BOSTON UNIVERSITY

GRADUATE SCHOOL OF ARTS AND SCIENCES

Dissertation

STRONG KEY DERIVATION FROM NOISY SOURCES

by

BENJAMIN WOODBURY FULLER

B.S., Rennselaer Polytechnic Institute, 2006
M.A., Boston University, 2011

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2015

Approved by

First Reader

Leonid Reyzin
Associate Professor of Computer Science

Second Reader

Ran Canetti
Professor of Computer Science

Third Reader

Daniel Wichs
Assistant Professor of Computer Science
Northeastern University

Acknowledgments

Looking at my time in graduate school, any success I have enjoyed is due to the people around me. First, I can not imagine having an advisor other than Leo Reyzin. Ignoring his technical abilities (though the more I learn, the more I appreciate Leo's technical skills), Leo is the greatest reason I am completing my graduate work. Through the last six years, Leo put my needs and goals foremost. I was always encouraged to work on problems that interested me. I am most thankful for the honesty of our conversations. If I did something wrong, Leo would tell me and explain how to change my behavior. I knew where I stood. Advice was always honest and considered.

The first couple of years were the hardest for me. It would have been easy to give up on me. Leo was critical in getting through this period. We had nuanced and tough conversations about my abilities, my probability of success, and how best to achieve my goals.

In addition to Leo, I am lucky to work with coauthors that constantly surprise and impress me with their creativity. John Bannick, Ran Canetti, Joe Cooley, Rob Cunningham, Ariel Hamlin, Kyle Ingols, Roger Khazan, Adam O'Neill, Xianrui Meng, Omer Paneth, Galen Pickard, Lee Rossey, Adam Smith, Merrielle Spain, and Tamara Yu challenge my thinking and allow me to see problems from different perspectives.

Ran Canetti, Sharon Goldberg, Steve Homer, Leonid Reyzin, and Daniel Wichs have helped refine and produce much of the research contained in this dissertation. My committee is full of talented researchers who are caring, consideration, and inspire the expansion of knowledge.

I have engaged in important technical discussions with many others including (I am sure this list is woefully incomplete): Jacob Alperin-Sheriff, Alexandra Berkoff, Nir Bitansky, Venkat Chandar, Nishanth Chandran, Kai-Min Chung, Yevgeniy Dodis, Nico Döttling, Sebastian Faust, Péter Gács, Chun-Yuan Hsiao, Gene Itkis, Feng-

Hao Liu, Danielle Micciancio, Christopher Peikert, Krzysztof Pietrzak, Oded Regev, Emily Shen, Salil Vadhan, Mayank Varia, Daniel Wichs, Sophia Yakoubov, Arkady Yerukhimovich, and Angela Zottarel.

The BU security group created a fluid and fun environment to do taxing research. Thanks to Foteini Baldimtsi, Raef Bassily, Kyle Brogle, Yilei Chen, Danny Cooper, Jeff Finklestein, Phillipa Gill, Jef Guarente, Ethan Heilman, Abhishek Jain, Bhavana Kanukurthi, Rachel Lin, Robert Lychev, Aanchal Malhotra, Xianrui Meng, Kobbi Nissim, Omer Paneth, Dimitris Papadopoulos, Oxana Poburinnaya, Davide Proserpio, Adam O'Neill, Sofya Raskhodnikova, Natali Ruchansky, Adam Smith, Nikos Triandopoulos, Rita Vald, Sachin Vasant, Sophia Yakoubov, and Angela Zotteral!

In addition, the staff at BU was wonderful my entire time. Charles, Chris, Ellen, Jennifer, Nora, Theresa, and Wes cleared hurdles from my path and let me focus on research.

While at Boston University, I worked at MIT Lincoln Laboratory and received mentorship from many great and experienced researchers. Of particular note are Rob Cunningham, Jon Herzog, Gene Itkis, and Roger Khazan. Rob has encouraged me throughout this entire process. From the moment I arrived at Lincoln, he deftly balanced my educational goals with project needs. Rob and I met weekly to discuss paper writing and how to organize thoughts. Rob changed graduate school from a possibility to a necessity. Roger helped build and expand my scientific exploration. With Roger, I wrote my first paper as the primary researcher. This process had significant growing pains. It was a challenge to have my ideas rephrased, streamlined, and changed. Looking back, this process was important and necessary for my intellectual growth. Jon served as my mentor for the majority of my graduate studies. He served as an independent source of advice to navigate political and intellectual issues. Lastly, Gene has helped me understand the practical importance of the research contained in this dissertation. Together, we explored the problem space and identified important parameters.

My family and friends have been instrumental in reaching this point. It is impossible to list all the important influences on my life but thanks to Ange, Brendan, Chris, Don, Hanah, Josh, John, Julie, Ken, Krystal, Lisa, Matt, Melissa, Merrielle, Reed, Suzy, and Yasha. Lisa, Merrielle, and Natali pushed me through the writing process. There were many late night panicked phone calls; they were a source of steady and continual support. I am blessed with friends that always pick up the phone. My support system instilled the foolish notion that I would finish this dissertation.

Lastly, none of this would have been possible without tremendous support and love from my parents Jeff and Susan. They always backed my choices. Nothing has off-limits or impossible. They encouraged exploration. Perhaps most importantly, they instilled the importance of hard work and showing up. To quote Randy Pausch, I won the "the parent lottery."

STRONG KEY DERIVATION FROM NOISY SOURCES

(Order No.)

BENJAMIN WOODBURY FULLER

Boston University, Graduate School of Arts and Sciences, 2015

Major Professor: Leonid Reyzin,
Associate Professor of Computer Science

ABSTRACT

A shared cryptographic key enables strong authentication. Candidate sources for creating such a shared key include biometrics and physically unclonable functions. However, these sources come with a substantial problem: noise in repeated readings.

A fuzzy extractor produces a stable key from a noisy source. It consists of two stages. At enrollment time, the generate algorithm produces a key from an initial reading of the source. At authentication time, the reproduce algorithm takes a repeated but noisy reading of the source, yielding the same key when the two readings are close. For many sources of practical importance, traditional fuzzy extractors provide no meaningful security guarantee.

This dissertation improves key derivation from noisy sources. These improvements stem from three observations about traditional fuzzy extractors.

First, the only property of a source that standard fuzzy extractors use is the entropy in the original reading. We observe that additional structural information about the source can facilitate key derivation.

Second, most fuzzy extractors work by first recovering the initial reading from the noisy reading (known as a secure sketch). This approach imposes harsh limitations on the length of the derived key. We observe that it is possible to produce a consistent key without recovering the original reading of the source.

Third, traditional fuzzy extractors provide information-theoretic security. However, security against computationally bounded adversaries is sufficient. We observe fuzzy extractors providing computational security can overcome limitations of traditional approaches.

The above observations are supported by negative results and constructions. As an example, we combine all three observations to construct a fuzzy extractor achieving properties that have eluded prior approaches. The construction remains secure even when the initial enrollment phase is repeated multiple times with noisy readings. Furthermore, for many practical sources, reliability demands that the tolerated noise is larger than the entropy of the original reading. The construction provides security for sources of this type by utilizing additional source structure, producing a consistent key without recovering the original reading, and providing computational security.

Contents

1 Introduction **1**

1.1 Overview of Contributions 4

1.1.1 Organization 7

1.2 Incorporating Structure of a Noisy Distribution 7

1.2.1 Techniques 10

1.2.2 Related Settings 12

1.3 Looking Beyond Sketch-then-Extract 13

1.4 Moving to Computational Security 16

1.4.1 Minimizing entropy loss 17

1.4.2 Adding reusability 18

1.4.3 Allowing correlated symbols 20

2 Preliminaries **22**

2.1 Entropy and Extraction 23

2.1.1 Min-Entropy 23

2.1.2 Randomness Extractors 23

2.2 Computational Tools 23

2.2.1 Computational Entropy 24

2.2.2 Extracting from Computational Entropy 24

2.3 Coding Theory 26

2.3.1 Shannon Codes 26

2.3.2 Hamming Codes 28

2.3.3 Random Linear Codes 29

2.4 Obfuscation 30

3 Key Derivation from Noisy Sources **32**

3.1 Key Derivation from Correlated Random Variables 33

3.2 The Interactive Setting 34

3.3 Fuzzy Extractors 36

3.3.1 Previous approaches 38

3.3.2 Computational Fuzzy Extractors 39

3.3.3 Reusable Computational Fuzzy Extractors 41

4 Measuring the Strength of a Noisy Distribution **45**

4.1 Fuzzy Min-Entropy: a Necessary Condition 45

4.2 Sufficiency of $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}$ When the Algorithms Know the Distribution . . . 47

4.2.1 Flat Distributions 47

4.2.2 Arbitrary Distributions 49

4.3 Impossibility of Secure Sketches for a Family with $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}$ 54

4.4 Impossibility of Fuzzy Extractors for a Family with $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}$ 59

5 Looking Beyond Sketch-then-Extract **66**

5.1 Impossibility of Computational Secure Sketches 66

5.1.1 Bounds on Secure Sketches using HILL entropy 66

5.1.2 Bounds on Secure Sketches using Unpredictability Entropy . . 70

5.1.3 Implications of negative results 72

5.2 Supporting more errors than entropy 73

6 Moving to Computational Security **78**

6.1 Computational Fuzzy Extractor based on LWE 78

6.1.1 Security of Construction 6.1.1 79

6.1.2 Efficiency of Construction 6.1.1 81

6.1.3 Lossless Computational Fuzzy Extractor 85

6.1.4 Comparison with computational extractor-based constructions 86

6.2 Moving to Nonuniform Sources 87

6.3 Proof of Theorem 6.2.2 89

6.4 Parameter Settings for Construction 6.1.1 94

6.4.1 Parameter Settings for Theorem 6.2.3 97

7 Computational Fuzzy Extractors from Point Obfuscation 100

7.1 A reusable computational fuzzy extractor 100

7.1.1 Security of Construction 7.1.1 102

7.1.2 Correctness of Construction 7.1.1 106

7.1.3 Reusability of Construction 7.1.1 107

7.2 Allowing Correlated Symbols 109

7.2.1 Security of Construction 7.2.3 111

7.2.2 Correctness of Construction 7.2.3 115

A A Definitional Equivalence 117

B Characterizing unguessable block sources 118

B.1 Positive Examples 118

B.2 Negative Examples 121

References 123

Curriculum Vitae 131

List of Figures

6·1 A PPT D' that distinguishes LWE using distinguisher for LWE w/ block fixing source 90

List of Abbreviations

$\mathcal{A}$-adversary.

B_t-ball around a point of radius t.

C-set of codewords of error-correcting code.

$\mathtt{cext}$-computational extractor. See Definition 2.2.5.

D-distinguisher.

d-length of seed in extractor.

$\mathcal{D}$-class of distinguishers.

dis-distance function for metric space.

$\mathtt{ext}$-randomness extractor. See Definition 2.1.3.

$\mathbb{F}$-finite field.

F-universal hash function.

GAPSVP-Gap Shortest Vector Problem. See [Reg10].

Gen-generate of fuzzy extractor. See Definition 3.3.1.

H_∞-min-entropy of random variable.

$\tilde{\mathrm{H}}_\infty$-conditional min-entropy of random variable.

H_0-support size of random variable. Also known as Hartley entropy.

$\tilde{H}_0$-conditional support size. Average Hartley entropy.

$H^{\mathtt{HILL}}_{\epsilon,s}$-HILL pseudoentropy. See Definition 2.2.1.

$H^{\mathtt{HILL\text{-}rlx}}_{\epsilon,s}$-relaxed HILL pseudoentropy. See Definition 2.2.2.

$\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}$-fuzzy min-entropy. See Definition 4.1.1.

$\mathtt{HILL}$-shorthand for standard definition of pseudoentropy (named for [HILL99]).

$H^{\mathtt{unp\text{-}rlx}}_{\epsilon,s}$-relaxed unpredictability entropy. See Definition 2.2.3.

key-key generated by fuzzy extractor. Key indicates random variable.

K-key for universal hash. The random variable is $\mathcal{K}$.

LWE-The learning with errors assumption. See [Reg10].

$\mathcal{M}$-metric space.

m-starting entropy of source.

$\tilde{m}$-residual entropy after either running fuzzy extractor or secure sketch.

n-security parameter. Often we consider metric spaces of size proportional to 2^n.

$\mathtt{ngl}$-shrinks faster than any inverse polynomial in the input or security parameter.

NP-non-deterministic polynomial time. See [Sip12].

Neigh_t-neighborhood of a point in the metric space. See Definition 2.3.1.

P-public info generated by fuzzy extractor. Lowercase for a specific instance.

$\mathtt{poly}$-polynomial size in the input parameter or security parameter.

PUF-physically unclonable function. See [PRTG02].

Rec-recover for secure sketch. See Definition 3.3.2.

Rep-reproduce of fuzzy extractor. See Definition 3.3.1.

seed-public value of randomness extractor. Length denoted by d.

$\mathbf{SD}$-statistical distance.

SIVP-Short independent vectors problem. See [Reg10].

SS-secure sketch. See Definition 3.3.2.

ss-public value produced by SS.

t-desired error tolerance.

U-uniform distribution, often parameterized by length.

W-source distribution.

$\mathcal{W}$-family of distributions accepted to fuzzy extractor.

w-initial reading.

w'-subsequent noisy reading provided to fuzzy extractor.

Wgt-Hamming weight of a string. Number of nonzero symbols.

$\mathcal{Z}$-alphabet for Hamming metric.

γ-number of symbols for Hamming constructions.

δ-error of fuzzy extractor.

ϵ-advantage of distinguisher.

κ-key length.

Chapter 1

Introduction

In today's online world, personal information is distributed among many services. Private details such as health records, bank accounts, and private relationships are stored online. A service storing sensitive details should authenticate a user's identity before granting access to resources. The standard mechanism for authenticating identity is a password shared between a user and the service.

Passwords are easy to deploy, update, and revoke. However, passwords have a significant weakness. Ideally, passwords would consist of random characters to make password guessing infeasible, however, there is a strong tradeoff between password strength and memorability [YBAG04, WACS10]. Large-scale system compromises have revealed large files of hashed password, allowing attackers to perform brute-force guessing attacks against passwords [Lys]. There is strong evidence that the average user's password can be guessed by a determined attacker [WACS10]. The *entropy* (uncertainty) of authentication information is critical.

There are two natural alternatives to passwords, something the user *has* or something the user *is* [KH11]. We collectively refer to one of these alternatives as a source. While many sources have higher entropy than passwords, they present a new problem. Sources instantiated from physical phenomena are often *noisy* [Dau04, MRW02, PRTG02, TSŠ+06]. That is, repeated readings from the same physical source are close (according to some distance metric) but not identical.

The classic way to use a noisy source for authentication is to take an initial reading of the source and store this reading as a template. Then subsequent readings are accepted if they are close enough. This approach has two significant weaknesses: 1) the original template can be stolen and used for enrollment [GRGB+12], and 2) there is a binary decision on access that can be manually forced to output accept [RCB03]. An alternative is to directly derive keys from noisy sources. However, when trying to derive a stable and consistent key, noise becomes a substantial problem.

Dodis, Ostrovsky, Reyzin, and Smith [DORS08] designed fuzzy extractors to derive keys from noisy sources. Let w represent an initial reading of the source and w' a nearby reading. A fuzzy extractor consists of two algorithms. Generate (Gen) takes w as an input, and produces key and some helper information p. The second algorithm Reproduce (Rep) takes a nearby reading w' and the helper information, p. We assume that the distance between w and w' is at most t. Rep and Gen should produce the same key if $\mathsf{dis}(w, w') \leq t$. History has shown that stored authentication information is often compromised, so key should be cryptographically strong even if an attacker knows the helper data p.

Bennett, Brassard, and Robert identified two crucial tasks for deriving keys from noisy data [BBR88].[1] The first, information-reconciliation removes errors from w'. The second, privacy amplification converts w to a uniform value. Traditionally, a fuzzy extractor uses two separate algorithms to accomplish these tasks. A secure sketch [DORS08] performs information-reconciliation and a randomness extractor [NZ93] performs information-reconciliation. We call a fuzzy extractor that separates information-reconciliation and privacy amplification the *sketch-and-extract* construction. In this work, we concentrate on fuzzy extractors and secure sketches.[2] A secure sketch consists of two algorithms: SS takes w and produces a public value ss, and Rec takes a nearby w' and ss to recover w. The goal of fuzzy extractors is to ensure that w has high entropy conditioned on ss.

[1] Bennett, Brassard, and Robert consider an interactive version of the problem. We discuss their setting in Section 3.2.

[2] Randomness extractors have matching upper and lower bounds on the security loss: for every extra two bits of output key, they lose one bit of security.

Limitations of Standard Techniques Fuzzy extractors and secure sketches must contain some information about the initial reading w in order to accept nearby w'. We call a point w' accepting if it is within distance t of the original reading w. A larger t means that more w' are accepting. For a fuzzy extractor, the adversary can use Rep on accepting w' to produce key. (For a secure sketch, the adversary can use Rec on accepting w' to obtain w.) This means if an adversary can find an accepting w' with noticeable probability, they can learn key with noticeable probability and break security. Key derivation becomes more difficult as more points are accepting. This creates a tension between the length of key and the error tolerance t.

This tension is quite strong for secure sketches. Secure sketches are closely linked to error correcting codes.[3] The syndrome of a linear code is used to compute where errors occurred in transmission. The syndrome can also serve as a secure sketch (Construction 3.3.5). The entropy of w conditioned on this secure sketch is at least the starting entropy minus the length of the syndrome. Standard analysis assumes this is the remaining entropy in w. The length of a syndrome increases as the error tolerance t increases. This means the lower bound on the remaining entropy of w decreases as t increases.[4]

This is not a limitation of this particular secure sketch. Dodis et al. show that secure sketches are linked to the best error-correcting code containing points of w [DORS08, Appendix C]. Upper bounds on the size of error-correcting codes translate to lower bounds on entropy loss of secure sketches. Error-correcting codes have a long and rich history, with many bounds on the best codes. All of these bounds are translate into limitations on the best secure sketches. Most fuzzy extractors use secure sketches (we discuss some exceptions in Section 1.2.1) for error correction. Fuzzy extractors that use secure sketches also inherit these bounds.

[3] We provide a limited introduction to error-correcting codes in Section 2.3.2.

[4] If a perfect error-correcting code is used, there are distributions with a matching upper bound on the remaining entropy. That is, there are distributions where the standard analysis is tight.

Standard fuzzy extractors and secure sketches only take as parameters the entropy of w and the number of errors to be corrected. However, secure sketches are connected to the best code containing *points of* w. This varies widely for different sources w. If all points of w are far apart, then correcting t errors is easy. Without information about the distribution of w, standard secure sketches must work for all distributions of entropy m and errors t. The worst case distribution has all points close together. For this distribution, to recover the correct w' from an original reading w, the value ss must disambiguate which point w within distance t was seen. For this distribution, a secure sketch must decrease entropy by the logarithm of the number of points within distance t of any w'. Thus, if a secure sketch provides a guarantee for the worst distribution, the bound on entropy loss is proportional to this quantity (in most settings, the number of points within distance t is exponential in t).

Losses due to secure sketches (and the resulting fuzzy extractors), prevent key derivation from many practical sources. For many sources, there are no known fuzzy extractors that provide meaningful security. As an example, the human iris is thought to be the strongest biometric [Dau04] and current fuzzy extractors provide no guarantee about the strength of a key derived from the human iris [BH09, Section 5].

1.1 Overview of Contributions

In this dissertation, we improve key derivation from noisy sources. Our improvements derive from three lessons about how to construct fuzzy extractors. We organize this dissertation around these lessons. We list the lessons below. Under each lesson we list our major technical results that serve as supporting evidence.

- **Incorporating structure of a noisy distribution** Traditional fuzzy extractors consider the worst-case distribution with entropy m for a desired error-tolerance t. We know more about the structure of physical sources. It may be possible to avoid the losses of traditional approaches by constructing fuzzy extractors whose security analysis uses structure of a physical source beyond its entropy. This motivates us to modify the definition of fuzzy-extractors to work for limited family of distributions rather than all distributions with entropy m. We describe a precise measure of a noisy distribution's suitability for key derivation called *fuzzy min-entropy*. Fuzzy min-entropy is a necessary condition for key derivation (Proposition 4.1.2).

  - Theorem 4.2.7: *Fuzzy min-entropy* is sufficient for security if a distribution is known exactly. This motivates fuzzy min-entropy as the right measure of a distribution's suitability. Furthermore, it shows that precise knowledge of a source's distribution allows key derivation.
  - Theorems 4.3.1 and 4.4.1: Unfortunately, it is imprudent to assume that high entropy distributions are known precisely. This uncertainty is handled by providing security for a family of distributions. We show there are families of distributions (where each distribution has fuzzy min-entropy) that no information-theoretic secure sketch or fuzzy extractor can provide meaningful security for most members of the family. This shows that uncertainty of a source's distribution comes at a cost to security.

- **Look beyond sketch-then-extract** Secure sketches are subject to considerably stronger negative results than fuzzy extractors. We provide additional negative results for computationally secure versions of secure sketches. We then construct improved fuzzy extractors that do not use secure sketches.

  - Corollary 5.1.5: Computational definitions of secure sketches are subject to upper bounds on remaining entropy. If computational secure sketches are defined using pseudoentropy, they are subject to almost the same bounds as information-theoretic secure sketches.

  - Construction 5.2.2: For many practical sources, reliability demands that the number of tolerated error patterns is greater than the starting entropy of the source. We call this condition *more errors than entropy*. Previous approaches have provided no security for such distributions. One cannot provide security for all such sources, restricted to a limited class of distributions is necessary (discussion in Section 1.3). We construct the first fuzzy extractor secure for large classes of distributions with more errors than entropy. This construction does not use a secure sketch.

- **Leverage Computational Security** Fuzzy extractors were defined with information theoretic security due to the use of information-theoretic tools. However, there is no compelling need for information-theoretic security. Fuzzy extractors can be improved by providing computational security. We provide computational constructions with new features. All of our constructions are for the Hamming metric (the number of symbols that differ between strings w and w').

  - Construction 6.1.1: A computational fuzzy extractor whose key is as long as the input entropy.
  - Construction 7.1.1: A computational fuzzy extractor that allows a source to be securely enrolled across multiple services. This is known as a reusable fuzzy extractor (see Definition 3.3.9).[5]
  - Construction 7.2.3: A computational fuzzy extractor that improves on the class of sources and error tolerance of the previous construction. These improvements come at a cost of a large symbol size in w.

[5] The work of [Boy04] contains some limited positive results on reusable fuzzy extractors. We discuss these in Section 1.4.2.

1.1.1 Organization

The results in this dissertation are drawn from three works [FMR13, CFP$^+$14, FRS14]. We cover preliminaries and common notation in Chapter 2. We discuss key derivation from noisy sources and fuzzy extractors in Chapter 3. We organize this dissertation by the lessons: incorporating structure of a noisy source (Section 1.2, technical results in Chapter 4), moving away from sketch-then-extract (Section 1.3, technical results in Chapter 5), and providing computational security (Section 1.4, technical results in Chapters 6 and 7).

1.2 Incorporating Structure of a Noisy Distribution

The goal of this section is to more precisely characterize the quality of a noisy distribution for key derivation. We begin by introducing a new notion that describes a noisy distribution's suitability for key derivation. The technical results described in this section can be found in Chapter 4.

Fuzzy Min-Entropy Usually, fuzzy extractors only take as parameters the entropy m of a source and desired error tolerance t. However, this ignores crucial structural information about the distribution W. The number and weight of points contained in neighborhoods of W is crucial for key derivation. We introduce a new entropy notion that combines entropy and error tolerance into a single measure. It measures a noisy distribution's suitability for key derivation.

Consider an adversary that tries to guess values w' close to the original reading w (without considering the helper string). If an adversary is able to guess some w' within distance t of the original reading w, they can subvert the security of key by running Rep. To have the maximum chance that w' is within distance t of w, the adversary would want to maximize the total probability mass of W within the ball

$B_t(w')$ of radius t around w'. We therefore define *fuzzy min-entropy*

$$\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) \overset{\text{def}}{=} -\log \max_{w'} \Pr[W \in B_t(w')].$$

Observe that this quantity can be bounded in terms of min-entropy: $\mathrm{H}_\infty(W) \geq \mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) \geq \mathrm{H}_\infty(W) - \log |B_t|$.

Super-logarithmic fuzzy min-entropy is *necessary* for nontrivial key extraction (Proposition 4.1.2). However, existing constructions do not measure their security in terms of fuzzy min-entropy; instead, their security is shown to be $\mathrm{H}_\infty(W)$ minus some loss that is at least $\log |B_t|$ due to error-tolerance. Since $\mathrm{H}_\infty(W) - \log |B_t| \leq \mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W)$, it is natural to ask whether this loss is necessary. This question is particularly relevant when the gap between the two sides of the inequality is high. As an example, iris scans appear to have significant $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W)$ (because iris scans for different people appear to be well-spread in the metric space [Dau06]) but negative $\mathrm{H}_\infty(W) - \log |B_t|$ [BH09, Section 5].[6] We therefore ask: *is fuzzy min-entropy sufficient for fuzzy extraction?* There is evidence that it may be when the security requirement is computational rather than information-theoretic—see Section 1.2.2.

Tight Characterization for the Case of a Known Distribution We show that for every source W with super-logarithmic $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W)$, it is possible to construct a fuzzy extractor with a super-logarithmic length key (Corollary 4.2.8). We thus show that $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W)$ is a necessary and sufficient condition for building a fuzzy extractor for a *known distribution* W. It is important to emphasize that these constructions incorporate the knowledge of the complete distribution of W (and, in particular, they are not polynomial-time).

A number of previous works in this known-distribution setting have provided

[6] When $\mathrm{H}_\infty(W) - \log |B_t|$ is negative we say a source has more errors than entropy. We discuss this condition further in Section 1.3.

efficient algorithms and tight bounds for specific distributions—generally the uniform distribution or i.i.d. sequences (for example, [JW99, LT03, TG04, HAD06, WRDI12, IW12]). Our characterization may be seen as unifying previous work, and justifies using $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W)$ as the measure of the quality of a noisy distribution, rather than cruder measures such as $\mathrm{H}_\infty(W) - \log|B_t|$.

Impossibility of Fuzzy Extractors for Families of Distributions Assuming full knowledge of a distribution is often unrealistic. Indeed, high-entropy distributions can never be fully observed directly and must therefore be modeled. It is imprudent to assume that the designer's model of a distribution is completely accurate—the adversary, with greater resources, would likely be able to build a better model. Therefore, fuzzy extractor designs cannot usually be tailored to one particular source. Existing designs work for a family of sources (for example, all sources of min-entropy at least m with at most t errors). Thus, the design is fixed before the distribution is fully known, and the adversary may know more about the distribution than the designer of the fuzzy extractor.

We show that this extra adversarial knowledge can be devastating (Theorem 4.4.1). Specifically, we describe a family of distributions $\mathcal{W}$ and show that not even a 2-bit fuzzy extractor can be secure for most distributions in $\mathcal{W}$. We emphasize that each distribution $W \in \mathcal{W}$ has super-logarithmic fuzzy min-entropy—in fact, $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \mathrm{H}_\infty(W)$, because all points in W are distance at least t apart. This result shows that distributional uncertainty is a real obstacle to key derivation from noisy sources. Our proof relies on high dimensionality of W and on perfect correctness of the Rep procedure.

Stronger Results for Secure Sketches As described above, fuzzy extractors often use secure sketches to perform information reconciliation (mapping w' back to

w).

We show comparable, but stronger, results for secure sketches. Namely, we show in Corollary 4.2.8 that secure sketches are possible if the distribution W is precisely known. (In fact, we obtain our fuzzy extractors for the case of a known distribution from this result by applying a randomness extractor.)

On the other hand, there is a family of sources with super-logarithmic $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \mathrm{H}_\infty(W)$ for which no secure sketch correcting even a few errors is possible (Theorem 4.3.1). The impossibility result applies even when Rec is allowed to be incorrect with probability up to $1/4$ (as opposed to our fuzzy extractor impossibility result).

