On the Randomness Requirements for Privacy

by

Carl Bosley

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

September 2010

Yevgeniy Dodis

To my family

Abstract

Most cryptographic primitives require randomness (for example, to generate 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 achievable 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 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 determin-

istically 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 primitives which are sufficiently *binding* and *hiding*, including *computationally* secure commitments and public-key encryption.

Contents

Chapter 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 substantial attention.

Extraction. The easiest such class of sources consists of *extractable* sources for which one can deterministically extract nearly perfect randomness, and then use it in any application. Although various examples of such non-trivial sources are known [53, 25, 9, 34, 15, 8, 1, 12, 21, 32, 50], most natural sources, such as the so called entropy sources[1] [47, 14, 54], are easily seen to be non-extractable. One can then ask the natural question of whether perfect randomness is indeed inherent

[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.

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 [52, 47, 14, 54, 3] 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. [19]. This line of work led to the introduction, by Nisan and Zuckerman [38], of the seeded extractor, which uses a short random "seed" of truly random bits to extract randomness from the source. If the seed is small enough, it is possible to enumerate through all random seeds and run the extractor on each.

Unfortunately, this is not enough for cryptography in general. For example, we cannot encrypt by sending a large collection of ciphertexts, only half of which hide the secret. Luckily, though, 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 [35] 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 [22] showed that smaller rate entropy sources are not sufficient to authenticate even a single bit). Dodis et al. [19] 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, [22] 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, such as our encryption example above, 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 [48], all existing methods for building secure encryptions schemes, as well as other privacy primitives, crucially depend on perfect randomness somewhere in their design. And this is true even in the computational setting. For example, the Goldreich-Levin [26] 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 [36] 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. [19], 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 [22], 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 [22] 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 [22] 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 [22] 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 [22] 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?

1.1 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 $b \approx \log n - \operatorname{loglog} n$ 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 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

[2]This result is a non-trivial extension of the separation of [22] from 1-bit to (roughly) $(\log n)$-bit encryption. Indeed, without the entropy constraints, our proof is considerably more involved than that of [22]. See also Section 4.5.

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 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 [50] (see also [21, 16]), 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, while the basic construction applies to information-theoretic private-key encryption, the technique extends to computationally secure privacy primitives which are sufficiently *binding* and *hiding*, which includes computationally secure commitments and computationally secure private- or public-key encryption. For example, if $\mathscr{S}$ allows an *efficient* computationally secure encryption of $b > \log n$ bits, then there exists an *efficient* deterministic extractor which outputs almost $b - \log n$ *pseudo-random* bits from $\mathscr{S}$.

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

1.2 Other Models

Privacy Amplification. The goal of Privacy Amplification, first described by Bennett, Brassard, and Robert [8], is to allow two parties holding correlated weak sources to perform key agreement. Unlike in our setting, they assume the use of local *perfect* non-secret randomness, which can be used as a seed to a strong extractor. [8] assumed access to an authenticated public channel, but Renner and Wolf [42] later showed that it is not necessary: an insecure channel is sufficient. The results of [42] were further improved by [13, 23, 33].

Leakage Resilience. The area of Leakage-Resilient Cryptography [46, 24, 2] is concerned with developing cryptographic primitives secure against arbitrary side-channel attacks, where the only limit on the attacker is an upper bound on the total entropy revealed to the attacker. Clearly, perfect randomness is available in this setting, but (parts of it) can leak to the attacker. Once again, this makes this setting different from our setting.

Multi-Source extractors. A variety of extractors have been constructed and analyzed for the case where the source consists of multiple *independent* entropy sources. Two-source extractors were first constructed by Chor and Goldreich [14]. Dodis and Oliveira [18] and Dodis et al [17] constructed strong two-source extractors, for which the extracted value is statistically independent from one of the sources, and therefore can be reused as a seed to a seeded extractor. Their constructions require the source to have rate at least $1/2$. Multi-source extractor constructions were subsequently improved by many works, including [4, 10, 5, 6, 40, 41].

Network Extractors. Sudan et al [27] construct protocols for Byzantine agreement in which parties have access to *independent* weak sources. This work

was further improved by Kalai et al. [30, 29], who construct network extractors, which allow parties with access to independent weak sources to extract private randomness. As we saw above, multiple independent sources are extractable. The challenge in this case is to tolerate a subset of dishonest players.

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, 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.

Chapter 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. When X is clear from context, we let $p_x = \Pr[X = x]$ denote the probability of sampling element x from distribution X. We denote the expected value by $\mathbb{E}$.

The Rényi entropy [45] of order α is defined for $\alpha \in (0,1) \cap (1,\infty)$ as $H_\alpha(X) = \frac{1}{1-\alpha} \log \left(\sum_{x \leftarrow X} p_x^\alpha\right)$. [1] We will be particularly interested in the Rényi entropy of order ∞, also known as the *min-entropy*, which is defined by $H_\infty(X) = \lim_{\alpha \to \infty} H_\alpha(X) = -\log \max_{x \in X} \Pr[X = x]$). We extend the notion of entropy to sources by defining $H_\alpha(\mathscr{S}) = \min_{X \in \mathscr{S}} H_\alpha(X)$.

We need a definition of the distance between two random variables.

Definition 1 The statistical distance $\mathsf{SD}\left(X_1,\, X_2\right)$ between two random variables

[1] Rényi entropy can be defined through limit arguments at $\alpha \in \{0, 1, \infty\}$. For $\alpha = 1$, the Rényi entropy is equivalent to Shannon entropy.

X_1, X_2 is

$$\begin{aligned} \mathsf{SD}\left(X_1,\, X_2\right) &= \frac{1}{2}\sum_{x\in\mathcal{X}} \left|\, \Pr[X_1 = x] - \Pr[X_2 = x]\right| && (2.1)\\ &= \max_{\mathcal{T}\subseteq\mathcal{X}} \left(\Pr[X_1 \in \mathcal{T}] - \Pr[X_2 \in \mathcal{T}]\right) && (2.2) \end{aligned}$$

If $\mathsf{SD}\left(X_1,\, X_2\right) \leq \varepsilon$, 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. $\diamondsuit$

We will use the following well-known fact.

Fact 1 *If f is a deterministic function, then for all X_1, X_2,*

$$\mathsf{SD}\left(f(X_1),\, f(X_2)\right) \leq \mathsf{SD}\left(X_1,\, X_2\right). \tag{2.3}$$

We use statistical distance to define a notion of randomness as follows.

Definition 2 A random variable R over $\mathcal{R}$ is ε-*random* if $\mathsf{SD}\left(R,\, U_{\mathcal{R}}\right) \leq \varepsilon$. 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-random:

$$\mathsf{SD}\left(\mathsf{Ext}(K),\, U_{\mathcal{R}}\right) \leq \varepsilon \tag{2.4}$$

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

We first define the security of an encryption algorithm.

Definition 3 An algorithm $\mathsf{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ is $(\delta, \mathscr{S})$-*hiding* if for all messages

$m_1, m_2 \in \mathcal{M}$ and all distributions $K \in \mathscr{S}$ we have

$$\mathsf{SD}\left(\mathsf{Enc}(K, m_1),\, \mathsf{Enc}(K, m_2)\right) \leq \delta \tag{2.5}$$

If $\delta = 0$, we say that Enc is *perfectly-hiding*. $\diamondsuit$

Using this definition, we can now define secure encryption schemes.

DEFINITION 4 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 $\mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m$. We say an encryption scheme $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ is $(\delta, \mathscr{S})$-*secure* if Enc is $(\delta, \mathscr{S})$-hiding.

If $\mathscr{S}$ admits some $(\delta, \mathscr{S})$-secure encryption scheme $\mathcal{E}$ we say that $\mathscr{S}$ is $(\delta, \mathcal{M})$-*encryptable*. When $\delta = 0$, we say that Enc 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 $(b - 2\log\left(\frac{1}{\varepsilon}\right), \varepsilon + \delta)$-extractable. Further, if the encryption scheme is efficient *(i.e., polynomial in n), then there exists an* efficient *extractor outputting $(b - \log n - 2\log\left(\frac{1}{\varepsilon}\right) - 2)$ 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 $b \leq \log n - \operatorname{loglog} n - 2$,[2] there exists a source $\mathscr{S}$ which is $(b, 0)$-encryptable, but not $(1, \varepsilon)$-extractable, where $\varepsilon = \frac{1}{2} - 2^{(2b - \frac{n}{2^b})} \geq \frac{1}{2} - \frac{1}{16n^2}$. Thus, even perfect encryption of nearly $\log n$ bits does not imply extraction of even a single slightly unbiased bit.

