Does Privacy Require True Randomness?

Carl Bosley and Yevgeniy Dodis*

New York University. {bosley,dodis}@cs.nyu.edu

Abstract. Most cryptographic primitives require randomness (for example, to generate their secret keys). Usually, one assumes that perfect randomness is available, but, conceivably, such primitives might be built under weaker, more realistic assumptions. This is known to be true for many authentication applications, when entropy alone is typically sufficient. In contrast, all known techniques for achieving privacy seem to fundamentally require (nearly) perfect randomness. We ask the question whether this is just a coincidence, or, perhaps, privacy inherently requires true randomness?

We completely resolve this question for the case of (information-theoretic) private-key encryption, where parties wish to encrypt a b-bit value using a shared secret key sampled from some imperfect source of randomness $\mathscr{S}$. Our main result shows that if such n-bit source $\mathscr{S}$ allows for a secure encryption of b bits, where $b > \log n$, then one can deterministically extract nearly b almost perfect random bits from $\mathscr{S}$. Further, the restriction that $b > \log n$ is nearly tight: there exist sources $\mathscr{S}$ allowing one to perfectly encrypt $(\log n - \operatorname{loglog} n)$ bits, but not to deterministically extract even a single slightly unbiased bit.

Hence, to a large extent, *true randomness is inherent for encryption*: either the key length must be exponential in the message length b, or one can deterministically extract nearly b almost unbiased random bits from the key. In particular, *the one-time pad scheme is essentially "universal"*.

Our technique also extends to related *computational* primitives which are *perfectly-binding*, such as perfectly-binding commitment and computationally secure private- or public-key encryption, showing the necessity to *efficiently* extract almost b *pseudorandom* bits.

1 Introduction

Randomness is important in many areas of computer science. It is especially indispensable in cryptography: secret keys must be random, and many cryptographic tasks, such as public-key encryption, secret sharing or commitment, require randomness for every use. Typically, one assumes that all parties have access to a perfect random source, but this assumption is at least debatable, and the question of what kind of *imperfect random sources* can be used for various applications has attracted a lot of attention.

Extraction. The easiest such class of sources consists of *extractable* sources for which one can deterministically extract nearly perfect randomness, and then use

* Supported by NSF Grants #0515121, #0133806, #0311095.

it in any application. Although various examples of such non-trivial sources are known (see [TV00,KRVZ06] and the references therein), most natural sources, such as the so called entropy sources[1] [SV86,CG88,Zuc96], are easily seen to be non-extractable. One can then ask the natural question of whether perfect randomness is indeed inherent for the considered application, or perhaps one can do with weaker, more realistic assumptions. Clearly, the answer depends on the application.

Positive Results. For one such application domain, a series of celebrated results [VV85,SV86,CG88,Zuc96,ACRT99] showed that entropy sources are sufficient for simulating probabilistic polynomial-time algorithms — namely, problems which do not *inherently* need randomness, but which could potentially be sped up using randomization. Thus, extremely weak imperfect sources can still be tolerated for this application domain. This result was later extended to interactive protocols by Dodis et al. [DOPS04].

Moving to cryptographic applications, entropy sources are typically sufficient for authentication applications, since entropy is enough to ensure unpredictability. For example, in the non-interactive (i.e., one-message) setting Maurer and Wolf [MW97] show that, for a sufficiently high entropy rate (specifically, more than 1/2), entropy sources are indeed sufficient for unconditional one-time authentication (while Dodis and Spencer [DS02] showed that smaller rate entropy sources are not sufficient to authenticate even a single bit). Moreover, in the interactive setting, Renner and Wolf [RW03] show information-theoretic authentication protocols capable of tolerating any constant-fraction entropy rate. Finally, Dodis et al. [DOPS04] consider the existence of computationally secure digital signature (and thus also message authentication) schemes, and, under (necessarily) strong, but plausible computational assumptions, once again showed that entropy sources are enough to build such signature schemes. From a different angle, [DS02] also show that for all entropy levels (in particular, below 1/2) there exist "severely non-extractable" imperfect sources which are nevertheless sufficient for non-trivial non-interactive authentication. Thus, good sources for authentication certainly do not require perfect randomness.

Randomness for Privacy? The situation is much less clear for privacy applications, whose security definitions include some kind of indistinguishability. Of those, the most basic and fundamental is the question of (private-key) encryption, whose definition requires that the encryptions of any two messages are indistinguishable. (Indeed, this will be the subject of this work.)

With one exception (discussed shortly), all known results indicate that true randomness might be inherent for privacy applications, such as encryption. First, starting with Shannon's one-time scheme [Sha49], all existing methods for building secure encryptions schemes, as well as other privacy primitives, crucially de-

[1] Informally, entropy sources guarantee that every distribution in the family has a non-trivial amount of entropy (and possibly more restrictions), but do not assume independence between different symbols of the source. Thus, they are the most general sources one would wish to tolerate, since cryptography clearly requires entropy.

pend on perfect randomness somewhere in their design. And this is true even in the computational setting. For example, the Goldreich-Levin [GL89] reduction from unpredictability to indistinguishability, as well the the entire theory of pseudorandomness, crucially use a random seed to obtain the desired constructions. Second, attempts to build secure encryption schemes (and other privacy primitives) based on known "non-extractable" sources, such as various entropy sources, *provably failed*, indicating that such sources are indeed insufficient for privacy. For example, McInnes and Pinkas [MP90] showed that unconditionally secure symmetric encryption cannot be based on entropy sources, even if one is restricted to encrypting a single bit. This result was subsequently strengthened by Dodis et al. [DOPS04], who showed that entropy sources are not sufficient even for *computationally* secure encryption (as well as essentially any other task involving "privacy", such as commitment, zero-knowledge and others).

The only reassuring result in the other direction is the work of Dodis and Spencer [DS02], who considered the setting of symmetric encryption, where the shared secret key comes from an imperfect random source, instead of being truly random. In this setting, they constructed a particular non-extractable imperfect source, nevertheless allowing one to perfectly encrypt *a single bit*. By itself, this result is not surprising. For example, a uniform distribution on $\{0, 1, 2\}$ allows one to encrypt a bit (by addition modulo 3), but not to extract a bit, which is obvious. Indeed, the actual contribution of [DS02] was not to show that the separation between one bit encryption and extraction *exists* — as we just saw, this is trivial — but to show that a very strong separation still holds even if one additionally requires all the distributions in the imperfect source to have high entropy (in fact, very close to n). In practice, however, we typically care about encrypting considerably more than a single bit. In such cases, it is certainly unreasonable to expect that, say, encryption of b bits will necessarily imply extraction of *exactly* b bits (which was indeed disproved by [DS02] for $b = 1$). One would actually *expect* that an implication, if true, would lose at least a few bits (perhaps depending on the statistical distance ε from the uniform distribution that we want our extraction to achieve).

In particular, the results of [DS02] leave open the following extreme possibilities: (a) perhaps any source encrypting already two bits must be extractable; or (b) perhaps there exists an n-bit source allowing one to perfectly encrypt almost n bits, and yet not to extract even a single bit. Clearly, possibility (a) would strongly indicate that true randomness *is* inherent for encryption, while possibility (b) that it is *not*. As we will see shortly, both (a) and (b) happen to be false, but our point is that the results of [DS02] regarding *one-bit* encryption and extraction do not answer what we feel is the more appropriate question:

Assume an imperfect source allows for a secure private-key encryption of b bits. Does this necessarily imply one can deterministically extract at least one (and, hopefully, close to b) nearly perfect bits from this source?

OUR RESULT. We resolve the above question. Our main result shows that if an n-bit source $\mathscr{S}$ allows for a secure (and even slightly biased) encryption of

b bits, where $b > \log n$, then one can deterministically extract almost b nearly perfect random bits from $\mathscr{S}$; see Theorem 1(a) for the precise bound. Moreover, the restriction that $b > \log n$ is essentially tight: there exist imperfect sources allowing one to perfectly encrypt $b \approx \log n - \log\log n$ bits, from which one cannot deterministically extract even a single slightly unbiased (let alone random!) bit; see Theorem 1(b).[2] Hence, to a large extent, *true randomness is inherent for (information-theoretic) private-key encryption*:

Either the key length n must be exponential in the message length b, or
One can deterministically extract almost b nearly random bits from the key.

In particular, in the case when b is large enough, so that it is infeasible to sample more than 2^b (imperfect) bits for one's secret key, our result implies the following. In order to build a secure b-bit encryption scheme, one must come up with a source of randomness from which one can already deterministically extract almost b nearly random bits! Notice, since such extracted bits can then be used as a one-time pad, we get that any b-bit encryption scheme can in principle be converted to a "one-time-pad-like" scheme capable of encrypting nearly b bits! In this sense, our results show that, *for the purpose of information-theoretically encrypting a "non-trivial" number of bits, the one-time pad scheme is essentially "universal"*.

EXTENSIONS. Our result can be extended in several ways.