1.2.1 Techniques

Techniques for Positive Results for Known Distributions We now explain how to construct a secure sketch for an arbitrary known distribution W. We begin with distributions in which all points in the support have the same probability (so-called "flat" distributions). Consider some subsequent reading w'. To achieve correctness, the sketch algorithm must disambiguate which point $w \in W$ within distance t of w' was sketched. Disambiguating multiple points can be accomplished by universal hashing, as long as the size of hash output space is slightly greater than the number of possible points. Thus, our sketch is computed via a universal hash of w. To determine the length of that sketch, consider the heaviest (according to W) ball of radius t. Because the distribution is flat, it is also the ball with the most points of nonzero probability. Thus, the length of the sketch needs to be slightly greater than the logarithm of the number of non-zero probability points in that ball. Since $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W)$ is determined by the weight of that ball, the number of points cannot be too high and there will be entropy left after the sketch is published.

For an arbitrary distribution, we cannot afford to disambiguate points in the ball with the greatest number of points, because there could be too many low-probability points in a single ball despite a high $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W)$. We solve this problem by splitting the arbitrary distribution into a number of nearly flat distributions we call "levels." We then write down, as part of the sketch, the level of the original reading w and apply the above construction considering only points in that level. We call this construction *leveled hashing*.

Techniques for Negative Results for Distribution Families We construct a family of distributions $\mathcal{W}$ and prove impossibility for a uniformly random $W \in \mathcal{W}$ (instead of proving impossibility for a worst-case W). We start by observing the following asymmetry: Gen sees only the sample w (obtained via $W \leftarrow \mathcal{W}$ and $w \leftarrow W$), while the adversary knows W. To exploit the asymmetry, we construct $\mathcal{W}$ so that conditioning on the knowledge of W reduces the distribution to a single affine line, but conditioning on w leaves the rest of the distribution uniform on a large fraction of the entire space.

Then we show how the adversary can exploit the knowledge of the affine line to reduce the uncertainty about w (in the secure sketch case) or key (in the fuzzy extractor case). In the secure sketch case, ss can be used to find fixed points of $\mathsf{Rec}(\cdot, ss)$ which, by the correctness requirement of the sketch, must be separated by minimum distance t. This means there aren't too many of them, so few can lie on an average line, permitting the adversary to guess one easily.

In the fuzzy extractor case, the nonsecret value p partitions the metric space into regions that produce a consistent value under Rep (preimages of each key under $\mathsf{Rep}(\cdot, p)$). For each of these regions, the adversary knows that possible w lie t-far from the boundary of the region. However, in the Hamming space, the vast majority of points lie near the boundary (this follows by combining the isoperimetric inequality [Har66] showing that the ball has the smallest boundary and Hoeffding's inequality [Hoe63] for bounding the volume that is t-away from this boundary). This allows the adversary to rule out so many possible w that, combined with the adversarial knowledge of the affine line, many regions become empty, leaving key far from uniform.

The result for fuzzy extractors is delicate. It uses the fact that p partitions the space into nonoverlapping regions, which is implied by perfect correctness. Extending this result to imperfect correctness seems challenging and is an interesting open problem. It also uses the fact that there are few points far from the boundary of every region, which is implied by the geometry of the high-dimensional Hamming space. This fact seems crucial: in contrast, in low-dimensional Euclidean space, which does not have this property, a single fuzzy extractor can work for any distribution with sufficient $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}$. (Such a construction would use quantization or tiling, similar to, for example, [CK03, LT03, CZC04, LC06, BDH+10, VTO+10]. Each sample from W would map to the "tile" containing it, from which the output key would be extracted. A randomly chosen quantizer would have the property that few samples lie near the boundary, giving almost-perfect correctness; if perfect correctness is desired, we can give up on security for those rare samples and simply use a special value of p to indicate that one of them was the input.)

1.2.2 Related Settings

Other settings with close readings: $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}$ is sufficient The security definition of fuzzy extractors and secure sketches can be weakened to protect only against computationally bounded adversaries [FMR13]. In this computational setting, fuzzy extractors and secure sketches can be constructed for the family of all distributions W with super-logarithmic $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}$ by using virtual grey-box obfuscation for all circuits [BCKP14]. The construction places into p the obfuscated program for testing proximity to w and outputting the appropriate value if the test passes. In addition to relying on strong assumptions for security (namely, the existence of semantically-secure multilinear maps), this construction is not of practical efficiency. Note that if this construction is used for a secure sketch, W will remain unpredictable conditioned on p, but will not have pseudoentropy (see Section 5.1.3 for details).

Furthermore, the functional definition of fuzzy extractors and secure sketches can be weakened to permit interaction between the party having w and the party having w' (we discuss this setting in Section 3.2). Such a weakening is useful for secure remote authentication [BDK$^+$05]. When both interaction and computational assumptions are allowed, secure two-party computation can produce a key that will be secure whenever the distribution W has fuzzy min-entropy. The two-party computation protocol needs to be secure without assuming authenticated channels; it can be built under the assumptions that collision-resistant hash functions and enhanced trapdoor permutations exist [BCL$^+$11].

Correlated rather than close readings A different model for the problem of key derivation from noisy sources does not explicitly consider the distance between w and w', but rather views w and w' as samples drawn from a correlated pair of random variables. This model is considered in multiple works, including [Wyn75, CK78, AC93, Mau93]; recent characterizations of when key derivation is possible in this model include [RW05] and [TW14]. We discuss this model in Section 3.1.

1.3 Looking Beyond Sketch-then-Extract

Secure sketches have significantly stronger results in the information-theoretic setting than fuzzy extractors. This is because secure sketches must precisely reproduce the original reading w while fuzzy extractors only need to produce a consistent value. In this section, we provide additional negative results on secure sketches, describe how to avoid secure sketches, and then describe a fuzzy extractor achieving a condition that has eluded secure sketches. We describe technical results in Chapter 5.

Computational secure sketches are also limited We ask whether negative results on secure sketches can be overcome by relaxing the definition to provide computational security (in Section 5.1). Recall, a secure sketch produces a public value ss used to reconstruct the original reading w. The traditional secrecy requirement is that w has high min-entropy conditioned on ss. This allows the fuzzy extractor of [DORS08] to form key by applying a randomness extractor [NZ93] to w, because randomness extractors produce random strings from strings with conditional min-entropy.

The most natural relaxation of the min-entropy requirement of the secure sketch is to require HILL entropy [HILL99] (namely, that the distribution of w conditioned on ss be *indistinguishable* from a high min-entropy distribution). Under this definition, we could still use a randomness extractor to obtain key from w, because it would yield a pseudorandom key. Unfortunately, it is unlikely that such a relaxation will yield fruitful results: we prove in Theorem 5.1.3 that the entropy loss of such secure sketches is subject to the same coding bounds as the ones that constrain information-theoretic secure sketches.

Another possible relaxation is to require that the value w is unpredictable conditioned on ss. This definition would also allow the use of a randomness extractor to get a pseudorandom key, although it would have to be a special extractor—one that has a reconstruction procedure (see [HLR07, Lemma 6]). We show a significantly weaker negative result for unpredictability entropy: we prove in Theorem 5.1.8 that the unpredictability is at most log the size of the metric space minus log the volume of the ball of radius t. For nearly uniform sources of w over the Hamming metric, this bound matches the best information-theoretic security sketches. However, for lower entropy sources this bound is not meaningful. Indeed, the result of [BCKP14] can be seen as constructing unpredictability secure sketches for all distributions with fuzzy min-entropy.

Constructing Fuzzy Extractors without Sketches Our negative results arise because Rec function acts as an error-correcting code for points of indistinguishable distributions. It is possible to avoid these negative results by outputting a fresh random variable.[7] Such an algorithm is called a fuzzy conductor [KR09]. Looking ahead, we construct information-theoretic fuzzy conductors, fuzzy conductors that have computational security, and fuzzy extractors with computational security. Our constructions will exploit the structure of the physical source beyond its entropy. We now describe a condition on practical sources that necessitates the use of some type of structure of a physical source (beyond its entropy).

More errors than entropy Fuzzy extractors and secure sketches have an inherent tension between security and correctness guarantees. Consider a distribution with starting entropy m and desired error tolerance t. If t is high enough that there are 2^m points in a ball of radius t, then there exists a distribution of w of min-entropy m *contained entirely in a single ball*. This distribution has no fuzzy min-entropy and thus cannot be securely used for key derivation. Thus, if the security guarantee of a given fuzzy extractor holds for *any* source of a given min-entropy m and the correctness guarantees holds for any t errors, then m must be greater than $\log |B_t|$.[8] If a source fails this condition, we will says that it has *more errors than entropy*. Distributions with more errors than entropy may have fuzzy min-entropy.

[7] If some efficient algorithm can invert this fresh value and recover W, the bounds of Corollary 5.1.5 and Theorem 5.1.8 both apply. This means that we need to consider constructions that are hard to invert (either information-theoretically or computationally).

[8] Fuzzy min-entropy is also a necessary condition for the computational and interactive settings (Proposition 4.1.2). Thus, even in these relaxed settings, to achieve security for all sources of a given entropy m and error level t, $m > \log |B_t|$. This further motivates our first lesson to incorporate the structure of a distribution.

A fuzzy extractor for more errors than entropy Current techniques for building secure sketches do not work for sources with more errors than entropy, because they lose at least $\log |B_t|$ bits of entropy regardless of the source. The negative results above show these limitations are unlikely to be overcome by secure sketches that retain pseudoentropy.

We provide the first construction of a fuzzy extractor that can be used for large classes of sources that have more errors than entropy (Construction 5.2.2). Our construction works for Hamming errors for strings w of length γ over some alphabet $\mathcal{Z}$. As argued above, our construction cannot work for all sources of a given entropy; Our construction can correct a constant fraction of errors, but requires that a constant fraction of the symbols contribute fresh entropy, even conditioned on previous symbols (Definition 5.2.3). This type of source is a subset of all sources with fuzzy min-entropy.

Our construction reduces the alphabet size by hashing each input symbol (which comes from a large alphabet) into a much smaller set, so that the resulting hash value has lower entropy deficiency. The intuition behind this approach is that it reduces the size of B_t by reducing the alphabet size, but preserves a sufficient portion of the input entropy. The resulting string no longer has more errors than entropy. We then apply a standard fuzzy extractor to the resulting string.

1.4 Moving to Computational Security

In the previous two sections, we showed that fuzzy extractors can be improved by providing security for families of distributions with additional structure (instead of all distributions with a given entropy) and by giving up on sketch-then-extract. In this section, we can provide further improvements by providing *computational* instead of information theoretic security. We construct three fuzzy extractors with computational security with novel properties. The first construction uses random linear codes, the second and third constructions use point obfuscation. For this reason, we split their discussion to Chapter 6 and Chapter 7 respectively. All construction in this section are for the Hamming metric. We assume w of length γ over some alphabet $\mathcal{Z}$.

1.4.1 Minimizing entropy loss

By considering this computational secrecy requirement, we construct the first fuzzy extractor (Construction 6.1.1), where key is as long as the entropy of the source w. Our construction uses the code-offset construction [JW99],[DORS08, Section 5] used in prior work, but with two crucial differences. First, key is not extracted from w like in the sketch-and-extract approach; rather w "encrypts" key in a way that is decryptable with the knowledge of some close w' (this idea is similar to the way the code-offset construction is presented in [JW99] as a "fuzzy commitment"). Second, the code used is a random linear code, which allows us to use the Learning with Errors (LWE) assumption due to Regev [Reg05, Reg10] and derive a longer key.

Specifically, we use the result of Döttling and Müller-Quade [DMQ13], which shows the hardness of decoding random linear codes when the error vector comes from the uniform distribution, with each coordinate ranging over a small interval. This allows us to use w as the error vector, assuming it is uniform. We also use a result of Akavia, Goldwasser, and Vaikuntanathan [AGV09], which says that LWE has many hardcore bits giving us a key.

Because we use a random linear code, our decoding is limited to reconciling a logarithmic number of differences. Unfortunately, we cannot utilize the results that improve the decoding radius through the use of trapdoors (such as [Reg05]), because in a fuzzy extractor, there is no secret storage place for the trapdoor. If improved decoding algorithms are obtained for random linear codes, they will improve error-tolerance of our construction. Given the hardness of decoding random linear codes [BMvT78], we do not expect significant improvement in the error-tolerance of our construction for general physical sources.

In Section 6.2, we are able to relax the assumption that w comes from the uniform distribution, and instead allow w to come from a symbol-fixing source [KZ07] (each dimension is either uniform or fixed). This relaxation follows from a result about the hardness of LWE when samples have a fixed (and adversarially known) error vector, which may be of independent interest (Theorem 6.2.2).

Improving Error Tolerance Construction 6.1.1 only tolerates a logarithmic number of errors. Most practical sources have substantially more errors. Subsequent to our construction, Herder et al. improved the error-tolerance of this construction for physical sources with an additional property [HRvD$^+$14]. For some physical sources it is possible to obtain a confidence vector with the subsequent reading w'. This confidence vector indicates how likely each symbol of w' is to contain an error. This confidence information can greatly the error tolerance of Construction 6.1.1 (from logarithmic number to a linear fraction of errors).[9] Furthermore, Herder et al. show that a ring oscillator physical unclonable function [SD07] produces such confidence information. If confidence information is not available the construction of Herder et al. reduces to our construction with logarithmic error tolerance. Finding other physical sources with similar confidence information is an open problem.

1.4.2 Adding reusability

A desirable security property of fuzzy extractors, introduced by Boyen [Boy04], is called reusability. This property is necessary if a user enrolls the same or correlated values multiple times. For example, if the source is a biometric reading, the user may

[9] Herder et al. base their construction on the learning parity with noise problem [BKW03]. Their approach can easily be extended to larger fields and the learning with errors problem.

enroll the same biometric with different organizations. Each of them will get a slightly different enrollment reading w_i, and will run $\mathsf{Gen}(w_i)$ to get key_i and a helper value p_i. Security for each key_i should hold even when an adversary is given all the values $p_1, \ldots, p_q$ (and, in case some organizations turn out to compromised or adversarial, a stronger security notion requires security for key_i even in the presence of key_j for $j \neq i$). Many traditional fuzzy extractors are not reusable [Boy04, STP09, BA12, BA13]. The only previous construction of reusable fuzzy extractors [Boy04] requires very particular relationships between w_i values, which are unlikely to hold in any practical source.

A reusable fuzzy extractor against strong correlation We construct a computational fuzzy extractor with strong reusability. Security holds even if the multiple readings w_i used in Gen are *arbitrarily correlated*, as long as each w_i *individually* comes from an allowed distribution. The construction is secure for distributions where sampling of symbols produces a high entropy output, such as those with k-wise independence among symbols for super-logarithmic k. We note that this construction also handles sources with more errors than entropy (discussed in Section 1.3). This construction requires that the fraction of errors is sub-constant. We note this construction requires each symbol of the source to contribute fresh entropy.

Approach Our reusable construction is based on obfuscated digital lockers [CD08]. Digital lockers output a secret value only when given the correct input to "unlock" the secret. An obfuscated digital locker does not provide information about the locked value or how to unlock it. The main idea of the construction is to pick a random key and lock key in a digital locker that is unlocked by a random subset of the symbols of w. To tolerate errors in the input, this process is repeated several times, so that at least one digital locker can be unlocked using w'. We use obfuscation in a way that does not leak partial information; this is crucial to arguing reusability.

1.4.3 Allowing correlated symbols

Our final construction addresses weaknesses in the previous construction. Construction 7.2.3 removes the need for fresh entropy in the symbols and allows a constant fraction of symbols of errors, at the cost of requiring a large alphabet size (super-polynomial in the security parameter). It is secure if symbols in w each have individual super-logarithmic min-entropy, even if they are arbitrarily correlated. Moreover, a constant fraction of symbols in w may have little or no entropy, as long as knowledge of their values does not reduce the entropy of the high-entropy symbols too much (see Definition 7.1.6).

Approach Our construction that allows correlated symbols tolerates more errors than the second because it uses digital lockers that are unlocked by single symbols of w. Since we do not assume that every symbol has high individual entropy, hiding all of key in every locker then becomes too risky, Instead, we hide a single bit per locker. To tolerate errors, these bits come from an error correcting code. To ensure an adversary who learns some bits doesn't learn anything useful about key, we don't encode key in the error-correcting code, but rather extract key (using an information-theoretic [NZ93] or computational [Kra10] extractor) from the decoded string.

The Required Notion of Obfuscation Constructions 7.1.1 and 7.2.3 use simulation secure obfuscation of digital lockers, however, we do not require full-fledged virtual black-box obfuscation [BGI$^+$01]. Instead, we rely on the relaxed notion of *virtual grey-box* obfuscation [BC10]. We also require that the obfuscation remains secure even when several digital lockers of correlated points are composed. Bitanski and Canetti constructed composable digital lockers with virtual grey-box security under particular number-theoretic assumptions [BC10]. Recent work of Brzuska and Mittelbach shows that if indistinguishability obfuscation exists then it is not possible to build composable virtual black-box digital lockers [BM14]. Thus, our use of virtual grey-box obfuscation is crucial.

Connection to General Obfuscation As described in Section 1.2.2, fuzzy extractors for all sources with fuzzy min-entropy can be trivially constructed from virtual grey-box obfuscation for all circuits [BCKP14]. The security of their construction is based on the strong assumption of *semantically secure graded encodings* [PST14]. The construction is based on multilinear encoding and is highly impractical. Our constructions use obfuscated digital lockers. Obfuscated digital lockers are instantiable under significantly weaker assumptions and can be implemented quite efficiently. Additionally, the known obfuscation for proximity point programs is not known to be composable and therefore does not yield a reusable fuzzy extractor.

Chapter 2

Preliminaries

Let $x \in X$ denote an element x in the support of X. Let $x \leftarrow X$ be the process of a sampling x from the distribution X. U_n is a random variable with the uniform distribution over $\{0,1\}^n$. Let $\mathbf{SD}(X,Y)$ be the statistical distance between random variables X, Y drawn from a set χ, defined as $\mathbf{SD}(X,Y) = \frac{1}{2}\sum_{x\in\chi} |\Pr(X = x) - \Pr(Y = x)|$. We consider randomized distinguishers that output a single bit. Given a circuit D, define the computational distance δ^D between X and Y as $\delta^D(X,Y) = |\mathbb{E}[D(X)] - \mathbb{E}[D(Y)]|$. For a circuit D, we use $|D|$ to denote its size. For a class of distinguishers, $\mathcal{D}_s$, each of size at most s, we write $\delta^{\mathcal{D}_s} = \max_{D\in\mathcal{D}_s} \delta^D(X,Y) = |\mathbb{E}[D(X)] - \mathbb{E}[D(Y)]|$. For a probability distribution X, let $\mathrm{supp}(X)$ denote the set of points with nonzero probability. Let $H_0(X)$ denote the logarithm of the support size of X, that is $H_0(X) = \log|\mathrm{supp}(X)|$. We use an average case notion of remaining support size $\tilde{H}_0(X|P) = \log(\mathbb{E}_{p\in P}|\mathrm{supp}(X|P = p)|)$. All logarithms without a base are considered base 2, that is, $\log x = \log_2 x$.

For a metric space $(\mathcal{M}, \mathsf{dis})$, the *(closed) ball of radius t around x* is the set of all points within radius t, that is, $B_t(x) = \{y|\mathsf{dis}(x,y) \le t\}$. If the size of a ball in a metric space does not depend on x, we denote by $|B_t|$ the size of a ball of radius t. We consider the Hamming metric over vectors in $\mathcal{Z}^\gamma$, defined via $\mathsf{dis}(x,y) = \{i|x_i \neq y_i\}$. For this metric, $|B_t| = \sum_{i=0}^{t}\binom{\gamma}{i}(|\mathcal{Z}|-1)^i$. For a vector w over $\mathbb{F}^\gamma$, let $\mathsf{Wgt}(w) = \{i|w_i \neq 0\}$.

Usually, we use capitalized letters for random variables and corresponding lowercase letters for their samples. We frequently use standard order notation (see [CLRS01]).

2.1 Entropy and Extraction

We begin introduction standard entropy and extraction notions.

2.1.1 Min-Entropy

We begin with the standard notion of min-entropy and proceed to computational notions.

Definition 2.1.1. *A distribution X has* min-entropy *at least m, denoted* $\mathrm{H}_\infty(X) \geq m$ *if*

$$\forall x \in X, \Pr[X = x] \leq 2^{-m}.$$

We use the average case notion of min-entropy defined by [DORS08].

Definition 2.1.2 ([DORS08])**.** *Let (X, Y) be a pair of random variables. The* average min-entropy *of X conditioned on Y is defined as*

$$\tilde{H}_\infty(X|Y) \overset{\text{def}}{=} -\log[\underset{Y}{\mathbb{E}}(2^{-H_\infty(X|Y)})] = -\log \sum_{y \in Y} \Pr[Y = y] 2^{-H_\infty(X|Y=y)}$$

2.1.2 Randomness Extractors

A *randomness extractor* takes a distribution X of (average) min-entropy m, and with the help of a uniform string called the seed, "extracts" the randomness contained in X and outputs a string of length κ that is *almost uniform* even given the seed.[1]

Definition 2.1.3 ([NZ93])**.** *Let $\mathcal{M}$, χ be finite sets. A function* $\texttt{ext} : \mathcal{M} \to \{0, 1\}^\kappa$ *a $(\tilde{m}, \epsilon)$-*average case extractor *if for all pairs of random variables X, Y over $\mathcal{M}, \chi$ such that $\tilde{H}_\infty(X|Y) \geq \tilde{m}$, we have* $\mathbf{SD}((\texttt{ext}(X, U_d), U_d, Y), U_\kappa \times U_d \times Y) \leq \epsilon$*.*

2.2 Computational Tools

We now describe computational notions of entropy. Our computational notions of entropy have two additional parameters: circuit size s and quality ϵ. Larger s and

[1] In all of our definitions of extractors we assume the extractor outputs its seed. We omit this from the function definition but all security definitions take in the d-bit seed.

smaller ϵ mean "better" entropy.

2.2.1 Computational Entropy

We use the average case notion [HLR07] of HILL entropy [HILL99].

Definition 2.2.1. *A joint distribution $X|Y$ has* conditional HILL entropy *at least m, denoted $H^{\mathtt{HILL}}_{\epsilon,s}(X) \geq m$ if there exists a distribution Z where $\tilde{\mathrm{H}}_\infty(Z|Y) \geq m$, such that $\delta^{\mathcal{D}_s}((X,Y),(Z,Y)) \leq \epsilon$.*

HILL entropy is a commonly used computational notion of entropy. It was extended to the conditional case by Hsiao, Lu, Reyzin [HLR07]. Here we recall a weaker definition due to Gentry and Wichs [GW11] (the term relaxed HILL entropy was introduced in [Rey11]).

Definition 2.2.2. *Let (X,Y) be a pair of random variables. W has* relaxed HILL entropy *at least m conditioned on S, denoted $H^{\mathtt{HILL\text{-}rlx}}_{\epsilon,s}(X|Y) \geq m$ if there exists a joint distribution (X',Y'), such that $\tilde{H}_\infty(X'|Y') \geq m$ and $\delta^{\mathcal{D}_s}((X,Y),(X',Y')) \leq \epsilon$.*

However, HILL entropy is a strong notion. We also consider a significantly weaker version where the value of X is hard to guess given public state. We use the definition of conditional unpredictability entropy [HLR07, Definition 7], which captures the notion of "hard to guess" (we relax the definition slightly, similarly to the relaxation of HILL entropy above).

Definition 2.2.3. *Let (X,Y) be a pair of random variables. X has* relaxed unpredictability entropy *at least m conditioned on Y, denoted by $H^{\mathtt{unp\text{-}rlx}}_{\epsilon,s}(X|Y) \geq m$, if there exists a pair of distributions (X',Y') such that $\delta^{\mathcal{D}_s}((X,Y),(X',Y')) \leq \epsilon$, and for all circuits $\mathcal{I}$ of size s,*

$$\Pr[\mathcal{I}(Y') = X'] \leq 2^{-m}.$$

2.2.2 Extracting from Computational Entropy

Extractors can be applied to distributions with HILL entropy to obtain pseudorandom, rather than random, outputs: that is, outputs that are computationally indistinguishable from, rather than statistically close to, uniformly random strings. We include a proof to provide intuition for manipulating computational entropy (a similar version of this theorem appeared in [FR12]).

Theorem 2.2.4. *Let* $\mathtt{ext} : \mathcal{M} \times \{0,1\}^d \rightarrow \{0,1\}^\kappa$ *be a* $(\tilde{m}, \epsilon_{ext})$*-extractor, computable by circuits of size* $s_{\mathtt{ext}}$*. Let* X, Y *be a distribution over* χ *with* $H^{\mathtt{HILL\text{-}rlx}}_{\epsilon_{\mathtt{HILL}}, s_{\mathtt{HILL}}}(X|Y) \geq \tilde{m}$*. Then* $\forall D \in \mathcal{D}_{s'}$*, where* $s' \approx s_{\mathtt{HILL}} - s_{\mathtt{ext}}$*,*

$$\delta^D((\mathtt{ext}(X, U_d), Y, U_d), U_m \times Y \times U_d) \leq \epsilon_{\mathtt{ext}} + \epsilon_{\mathtt{HILL}}\,.$$

Proof. We proceed by contradiction. Suppose not, that is, $\exists D \in \mathcal{D}_{s'}$ such that

$$\delta^D((\mathtt{ext}(X, U_d), Y, U_d), (U_\kappa \times Y \times U_d)) > \epsilon_{\mathtt{ext}} + \epsilon_{\mathtt{HILL}}.$$

We use D to construct a distinguisher D' to distinguish X, Y from all distributions X', Y' where $\tilde{\mathrm{H}}_\infty(X'|Y') \geq \tilde{m}$, violating the $\mathtt{HILL\text{-}rlx}$ entropy of $X|Y$. We define D' as follows: upon receiving input $\alpha \in \mathcal{M}, \beta \in \chi$, D' samples $\mathsf{seed} \leftarrow U_d$, runs $\eta \leftarrow \mathtt{ext}(\alpha, \mathsf{seed})$ and then runs $D(\eta, \beta, \mathsf{seed})$ on the result. Note that $D' \in \mathcal{D}_s$ where $s \approx s' + s_{\mathtt{ext}} = s_{HILL}$. Thus we have the following $\forall X', Y'$, where $H_\infty(X'|Y') \geq \tilde{m}$:

$$\begin{aligned}
\delta^{D'}(X, Y), (X', Y')) &= \delta^D((\mathtt{ext}(X, U_d), Y, U_d), (\mathtt{ext}(X', U_d), Y', U_d)) \\
&\geq \delta^D((\mathtt{ext}(X, U_d), Y, U_d), (U_\kappa \times Y \times U_d)) \\
&- \delta^D((\mathtt{ext}(X', U_d), Y', U_\kappa) \times U_\kappa \times Y' \times U_d) \\
&> \epsilon_{\mathtt{ext}} + \epsilon_{\mathtt{HILL}} - \epsilon_{\mathtt{ext}} = \epsilon_{\mathtt{HILL}}
\end{aligned}$$

Thus D' is able to distinguish $X|Y$ from all $X'|Y'$ with sufficient entropy. This is a contradiction. □

When working with computational entropy, there is no reason to use an information-theoretic randomness extractor. A computational extractor [Kra10] is the adaption of a randomness extractor to the computational setting. Any information-theoretic randomness extractor is also a computational extractor; however, unlike information-theoretic extractors, computational extractors can expand their output via pseudorandom generators once a long-enough output is obtained. We adapt the definition of Krawczyk [Kra10] to the average case:

Definition 2.2.5. *A function* $\texttt{cext} : \mathcal{M} \to \{0,1\}^\kappa$ *a* $(\tilde{m}, \epsilon_{sec}, s_{sec})$-average-case computational extractor *if for all pairs of random variables* X, Y *(with* X *over* $\mathcal{M}$*) such that* $\tilde{H}_\infty(X|Y) \geq \tilde{m}$*, we have* $\delta^{\mathcal{D}_{s_{sec}}}((\texttt{cext}(X; U_d), U_d, Y), U_\kappa \times U_d \times Y) \leq \epsilon_{sec}$.