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

Chapter 3

Encryption Implies 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 we are given $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ over message space $\mathcal{M} = \{1, \ldots, B\}$, key space $\mathcal{K}$, and ciphertext space $\mathcal{C}$. 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 will prove the result for the most basic case in Section 3.1. In Section 3.2 we describe the construction of *efficient* extractors. Then we consider computational extensions in Section 3.3. Finally, in Section 3.4 we consider further generalizations.

3.1 Encryption Implies Extraction

Assume $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ is $(\delta, \mathscr{S})$-secure. 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

$$\mathscr{S}' = \{\mathsf{Enc}(k, U_{\mathcal{M}}) \mid k \in \mathcal{K}\} \tag{3.1}$$

Lemma 1 *If $\mathscr{S}'$ is (ℓ, ε)-extractable and $\mathcal{E}$ is $(\delta, \mathscr{S})$-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 $\mathscr{S}$:*

$$\mathsf{Ext}(k) = \mathsf{Ext}'(\mathsf{Enc}(k, 1)) \tag{3.2}$$

Proof: Take any distribution $K \in \mathscr{S}$. Then Also, let Ext' be the assumed $(\mathscr{S}', \varepsilon)$-extractor. Thus, $\mathsf{SD}\left(\mathsf{Ext}'(\mathsf{Enc}(k, U_{\mathcal{M}})), U_{\mathcal{R}}\right) \leq \varepsilon$ for all $k \in \mathcal{K}$.

$$\begin{aligned}
&\mathsf{SD}\left(\mathsf{Ext}(K), U_{\mathcal{R}}\right) \\
&= \mathsf{SD}\left(\mathsf{Ext}'(\mathsf{Enc}(K, 1)), U_{\mathcal{R}}\right) && (3.3)\\
&\leq \mathsf{SD}\left(\mathsf{Ext}'(\mathsf{Enc}(K, 1)), \mathsf{Ext}'(\mathsf{Enc}(K, U_{\mathcal{M}}))\right) + \mathsf{SD}\left(\mathsf{Ext}'(\mathsf{Enc}(K, U_{\mathcal{M}}), U_{\mathcal{R}})\right) && (3.4)\\
&\leq \mathsf{SD}\left(\mathsf{Enc}(K, 1), \mathsf{Enc}(K, U_{\mathcal{M}})\right) + \mathsf{SD}\left((K, \mathsf{Ext}'(\mathsf{Enc}(K, U_{\mathcal{M}}))), (K, U_{\mathcal{R}})\right) && (3.5)\\
&\leq \delta + \mathsf{SD}\left((K, \mathsf{Ext}'(\mathsf{Enc}(K, U_{\mathcal{M}}))), (K, U_{\mathcal{R}})\right) && (3.6)\\
&= \delta + \mathbb{E}_{k \in K}\, \mathsf{SD}\left(\mathsf{Ext}'(\mathsf{Enc}(k, U_{\mathcal{M}})), U_{\mathcal{R}}\right) && (3.7)\\
&\leq \delta + \varepsilon && (3.8)
\end{aligned}$$

Equation (3.3) is a consequence of the definition of Ext in Equation (3.2). Equation (3.4) follows from the triangle inequality. Equation (3.5) follows from two applications of Equation (2.3) on statistical distance, with $f(x) = \mathsf{Ext}'(x)$ in the first, and $f(k, x) = x$ in the second. Equation (3.6) follows from rewriting

the security of Enc. Equation (3.7) is obtained by rewriting the joint distributions as an expected value. Equation (3.8) follows from the fact that Ext' is an (ℓ, ε)-extractor for $\mathscr{S}'$. We note that in Equation (3.5), we are giving the key to free to the attacker while using a random message. Even though we give the attacker the key, with a random message, the encryption remains secure. $\diamondsuit$

The point of this reduction (which is the only place in our argument using the $(\delta, \mathscr{S})$-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.[1] Thus, each D_k is a uniform distribution on some B-element subset of the ciphertext space $\mathcal{C}$, and in particular, $H_\infty(D_k) \geq b$. 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 2^n distributions over some space $\mathcal{C}$, where $b > \log n + 2\log\left(\frac{1}{\varepsilon}\right)$, and for all k, $H_\infty(D_k) \geq b$. Then $\mathscr{S}'$ is $\left(b - 2\log\left(\frac{1}{\varepsilon}\right), \varepsilon\right)$-extractable.*

The first assertion of Theorem 1(a) follows immediately by combining Lemma 1 and Lemma 2, whose proof we defer to Appendix A.1. In the following subsections we describe extensions to efficient extraction, computational security, and other "binding" primitives.

[1] This is the only place where we use the existence of the decryption algorithm. This is why our result will later extend in Section 3.4 to any sufficiently "binding" primitive.

3.2 Efficient Encryption Implies Efficient Extraction

Using Lemma 1 (and, in particular, Equation (3.2)), 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 min-entropy b 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 [51], which was later explored in more detail by [16].

The idea is to sample the function f (which will define Ext') at random from any family $\mathcal{F}_\alpha$ of *α-wise independent functions* from $\mathcal{C}$ to $\mathcal{R}$. Recall, such families have the property that for any distinct $c_1 \dots c_\alpha \in \mathcal{C}$, the values $f(c_1) \dots f(c_\alpha)$ are random and independent from each other, if f is chosen at random from $\mathcal{F}_\alpha$. Also, one can construct α-wise independent function families where each f can be evaluated in time polynomial in α 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 α is polynomial in n, every member $f \in \mathcal{F}_\alpha$ will be efficiently computable. As was shown by [51, 16], setting $\alpha = O(n)$ is already enough. The following Lemma (essentially from [16]) is proven for self-containment and because we use a slightly different parameter setting.

Lemma 3 ([16]) *Choose ℓ such that $\ell \leq b - \log n - 2\log\left(\frac{1}{\varepsilon}\right) - 2$. Let f be 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 source $\mathscr{S}' = \{D_k \mid k \in \mathcal{K}\}$ of cardinality 2^n satisfying*

$H_\infty(\mathscr{S}') \geq b$, *it follows that* $\Pr_f[\ f$ *is not an* $(\mathscr{S}', \varepsilon)$*-extractor* $] < 2^{-n}$.

The above lemma, which is proven in Appendix A.2, 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 [51] for the efficiently samplable source $\mathscr{S}'$. Assuming some strong complexity assumptions (see [51]), this would give us an explicit way to deterministically extract $\Omega(b)$ bits, provided $s < (1+\gamma)b$ for a small enough constant γ.

3.3 Computational Security

We will now extend our results to the computational setting. For this, we will need the following natural generalization of Definitions 1-4 taking into account the efficiency of corresponding attackers. We will now define computational distance, generalizing Equation (2.2) as follows.

DEFINITION 5 We define the *computational distance* $\mathsf{CD}_t(X_1, X_2)$ between two random variables X_1, X_2 to be the maximum

$$\max_D \left(\Pr[D(X_1) = 1] - \Pr[D(X_2) = 1]\right) \tag{3.9}$$

over all distinguishers D which are (possibly probabilistic) [2] Turing Machines running in time at most t. Hence, if $\mathsf{CD}_t(X_1, X_2) \leq \varepsilon$, then that no (possibly probabilistic) distinguisher D running in time t can tell apart a sample from X_1 from a sample from X_2 with an advantage greater than ε. $\diamondsuit$

We will use the following well-known fact.

