simply runs A until $A \neq \perp$ (restarting the game if $A = \perp$) and outputs whatever A returns. Let $\alpha = \min(\alpha^A_{\mathcal{P}}, \alpha^A_{\mathcal{Q}})$ and note that B's resources are $T^B < T^A/\alpha$, its output probability is 1 and the (conditional) success probability is $\beta^B_{\mathcal{P}} = \beta^A_{\mathcal{P}}$ (or $\beta^B_{\mathcal{Q}} = \beta^A_{\mathcal{Q}}$) if playing $\hat{S}^{\mathcal{P}}$ (or $\hat{S}^{\mathcal{Q}}$, respectively).

By the properties of δ and Δ_{SD}, we have $\beta^B_{\mathcal{P}} \geq \beta^B_{\mathcal{Q}} - \sqrt{T^B}\delta(\mathcal{P}_\theta, \mathcal{Q}_\theta)$ and so $2\beta^B_{\mathcal{P}} - 1 \geq 2\beta^B_{\mathcal{Q}} - 1 - 2\sqrt{T^B/2^\kappa}$. By assumption we also have that $2\beta^A_{\mathcal{P}} - 1 \leq \sqrt{T^A/\alpha^A_{\mathcal{P}}2^\kappa}$, which yields

$$\sqrt{\frac{T^A}{\alpha 2^\kappa}} \geq \sqrt{\frac{T^A}{\alpha^A_{\mathcal{P}}2^\kappa}} \geq 2\beta^B_{\mathcal{Q}} - 1 - 2\sqrt{\frac{T^A}{\alpha 2^\kappa}}$$

because $\beta^B_{\mathcal{P}} = \beta^A_{\mathcal{P}}$, and so

$$2\beta^A_{\mathcal{Q}} - 1 = 2\beta^B_{\mathcal{Q}} - 1 \leq 3\sqrt{\frac{T^A}{\alpha 2^\kappa}}.$$

If $\alpha^A_{\mathcal{Q}} \leq \alpha^A_{\mathcal{P}}$, then $\alpha = \alpha^A_{\mathcal{Q}}$ and the above inequality immediatly yields the contradiction. Otherwise, we can derive an upper bound on $\alpha^A_{\mathcal{P}}$ from it:

$$\alpha^A_{\mathcal{P}} \leq \frac{9T^A}{2^\kappa(2\beta^A_{\mathcal{Q}}-1)^2} < \frac{\alpha^A_{\mathcal{Q}}}{2^4}$$

where the latter inequality follows from the assumption. This contradicts our lower bound above. □

Acknowledgment

We would like to thank Krzysztof Pietrzak, Russell Impagliazzo, and Mihir Bellare for helpful discussions and pointers to relevant literature.

References

[1] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange - A new hope. In *USENIX Security Symposium*, pages 327–343. USENIX Association, 2016.

[2] S. Bai, A. Langlois, T. Lepoint, D. Stehlé, and R. Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. In T. Iwata and J. H. Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Part I*, volume 9452 of *Lecture Notes in Computer Science*, pages 3–24. Springer, Heidelberg, Nov. / Dec. 2015.

[3] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science*, pages 394–403. IEEE Computer Society Press, Oct. 1997.

[4] M. Bellare, M. Fischlin, S. Goldwasser, and S. Micali. Identification protocols secure against reset attacks. In B. Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 495–511. Springer, Heidelberg, May 2001.

[5] M. Bellare and P. Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In U. M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer, Heidelberg, May 1996.

[6] D. J. Bernstein and T. Lange. Non-uniform cracks in the concrete: The power of free precomputation. In K. Sako and P. Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part II*, volume 8270 of *Lecture Notes in Computer Science*, pages 321–340. Springer, Heidelberg, Dec. 2013.

[7] A. De, L. Trevisan, and M. Tulsiani. Time space tradeoffs for attacks against one-way functions and PRGs. In T. Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 649–665. Springer, Heidelberg, Aug. 2010.

[8] Y. Dodis and J. P. Steinberger. Message authentication codes from unpredictable block ciphers. In S. Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 267–285. Springer, Heidelberg, Aug. 2009.