Computational extractors also work when given HILL entropy. The proof is the same as the proof of Theorem 2.2.4 and is omitted.

Extracting from unpredictability entropy Standard extractors cannot extract from distributions with unpredictability entropy. This requires a special type of extractor with a *reconstruction.* The best known example of a reconstructive extractor is the Goldreich-Levin hardcore bit [GL89].

Definition 2.2.6 (Reconstruction procedure)**.** *An* (κ, ϵ)*-reconstruction for a function ext* $: \mathcal{M} \to \{0,1\}^\kappa$ *is a pair of machines* $\mathsf{Compress}$ *and* Decomp*, where* $\mathsf{Compress} : \chi \to \{0,1\}^\kappa$ *is a randomized Turing machine, and* $\mathsf{Decomp}^{(\cdot)} : \{0,1\}^\kappa \to \mathcal{M}$ *is a randomized oracle Turing machine which runs in time polynomial in* $\log|\mathcal{M}|$*. Furthermore, for every* x *and* T*, if* $|\Pr[T(\texttt{ext}(x, U_d)) = 1] - \Pr[T(U_m \times U_d) = 1]| > \epsilon$*, then* $\Pr[\mathsf{Decomp}^T(\mathsf{Compress}^T(x)) = x] > 1/2$ *(the probability is over the random choices of* $\mathsf{Compress}$ *and* Decomp*).*

Lemma 2.2.7. *[HLR07, Lemma 6] Let* X, Y *be random variables with* $H^{unp}_{\epsilon,s}(X|Y) \geq \tilde{m}$*, and let* $\texttt{ext}$ *be an extractor with a* $(k - \log\frac{1}{\epsilon}, \epsilon)$*-reconstruction* $(\mathsf{Compress}, \mathsf{Decomp})$*. Then*

$$\delta^{\mathcal{D}_{s'}}((\texttt{ext}(X, U_d), Y, U_d), (U_\kappa \times Y \times U_d)) \leq 5\epsilon,$$

where $s' = s/(|\mathsf{Compress}| + |\mathsf{Decomp}|)$.

2.3 Coding Theory

Definition 2.3.1. *The* t-neighborhood *of* c*, denoted* $\mathsf{Neigh}_t(c)$*, is the set of all points distance* t *from* c*. That is* $\mathsf{Neigh}_t(c) = \{c' | \mathsf{dis}(c, c') = t\}$.

2.3.1 Shannon Codes

We use the definition of a Shannon code [SWBH49]:

Definition 2.3.2. *Let C be a set over space $\mathcal{M}$. We say that C is an (t, δ)-*Shannon code *if there exists a procedure* Rec *such that for all $t' \le t$ and for all $c \in C$, $\Pr[c' \leftarrow \mathsf{Neigh}_{t'}(c) \wedge \mathsf{Rec}(c') \ne c] \le \delta$. To distinguish it from the average-error Shannon code defined below, we will sometimes call it a* maximal-error *Shannon code.*

This is a slightly stronger formulation than usual, in that for every size $t' < t$ we require the code to correct t' random errors.[2] Shannon codes work for all codewords. We can also consider a formulation that works for an "average" codeword.

Definition 2.3.3. *Let C be a distribution over space $\mathcal{M}$. We say that C is an (t, ϵ)-*average error Shannon code *if there exists an efficient procedure* Rec *such that for all $t' \le t$ $\Pr_{c \leftarrow C}[\mathsf{Rec}(\mathsf{Neigh}_{t'}(c)) \ne c] \le \epsilon$.*

An average error Shannon code is one whose average probability of error is bounded by ϵ. See [CT06, Pages 192-194] for definitions of average and maximal error probability. An average-error Shannon code is convertible to a maximal-error Shannon code with a small loss. We use the following pruning argument from [CT06, Pages 202-204]:

Lemma 2.3.4. *Let C be a (t, ϵ)-average error Shannon code with recovery procedure* Rec *such that $\mathrm{H}_\infty(C) \ge m$. There is a set $\mathcal{C}'$ with $|\mathcal{C}'| \ge 2^{m-1}$ that is a $(t, 2\epsilon)$-(maximal error) Shannon code with recovery procedure* Rec*.*

Proof. Let C be the (t, ϵ)-average error Shannon code with recovery procedure Rec such that $\mathrm{H}_\infty(C) \ge m$. Then for all $t' \le t$

$$\sum_{c \in C} \Pr[C = c] \Pr[c' \leftarrow \mathsf{Neigh}(c, t') \wedge \mathsf{Rec}(c') \ne c] \le \epsilon.$$

For c denote by $\epsilon_c = \Pr[c' \leftarrow \mathsf{Neigh}(c, t') \wedge \mathsf{Rec}(c') \ne c]$. Then by Markov's inequality:

$$\Pr_{c \in C}[\epsilon_c \le 2 \mathop{\mathbb{E}}_{c \leftarrow C}[\epsilon_c]] = \Pr_{c \in C}[\epsilon_c \le 2\epsilon] \ge \frac{1}{2}$$

[2] In the standard formulation, the code must correct a random error of size up to t, which may not imply that it can correct a random error of a much smaller size t', because the volume of the ball of size t' may be negligible compared to the volume of the ball of size t. For codes that are monotone (if decoding succeeds on a set of errors, it succeeds on all subsets), these formulations are equivalent. However, we will work with an arbitrary recover functionality that is not necessarily monotone.

Let C' denote the of set all $c \in C$ where $\epsilon_c \leq 2\epsilon$. Note that $\Pr_{c \leftarrow C}[c \in C'] \geq 1/2$. Since $H_\infty(C) \geq m$, we know $|C'| \geq 2^{m-1}$ (otherwise $\Pr_{c \leftarrow C}[c \in C'] = \sum_{c \in C'} \Pr[C = c]$ would be less than $2^{m-1}\frac{1}{2^m} = 1/2$). This completes the proof of the Lemma 2.3.4. □

2.3.2 Hamming Codes

Definition 2.3.5 (Minimum distance)**.** *Let C be a set. The* minimum distance *of C is* $\min_{c,c' \in C} \mathsf{dis}(c, c')$.

Definition 2.3.6 (Error-correcting code)**.** *A set C is an* $(\mathcal{Z}^\gamma, |C|, d)$-error-correcting code *if its minimum distance is at least d. The elements $c \in C$ are known as codewords.*

If a message c is transmitted and at most $t = \lfloor \frac{d-1}{2} \rfloor$ of the symbols of c are modified, it is possible to uniquely recover the transmitted message c. A code is efficient if there exist polynomial time algorithms that sample $c \leftarrow C$ and $\mathsf{Dec}(c^*)$ that finds the unique $c \in C$ such that $\mathsf{dis}(c, c^*) \leq t$ if one exists. Many error-correcting codes have additional properties that facilitate encoding and decoding.

Linear error-correcting codes Let $\mathcal{Z} = \mathbb{F}_q$ for some field $\mathbb{F}_q$.

Definition 2.3.7. *A code C is a* $(\mathbb{F}_q^\gamma, \mathbb{F}_q^k, d)$-linear code *if is a k-dimensional linear subspace of $\mathbb{F}_q^\gamma$ with minimum distance d.*

Linear codes have two associated matrices G and H, known as the generating matrix and the parity check matrix respectively.

Definition 2.3.8 (Generating Matrix)**.** *For any $(\mathbb{F}_q^\gamma, \mathbb{F}_q^k, d)$-linear code C there exists a matrix $G \in \mathbb{F}_q^{\gamma \times k}$ where $span(G) = C$.*

Sampling a random $x \in \mathbb{F}_q^k$ and computing Gx is an efficient encoding function for C. Recall that the kernel, or **ker**, of a matrix is the set of all vectors that map to the 0 vector.

Definition 2.3.9 (Parity Check Matrix)**.** *For any $(\mathbb{F}_q^\gamma, \mathbb{F}_q^k, d)$-linear code C there exists a matrix $H \in \mathbb{F}_q^{(\gamma-k)\times\gamma}$ such that* $\mathbf{ker}(H) = C$.

Fix some $c \in C$, an important property of H is that for any c^* such that $\mathsf{dis}(c, c^*) \leq t$, the value Hc^* is unique. We call Hc^* the *syndrome* of c^*. Indeed, decoding usually consists three steps:

- Compute $s = Hc^*$.
- Map Hc^* to an error vector $e \in \mathbb{F}_q^\gamma$ where $\mathsf{Wgt}(e) \leq t$.
- Subtract e from c^* to obtain c.

2.3.3 Random Linear Codes

We will use the q-ary entropy function, denoted $H_q(x)$ and defined as $H_q(x) = x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x)$. Note that $H_2(x) = -x\log x - (1-x)\log(1-x)$. In the region $[0, \frac{1}{2}]$ for any value $q' \geq q$, $H_{q'}(x) \leq H_q(x)$. The following theorem is standard in coding theory:

Theorem 2.3.10. *[Gur10, Theorem 8] For prime $q, \rho \in [0, 1-1/q), 0 < \epsilon < 1-H_q(\rho)$ and sufficiently large γ, the following holds for $\mu = \lceil(1-H_q(\rho)-\epsilon)\gamma\rceil$. If $\mathbf{A} \in \mathbb{F}_q^{\gamma\times\mu}$ is drawn uniformly at random, then the linear code with $\mathbf{A}$ as a generator matrix has rate at least $(1-H_q(\rho)-\epsilon)$ and relative distance at least ρ with probability at least $1-e^{-\Omega(\gamma)}$.*

We use the following claim (techniques from Cooper [Coo00]):

Claim 2.3.11. *Let $q \geq 2$ be a prime. Let α, β be integers and let let $\mathbf{S} \overset{\$}{\leftarrow} \mathbb{F}_q^{\alpha\times(\alpha+\beta)}$ be uniformly generated. Then $\Pr[\mathtt{rank}(\mathbf{S}) = \alpha] > 1 - q^{-\beta}$.*

Proof. Let p_i be the probability that the i-th row is linearly dependent on the previous $i-1$ rows. By the union bound, the probability that α rows are linearly dependent is bounded by $\sum_{i=1}^{\alpha} p_i$. Since $i-1$ rows can span a space of size at most q^{i-1}, the probability p_i that a randomly chosen ith row is in that space is at most $q^{i-1}/q^{\alpha+\beta}$. So

$$\Pr[\mathtt{rank}(\mathbf{S}) < \alpha] = \sum_{i=1}^{\alpha} \frac{q^{i-1}}{q^{\alpha+\beta}} = \frac{q^\alpha - 1}{q-1}\frac{1}{q^{\alpha+\beta}} < q^{-\beta}.$$

□

2.4 Obfuscation

Our constructions will use obfuscation for two types of circuits: point functions and digital lockers. The family of point functions $\mathtt{I}_n = \{I_w\}_{w \in \{0,1\}^n}$ defined as follows:

$$I_w(x) : \begin{cases} 1 & x = w \\ 0 & \text{otherwise} \end{cases} .$$

and the class of digital lockers is $\mathtt{I}_n = \{I_{w,\mathsf{key}}\}_{w \in \{0,1\}^n, \mathsf{key} \in \{0,1\}^\kappa}$ defined as follows:

$$I_{w,\mathsf{key}}(x) : \begin{cases} \mathsf{key} & x = w \\ \perp & \text{otherwise} \end{cases} .$$

The required notion of obfuscation is virtual grey-box (VGB) introduced in [BC10]. This notion is weaker then the standard notion of virtual black-box ([BGI$^+$01]), as it allows the simulator to run in unbounded time while making at a polynomial number of oracle queries to the function.

We require that the obfuscation is composable and secure with respect to auxiliary input. Composable auxiliary-input VGB obfuscators for point functions and digital lockers are constructed in [BC10, Theorem 6.1] from the Strong Vector Decision Diffie-Hellman assumption, which is a generalization of the strong DDH assumption of [Can97] for tuples of points. They can also be constructed by assuming strong properties of cryptographic hash functions [Can97].

Definition 2.4.1 (composable obfuscation VGB obfuscation with auxiliary input [BC10])**.** *A PPT algorithm $\mathcal{O}$ is an ℓ-composable VGB obfuscator for $\mathtt{I}_n$ (resp. $\mathtt{I}_{n+\kappa}$) with auxiliary-input if the following conditions are met:*

1. Functionality: *for every n and $I \in \mathtt{I}_n$, $\mathcal{O}(I)$ is a circuit that computes the same function as I.*

2. Virtual grey-box: *For every PPT adversary A and polynomial p, there exists a*

(possibly inefficient) simulator S and a polynomial q such that for all sufficiently large n, any sequence of circuits $I^1, \ldots, I^\ell \in \mathtt{I}_n$, (where $\ell = \mathtt{poly}(n)$) and for all auxiliary inputs $z \in \{0,1\}^*$:

$$|\Pr_{A,\mathcal{O}}[A(z, \mathcal{O}(I^1), \ldots, \mathcal{O}(I^\ell)) = 1] - \Pr_{S}[S^{(I^1,\ldots,I^\ell)[q(n)]}(z, 1^{|I^1|+|I^\ell|}|) = 1]| < \frac{1}{p(n)},$$

where $(I^1, \ldots, I^\ell)[q(n)]$ is an oracle that answers at most $q(n)$ queries, and where every query of the form (i, x) is answered by $I^i(x)$.

For notational convenience, when we use point function obfuscation, we denote the oracle provided to the simulator as $I_w(\cdot,\cdot)$ where $w = w_1, ..., w_\gamma$ is the vector of obfuscated points. When we use digital lockers we denote the oracle provided to the simulator as $I_{w,\mathsf{key}}(\cdot,\cdot)$ where w is the vector of obfuscated points and key is the hidden value (we will hide the same value in each obfuscation).

Chapter 3

Key Derivation from Noisy Sources

In this chapter, we discuss prior approaches to key derivation from noisy sources. The term fuzzy extractors was introduced by Dodis et al. [DORS08]. There was substantial work on key derivation from noisy sources prior to the work of Dodis et al. In the discussion until now, we assumed a single user wished to generate key from a noisy source (with initial reading w) and be able to regenerate key from a nearby reading w'. In the single user setting, this task must be accomplished non-interactively, generating key and any necessary helper information p when w is observed. At a later time key can be regenerated from w' and p.

In this section, we explore alternative models for key derivation from noisy sources. We first consider a more general problem where users wish to derive keys from arbitrarily correlated random variables W and W' (Section 3.1). Bounded distance is one possible type of correlation between repeated readings, however not all types of correlation can be expressed as a metric. We then return to the bounded distance setting where two users hold w and w' respectively and engage in an interactive protocol to derive key protocol (Section 3.2). Finally, we consider the non-interactive setting where a single user wishes to derive key and p, and regenerate key from w' and p (Section 3.3). In all models, there are two fundamental tasks: information-reconciliation and privacy amplification [BBR88]. Information-reconciliation ensures that related w and w' are mapped to the same key (often this is done by first correcting w' to w). Privacy amplification ensures that key is uniformly distributed conditioned on the adversary's view.

3.1 Key Derivation from Correlated Random Variables

In the introduction, we considered two readings w and w' of a single physical source. We assumed the two readings were close according to some metric dis. The two readings w and w' can be modeled as a draw of correlated random variables where for all $(w, w') \leftarrow (W, W'), \mathsf{dis}(w, w') \leq t$.[1] Bounded distance is just one possible way that random variables can be correlated. Another line of work considers the problem of key derivation from arbitrarily correlated random variables [Mau93]. In this model, two parties Alice and Bob are able to draw from correlated random variables W and W' respectively. The goal of Alice and Bob is to agree on key by discussion over a public channel using W and W'. There is a (passive) eavesdropper Eve that can observes messages sent between Alice and Bob.

Maurer shows two results. First, the length of key is bounded by the mutual information between W and W' [Mau93, Theorem 1]. Second, if the two parties can make i.i.d. draws from W and W', then it is possible to achieve a secret key rate approaching the mutual information [Mau93, Theorem 2]. (The secret key rate is the average length of the secret key across independent executions of the protocol.) When authenticating using physical sources, for parties to make i.i.d. draws from W, W' they must have multiple instances of the physical source. The use of repeated draws from W and W' is crucial for a secret key rate that is proportional to an average-case information notion.[2] In this setting, mutual information between W and W' is a necessary and sufficient condition for key derivation.

Renner and Wolf [RW05] ask what length key is possible from a single draw of W and W' using non-interactive protocols. They first consider information-reconciliation and privacy amplification separately. They show that $\mathrm{H}_\infty(W)$ is a necessary and

[1] In the definition of fuzzy extractors (Definition 3.3.1), correctness is provided for all w' within distance t. That is, w' is assumed to be worst case and is not modeled as a random variable.

[2] If only one draw is allowed, even If $W = W'$, min-entropy of W is an upper bound on the achievable secret key length.

sufficient condition for privacy amplification.[3] Second, they show that the length of p must grow with the worst case number of possible outcomes for W conditioned on W'.[4] That is, the length of the public value

$$|p| \geq \max_{w' \in W} \log |\{w| \Pr[W = w|W' = w'] > 0\}|.$$

Furthermore, they show there exists a protocol with this length p using optimal encoding functions. Intuitively, the public information must describe which possible outcome for W actually occurred. This result describes the maximal length of p and does not argue how p effects security. It may be possible to construct a p that reduces the entropy of w by less than $\log |p|$. In Chapter 4, we will construct schemes with variable length p, providing information-reconciliation for distributions where the bounds of Renner and Wolf provide no security guarantees.

Lastly, the work of Renner and Wolf shows characterize when key derivation is possible from correlated random variables [RW05, Theorem 3]. We compare this characterization to fuzzy min-entropy in Chapter 4.

Most work on key derivation from correlated random variable setting is highly non-constructive. Restricting to correlation captured by a distance metric allows for ideas from coding theory and more efficient constructions.

3.2 The Interactive Setting

The work of Bennett, Brassard, and Robert [BBR88] introduced the problem of key agreement from nearby values. Two parties Alice and Bob which hold W and W' respectively. For any outcome of the two random variables $(w, w') \leftarrow (W, W')$,

[3] The results of Renner and Wolf use smooth notions of entropy. A random variable has smooth entropy if it is statistically close to a distribution with true entropy. We describe their results in the terminology of non smooth entropy.

[4] This result also uses a smooth notion of entropy. We describe the non smooth version for simplicity.

$\mathsf{dis}(w, w') \leq t$. Bounded distance between W and W' implies mutual information between W and W'. As in the correlated random variable case, information-reconciliation and privacy amplification are the two fundamental tasks. Min-entropy of W remains a necessary condition for privacy amplification. The goal of Alice and Bob is to create key using this information and a public channel. There is good reason to expect that interactive information-reconciliation can outperform non-interactive information-reconciliation.

We do not attempt to survey the interactive setting. We present the *Cascade* protocol to demonstrate the power of interactivity [BS94]. In this protocol, Alice and Bob hold $w \in \{0,1\}^\gamma$ and $w' \in \{0,1\}^\gamma$ respectively. We give a high level synopsis of the protocol:

1. Pass 1: Alice and Bob choose a length parameter α. The strings w and w' are split into length α blocks, $v_1, ..., v_{\gamma/\alpha}$.

2. Pass 1: Alice sends the parity of each block to Bob. For blocks where the parity differs, Alice and Bob use binary search to identify which bit(s) have errors (by repeatedly sending parity of subblocks). This identifies and eliminates any odd number of errors between blocks.

3. Passes 2 and up: Alice and Bob choose a length parameter α. They also choose a random function from $f : [1, ..., \gamma] \to [1, ..., \gamma/\alpha]$. Blocks are formed by all bits that have the same output f, that is $v_j = \{i | f(i) = j\}$. Alice sends parity of each block to Bob.

4. Passes 2 and up: Alice and Bob use binary search to correct odd number of errors in blocks whose parity does not match.

5. Passes 2 and up: Alice and Bob use binary search on the smallest blocks v_j to see if any errors exist in that block. If so, Alice and Bob find blocks in previous

passes that contained the position in error. These blocks must have an even number of errors. Use binary search to find errors in those blocks.

This protocol involves several rounds and Alice and Bob send multiple messages. It is unclear how to translate the ideas of the *Cascade* (and other similar) protocols to the non-interactive setting.

3.3 Fuzzy Extractors

In this section, we define fuzzy extractors and secure sketches. Definitions and lemmas are drawn from the work of Dodis et al. [DORS08, Sections 2.5–4.1] with modifications. First we allow for error, as discussed in [DORS08, Section 8]. Second, we consider an arbitrary family $\mathcal{W}$ of distributions instead of families containing all distributions of a given min-entropy. Let $\mathcal{M}$ be a metric space with distance function dis.

Definition 3.3.1. *An* $(\mathcal{M}, \mathcal{W}, \kappa, t, \epsilon)$-fuzzy extractor *with error* δ *is a pair of randomized procedures, "generate"* (Gen) *and "reproduce"* (Rep). Gen *on input* $w \in \mathcal{M}$ *outputs an extracted string* $\mathsf{key} \in \{0,1\}^\kappa$ *and a helper string* $p \in \{0,1\}^*$. Rep *takes* $w' \in \mathcal{M}$ *and* $p \in \{0,1\}^*$ *as inputs.* $(\mathsf{Gen}, \mathsf{Rep})$ *have the following properties:*

1. Correctness: *if* $\mathsf{dis}(w, w') \leq t$ *and* $(\mathsf{key}, p) \leftarrow \mathsf{Gen}(w)$*, then* $\Pr[\mathsf{Rep}(w', p) = \mathsf{key}] \geq 1 - \delta$ *(note that correctness holds for any* w' *with probability* $1 - \delta$ *over the coins on* Gen *and* Rep*, but* w' *cannot be a function of* p*).*

2. Security: *for any distribution* $W \in \mathcal{W}$*, if* $(\mathsf{Key}, P) \leftarrow \mathsf{Gen}(W)$*,*

$$\mathbf{SD}((\mathsf{Key}, P), (U_\kappa, P)) \leq \epsilon.$$

The standard construction is *sketch-and-extract:* the uniform key is extracted from w (using a randomness extractor [NZ93]) and error-tolerance is obtained by using a secure sketch [DORS08, Lemma 4.1]. Secure sketches produce a string ss that minimally decreases the entropy of w, while mapping nearby w' to w:

Definition 3.3.2. *An* $(\mathcal{M}, \mathcal{W}, \tilde{m}, t)$-secure sketch *with error* δ *is a pair of randomized procedures, "sketch"* (SS) *and "recover"* (Rec). SS *on input* $w \in \mathcal{M}$ *returns a bit string* $ss \in \{0,1\}^*$. Rec *takes an element* $w' \in \mathcal{M}$ *and* $ss \in \{0,1\}^*$. $(\mathsf{SS}, \mathsf{Rec})$ *have the following properties:*

1. Correctness*:* $\forall w, w' \in \mathcal{M}$ *if* $\mathsf{dis}(w, w') \leq t$ *then* $\Pr[\mathsf{Rec}(w', \mathsf{SS}(w)) = w] \geq 1 - \delta$ *(as before, correctness holds for any* w' *with probability* $1 - \delta$ *over the coins of* SS *and* Rec*, but not if* w' *is a function of* $\mathsf{SS}(w)$*).*

2. Security*: for any distribution* $W \in \mathcal{W}$, $\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) \geq \tilde{m}$.

The Case of Known Distribution If in the above definitions we take $\mathcal{W}$ to be a one-element set containing a single distribution W, then the fuzzy extractor/secure sketch is said to be constructed for a *known distribution*. In this case, we need to require correctness only for w that have nonzero probability[5].

We have no requirement that the algorithms are compact or efficient, and so the distribution can be fully known to them. Finding a natural model of specifying distributions that allows for efficient (yet generic) known distribution constructions of sketches and extractors is an interesting problem.

From Secure Sketches to Fuzzy Extractors A fuzzy extractor can be produced from a *secure sketch* and an *average case randomness extractor*:

Lemma 3.3.3. *Assume* $(\mathsf{SS}, \mathsf{Rec})$ *is an* $(\mathcal{M}, \mathcal{W}, \tilde{m}, t)$*-secure sketch with error* δ*, and let* $\mathtt{ext} : \mathcal{M} \to \{0,1\}^\kappa$ *be a* $(\tilde{m}, \epsilon)$*-average case extractor. Then the following* $(\mathsf{Gen}, \mathsf{Rep})$ *is an* $(\mathcal{M}, \mathcal{W}, \kappa, t, \epsilon)$*-fuzzy extractor with error* δ*:*

- $\mathsf{Gen}(w)$: *generate* $\mathsf{seed} \leftarrow \{0,1\}^d$*, set* $p = (\mathsf{SS}(w), \mathsf{seed})$, $\mathsf{key} = \mathtt{ext}(w; \mathsf{seed})$*, and output* (key, p).

- $\mathsf{Rep}(w', (ss, \mathsf{seed}))$: *recover* $w = \mathsf{Rec}(w', ss)$ *and output* $\mathsf{key} = \mathtt{ext}(w; \mathsf{seed})$.

[5] We can extend correctness to all of $\mathcal{M}$ by defining Gen/SS to output the point w as part of p/ss on zero-probability inputs, which will ensure that $\mathsf{Rep}/\mathsf{Rec}$ can always be correct; this does not affect security.

3.3.1 Previous approaches

As stated above, fuzzy extractors perform information-reconciliation and privacy amplification non-interactively. Secure sketches perform non-interactive information-reconciliation. In this section, we provide a brief overview of standard constructions. We focus on secure sketches as most known fuzzy extractors are formed using a secure sketch followed by a randomness extractor. Throughout this dissertation, we focus on the Hamming metric (many of our negative results hold for arbitrary metric spaces). Let $\mathcal{Z}$ be some alphabet and let w, w' be strings over $\mathcal{Z}^\gamma$, define $\mathsf{dis}(w, w') = \{i | w_i \neq w'_i\}$. We use Hamming codes as described in Section 2.3.2.

The code-offset construction Juels and Wattenberg [JW99] constructed an object called a fuzzy commitment scheme. The goal of a fuzzy commitment scheme is to allow a user to decommit if they know a close value (while retaining standard hiding properties).[6] This goal is very related to a secure sketch. Let h be some hash function and let C be an error correcting code with decoding function Dec.

```
Commit

  1. Input: $w$.
  2. Sample $c \leftarrow C$.
  3. Compute $com = h(c), c - w$.
```

```
Open

  1. Input: $(w', y, \delta)$
  2. Compute $c' = \delta + w'$.
  3. Decode $c^* = \mathsf{Dec}(c')$.
  4. If $h(c^*) = y$ output 1.
  5. Else output 0.
```

This construction was the precursor to the *code-offset sketch*. The code-offset sketch stores a public value $ss = c - w$ that is a "one-time pad" but with c that

[6] We do not describe cryptographic commitment here. It was defined by Brassard, Chaum, and Crépeau [BCC88].

contains redundancy.

Construction 3.3.4. *Let C be an efficient $(\mathbb{F}_q^\gamma, |C|, d)$-code. Define $\mathsf{SS}, \mathsf{Rec}$ as follows:*

```
Gen(w)
    1. Sample $c \leftarrow C$.
    2. Compute $ss = c - w$.
```

```
Rec(ss, w')
    1. Compute $c' = \delta + w'$.
    2. Decode $c^* = \mathsf{Dec}(c')$.
    3. Output $w^* = c^* - ss$.
```

Then $(\mathsf{SS}, \mathsf{Rec})$ is a $(\mathbb{F}_q^\gamma, m, m - (\gamma - \log|C|)\log q, \lfloor\frac{d-1}{2}\rfloor)$-secure sketch.