Fact 2 *If f is a deterministic function computable in time t_2, then for all X_1, X_2,*

$$\mathsf{CD}_{t_1}(f(X_1), f(X_2)) \leq \mathsf{CD}_{t_1+t_2}(X_1, X_2). \tag{3.10}$$

DEFINITION 6 A random variable R over $\mathcal{R}$ is (t, ε)-*pseudorandom* if $\mathsf{CD}_t(R, U_{\mathcal{R}}) \leq \varepsilon$. Given a source $\mathscr{S}$ over some set $\mathcal{K}$, a function $\mathsf{Ext} : \mathcal{K} \to \mathcal{R}$ is an $(t, \varepsilon, \mathscr{S})$-*computational extractor* if for all $K \in \mathscr{S}$, $\mathsf{Ext}(K)$ is (t, ε)-pseudorandom:

$$\mathsf{CD}_t(\mathsf{Ext}(K), U_{\mathcal{R}}) \leq \varepsilon \tag{3.11}$$

If such Ext exists for $\mathscr{S}$, we say that $\mathscr{S}$ is (t, ℓ, ε)-*computationally extractable.* $\diamondsuit$

DEFINITION 7 An algorithm $\mathsf{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ is $(t, \delta, \mathscr{S})$-*computationally hiding* if for all messages $m_1, m_2 \in \mathcal{M}$ and all distributions $K \in \mathscr{S}$ we have

$$\mathsf{CD}_t(\mathsf{Enc}(K, m_1), \mathsf{Enc}(K, m_2)) \leq \delta \tag{3.12}$$

$\diamondsuit$

DEFINITION 8 An *encryption scheme* $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ over message space $\mathcal{M}$, key space $\mathcal{K}$ and ciphertext space $\mathcal{C}$ is $(t, \delta, \mathscr{S})$-*computationally secure* if Enc is $(t, \delta, \mathscr{S})$-

[2] We allow the adversary to have access to true randomness, in order to achieve the most general result.

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

We note that when $t = \infty$, statistical distance is equivalent to computational distance. In particular, definitions 6,7,8 become generalizations of earlier definitions 2,3,4. Thus we can state Lemma 1 and Lemma 2 in terms of $(\infty, \delta, \mathscr{S})$-security and (∞, ε)-pseudorandomness. With this in mind, it should be no surprise that we can obtain the following computational extension of Lemma 1.

Lemma 4 *Assume $\mathcal{E}$ is $(t_1, \delta, \mathscr{S})$-computationally secure and Ext' is a (t_2, ℓ, ε)-computational extractor for $\mathscr{S}' = \{\mathsf{Enc}(k, 1)\}_{k \in K}$. Then $\mathsf{Ext} = \mathsf{Ext}'(\mathsf{Enc}(k, 1))$ is a $(t_3, \ell, \varepsilon + \delta)$-computational extractor for $\mathscr{S}$, where $t_3 = \min(t_1 - t_{\mathsf{Samp}}, t_2 - t_K)$, t_{Samp} is the running time of Ext', and t_K is the time required to sample a key from $K \in \mathscr{S}$.*

Proof: Take any distribution $K \in \mathscr{S}$. Then

$$
\begin{aligned}
&\mathsf{CD}_{t_3}(\mathsf{Ext}(K),\ U_{\mathcal{R}}) \\
&= \mathsf{CD}_{t_3}(\mathsf{Ext}'(\mathsf{Enc}(K,1)),\ U_{\mathcal{R}}) && (3.13)\\
&\le \mathsf{CD}_{t_3}(\mathsf{Ext}'(\mathsf{Enc}(K,1)),\ \mathsf{Ext}'(\mathsf{Enc}(K,U_{\mathcal{M}}))) + \mathsf{CD}_{t_3}(\mathsf{Ext}'(\mathsf{Enc}(K,U_{\mathcal{M}}),\ U_{\mathcal{R}})) && (3.14)\\
&\le \mathsf{CD}_{t_1}(\mathsf{Enc}(K,1),\ \mathsf{Enc}(K,U_{\mathcal{M}})) + \mathsf{CD}_{t_2}((K,\mathsf{Ext}'(\mathsf{Enc}(K,U_{\mathcal{M}}))),\ (K,U_{\mathcal{R}})) && (3.15)\\
&\le \delta + \mathsf{CD}_{t_2}((K,\mathsf{Ext}'(\mathsf{Enc}(K,U_{\mathcal{M}}))),\ (K,U_{\mathcal{R}})) && (3.16)\\
&= \delta + \mathbb{E}_{k\in K}\,\mathsf{CD}_{t_2}(\mathsf{Ext}'(\mathsf{Enc}(k,U_{\mathcal{M}})),\ U_{\mathcal{R}}) && (3.17)\\
&\le \delta + \varepsilon && (3.18)
\end{aligned}
$$

The proof is virtually identical to the proof of Lemma 1. Equation (3.14) follows from the triangle inequality. Equation (3.15) follows from two applications of Equation (3.10) on computational distance, with $f(x) = \mathsf{Ext}'(x)$ in the first, and $f(k,x) = x$ in the second. Equation (3.16) is a consequence of the $(\delta, \mathscr{S})$-security of Enc. Equation (3.17) follows from rewriting the joint distributions as an expected value. Equation (3.18) follows from the fact that Ext' is an (ℓ,ε)-extractor for $\mathscr{S}'$. $\diamondsuit$

We now combine Lemma 4 and Lemma 3 to derive a computational version of Theorem 1(a). Let Ext' be an (ℓ,ε)-extractor (equivalently, an $(\infty,\ell,\varepsilon)$-extractor) for $\mathscr{S}' = \{\mathsf{Enc}(k,1)\}_{k\in K}$. By Lemma 3, there exists such an Ext' from the family of polynomials of degree $2n$ over the ciphertext space, which is a family of $2n$-wise independent functions computable in time $t_{\mathsf{Samp}} = O(ns\log s)$ using fast multiplication. Applying Lemma 4 with $t_2 = \infty$, $t_{\mathsf{Samp}} = O(ns\log s)$, we obtain

Corollary 5 *$\forall \varepsilon > 0$, if $\mathscr{S}$ is (t, b, δ)-computationally encryptable, and $b > \log n + 2\log\left(\frac{1}{\varepsilon}\right)$, then $\mathscr{S}$ is $(t - O(ns\log s), b - \log n - 2\log\left(\frac{1}{\varepsilon}\right) - 2, \varepsilon + \delta)$-computationally extractable for some extractor* Ext. *Further, if the encryption scheme is* efficient *(i.e., runs in time $t_{\mathsf{Enc}} = \mathsf{poly}(n)$), then there exists an efficient extractor* Ext *which runs in time $t_{\mathsf{Enc}} + O(ns\log s)$. Thus, even computationally secure encryption of $b > \log n$ bits implies efficient extraction of almost $b - \log n$ pseudorandom bits.*

A consequence of Corollary 5 is that any source which is not efficiently extractable cannot be efficiently encryptable. Of particular interest is a family of efficiently samplable sources considered by Trevisan and Vadhan [50]. For some polynomial $t(n)$, let $\mathscr{S}_{t(n)}$ be the source of all n-bit distributions with min-entropy at least $n-1$ which are samplable in time $t(n)$. Trevisan and Vadhan [51] showed that there exists a constant $c > 0$ such that any $(1, 1/5)$-extractor for $\mathscr{S}_{t(n)}$ cannot be computable in time less than $c \cdot t(n)$. Combining this fact with Corollary 5, we obtain the following corollary.

Corollary 6 *Any $(t, \log n + 8, 1/5)$-encryption scheme* Enc *on $\mathscr{S}_{t(n)}$ with s-bit ciphertexts must require time $t_{\mathsf{Enc}} \geq c \cdot t(n) - O(ns\log s)$ to compute. In particular, either $s = \Omega\left(\frac{t(n)/n}{\log(t(n)/n)}\right)$ or $t_{\mathsf{Enc}} = \Omega(t(n))$.*

This rules out the possibility of a generic construction of efficient encryption for non-extractable sources.

Proof: Suppose Enc on $\mathscr{S}_{t(n)}$ is computable in time t_{Enc}. It follows from Corollary 5 that there exists an $(b - \log n - 2\log\left(\frac{1}{\varepsilon}\right) - 2, \varepsilon)$-extractor computable in time $t_{\mathsf{Enc}} + t_{\mathsf{Samp}}$, where t_{Samp} is the amount of time required to sample an α-wise independent function f from $\mathcal{C}$ to $\mathcal{R}$. Setting parameters $\varepsilon = 1/5$ and $b = \log n + 8 > \log n + 2\log\left(\frac{1}{\varepsilon}\right) + 2 + 1$, it follows that there exists a $(t, 1, 1/5)$-computational extractor

computable in time $t_{\mathsf{Enc}} + t_{\mathsf{Samp}}$. But any 1-bit pseudorandom bit is also a random bit, so any $(t, 1, 1/5)$-computational extractor is also a $(1, 1/5)$-extractor. Trevisan and Vadhan (see Prop. A.4) showed that no such extractor is computable in time $t(n)$. Recall that $t_{\mathsf{Samp}} = O(ns \log s)$. Therefore, $t_{\mathsf{Enc}} + t_{\mathsf{Samp}} = t_{\mathsf{Enc}} + O(ns \log s) \geq c \cdot t(n)$.