First, the basic extractor we construct is inefficient, even if the encryption scheme is efficient (i.e., runs in time polynomial in n). However, using the technique of Trevisan and Vadhan [TV00] (see also [DSS01,Dod00]), we can obtain the following marginally weaker result which maintains efficiency: if a source $\mathscr{S}$ enables an *efficient* encryption of $b > \log n$ bits, then there exists an *efficient* deterministic extractor allowing one to extract roughly $(b - \log n)$ nearly perfect bits from $\mathscr{S}$. Despite the small loss of $\log n$ bits, we still get the same pessimistic conclusion: unless the key is exponential in the message length, efficient encryption implies efficient extraction of nearly the same number of bits.

Second, our technique extends to computationally secure privacy primitives which are *perfectly (or statistically) binding*, which includes perfectly-binding commitment (which, therefore, must be computationally hiding) and computationally secure private- or public-key encryption. Specifically, let λ be the security parameter, $n = \mathsf{poly}(\lambda)$ be the number of random bits coming from the imperfect source $\mathscr{S}$, and assume that $\mathscr{S}$ is good enough to *efficiently* (i.e., in time polynomial in λ) implement the required *computationally secure* (but perfectly-binding) primitive on $b = \omega(\log \lambda)$ bits. Then we show that there exists an *efficient* extractor capable of extracting $b(1 - o(1))$ *pseudorandom* bits from $\mathscr{S}$. Of course, at this point one can also apply a pseudorandom generator, whose existence is typically implied by the existence of the corresponding

[2] This result is a non-trivial extension of the separation of [DS02] from 1-bit to (roughly) $(\log n)$-bit encryption. Indeed, without the entropy constraints, our proof is considerably more involved than that of [DS02]. See also Section 4.5.

computational primitive, to stretch the extracted (pseudo)randomness further by any polynomial amount. Also, since every *individual* pseudorandom bit must actually be *statistically* random (otherwise, the distinguisher succeeds by simply outputting this bit), we still get that any of the above computationally secure primitives on $b = \omega(\log \lambda)$ bits requires *at least some* nearly perfect randomness.

To summarize, non-trivial computationally secure primitives which are perfectly binding require some *efficiently extractable true randomness.*

Organization. We define the needed notation in Section 2, which also allows us to formally state our main result (Theorem 1). In Section 3 we prove that encryption of $b > \log n$ bits using an n-bit key implies extraction of roughly b random bits, and mention the "computational" extensions of this result. In Section 4, which is the main technical section, we show that encryption of up to $(\log n - \operatorname{loglog} n)$ bits does not necessarily imply extraction of even a single bit. Finally, in Section 5 we conclude and state some open problems.

2 Notation and Definitions

We use calligraphic letters, like $\mathcal{X}$, to denote finite sets. The corresponding large letter X is then used to denote a random variable over $\mathcal{X}$, while the lowercase letter x denotes a particular element from $\mathcal{X}$. $U_{\mathcal{X}}$ denotes the uniform distribution over $\mathcal{X}$. A source $\mathscr{S}$ over $\mathcal{X}$ is a set of distributions over $\mathcal{X}$. We write $X \in \mathscr{S}$ to state that $\mathscr{S}$ contains a distribution X.
The statistical distance $\mathsf{SD}(X_1, X_2)$ between two random variables X_1, X_2 is

$$\mathsf{SD}(X_1, X_2) = \frac{1}{2} \sum_{x \in \mathcal{X}} \big| \Pr[X_1 = x] - \Pr[X_2 = x] \big| \tag{1}$$

$$= \max_{\mathcal{T} \subseteq \mathcal{X}} \left(\Pr[X_1 \in \mathcal{T}] - \Pr[X_2 \in \mathcal{T}] \right) \tag{2}$$

If $\mathsf{SD}(X_1, X_2) \leq \varepsilon$, this means that no (even computationally unbounded) distinguisher D can tell apart a sample from X_1 from a sample from X_2 with an advantage greater than ε.

Definition 1. *A random variable R over $\mathcal{R}$ is ε-*fair *if $\mathsf{SD}(R, U_{\mathcal{R}}) \leq \varepsilon$. Given a source $\mathscr{S}$ over some set $\mathcal{K}$, a function $\mathsf{Ext} : \mathcal{K} \to \mathcal{R}$ is an $(\mathscr{S}, \varepsilon)$-*extractor *if for all $K \in \mathscr{S}$, $\mathsf{Ext}(K)$ is ε-fair:*

$$\mathsf{SD}(\mathsf{Ext}(K), U_{\mathcal{R}}) \leq \varepsilon \tag{3}$$

*If such Ext exists for $\mathscr{S}$, we say that $\mathscr{S}$ is $(\mathcal{R}, \varepsilon)$-*extractable. $\diamondsuit$

Definition 2. *An* encryption scheme *$\mathcal{E}$ over message space $\mathcal{M}$, key space $\mathcal{K}$ and ciphertext space $\mathcal{C}$ is a pair of algorithms $\mathsf{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ and $\mathsf{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$, which for all keys $k \in \mathcal{K}$ and messages $m \in \mathcal{M}$ satisfies $\mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m$.*

*Given a source $\mathscr{S}$ over $\mathcal{K}$, we say that the encryption scheme $\mathcal{E}$ is $(\mathscr{S}, \delta)$-*secure *if for all messages $m_1, m_2 \in \mathcal{M}$ and all distributions $K \in \mathscr{S}$ we have*

$$\mathsf{SD}(\mathsf{Enc}(K, m_1), \mathsf{Enc}(K, m_2)) \leq \delta \tag{4}$$

*If $\mathscr{S}$ admits some $(\mathscr{S},\delta)$-secure encryption $\mathcal{E}$ over $\mathcal{M}$, we say that $\mathscr{S}$ is $(\mathcal{M},\delta)$-*encryptable*. When $\delta = 0$, we say that $\mathcal{E}$ is* perfect *on $\mathscr{S}$, and $\mathscr{S}$ is perfectly encryptable (on $\mathcal{M}$).* $\diamondsuit$

Throughout we will use the following capital letters to denote the cardinalities of various sets: key set cardinality $|\mathcal{K}| = N$, message set cardinality $|\mathcal{M}| = B$, ciphertext set cardinality $|\mathcal{C}| = S$, and extraction space cardinality $|\mathcal{R}| = L$. Although our results are general, for historical reasons it is customary to translate the results into "bit-notation". To accommodate these conventions, we let $b = \log B$, $\ell = \log L$, $n = \log N$ (here and elsewhere, all the logarithms are base 2), and will use the terms "b-bit encryption", "ℓ-bit extraction" or "n-bit key" with the obvious meanings attached. Moreover, we will slightly abuse the terminology and say that a source $\mathscr{S}$ is (1) *n-bit* if it is over a set $\mathcal{K}$ and $|\mathcal{K}| = N$; (2) *(ℓ,ε)-extractable* if it is $(\mathcal{R},\varepsilon)$-extractable and $|\mathcal{R}| = L$, and (2) *(b,δ)-encryptable* if it is $(\mathcal{M},\delta)$-encryptable and $|\mathcal{M}| = B$. Clearly, when b, ℓ or n are integers, this terminology is consistent with our intuitive understanding.

With this in mind, our main result can be restated as follows:

Theorem 1. *Secure encryption of b bits with an n-bit key requires nearly perfect randomness (in fact, almost b random bits!) if and only if b is greater than $\log n$. More precisely,*

(a) $\forall\varepsilon > 0$, if $\mathscr{S}$ is (b,δ)-encryptable, and $b > \log n + 2\log\left(\frac{1}{\varepsilon}\right)$, then $\mathscr{S}$ is $\left(b - 2\log\left(\frac{1}{\varepsilon}\right), \varepsilon+\delta\right)$-extractable. Further, if the encryption scheme is efficient *(i.e., polynomial in n), then there exists an* efficient *extractor outputting $\left(b - \log n - 2\log\left(\frac{1}{\varepsilon}\right) - 2\right)$ bits within statistical distance $(\varepsilon+\delta)$ from uniform. Thus, encryption of $b > \log n$ bits implies extraction of almost b nearly perfect bits.*

(b) For any $b \le \log n - \log\log n - 2$,[3] there exists a source $\mathscr{S}$ which is $(b,0)$-encryptable, but not $(1,\varepsilon)$-extractable, where $\varepsilon = \frac{1}{2} - 2^{(2b - \frac{n}{2^b})} \ge \frac{1}{2} - \frac{1}{16n^2}$. Thus, even perfect encryption of nearly $\log n$ bits does not imply extraction of even a single slightly unbiased bit.