By using linear codes the construction can be derandomized. The syndrome of a linear code is a secure sketch:

Construction 3.3.5. *Let H be the parity check matrix of a $(\mathbb{F}_q^\gamma, \mathbb{F}_q^k, d)$-linear code. Then $\mathsf{SS}(w) = Hw, \mathsf{Rec}(ss, w') = w' + \mathsf{Dec}(ss)$ is a $(\mathbb{F}_q^\gamma, m, m - (\gamma - k)\log q, \lfloor\frac{d-1}{2}\rfloor)$-secure sketch.*

3.3.2 Computational Fuzzy Extractors

Definition 3.3.6 (Computational Fuzzy Extractor)**.** *Let $\mathcal{W}$ be a family of probability distributions over $\mathcal{M}$. A pair of randomized procedures "generate" (Gen) and "reproduce" (Rep) is a $(\mathcal{M}, \mathcal{W}, \kappa, t)$-*computational fuzzy extractor *that is (ϵ, s)-hard with error δ if Gen and Rep satisfy Definition 3.3.1 with the security property replaced with:*

- *For any distribution $W \in \mathcal{W}$, the string Key is pseudorandom conditioned on P, that is $\delta^{\mathcal{D}_s}((\mathsf{Key}, P), (U_\kappa, P)) \le \epsilon$.*

Any efficient fuzzy extractor is also a computational fuzzy extractor. We now define a weaker object that outputs computational entropy (instead of a pseudorandom key). We call this object a computational fuzzy conductor. It is the computational analogue of a fuzzy conductor (introduced by Kanukurthi and Reyzin [KR09]).

Definition 3.3.7. *A pair of randomized procedures "generate" (*Gen*) and "reproduce" (*Rep*) is an* $(\mathcal{M}, \mathcal{W}, \tilde{m}, t)$*-computational fuzzy conductor that is* (ϵ, s)*-hard with error* δ *if* Gen *and* Rep *satisfy Definition 3.3.6, except the last condition is replaced with the following weaker condition:*

- *for any distribution* $W \in \mathcal{W}$*, the string* r *has high HILL entropy conditioned on* P*. That is* $H^{\mathtt{HILL}}_{\epsilon,s}(R|P) \geq \tilde{m}$*.*

A computational fuzzy conductor (Definition 3.3.7) can be transformed to a computational fuzzy extractor (Definition 3.3.6) using a computational extractor (Definition 2.2.5).

Lemma 3.3.8. *Let* $(\mathsf{Gen}', \mathsf{Rep}')$ *be a* $(\mathcal{M}, \mathcal{W}, \tilde{m}, t)$*-computational fuzzy conductor that is* $(\epsilon_{cond}, s_{cond})$*-hard with error* δ *and outputs in* $\{0,1\}^\gamma$*. Let* $\mathtt{cext} : \{0,1\}^\gamma \to \{0,1\}^\kappa$ *be a* $(\tilde{m}, \epsilon_{ext}, s_{ext})$*-average case computational extractor. Define* $(\mathsf{Gen}, \mathsf{Rep})$ *as:*

- $\mathsf{Gen}(w; seed)$ *(where* $seed \in \{0,1\}^d$*): run* $(r', p') = \mathsf{Gen}'(w)$ *and output* $r = \mathtt{cext}(r'; seed)$*,* $p = (p', seed)$*.*
- $\mathsf{Rep}(w', (p', seed))$ *: run* $r' = \mathsf{Rep}'(w'; p')$ *and output* $r = \mathtt{cext}(r'; seed)$*.*

Then $(\mathsf{Gen}, \mathsf{Rep})$ *is a* $(\mathcal{M}, \mathcal{W}, \kappa, t)$*-computational fuzzy extractor that is* $(\epsilon_{cond}+\epsilon_{ext}, s)$*-hard with error* δ *where* $s = \min\{s_{cond} - |\mathtt{cext}| - d, s_{ext}\}$*.*

Proof. It suffices to show if there is some distinguisher D of size s where

$$\delta^D((\mathtt{cext}(W; U_d), U_d, P), (U_\kappa, U_d, P)) > \epsilon_{cond} + \epsilon_{ext}$$

then there is an distinguisher D' of size s_{cond} such that for all Y with $\tilde{\mathrm{H}}_\infty(Y|P') \geq \tilde{m}$,

$$\delta^{D'}((W, P'), (Y, P')) \geq \epsilon_{cond}.$$

Let D be such a distinguisher. That is,

$$\delta^D((\mathtt{cext}(W, U_d), U_d, P), (U_\kappa, U_d, P)) > \epsilon_{ext} + \epsilon_{cond}.$$

Define D' as follows. On input (y, p') sample $\mathsf{seed} \leftarrow U_d$, compute $\mathsf{key} \leftarrow \mathtt{cext}(y; seed)$, set $p = (p', \mathsf{seed})$ and output $D(\mathsf{key}, p)$. Note that D' is of size approximately

$s + |\mathtt{cext}| + d \leq s_{cond}$. Then we have the following:

$$\begin{aligned}\delta^{D'}((W, P'), (Y, P')) &= \delta^{D}((\mathtt{cext}(W, U_d), P), \mathtt{cext}(Y, U_d), P) \\ &\geq \delta^{D'}((\mathtt{cext}(W, U_d), P), (U_\kappa, P)) \\ &- \delta^{D'}((U_\kappa \times P), (\mathtt{cext}(Y, U_d), P)) \\ &> \epsilon_{cond} + \epsilon_{ext} - \epsilon_{ext} = \epsilon_{cond}.\end{aligned}$$

Where the last line follows by noting that D is of size at most s_{ext}. Thus D' distinguishes W from all Y with sufficient entropy. This is a contradiction. □

3.3.3 Reusable Computational Fuzzy Extractors

An additional desirable feature of fuzzy extractors is reusability [Boy04]. It is the ability to support multiple independent enrollments of the same value, allowing users to reuse the same biometric or physical unclonable function, for example, with multiple noncommunicating providers. More precisely, the algorithm Gen may be run multiple times on correlated readings $w_1, ..., w_q$ of a given source. Each time, Gen will produce a different pair of values $(\mathsf{key}_1, p_1), ..., (\mathsf{key}_q, p_q)$. Security for each extracted string key_i should hold even in the presence of all the helper strings $p_1, \ldots, p_q$ (the reproduction procedure Rep at the i-th provider still obtains only a single w_i' close to w_i and uses a single helper string p_i). Because the providers may not trust each other, a stronger security feature (which we satisfy) ensures that each key_i is secure even when all key_j for $j \neq i$ are also given to the adversary.

Constructions of reusable fuzzy extractors depend on the types of correlations allowed among $w_1, \ldots, w_q$. Boyen [Boy04] showed how to do so when each w_i is a shift of w_1 by a value that is oblivious to the value of w_1 itself (formally, w_i is a result of a transitive isometry applied to w_1). Boyen also showed that even for this weak class of correlations, any secure sketch must lose at least $\log |B_t|$ entropy [Boy04, Theorem 11].

We modify the definition of Boyen [Boy04, Definition 6] for the computational setting. We then compare the two definitions.

Definition 3.3.9 (Reusable Fuzzy Extractors)**.** *Let $\mathcal{W}$ be a family of distributions over $\mathcal{M}$. Let $(\mathsf{Gen}, \mathsf{Rep})$ be a $(\mathcal{M}, \mathcal{W}, \kappa, t)$-computational fuzzy extractor that is $(\epsilon_{sec}, s_{sec})$-hard with error δ. Fix some $W_1 \in \mathcal{W}$. Let $(f_2, .., f_q), D$ be a split adversary. Define the following game for all $j = 1, ..., q$:*

- ***Sampling*** *The challenger samples $w_1 \leftarrow W_1, u \leftarrow \{0,1\}^\kappa$.*
- ***Perturbation*** *For $i = 2, ..., q$: the challenger computes $(\mathsf{key}_i, p_i) \leftarrow \mathsf{Gen}(w_i)$. Set $w_{i+1} = f_i(w_1, p_1, ..., p_i)$.*
- ***Distinguishing*** *The advantage of D is*

$$\begin{aligned} Adv(D) \overset{def}{=} &\Pr[D(\mathsf{key}_1, ..., \mathsf{key}_{j-1}, \mathsf{key}_j, \mathsf{key}_{j+1}, ..., \mathsf{key}_q, p_1, ..., p_q) = 1] \\ &- \Pr[D(\mathsf{key}_1, ..., \mathsf{key}_{j-1}, u, \mathsf{key}_{j+1}, ..., \mathsf{key}_q, p_1, ..., p_q) = 1]. \end{aligned}$$

$(\mathsf{Gen}, \mathsf{Rep})$ *is $(q, \epsilon_{sec}, s_{sec}, f_2, ..., f_q)$-reusable if for all $D \in \mathcal{D}_{s_{sec}}$ the advantage is at most ϵ_{sec}.*

The definition is parameterized by $f_2, ..., f_q$. This adversary implicitly defines distributions $W_2, ..., W_q$ (which depend on W_1 and the public values $P_1, ...P_i$). Security seems hopeless if fuzzy extractor is not secure on each of these distributions on their own. This is the only requirement we make on these functions. We call these types of functions admissible:

Definition 3.3.10. *Let $(\mathsf{Gen}, \mathsf{Rep})$ be a $(\mathcal{M}, \mathcal{W}, \kappa, t)$-computational fuzzy extractor that is $(\epsilon_{sec}, s_{sec})$-hard with error δ. In the reusability game above, a set of functions $f_2, ..., f_q$ is* admissible *if for all $W_1 \in \mathcal{W}$ for all $w_1 \in W_1$ and $\forall p_1, ..., p_q$ that are the public outputs of Gen the distribution $W_{i,w_1,p_1,...,p_{i-1}} = f_i(w_1, p_1, ..., p_{i-1})$ is a member of $\mathcal{W}$.*

Comparison with the Definition of [Boy04] The goal of a reusable fuzzy extractor is to allow enrollment of a source across multiple services. A service i sees an reading of the source w_i. Boyen considers two versions of reusable fuzzy extractors, first where the adversary sees $p_1, ..., p_q$ (outsider security [Boy04, Definition 6]) and tries to learn about the values $w_1, ..., w_q$ or $\mathsf{key}_1, ..., \mathsf{key}_q$. Second, where the adversary controls some subset of the servers and can generate keys on arbitrary p_i' (insider security [Boy04, Definition 7]). This allows the adversary to learn a subset of keys r_i (by performing key generation on the valid p_i). This definition makes sense when servers are compromised (after enrollment) and act maliciously. In both definitions, the adversary creates a perturbation function f_i after seeing $p_1, ..., p_{i-1}$ (and generated keys for outsider security) and the challenger generates $w_i = f_i(w_1)$. The definition is parameterized by the class of allowed perturbation functions.

Boyen constructs a outsider reusable cryptographic fuzzy extractor for unbounded q when the perturbation family is a transitive isometric permutation groups. Boyen transforms this construction to insider security using random oracles.

Insider security strengthens outsider security in two ways. First, it allows the adversary to see some subset of keys, second it allows the adversary to perform key generation on arbitrary p_i. This mixes two properties of a fuzzy extractor: reusability and robustness [DKRS06]. Robust fuzzy extractors provide security against modified p. In this work, we show reusability when key_i are observed but do not handle the issue of robustness. That is, we assume keys may be exposed but servers keep honest state. Our definition lies between outsider and insider security.

We adapt the definition of Boyen to the computational setting (Definition 3.3.9). The definition of Boyen considers a single adversary. We split the adversary into two parts, one of which is information-theoretic and another that is computationally bounded. The functions $f_2, ..., f_q$ can be thought of as a single adversary that sees all prior state. However, to provide meaningful security in the computational setting, we cannot have communication between $f_2, ..., f_q$ and D.[7] Because these two adver-

[7] An alternative would be to have a single computationally bounded adversary. Construction 7.1.1 satisfies this alternative adaption as well.

saries do not communicate we strengthen the definition by allowing the perturbation functions, f_i, to see the original sample w_1. This was not allowed in the definition of Boyen as it would make security impossible.

Chapter 4

Measuring the Strength of a Noisy Distribution

In this chapter, we show how to improve fuzzy extractors by incorporating additional structure of the source distribution W. We begin by definition fuzzy min-entropy which describes a noisy distribution's suitability for key derivation (Section 4.1). We then show how to construct fuzzy extractors if a distribution is exactly known (Section 4.2). However, we show that distributional uncertainty comes at a cost. We show if a distribution W is only known to come from a family of distributions $\mathcal{W}$, then it may be impossible to construct a fuzzy extractor (Section 4.3).

4.1 Fuzzy Min-Entropy: a Necessary Condition

The value p allows everyone, including the adversary, to find the output of $\mathsf{Rep}(\cdot, p)$ on any input w'. Ideally, p should not provide any useful information beyond this ability, and the outputs of Rep on inputs that are too distant from w should provide no useful information, either. In this ideal scenario, the adversary is limited to trying to guess a w' that is t-close to w. Letting w' be the center of the maximum-weight ball in W would be optimal for the adversary. We therefore measure the quality of a source by (the negative logarithm of) this weight.

Definition 4.1.1. *The t-fuzzy min-entropy of a distribution W in a metric space $(\mathcal{M}, \mathsf{dis})$ is:*

$$\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = -\log\left(\max_{w'} \sum_{w \in W | \mathsf{dis}(w,w') \le t} \Pr[W = w]\right)$$

Fuzzy min-entropy is a necessary condition for security:

Proposition 4.1.2. *Let W be a distribution over $(\mathcal{M}, \mathsf{dis})$ and let $n = \log |\mathcal{M}|$. If $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \Theta(\log n)$ there is no $(\mathcal{M}, W, \kappa, t)$-fuzzy extractor with error $\delta = \mathtt{ngl}(n)$ for $\kappa = \omega(\log n)$.*

Proof. Let W be a distribution where $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \Theta(\log n)$. This means that there exists a point $w' \in \mathcal{M}$ such that $\Pr_{w \in W}[\mathsf{dis}(w, w') \leq t] \geq 1/\mathtt{poly}(n)$. Consider the following distinguisher D:

- Input key, p.
- If $\mathsf{Rep}(w', p) = \mathsf{key}$, output 1.
- Else output 0.

Clearly, $\Pr[D(\mathsf{Key}, P) = 1] \geq 1/\mathtt{poly}(n) - \delta$, while $\Pr[D(U_\kappa, P) = 1] = 1/2^{-\kappa}$. Thus, when $\kappa = \omega(\log n)$:

$$\delta^D((\mathsf{Key}, P), (U_\kappa, P)) \geq \frac{1}{\mathtt{poly}(n)} - \delta - \frac{1}{2^{-\kappa}} = 1/\mathtt{poly}(n).$$

Note that D only provides an input and looks at the output, thus it extends to an interactive protocol. Also, D is of size $\max|\mathcal{M}| + |\mathsf{Rep}|$ where $\max|\mathcal{M}|$ is the longest description of an item in the metric space. Thus, D is also a distinguisher in the computational setting. □

Generalizing to correlated random variables Instead of considering w, w' that have bounded distance, we treat W, W' as a pair of correlated random variables. Previous results in this setting are discussed in Section 3.1. Fuzzy min-entropy can be generalized to this setting. Fuzzy extractors consider the worst case w'. When considing correlated readings, it is natural to treat W' as a random variable:[1]

Definition 4.1.3. *Let (W, W') be a pair of correlated random variables. The correlated fuzzy min-entropy of W, W' is:*

$$\mathrm{H}^{\mathsf{corr}}(W, W') = -\log\left(\max_{w' \in \mathrm{supp}(W')} \sum_{w \in W \mid \Pr[W = w \mid W' = w'] > 0} \Pr[W = w]\right).$$

[1] Fuzzy extractors are defined to require high probability of correctness for all pairs w, w'. In the correlated setting, it may make sense to provide an average-case guarantee, where the probability of correctness is also over the draw of w, w'. Renner and Wolf use a smoothed notion of entropy that removes the δ fraction of the probability mass of $W = w|W' = w'$ with the most points to improve parameters under such a definition. In this work, we consider worst case correctness and use unsmoothed entropy.

In Definition 4.1.1, the sum is implicitly over $W = w | W' = w'$ since we assume any w' within distance t is possible. For now, we consider sufficiency of $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W)$ for key derivation from noisy sources (Definition 4.1.1). We then consider the implications of our results on the correlated reading setting (Definition 4.1.3).

4.2 Sufficiency of $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}$ When the Algorithms Know the Distribution

In this section, we consider fuzzy extractors that precisely know the input distribution W. We call this setting the *known-distribution* setting (see discussion after Definition 3.3). We show it is possible to build known-distribution secure sketches (and thus fuzzy extractors through Lemma 3.3.3) whenever $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \omega(\log n)$. We first consider flat distributions and show that hashing maintains fuzzy min-entropy and suffices to disambiguate points. We then turn to arbitrary distributions.

4.2.1 Flat Distributions

A distribution is flat if all points in its support have the same probability.

Definition 4.2.1. *A distribution W is* flat *if for all $w_0, w_1 \in \mathrm{supp}(W)$,* $\Pr[W = w_0] = \Pr[W = w_1]$.

Denote the largest number of points in a ball of radius t in the support of W as $\beta_t = \max_{w' \in \mathcal{M}} |\{w | w \in \mathrm{supp}(W) \wedge \mathsf{dis}(w, w') \leq t\}|$. For flat distributions, the weight of this maximum-probability ball (which determines $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W)$ by Definition 4.1.1) is proportional to the number of points in it. More precisely,

$$\begin{aligned}
\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) &= -\log\left(\max_{w' \in \mathcal{M}} |\{w | w \in \mathrm{supp}(W) \wedge \mathsf{dis}(w, w') \leq t\}| \cdot \Pr[W = w]\right) \\
&= -\log\left(\max_{w' \in \mathcal{M}} |\{w | w \in \mathrm{supp}(W) \wedge \mathsf{dis}(w, w') \leq t\}| \cdot 2^{-\mathrm{H}_\infty(W)}\right) \\
&= \mathrm{H}_\infty(W) - \log \beta_t. \qquad (4.1)
\end{aligned}$$

We use universal hashes to construct secure sketches for flat distributions. Skoric et al. constructed secure sketches from universal hashes to correct a polynomial number of error patterns [ŠTGP09].

Definition 4.2.2 ([CW79])**.** *Let $F : \mathcal{K} \times \mathcal{M} \to R$ be a function. We say that F is* universal *if for all distinct $x_1, x_2 \in \mathcal{M}$:*

$$\Pr_{K \leftarrow \mathcal{K}}[F(K, x_1) = F(K, x_2)] = \frac{1}{|R|} \ .$$

Construction 4.2.3. *Let $F : \mathcal{K} \times \mathcal{M} \to R$ be a universal hash function. Let W be a distribution. Define $\mathsf{SS}_W, \mathsf{Rec}_W$ as:*

```
$\mathsf{SS}_W$
    1. Input: $w$.
    2. Sample $K \leftarrow \mathcal{K}$.
    3. Set $p = F(K, w), K$.
```

```
$\mathsf{Rec}_W$
    1. Input: $(w', p = y, K)$
    2. Let $W^* = \{w \in \mathrm{supp}(W) | \mathsf{dis}(w, w') \leq t\}$.
    3. For $w^* \in W^*$, if $F(K, w^*) = y$,
       output $w^*$.
    4. Output $\perp$.
```

Lemma 4.2.4. *Let W be a flat distribution with $\mathrm{H}_\infty(W) \geq m$. Then Construction 4.2.3 is a $(\mathcal{M}, \{W\}, m - \log|R|, t)$-known distribution secure sketch with error $\delta \leq \frac{\beta_t - 1}{|R|}$.*

Proof. We first argue security. Fix some $W \in \mathcal{W}$. Since $\mathcal{K}$ and W are independent $\tilde{\mathrm{H}}_\infty(W|\mathcal{K}) = \mathrm{H}_\infty(W) = m$. Then by [DORS08, Lemma 2.2b], $\tilde{\mathrm{H}}_\infty(W|\mathcal{K}, F(\mathcal{K}, W)) \geq \mathrm{H}_\infty(W) - \log|F(\mathcal{W}, W)| \geq m - \log|R|$.

We now argue correctness. Fix some w, w'. Let W^* denote the set of elements in W within distance t of w'. The size of W^* is at most β_t. Since w, w' are independent of SS this set is independent of the choice of $\mathcal{K}$. The algorithm Rec will never output $\perp$ as the correct w will match the hash. The probability that another element w^*

collides is:

$$\begin{aligned}\Pr[\exists w^* \in W^* | w^* \neq w \wedge F(K, w^*) = F(K, w)] &\leq \sum_{w^* \in W^* | w^* \neq w} \Pr[F(K, w^*) = F(K, w)] \\ &= \sum_{w^* \in W^* | w^* \neq w} \frac{1}{|R|} \leq \frac{\beta_t - 1}{|R|}\end{aligned}$$

The inequality proceeds by union bound. The first equality proceeds by the universality of F and the second inequality proceeds by noting the number of wrong neighbors is bounded by $\beta_t - 1$. This completes the proof. □

Corollary 4.2.5. *Let* $n = \log |\mathcal{M}|$. *If* $|R| \geq |\beta_t| \cdot n^{\omega(1)}$ *then Construction 4.2.3 is correct with overwhelming probability. That is, setting* $\log |R| = \log \beta_t + \omega(\log n)$ *suffices.*

Construction 4.2.3 writes down enough information to disambiguate any ball of points. The remaining entropy for this construction is $\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) = \mathrm{H}_\infty(W) - \log \beta_t - \omega(\log n)$. For a flat distribution this is within a super-logarithmic factor of optimal (see Equation (4.1)). By choosing δ based on $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W)$ we build $(\mathsf{SS}, \mathsf{Rec})$ such that $\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) = \omega(\log n)$.

4.2.2 Arbitrary Distributions

The worst-case hashing approach does not work for arbitrary sources. The reason is that some balls may have many points but low total weight. For example, let W be a distribution consisting of the following balls. Denote by B_t^1 a ball with $2^{\mathrm{H}_\infty(W)}$ points with probability $\Pr[W \in B_t^1] = 2^{-\mathrm{H}_\infty(W)}$. Let $B_t^2, ..., B_t^{2^{-\mathrm{H}_\infty(W)}}$ be balls with one point each with probability $\Pr[W \in B_t^i] = 2^{-\mathrm{H}_\infty(W)}$. Then the hashing algorithm needs to write down $\mathrm{H}_\infty(W)$ bits to achieve correctness on B_t^1. However, with probability $1 - 2^{-\mathrm{H}_\infty(W)}$ the initial reading is outside of B_t^1, and the hash completely reveals the point.

Dealing with non-flat distributions requires a new strategy. Many solutions for manipulating high entropy distributions leverage a solution for flat distributions and use the fact that high entropy distributions are convex combinations of flat distributions. However, a distribution with high fuzzy min-entropy may be formed from component distributions with little or no fuzzy min-entropy. It is unclear how to leverage the convex combination property in this setting.

The main obstacle in the arbitrary setting is distinguishing between a setting where a ball has a few high probability points and a large number of low probability points. To overcome this problem, we write the probability of $w \in W$ in the sketch output. To ensure this information does not completely reveal w we write down $\lfloor \log \Pr[W = w] \rfloor$. We then use a universal hash whose output length is proportional to the number of close points of the same probability as w. This construction divides the distribution W into probability levels. Each level is nearly flat.

Construction 4.2.6. *Let $\mathcal{M}$ be a metric space and let $n = \log|\mathcal{M}|$. Let W be a distribution with $\mathrm{H}_\infty(W) = m$. Let $\ell \in \mathbb{Z}^+$ be a parameter. Let $L_i = (2^{-(i+1)}, 2^{-i}]$ for $i = m, ..., m + \ell$. Let $F_i : \mathcal{K}_i \times \mathcal{M} \to R_i$ be a parameterized family of universal hash functions. Define $\mathsf{SS}_W, \mathsf{Rec}_W$ as:*

```
$\mathsf{SS}_W$
  1. Input: $w$.
  2. If $\Pr[W = w] \leq 2^{-(m+\ell)}$. Set $p = 0, w$.
  3. Else
     (a) Find $i$ such that $\Pr[W = w] \in L_i$.
     (b) Sample $K \leftarrow \mathcal{K}_i$.
     (c) Set $ss = 1, i, F_i(K, w), K$.
```

```
$\mathsf{Rec}_W$
  1. Input: $(w', ss)$
  2. If $ss_0 = 1$, output $ss_{1,...,|y|}$.
  3. Else
     (a) Parse $(i, y, K) = ss_{1,...,|y|}$.
     (b) $W^* = \{w \in \mathrm{supp}(W) |$
         $\mathsf{dis}(w, w') \leq t,$
         $\Pr[W = w] \in L_i\}$.
     (c) For $w^* \in W^*$,
         if $F_i(K, w^*) = z$, output $w^*$.
     (d) Output $\perp$.
```

We extend our notation for the maximum likelihood ball to the leveled case. Define

$\beta_{t,i}$ as the maximum number of points in a ball in level i. That is,

$$\beta_{t,i} = \max_{w' \in \mathcal{M}} \left| \{ w | w \in \mathrm{supp}(W) \wedge \mathsf{dis}(w, w') \le t \wedge \Pr[W = w] \in L_i \} \right| .$$

Theorem 4.2.7. *Let W be a distribution over $\mathcal{M}$ where $n = \log \mathcal{M}$. Let $\delta > 0$ be an function of n. Let $F_i : \mathcal{K}_i \times \mathcal{M} \to R_i$ be a parameterized family of universal hash functions where $|R_i| = (\beta_{t,i} - 1)/\delta$. When $\ell = n$ Construction 4.2.6 is a $(\mathcal{M}, \{W\}, \tilde{m}, t)$-known distribution secure sketch with error δ for $\tilde{m} = \mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) - \log n - \log 1/\delta - 3$.*

Proof. Throughout the proof we assume that $\ell = n$ is the number of levels. The proof can be carried out for an arbitrary ℓ but it leads to a complicated theorem statement. **Correctness:** Fix some w, w'. If $\Pr[W = w] \le 2^{-(m+\ell)} = 2^{-(m+n)}$, then w is simply transmitted to Rec and correctness is clear. When $\Pr[W = w] > 2^{-(m+n)}$ let L_i^* be the level of $\Pr[W = w]$.

Let W^* denote the set of elements of W in L_i within distance t of w'. The size of W^* is at most $\beta_{t,i}$. The choice of w, w' is independent of SS, so this set is independent of $\mathcal{K}_i$ (it does effect the value of i but not the particular outcome from $\mathcal{K}_i$). The probability that another element w^* matches the hash is:

$$\begin{aligned}\Pr[\exists w^* \in W^* | w^* \neq w \wedge F(K, w^*) = F(K, w)] &\le \sum_{w^* \in W^* | w^* \neq w} \Pr[F(K, w^*) = F(K, w)] \\ &= \sum_{w^* \in W^* | w^* \neq w} \frac{1}{|R_i|} \le \frac{\beta_{t,i} - 1}{|R_i|} = \delta\end{aligned}$$

The inequality is by union bound. The first equality follows from the universality of F. The second inequality follows since the number of neighbors is bounded by $\beta_{t,i}$. **Ideal Adversary with access to Level Information:** To aid in the argument in security, we show the level information on its own is not too harmful.