Suppose t_{Enc} is not $\Omega(t(n))$, and assume that $t(n) > e \cdot n$. Then it follows that $ns \log s = \Omega(c \cdot t(n)) = \Omega(t(n))$, which can be written as $s \log s = \Omega(t(n)/n)$. Taking logarithms, it follows that $\log(s + \log s) = \log s + \log\log s = \Omega(\log(t(n)/n))$. Since the function $f(x) = \frac{x}{\log x}$ is monotone increasing for $x > e$, it follows that $\frac{s \log s}{\log s + \log\log s} = \Omega\left(\frac{t(n)/n}{\log(t(n)/n)}\right)$. But $\frac{s \log s}{\log s + \log\log s} = \frac{s}{1 + \frac{\log\log s}{\log s}} = O(s)$. Thus $s = \Omega\left(\frac{t(n)/n}{\log(t(n)/n)}\right)$, as required. ♢

The same result can be derived in the nonuniform model by substituting Proposition A.3 of [51] in place of Proposition A.4.

3.4 Extension to Decryption Error γ and Binding Commitments

In essence, we only used the fact that for all keys k, $\mathsf{Enc}(k, U_{\mathcal{M}})$ has min-entropy. Other than that, we did not need to use the existence of the decryption algorithm at all. Here we describe a general relaxation of the definition of encryption, which we shall apply to allow imperfect decryption with error γ and commitment schemes.

DEFINITION 9 We say that an algorithm Enc is $(t, \tau, \delta, b, \mathscr{S})$-good if Enc is $(t, \delta, \mathscr{S})$-

computationally hiding and there exists a source $\mathscr{S}'' = \{X_k\}_{k \in \mathcal{K}}$ such that

$$\forall k \in \mathcal{K} \qquad H_\infty(X_k) = b \tag{3.19}$$

$$\forall K \in \mathscr{S} \qquad \mathbb{E}_{k \in K} \mathsf{SD}\left(X_k, \mathsf{Enc}(k, U_{\mathcal{M}})\right) \leq \tau, \tag{3.20}$$

where the expected value is taken over keys k sampled from distribution K. $\diamondsuit$

Note that our auxiliary source $\mathscr{S}'$ no longer needs to have min-entropy b. Instead, we only require $\mathscr{S}'$ to be close to a source $\mathscr{S}''$ with min-entropy b. Furthermore, distance is measured with respect to $\mathscr{S}$ rather than with respect to $\mathcal{K}$: we only require $\mathsf{Enc}(k, U_{\mathcal{M}})$ to be close to X_k on average (over any $K \in \mathscr{S}$), which is less strict than requiring every distribution $\mathsf{Enc}(k, U_{\mathcal{M}}) \in \mathscr{S}'$ to be close to the corresponding distribution $X_k \in \mathscr{S}''$.

For the preceding results, it was sufficient to let $X_k = \mathsf{Enc}(k, U_{\mathcal{M}})$ and $\tau = 0$. The next lemma shows that sources which permit "good" encryption are also extractable. As a result, we can now consider X_k to be any distribution whose expected statistical distance to $\mathsf{Enc}(k, U_{\mathcal{M}})$ is bounded for all $K \in \mathscr{S}$. With the new definition, we can build extractors for more general sources. Using Definition 9, we get the following generalization of Corollary 5, which combines a generalization of Lemma 3 and Lemma 4.

Lemma 7 *If* Enc *is* $(t, \tau, \delta, b, \mathscr{S})$*-good and* $b > \log n + 2 \log \frac{1}{\varepsilon}$*, then* $\mathscr{S}$ *is* $(t - O(ns \log s), b - \log n - 2 \log\left(\frac{1}{\varepsilon}\right) - 2, \tau + \varepsilon + \delta)$*-computationally extractable.*

Lemma 7 accommodates the broader definition of encryption while increasing the distance to uniform by τ. Note that the number of bits extracted may remain the same or decrease, since b is now a parameter separate from the number of bits in the message space of Enc. As in Corollary 5, instead of using a generic extractor,

we will use a particular extractor which runs in time $O(ns \log s)$. The proof is a variant of the proof of Lemma 4. The argument is essentially identical except for the addition of another triangle inequality.

Proof: As before, define $\mathsf{Ext}(K) = \mathsf{Ext}'(\mathsf{Enc}(K, 1))$. Let $t = t_3 + O(ns \log s)$ and let t_K be the time required to sample a key from $K \in \mathscr{S}$.

$$\mathsf{CD}_{t_3}(\mathsf{Ext}(K),\ U_{\mathcal{R}})$$

$$= \mathsf{CD}_{t_3}(\mathsf{Ext}'(\mathsf{Enc}(K, 1)),\ U_{\mathcal{R}}) \tag{3.21}$$

$$\leq \mathsf{CD}_{t_3}(\mathsf{Ext}'(\mathsf{Enc}(K, 1)),\ \mathsf{Ext}'(\mathsf{Enc}(K, U_{\mathcal{M}}))) + \mathsf{CD}_{t_3}(\mathsf{Ext}'(\mathsf{Enc}(K, U_{\mathcal{M}}),\ U_{\mathcal{R}})) \tag{3.22}$$

$$\leq \mathsf{CD}_{t}(\mathsf{Enc}(K, 1),\ \mathsf{Enc}(K, U_{\mathcal{M}})) + \mathsf{CD}_{t_3+t_K}((K, \mathsf{Ext}'(\mathsf{Enc}(K, U_{\mathcal{M}}))),\ (K, U_{\mathcal{R}})) \tag{3.23}$$

$$\leq \delta + \mathsf{CD}_{t_3+t_K}((K, \mathsf{Ext}'(\mathsf{Enc}(K, U_{\mathcal{M}}))),\ (K, U_{\mathcal{R}})) \tag{3.24}$$

$$= \delta + \mathbb{E}_{k \in K}\, \mathsf{CD}_{t_3+t_K}(\mathsf{Ext}'(\mathsf{Enc}(k, U_{\mathcal{M}})),\ U_{\mathcal{R}}) \tag{3.25}$$

$$\leq \delta + \mathbb{E}_{k \in K}\ (\mathsf{CD}_{t_3+n}(\mathsf{Ext}'(\mathsf{Enc}(k, U_{\mathcal{M}})), \mathsf{Ext}'(X_k)) + \mathsf{CD}_{t_3+n}(\mathsf{Ext}'(X_k), U_{\mathcal{R}})) \tag{3.26}$$

$$\leq \delta + \mathbb{E}_{k \in K}\, \mathsf{CD}_{t_3+n}(\mathsf{Ext}'(\mathsf{Enc}(k, U_{\mathcal{M}})), \mathsf{Ext}'(X_k)) + \varepsilon \tag{3.27}$$

$$\leq \delta + \mathbb{E}_{k \in K}\, \mathsf{CD}_{t_3+n}(\mathsf{Enc}(k, U_{\mathcal{M}}), X_k) + \varepsilon \tag{3.28}$$

$$\leq \delta + \mathbb{E}_{k \in K}\, \mathsf{CD}_{t}(\mathsf{Enc}(k, U_{\mathcal{M}}), X_k) + \varepsilon \tag{3.29}$$

$$\leq \delta + \tau + \varepsilon. \tag{3.30}$$

The proof begins identically to the proof of Lemma 4. Equation (3.22) follows from the triangle inequality. Equation (3.23) follows from two applications

of Equation (3.10) on statistical distance, with $f(x) = \mathsf{Ext}'(x)$ in the first, and $f(k, x) = x$ in the second. Equation (3.24) is a consequence of the $(\delta, \mathscr{S})$-security of Enc. Equation (3.25) follows from rewriting the joint distributions as an expected value. Equation (3.26) follows from the triangle inequality. Equation (3.27) follows from the fact that Ext' is an (ℓ, ε)-extractor for $\mathscr{S}'$ for $\ell = b - \log n - 2\log\left(\frac{1}{\varepsilon}\right)$. Equation (3.28) follows from Equation (3.10) with $f(x) = \mathsf{Ext}'(x)$. Equation (3.29) follows from $t = t_3 + O(ns \log s) > t_3 + n$. Equation (3.30) follows from Equation (3.20). ♢

3.4.1 Extension to Decryption Error γ

Next, we extend our results to allow errors in decryption. The difficulty in this case is that $H_\infty(\mathsf{Enc}(k, U_\mathcal{M})) = b$ no longer holds. We can correct for this by finding X_k close to $\mathsf{Enc}(k, U_\mathcal{M})$ with min-entropy b.