[9] O. Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001.

[10] O. Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004.

[11] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *21st Annual ACM Symposium on Theory of Computing*, pages 25–32. ACM Press, May 1989.

[12] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[13] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[14] L. A. Levin. Randomness and non-determinism. *Journal of Symbolic Logic*, 58:1102–1103, 1993.

[15] D. Micciancio and M. Walter. Gaussian sampling over the integers: Efficient, generic, constant-time. In *CRYPTO (2)*, volume 10402 of *Lecture Notes in Computer Science*, pages 455–485. Springer, 2017.

[16] T. Pöppelmann, L. Ducas, and T. Güneysu. Enhanced lattice-based signatures on reconfigurable hardware. In L. Batina and M. Robshaw, editors, *Cryptographic Hardware and Embedded Systems – CHES 2014*, volume 8731 of *Lecture Notes in Computer Science*, pages 353–370. Springer, Heidelberg, Sept. 2014.

[17] T. Prest. Sharper bounds in lattice-based cryptography using the Rényi divergence. In T. Takagi and T. Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 347–374. Springer, Heidelberg, Dec. 2017.

A Proof of Theorem 1

Proof [of Theorem 1] From the definition of Y in Definition 7 we get for any $x, y \in \{0,1\}^n$ with $y \neq x$

- $\Pr[Y = \perp | X = x] = 1 - \alpha$
- $\Pr[Y = x | X = x] = \alpha\beta$
- $\Pr[Y = y | X = x] = \frac{\alpha(1-\beta)}{2^n - 1}$.

From this we compute

- $\Pr[Y = \perp] = 1 - \alpha$
- $\Pr[Y = y] = \Pr[Y = y | X = y]\Pr[X = y] + \Pr[Y = y | X \neq y]\Pr[X \neq y] = \frac{\alpha\beta}{2^n} + \frac{2^n - 1}{2^n}\frac{\alpha(1-\beta)}{2^n - 1} = \frac{\alpha}{2^n}$.

Now we calculate the conditional entropy

$$\begin{aligned}
H(X|Y) =& \sum_{x,y} \Pr[Y = y | X = x]\Pr[X = x] \log \frac{\Pr[Y = y]}{\Pr[Y = y | X = x]\Pr[X = x]} \\
=& \sum_{x} \Pr[Y = \perp | X = x]\Pr[X = x] \log \frac{\Pr[Y = \perp]}{\Pr[Y = \perp | X = x]\Pr[X = x]} \\
&+ \Pr[Y = x | X = x]\Pr[X = x] \log \frac{\Pr[Y = x]}{\Pr[Y = x | X = x]\Pr[X = x]} \\
&+ \sum_{y \neq x \wedge y \neq \perp} \Pr[Y = y | X = x]\Pr[X = x] \log \frac{\Pr[Y = y]}{\Pr[Y = y | X = x]\Pr[X = x]} \\
=& \sum_{x} \frac{1 - \alpha}{2^n} \log \frac{(1-\alpha)2^n}{1 - \alpha} + \frac{\alpha\beta}{2^n} \log \frac{\alpha 2^n}{\alpha\beta 2^n} \\
&+ (2^n - 1)\frac{\alpha(1-\beta)}{(2^n - 1)2^n} \log \frac{\alpha 2^n (2^n - 1)}{2^n \alpha(1-\beta)} \\
=& (1-\alpha)n + \alpha\beta \log \frac{1}{\beta} + \alpha(1-\beta) \log \frac{2^n - 1}{1 - \beta} \\
=& (1-\alpha)n + \alpha((1-\beta)\log(2^n - 1) + H(\mathcal{B}_\beta))
\end{aligned}$$

Finally, we compute the advantage

$$\begin{aligned}
\mathrm{adv}^A &= 1 - \frac{H(X|Y)}{n} \\
&= 1 - (1-\alpha) - \alpha\frac{(1-\beta)\log(2^n - 1) + H(\mathcal{B}_\beta)}{n} \\
&= \alpha\left(1 - \frac{(1-\beta)\log(2^n - 1) + H(\mathcal{B}_\beta)}{n}\right).
\end{aligned}$$