The best strategy for an adversary that receives i as is to guess a point that has the most nearby weight in that level. The adversary chooses

$$w^* = \arg\max_{w' \in \mathcal{M}} \Pr_{w \in W | 2^{-(i+1)} < \Pr[W = w] \le 2^{-i} \wedge \mathsf{dis}(w, w^*)} [W = w].$$

The success of this adversary is at least $2^{-(i+1)}\beta_{t,i}$ as there at $\beta_{t,i}$ nearby points in that layer each with probability at least $2^{-(i+1)}$. There are n outcomes for i. The overall success of such an adversary is at most n better than an adversary without such input (by [DORS08, Lemma 2.2]). That is,

$$\begin{aligned}
&\mathbb{E}_{i|m\le i\le m+n}2^{-(i+1)}\beta_{t,i}\\
&\quad\le \mathop{\mathbb{E}}_{i|m\le i\le n+m}\left(\max_{w^*\in W}\sum_{w\in W|2^{-(i+1)}<\Pr[W=w]\le 2^{-i}\wedge\mathsf{dis}(w,w^*)\le t}\Pr[W=w]\right)\\
&\quad\le n\left(\max_{w^*\in W}\sum_{w\in W|\mathsf{dis}(w,w^*)\le t}\Pr[W=w]\right)\\
&\quad= n2^{-\mathrm{H}^{\mathbf{fuzz}}_{t,\infty}(W)} \qquad (4.2)
\end{aligned}$$

Security: We now argue security. First note that the total weight of points whose probability is less than $2^{-(n+m)}$ is at most 2^{-m} (there are at most 2^n points in the distribution). Let 1_{low} be the indicator random variable for $\Pr[W=w]\le 2^{-(n+m)}$. Then

$$\begin{aligned}
\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) = -\log\left(\Pr[1_{\text{low}}=1]*1+\Pr[1_{\text{low}}=0]2^{-\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)\wedge 1_{\text{low}}=0)}\right)\\
-\log\left(2^{-m}+(1-2^{-m})2^{-\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)\wedge 1_{\text{low}}=0)}\right)
\end{aligned}$$

For the remainder of the proof, we seek a bound on

$$2^{-\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)\wedge 1_{\text{low}}=0}=\max_{w\in W|2^{-(n+m)}<\Pr[W=w]}\Pr[W=w|\mathsf{SS}(W)].$$

We separate out this quantity into levels:

$$\begin{aligned}
&\max\nolimits_{w\in W|\Pr[W=w]>2^{-(m+n)}}\left(\Pr[W=w|\mathsf{SS}(W)]\right)\\
&\quad=\mathop{\mathbb{E}}_{i|m\le i\le m+n}\left(\max_{w\in W|\Pr[W=w]\in L_i}\Pr[W=w|\mathsf{SS}(W),i]\right)\\
&\quad=\mathop{\mathbb{E}}_{i|m\le i\le m+n}\left(\max_{w\in W|\Pr[W=w]\in L_i}\Pr[W=w]\cdot 2^{|\mathsf{SS}(W)|i|}\right)\\
&\quad\le\mathop{\mathbb{E}}_{i|m\le i\le m+n}\left(\max_{w\in W|\Pr[W=w]\in L_i}\Pr[W=w]\cdot 2^{H_0(\mathsf{SS}(W)|i)}\right)\\
&\quad\le\mathop{\mathbb{E}}_{i|m\le i\le m+n}\left(2^{-i}*\beta_{t,i}/\delta\right)\le\frac{\mathbb{E}_{i|m\le i\le m+n}\left(2^{-(i+1)}\cdot\beta_{t,i}\right)}{2\delta}\\
&\quad=\frac{n2^{-\mathrm{H}^{\mathbf{fuzz}}_{t,\infty}(W)}}{2\delta}.
\end{aligned}$$

Where the last line follows by Equation (4.2). Combining both cases we have:

$$\begin{aligned}
\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) &= -\log\left(2^{-m} + \frac{(1-2^{-m})(n)2^{-\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W)}}{2\delta}\right)\\
&\geq -\log\min\{2^{-m}, \frac{(1-2^{-m})n2^{-\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W)}}{2\delta}\}) - 1\\
&\geq \mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) - \log n + \log\delta - \log(1-2^{-m}) - 2\\
&\geq \mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) - \log n + \log\delta - 3
\end{aligned}$$

Where the third line follows from the second because $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) \leq \mathrm{H}_\infty(W) = m$. The last line follows from the fourth because if $m \geq 1$ then $\log(1-2^{-m}) \leq 1$ and if $m < 1$ the entire bound is vacuous as $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) < 1$. □

Corollary 4.2.8. *Let $\mathcal{M}$ be a metric space where $n = \log|\mathcal{M}|$. For any distribution W over $\mathcal{M}$ with $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \omega(\log n)$, there exists a $(\mathcal{M}, \{W\}, \tilde{m}, t)$-known distribution secure sketch with $\tilde{m} = \omega(\log n)$ and $\delta = \mathtt{ngl}(n)$. (Extendible to a fuzzy extractor using Lemma 3.3.3.)*

Connection to the characterization of [RW05] Renner and Wolf characterize when it is possible to derive keys from correlated random variables [RW05, Theorem 3]. They consider all possible (randomized) transforms T, T' of W, W' into a new pair of variables V, V'. They show that

$$|\mathsf{key}| \leq \sup_{(V,V')\leftarrow(T(W),T'(W))}\left(\mathrm{H}_\infty(V|T') - \log\max_{v'\in V'}|\{v|\Pr[V=v|T'\wedge V'=v']>0\}|\right).$$

Furthermore, they show that there is a transformation that achieves a key of nearly this length. The result is nonconstructive as there is no guidance on how to find the transforms T, T'. Since there is no known bound on the length of T, T' it is not clear how to search the transform space even with unlimited time.

Construction 4.2.6 can be used to derive keys from correlated random variables.

The main change is to define

$$W^* = \{w | \Pr[W = w | W' = w'] > 0 \wedge \Pr[W = w] \in L_i\}.$$

Our result shows if one is satisfied with obtaining a strong key when possible (our protocol has losses of $2 \log 1/\epsilon + \log n + \log 1/\delta$), then a protocol is possible (and explicitly constructible) in the original space.

4.3 Impossibility of Secure Sketches for a Family with $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}$

In the previous section, we showed the sufficiency of $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W)$ for known distribution algorithms. Unfortunately, it is unrealistic to assume that W is completely known. Traditionally, algorithms deal with this uncertainty by providing security for a family of distributions $\mathcal{W}$.

In this section, we show uncertainty of W comes at a real cost. The security game of a fuzzy extractor can be thought of as a three stage process: 1) the challenger specifies $(\mathsf{SS}, \mathsf{Rec})$, 2) the adversary sees $(\mathsf{SS}, \mathsf{Rec})$ and specifies $W \in \mathcal{W}$ 3) the adversary wins if $\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) < \tilde{m}$. We prove impossibility in a game that is harder for the adversary to win: 1) the challenger specifies $(\mathsf{SS}, \mathsf{Rec})$ 2) the adversary samples a random distribution from $W \leftarrow \mathcal{W}$ 3) the adversary wins if $\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) < \tilde{m}$.

Let V be the process of uniformly sampling $W \leftarrow \mathcal{W}$ and then sampling $w \leftarrow W$. Let the random variable Z indicate which W was sampled. The view of the challenger is V, while the view of the adversary is a distribution $V|Z$. Our results rule out security for an average member of $\mathcal{W}$. It may be possible to improve parameters by ruling out only a worst case W. In Chapter A, we show that providing security for a family $\mathcal{W}$ is equivalent to providing security for all distributions over that family. We now show a family of distributions $\mathcal{W}$ that does not admit a secure sketch. Our negative results in this chapter are specific to the Hamming metric.

Theorem 4.3.1. *Let n be a security parameter. There exists a family of distributions $\mathcal{W}$ over $\mathcal{Z}^\gamma$ such that for each element $W \in \mathcal{W}$, $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \omega(\log n)$, and yet for any $(\mathcal{M}, \mathcal{W}, \tilde{m}, t)$-secure sketch $(\mathsf{SS}, \mathsf{Rec})$ with error $\delta < 1/4$ and distance $\gamma > t \geq 4$, the remaining entropy $\tilde{m} < 2$.*

Furthermore, this is true on average. Let V be process of uniformly sampling $W \leftarrow \mathcal{W}$ and sampling $w \leftarrow W$, and let Z indicate which W is sampled. Then

$$\tilde{\mathrm{H}}_\infty(V | \mathsf{SS}(V), Z) < 2.$$

Proof. We prove the stronger average case statement. We first describe a family $\mathcal{W}$. Let $\mathbb{F}$ be some field of size $q = \omega(\mathtt{poly}(n))$. Let $\mathcal{W}$ be the set of all distributions of the form

$$W = \begin{pmatrix} \vec{1} \\ a_2 \\ \vdots \\ a_\gamma \end{pmatrix} W_1 + \begin{pmatrix} 0 \\ b_2 \\ \vdots \\ b_\gamma \end{pmatrix}$$

where W_1 is uniform and $W_i = a_i W_1 + b_i$ for $2 \leq i \leq \gamma$ and $a_i, b_i \in \mathbb{F}, a_i \neq 0$. This type of distribution is an affine line in space $\mathbb{F}^\gamma$. Define V as the process of uniformly choosing $W \leftarrow \mathcal{W}$ and then sampling from $w \leftarrow W$. The adversary sees $\mathsf{SS}(V)$ and Z. Z is the description of the line $Z = a_2, b_2, ..., a_\gamma, b_\gamma$. The algorithms $\mathsf{SS}, \mathsf{Rec}$ never see Z. Fix some $4 \leq t < \gamma$. We show the following:

- Proposition 4.3.2: for all $W \in \mathcal{W}$, $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \omega(\log n)$. That is, $\forall z, \mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(V | Z = z) = \omega(\log n)$.
- Proposition 4.3.3: the distribution V is uniform.
- Lemma 4.3.4: for any secure sketch on V, the support size of $V | \mathsf{SS}(V)$ decreases significantly. Here we show the minimum distance of $V | \mathsf{SS}(V)$ is at least t.
- Lemma 4.3.5: for most lines Z, the intersection of the support of $V | Z$ and $V | \mathsf{SS}(V)$ is small. That is, $\tilde{H}_0(V | \mathsf{SS}(V), Z) < 2$.

The proof of Theorem 4.3.1 uses Shannon codes (Definition 2.3.2). We now prove item in the above outline.

Proposition 4.3.2. *For each $W \in \mathcal{W}$, $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \omega(\log n)$.*

Proof. Consider some $W \in \mathcal{W}$. The value w_1 is uniform in a field of size $\omega(\mathtt{poly}(n))$, so $\mathrm{H}_\infty(W) = \omega(\log n)$. We now show that for any $w, w' \in W$, $\mathsf{dis}(w, w') = \gamma > t$.

This shows that $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \mathrm{H}_\infty(W)$. Fix some $w, w' \in W$. Clearly, $w_1 \neq w_1'$, for any i, $w_i = a_i w_1 + b_i$ and $w_i' = a_i w_1' + b_i$. Since $a_i \neq 0$, $a_i w_1 \neq a_i w_1'$ and thus $a_i w_1 + b_i \neq a_i w_1' + b_i$. That is, $\mathsf{dis}(w, w') = \gamma$. □

Proposition 4.3.3. *V is the uniform distribution over $\mathbb{F}^\gamma$.*

Proof. Consider some $w \in V$. Then w was drawn from some intermediate distribution W with coefficients $a_2, b_2, ..., a_\gamma, b_\gamma$. The value w_1 is uniformly random and w_i are uniformly random since $b_2, ..., b_\gamma$ are uniformly random. □

Lemma 4.3.4. *Fix some $\mathsf{SS}, \mathsf{Rec}$ algorithm with error $\delta < 1/4$, then $\tilde{H}_0(V|\mathsf{SS}(V)) \leq (\gamma - t + 1)\log q + 1$.*

Proof. We assume that Rec is deterministic in our analysis. Any randomness necessary for the Rec algorithm can be provided by SS. This is the same as considering Rep that outputs any coin it flips. Since w, w' are independent of p this does not effect correctness. Security is defined based on the output of Rec so outputting the coins of Rep does not effect security. By the definition of correctness for $(\mathsf{SS}, \mathsf{Rec})$,

$$\forall w, w', \Pr_{ss \leftarrow \mathsf{SS}(w)}[\mathsf{Rec}(w', ss) \neq w] < 1/4.$$

Fix some w. By Markov's inequality, there exists a set A_{ss} such that $\Pr[ss \in A_{ss}] \geq 1/2$ and $\forall ss \in A_{ss}$,

$$\{w' | \mathsf{dis}(w', w) \leq t \wedge \mathsf{Rec}(w', p) \neq w\} \leq 2\delta < 1/2.$$

Consider some $ss^* \in A_{ss}$. We now show that $H_0(V|\mathsf{SS}(V) = ss^*) \leq (\gamma - t + 1)\log q$. For the sketched value w, $\{w' | \mathsf{dis}(w, w') \leq t \wedge \mathsf{Rec}(w', p) \neq w] \leq 2\delta$.

For every value in $V|\mathsf{SS}(V) = ss^*$ this is also true. This makes the support of $V|\mathsf{SS}(V) = ss^*$ a $(t, 2\delta)$-Shannon code (see Definition 2.3.2). This implies that for all $w_1, w_2 \in V|\mathsf{SS}(V) = ss^*$, $\mathsf{dis}(w_1, w_2) \geq t$ (since $2\delta < 1/2$). That is $V|\mathsf{SS}(V) = ss^*$ is a set with minimum distance at least t.

By the Singleton bound, this implies that $H_0(V|\mathsf{SS}(V) = ss^*) \leq (\gamma - t + 1)q$. Averaging over $\mathsf{SS}(V) = ss^*$ one has that $\tilde{H}_0(V|P) \leq (\gamma - t + 1)\log q + 1$. □

Lemma 4.3.5. $\tilde{\mathrm{H}}_\infty(V|\mathsf{SS}(V), Z) < 2$.

Proof. Recall that Z consists of 2γ coefficients and there are $(q-1)^{\gamma-1}q^{\gamma-1}$ equally likely values for Z. As described above, the view of $\mathsf{SS}, \mathsf{Rec}$ is a uniform distribution

V. The only information seen by SS algorithm is in the point $V = v$. The length of this point is $\gamma \log q$. Conditioned on this information there are still many possible values for Z. That is,

$$\forall v, H_0(Z|V = v) = \log\left(\frac{(q-1)^{\gamma-1}q^{\gamma-1}}{q^\gamma}\right) = \log\left((q-1)^{\gamma-1}/q\right).$$

Consider two possible z_1, z_2 that are possible values of Z (having seen v). The distributions $V|Z = z_1$ and $V|Z = z_2$ intersect at one point (namely v).

We now show for any sketch algorithm there are few possible values of $V|Z$ in the support of $V|\mathsf{SS}(V)$. The distributions $V|Z = z_1$ and $V|Z = z_2$ for possible z_1, z_2 (having seen v) overlap only at the point v. This means for any $v^* \in V|\mathsf{SS}(V)$ (other than the true v) there is at most one z such that $v^* \in V|\mathsf{SS}(V), Z = z$.

The optimum strategy is to include these values uniformly from different Z values. We show this across different sketch values. Consider some fixed sketch value s and let $h_s = H_0(V|\mathsf{SS}(V) = s)$. Recall that

$$\tilde{H}_0(V|\mathsf{SS}(V)) = \log \mathop{\mathbb{E}}_{s\in\mathsf{SS}(V)} 2^{H_0(V|\mathsf{SS}(V)=s)} = \log \mathop{\mathbb{E}}_{s\in\mathsf{SS}(V)} 2^{h_s}$$

Conditioned on seeing the point V there are $(q-1)^{\gamma-1}/q$ possible values for Z with disjoint support outside of the sketched point. Consider these possible values for Z as containers to be filled with the $2^{h_{ss}}$ items (possible values of $V|\mathsf{SS}(V) = ss$). Each container receives automatically receives one free point (all the distributions share v). The average number of items in each container is maximized when the containers are filled equally. That is, the average number of items in each container is bounded by the number of items divided by the number of container. That is,

$$\begin{aligned}\tilde{H}_0(V|Z, \mathsf{SS}(V) = ss) &\le \log\left(\frac{\#\text{ items} + \#\text{ containers}}{\#\text{ containers}}\right)\\ &= \log\left(\frac{2^{h_{ss}}q}{(q-1)^{\gamma-1}} + 1\right)\end{aligned}$$

Then averaging over the possible values of s, we have the following as long as $t \ge$

4 (using Lemma 4.3.6, which appears below):

$$\begin{aligned}\tilde{H}_0(V|Z,\mathsf{SS}(V)) &= \log \mathop{\mathbb{E}}_{s\in\mathsf{SS}(V)} 2^{\tilde{H}_0(V|\mathsf{SS}(V)=ss,(Z|\mathsf{SS}(V)=ss))} \\ &= \log \mathop{\mathbb{E}}_{s\in\mathsf{SS}(V)} \left(\frac{2^{h_s}q}{(q-1)^{\gamma-1}}+1\right) \\ &\le \max\left\{\log\left(\frac{q}{(q-1)^{\gamma-1}}\mathop{\mathbb{E}}_{s\in\mathsf{SS}(V)} 2^{h_s}\right)+1,1\right\}.\end{aligned}$$

Where the inequality follows because $\log x + 1 \le \max\{1+\log x, 1\}$ for $x \ge 0$. The left operand to max is bounded by 2 (bounding the max by 2):

$$\begin{aligned}&\log\left(\frac{q}{(q-1)^{\gamma-1}}\mathop{\mathbb{E}}_{s\in\mathsf{SS}(V)} 2^{h_s}\right)+1 \\ &\quad= \log q - (\gamma-1)\log(q-1) + \log\left(\mathop{\mathbb{E}}_{s\in\mathsf{SS}(V)} 2^{h_s}\right)+1 \\ &\quad= \log q - (\gamma-1)\log(q-1) + \tilde{H}_0(V|\mathsf{SS}(V))+1 \\ &\quad\le \log q - (\gamma-1)\log(q-1) + (\gamma-t+1)\log q + 2 \\ &\quad\le (\gamma-t+2)\log q - (\gamma-1)\log(q-1) + 2 \\ &\quad< (\gamma-t+2)\log q - (\gamma-2)\log q + 2 \qquad \text{(by Lem 4.3.6)} \\ &\quad\le (4-t)\log q + 2 < 2\,.\end{aligned}$$

□

Lemma 4.3.6. *For any real numbers $\alpha \le \eta$ with $\eta \ge e+1$ (in particular, $\eta \ge 4$ suffices), the following holds: $\alpha\log(\eta-1) > (\alpha-1)\log\eta$.*

Proof. Because $\eta - 1$ is positive, and $1 + x < e^x$ for positive x,

$$1+\frac{1}{\eta-1} < e^{\frac{1}{\eta-1}}\,.$$

Therefore,

$$\left(1+\frac{1}{\eta-1}\right)^{\alpha-1} < e^{\frac{\alpha-1}{\eta-1}} \le e < \eta - 1$$

(since $\alpha \le \eta$). Multiplying both sides by $(\eta-1)^{\alpha-1}$, we obtain

$$\eta^{\alpha-1} < (\eta-1)^{\alpha}\,.$$

Taking the logarithm of both sides yields the statement of the lemma. □

□

Note: There is a tradeoff between the size of $\mathbb{F}$ and the error tolerance required for the counter example. By increasing t it is possible to show a counter example for a smaller $\mathbb{F}$.

4.4 Impossibility of Fuzzy Extractors for a Family with $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}$

In the previous section, we showed a family of distributions that does not admit a secure sketch. We provide a similar result for fuzzy extractors.

Theorem 4.4.1. *Let n be a security parameter. There exists a family of distributions $\mathcal{W}$ over $\{0,1\}^n$ satisfying the following conditions. For each element $W \in \mathcal{W}$, $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \omega(\log n)$. Let $\kappa \geq 2$ and $t = \omega(n^{1/2}\log n)$. Any $(\mathcal{M}, \mathcal{W}, \kappa, t, \epsilon)$-fuzzy extractor with error $\delta = 0$ has $\epsilon > 1/8 - \mathtt{ngl}(n)$.*

Furthermore, this is true on average. Let V be process of uniformly sampling $W \leftarrow \mathcal{W}$ and sampling $w \leftarrow W$ and let Z indicate which W is sampled. Let $(\mathsf{Key}, P) \leftarrow \mathsf{Gen}(V)$. Then,

$$\mathbf{SD}((\mathsf{Key}, P, Z), (U_\kappa, P, Z)) > 1/8 - \mathtt{ngl}(n)\,.$$

Proof Outline. We prove the stronger average case statement. Let $\nu = \omega(\log n)$ and $\nu = o(n^{1/2}/\log n)$. Let $t = 4\nu n^{1/2}$ and note that $n/\nu > t$.

Our counterexample uses a slightly different family of distributions $\mathcal{W}$ than the counterexample for secure sketches. We will work over a binary alphabet (we used a large alphabet in our counterexample for secure sketches). A property of the binary Hamming space is that a large fraction of any set of bounded size is the near "boundary" of that set. This will be crucial in our proof. We will embed the larger alphabet we used into the binary Hamming metric. Let $x_1, ..., x_\nu \in \{0,1\}^\nu$. Let $\mathbb{F}$ denote the field of size 2^ν. Let $a_2, ..., a_{n/\nu} \in \mathbb{F}$ such that $a_i \neq 0$ and let $b_2, ..., b_{n/\nu} \in \mathbb{F}$. Interpret $x_1, ..., x_\nu$ as a element $x \in \mathbb{F}$ and let

$$w = \begin{pmatrix} \vec{1} \\ a_2 \\ \vdots \\ a_{n/\nu} \end{pmatrix} x + \begin{pmatrix} 0 \\ b_2 \\ \vdots \\ b_{n/\nu} \end{pmatrix}.$$

The multiplication is in $\mathbb{F}$. Define a distribution W as the uniform distribution over values of x for a particular value of $a_2, ..., a_{n/\nu}$, $b_2, ..., b_{n/\nu}$. Let $\mathcal{W}$ be the set of all such W.

Define V as the process of uniformly choosing $W \leftarrow \mathcal{W}$ and then sampling from $w \leftarrow W$. The adversary sees $\mathsf{SS}(V)$ and Z, where Z is the description of the line $Z = a_2, ..., a_{n/\nu}, b_2, ..., b_{n/\nu}$.

We now present an outline of the proof (formal statements and proofs follow):

- Proposition 4.4.2: for all $W \in \mathcal{W}$, $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \omega(\log n)$. That is, $\forall z, \mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(V|Z = z) = \omega(\log n)$.
- Proposition 4.4.3: the distribution V is uniform.
- Lemma 4.4.4: In expectation across Z, a large subset of keys that are not possible. In more detail,
  - Half the keys have at most $2^{n-\kappa}$ pre images in the metric space (this is at most half the metric space). Denote this set as R_{sml}.
  - Consider some $\mathsf{key} \in R_{sml}$. Consider the set of $V_{\mathsf{key}} = \{w|\mathsf{Rep}(w, p) = \mathsf{key}\}$. All points in $V|\mathsf{SS}(V)$ are distance t from a boundary of V_{key} (the functionality of Rep guarantees that for the true w all nearby points map to the same key). We show that most of V_{key} is near a boundary. A result of Frankel and Füredi says that the boundary of a region is minimized by a ball containing the same number of points [FF81]. Hoeffding's inequality says that most of a ball lies near its boundary [Hoe63]. Together these two results imply that V_{key} is small.
  - As before, there are many possible values for z_1, z_2 for the side information Z (and these possible values are equally likely). Furthermore, the distributions $V|Z = z_1$ and $V|Z = z_2$ have disjoint support outside of v.
  - For most values of possible Z, the intersection between the viable pre images of $V|Z$ and V_{key} contains at most one point (the received point v). Checking if $V|Z \cap V_{\mathsf{key}}$ is nonempty is an effective distinguisher.

□

Proposition 4.4.2. *For each $W \in \mathcal{W}$, $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \omega(\log n)$.*

Proof. Consider some fixed $W \in \mathcal{W}$. The bits $w_{1,...,\nu}$ are uniform, so $\mathrm{H}_\infty(W) = \omega(\log n)$. Recall that $t = o(n/\nu)$. Fix some $w, w' \in W$. Denote by x, x' the values that produce w, w' respectively. Clearly, $x \neq x'$. Thus, for any i, $a_i x + b_i \neq a_i x' + b_i$. This implies that $w_{i\nu+1,....,(i+1)\nu} \neq w'_{i\nu+1,...,(i+1)\nu}$. That is, at least one of the bits in each block differs between w and w', and so $\mathsf{dis}(w, w') \geq n/\nu$. Since no two values in the support of W lie in the same ball of radius t, we have $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \mathrm{H}_\infty(W) = \omega(\log n)$. □

Proposition 4.4.3. *V is the uniform distribution over $\mathbb{F}^\gamma$.*

Proof. Consider some $w \in V$ over $\{0,1\}^n$. Then $w \leftarrow W$ with coefficients $a_2, ..., a_\gamma$ and $b_2, ..., b_\gamma$. The value $w_{1,...,\nu} = x$ is uniformly random and $w_{i\nu+1,...,(i+1)\nu}$ are uniformly random since $b_2, ..., b_\gamma$ are random. □

Lemma 4.4.4. *Fix some $(\mathsf{Gen}, \mathsf{Rep})$ algorithm with $\kappa \geq 2$. There exists an information theoretic distinguisher between (R, P, Z) and (U_κ, P, Z) with advantage $\epsilon = 1/8 - \mathtt{ngl}(n)$.*

Proof. As in the proof of Theorem 4.3.1, we assume that Rep is deterministic. Denote by $(\mathsf{Key}, P) \leftarrow \mathsf{Gen}(V)$. By Markov's inequality, there exists a set A_p such that $\Pr[p \in A_p] \geq 1/2$ and $\forall p \in A_p$,

$$(\mathsf{Key}|P = p, P = p) \approx_{2\epsilon} (U_\kappa, P = p).$$

Consider some $p^* \in A_p$. The distribution $\mathsf{Key}|P = p^*$ is the set of possible keys. The distribution $\mathsf{Key}|P = p^*$ induces a partition on the metric space. That is, for every $w \in \mathcal{M}$, there exists a unique value key such that $\mathsf{Rep}(w, p^*) = \mathsf{key}$. Denote this partition by $Q_{p^*,\mathsf{key}} = \{w | \mathsf{Rep}(w, p^*) = \mathsf{key}\}$.

There exists a set R_{sml} where $|R_{sml}| \geq 2^{\kappa-1}$ such that for all $\mathsf{key} \in R_{sml}$, $|Q_{p^*,r}| \leq \mathcal{M}/2^\kappa = 2^{n-\kappa}$. If not, then $\cup_{\mathsf{key}} |Q_{p^*,\mathsf{key}}| > |\mathcal{M}|$. For the remainder of the proof we restrict ourselves to elements in R_{sml}. Only points that are distance t from points outside of $Q_{p^*,r}$ are viable points in the metric space. These are the interior of $Q_{p^*,r}$:

$$\mathsf{Inter}(Q_{p^*,\mathsf{key}}) = \{w | \mathsf{Rep}(w, p^*) = \mathsf{key} \wedge \forall w', \mathsf{dis}(w, w') \leq t \wedge \mathsf{Rep}(w', p^*) = \mathsf{key}\},$$

We will use the term deficient ball[2]:

[2] In most statements of the isoperimetric inequality, this type of set is simply called a ball. We use the term deficient ball for emphasis.

Definition 4.4.5. *A set S is a η-deficient ball if there exists a point x such that $B_{\eta-1}(x) \subseteq S \subseteq B_\eta(x)$.*

Consider some $\mathsf{key}^* \in R_{sml}$. We now show that the interior of each Q_{p^*,key^*} is small:

Lemma 4.4.6. $|\mathsf{Inter}(Q_{p^*,\mathsf{key}^*})| \leq 2^{n-4\nu}$.

Proof. By the isoperimetric inequality on the Hamming space (we use a version due to [FF81, Theorem 1], the original result is due to Harper [Har66]), there exists a η-deficient ball S_{p^*,key^*} centered at 0 and a set D such that $|S_{p^*,\mathsf{key}^*}| = |\mathsf{Inter}(Q_{p^*,\mathsf{key}^*})|$, $|D| = |Q^{\mathsf{C}}_{p^*,\mathsf{key}^*}|$ and $\forall s \in S_{p^*,\mathsf{key}^*}, d \in D$, $\mathsf{dis}(s,d) \geq t$ (alternatively, the distance between the sets is t). Furthermore, note that $S_{p^*,\mathsf{key}^*} \cup D$ is a deficient ball (and its radius is $\eta + t$). We now find bound the size of S_{p^*,key^*}.