DEFINITION 10 We say that an encryption scheme $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ is is $(\gamma, \mathscr{S})$-*correct* for $\gamma < 1$ if for all $K \in \mathscr{S}$,

$$\Pr_{k \leftarrow K, m \leftarrow \mathcal{M}}[\mathsf{Dec}(k, \mathsf{Enc}(k, m)) \neq m] \leq \gamma. \tag{3.31}$$

The following lemma shows that if $\mathcal{E}$ is δ-computationally secure and $(\gamma, \mathscr{S})$-correct, we can construct an extractor in exchange for losing γ statistical distance to uniform.

Lemma 8 *Suppose $\mathcal{E}$ is $(t, \delta, \mathscr{S})$-computationally secure and $(\gamma, \mathscr{S})$-correct. If $b > \log n + 2\log\left(\frac{1}{\varepsilon}\right)$, then $\mathscr{S}$ is $(t - O(ns \log s), b - \log n - 2\log\left(\frac{1}{\varepsilon}\right) - 2, \delta + \gamma + \varepsilon)$-computationally extractable.*

Proof: By Lemma 7, it is sufficient to show that $\mathcal{E}$ is $(t, \gamma, \delta, b, \mathscr{S})$-good. For all $k \in K$, let $\varphi_k : \mathcal{M} \times \mathcal{C} \to \mathcal{C}$ be an arbitrary injective mapping from the set $\{m : \mathsf{Dec}(k, \mathsf{Enc}(k, m)) \neq m\}$ of incorrectly decrypted messages under key k, whose support consists solely of ciphertexts $s \in \mathcal{C}$ for which no message m encrypts to s. More formally, if $\varphi_k(m) = s$, then $\mathsf{Dec}(k, \mathsf{Enc}(k, m)) \neq m$, and furthermore there does not exist m_1 such that $\mathsf{Enc}(k, m_1) = s$. Since the ciphertext space is larger than the message space, φ_k must exist, although it may be inefficient. Intuitively, we use φ_k to change Enc so that it is potentially inefficient, but has no decryption error. Instead, we can think of the decryption error γ as being converted into an additional hiding error. We choose the distribution X_k, considered as a function of m (with ciphertexts in $\mathcal{C}$ disjoint from messages in $\mathcal{M}$) as follows:

$$X_k(m) = \begin{cases} \mathsf{Enc}(k, m), & \mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m, \\ \varphi_k(m) & \text{otherwise.} \end{cases}$$

First we examine the min-entropy of X_k. Suppose $X_k(m_1) = X_k(m_2)$. This cannot happen if $X_k(m_1) = m_1$ or $X_k(m_2) = m_2$. Therefore $\mathsf{Enc}(k, m_1) = \mathsf{Enc}(k, m_2)$, which implies $\mathsf{Dec}(k, \mathsf{Enc}(k, m_1)) = \mathsf{Dec}(k, \mathsf{Enc}(k, m_2))$. It follows that one of m_1, m_2 is decrypted incorrectly, which contradicts our assumption. It follows that $H_\infty(X_k) = b$.

Next we consider Equation (3.20). $\mathbb{E}_{k \in K} \mathsf{SD}\left(X_k, \mathsf{Enc}(k, U_{\mathcal{M}})\right)$ can be rewritten as $\Pr_{k \leftarrow k, m \leftarrow \mathcal{M}}[\mathsf{Dec}(k, \mathsf{Enc}(k, m) \neq m)]$, which is $\leq \gamma$ by $(\gamma, \mathscr{S})$-correctness. Therefore Equation (3.20) is satisfied with parameter $\tau = \gamma$. It follows that Enc is $(t, \gamma, \delta, \ell, \mathscr{S})$-good. The claim now follows immediately from Lemma 7. $\diamondsuit$

Setting $t = \infty$ and using Lemma 2 in place of Lemma 3, we obtain the following information-theoretic corollary.

Corollary 9 *Suppose $\mathcal{E}$ is $(\delta, \mathscr{S})$-secure and $(\gamma, \mathscr{S})$-correct. If $b > \log n + 2\log\left(\frac{1}{\varepsilon}\right)$, then $\mathscr{S}$ is $\left(b - 2\log\left(\frac{1}{\varepsilon}\right), \delta + \gamma + \varepsilon\right)$-extractable.*

3.4.2 Commitments

We use $(t, \tau, \delta, \ell, \mathscr{S})$-goodness to extend our results above to handle privacy primitives which are sufficiently "*binding*," which includes commitments. This means that there exists an algorithm Enc, which takes input $m \in \mathcal{M}$ and "randomness" $k \in \mathcal{K}$, and outputs a 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 define binding and hiding as follows.

Definition 11 An algorithm Enc is $(t, \beta, \mathscr{S})$-*computationally binding* for $\beta \geq 0$ if for any adversary $\mathcal{A}$ running in time t, for any $K \in \mathscr{S}$,

$$\Pr_{k \leftarrow K}[\mathcal{A}(k) \to (m_0, m_1) : m_0 \neq m_1, \mathsf{Enc}(k, m_0) = \mathsf{Enc}(k, m_1)] \leq \beta$$

If $\beta = 0$, we call Enc *perfectly binding*. ♢

A note on perfect binding. Clearly, Definition 11 applies to the perfectly-binding encryption and commitment applications above, with $\beta = 0$. Our notion of perfect binding even includes some primitives which are traditionally *not* considered perfectly-binding. For example, Pedersen's commitment [39] 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.

We begin with an observation about Rényi entropy.

Lemma 10 *Assume* Enc *is* $(\delta,\mathscr{S})$*-secure and* $(t,\beta,\mathscr{S})$*-binding, for* $t > 2b$*. Then* $\forall K \in \mathscr{S}$, $\mathbb{E}_{k\leftarrow K} H_2(\mathsf{Enc}(k,U_{\mathcal{M}})) \geq \log\left(\frac{1}{\beta+2^{-b}}\right)$.

Proof: Recall that for any distribution X, the Rényi entropy of order 2 is defined as $H_2(X) = -\log\sum_{x\in X} p_x^2$, where p_x is the probability of sampling x from X. Let $X = \mathsf{Enc}(k,U_{\mathcal{M}})$. Consider the distinguisher which runs in time $t = 2b$, simply picking two random messages. This distinguisher finds a collision (possibly involving the same message) with probability $2^{-H_2(X)} = \sum_{x\in X} p_x^2$. Since the probability of drawing the same message twice is 2^{-b}, the distinguisher succeeds with probability $2^{-H_2(X)} - 2^{-b} \leq \beta$. It follows that $H_2(X) \geq \log\left(\frac{1}{\beta+2^{-b}}\right)$. ♢

Existence of Extractor. Our approach will be to consider the auxiliary source $\mathscr{S}' = \{\mathsf{Enc}(k,U_{\mathcal{M}})\}_{k\in K}$, as before. By Lemma 10, it follows that $\mathscr{S}'$ has high Rényi entropy of order 2. Next we use a well-known lemma which shows that a distribution having high Rényi entropy of order 2 is close to a distribution having high min-entropy. The lemma is closely related to a more general lemma of Renner

and Wolf, which appears as Lemma I.3 of [43] and Lemma 2 of [44]. For simplicity and self-containment, we state and prove our version of the lemma for $\alpha = 2$.