3 Encryption $\Rightarrow$ Extraction if $b > \log n$

In this section we prove the implication given in Theorem 1(a), which shows that encryption of b bits implies extraction of nearly b bits. Assume $\mathcal{E} = (\mathsf{Enc},\mathsf{Dec})$ is $(\mathscr{S},\delta)$-secure over message space $\mathcal{M}$, ciphertext space $\mathcal{C}$ and key space $\mathcal{K}$. For convenience, let us identify the message space $\mathcal{M}$ with $\{1,\ldots,B\}$. Also, let ℓ (to be specified later) denote the number of bits we wish to extract, $L = 2^\ell$, and $\mathcal{R}$ be an arbitrary set of cardinality L.

We start constructing the needed extractor $\mathsf{Ext} : \mathcal{K} \to \mathcal{R}$ by showing that it is sufficient to construct a good extractor $\mathsf{Ext}' : \mathcal{C} \to \mathcal{R}$ for an auxiliary source $\mathscr{S}'$, defined by

$$\mathscr{S}' = \{\mathsf{Enc}(k, U_{\mathcal{M}}) \mid k \in \mathcal{K}\}$$

[3] The formula also holds for $b = \log n - \log\log n - 1$, but yields a slightly smaller $\varepsilon = \frac{1}{2} - \frac{1}{4\log n}$.

Lemma 1. *If $\mathscr{S}'$ is (ℓ, ε)-extractable and $\mathcal{E}$ is $(\mathscr{S}, \delta)$-secure, then $\mathscr{S}$ is $(\ell, \varepsilon+\delta)$-extractable. In fact, if* Ext' *is the assumed extractor for $\mathscr{S}'$, then the following extractor* Ext *is the claimed extractor for $\mathscr{S}$:*

$$\mathsf{Ext}(k) = \mathsf{Ext}'(\mathsf{Enc}(k, 1)) \tag{5}$$

Proof. Take any distribution $K \in \mathscr{S}$, and let $p_k = \Pr[K = k]$. Also, let Ext' be the assumed $(\mathscr{S}', \varepsilon)$-extractor. Thus, $\mathsf{SD}(\mathsf{Ext}'(\mathsf{Enc}(k, U_{\mathcal{M}})),\ U_{\mathcal{R}}) \le \varepsilon$ for all $k \in \mathcal{K}$. Then, using definition of Ext in Equation (5), we have

$$\begin{aligned}
\mathsf{SD}(\mathsf{Ext}(K),\ U_{\mathcal{R}}) &= \mathsf{SD}(\mathsf{Ext}'(\mathsf{Enc}(K, 1)),\ U_{\mathcal{R}}) \\
&\le \mathsf{SD}(\mathsf{Enc}(K, 1),\ \mathsf{Enc}(K, U_{\mathcal{M}})) + \mathsf{SD}(\mathsf{Ext}'(\mathsf{Enc}(K, U_{\mathcal{M}})),\ U_{\mathcal{R}}) \\
&\le \delta + \sum_k p_k \cdot \mathsf{SD}(\mathsf{Ext}'(\mathsf{Enc}(k, U_{\mathcal{M}})),\ U_{\mathcal{R}}) \\
&\le \delta + \sum_k p_k \cdot \varepsilon \ = \ \delta + \varepsilon
\end{aligned}$$

The first inequality follows from the triangle inequality on statistical distance. The second — from the δ-security of the encryption (stating that encryption of 1 is δ-close to the encryption of a random message $U_{\mathcal{M}}$) and the convexity of statistical distance (when expanding K as the convex combination of "point" distributions). Finally, the last inequality follows from the fact that Ext' is an ε-fair extractor for $\mathscr{S}'$. $\square$

The point of this reduction (which is the only place in our argument using the δ-security of $\mathcal{E}$) is to reduce the task of constructing an extractor for our (potentially infinite) source $\mathscr{S}$ to an extractor for a source $\mathscr{S}'$ containing "only" N distributions. Moreover, every distribution $D_k \stackrel{\text{def}}{=} \mathsf{Enc}(k, U_{\mathcal{M}})$ in $\mathscr{S}'$ contains b bits of entropy. Indeed, for any $k \in \mathcal{K}$ and $m_1 \neq m_2$, we have $\mathsf{Enc}(k, m_1) \neq \mathsf{Enc}(k, m_2)$, since otherwise one would not be able to recover the message from the ciphertext.[4] Thus, each D_k is a uniform distribution on some B-element subset of the ciphertext space $\mathcal{C}$: we call such distributions *b-flat*. It turns out that this is the only thing we need to know to ensure the existence of a good extractor for $\mathscr{S}'$!

Lemma 2. *Assume $\mathscr{S}' = \{D_k \mid k \in \mathcal{K}\}$ is any collection of b-flat distributions of cardinality N over some space $\mathcal{C}$, where $b > \operatorname{loglog} N + 2\log\left(\frac{1}{\varepsilon}\right)$. Then $\mathscr{S}'$ is $(b - 2\log\left(\frac{1}{\varepsilon}\right), \varepsilon)$-extractable.*

Proof. Let $\ell = b - 2\log\left(\frac{1}{\varepsilon}\right)$, so that $L = \varepsilon^2 B$. We show that a completely *random* function $f : \mathcal{C} \to \mathcal{R}$ gives a required *deterministic* extractor Ext' with non-zero (in fact, overwhelming!) probability, implying that the claimed Ext' exists. Take

[4] This is the only place where we use the existence of the decryption algorithm. This is why our result will later extend to any perfectly (or statistically) binding primitive.

any fixed $k \in \mathcal{K}$ and any fixed subset $\mathcal{T} \subseteq \mathcal{R}$. Let $p \stackrel{\text{def}}{=} |\mathcal{T}|/|\mathcal{R}|$ be the density of $\mathcal{T}$. For any fixed f, define the quantity

$$\Delta_f(k, \mathcal{T}) \stackrel{\text{def}}{=} \Pr[f(D_k) \in \mathcal{T}] - \Pr[U_{\mathcal{R}} \in \mathcal{T}] \tag{6}$$

and let us estimate $\Pr_f[\Delta_f(k, \mathcal{T}) > \varepsilon]$ as follows. First, it is clear that $\Pr[U_{\mathcal{R}} \in \mathcal{T}] = p$. Second, assume D_k is a uniform distribution over some set $\{c_1, \ldots, c_B\} \subseteq \mathcal{C}$, and let X_m denote an indicator random variable which is 1 if and only if $f(c_m) \in \mathcal{T}$. Clearly, if f is random, we have $\Pr_f[X_m = 1] = p$. Also, letting $\hat{X} = \frac{1}{B} \cdot \sum_m X_m$ be the average of B independent indicator variables X_m, for any fixed f we get $\Pr[f(D_k) \in \mathcal{T}] = \frac{1}{B} \cdot \sum_m X_m = \hat{X}$. Thus, recalling the definition of $\Delta_f(k, \mathcal{T})$ from Equation (6), using $\mathbb{E}[\hat{X}] = p = \Pr[U_{\mathcal{R}} \in \mathcal{T}]$, and applying the standard additive Chernoff bound to $\hat{X}$, we get

$$\Pr_f[\ \Delta_f(k, \mathcal{T}) > \varepsilon\ ] = \Pr_f[\ \hat{X} - p > \varepsilon\ ] \leq e^{-2\varepsilon^2 B}$$

We now take a union bound over all $\mathcal{T} \subseteq \mathcal{R}$ and all $k \in \mathcal{K}$. Recalling definition of $\Delta_f(k, \mathcal{T})$ (Equation (6)), using $b > \operatorname{loglog} N + 2\log\left(\frac{1}{\varepsilon}\right)$ (so $N < 2^{\varepsilon^2 B}$) and $\ell = b - 2\log\left(\frac{1}{\varepsilon}\right)$ (so $2^L = 2^{\varepsilon^2 B}$), we conclude that

$$\Pr_f[\ \exists\, k, \mathcal{T} \text{ s.t. } \Pr[f(D_k) \in \mathcal{T}] - \Pr[U_{\mathcal{R}} \in \mathcal{T}] > \varepsilon\ ] \leq N \cdot 2^L \cdot e^{-2\varepsilon^2 B} = 2^{-\Omega(\varepsilon^2 B)} \ll 1$$

Thus, there exists a specific f such that $\Pr[f(D_k) \in \mathcal{T}] - \Pr[U_{\mathcal{R}} \in \mathcal{T}] \leq \varepsilon$, for *all* subsets $\mathcal{T}$ and keys k. Using the definition of statistical distance (Equation (2)), this means that $\mathsf{SD}(f(D_k), U_{\mathcal{R}}) \leq \varepsilon$ for all $k \in \mathcal{K}$, completing the proof. □

The first assertion of Theorem 1(a) follows immediately by combining Lemma 1 and Lemma 2. In the following subsections we mention the extensions to efficient extraction and other computational primitives which are perfectly-binding.