Recall that $|S_{p^*,\mathsf{key}^*} \cup D| = |Q_{p^*,\mathsf{key}^*}| \leq 2^{n-\kappa} \leq |\mathcal{M}|/2$. Since this set contains less than half the points in the metric space we know its radius at most $n/2$. This means that $|S_{p^*,\mathsf{key}^*}|$ is a deficient sphere of radius at most $n/2 - t$. Let X denote a uniform string on $\{0,1\}^n$. We use Hoeffding's inequality [Hoe63]:

$$\begin{aligned}|S_{p^*,\mathsf{key}^*}| \leq \{x | \mathsf{dis}(x,0) \leq n - t\} &= 2^n \Pr_{X \leftarrow \{0,1\}^n}[\mathsf{Wgt}(X) \leq (1/2 - t/n)n] \\ &\leq 2^n e^{-n((t/n)^2)} = 2^n e^{-4\nu} \leq 2^{n-4\nu}\end{aligned}$$

□

We have shown that $|\mathsf{Inter}(Q_{p^*,\mathsf{key}^*})| \leq 2^{n-4\nu}$. To complete the proof it suffices to show that for most values of the auxiliary information Z there are many parts Q_{p^*,key^*} that do not receive any points. Recall that Z consists of $2n/\nu$ coefficients and there are $(2^{n/\nu} - 1)^{\nu-1} 2^{n-\nu}$ equally likely values for Z. As described above, the view of $\mathsf{Gen}, \mathsf{Rep}$ is a uniform distribution V. We know show there are many possible values for $Z | P = p^*$. The only information about Z is contained in the point $V = v$. The length of this point is 2^n. Conditioned on this information there are still many possible values for Z. That is,

$$\begin{aligned}\forall v, H_0(Z|V=v) &= \log\left(\frac{(2^{n/\nu}-1)^{\nu-1}2^{n-\nu}}{2^n}\right)\\ &= \log\frac{(2^{n/\nu}-1)^{\nu-1}}{2^\nu}\\ &> \log\frac{(2^{n/\nu})^{\nu-2}}{2^\nu} \qquad \text{(by Lemma 4.3.6)}\\ &= \log\frac{2^{(n-2\nu))}}{2^\nu} = n-3\nu.\end{aligned}$$

Consider two possible z_1, z_2 that are possible values of Z. The distributions $V|Z=z_1$ and $V|Z=z_2$ intersect at one point (namely v).

This means that the Gen algorithm may include points for possible Z values into parts Q_{p^*,key^*} (other than v) and these values are disjoint. The optimum strategy is to include these values uniformly from different Z values. Consider the set of all preimages of R_{sml} denoted $Q_{sml} = \cup_{\mathsf{key}\in R_{sml}}\mathsf{Inter}(Q_{\mathsf{key},p^*})$. Note that $Q_{sml} \leq 2^{n-4\nu}|R_{sml}|$. We now show that the intersection between Q_{key,p^*} is small for most possible values z. As before each container (the values of z) receives one item for free (the point v).

$$\begin{aligned}\mathop{\mathbb{E}}_{z}|Q_{sml}\cap(V|P=p^*\wedge Z=z)| &\leq \left(\frac{\#\text{ items} + \#\text{ containers}}{\#\text{ containers}}\right)\\ &\leq \frac{2^{n-4\nu}|R_{sml}|}{2^{n-3\nu}}+1\\ &= \frac{|R_{sml}|}{2^\nu}+1\end{aligned}$$

In expectation across Z,

$$\frac{\frac{|R_{sml}|}{2^\nu}+1}{|R_{sml}|} \leq \frac{1}{2^\nu}+\frac{1}{|R_{sml}|}$$

fraction of R_{sml} receive any support. We now present a distinguisher D_{p^*} for a particular p^*:

1. On input x, z.
2. Compute $V|P=p^*\wedge Z=z$ and $Q_{p^*,x}$.
3. If $(Q_{p^*,x}\cap V|P=p^*\wedge Z=z)=\emptyset$ output $b=0$.
4. Else output $b=1$.

The distinguisher $D(x,p,z)$ is formed by calling $D_p(x,z)$ when $p \in A_p$ and outputting a random bit otherwise. The advantage of D is

$$
\begin{aligned}
&\Pr[D(\mathsf{Key},P,Z)=1] - \Pr[D(U,P,Z)=1]\\
&\quad= (\Pr[D(\mathsf{Key},P,Z)=1 | P \in A_p]\\
&\qquad - \Pr[D(U,P,Z)=1|P\in A_p])\Pr[P\in A_p]\\
&\quad\geq \sum_{p^*\in A_p} \Pr[P=p^*]\,(1-\Pr[D_{p^*}(U,Z)=1])\\
&\quad\geq \sum_{p^*\in A_p} \Pr[P=p^*](1-\\
&\qquad \Pr[D_{p^*}(U,Z)=1|U\in R_{sml}]\Pr[U\in R_{sml}])\\
&\qquad - \sum_{p^*\in A_p} \Pr[P=p^*]\Pr[U\notin R_{sml}]\\
&\quad\geq \sum_{p^*\in A_p} \Pr[P=p^*]\left(1-\left(\left(\frac{1}{|R_{sml}|}+\frac{1}{2^\nu}\right)\Pr[U\in R_{sml}]\right)\right)\\
&\qquad - \sum_{p^*\in A_p} (\Pr[P=p^*]\Pr[U\notin R_{sml}])\\
&\quad\geq \sum_{p^*\in A_p} \Pr[P=p^*]\left(1-\frac{1}{2^\nu}-\frac{1}{2}\Pr[U\in R_{sml}]-\Pr[U\notin R_{sml}]\right)\\
&\quad\geq \sum_{p^*\in A_p} \Pr[P=p^*]\left(1-\frac{1}{2^\nu}-\frac{1}{2}\Pr[U\in R_{sml}]-\Pr[U\notin R_{sml}]\right)\\
&\quad\geq \sum_{p^*\in A_p} \Pr[P=p^*]\left(1-\frac{1}{2^\nu}-1+\frac{1}{2}\Pr[U\in R_{sml}]\right)\\
&\quad\geq \sum_{p^*\in A_p} \Pr[P=p^*]\,(1/4-\mathtt{ngl}(n)) \geq \frac{1}{8}-\mathtt{ngl}(n).
\end{aligned}
$$

The sixth line follows since $R_{sml} \geq 2^{\kappa-1} \geq 2$. The eighth line follows because $\Pr[U \in R_{sml}] \geq 1/2$. The last inequality proceeds because $\Pr[P \in A_p] \geq 1/2$. This completes the proof of Lemma 4.4.4. □

Note: As stated in Section 1.2.2, using strong computational assumptions it is possible to avoid this result. Furthermore, for the specific family used in the secure sketch, we construct computational fuzzy extractors for this family of distributions when $\mathbb{F}$

is large enough under weaker assumptions in Construction 7.2.3. The construction is stated with imperfect correctness. A construction with perfect correctness is obtained by using a code that corrects t bidirectional errors instead of a code that corrects t unidirectional errors.

Comparison with Theorem 4.3.1 The parameters in this result are weaker than those in Theorem 4.3.1. This result requires: 1) higher error tolerance $t = \omega(n^{1/2} \log n)$ 2) the fuzzy extractor must have perfect correctness. The secure sketch counter example needs $t = 4$ and allows the Rec to be wrong almost $1/4$ of the time.

Chapter 5

Looking Beyond Sketch-then-Extract

As described in the introduction, secure sketches are subject to considerably stronger negative results than fuzzy extractors. In this chapter, we first show that computational versions of secure sketches are also subject to strong negative results. We then show how to construct a fuzzy extractor (without using a secure sketch), that supports sources with more errors than entropy.

5.1 Impossibility of Computational Secure Sketches

In this section, we consider whether it is possible in build a secure sketch that retains significantly more computational than information-theoretic entropy. We consider two different notions for computational entropy, and for both of them show that corresponding secure sketches are subject to upper bounds on the residual entropy. In particular, we show how to transform any sketch retaining HILL entropy into an information-theoretic sketch that retains a similar amount of min-entropy. Thus, it seems that relaxing security of sketches from information-theoretic to computational does not help.

In conjunction with previous results on upper bounds for the information-theoretic entropy of a secure sketch, this motivates us to build fuzzy extractors that do not incorporate secure sketches.

5.1.1 Bounds on Secure Sketches using HILL entropy

HILL entropy is a commonly used computational notion of entropy (Definition 2.2.1). Intuitively, HILL entropy is as good as average min-entropy for all computationally-bounded observers. Thus, redefining secure sketches using HILL entropy is a natural relaxation of the original information-theoretic definition; in particular, the sketch-and-extract construction in Lemma 3.3.3 would yield pseudorandom outputs if the secure sketch ensured high HILL entropy. We will consider secure sketches that retain relaxed HILL entropy (Definition 2.2.2).

Definition 5.1.1. *We say that* $(\mathsf{SS}, \mathsf{Rec})$ *is a* HILL-entropy $(\mathcal{M}, m, \tilde{m}, t)$ secure sketch *that is* (ϵ, s_{sec})*-hard with error* δ *if it satisfies Definition 3.3.2, with the security requirement replaced by* $H^{\texttt{HILL-rlx}}_{\epsilon,s_{sec}}(W|\mathsf{SS}(W)) \geq \tilde{m}$*.*

Unfortunately, we will show below that such a secure sketch implies an error correcting code with approximately $2^{\tilde{m}}$ points that can correct t random errors (see [DORS08, Lemma C.1] for a similar bound on information-theoretic secure sketches). For the Hamming metric, our result essentially matches the bound on information-theoretic secure sketches of [DORS08, Proposition 8.2]. In fact, we show that, for the Hamming metric, HILL-entropy secure sketches imply information-theoretic ones with similar parameters, and, therefore, the HILL relaxation gives no advantage.

The intuition for building error-correcting codes from HILL-entropy secure sketches is as follows. In order to have $H^{\texttt{HILL-rlx}}_{\epsilon,s_{sec}}(W|\mathsf{SS}(W)) \geq \tilde{m}$, there must be a distribution X, Y such that $\tilde{\mathrm{H}}_\infty(X|Y) \geq \tilde{m}$ and (X, Y) is computationally indistinguishable from $(W, \mathsf{SS}(W))$. Sample a sketch $s \leftarrow \mathsf{SS}(W)$. We know that SS followed by Rec likely succeeds on $W|s$ (i.e., $\mathsf{Rec}(w', s) = w$ with high probability for $w \leftarrow W|s$ and $w' \leftarrow B_t(w)$). Consider the following experiment: 1) sample $y \leftarrow Y$, 2) draw $x \leftarrow X|y$ and 3) $x' \leftarrow B_t(x)$. By indistinguishability, $\mathsf{Rec}(x', y) = x$ with high probability. This means we can construct a large set C from the support of $X|y$. C will be an error correcting code and Rec an efficient decoder. We can then use standard arguments to turn this code into an information theoretic sketch.

To make this intuition precise, we need an additional technical condition: sampling a random neighbor of a point is efficient.

Definition 5.1.2. *We say a metric space* $(\mathcal{M}, \mathsf{dis})$ *is* (s_{neigh}, t)-neighborhood samplable *if there exists a randomized circuit* Neigh *of size* s_{neigh} *that for all* $t' \leq t$, $\mathsf{Neigh}_{t'}(w)$ *outputs a random point at distance* t' *of* w.

We use the definition of a maximal and average error Shannon codes (Definitions 2.3.2 and 2.3.3). Recall, when we use the term Shannon code, we mean a maximal error Shannon code. A sketch that retains $\tilde{m}$-bits of relaxed HILL entropy implies a maximal error Shannon code with nearly $2^{\tilde{m}}$ points.

Theorem 5.1.3. *Let* $(\mathcal{M}, \mathsf{dis})$ *be a metric space that is* (s_{neigh}, t)*-neighborhood samplable. Let* $(\mathsf{SS}, \mathsf{Rec})$ *be an HILL-entropy* $(\mathcal{M}, m, \tilde{m}, t)$*-secure sketch that is* (ϵ, s_{sec})*-secure with error* δ*. Let* s_{rec} *denote the size of the circuit that computes* Rec*. If* $s_{sec} \geq (t(s_{neigh} + s_{rec}))$*, then there exists a value* s *and a set* $\mathcal{C}$ *with* $|\mathcal{C}| \geq 2^{\tilde{m}-2}$ *that is a* $(t, 4(\epsilon + t\delta))$*-Shannon code with recovery procedure* $\mathsf{Rec}(\cdot, s)$*.*

Proof. Let W be a distribution of min-entropy m. Let (X, Y) be a joint distribution such that $\tilde{\mathrm{H}}_\infty(X|Y) \geq \tilde{m}$ and

$$\delta^{\mathcal{D}_{s_{sec}}}((W, \mathsf{SS}(W)), (X, Y)) \leq \epsilon\,,$$

where $s_{sec} \geq t(s_{neigh} + s_{rec})$. One such (X, Y) must exist by the definition of relaxed HILL entropy. Define D as:

1. Input $w \in \mathcal{M}, z \in \{0,1\}^*, t$.
2. For all $1 \leq t' \leq t$:

   $w' \leftarrow \mathsf{Neigh}_{t'}(w)$.

   If $\mathsf{Rec}(w', z) \neq w$ output 0.
3. Output 1.

By correctness of the sketch $\Pr[D(W, \mathsf{SS}(W)) = 1] \geq 1 - t\delta$. Since

$$\delta^{D}((W, \mathsf{SS}(W)), (X, Y)) \leq \epsilon,$$

we know $\Pr[D(X, Y) = 1] \geq 1 - \epsilon - t\delta$. Let X_y denote the random variable $X|Y = y$. By Markov's inequality, there exists a set S_Y such that $\Pr[Y \in S_Y] \geq 1/2$ and for all $y \in S_Y$, $\Pr[D(X_y, y) = 1] \geq 1 - 2(\epsilon + t\delta)$.

Because $\tilde{\mathrm{H}}_\infty(X|Y) \geq \tilde{m}$, we know that $\mathbb{E}_{y\leftarrow Y} \max_x \Pr[X_y = x] \leq 2^{-\tilde{m}}$. Applying Markov's inequality to the random variable $\max_x \Pr[X_y = x]$, there exists a set S'_Y such that $\Pr[y \in S'_Y] > 1/2$, and for all $y \in S'_Y$, $\mathrm{H}_\infty(X_y) \geq \tilde{m}-1$ (we can use the strict version of Markov's inequality here, because the random variable $\max_x \Pr[X_y = x]$ is positive). Fix one value $y \in S_Y \cap S'_Y$ (which exists because the sum of probabilities of S_Y and S'_Y is greater than 1). Thus, for all such that $t', 1 \leq t' \leq t$,

$$\Pr_{x\leftarrow X_y}[x' \leftarrow \mathsf{Neigh}(x, t') \wedge \mathsf{Rec}(x', z) = x] \geq 1 - 2(\epsilon + t\delta).$$

Thus, X_y is a $(t, 2(\epsilon + t\delta))$-average error Shannon code with recovery $\mathsf{Rec}(\cdot, y)$ and $2^{\tilde{m}-1}$ points. The statement of the theorem follows by application of Lemma 2.3.4. □

For the Hamming metric, any Shannon code (as defined in Definition 2.3.2) can be converted into an information-theoretic secure sketch (as described in [DORS08, Section 8.2] and references therein). The idea is to use the code offset construction, and convert worst-case errors to random errors by randomizing the order of the symbols of w first, via a randomly chosen permutation π (which becomes part of the sketch and is applied to w' during Rec). The formal statement of this result can be expressed in the following Lemma (which is implicit in [DORS08, Section 8.2]).

Lemma 5.1.4. *For an alphabet $\mathcal{Z}$, let $\mathcal{C}$ be a (t, δ) Shannon code over $\mathcal{Z}^\gamma$. Then there exists a $(\mathcal{Z}^\gamma, m, m - (\gamma \log|\mathcal{Z}| - \log|\mathcal{C}|), t)$ secure sketch with error δ for the Hamming metric on $\mathcal{Z}^\gamma$.*

Combining Theorem 5.1.3 and Lemma 5.1.4 gives us the negative result for the Hamming metric: a HILL-entropy secure sketch (for the uniform distribution) implies an information-theoretic one with similar parameters:

Corollary 5.1.5. *Let $\mathcal{Z}$ be an alphabet. Let $(\mathsf{SS}', \mathsf{Rec}')$ be an (ϵ, s_{sec})-HILL-entropy $(\mathcal{Z}^\gamma, \gamma \log|\mathcal{Z}|, \tilde{m}, t)$-secure sketch with error δ for the Hamming metric over $\mathcal{Z}^\gamma$, with Rec' of circuit size s_{rec}. If $s_{sec} \geq t(s_{rec}+\gamma \log|\mathcal{Z}|)$, then there exists a $(\mathcal{Z}^\gamma, \gamma \log|\mathcal{Z}|, \tilde{m}-2, t)$ (information-theoretic) secure sketch with error $4(\epsilon + t\delta)$.*

Note In Corollary 5.1.5, the resulting $(\mathsf{SS}, \mathsf{Rec})$ is not guaranteed to be efficient because the proof of Theorem 5.1.3 is not constructive.

Corollary 5.1.5 extends to non-uniform distributions: if there exists a distribution whose HILL sketch retains $\tilde{m}$ bits of entropy, then for all distributions W, there is an information theoretic sketch that retains $\mathrm{H}_\infty(W) - (\gamma \log |\mathcal{Z}| - \tilde{m}) - 2$ bits of entropy.

5.1.2 Bounds on Secure Sketches using Unpredictability Entropy

In the previous section, we showed that any sketch that retained HILL entropy could be transformed into an information theoretic sketch. However, HILL entropy is a strong notion. In this section, we therefore ask whether it is useful to consider a sketch that satisfies a minimal requirement: the value of the input is computationally hard to guess given the sketch. We use the notion of relaxed unpredictability entropy (Definition 2.2.3) which captures the notion of "hard to guess."

Definition 5.1.6. $(\mathsf{SS}, \mathsf{Rec})$ *are an* unpredictability-entropy $(\mathcal{M}, m, \tilde{m}, t)$ secure sketch *that is* (ϵ, s_{sec})*-hard with error* δ *if it satisfies Definition 3.3.2, with the security requirement replaced by* $H^{\texttt{unp-rlx}}_{\epsilon, s_{sec}}(W|\mathsf{SS}(W)) \geq \tilde{m}$*.*

Combining such a secure sketch with a reconstructive extractor yields a computational fuzzy extractor (Lemma 2.2.7). The conditional unpredictability entropy $\tilde{m}$ must decrease as t increases. We will prove the result for any metric space that is both neighborhood samplable (Definition 5.1.2) and where picking a random point in the space is easy.

Definition 5.1.7. *A metric space space* $(\mathcal{M}, \mathsf{dis})$ *is* s_{sam}-efficiently-samplable *if there exists a randomized circuit* Sample *of size* s_{sam} *that outputs a uniformly random point in* $\mathcal{M}$*.*

Theorem 5.1.8. *Let* W *be a distribution over a metric space* $(\mathcal{M}, \mathsf{dis})$ *that is* s_{sam} *samplable and* (s_{neigh}, t) *neighborhood samplable. Furthermore, assume that the number of points within distance* t *in* $\mathcal{M}$ *is at least some fixed value* $B_t(\cdot)$*. Let* $(\mathsf{SS}, \mathsf{Rec})$ *be an unpredictability-entropy* $(\mathcal{M}, \mathrm{H}_\infty(W), \tilde{m}, t)$ *secure sketch that is* (ϵ, s_{sec})*-secure with error* δ*. If* $s_{sec} \geq \max\{t(|\mathsf{Rec}| + s_{neigh}), |\mathsf{Rec}| + s_{sam}\}$*, then* $\tilde{m} \leq \log |\mathcal{M}| - \log |B_t(\cdot)| + \log(1 - \epsilon - t\delta)$*.*

Proof. Let (X, Y) be two random variables such that $\delta^{\mathcal{D}_{s_{sec}}}((W, \mathsf{SS}(W)), (X, Y)) \leq \epsilon$. It suffices to show that $\exists \mathcal{I}$ of size s_{sec} such that $\Pr[\mathcal{I}(Y) = X] \geq |\mathcal{M}|(1-\epsilon-t\delta)/|B_t(\cdot)|$.

Let $B_t(x)$ denote the random variable representing a random neighbor of distance at most t from x (note that B_t may not be efficiently samplable, because we are assuming only that a neighbor a fixed distance is efficiently samplable). We begin by showing that Rec must recover points of X.

Claim 5.1.9.

$$\Pr[\mathsf{Rec}(B_t(X), Y) = X] =$$
$$\Pr[(x, y) \leftarrow (X, Y) \wedge x' \leftarrow B_t(x) \wedge \mathsf{Rec}(x', y) = x] \geq 1 - \epsilon - t\delta.$$

Proof. Suppose that $\Pr[\mathsf{Rec}(B_t(X), Y) = X] < 1 - \epsilon - t\delta$. We construct the following distinguisher $D \in \mathcal{D}_{s_{sec}}$ (the distinguisher design is slightly complicated by the fact that we don't know at which particular distance t' the recover procedure is most likely to fail, so we have to try all distances):

- Input $w \in \mathcal{M}, s \in \{0, 1\}^*$.
- For all $1 \leq t' \leq t$:

  $w' \leftarrow \mathsf{Neigh}(w, t')$.

  If $\mathsf{Rec}(w', z) \neq w$ output 0.
- Output 1.

First note that $|D| = t(|\mathsf{Rec}| + s_{neigh})$. Since $(\mathsf{SS}, \mathsf{Rec})$ has error δ we know that $\forall w, w' \in \mathcal{M}$ where $\mathsf{dis}(w, w') \leq t$

$$\Pr[s \leftarrow \mathsf{SS}(w) \wedge \mathsf{Rec}(w', s) = w] \geq 1 - \delta.$$

This implies that for all $1 \leq t' \leq t$, $\Pr[\mathsf{Rec}(\mathsf{Neigh}(W, t'), \mathsf{SS}(W)) = W)] \geq 1 - \delta$ and thus $\Pr[D(W, \mathsf{SS}(W)) = 1] \geq 1 - t\delta$. If $\Pr[\mathsf{Rec}(B_t(X), Y) = X] < 1 - \epsilon - t\delta$ there must exist at least one $1 \leq t' \leq t$ for which $\Pr[\mathsf{Rec}(\mathsf{Neigh}(X, t'), Y) = X] < 1 - \epsilon - t\delta$. Then

$$\begin{aligned}\Pr[D(W, \mathsf{SS}(W)) = 1] - \Pr[D(X, Y) = 1] &\geq (1 - t\delta) - \Pr[\mathsf{Rec}(\mathsf{Neigh}(X, t'), Y) = X] \\ &> (1 - t\delta) - (1 - t\delta - \epsilon) > \epsilon.\end{aligned}$$

This is a contradiction and the statement of the claim follows. □

We now return to the proof of Theorem 5.1.8. Now define $\mathcal{I}$ as follows:

- Input $y \in \{0,1\}^*$.
- Sample $x' \leftarrow \mathsf{Sample}$.
- Output $\mathsf{Rec}(x', y)$.

Note that $|\mathcal{I}| = |\mathsf{Rec}| + s_{sam}$. We now show that $\mathcal{I}$ predicts X:

$$\begin{aligned}
&\Pr[\mathcal{I}(Y) = X] = \\
&\quad = \sum_{x,y\in\mathcal{M}} \Pr[(X,Y) = (x,y)] \Pr[\mathcal{I}(y) = x] \\
&\quad = \sum_{x,y\in\mathcal{M}} \Pr[(X,Y) = (x,y)] \sum_{x'\in\mathcal{M}} \Pr[\mathsf{Sample} = x'] \Pr[\mathsf{Rec}(x', y) = x] \\
&\quad \geq \sum_{x,y\in\mathcal{M}} \Pr[(X,Y) = (x,y)] \sum_{x' | \mathsf{dis}(x',x)\leq t} \Pr[\mathsf{Sample} = x'] \Pr[\mathsf{Rec}(x', y) = x] \\
&\quad \geq \sum_{x,y\in\mathcal{M}} \Pr[(X,Y) = (x,y)] \sum_{x' | \mathsf{dis}(x',x)\leq t} \frac{|B_t(\cdot)| \Pr[B_t(x) = x'] \Pr[\mathsf{Rec}(x', y) = x]}{|\mathcal{M}|} \\
&\quad \geq \frac{|B_t(\cdot)|}{|\mathcal{M}|}(1 - \epsilon - t\delta)
\end{aligned}$$

(the last step follows by Claim 5.1.9). □

Note: If the input is uniform, the entropy loss is about $\log |B_t(\cdot)|$. An alternative interpretation of this theorem is that fuzzy min-entropy is at most $\gamma \log |\mathcal{Z}| - \log |B_t(\cdot)|$.

As mentioned at the beginning of Section 5.1, the same entropy loss can be achieved with information-theoretic secure sketches on the uniform distribution by using the randomized code-offset construction. One interpretation of this result is that unpredictability secure sketches are not useful on high entropy distributions.

5.1.3 Implications of negative results

In this chapter, we show that secure sketches that provide pseudoentropy suffer from similar lower bounds as information-theoretic secure sketches. In Chapter 4 we showed a family of distributions that cannot be sketched. This result extends to the computational setting. By Theorem 4.3.1 and the contrapositive of Corollary 5.1.5, no sketch can retain HILL entropy for the same family of distributions:

Corollary 5.1.10. *Let n be a security parameter and let $\mathcal{M} = |\mathbb{F}|^\gamma$. There exists a family of distributions $\mathcal{W}$ over $\mathcal{M}$ such that for each element $W \in \mathcal{W}$, $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) = \omega(\log n)$ and for any $(\mathcal{M}, \mathcal{W}, \tilde{m}, t)$-HILL secure sketch $(\mathsf{SS}, \mathsf{Rec})$ that is $(s_{sec}, \epsilon_{sec})$-hard and error δ. If $s_{sec} \geq t(|\mathsf{Rec}| + \gamma \log |\mathbb{F}|)$, $t \geq 4$, and $\epsilon_{sec} + t\delta < 1/16$, then $\tilde{m} < 4$.*

Secure sketches that provide computational unpredictability are implied the virtual-grey box obfuscation of all polynomial time circuits [BCKP14]. Our negative result bounds unpredictability away from the size of the metric space. Extraction from unpredictability entropy can be done using an extractor with a reconstruction property (Lemma 2.2.7); however, a virtual-grey box obfuscator for all polynomial size circuits can simply hide a randomly generated key, and therefore extraction is not necessary to obtain a fuzzy extractor.

Avoiding bounds Both of lower bounds arise because Rec must function as an error-correcting code for many points of any indistinguishable distribution. It may be possible to avoid these bounds if Rec outputs a fresh random variable[1]. Such an algorithm is called a computational fuzzy conductor (Definition 3.3.7). Some of our constructions will be computational fuzzy conductors while some will have pseudorandom outputs and thus be computational fuzzy extractors (Definition 3.3.6).

5.2 Supporting more errors than entropy

In the previous section, we showed that computational versions of secure sketches are subject to upper bounds on output entropy. We now show to build constructing fuzzy

[1] If some efficient algorithm can take the output of Rec and efficiently transform it back to the source W, the bounds of Corollary 5.1.5 and Theorem 5.1.8 both apply. This means that we need to consider constructions that are hard to invert (either information-theoretically or computationally).

extractors that do not contain a secure sketch (achieving properties that have eluded secure sketches). In particular, we show an information-theoretic fuzzy extractor that supports *more errors than entropy.* We describe this condition in Section 1.3.

The construction first condenses entropy from each block of the source and then applies a different fuzzy extractor to the condensed blocks. We'll denote the fuzzy extractor on the smaller alphabet as $(\mathsf{Gen}', \mathsf{Rep}')$. A condenser is like a randomness extractor but the output is allowed to be slightly entropy deficient. Condensers are known with smaller entropy loss than possible for randomness extractors (e.g. [DPW14]).

Definition 5.2.1. *A function* $\mathsf{cond} : \mathcal{Z} \rightarrow \mathcal{Y}$ *is a* $(m, \tilde{m}, \epsilon)$*-randomness condenser if whenever* $\mathrm{H}_\infty(W) \geq m$*, then there exists a distribution* Y *with* $\tilde{\mathrm{H}}_\infty(Y|\mathsf{seed}) \geq \tilde{m}$ *and*

$$(\mathsf{cond}(W, \mathsf{seed}), \mathsf{seed}) \approx_\epsilon (Y, \mathsf{seed}).$$

The main idea of the construction is that errors are "corrected" on the large alphabet (before condensing) while the entropy loss for the error correction is incurred on a smaller alphabet (after condensing).