Lemma 11 *For all K, ε, there exists K' such that $H_\infty(K') \geq H_2(K) - \log \frac{1}{\varepsilon}$, and $\mathsf{SD}(K,\, K') \leq \varepsilon$.*

Proof: Let $p_k = \Pr[k = K]$ and $p'_k = \Pr[k = K']$ be the probabilities of obtaining key k from distributions K and K', respectively. Let $\alpha = e^{-H_2(K)} = \sum p_k^2$. For a parameter $0 \leq p \leq 1$, let $K_p = \{k : p_k > p\}$ be the set of all "heavy" elements $k \in K$ which occur with probability greater than p. It is easy to see that there exists a probability distribution K' such that $\max_{k \leftarrow K'} p'_k = p$ and $\mathsf{SD}(K,\, K') = \sum_{k \in K_p} (p_k - p)$. K' can be obtained from K by setting all probabilities larger than p to p. We can then compensate for the loss of probability mass by either increasing the value of the smallest probabilities or adding new elements, each of which occurs with probability at most p. It follows that $\alpha = \sum_{k \in K} p_k^2 \geq \sum_{k \in K_p} p_k^2 \geq p \sum_{k \in K_p} p_k \geq p \sum_{k \in K_p} (p_k - p) \geq p\mathsf{SD}(K,\, K')$. Setting $p = \frac{\alpha}{\varepsilon}$ gives $\varepsilon \geq \mathsf{SD}(K,\, K')$, and $H_\infty(K') = -\log \alpha + \log \varepsilon = H_2(K) - \log\left(\frac{1}{\varepsilon}\right)$. $\diamondsuit$

As a corollary, it follows that if Enc is binding and hiding, then it must be "good".

Corollary 12 *$\forall \varepsilon > 0$, if Enc is $(t, \beta, \mathscr{S})$-binding and $(t, \delta, \mathscr{S})$-hiding, and $b > \log n + 2 \log\left(\frac{1}{\varepsilon}\right)$, then Enc is $\left(t, \varepsilon, \delta, \log\left(\frac{1}{\beta + 2^{-b}}\right) - \log\left(\frac{1}{\varepsilon}\right), \mathscr{S}\right)$-good.*

Proof: Consider a $(t_1, \beta, \mathscr{S})$-binding and $(t_2, \delta, \mathscr{S})$-hiding Enc. By Lemma 10, we know that $H_2(\mathsf{Enc}(k, U_{\mathcal{M}})) \geq \log\left(\frac{1}{\beta + 2^{-b}}\right)$. It follows from Lemma 11 that that there exists X_k such that $\mathsf{SD}(X_k,\, \mathsf{Enc}(k, U_{\mathcal{M}})) \leq \varepsilon$ and $H_\infty(X_k) \geq \log\left(\frac{1}{\beta + 2^{-b}}\right) - \log\left(\frac{1}{\varepsilon}\right)$. Since Enc is also $(t_2, \delta, \mathscr{S})$-hiding, it follows that Enc is $\left(t, \varepsilon, \delta, \log\left(\frac{1}{\beta + 2^{-b}}\right) - \log\left(\frac{1}{\varepsilon}\right), \mathscr{S}\right)$-good. $\diamondsuit$

Combining Corollary 12 and Lemma 7, we immediately obtain a version of Theorem 1(a) for commitments.

Theorem 2 *$\forall \varepsilon > 0$, if* Enc *is $(t, \beta, \mathscr{S})$-binding and $(t, \delta, \mathscr{S})$-hiding, and $b > \log n + 2\log\left(\frac{1}{\varepsilon}\right)$, there exists an extractor which is $\left(t, \log\left(\frac{1}{\beta+2^{-b}}\right) - 3\log\left(\frac{1}{\varepsilon}\right), 2\varepsilon + \delta, \mathscr{S}\right)$-pseudorandom. Further, if the commitment scheme is* efficient *(i.e., polynomial in n), then there exists an* efficient *extractor which is $\left(t, \log\left(\frac{1}{\beta+2^{-b}}\right) - \log n - 3\log\left(\frac{1}{\varepsilon}\right) - 2, 2\varepsilon + \delta, \mathscr{S}\right)$-pseudorandom.*

As a result, even commitment of $b > \log n$ bits implies extraction of almost $\log\left(\frac{1}{\beta+2^{-b}}\right) - \log n$ nearly perfect bits.

Chapter 4

Encryption Does Not Require Extraction if $b < \log n - \operatorname{loglog} n$

In this section we prove the non-implication given in Theorem 1(b), which shows that even perfect encryption of up to $(\log n - \operatorname{loglog} n)$ 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

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 $\mathsf{Enc}(k, m) = c_m$. Technically, some B-tuples might repeat for several keys, but it is easy to see that such "repeated" keys will only complicate our job.[1] More interestingly, 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 $S > B$.[2] 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 $\mathcal{C}$:

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

[1] We omit the argument, since it is not very illuminating. Essentially, such keys force us to consider more extractors when arguing lack of extraction, without expanding the "geometry" of perfect key distributions.

[2] 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.

We then define $\mathsf{Enc}((c_1 \dots c_B), m) = c_m$, while $\mathsf{Dec}((c_1, \dots, c_B), c)$ is defined to be the (necessarily unique) m such that $c_m = c$, and arbitrarily if no such m exists.

4.2 Defining Bad Extraction

Let us now fix 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 will show that either Ext can be completely biased to output 0 on some distribution, or there must exist a distribution for which Ext is almost completely biased to output 1. More specifically, we will show that either there exists K such that $\Pr[\mathsf{Ext}(K) = 0] = 1$, implying $\mathsf{SD}(\mathsf{Ext}(K), U_1) = \frac{1}{2}$; or there exists K such that $\Pr[\mathsf{Ext}(K) = 0] \leq \frac{B^2}{S}$. Clearly, in the first case, $\mathsf{SD}(\mathsf{Ext}(K), U_1) = \frac{1}{2}$ (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 K such that $\Pr[\mathsf{Ext}(K) = 0] = 1$. The heart of the proof then will consist of designing a perfect encryption distribution K such that

$$\Pr[\mathsf{Ext}(K) = 0] \leq \frac{B^2}{S} \tag{4.1}$$

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

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

as claimed by Theorem 1(b). Thus, we concentrate on building a perfect distribution K satisfying Equation (4.1). 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

We say that a distribution K is 0-*monochromatic* (with respect to Ext) if $\Pr[\mathsf{Ext}(K) = 0] = 1$. As in the previous section, we can assume that the set of perfect distributions $\mathscr{S}(\mathcal{E})$ does not contain any 0-monochromatic 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

$$\Pr[\mathsf{Enc}(K, 1) = c] = \Pr[\mathsf{Enc}(K, m) = c] \tag{4.2}$$

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 (4.2) can be rewritten as

$$\sum_{\{(c_1 \dots c_B) : c_1 = c\}} p_{(c_1 \dots c_B)} - \sum_{\{(c_1 \dots c_B) : c_m = c\}} p_{(c_1 \dots c_B)} = 0 \tag{4.3}$$

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 (4.3), we first make our notation more suggestive. We index the N columns of V by tuples $(c_1, \ldots c_B)$, and the remaining $(B-1)S$ rows of V by tuples (m, c), where $m \in \{2, \ldots B\}$ and $c \in \{1 \ldots S\}$. Then, we define

$$v_{(m,c),(c_1,\ldots,c_B)} = \begin{cases} 1, & c = c_1, \\ -1, & c = c_m, \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

Now, Equation (4.3) 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 13 *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 13 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 [49]):

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 the $(1+(B-1)S) \times |Z|$-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 13, 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

$$y_1 + \sum_{m>1} \sum_{c} y_{(m,c)} \cdot v_{(m,c),(c_1,\ldots,c_B)} \geq 0 \tag{4.5}$$

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 (4.4), we conclude that for all $k = (c_1, \ldots, c_B)$, such that $\mathsf{Ext}(k) = 0$, we have

$$\sum_{m>1} \left(y_{(m,c_1)} - y_{(m,c_m)} \right) > 0 \tag{4.6}$$

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

Theorem 3 *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*

$$y_{(m,c_1)} - y_{(m,c_m)} \leq 0 \qquad \text{for all } m > 1, \tag{4.7}$$

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

Proof: Summing Equation (4.7) for all $m > 1$ we get a contradiction to Equation (4.6), 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 [22]. Theorem 3 tells us that if Ext cannot be fixed to 0, there exists real numbers $y_1 \ldots y_S$ 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 $y_1 \leq y_2 \leq \ldots \leq y_S$, we get that $\mathsf{Ext}((i, j)) = 1$ for any $i < j$. In particular, the uniform distribution on S keys $\{(1, 2), (2, 3), \ldots, (S - 1, S), (S, 1)\}$ 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 (4.1) and thus completing this special case.