3.1 Efficient Encryption Implies Efficient Extraction

Using Lemma 1 (and, in particular, Equation (5)), we see that when the encryption algorithm Enc is efficient (i.e., runs in time polynomial in n), to construct an efficient extractor Ext for $\mathscr{S}$ it suffices to construct an efficient extractor Ext' for the source $\mathscr{S}'$ consisting of N efficiently samplable b-flat distributions $D_k = \mathsf{Enc}(k, U_{\mathcal{M}})$, where $k \in \mathcal{K}$. Unfortunately, the extractor Ext' that we built for $\mathscr{S}'$ via Lemma 2 was generally inefficient. Luckily, we can build an efficient extractor for $\mathscr{S}'$ using the technique of Trevisan and Vadhan [TV00], which was later explored in more detail by [Dod00].

The idea is to sample the function f (which will define Ext') at random from any family $\mathcal{F}_t$ of *t-wise independent functions* from $\mathcal{C}$ to $\mathcal{R}$. Recall, such families have the property that for any distinct $c_1 \ldots c_t \in \mathcal{C}$, the values $f(c_1) \ldots f(c_t)$ are random and independent from each other, if f is chosen at random from $\mathcal{F}_t$. Also, one can construct t-wise independent function families where each f can be evaluated in time polynomial in t and s, where s is the length of an element of $\mathcal{C}$. Since the encryption scheme is efficient, s is polynomial in n. Thus, as long as t is polynomial in n, every member $f \in \mathcal{F}_t$ will be efficiently computable. As was shown by [TV00,Dod00], setting $t = O(n)$ is already enough: the following Lemma (essentially from [Dod00]) is proven for self-containment and because it uses a slightly different parameter setting.

Lemma 3 ([Dod00]). *Assume* $\ell \leq b - \log n - 2\log\left(\frac{1}{\varepsilon}\right) - 2$, *and* f *is chosen at random from a family of* $2n$*-wise independent functions from* $\mathcal{C}$ *to* $\mathcal{R}$, *where* $|\mathcal{R}| = L = 2^\ell$. *Then for any collection* $\mathscr{S}' = \{D_k \mid k \in \mathcal{K}\}$ *of* b*-flat distributions of cardinality* 2^n *over* $\mathcal{C}$, $\Pr_f[\ f$ *is not an* $(\mathscr{S}', \varepsilon)$*-extractor* $] < 2^{-n}$.

Proof. The first attempt to prove this result would be to use the same proof template as in Lemma 2. Namely, to prove that for any subset $\mathcal{T} \subseteq \mathcal{R}$ and any b-flat distribution $D_k \in \mathscr{S}'$, $\Pr_f[f(D_k) \in \mathcal{T}]$ is unlikely to be different from its expectation $\Pr[U_\mathcal{R} \in \mathcal{T}]$ by more then ε. Unfortunately, with "only" a t-wise independent function f, the tail bound we would get for this undesirable event is not strong enough to take the union bound over all subsets $\mathcal{T}$ (unless t is exponential in b, which was the case when a truly random f was chosen in Lemma 2). Instead, we will only consider "singleton" sets $\mathcal{T} = \{r\}$, for $r \in \mathcal{R}$, but will prove a stronger bound on $\Delta_f(k, \{r\}) \stackrel{\text{def}}{=} (\Pr_f[f(D_k) = r] - \frac{1}{L})$ when $\ell \leq b - 2\log\left(\frac{1}{\varepsilon}\right) - \log n - 2$. This stronger bound will enable us to use Equation (1) (rather than Equation (2)) when bounding the statistical distance, and then take a union bound over "only" L singleton sets $\{r\}$ instead of 2^L subsets $\mathcal{T}$. Details follow.

We fix any $k \in \mathcal{K}$, $r \in \mathcal{R}$, and estimate $\Pr_f[\ |\Delta_f(k, \{r\})| > \frac{2\varepsilon}{L}\ ]$. We do it similarly to Lemma 2. Assume D_k is a uniform distribution over some set $\{c_1, \ldots, c_B\} \subseteq \mathcal{C}$, and let X_m denote an indicator random variable which is 1 if and only if $f(c_m) = r$. Since f is $2n$-wise independent, so are the variables $\{X_m\}$: any $2n$ of them are random and independent from each other. Let $X = \sum_m X_m$. Then $\Pr_f[X_m = 1] = \Pr_f[f(c_m) = r)] = \frac{1}{L}$, and $\mathbb{E}[X] = \frac{B}{L}$. Also,

$$\Delta_f(k, \{r\}) = \frac{1}{B} \cdot \sum_m \Pr[f(c_m) = r] - \frac{1}{L} = \frac{1}{B} \cdot (X - \mathbb{E}[X]) \tag{7}$$

Next, we use the tail bound for the sum X of t-wise independent random variables from [Dod00] (Theorem 5, page 48). It says that if $t \geq 8$ is an even integer and $\varepsilon < \frac{1}{2}$, then $\Pr(|X - \mathbb{E}[X]| \geq 2\varepsilon \cdot \mathbb{E}[X]) \leq \left(\frac{t}{4\varepsilon^2 \mathbb{E}[X]}\right)^{t/2}$. In our case, $t = 2n$, $\mathbb{E}[X] = \frac{B}{L}$, and we get by Equation (7)

$$\Pr_f\left[\ |\Delta_f(k, \{r\})| > \frac{2\varepsilon}{L}\ \right] = \Pr_f[\ |X - \mathbb{E}[X]| > 2\varepsilon \cdot \mathbb{E}[X]\ ] \leq \left(\frac{2nL}{4\varepsilon^2 B}\right)^n \leq 2^{-3n}$$

where the last inequality used $\ell \leq b - 2\log\left(\frac{1}{\varepsilon}\right) - \log n - 2$. Taking now the union bound over all $k \in \mathcal{K}$ and $r \in \mathcal{R}$, we get that with probability at least $(1 - 2^{-n})$ over the choice of f, we have $|\Delta_f(k, \{r\})| \leq \frac{2\varepsilon}{L}$ for all $k \in \mathcal{K}$ and $r \in \mathcal{R}$. In other words, for any $k \in \mathcal{K}$, $f(D_k)$ hits *every* element $r \in \mathcal{R}$ with probability between $(1 \pm 2\varepsilon)/L$. Using the definition of statistical distance in Equation (1), this implies that with probability at least $(1 - 2^{-n})$ over the choice of f, $\mathsf{SD}(f(D_k), U_{\mathcal{R}}) \leq \varepsilon$ for all $k \in \mathcal{K}$, which completes the proof. □

The above lemma immediately gives a *constructive probabilistic method* for showing the existence of an efficient *deterministic* extractor claimed by the second part of Theorem 1(a). Namely, combining Lemma 1 and Lemma 3 we get a concrete family of efficient functions most of which are guaranteed to be good deterministic extractors for $\mathscr{S}$. However, to actually fix a concrete extractor, one must either directly look at the source $\mathscr{S}$ in question, or choose the extractor *obliviously* by sampling it (using good randomness) from our family *once and for all*, or rely on non-uniformity. Alternatively, in case the length s of the ciphertext c is only slightly larger than the length b of the plaintext m, we can use an explicit deterministic extractor of Trevisan and Vadhan [TV00] for the efficiently samplable source $\mathscr{S}'$. Assuming some strong complexity assumptions (see [TV00]), this would give us an explicit way to deterministically extract $\Omega(b)$ bits, provided $s < (1 + \gamma)b$ for a small enough constant γ.

3.2 Other Perfectly-Binding Computational Primitives

We now extend our results above to handle *computationally* secure privacy primitives which are *perfectly binding*, which includes perfectly-binding commitment (which, therefore, must be computationally hiding) and computationally secure private- or public-key encryption.

Let λ be the security parameter, $n = \mathsf{poly}(\lambda)$ be the number of random bits coming from the imperfect source $\mathscr{S}$, and assume that $\mathscr{S}$ is good enough to *efficiently* (i.e., in time polynomial in λ) implement the required *computationally secure* (but perfectly-binding) primitive P on $b = \omega(\log \lambda)$ bits. Trying to unify all the above examples into one template, this means that there exists a polynomial-time algorithm Enc, which takes input $m \in \mathcal{M}$ and "randomness" $k \in \mathcal{K}$, and outputs a perfectly-binding "commitment" c to m. Here k denotes *all* the randomness needed to evaluate Enc once. For example, for secret- or public-key encryption, k includes the randomness used to sample the secret and/or public key, and, if required, the local randomness used to encrypt the message. On the other hand, for commitment, k includes the randomness used to set-up the global commitment parameters, as well as the randomness used to commit to the messages.