Construction 5.2.2. *Let* $\mathcal{Z}$ *be an alphabet and let* $W = W_1, ..., W_\gamma$ *be a distribution over* $\mathcal{Z}^\gamma$*. We describe* $\mathsf{Gen}, \mathsf{Rep}$ *as follows:*

```
Gen
1. Input: $w = w_1, ..., w_\gamma$
2. For $j = 1, ..., \gamma$:
     (i) Sample $\mathsf{seed}_i \leftarrow \{0,1\}^d$.
     (ii) Set $v_i = \mathsf{cond}(w_i, \mathsf{seed}_i)$.
3. Set $(\mathsf{key}, p') \leftarrow \mathsf{Gen}'(v_1, ..., v_\gamma)$.
4. Set $p = (p', \mathsf{seed}_1, ..., \mathsf{seed}_\gamma)$.
5. Output $(\mathsf{key}, p)$.
```

```
Rep
1. Input: $(w', p = (p', \vec{\mathsf{seed}}))$
2. For $j = 1, ..., \gamma$:
     (i) Set $v_i' = \mathsf{cond}(w_i', \mathsf{seed}_i)$.
3. Output $\mathsf{key} = \mathsf{Rep}'(v', p')$.
```

For Construction 5.2.2 to be secure we need most blocks to contribute some entropy to the output. We call this notion a partial block source.

Definition 5.2.3. *A distribution $W = W_1, ..., W_\gamma$ is an (α, β)-partial block source if there exists a set of indices J where $|J| \geq \gamma - \beta$ such that the following holds:*

$$\forall j \in J, \forall w_1, ..., w_{j-1} \in W_1, ..., W_{j-1}, \mathrm{H}_\infty(W_j | W_1 = w_1, ..., W_{j-1} = w_{j-1}) \geq \alpha.$$

Definition 5.2.3 is a weakening of block sources (introduced by Chor and Goldreich [CG88]), as only some blocks are required to have entropy conditioned on the past. The choice of conditioning on the past is arbitrary: a more general sufficient condition is that there exists some ordering of indices where most items have entropy conditioned on all previous items in this ordering (for example, a "partial" reverse block source [Vad03]). This construction is secure and it supports distributions with more errors than entropy.

Lemma 5.2.4. *Let $\mathcal{W}$ be the family of $(\alpha = \Omega(1), \beta \leq \gamma(1 - \Theta(1)))$-partial block sources over $\mathcal{Z}^\gamma$ and let $\mathsf{cond} : \mathcal{Z} \times \{0,1\}^d \rightarrow \mathcal{Y}$ be a $(\alpha, \tilde{\alpha}, \epsilon_{cond})$-randomness conductor. Define $\mathcal{V}$ as the family of all distributions with min-entropy at least $\tilde{\alpha}(\gamma - \beta)$ and let $(\mathsf{Gen}', \mathsf{Rep}')$ be $(\mathcal{Y}^\gamma, \mathcal{V}, \kappa, t, \epsilon_{fext})$-fuzzy extractor with error δ.[2] Then $(\mathsf{Gen}, \mathsf{Rep})$ is a $(\mathcal{Z}^\gamma, \mathcal{W}, \kappa, t, \gamma\epsilon_{cond} + \epsilon_{fext})$-fuzzy extractor with error δ.*

Proof of Lemma 5.2.4. Let $W \in \mathcal{W}$. It suffices to argue correctness and security. We first argue correctness. When $w_i = w_i'$, then $\mathsf{cond}(w_i, \mathsf{seed}_i) = \mathsf{cond}(w_i', \mathsf{seed}_i)$ and thus $v_i = v_i'$. Thus, for all w, w' where $\mathsf{dis}(w, w') \leq t$, then $\mathsf{dis}(v, v') \leq t$. Then by correctness of $(\mathsf{Gen}', \mathsf{Rep}')$, $\Pr[(r,p) \leftarrow \mathsf{Gen}'(v) \wedge r' \leftarrow \mathsf{Rep}(v', p) \wedge r' = r] \geq 1 - \delta$.

We now argue security. Denote by seed the random variable consisting of all γ seeds and V the entire string of generated $V_1, ..., V_\gamma$. To show that

$$\mathsf{Key} | P, \mathsf{seed} \approx_{\gamma\epsilon_{cond} + \epsilon_{fext}} U | P, \mathsf{seed},$$

it suffices to show that $\tilde{\mathrm{H}}_\infty(V | seed)$ is $\gamma\epsilon_{cond}$ close to a distribution with average min-entropy $\tilde{\alpha}(\gamma - \beta)$. The lemma then follows by the security of $(\mathsf{Gen}', \mathsf{Rep}')$.

[2] We actually need $(\mathsf{Gen}', \mathsf{Rep}')$ to be an average case fuzzy extractor (see [DORS08, Definition 4] and the accompanying discussion). Most known constructions of fuzzy extractors are average-case fuzzy extractors. For simplicity we refer to $\mathsf{Gen}', \mathsf{Rep}'$ as simply a fuzzy extractor.

We now argue that there exists a distribution Y where $\tilde{\mathrm{H}}_\infty(Y|seed) \geq \tilde{\alpha}(\gamma - \beta)$ and $(V, seed_1, ..., seed_\gamma) \approx (Y, seed_1, .., seed_\gamma)$. First note since W is (α, β)-partial block distribution that there exists a set of indices J where $|J| \geq \gamma - \beta$ such that the following holds:

$$\forall j \in J, \forall w_1, ..., w_{j-1} \in W_1, ..., W_{j-1}, \mathrm{H}_\infty(W_j | W_1 = w_1, ..., W_{j-1} = w_{j-1}) \geq \alpha.$$

Then consider the first element of $j_1 \in J$, $\forall w_1, ..., w_{j_1-1} \in W_1, ..., W_{j_1-1}$,

$$\mathrm{H}_\infty(W_{j_1} | W_1 = w_1, ..., W_{j_1-1} = w_{j_1-1}) \geq \alpha.$$

Thus, there exists a distribution Y_{j_1} with $\tilde{\mathrm{H}}_\infty(Y_{j_1}|seed_{j_1}) \geq \tilde{\alpha}$ such that

$$(\mathtt{cond}(W_{j_1}, seed_{j_1}), seed_{j_1}, W_1, ..., W_{j_1-1}) \approx_{\epsilon_{cond}} (Y_{j_1}, seed_{j_1}, W_1, ..., W_{j_1-1})$$

and since $(seed_1, ..., seed_{j_1})$ are independent of these values

$$\begin{aligned}(\mathtt{cond}(W_{j_1}, seed_{j_1}), W_{j_1-1}, ..., W_1, seed_{j_1}, ..., seed_1)\\ \approx_{\epsilon_{cond}} (Y_{j_1}, W_{j_1-1}, ..., W_1, seed_{j_1}, , ..., seed_1)\end{aligned}$$

let $Z_{j_1} \overset{def}{=} (Y_{j_1}, \mathtt{cond}(W_{j_1-1}, seed_{j_1-1}), ..., \mathtt{cond}(W_1, seed_1))$ and note that

$$\tilde{\mathrm{H}}_\infty(Z_{j_1}|seed_1, ..., seed_{j_1}) \geq \alpha'.$$

Applying a deterministic function does not increase statistical distance and thus,

$$\begin{aligned}(\mathtt{cond}(W_{j_1}, seed_{j_1}), \mathtt{cond}(W_{j_1-1}, seed_{j_1-1}), ..., \mathtt{cond}(W_1, seed_1), seed_{j_1}, ..., seed_1)\\ \approx_{\gamma\epsilon_{cond}} (Z_{j_1}, seed_{j_1}, ..., seed_1)\end{aligned}$$

By a hybrid argument there exists a distribution Z with $\tilde{\mathrm{H}}_\infty(Z|seed) \geq \tilde{\alpha}(\gamma - \beta)$ where

$$(\mathtt{cond}(W_\gamma, seed_\gamma), ..., \mathtt{cond}(W_1, seed_1), seed_\gamma, ..., seed_1) \approx_{\gamma\epsilon_{cond}} (Z, seed_\gamma, ..., seed_1).$$

This completes the proof. □

More errors than entropy In this section we show that Construction 5.2.2 supports partial block sources with more errors than entropy. The structure of a partial block source implies that $\mathrm{H}_\infty(W) \geq \alpha(\gamma - \beta) = \Theta(\gamma)$. We assume that $\mathrm{H}_\infty(W) = \Theta(\gamma)$. The condenser of Dodis et al [DPW14] has a constant entropy loss, so $\alpha - \tilde{\alpha} = \Theta(1)$. This means that the input entropy to $(\mathsf{Gen}', \mathsf{Rep}')$ is $\Theta(\gamma)$. We assume that the new alphabet $\mathcal{Y}$ is of constant size. Standard fuzzy extractors on constant size alphabets correct a constant fraction of errors at a entropy loss of $\Theta(\gamma)$, yielding $\kappa = \Theta(\gamma)$. Thus, our construction is secure for distributions with more errors than entropy whenever $|\mathcal{Z}| = \omega(1)$. More formally:

$$\#\text{ Errors}-\text{Entropy} = \log|B_t| - \mathrm{H}_\infty(W) \geq t\log|\mathcal{Z}| - \Theta(\gamma) - = \Theta(\gamma)\log|\mathcal{Z}| - \Theta(\gamma) > 0$$

That is, there exists a super-constant alphabet size for which Construction 5.2.2 is secure with more errors than entropy.

Chapter 6

Moving to Computational Security

For the remainder of this work we switch to fuzzy extractors that provide computational security (Definition 3.3.6). We begin by showing a fuzzy extractor whose output key is as long as the starting entropy. This is impossible in the information-theoretic setting (unless all points of the distribution are far apart and $\mathrm{H}^{\mathsf{fuzz}}_{t,\infty}(W) = \mathrm{H}_\infty(W)$).

6.1 Computational Fuzzy Extractor based on LWE

In this section, a computational fuzzy extractor based on the learning with errors assumption. Security of our construction depends on the source W. We first consider a uniform source W; we consider other distributions in Section 6.2. Our construction uses the code-offset construction (described in Construction 3.3.4) instantiated with a random linear code over a finite field $\mathbb{F}_q$. Let Decode_t be an algorithm that decodes a random linear code with at most t errors (we will present such an algorithm later, in Section 6.1.2).

Construction 6.1.1. *Let n be a security parameter and let $\gamma \geq n$. Let q be a prime. Define* $\mathsf{Gen}, \mathsf{Rep}$ *as follows:*

Gen

1. *Input: $w \leftarrow W$ (where W is some distribution over $\mathbb{F}_q^\gamma$).*
2. *Sample $\mathbf{A} \in \mathbb{F}_q^{\gamma \times n}, \mathbf{x} \in \mathbb{F}_q^n$ uniformly.*
3. *Compute $p = (\mathbf{A}, \mathbf{A}\mathbf{x} + w)$,* $\mathsf{key} = \mathbf{x}_{1,\dots,n/2}$.
4. *Output* (key, p).

Rep

1. *Input: (w', p).*
2. *Parse p as $(\mathbf{A}, \mathbf{c})$; let $\mathbf{b} = \mathbf{c} - w'$.*
3. *Let $x = \mathsf{Decode}_t(\mathbf{A}, \mathbf{b})$*
4. *Output* $\mathsf{key} = x_{1,\dots,n/2}$.

Intuitively, security comes from the computational hardness of decoding random linear codes with a high number of errors (introduced by w). In fact, we know that decoding a random linear code is NP-hard [BMvT78]; however, this statement is not sufficient for our security goal, which is to show $\delta^{\mathcal{D}_{s_{sec}}}((X_{1,\ldots,n/2}, P), (U_{n/2\log q}, P)) \leq \epsilon$. Furthermore, this construction is only useful if Decode_t can be efficiently implemented.

The rest of this section is devoted to making these intuitive statements precise. We describe the LWE problem and the security of our construction in Section 6.1.1. We describe one possible polynomial-time Decode_t (which corrects more errors than is possible by exhaustive search) in Section 6.1.2. In Section 6.1.3, we describe parameter settings that allow us to extract as many bits as the input entropy, resulting in a lossless construction. In Section 6.1.4, we compare Construction 6.1.1 to using a sketch-and-extract approach (Lemma 3.3.3) instantiated with a computational extractor.

6.1.1 Security of Construction 6.1.1

The LWE problem was introduced by Regev [Reg05, Reg10] as a generalization of "learning parity with noise." For a complete description of the LWE problem and related lattices problems (which we do not define here) see [Reg05]. We now recall the decisional version of the problem.

Definition 6.1.2 (Decisional LWE). *Let n be a security parameter. Let $\gamma = \gamma(n) = \mathtt{poly}(n)$ be an integer and $q = q(n) = \mathtt{poly}(n)$ be a prime*[1]*. Let $\mathbf{A}$ be the uniform distribution over $\mathbb{F}_q^{\gamma\times n}$, X be the uniform distribution over $\mathbb{F}_q^n$ and χ be an arbitrary distribution on $\mathbb{F}_q^\gamma$. The decisional version of the LWE problem, denoted $\mathsf{dist\text{-}LWE}_{n,\gamma,q,\chi}$, is to distinguish the distribution $(\mathbf{A}, \mathbf{A}X + \chi)$ from the uniform distribution over $(\mathbb{F}_q^{\gamma\times n}, \mathbb{F}_q^\gamma)$.*

We say that $\mathsf{dist\text{-}LWE}_{n,\gamma,q,\chi}$ is (ϵ, s_{sec})-secure if no (probabilistic) distinguisher of size s_{sec} can distinguish the LWE instances from uniform except with probability ϵ. If

[1] Unlike in common formulations of LWE, where q can be any integer, we need q to be prime for decoding.

for any $s_{sec} = \texttt{poly}(n)$, there exists $\epsilon = \texttt{ngl}(n)$ such that $\mathsf{dist\text{-}LWE}_{n,\gamma,q,\chi}$ *is (ϵ, s_{sec})-secure, then we say it is* secure.

Regev [Reg05] and Peikert [Pei09] show that $\mathsf{dist\text{-}LWE}_{n,\gamma,q,\chi}$ is secure when the distribution χ of errors is Gaussian, as follows. Let $\bar{\Psi}_\rho$ be the discretized Gaussian distribution with variance $(\rho q)^2/2\pi$, where $\rho \in (0,1)$ with $\rho q > 2\sqrt{n}$. If GAPSVP and SIVP are hard to approximate (on lattices of dimension n) within polynomial factors for quantum algorithms, then $\mathsf{dist\text{-}LWE}_{n,\gamma,q,\bar{\Psi}_\rho^m}$ is secure. (A recent result of Brakerski et al. [BLP$^+$13] shows security of LWE based on hardness of approximating lattices problems for classical algorithms. We have not considered how this result can be integrated into our analysis.)

The above formulation of LWE requires the error term to come from the discretized Gaussian distribution, which makes it difficult to use it for constructing fuzzy extractors (because using w and w' to sample Gaussian distributions will increase the distance between the error terms and/or reduce their entropy). Recent work of Döttling and Müller-Quade [DMQ13] shows the security of LWE, under the same assumptions, when errors come from the uniform distribution over a small interval[2]. This allows us to directly encode w as the error term in an LWE problem by splitting it into γ blocks. The size of these blocks is dictated by the following result of Döttling and Müller-Quade:

Lemma 6.1.3. *[DMQ13, Corollary 1] Let n be a security parameter. Let $q = q(n) = \texttt{poly}(n)$ be a prime and $\gamma = \gamma(n) = \texttt{poly}(n)$ be an integer with $\gamma \geq 3n$. Let $\sigma \in (0,1)$ be an arbitrarily small constant and let $\rho = \rho(n) \in (0, 1/10)$ be such that $\rho q \geq 2n^{1/2+\sigma}\gamma$. If the approximate decision-version of the shortest vector problem (GAPSVP) and the shortest independent vectors problem (SIVP) are hard within a factor of $\tilde{O}(n^{1+\sigma}\gamma/\rho)$ for quantum algorithms in the worst case, then, for χ the uniform distribution over $[-\rho q, \rho q]^\gamma$,* $\mathsf{dist\text{-}LWE}_{n,\gamma,q,\chi}$ *is secure.*

[2] Micciancio and Peikert provide a similar formulation in [MP13]. The result Döttling and Müller-Quade provides better parameters for our setting.

To extract pseudorandom bits, we use a result of Akavia, Goldwasser, and Vaikuntanathan [AGV09] to show that X has simultaneously many hardcore bits. The result says that if $\mathsf{dist\text{-}LWE}_{(n-k,\gamma,q,\chi)}$ is secure then any k variables of X in a $\mathsf{dist\text{-}LWE}_{(n,\gamma,q,\chi)}$ instance are hardcore. We state their result for a general error distribution (noting that their proof does not depend on the error distribution):

Lemma 6.1.4. *[AGV09, Lemma 2] If* $\mathsf{dist\text{-}LWE}_{(n-k,\gamma,q,\chi)}$ *is* (ϵ, s_{sec}) *secure, then*

$$\delta^{\mathcal{D}_{s_{sec'}}}((X_{1,...,k}, \mathbf{A}, \mathbf{A}X+\chi), (U_{k\log q}, \mathbf{A}, \mathbf{A}X+\chi)) \leq \epsilon\,,$$

where $\mathbf{A}$ *denotes the uniform distribution over* $\mathbb{F}_q^{m\times n}$*,* X *denotes the uniform distribution over* $\mathbb{F}_q^n$*,* $X_{1,...,k}$ *denote the first* k *coordinates of* x*, and* $s'_{sec} \approx s_{sec} - n^3$*.*

The security of Construction 6.1.1 follows from Lemmas 6.1.3 and 6.1.4 when parameters are set appropriately (see Theorem 6.1.8), because we use the hardcore bits of X as our key.

6.1.2 Efficiency of Construction 6.1.1

Construction 6.1.1 is useful only if Decode_t can be efficiently implemented. We need a decoding algorithm for a random linear code with t errors that runs in polynomial time. We present a simple Decode_t that runs in polynomial time and can correct $\Theta(\log n)$ errors (note that this corresponds to a super-polynomial number of possible error patterns). This algorithm is a proof of concept, and neither the algorithm nor its analysis have been optimized for constants. An improved decoding algorithm can replace our algorithm, which will increase our correcting capability and improve Construction 6.1.1.

Construction 6.1.5. *We consider a setting of* (n, γ, q, χ) *where* $\gamma \geq 3n$*. We describe* Decode_t*:*

1. *Input* $\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{x} + w - w'$
2. *Randomly select rows without replacement* $i_1, ..., i_{2n} \leftarrow [1, \gamma]$*.*

3. *Restrict* $\mathbf{A}, \mathbf{b}$ *to rows* $i_1, ..., i_{2n}$*; denote these* $\mathbf{A}_{i_1,...,i_{2n}}, \mathbf{b}_{i_1,...,i_{2n}}$.

4. *Find* n *rows of* $\mathbf{A}_{i_1,...,i_{2n}}$ *that are linearly independent.*
   If no such rows exist, output $\perp$ *and stop.*

5. *Denote by* $\mathbf{A}', \mathbf{b}'$ *the restriction of* $\mathbf{A}_{i_1,...,i_{2n}}, \mathbf{b}_{i_1,...,i_{2n}}$ *(respectively) to these rows.*
   Compute $\mathbf{x}' = (\mathbf{A}')^{-1}\mathbf{b}'$.

6. *If* $\mathbf{b} - \mathbf{A}\mathbf{x}'$ *has more than* t *nonzero coordinates, go to step (2).*

7. *Output* $\mathbf{x}'$.

Each step is computable in time $O(n^3)$. For Decode_t to be efficient, we need t to be small enough so that with probability at least $\frac{1}{\mathtt{poly}(n)}$, none of the $2n$ rows selected in step 2 have errors (i.e., so that w and w' agree on those rows). If this happens, and $\mathbf{A}_{i_1,...,i_{2n}}$ has rank n (which is highly likely), then $\mathbf{x}' = \mathbf{x}$, and the algorithm terminates. However, we also need to ensure correctness: we need to make sure that if $\mathbf{x}' \neq \mathbf{x}$, we detect it in step 6. This detection will happen if $\mathbf{b} - \mathbf{A}\mathbf{x}' = \mathbf{A}(\mathbf{x} - \mathbf{x}') + (w - w')$ has more than t nonzero coordinates. It suffices to ensure that $\mathbf{A}(\mathbf{x} - \mathbf{x}')$ has at least $2t + 1$ nonzero coordinates (because at most t of those can be zeroed out by $w - w'$), which happens whenever the code generated by $\mathbf{A}$ has distance $2t + 1$.

Remark: Fuzzy extractor definitions make no guarantee about Rep behavior when the distance between w and w' is larger than t. Our Decode algorithm will never output an incorrect key (with high probability over the coins of Gen) but may not terminate. It may be preferable to output the wrong key or $\perp$ when $\mathsf{dis}(w, w') > t$.

Setting $t = \Theta(\frac{\gamma}{n} \log n)$ is sufficient to ensure efficiency when $\mathsf{dis}(w, w') \leq t$. Random linear codes have distance at least $\Theta(\frac{\gamma}{n} \log n)$ with probability $1 - e^{-\Omega(n)}$ (the exact statement is in Corollary 6.1.7), so this also ensures correctness. The formal statement is below:

Lemma 6.1.6 (Efficiency of Decode_t when $t \leq d(\gamma/n - 2)\log n$)**.** *Let* d *be a positive constant and assume that* $\mathsf{dis}(W, W') \leq t$ *where* $t \leq d(\frac{\gamma}{n} - 2)\log n$*. Then* Decode_t

runs in expected time $O(n^{4d+3})$ operations in $\mathbb{F}_q$ (this expectation is over the choice of random coins of Decode_t*, regardless of the input, as long as* $\mathsf{dis}(w, w') \leq t$*). It outputs X with probability $1 - e^{-\Omega(n)}$ (this probability is over the choice of the random matrix $\mathbf{A}$ and random choices made by* Decode_t*).*

Proof. We first show that our code has high distance with overwhelming probability. In our construction $\gamma = poly(n) \geq 2n$ and $\delta = O(\log n/n)$. This setting of parameters satisfies Theorem 2.3.10:

Lemma 6.1.7. *Let n be a parameter and let $\gamma = \texttt{poly}(n) \geq 2n$. Let q be a prime and $\tau = \Theta(\frac{\gamma}{n} \log n)$. For large enough values of n, when $\mathbf{A} \in \mathbb{F}_q^{\gamma \times n}$ is drawn uniformly, the code generated by $\mathbf{A}$ has distance at least τ with probability at least $1 - e^{-\Omega(\gamma)} \geq 1 - e^{-\Omega(n)}$.*

Proof. Let c be some constant. Let $\delta = \tau/\gamma = \frac{c \log n}{n}$. We show the corollary for the case when $\gamma = 2n$ (increasing the size of γ only increases the relative distance). It suffices to show that for sufficiently large n, there exists $\epsilon > 0$ where $1 - H_q(\frac{c \log n}{n}) - \epsilon = 1/2$ or equivalently that $H_q(\frac{c \log n}{m}) < 1/2$ as then setting $\epsilon = 1/2 - H_q(\frac{c \log n}{n})$ satisfies Theorem 2.3.10. For sufficiently large n:

- $\frac{c \log n}{n} < 1/2$, so we can work with the binary entropy function H_2.
- $\frac{c \log n}{n} < .1 < 1/2$ and thus $H_q(\frac{c \log n}{n}) < H_q(.1)$.

Putting these statements together, for large enough n, $H_q(\frac{c \log n}{n}) < H_q(.1) < H_2(.1) < 1/2$ as desired. This completes the proof. □

Note that Decode_t will stop if w and w' agree on all the rows selected in Step 2 (it may also stop for other reasons—namely, in step 4; but we do not use this fact to bound the expected running time). The probability of each selected row having an error is at most $\frac{t}{\gamma - i}$ where i is the number of rows already selected. That is,

$$\begin{aligned}
\Pr[i_1, ..., i_{2n} \text{ have no errors}] &\geq \prod_{i=0}^{2n-1}\left(1 - \frac{t}{\gamma - i}\right) \geq \prod_{i=0}^{2n-1}\left(1 - \frac{d\left(\frac{\gamma}{n} - 2\right)\log n}{\gamma - i}\right) \\
&\geq \prod_{i=0}^{2n-1}\left(1 - \frac{d\log n}{n}\left(\frac{\gamma - 2n}{\gamma - i}\right)\right) \geq \prod_{i=0}^{2n-1}\left(1 - \frac{d\log n}{n}\right) \\
&= \left(1 - \frac{d\log n}{n}\right)^{2n} = \left(\left(1 - \frac{d\log n}{n}\right)^{\frac{n}{d\log n}}\right)^{2d\log n} \\
&\geq \frac{1}{4^{2d\log n}} = \frac{1}{n^{4d}}.
\end{aligned}$$

(The second-to-last step holds as long as $n \geq 2d\log n$.) Because at each iteration, we select $2n$ rows independently at random, the expected number of iterations is at most n^{4d}; each iteration takes $O(n^3)$ operations in $\mathbb{F}_q$, which gives us the expected running time bound.

The probability that Decode_t outputs $\perp$ is bounded by

$$\begin{aligned}
\Pr[\mathsf{Decode}_t \rightarrow \perp] &\leq \sum_{j=1}^{\infty} \Pr[\mathsf{Decode}_t \rightarrow \perp \text{ in } j\text{-th iteration of step 4}] \\
&= \sum_{j=1}^{\infty} \Pr[\mathsf{Decode}_t \text{ reaches } j \text{ iterations} \wedge \mathtt{rank}(\mathbf{A}_{i_1,\ldots,i_{2n}}) < n] \\
&\leq \sum_{j=1}^{\infty} \Pr[i_1, ..., i_{2n} \text{ had errors } j-1 \text{ times} \wedge \mathtt{rank}(\mathbf{A}_{i_1,\ldots,i_{2n}}) < n] \\
&= \sum_{j=1}^{\infty} \Pr[i_1, ..., i_{2n} \text{ had errors } j-1 \text{ times}] \cdot \Pr[\mathtt{rank}(\mathbf{A}_{i_1,\ldots,i_{2n}}) < n] \\
&\leq \sum_{j=1}^{\infty} \left(1 - \frac{1}{n^{4d}}\right)^{j-1} \cdot q^{-n} \\
&= n^{4d} e^{-\Omega(n)} = e^{-\Omega(n)}.
\end{aligned}$$

The third line from the bottom follows from the fact that the locations of the errors are assumed to be independent of the sketch, and therefore independent of the matrix $\mathbf{A}$. The second line from the bottom follows from Claim 2.3.11 when $\beta = n$; note that, because we use the union bound and evaluate the probability separately for each value of j, we can treat $\mathbf{A}_{i_1,\ldots,i_{2n}}$ as a randomly chosen $2n \times n$ matrix, ignoring the fact that these matrices are correlated.

We claim that if the code generated by $\mathbf{A}$ has distance at least $2t+1$, then Decode_t will output $\perp$ or the correct $\mathbf{x}' = \mathbf{x}$. Indeed, suppose $\mathbf{x}' \neq \mathbf{x}$. Since $\mathbf{A}(\mathbf{x} - \mathbf{x}')$ has at least $2t+1$ nonzero coordinates by the minimum distance of the code generated by $\mathbf{A}$, and at most t of those can be zeroed out by the addition of $w - w'$, such an $\mathbf{x}'$ will not pass Step 6.

The probability that the code generated by $\mathbf{A}$ has distance lower than $2t+1$ is at most $e^{-\Omega(n)}$ (see Corollary 6.1.7), the probability of outputting $\perp$ is also $e^{-\Omega(n)}$ (computed above). This gives the correctness bound for Decode_t. □

6.1.3 Lossless Computational Fuzzy Extractor

We now state a setting of parameters that yields a lossless construction. The intuition is as follows. We are splitting our source into γ blocks each of size $\log \rho q$ (from Lemma 6.1.3) for a total input entropy of $\gamma \log \rho q$. Key is derived from hardcore bits of X: $X_{1,\ldots,k}$ and is of size $k \log q$ (from Lemma 6.1.4). Thus, to achieve a lossless construction we need $k \log q = \gamma \log \rho q$. In other words, in order to decode a meaningful number of errors, the vector w is of higher dimension than the vector X, but each coordinate of w is sampled using fewer bits than each coordinate of X. Thus, by increasing the size of q (while keeping ρq fixed) we can set $k \log q = \gamma \log \rho q$, yielding a $|\mathsf{key}| = |W|$. The formal statement is below.