Interestingly, Dodis and Spencer [22] 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 $c_1 \ldots c_S$ 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 3.

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 3) 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 (4.7), meaning that $\mathsf{Ext}(K)$ is very biased towards 1. We will construct such K having a very special form below.

DEFINITION 12 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 14 *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 $K_{(\pi_1,\ldots,\pi_B)}$ is perfect. $\diamondsuit$

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 $1 < m \leq d$:[3]

$$y_{(m,c)} - y_{(m,\pi_m(c))} \leq 0 \tag{4.8}$$

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 14 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 (4.7). Thus, by Theorem 3 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 (4.1) and completes our proof (modulo the inductive step).

[3]To get some intuition, we will see shortly that "good" ciphertexts c will lead to keys k_c satisfying Equation (4.7), so that $\mathsf{Ext}(k_c) = 1$ by Theorem 3.

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 15 *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. $\diamondsuit$

Corollary 16 *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 $|\mathcal{N}(A)| \geq \mathsf{deg}_G(c_{i_a}) \geq i_a \geq a = |A|$. $\diamondsuit$

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 $F_c = \{c, \pi_2(c), \ldots, \pi_d(c)\}$. 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 17 *Every vertex $v \in L \cup R$ has degree at least $S - d$: $\mathsf{deg}_G(v) \geq S - d$. In particular, by Corollary 15 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 $v, \pi_2^{-1}(v), \ldots, \pi_d^{-1}(v)$. $\diamondsuit$

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 (4.8) 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 (4.8) 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 $c \in T_{d+1}$

$$y_{(d+1,c)} - y_{(d+1,\pi_{d+1}(c))} \leq 0 \tag{4.9}$$

This constraint motivates us to define a subgraph G' of our constraint graph G as follows. An 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 (4.9) if we were to define $\pi_{d+1}(c) = c'$. The key property of G' turns out to be

Lemma 18 *G' contains a matching P' of size at least $S - d$.*

Proof: We will use Corollary 16. 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 16, $\{v_{S-d}, \dots, v_1\}$ 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 18. 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 17. 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 17, 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 (4.9), 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.

Chapter 5

Conclusions

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 [20]) or general multi-party computation? 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.

Appendix A

Proofs of Lemma 2 and Lemma 3

A.1 Proof of Lemma 2

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 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

$$\Delta_f(k, \mathcal{T}) \stackrel{\text{def}}{=} \Pr[f(D_k) \in \mathcal{T}] - \Pr[U_{\mathcal{R}} \in \mathcal{T}] \tag{A.1}$$

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 distribution of min-entropy $\geq b$ over some set $\{c_1, \ldots, c_\beta\}$ in $\mathcal{C}$ for some $\beta \geq B$, and let X_m denote an indicator random variable which is 1 if and only if $f(c_m) \in \mathcal{T}$. Let $p_m = \Pr_{Y \leftarrow D_k}(Y = c_m)$ denote the probability that c_m is drawn from D_k. Then $\sum_{m \subseteq \mathcal{C}} p_m = 1$. Let $Z_m = X_m \cdot p_m$. Clearly,

if f is random, we have $\Pr_f[Z_m = c_m] = p$. Also, letting $\hat{X} = \sum_m p_m \cdot X_m$ be the average of independent indicator variables X_m, for any fixed f we get $\Pr[f(D_k) \in \mathcal{T}] = \sum_m p_m \cdot X_m = \hat{X}$.

We will apply the standard additive Hoeffding bound, Theorem 2.6 of [28]:

$$\Pr(\hat{Z} - \mu \geq tn) \leq e^{-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}},$$

where $\hat{Z} = \sum Z_i$, $\mu = \mathbb{E}[\hat{Z}]$, and $a_m \leq Z_m \leq b_m$. Recalling the definition of $\Delta_f(k, \mathcal{T})$ from *Equation* (*A*.1), we have $\mu = \mathbb{E}[\hat{Z}] = p = \Pr[U_{\mathcal{R}} \in \mathcal{T}]p$. Setting $n = \beta \geq B$, $a_m = 0$, $b_m = p_m$, and $t = \varepsilon/n$, we find that

$$\begin{aligned}
\Pr_f[\ \Delta_f(k, \mathcal{T}) > \varepsilon\ ] &= \Pr_f[\ \hat{Z} - p > \varepsilon\ ] \\
&\leq e^{-2\varepsilon^2 / \sum_{m=1}^{\beta} p_m^2} \\
&= e^{-2\varepsilon^2 2^{H_2(D_k)}} \\
&\leq e^{-2B\varepsilon^2},
\end{aligned}$$

since the Rényi entropy $-\log \sum_{m=1}^{\beta} p_m^2 = H_2(D_k) \geq H_\infty(D_k) = b$. We now take a union bound over all $\mathcal{T} \subseteq \mathcal{R}$ and all $k \in \mathcal{K}$. Recalling the definition of $\Delta_f(k, \mathcal{T})$ (Equation (A.1)), using $b > \text{loglog}\, N + 2\log\left(\frac{1}{\varepsilon}\right)$ (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

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

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 (2.2)),

this means that $\mathsf{SD}(f(D_k), U_{\mathcal{R}}) \leq \varepsilon$ for all $k \in \mathcal{K}$, completing the proof. $\diamondsuit$

A.2 Proof of Lemma 3

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 distribution $D_k \in \mathscr{S}'$ with min-entropy $\geq b$, $\Pr_f[f(D_k) \in \mathcal{T}]$ is unlikely to be different from its expectation $\Pr[U_{\mathcal{R}} \in \mathcal{T}]$ by more than 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 (2.1) (rather than Equation (2.2)) 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 distribution over some set $\{c_1, \ldots, c_\beta\} \subseteq \mathcal{C}$, with $H_\infty(D_k) \geq b$, and let X_m denote an indicator random variable which is 1 if and only if $f(c_m) = r$. Let $p_m = \Pr_{Y \leftarrow D_k}(Y = c_m)$ denote the probability that c_m is drawn from D_k. Then $\sum_{m \subseteq \mathcal{C}} p_m = 1$. Let $Z_m = X_m \cdot p_m \cdot B \leq X_m \leq 1$.

Since f is $2n$-wise independent, so are the variables $\{Z_m\}$: any $2n$ of them are random and independent from each other. Let $Z = \sum_m Z_m$. Then $\Pr_f[X_m = 1] =$

$\Pr_f[f(c_m) = r)] = \frac{1}{L}$, and $\mathbb{E}[Z] = \sum_m \frac{B \cdot p_m}{L} = \frac{B}{L}$. Also,

$$\Delta_f(k, \{r\}) = \sum_m \Pr[f(c_m) = r] - \frac{1}{L} = \frac{1}{B}\left(Z - \mathbb{E}[Z]\right) \tag{A.2}$$

Next, we use the tail bound for the sum Z of t-wise independent random variables from [16] (Theorem 5, page 48), which is a special case of a more general bound from [7]. It says that if $t \geq 8$ is an even integer and $\varepsilon < \frac{1}{2}$, then $\Pr[|Z - \mathbb{E}[Z]| \geq 2\varepsilon \cdot \mathbb{E}[Z]] \leq \left(\frac{t}{4\varepsilon^2 \mathbb{E}[Z]}\right)^{t/2}$. In our case, $t = 2n$, $\mathbb{E}[Z] = \frac{B}{L}$, and we get by Equation (A.2)

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

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 (2.1), 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. $\diamondsuit$

Bibliography

[1] Ajtai, M., and Linial., N. The influence of large coalitions. *Combinatorica 13*, 2 (1993), 129–145.

[2] Akavia, A., Goldwasser, S., and Vaikuntanathan, V. Simultaneous hardcore bits and cryptography against memory attacks. In *Proc. 6th Theory of Cryptography Conference (TCC)* (Jan 2009).