We assume that c is *perfectly-binding* in the following sense: for any randomness k and any $m_1 \neq m_2$, we have $\mathsf{Enc}(k, m_1) \neq \mathsf{Enc}(k, m_2)$. Notice, we do not require any efficient "decryption" algorithm recovering m from c and k (which we have in the case of encryption, but not commitment). Clearly, this includes the perfectly-binding encryption and commitment applications above. In fact, it even includes some primitives which are traditionally *not* considered perfectly-binding. For example, Pedersen's commitment [Ped91] computes $\mathsf{Enc}((r, g, h, p), m) = g^r h^m \bmod p$, where $k = (r, g, h, p)$ includes a prime p, two generators g and h of some large-enough subgroup G of $\mathbb{Z}_p^*$ of prime order q, and local randomness $r \in \mathbb{Z}_q$ used to mask the message $m \in \mathbb{Z}_q$. Traditionally, this commitment scheme is considered *perfectly-hiding* (in the setting of ideal randomness), since for any m, the value $\mathsf{Enc}((r, \ldots), m)$ is uniformly distributed for a *random* r. However, it is *perfectly-binding* according to our definition, since for any *fixed* value of r, the value of m is (inefficiently but) uniquely determined given c (and g, h, p). Thus, our notion of perfect binding is a weaker restriction than what might originally appear.

Also, in terms of computational security of P w.r.t. a source of randomness $\mathscr{S}$, we require that for any distribution $K \in \mathscr{S}$ and any $m \in \mathcal{M}$, no efficient attacker A can distinguish $\mathsf{Enc}(K, m)$ from $\mathsf{Enc}(K, U_{\mathcal{M}})$ with non-negligible probability (in λ). Finally, we say that an efficient algorithm Ext extracts ℓ *pseudorandom* bits from some source $\mathscr{S}$, if for any $K \in \mathscr{S}$ and any efficient attacker A, A has at most a negligible in λ chance of telling apart a sample of $\mathsf{Ext}(K)$ from a sample of U_ℓ. Needless to say, any ε-fair "statistical" extractor satisfies this definition *as long as ε is negligible in λ.*

With these clarifications in mind, we can generalize Lemma 1 and Lemma 3 as follows. Lemma 1 trivially extends to show that if some *efficient* Ext' extracts b' *pseudorandom* bits from the source $\mathscr{S}' \stackrel{\text{def}}{=} \{\mathsf{Enc}(k, U_{\mathcal{M}})\}$, then $\mathsf{Ext}(k) \stackrel{\text{def}}{=} \mathsf{Ext}'(\mathsf{Enc}(k, 1))$ also extracts b' *pseudorandom* bits from $\mathscr{S}$. This is the only place using the computational security of P, the rest of the proofs stays information-theoretic. As for Lemma 3, it stays the same, but we use it with any value ε which is negligible in λ, but still such that $\log\left(\frac{1}{\varepsilon}\right) = o(b)$. This is possible since we assumed that $b = \omega(\log \lambda)$. Then Lemma 3 implies the existence of an efficient extractor Ext' for $\mathscr{S}'$ (since $n = \mathsf{poly}(\lambda)$, so that one can efficiently evaluate a $2n$-wise independent function) which extracts $b - 2\log\left(\frac{1}{\varepsilon}\right) - \log n - O(1) = b - o(b) - O(\log \lambda) = b(1 - o(1))$ bits of negligible statistical distance ε from the uniform distribution, implying that these $b(1-o(1))$ bits are also pseudorandom.

To summarize, for any perfectly-binding primitive P on $b = \omega(\log \lambda)$ bits, we get the possibility of efficiently extracting $b(1 - o(1))$ pseudorandom bits.

4 Encryption $\not\Rightarrow$ Extraction if $b < \log n - \log\log n$

In this section we prove the non-implication given in Theorem 1(b), which shows that even perfect encryption of up to $(\log n - \log\log n)$ bits does not necessarily imply extraction of even a single bit. For that we need to define a specific b-bit encryption scheme $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ and a source $\mathscr{S}$, such that $\mathscr{S}$ is perfect on $\mathcal{E}$, but "non-extractable". The proof will proceed in several stages.

4.1 Defining Good Encryption $\mathcal{E}$

As the first observation, we claim that we only need to define the encryption scheme $\mathcal{E}$, and then let the source $\mathscr{S} = \mathscr{S}(\mathcal{E})$ be the set of all key distributions K making $\mathcal{E}$ perfect:

$$\mathscr{S}(\mathcal{E}) = \{K \mid \forall\, m_1, m_2 \in \mathcal{M}, c \in \mathcal{C} \Rightarrow \Pr[\mathsf{Enc}(K, m_1) = c] = \Pr[\mathsf{Enc}(K, m_2) = c]\}$$

Indeed, $\mathscr{S}(\mathcal{E})$ is the largest source which is $(b,0)$-encryptable by means of $\mathcal{E}$, so it is the hardest one to extract even a single bit from. We call distributions in $\mathscr{S}(\mathcal{E})$ *perfect* (for $\mathcal{E}$).

Although we are not required to do so, let us intuitively motivate our choice of $\mathcal{E}$ before actually defining it. For that it is very helpful to view our key space $\mathcal{K}$ in terms of the encryption scheme $\mathcal{E}$ as follows. Given any $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$, we identify each key $k \in \mathcal{K}$ with an ordered B-tuple of ciphertexts $(c_1, \ldots, c_B)$, where $\mathsf{Enc}(k, m) = c_m$. Notice, some B-tuples might not correspond to valid keys. For example, this is the case when $c_i = c_j$ for some $i \neq j$, since then encryptions of i and j are the same under this key. Intuitively, however, the larger is the set of valid B-tuples of ciphertexts, the more variety we have in the set of perfect distributions $\mathscr{S}(\mathcal{E})$, and the harder it would be to extract from $\mathscr{S}(\mathcal{E})$. This suggests that every B-tuple $(c_1, \ldots, c_B)$ of ciphertexts should correspond to a potential key, except for the necessary constraint that all the c_m's must be distinct to enable unique decryption.

A bit more formally, we assume that N can be written as $N = S(S-1)\ldots(S-B+1)$ for some integer S.[5] Then we define the set $\mathcal{C} = \{1, \ldots S\}$ to be the set of ciphertexts, $\mathcal{M} = \{1, \ldots, B\}$ be the set of plaintexts, and view the key set $\mathcal{K}$ as the set of distinct B-tuples over $\mathcal{C}$:

$$\mathcal{K} = \{k = (c_1, \ldots c_B) \mid \forall\ i \neq j \Rightarrow c_i \neq c_j\}$$

We then define $\mathsf{Enc}((c_1 \ldots c_B), m) = c_m$, while $\mathsf{Dec}((c_1, \ldots, c_B), c)$ to be the (necessarily unique) m such that $c_m = c$, and arbitrarily if no such m exists. Notice, $N < S^B$, so that $S > N^{1/B}$, which is strictly greater than B when $b < \log n - \operatorname{loglog} n$. Thus, S contains enough ciphertexts to allow for B distinct encryptions.

4.2 Excluding 0-monochromatic Distributions

Let us now take an arbitrary bit extractor $\mathsf{Ext} : \mathcal{K} \to \{0,1\}$ and argue that it is not very good on the set of perfect distributions $\mathscr{S}(\mathcal{E})$. We say that a distribution K is 0-*monochromatic* if $\Pr[\mathsf{Ext}(K) = 0] = 1$. Clearly, if the set of perfect distributions $\mathscr{S}(\mathcal{E})$ contains a 0-monochromatic distribution K, then $\mathsf{SD}(\mathsf{Ext}(K), U_1) = \frac{1}{2}$ (here and below, U_1 is the uniform distribution of $\{0,1\}$), and we would be done. Thus, for the remainder of the proof we assume that $\mathscr{S}(\mathcal{E})$ *does not contain a* 0-*monochromatic distribution*. The heart of the proof then will consist of designing a perfect encryption distribution K such that

$$\Pr[\mathsf{Ext}(K) = 0] \leq \frac{B^2}{S} \tag{8}$$

Once this is done, recalling that $S > N^{1/B} = 2^{n/2^b}$ we immediately get

$$\mathsf{SD}(\mathsf{Ext}(K), U_1) = \left|\frac{1}{2} - \Pr[\mathsf{Ext}(K) = 0]\right| \geq \frac{1}{2} - 2^{(2b - \frac{n}{2^b})}$$

[5] If not, take largest S such that $N \geq S(S-1)\ldots(S-B+1)$, and work on the subset of $N' = S(S-1)\ldots(S-B+1)$ keys, but this will not change our bounds.

as claimed by Theorem 1(b). Thus, we concentrate on building a perfect distribution K satisfying Equation (8). For that, in the following subsections we will (1) characterize perfect distributions using linear algebra; (2) use this characterization to understand the implication of the lack of 0-monochromatic perfect distributions; and, finally, (3) use this implication to construct the required perfect distribution K.

4.3 Characterizing Perfect Distributions

Let K be any distribution on $\mathcal{K}$. Given a key $k = (c_1 \dots c_B)$, let $p_k = p_{(c_1 \dots c_B)} = \Pr[K = (c_1 \dots c_B)]$ and p be the N-dimensional column vector whose k-th component is equal to p_k. Notice, being a probability vector, we know that $\sum p_k = 1$ and $p \geq 0$ (which is a shorthand for $p_k \geq 0$ for all k). Conversely, any such p defines a unique distribution K.