□

B Missing Details of Proof for Lemma 2

With the notation of Section 5.3.1, the goal of this section is to prove

$$\sum_{i=1}^{k} \mathrm{adv}_{i,i+1} \geq \frac{\alpha_1 + \alpha_k}{\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k} \mathrm{adv}_{1,k}.$$

By Equation (5)

$$\sum_{i=1}^{k} \text{adv}_{i,i+1} = \sum_{i=1}^{k} \frac{(2(\gamma_i - \gamma_{i+1}) + \alpha_{i+1} - \alpha_i)^2}{2(\alpha_i + \alpha_{i+1})}.$$

Applying the induction hypothesis, this is lower bounded by

$$f(\gamma_{k-1}) = \frac{(2(\gamma_1 - \gamma_{k-1}) + \alpha_{k-1} - \alpha_1)^2}{2(\alpha_1 + 2\sum_{i=2}^{k-2} \alpha_i + \alpha_{k-1})} + \frac{(2(\gamma_{k-1} - \gamma_k) + \alpha_k - \alpha_{k-1})^2}{2(\alpha_{k-1} + \alpha_k)}.$$

Taking f's derivative

$$f'(\gamma_{k-1}) = \frac{2(2(\gamma_{k-1} - \gamma_k) + \alpha_k - \alpha_{k-1})}{\alpha_{k-1} + \alpha_k} - \frac{2(2(\gamma_1 - \gamma_{k-1}) + \alpha_{k-1} - \alpha_1)}{\alpha_1 + 2\sum_{i=2}^{k-2} \alpha_i + \alpha_{k-1}}$$

Note that the second derivative is a positive constant, so if f has an extremum it must be a minimum, and since it is a quadratic function, it is a global minimum. Setting $f'(\gamma_{k-1}) = 0$ and solving for $2\gamma_{k-1}$, we get:

$$2\gamma_{k-1}\left(\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k\right) = 2(\gamma_1 + \alpha_{k-1} - \alpha_1)(\alpha_{k-1} + \alpha_k) + 2(\gamma_k + \alpha_k - \alpha_{k-1})\left(\alpha_1 + 2\sum_{i=2}^{k-2} \alpha_i + \alpha_k\right)$$

Plugging this into the terms of f:

$$(2(\gamma_1 - \gamma_{k-1}) + \alpha_{k-1} - \alpha_1) = \frac{2(\gamma_1 - \gamma_k) - \alpha_1 + \alpha_k)\left(\alpha_1 + 2\sum_{i=2}^{k-2} \alpha_i + \alpha_k\right)}{\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k}$$

and

$$(2(\gamma_{k-1} - \gamma_k) + \alpha_k - \alpha_{k-1}) = \frac{(2(\gamma_1 - \gamma_k) - \alpha_1 + \alpha_k)(\alpha_{k-1} + \alpha_k)}{\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k}$$

which yields that

$$\begin{aligned}
f(\gamma_{k-1}) &\geq \frac{(2(\gamma_1 - \gamma_k) - \alpha_1 + \alpha_k)^2 \left(\alpha_1 + 2\sum_{i=2}^{k-2} \alpha_i + \alpha_k\right)^2}{\left(\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k\right)^2 \left(\alpha_1 + 2\sum_{i=2}^{k-2} \alpha_i + \alpha_k\right)} \\
&\quad + \frac{(2(\gamma_1 - \gamma_k) - \alpha_1 + \alpha_k)^2 (\alpha_{k-1} + \alpha_k)^2}{\left(\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k\right)^2 (\alpha_{k-1} + \alpha_k)} \\
&= \frac{(2(\gamma_1 - \gamma_k) - \alpha_1 + \alpha_k)^2}{\left(\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k\right)^2} \left(\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k\right) \\
&= \frac{(2(\gamma_1 - \gamma_k) - \alpha_1 + \alpha_k)^2}{\left(\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k\right)} \\
&= \frac{\alpha_1 + \alpha_k}{\left(\alpha_1 + 2\sum_{i=2}^{k-1} \alpha_i + \alpha_k\right)} \text{adv}_{1,k}
\end{aligned}$$

as desired.