[3] Andreev, A., Clementi, A., Rolim, J., and Trevisan., L. Dispersers, deterministic amplification and weak random sources. *SIAM J. on Computing 28*, 6 (1999), 2103–2116.

[4] Barak, B., Impagliazzo, R., and Wigderson, A. Extracting randomness using few independent sources. *SIAM J. Comput. 36*, 4 (2006), 1095–1118. Preliminary version appears in ACM FOCS 2004.

[5] Barak, B., Kindler, G., Shaltiel, R., Sudakov, B., and Wigderson, A. Simulating independence: New constructions of condensers, Ramsey graphs, dispersers, and extractors. In *Proc. 37th ACM STOC* (2005), ACM, p. 10.

[6] Barak, B., Rao, A., Shaltiel, R., and Wigderson, A. 2-source dispersers for sub-polynomial entropy and Ramsey graphs beating the Frankl-Wilson construction. In *Proc. 38th ACM STOC* (Jan 2006).

[7] Bellare, M., and Rompel, J. Randomness-efficient oblivious sampling. In *Proc. 35th IEEE FOCS* (1994), pp. 276–287.

[8] Bennett, C. H., Brassard, G., and Robert., J.-M. Privacy amplification by public discussion. *SIAM J. on Computing 17*, 2 (1988), 210–229.

[9] Blum, M. Independent unbiased coin flips from a correlated biased source — a finite state Markov chain. *Combinatorica 6*, 2 (1986), 97–108.

[10] Bourgain, J. More on the sum-product phenomenon in prime fields and its applications. *International Journal of Number Theory* (Jan 2005).

[11] Boyd, S., and Vandenberghe, L. Convex optimization. *books.google.com* (Jan 2004).

[12] Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E., and Sahai, A. Exposure-resilient functions and all-or-nothing transforms. In *Proc. EUROCRYPT* (2000), pp. 453–469.

[13] Chandran, N., Kanukurthi, B., Ostrovsky, R., and Reyzin, L. Privacy amplification with asymptotically optimal entropy loss. In *Proc. 42nd ACM STOC* (2010).

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

[15] Chor, B., Goldreich, O., Håstad, J., Friedman, J., Rudich, S., and Smolensky, R. The bit extraction problem of t-resilient functions. In *Proc. 26th IEEE FOCS* (1985), pp. 396–407.

[16] Dodis, Y. *Exposure-Resilient Cryptography.* PhD thesis, MIT, 2000.

[17] Dodis, Y., Elbaz, A., Oliveira, R., and Raz, R. Improved randomness extraction from two independent sources. In *Proc. RANDOM* (Jan 2004).

[18] Dodis, Y., and Oliveira, R. On extracting private randomness over a public channel. In *Proc. RANDOM* (Jan 2003).

[19] Dodis, Y., Ong, S. J., Prabhakaran, M., and Sahai, A. On the (im)possibility of cryptography with imperfect randomness. In *Proc. 45th IEEE FOCS* (2004), pp. 196–205.

[20] Dodis, Y., Pietrzak, K., and Przydatek, B. Separating sources for encryption and secret-sharing. In *Proc. 3rd Theory of Cryptography Conference (TCC)* (2006), pp. 601–616.

[21] Dodis, Y., Sahai, A., and Smith, A. On perfect and adaptive security in exposure-resilient cryptography. In *Proc. EUROCRYPT* (2001), pp. 301–324.

[22] Dodis, Y., and Spencer, J. On the (non-)universality of the one-time pad. In *Proc. 43rd IEEE FOCS* (2002), pp. 376–388.

[23] Dodis, Y., and Wichs, D. Non-malleable extractors and symmetric key cryptography from weak secrets. In *Proc. 41st ACM STOC* (2009), pp. 601–610.

[24] Dziembowski, S., and Pietrzak, K. Leakage-resilient cryptography. In *Proc. 49th IEEE FOCS* (2008).

[25] Elias, P. The efficient construction of an unbiased random sequence. *Ann. Math. Stat. 43*, 2 (1972), 865–870.

[26] Goldreich, O., and Levin, L. A hard-core predicate for all one-way functions. In *Proc. 21st ACM STOC* (1989), pp. 25–32.

[27] Goldwasser, S., Sudan, M., and Vaikuntanathan, V. Distributed computing with imperfect randomness. *Distributed Computing* (2005).

[28] Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* (Jan 1963).

[29] Kalai, Y. T., Li, X., and Rao, A. 2-source extractors under computational assumptions and cryptography with defective randomness. In *Proc. 50th IEEE FOCS* (2009).

[30] Kalai, Y. T., Li, X., Rao, A., and Zuckerman, D. Network extractor protocols. In *Proc. 49th IEEE FOCS* (2008).

[31] Kamp, J., Rao, A., Vadhan, S., and Zuckerman, D. Deterministic extractors for small-space sources. In *Proc. 38th ACM STOC* (2006), pp. 691–700.

[32] Kamp, J., and Zuckerman, D. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. In *Proc. 44th IEEE FOCS* (2003), pp. 92–101.

[33] Kanukurthi, B., and Reyzin, L. Key agreement from close secrets over unsecured channels. In *Proc. EUROCRYPT* (2009), pp. 206–223.

[34] Lichtenstein, D., Linial, N., and Saks, M. Some extremal problems arising from discrete control processes. *Combinatorica 9*, 3 (1989), 269–287.

[35] Maurer, U., and Wolf, S. Privacy amplification secure against active adversaries. In *Proc. CRYPTO* (1997), pp. 307–321.

[36] McInnes, J. L., and Pinkas, B. On the impossibility of private key cryptography with weakly random keys. In *Proc. CRYPTO* (1990), pp. 421–436.

[37] Nisan, N., and Zuckerman, D. More deterministic simulation in logspace. In *Proc. 25th ACM STOC* (Jan 1993).

[38] Nisan, N., and Zuckerman, D. Randomness is linear in space. *Journal of Computer and System Sciences* (Jan 1996).

[39] Pedersen, T. P. Non-interactive and information-theoretic secure verifiable secret sharing. In *Proc. CRYPTO* (1991), pp. 129–140.

[40] Rao, A. Extractors for a constant number of polynomially small min-entropy independent sources. In *Proc. 38th ACM STOC* (Jan 2006).

[41] Rao, A. An exposition of Bourgain's 2-source extractor, Jan 2007.

[42] Renner, R., and Wolf, S. Unconditional authenticity and privacy from an arbitrarily weak secret. In *Proc. CRYPTO* (Jan 2003).

[43] Renner, R., and Wolf, S. Smooth Rényi entropy and applications. In *Proc. ISIT* (2004).

[44] Renner, R., and Wolf, S. Simple and tight bounds for information reconciliation and privacy amplification. In *Proc. ASIACRYPT* (2005), pp. 199–216.

[45] Rényi, A. On measures of information and entropy. In *Proc. 4th Berkeley Symposium on Mathematics, Statistics and Probability* (1961), vol. 1, pp. 547–561.

[46] Rivest, R. All-or-nothing encryption and the package transform. In *Proc. Fast Software Encryption* (Jan 1997), pp. 210–218.

[47] Santha, M., and Vazirani, U. V. Generating quasi-random sequences from semi-random sources. *Journal of Computer and System Sciences 33*, 1 (1986), 75–87.

[48] Shannon, C. Communication theory of secrecy systems. *Bell Systems Technical J. 28* (1949), 656–715.

[49] Strang, G. *Linear Algebra and Its Applications*. Academic Press, London, 1980.

[50] Trevisan, L., and Vadhan, S. Extracting randomness from samplable distributions. In *Proc. 41st IEEE FOCS* (November 2000), pp. 32–42.

[51] Trevisan, L., and Vadhan, S. Extracting randomness from samplable distributions, April 2000. Full version of [50].

[52] Vazirani, U. V., and Vazirani, V. V. Random polynomial time is equal to slightly-random polynomial time. In *Proc. 26th IEEE FOCS* (1985), pp. 417–428.

[53] VON NEUMANN, J. Various techniques used in connection with random digits. *National Bureau of Standards, Applied Mathematics Series 12* (1951), 36–38.

[54] ZUCKERMAN, D. Simulating BPP using a general weak random source. *Algorithmica 16*, 4/5 (1996), 367–391.