Assume now that K is a perfect encryption distribution for $\mathcal{E}$. This adds several more constraints on p. Specifically, a necessary and sufficient condition for a perfect encryption distribution is to require that for all $c \in \mathcal{C}$ and all $m > 1$, we have

$$\Pr[c_1 = c \mid (c_1 \dots c_B) \leftarrow K] = \Pr[c_m = c \mid (c_1 \dots c_B) \leftarrow K] \tag{9}$$

We can translate this into a linear equation by noticing that the left probability is equal to $\sum_{\{(c_1 \dots c_B):c_1=c\}} p_{(c_1 \dots c_B)}$, while the second — to $\sum_{\{(c_1 \dots c_B):c_m=c\}} p_{(c_1 \dots c_B)}$. Thus, Equation (9) can be rewritten as

$$\sum_{\{(c_1 \dots c_B):c_1=c\}} p_{(c_1 \dots c_B)} - \sum_{\{(c_1 \dots c_B):c_m=c\}} p_{(c_1 \dots c_B)} = 0 \tag{10}$$

We can then rewrite all these constraints on p into a more compact notation by defining a *constraint matrix* $V = \{v_{i,j}\}$, which has $(1 + (B-1)S)$ rows (corresponding to the constraints) and N columns (corresponding to keys). The first row of V will consist of all 1's: $v_{1,k} = 1$ for all $k \in \mathcal{K}$. This will later correspond to the fact that $\sum p_k = 1$. To define the rest of V, which would correspond to $(B-1)S$ constraints from Equation (10), we first make our notation more suggestive. We index the N columns of V by tuples $(c_1, \dots c_B)$, and the remaining $(B-1)S$ rows of V by tuples (m, c), where $m \in \{2, \dots B\}$ and $c \in \{1 \dots S\}$. Then, we define

$$v_{(m,c),(c_1,\dots,c_B)} = \begin{cases} 1, & c = c_1, \\ -1, & c = c_m, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Now, Equation (10) simply becomes $\sum_k v_{(m,c),k} \cdot p_k = 0$. Finally, we define a $(1 + (B-1)S)$-column vector e by $e_1 = 1$ and $e_i = 0$ for $i > 1$. Combining all this notation, we finally get

Lemma 4. *An N-dimensional real vector p defines a perfect distribution K for $\mathcal{E}$ if and only if $Vp = e$ and $p \geq 0$.*

4.4 Using the Lack of 0-Monochromatic Distributions

Next, we use Lemma 4 to understand our assumption that no perfect distribution K is 0-monochromatic with respect to Ext. Before that, we remind the reader of a well known Farkas Lemma (e.g., see [Str80]):

Farkas Lemma. *For any matrix A and column vector e, the linear system $Ax = e$ has no solution $x \geq 0$ if and only if there exists a row vector y s.t. $yA \geq 0$ and $ye < 0$.*

Now, let $Z = \{k \mid \mathsf{Ext}(k) = 0\}$ be the set of "0-keys" under Ext, and let A denote $(1 + (B-1)S) \times |Z|$-matrix equal to the constraint matrix V restricted its $|Z|$ columns in Z. Take any real vector p such that $p_k = 0$ for all $k \notin Z$. By Lemma 4, p corresponds to a (necessarily 0-monochromatic) perfect distribution K if and only if $Vp = e$ and $p \geq 0$. But since $p_k = 0$ for all $k \notin Z$, the above conditions are equivalent to saying that the $|Z|$-dimensional restriction $x = p|_Z$ of p to its coordinates in Z satisfies $Ax = e$ and $x \geq 0$. Conversely, any x satisfying the above constraints defines a 0-monochromatic perfect distribution p by letting $p|_Z = x$ and $p_k = 0$ for $k \notin Z$.

Thus, Ext defines no 0-monochromatic perfect distributions if and only if the constraints $Ax = e$ and $x \geq 0$ are unsatisfiable. But this is exactly the precondition to the Farkas' Lemma above! Using the Farkas Lemma on our A and e, we get the existence of the $(1 + (B-1)S)$-dimensional row vector y such that $yA \geq 0$ and $ye < 0$. Just like we did for the rows of V, we denote the first element of y by y_1, and use the notation $y_{(m,c)}$ to denote the remaining elements of y. We now translate the constraints $yA \geq 0$ and $ye < 0$ using our specific choices of A and e.

Notice, since $e_1 = 1$ and $e_i = 0$ for $i > 1$, it means that $ye = y_1$, so the constraint that $ye < 0$ is equivalent to $y_1 < 0$. Next, recalling that A is just the restriction of V to its columns in Z, and that the first row of V is the all-1 vector, we get that $yA \geq 0$ is equivalent to saying that for all $(c_1, \ldots, c_B) \in Z$ we have

$$y_1 + \sum_{m>1} \sum_{c} y_{(m,c)} \cdot v_{(m,c),(c_1,\ldots,c_B)} \geq 0 \tag{12}$$

Notice, since $y_1 < 0$, this equation implies that the double sum above is *strictly* greater than 0. Thus, recalling the definition of $v_{(m,c),(c_1,\ldots,c_B)}$ given in Equation (11), we conclude that for all $k = (c_1, \ldots, c_B)$, such that $\mathsf{Ext}(k) = 0$, we have

$$\sum_{m>1} \left(y_{(m,c_1)} - y_{(m,c_m)} \right) > 0 \tag{13}$$

The last equation finally allows us to derive the implication we need:

Theorem 2. *Assume Ext defines no 0-monochromatic perfect distributions. Then there exist real numbers $\left\{ y_{(m,c)} \mid m \in \{2 \ldots B\}, c \in \{1 \ldots S\} \right\}$ such that the following holds. If a key $k = (c_1, \ldots, c_B)$ is such that*

$$y_{(m,c_1)} - y_{(m,c_m)} \leq 0 \qquad \text{for all } m > 1, \tag{14}$$

then $\mathsf{Ext}(k) = 1$.

Proof. Summing Equation (14) for all $m > 1$ we get a contradiction to Equation (13), which means that $\mathsf{Ext}(k) \neq 0$; i.e., $\mathsf{Ext}(k) = 1$. □

4.5 Developing Intuition: Special Case $b = 1$

To get some intuition, we take a momentary detour and consider the special case $b = 1$, therefore reproving the result of [DS02]. Theorem 2 tells us that if Ext cannot be fixed to 0, there exists real numbers $y_1 \dots y_S$ such that $y_i \leq y_j$ implies that the key $k = (i, j)$ gets mapped to 1 by Ext. Thus, by rearranging the y's in the non-decreasing order $y_1 \leq y_2 \leq \dots \leq y_S$, we get that $\mathsf{Ext}((i, j)) = 1$ for any $i < j$. In particular, the uniform distribution on S keys $\{(1, 2), (2, 3), \dots, (S - 1, S), (S, 1)\}$ is easily seen to define a perfect encryption distribution K (as both $\mathsf{Enc}(K, 1)$ and $\mathsf{Enc}(K, 2)$ sample a uniformly random ciphertext) at most one of whose components — the key $(S, 1)$ — could conceivably get mapped to 0 by Ext. Thus, $\Pr[\mathsf{Ext}(K) = 0] \leq 1/S$, showing (even stronger) Equation (8) and thus completing this special case.

Interestingly, Dodis and Spencer [DS02] used a simpler "graph-theoretic" method to show the existence of exactly the same perfect distribution K as above. They viewed ciphertexts as vertices of the complete directed graph G on S vertices, and keys $k = (c_1, c_2)$ (where $c_1 \neq c_2$) — as directed edges connecting $c_1 = \mathsf{Enc}(k, 1)$ to $c_2 = \mathsf{Enc}(k, 2)$. With this notation, it is easy to see that a uniform distribution on any cycle in this graph defines a perfect encryption distribution. Now, considering first 2-cycles $\{(c_1, c_2), (c_2, c_1)\}$, the fact that none of them is 0-monochromatic implies that at least one of $\mathsf{Ext}((c_1, c_2)) = 1$ or $\mathsf{Ext}((c_2, c_1)) = 1$ is true, for any $c_1 \neq c_2$. Taking one such edge from every 2-cycle yields what is called a *tournament* graph, every one of whose edges extracts to 1. Now, a well known (and simple to prove) result in graph theory states that every tournament graph has a Hamiltonian path. In other words, there exists an ordering of ciphertexts $c_1 \dots c_S$ such that every edge (c_i, c_j) belongs to the 1-monochromatic tournament subgraph whenever $i < j$; i.e., $\mathsf{Ext}((c_i, c_j)) = 1$ if $i < j$. Completing this Hamiltonian path to a Hamiltonian cycle (by adding the edge (c_S, c_1)) yields the same kind of perfect distribution K we built earlier using Theorem 2.

Unfortunately, it seems hard to extend this graph-theoretic argument to "hypergraphs" corresponding to $b > 1$. Instead, we chose to rely on linear algebra (i.e., Theorem 2) to get a better handle on the problem. Still, our proof below for general $b > 1$ is quite more involved than the proof above for $b = 1$.

4.6 Building Non-Extractable yet Perfect K

Returning to the general case, we build a special perfect distribution K which contains many keys satisfying Equation (14), meaning that $\mathsf{Ext}(K)$ is very biased towards 1. We will construct such K having a very special form below.

Definition 3. *Assume $\pi_1, \ldots, \pi_d : \mathcal{C} \to \mathcal{C}$ are d permutations over the ciphertext space $\mathcal{C} = \{1 \ldots S\}$. We say that $\pi_1, \ldots, \pi_d$ are* d-valid *if for every $c \in C$, and distinct $i, j \in \{1 \ldots d\}$, we have $\pi_i(c) \neq \pi_j(c)$.* $\diamondsuit$

The reason for this terminology is the following. Given any B-valid $\pi_1, \ldots, \pi_B$, where recall that $B = |\mathcal{M}|$, we can define S valid keys $k_1, \ldots, k_S \in \mathcal{K}$ by $k_c = (\pi_1(c), \ldots, \pi_B(c))$, where the B-validity constraint precisely ensures that all the B ciphertexts inside k_c are distinct, so that k_c is a legal key in $\mathcal{K}$. Now, we denote by $K_{(\pi_1,\ldots,\pi_B)}$ the uniform distribution over these S keys $k_1, \ldots, k_S$.

Lemma 5. *If $\pi_1, \ldots, \pi_B$ are B-valid permutations, then $K_{(\pi_1,\ldots,\pi_B)}$ is a perfect encryption distribution.*

Proof. For any message m, $\mathsf{Enc}(K_{(\pi_1,\ldots,\pi_B)}, m)$ is equivalent to outputting $\pi_m(U_{\mathcal{C}})$, where $U_{\mathcal{C}}$ is the uniform distribution over $\mathcal{C}$. Since each π_m is a permutation over $\mathcal{C}$, this is equivalent to $U_{\mathcal{C}}$. Thus, encryption of every message m yields a truly random ciphertext $c \in \mathcal{C}$, which means that $K_{(\pi_1,\ldots,\pi_B)}$ is perfect. $\square$

Choosing Good Permutations. We will construct our perfect distribution $K = K_{(\pi_1,\ldots,\pi_B)}$ by carefully choosing a B-valid family $(\pi_1, \ldots, \pi_B)$ such that $\mathsf{Ext}(K)$ is very biased towards 1. We start by choosing π_1 to be the identity permutation $\pi_1(c) = c$ (for all c), and proceed by defining $\pi_2 \ldots \pi_B$ iteratively. After defining each π_d, we will maintain the following invariants which clearly hold for the base case $d = 1$:

(i) $\pi_1, \ldots, \pi_d$ are d-valid.
(ii) There exists a large set T_d of "good" ciphertexts (where, initially, $T_1 = \mathcal{C}$) of size $q_d > S - d^2$, which satisfies the following equation for all $c \in T_d$ and $1 < m \leq d$:[6]

$$y_{(m,c)} - y_{(m,\pi_m(c))} \leq 0 \tag{15}$$

Now, assuming inductively that we have defined $\pi_1 = id, \pi_2, \ldots, \pi_d$ which satisfy properties (i) and (ii) above, we will construct π_{d+1} still satisfying (i) and (ii).

This inductive step is somewhat technical, and we will come back to it in the next subsections. But first, assuming it is true, we show that we can easily finish our proof. Indeed, we apply the induction for $B - 1$ iterations and get B permutations $\pi_1, \ldots, \pi_B$ satisfying properties (i) and (ii) above. Then, property (i) and Lemma 5 imply that $K_{(\pi_1,\ldots,\pi_B)}$ is a perfect encryption distribution. On the other hand, property (ii) and the definition of $k_c = \{c, \pi_2(c), \ldots, \pi_B(c)\}$ imply that any key $k_c \in T_B$ satisfies Equation (14). Thus, by Theorem 2 we get that $\mathsf{Ext}(k_c) = 1$ for every $c \in T_B$. Since, $|T_B| > S - B^2$, we get that at most B^2 out of S keys k_c extract to 0. Thus, since $K_{(\pi_1,\ldots,\pi_B)}$ is uniform over its S keys, we get

$$\Pr[\mathsf{Ext}(K_{(\pi_1,\ldots,\pi_B)}) = 0] \leq \frac{B^2}{S}$$

which shows Equation (8) and completes our proof (modulo the inductive step).

[6] To get some intuition, we will see shortly that "good" ciphertexts c will lead to keys k_c satisfying Equation (14), so that $\mathsf{Ext}(k_c) = 1$ by Theorem 2.

4.7 Preparing for Induction: Detour to Matchings

Before doing the inductive step, we recall some basic facts about bipartite graphs, which we will need soon. A (balanced) bipartite graph G is given by two vertex sets L and R of cardinality S and an edge set $E = E(G) \subseteq L \times R$. A *matching* P in G is a subset of node-disjoint edges of E. P is *perfect* if $|P| = S$. In this case every $i \in L$ is matched to a unique $j \in R$ and vice versa.

We say that a subset $L' \subseteq L$ is *matchable* (in G) if there exists a matching P containing L' as the set of its endpoints in L. In this case we also say that L' is *matchable with* R', where $R' \subseteq R$ is the set of P's endpoints in R. (Put differently, L' is matchable with R' precisely when the subgraph induced by L' and R' contains a perfect matching.) The famous Hall's marriage theorem gives a necessary and sufficient condition for L' to be matchable.

Hall's Marriage Theorem. *L' is matchable if and only if every subset A of L' contains at least $|A|$ neighbors in R. Notationally, if $\mathcal{N}(A)$ denotes the set of elements in R containing an edge to A, then L' is matchable iff $|\mathcal{N}(A)| \geq |A|$, for all $A \subseteq L'$.*

We will only use the following two special cases of Hall's theorem.

Corollary 1. *Assume every vertex $v \in L \cup R$ has degree at least $S - d$: $\mathsf{deg}_G(v) \geq S - d$. Then, for any $L' \subset L$ and $R' \subset R$ of cardinality $2d$, we have that L' is matchable with R'.*

Proof. Let us consider the $2d \times 2d$ bipartite subgraph G' of G induced by L' and R'. Clearly, that every vertex $v \in L' \cup R'$ has degree at least d in G', since each such v is not connected to at most d opposite vertices in the entire G, let alone G'. We claim that L' meets the conditions of the Hall's theorem in G'. Consider any non-empty $A \subseteq L'$. If $|A| \leq d$, then any vertex v in A had $\mathsf{deg}_{G'}(v) \geq d \geq |A|$ neighbors, so $|\mathcal{N}(A) \geq |A|$. If $d < |A| \leq 2d$, let us assume for the sake of contradiction that $|\mathcal{N}(A)| < |A|$. Consider now any vertex $v \in R \backslash \mathcal{N}(A)$. Such v exists as $|\mathcal{N}(A)| < |A| \leq 2d = |R'|$. Then no element in A can be connected to v, since $v \notin \mathcal{N}(A)$. Thus, the degree of v can be at most $2d - |A| < d$, which is a contradiction. $\square$

Corollary 2. *Assume L contains a subset $L' = \{c_1, \ldots, c_\ell\}$ such that $\mathsf{deg}_G(c_i) \geq i$, for $1 \leq i \leq \ell$. Then L' is matchable in G. In particular, G contains a matching of size at least ℓ.*

Proof. We show that L' satisfies the conditions of Hall's theorem. Assume $A = \{c_{i_1}, \ldots, c_{i_a}\}$, where $1 \leq i_1 < i_2 < \ldots < i_a \leq \ell$. Notice, this means $i_j \geq j$ for all j. Then the neighbors of A at least include the neighbors of i_a, so that $|\mathcal{N}(A)| \geq \mathsf{deg}_G(c_{i_a}) \geq i_a \geq a = |A|$. $\square$

4.8 Mapping Induction into a Matching Problem

We return to our induction. Recall, we are given permutations $\pi_1 = id, \pi_2, \ldots, \pi_d$ satisfying properties (i) and (ii), and need to construct π_{d+1} also satisfying properties (i) and (ii). We translate this task into some graph matching problem, starting with the property (i) first.

For every $c \in C$, we define the "forbidden" set $F_c = \{c, \pi_2(c), \dots, \pi_d(c)\}$. Then, the $(d+1)$-validity constraint (i) is equivalent to requiring $\pi_{d+1}(c) \notin F_c$ for all $c \in \mathcal{C}$. Next we define a bipartite "constraint graph" G on two copies L and R of $\mathcal{C}$ containing all the non-forbidden edges: $(c, c') \in E(G)$ if and only if $c' \notin F_c$. We observe two facts about G. First,

Lemma 6. *Every vertex $v \in L \cup R$ has degree at least $S-d$: $\mathsf{deg}_G(v) \geq S-d$. In particular, by Corollary 1 every two $2d$-element subsets of L and R are matchable with each other in G.*

Proof. The claim is obvious for $v \in L$ as $|F_v| = c$. It is also true for $v \in R$, since any value $v \in R$ is forbidden by exactly d (necessarily distinct) elements $v, \pi_2^{-1}(v), \dots, \pi_d^{-1}(v)$. □

Second, any perfect matching P of G uniquely defines a permutation π on S elements such that $P = \{(c, \pi(c))\}_{c \in L}$. Since, by definition, $\pi(c) \notin F_c$, it is clear that this π will always satisfy constraint (i). Thus, we only need to find a perfect matching P for G which will define a permutation π_{d+1} satisfying condition (ii).

Notice, our inductive assumption implies the existence of a subset T_d of L (recall, L is just a copy of $\mathcal{C}$) of size $q_d > S - d^2$ such that Equation (15) is satisfied for all $c \in T_d$ and $1 < m \leq d$. Irrespective of the permutation π_{d+1} we will construct later, we will restrict T_{d+1} to be a *subset* of T_d. This means that Equation (15) will already hold for all $c \in T_{d+1}$ and $1 < m \leq d$. Thus, we will only need to ensure this equation for $m = d+1$; i.e., that for all $c \in T_{d+1}$

$$y_{(d+1,c)} - y_{(d+1,\pi_{d+1}(c))} \leq 0 \tag{16}$$

This constraint motivates us to define a subgraph G' of our constraint graph G as follows. As edge $(c, c') \in E(G')$ if and only if $(c, c') \in E(G)$ (i.e., $c' \notin F_c$) and $y_{(d+1,c)} - y_{(d+1,c')} \leq 0$. In other words, we only leave edges (c, c') which will satisfy Equation (16) if we were to define $\pi_{d+1}(c) = c'$. The key property of G' turns out to be

Lemma 7. *G' contains a matching P' of size at least $S - d$.*

Proof. We will use Corollary 2. Let us sort the vertices $v_1 \dots v_S$ of L and R in the order of non-decreasing $y_{(d+1,\cdot)}$ values; i.e.

$$y_{(d+1,v_1)} \leq y_{(d+1,v_2)} \leq \dots \leq y_{(d+1,v_S)}$$

Then, the edge (v_i, v_j) satisfies $y_{(d+1,v_i)} - y_{(d+1,v_j)} \leq 0$ whenever $i \leq j$. Thus, such (v_i, v_j) belongs to G' if and only if it also belongs to the larger constraint graph G; i.e., $v_j \notin F_{v_i}$. But since each v_i has at most d forbidden edges in G, and $|\{j \mid j \geq i\}| = S - i + 1$, we have that $\mathsf{deg}_{G'}(v_i) \geq (S - i + 1) - d$. In particular, $\mathsf{deg}_{G'}(v_{S-d}) \geq 1, \dots, \mathsf{deg}_{G'}(v_1) \geq S - d$. By Corollary 2, $\{v_{S-d}, \dots, v_1\}$ is matchable in G', completing the proof. □

4.9 Finishing the Proof

Finally, we can collect all the pieces together and define a good matching P in G (corresponding to π_{d+1}). With an eye on satisfying property (ii), we start with a large (but not yet perfect) matching P' of G' of size at least $S-d$, guaranteed by Lemma 7. Ideally, we would like to extend P' to some perfect matching in the full graph G, by somehow matching the vertices currently unmatched by P'. Unfortunately, we do not know how to argue that such extension is possible, since there are at most d vertices unmatched, and we can only match arbitrary sets of size at least $2d$ by Lemma 6. So we simply take an arbitrary sub-matching P'' of P' of size $S-2d$, just throwing away any $|P'|-(S-2d)$ edges of P'.

Notice, P'' is also a matching of G which has exactly $2d$ unmatched vertices on both sides. By Lemma 6, we know that we can always match these missing vertices, and get a perfect matching P of the entire G. We finally claim that this perfect matching P defines a permutation π_{d+1} on $\mathcal{C}$ satisfying properties (i) and (ii).

Property (i) is immediate since P is a perfect matching of G. As for property (ii), let L' denote the $S-2d$ endpoints of P'' in L. Now, every $c \in L'$ satisfies Equation (16), since this is how the graph G' was defined and $(c, \pi_{d+1}(c)) \in P'' \subseteq E(G')$. Thus, we can inductively define $T_{d+1} = T_d \cap L'$ and have T_{d+1} satisfy property (ii). We only need to argue that T_{d+1} is large enough, but this is easy. Since L' misses only $2d$ ciphertexts, we get by induction that

$$|T_{d+1}| \geq |T_d| - 2d > S - d^2 - 2d > S - (d+1)^2$$

completing the induction and the whole proof.

5 Conclusions and Open Problems

We study the question of whether true randomness is inherent for achieving privacy, and show a largely positive answer for the case of information-theoretic private-key encryption, as well as computationally secure perfectly-binding primitives. The most interesting question is to study other privacy primitives (either information-theoretic or computational) not immediately covered by our technique. For example, what about 2-out-2 secret sharing (which is strictly implied by private-key encryption [DPP06]) or computationally binding commitment schemes? Do they still require true randomness?

More generally, we hope that our result and techniques will stimulate further interest in understanding the extent to which cryptographic primitives can be based on imperfect randomness.

Acknowledgments. We would like to thank Amit Sahai, Salil Vadhan and the anonymous referees for suggesting most of the "computational" extensions of our result. We would also like to thank Shien Jin Ong and Salil Vadhan for suggesting to use a better Chernoff bound in the proof of Theorem 1(a).

References

[ACRT99] Alexander Andreev, Andrea Clementi, Jose Rolim, and Luca Trevisan. Dispersers, deterministic amplification, and weak random sources. *SIAM J. on Computing*, 28(6):2103–2116, 1999.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM J. on Computing*, 17(2):230–261, 1988.

[Dod00] Yevgeniy Dodis. Exposure-Resilient Cryptography (PhD Thesis). *MIT PhD Thesis*, 2000.

[DOPS04] Yevgeniy Dodis, Shien Jin Ong, Manoj Prabhakaran, and Amit Sahai. On the (im)possibility of cryptography with imperfect randomness. In *Proc. 45th IEEE FOCS*, pages 196–205, 2004.

[DPP06] Yevgeniy Dodis, Krzysztof Pietrzak and Bartosz Przydatek. Separating Sources for Encryption and Secret-Sharing. In *Proc. Theory of Cryptography Conference (TCC)*, pages 601–616, 2006.

[DS02] Yevgeniy Dodis and Joel Spencer. On the (non-)universality of the one-time pad. In *Proc. 43rd IEEE FOCS*, pages 376–388, 2002.

[DSS01] Yevgeniy Dodis, Amit Sahai, and Adam Smith. On perfect and adaptive security in exposure-resilient cryptography. In *Proc. EUROCRYPT'01*, pages 301–324, 2001.

[GL89] Oded Goldreich and Leonid Levin. A Hard-Core Predicate for all One-Way Functions. In *Prof. STOC*, pp. 25–32, 1989.

[KRVZ06] Jesse Kamp, Anup Rao, Salil Vadhan and David Zuckerman. Deterministic extractors for small-space sources. In *Proc of STOC*, pp. 691–700, 2006.

[MP90] James L. McInnes and Benny Pinkas. On the impossibility of private key cryptography with weakly random keys. In *Proc. CRYPTO'90*, pages 421–436, 1990.

[MW97] Ueli Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In *Proc. CRYPTO'97*, pages 307–321, 1997.

[Ped91] Torben P. Pedersen Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *Proc. of CRYPTO*, pp. 129–140, 1991.

[RW03] Renato Renner and Stefan Wolf. Unconditional authenticity and privacy from an arbitrary weak secret. In *Proc. CRYPTO'03*, pages 78–95, 2003.

[SV86] Miklos Santha and Umesh V. Vazirani. Generating quasi-random sequences from semi-random sources. *JCSS*, 33(1):75–87, 1986.

[Sha49] Claude Shannon. Communication Theory of Secrecy systems. In *Bell Systems Technical J.*, 28:656–715, 1949.

[Str80] Gilbert Strang. Linear Algebra and Its Applications. *Academic Press*, London, 1980.

[TV00] Luca Trevisan and Salil Vadhan. Extracting randomness from samplable distributions. In *Proc. 41st IEEE FOCS*, pages 32–42, 2000.

[vN51] John von Neumann. Various techniques used in connection with random digits. *National Bureau of Standards, Applied Mathematics Series*, 12:36–38, 1951.

[VV85] Umesh V. Vazirani and Vijay V. Vazirani. Random polynomial time is equal to slightly-random polynomial time. In *Proc. 26th IEEE FOCS*, pages 417–428, 1985.

[Zuc96] David Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16(4/5):367–391, 1996.