Theorem 6.1.8. *Let n be a security parameter and let the number of errors $t = c \log n$ for some positive constant c. Let d be a positive constant (giving us a tradeoff between running time of Rep and $|w|$). Consider the Hamming metric over the alphabet $\mathcal{Z} = [-2^{b-1}, 2^{b-1}]$, where $b = \log 2(c/d+2)n^2 = O(\log n)$. Let W be uniform over $\mathcal{M} = \mathcal{Z}^\gamma$, where $\gamma = (c/d+2)n = O(n)$. If GAPSVP and SIVP are hard to approximate within polynomial factors using quantum algorithms, then there is a setting of $q = \mathtt{poly}(n)$ such that for any polynomial $s_{sec} = \mathtt{poly}(n)$ there exists $\epsilon = \mathtt{ngl}(n)$ such that the following holds: Construction 6.1.1 is a $(\mathcal{M}, W, \gamma \log |\mathcal{Z}|, t)$-computational fuzzy extractor that is (ϵ, s_{sec})-hard with error $\delta = e^{-\Omega(n)}$. The generate procedure Gen takes $O(n^2)$ operations over $\mathbb{F}_q$, and the reproduce procedure Rep takes expected time $O(n^{4d+3})$ operations over $\mathbb{F}_q$.*

Proof. Security follows by combining Lemmas 6.1.3 and 6.1.4; efficiency follows by Lemma 6.1.6. For a more detailed explanation of the various parameters and constraints see Section 6.4. □

6.1.4 Comparison with computational extractor-based constructions

An alternative approach to building a computational fuzzy extractor is to use a computational extractor (e.g., [Kra10, BDK+11, DSGKMk12]) in place of the information-theoretic extractor in the sketch-and-extract construction. We will call this approach *sketch-and-comp-extract*. (A simple example of a computational extractor is a pseudorandom generator applied to the output of an information-theoretic extractor; note that LWE-based pseudorandom generators exist [AIK06].)

This approach (specifically, its analysis via Lemma 3.3.3) works as long as the amount of entropy $\tilde{m}$ of w conditioned on the sketch s remains high enough to run a computational extractor. However, as discussed in the introduction, secure sketches are subject to strong lower bounds. For many practical sources, there are no known constructions of secure sketches.

In contrast, our approach does not require the entropy of w conditioned on $p = (\mathbf{A}, \mathbf{A}X + w)$ to be high enough for a computational extractor. Instead, we require that w is not computationally recoverable given p. This requirement is weaker—in particular, in our construction, w may have no information-theoretic entropy conditioned on p.

Unfortunately, the above construction comes with strong limitations on the error tolerance and supported sources. Herder et al. [HRvD+14] subsequently improved the error-tolerance for some sources using confidence information (discussion in the introduction). In Chapter 7, we show practical constructions based on point obfuscation. These constructions do not use *sketch-then-comp-extract*. In the next section, we show that Construction 6.1.1 is secure for more sources than just uniform W.

6.2 Moving to Nonuniform Sources

In this section, we show that Construction 6.1.1 is secure for a particular class of distributions called symbol-fixing. First we define a symbol fixing source (from [KZ07, Definition 2.3]):

Definition 6.2.1. *Let $W = (W_1, ..., W_{\gamma+\alpha})$ be a distribution where each W_i takes values over an alphabet $\mathcal{Z}$. We say that it is a $(\gamma + \alpha, \gamma, |\mathcal{Z}|)$* symbol fixing source *if for α indices $i_1, \ldots, i_\alpha$, the symbols W_{i_α} are fixed, and the remaining m symbols are chosen uniformly at random. Note that $H_\infty(W) = \gamma \log |\mathcal{Z}|$.*

Symbol-fixing sources are a very structured class of distributions. However, extending Construction 6.1.1 to such a class is not obvious. Although symbol-fixing sources are deterministically extractible [KZ07], we cannot first run a deterministic extractor before using Construction 6.1.1. This is because we need to preserve distance between w and w' and an extractor must not preserve distance between input points. We present an alternative approach, showing security of LWE directly with symbol-fixing sources.

The following theorem states the main technical result of this section, which is of potential interest outside our specific setting. The result is that $\mathsf{dist\text{-}LWE}$ with symbol-fixing sources is implied by standard $\mathsf{dist\text{-}LWE}$ (but for n and m reduced by the amount of fixed symbols).

Theorem 6.2.2. *Let n be a security parameter, γ, α be polynomial in n, and $q = \mathtt{poly}(n)$ be a prime and $\beta \in \mathbb{Z}^+$ be such that $q^{-\beta} = \mathtt{ngl}(n)$. Let U denote the uniform distribution over $\mathcal{Z}^\gamma$ for an alphabet $\mathcal{Z} \subset \mathbb{F}_q$, and let W denote an $(\gamma+\alpha, \gamma, |\mathcal{Z}|)$ symbol fixing source over $\mathcal{Z}^{\gamma+\alpha}$. If $\mathsf{dist\text{-}LWE}_{n,\gamma,q,U}$ is secure, then $\mathsf{dist\text{-}LWE}_{n+\alpha+\beta,\gamma+\alpha,q,W}$ is also secure.*

Theorem 6.2.2 also holds for an arbitrary error distribution (not just uniform error) in the following sense. Let χ' be an arbitrary error distribution. Define χ as the distribution where γ dimensions are sampled according to χ' and the remaining dimensions have some fixed error. Then, security of $\mathsf{dist\text{-}LWE}_{n,\gamma,q,\chi'}$ implies security of $\mathsf{dist\text{-}LWE}_{n+\alpha+\beta,\gamma+\alpha,q,\chi}$. We show this stronger version of the theorem in Section 6.3.

The intuition for this result is as follows. Providing a single sample with no error "fixes" at most a single variable. Thus, if there are significantly more variables than samples with no error, search LWE should still be hard. We are able to show a stronger result that dist-LWE is still hard. The nontrivial part of the reduction is using the additional $\alpha + \beta$ variables to "explain" a random value for the last α samples, without knowing the other variables. The β parameter is the slack needed to ensure that the "free" variables have influence on the last α samples. A similar theorem for the case of a single fixed dimension was shown in concurrent work by Brakerski et al. [BLP+13, Lemma 4.3]. The proof techniques of Brakerski et al. can be extended to our setting with multiple fixed dimensions, improving the parameters of Theorem 6.2.2 (specifically, removing the need for β).

Theorem 6.2.2 allows us to construct a lossless computational fuzzy extractor from block-fixing sources:

Theorem 6.2.3. *Let n be a security parameter and let $t = c \log n$ for some positive constant c. Let $d \le c$ be a positive constant and consider the Hamming metric over the alphabet $\mathcal{Z} = [-2^{b-1}, 2^{b-1}]$, where $b \approx \log 2(c/d + 2)n^2 = O(\log n)$. Let $\mathcal{M} = \mathcal{Z}^{\gamma+\alpha}$ where $\gamma = (c/d + 2)n = O(n)$ and $\alpha \le n/3$. Let $\mathcal{W}$ be the class of all $(\gamma + \alpha, \gamma, |\mathcal{Z}|)$-symbol fixing sources. If GAPSVP and SIVP are hard to approximate within polynomial factors using quantum algorithms, then there is a setting of $q = \mathtt{poly}(n)$ such that for any polynomial $s_{sec} = \mathtt{poly}(n)$ there exists $\epsilon = \mathtt{ngl}(n)$ such that the following holds: Construction 6.1.1 is a $(\mathcal{M}, \mathcal{W}, \gamma \log |\mathcal{Z}|, t)$-computational fuzzy extractor that is (ϵ, s_{sec})-hard with error $\delta = e^{-\Omega(n)}$. The generate procedure* Gen *takes $O(n^2)$ operations over $\mathbb{F}_q$, and the reproduce procedure* Rep *takes expected time $O(n^{4d+3} \log n)$ operations over $\mathbb{F}_q$.*

Proof. Security follows by Lemmas 6.1.3 and 6.1.4 and Theorem 6.2.2 . Efficiency follows by Lemma 6.1.6. For a more detailed explanation of parameters see Section 6.4.1.

□

6.3 Proof of Theorem 6.2.2

Proof. We assume that all of the fixed blocks are located at the end and their fixed value is 0. If the blocks are fixed to some other value, the reduction is essentially the same. In the reduction, the distinguisher is allowed to depend on the source and can know the positions of the fixed blocks and their values. For a matrix $\mathbf{A}$ we will denote the i-th row by $\mathbf{a}_i$. For a set T of column indices, we denote by $\mathbf{A}_T$ the restriction of the matrix $\mathbf{A}$ to the columns contained in T. Similarly, for a vector $\mathbf{x}$ we denote by $\mathbf{x}_T$ the restriction of $\mathbf{x}$ to the variables contained in T. We use similar notation for the complement of T, denoted T^c. For a matrix or vector we use T to denote the transpose. We use i as a index into matrix rows and the error vector and j as an index into columns and the solution vector.

Let n be a security parameter, $\gamma, q, \alpha = \mathtt{poly}(n)$. Let β be such that $q^{-\beta} = \mathtt{ngl}(n)$. All operations are computed modulo q, and we omit " mod q" notation. Let χ' be some error distribution over $\mathbb{F}_q^m$ and let χ over $\mathbb{F}_q^{m+n}$ be defined by sampling χ' to obtain values on dimensions $1, ..., m$ and then appending α 0s.

Let D be a distinguisher that breaks $\mathsf{dist\text{-}LWE}_{(\gamma+\alpha),(n+\alpha+\beta),q,\chi}$ with advantage $\epsilon > 1/\mathtt{poly}(n)$. Let $\mathbf{A}$ denote the uniform distribution over $\mathbb{F}_q^{(\gamma+\alpha)\times(n+\alpha+\beta)}$, X denote the uniform distribution over $\mathbb{F}_q^{(n+\alpha+\beta)}$, and U denote the uniform distribution over $\mathbb{F}_q^{\gamma+\alpha}$. Then

$$|\Pr[D(\mathbf{A}, \mathbf{A}X + \chi) = 1] - \Pr[D(\mathbf{A}, U) = 1]| > \epsilon.$$

We build a distinguisher that breaks $\mathsf{dist\text{-}LWE}_{\gamma,n,q,\chi}$. Let $\mathbf{A}'$ denote the uniform distribution over $\mathbb{F}_q^{\gamma\times n}$, X' denote the uniform distribution over $\mathbb{F}_q^n$, and U' denote the uniform distribution over $\mathbb{F}_q^{\gamma}$. We will build a distinguisher D' of polynomial size for which

$$|\Pr[D'(\mathbf{A}', \mathbf{A}'X' + \chi') = 1] - \Pr[D'(\mathbf{A}', U') = 1]| > (\epsilon - \mathtt{ngl}(n))(1 - \mathtt{ngl}(n)) \approx \epsilon. \tag{6.1}$$

D' will make a single call to D, so we focus on how to prepare a random block-fixing instance for D from the random instance that D' is given. The code for D' is given in Figure 6·1.

The distinguisher D' has an advantage when $\mathbf{S}$ is of rank α. This occurs with overwhelming probability:

Claim 6.3.1. *Let* $\mathbf{S} \xleftarrow{\$} \mathbb{F}_q^{\alpha\times(\alpha+\beta)}$ *be randomly generated. Then* $\Pr[\mathtt{rank}(\mathbf{S}) = \alpha] \geq$

```
1. Input $\mathbf{A}', \mathbf{b}'$, where $\mathbf{A}' \xleftarrow{\$} \mathbb{F}_q^{\gamma \times n}$ and $\mathbf{b}'$ is either uniform over $\mathbb{F}_q^{\gamma}$ or $\mathbf{b}' = \mathbf{A}'\mathbf{x}' + \mathbf{e}'$ for $\mathbf{e}' \xleftarrow{\$} \chi'$ and uniform $\mathbf{x}' \xleftarrow{\$} \mathbb{F}_q^n$.
2. Choose $\mathbf{R} \xleftarrow{\$} \mathbb{F}_q^{\alpha \times n}$ uniformly at random. Initialize $\mathbf{Q} \in \mathbb{F}_q^{\gamma \times (\alpha+\beta)}$ to be the zero matrix.
3. Let $\mathbf{b}^* = (\mathbf{b}', b^*_{\gamma+1}, \ldots, b^*_{\gamma+\alpha})$, for uniformly chosen $(b^*_{\gamma+1}, \ldots, b^*_{\gamma+\alpha}) \xleftarrow{\$} \mathbb{F}_q^{\alpha}$.
4. Choose $\mathbf{S} \xleftarrow{\$} \mathbb{F}_q^{\alpha \times (\alpha+\beta)}$ uniformly at random.
       If $\mathtt{rank}(\mathbf{S}) < \alpha$, stop and output a random bit.
5. Find a set of $\alpha$ linearly independent columns in $\mathbf{S}$. Let $T$ be the set of indices of these columns.
6. For all $1 \leq j \leq \alpha + \beta$, $j \notin T$:
       Choose $x_{n+j} \xleftarrow{\$} \mathbb{F}_q$ uniformly at random.
       For $i = 1, ..., \gamma$:
           Choose $\mathbf{Q}_{i,j} \xleftarrow{\$} \mathbb{F}_q$ uniformly at random.
           Set $b^*_i = b^*_i + \mathbf{Q}_{i,j} x_{n+j}$.
7. Initialize $\mathbf{A}^* = \left(\begin{array}{c|c} \mathbf{A}' & \mathbf{Q} \\ \hline \mathbf{R} & \mathbf{S} \end{array}\right)$.
8. For $i = 1, ..., \gamma$:
       Choose a row vector $\pi_i \leftarrow \mathbb{F}_q^{1 \times \alpha}$ uniformly at random.
       Set $\mathbf{a}_i \leftarrow \mathbf{a}^*_i + \pi_i(\mathbf{R}||\mathbf{S})$
       Set $b_i \leftarrow b^*_i + \pi_i(b^*_{\gamma+1}, ..., b^*_{\gamma+\alpha})^{\mathsf{T}}$
9. For $i = \gamma + 1, \ldots, \gamma + \alpha$:
       Set $\mathbf{a}_i \leftarrow \mathbf{a}^*_i$
       Set $b_i = b^*_i$.
10. Output $D(\mathbf{A}, \mathbf{b})$.
```

Figure 6·1: A PPT D' that distinguishes LWE using distinguisher for LWE w/ block fixing source

$1 - \mathtt{ngl}(n)$.

Proof. Direct result of Claim 2.3.11 because $q^{-\beta} = \mathtt{ngl}(n)$. □

The probability that a random $\mathbf{S}$ is not full rank is $\mathtt{ngl}(n)$ so the distinguisher D must still have an advantage when the matrix $\mathbf{S}$ is full rank. That is,

$$|\Pr[D(\mathbf{A}, \mathbf{A}X + \chi) = 1 | \mathtt{rank}(\mathbf{S}) = \alpha] - \Pr[D(\mathbf{A}, U) = 1 | \mathtt{rank}(\mathbf{S}) = \alpha]| > \epsilon - \mathtt{ngl}(n).$$

It suffices to show that D' prepares a good instance for D conditioned on $\mathbf{S}$ being full rank. We show this in the following three claims:

1. If $\mathbf{A}'$ is a random matrix then $\mathbf{A}$ is a random matrix subject to the condition that $\mathtt{rank}(\mathbf{S}) = \alpha$.
2. If $\mathbf{b}' = \mathbf{A}'\mathbf{x}' + \mathbf{e}'$ for uniform $\mathbf{A}'$ and $\mathbf{x}'$, then $\exists \mathbf{x}$ (uniformly distributed and independent of $\mathbf{A}$ and $\mathbf{e}'$) such that $\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $\mathbf{e}_i = \mathbf{e}'_i$ for $1 \le i \le \gamma$ and $\mathbf{e}_i = 0$ otherwise.
3. If the conditional distribution $\mathbf{b}' \,|\, \mathbf{A}'$ is uniform, then the conditional distribution $\mathbf{b} \,|\, \mathbf{A}$ is also uniform.

Claim 6.3.2. *The matrix $\mathbf{A}$ is distributed as a uniformly random choice from the set of all matrices whose bottom-right $\alpha \times (\alpha + \beta)$ submatrix $\mathbf{S}$ satisfies $\mathtt{rank}(\mathbf{S}) = \alpha$.*

Proof. The bottom α rows of $\mathbf{A}$ (namely, $\mathbf{R}|\mathbf{S}$) are randomly generated (conditioned on $\mathtt{rank}(\mathbf{S}) = \alpha$). The top left $\gamma \times n$ quadrant of $\mathbf{A}$ is also random, because it is produced as a sum of a uniformly random $\mathbf{A}'$ with some values that are uncorrelated with $\mathbf{A}'$. The submatrix of the top-right $\gamma \times (\alpha + \beta)$ quadrant corresponding to $\mathbf{Q}_{T^c}$ (recall this is the restriction of $\mathbf{Q}$ to the columns not in T) is also random, because it is initialized with random values to which some uncorrelated values are then added. It is important to note that all these values are independent of π_i values.

Thus, we restrict attention to the $\gamma \times \alpha$ submatrix of $\mathbf{A}$ that corresponds to $\mathbf{Q}_T$ in $\mathbf{A}^*$ (note that these values are 0 in $\mathbf{A}^*$). Consider a particular row i. That row is computed as $\pi_i \mathbf{S}_T$. Since $\mathbf{S}_T$ is a full rank square matrix and π_i is uniformly and independently generated, that row is also uniform and independent of other entries in $\mathbf{A}$. □

Claim 6.3.3. *If D' is provided with input distributed as $\mathbf{A}', \mathbf{b}' = \mathbf{A}'\mathbf{x}' + \mathbf{e}'$ then $\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where*

- *$e_i = e'_i$ for $1 \leq i \leq \gamma$,*
- *$e_i = 0$ for $\gamma < i \leq \gamma + \alpha$,*
- *$x_j = x'_j$ for $1 \leq j \leq n$,*
- *and x_j is uniform and independent of $\mathbf{A}$ and $\mathbf{e}'$ for $n < j \leq n + \alpha + \beta$,*

Proof. Partially define $\mathbf{x}$ as $x_j = x'_j$ if $1 \leq j \leq n$ and x_j as the value generated in step 6 for $j > n$ and $j \notin T$. Define the remaining variables $\mathbf{x}_T$ as the solution to the following system of equations.

$$\mathbf{S}_T \mathbf{x}_T = \begin{pmatrix} b^*_{\gamma+1} \\ \vdots \\ b^*_{\gamma+\alpha} \end{pmatrix} - \mathbf{R}\mathbf{x}' - \mathbf{S}_{T^c}\mathbf{x}_{T^c} \tag{6.2}$$

A solution $\mathbf{x}_T$ exists as $\mathbf{S}_T$ is full rank. Moreover, it is uniform and independent of $\mathbf{A}$ and $\mathbf{e}$, because $b^*_{\gamma+1}, \ldots, b^*_{\gamma+\alpha}$ are uniform and independent of $\mathbf{A}$ and $\mathbf{e}$.

We now show that $\mathbf{b}^* = \mathbf{A}^*\mathbf{x} + \mathbf{e}$. All entries in matrix $\mathbf{Q}$ corresponding to variables in T are set to zero. Thus, the values of $\mathbf{x}^T$ do not affect b^*_i for $1 \leq i \leq \gamma$. The values of $\mathbf{x}_{T^c}$ are manually set, and $\mathbf{Q}_{i,j}\mathbf{x}_j$ is added to the corresponding b^*_i. Thus, for $1 \leq i \leq \gamma$, we have $\mathbf{b}^* = \mathbf{A}^*\mathbf{x} + \mathbf{e}$. For $\gamma < i$, this constraint is also satisfied by the values of $\mathbf{x}_T$ set in Equation 6.2.

Thus, it remains to show that step 8 preserves this solution. We now show that for all rows $1 \leq i \leq \gamma$, if $b^*_i = \mathbf{a}^*_i\mathbf{x} + e_i$ then $b_i = \mathbf{a}_i\mathbf{x} + e_i$. Recall the other rows are not modified. We have the following for $1 \leq i \leq \gamma$:

$$\begin{aligned} \mathbf{a}_i\mathbf{x} + e_i &= \left(\mathbf{a}^*_i + \pi_i(\mathbf{R}||\mathbf{S})\right)\mathbf{x} + e_i \\ &= \mathbf{a}^*_i\mathbf{x} + e_i + \pi_i(\mathbf{R}||\mathbf{S})\mathbf{x} \\ &= b^*_i + \pi_i(\mathbf{R}||\mathbf{S})\mathbf{x} \end{aligned}$$

Recall that $b_i = b^*_i + \pi_i(b^*_{\gamma+1}, ..., b^*_{\gamma+k})$. We consider the product $(\mathbf{R}||\mathbf{S})\mathbf{x}$. It suffices to show that $(\mathbf{R}||\mathbf{S})\mathbf{x} = (b^*_{\gamma+1}, ..., b^*_{\gamma+\alpha})$,

$$\begin{aligned}(\mathbf{R}||\mathbf{S})\mathbf{x} &= \mathbf{R}\begin{pmatrix}\mathbf{x}_1\\ \vdots \\ \mathbf{x}_n\end{pmatrix} + \mathbf{S}_{T^c}\mathbf{x}_{T^c} + \mathbf{S}_T\mathbf{x}_T \\ &= \mathbf{R}\begin{pmatrix}\mathbf{x}_1\\ \vdots \\ \mathbf{x}_n\end{pmatrix} + \mathbf{S}_{T^c}\mathbf{x}_{T^c} + \begin{pmatrix}b^*_{\gamma+1}\\ \vdots \\ b^*_{\gamma+\alpha}\end{pmatrix} - \mathbf{R}\begin{pmatrix}\mathbf{x}_1\\ \vdots \\ \mathbf{x}_n\end{pmatrix} - \mathbf{S}_{T^c}\mathbf{x}_{T^c} \\ &= \begin{pmatrix}b^*_{\gamma+1}\\ \vdots \\ b^*_{\gamma+\alpha}\end{pmatrix}\end{aligned}$$

This completes the proof of the claim. □

Claim 6.3.4. *If the conditional distribution* $\mathbf{b}' \,|\, \mathbf{A}'$ *is uniform, then* $\mathbf{b} \,|\, \mathbf{A}$ *is also uniform.*

Proof. Since $\mathbf{R}, \mathbf{S}$, and $\mathbf{Q}$ are chosen independently of $\mathbf{b}'$, the distribution $\mathbf{b}' \,|\, \mathbf{A}^*$ is uniform. Let $\mathbf{b}^*$ be the vector generated after step 6. Its first γ coordinates are computed by adding the uniform vector $\mathbf{b}'$ to values that are independent of $\mathbf{b}^*$, and its remaining α coordinates $b^*_{\gamma+1}, \ldots, b^*_{\gamma+\alpha}$ are chosen uniformly. Thus $\mathbf{b}^* \,|\, \mathbf{A}^*$ is uniform.

Let $\mathbf{\Pi}$ represent the matrix formed by π_i. It is independent of $\mathbf{b}^*$ and $\mathbf{A}^*$, so $\mathbf{b}^* \,|\, (\mathbf{A}^*, \mathbf{\Pi})$ is uniform. Let $\mathbf{\Pi}' = \left(\begin{array}{c|c}\mathbf{I}_\gamma & \mathbf{\Pi} \\ \hline \mathbf{0} & \mathbf{I}_\alpha\end{array}\right)$. Note that $\mathbf{b} = \mathbf{\Pi}'\mathbf{b}^*$. Since $\mathbf{b}^* \,|\, (\mathbf{A}^*, \mathbf{\Pi})$ is uniform, and $\mathbf{\Pi}'$ is invertible, $\mathbf{b} \,|\, (\mathbf{A}^*, \mathbf{\Pi})$ must also be uniform. Since $\mathbf{A}$ is a deterministic function of $\mathbf{A}^*$ and $\mathbf{\Pi}$ (assuming Step 5 is deterministic—if not, we can fix the coins used), the distribution $\mathbf{b} \,|\, \mathbf{A}$ is the same as $\mathbf{b} \,|\, (\mathbf{A}^*, \mathbf{\Pi})$ and is thus also uniform. □

Finally, the reduction runs in polynomial time and together Claims 6.3.2, 6.3.3, and 6.3.4 show that when $\texttt{rank}(\mathbf{S}) = \alpha$ the distinguisher D' properly prepares the instance thus,

$$\begin{aligned}
&|\Pr[D'(\mathbf{A}, \mathbf{A}X + \chi) = 1] - \Pr[D'(\mathbf{A}, U) = 1]| \\
&= (|\Pr\left[D'(\mathbf{A}', \mathbf{u}') = 1 | \texttt{rank}(\mathbf{S}) = \alpha\right] \\
&\quad - \Pr\left[D'(\mathbf{A}', \mathbf{b}' = \mathbf{A}'\mathbf{x} + \mathbf{e}) = 1 | \texttt{rank}(\mathbf{S}) = \alpha\right]|) \Pr[\texttt{rank}(\mathbf{S}) = \alpha] \\
&= (|\Pr[D(\mathbf{A}, \mathbf{A}X + \chi) = 1 | \texttt{rank}(\mathbf{S}) = \alpha] \\
&\quad - \Pr[D(\mathbf{A}, U) = 1 | \texttt{rank}(\mathbf{S}) = \alpha]|) \Pr[\texttt{rank}(\mathbf{S}) = \alpha] \\
&\geq (\epsilon - \texttt{ngl}(n))(1 - \texttt{ngl}(n)) \approx \epsilon
\end{aligned}$$

Where the second line follows because we can detect when $\texttt{rank}(\mathbf{S}) < \alpha$ and output a random bit in this case. Thus, Equation (6.1) is satisfied, this completes the proof. □

6.4 Parameter Settings for Construction 6.1.1

In this section, we explain the different parameters that go into our construction. In Theorem 6.1.8 we give a lossless fuzzy extractor from a security parameter n and an error t. In this section, we discuss constraints imposed by 1) efficient decoding 2) maintaining security of the LWE instance and 3) ensuring no entropy loss of the construction. We begin by reviewing the parameters that make up our construction:

- $|W|$: The length of the source.
- t: Number of errors that can be supported.
- n: LWE security parameter (i.e., number of field elements in X), which must be greater than some minimum value n_0 for security.
- q: The size of the field.
- ρ: The fraction of the field needed for error sampling.
- γ: The length of the string w in symbols.
- k: The number of hardcore bits in X (from Lemma 6.1.4).

We will split the source $|W|$ into γ blocks each of size $2\rho q+1$ (that is, $|W| = \gamma\log(2\rho q+1)$). We will ignore the parameter $|W|$ and focus on t, n, q, ρ, and γ. As stated above we have three constraints:

- Maintain security of LWE. If we assume GAPSVP and SIVP are hard to approximate within polynomial factors then Lemma 6.1.3 says that we get security for all n greater than some minimum n_0 and $q = \texttt{poly}(n)$ and $\rho q \geq 2n^{1/2+\sigma}\gamma = \texttt{poly}(n)$. The only reason to increase ρq over this minimum amount (other than security) is if the number of errors in W decreases with a slightly larger block size. We ignore this effect and assume that $\rho q = 2n^{1/2+\sigma}\gamma$.
- Maintain efficient decoding of Construction 6.1.5. Using Lemma 6.1.6, this means that $t \leq d\log n(\gamma/n-2)$.
- Minimize entropy loss of the construction. We will output $X_{1,\ldots,k}$ so the entropy loss of the construction is $|W| - |X_{1,\ldots,k}|$. We want the entropy loss to be zero, that is, $|W| = |X_{1,\ldots,k}|$. Substituting, one has $\gamma\log 2\rho q+1 = k\log q$.

Collecting constraints we can support any setting where t, n, q, ρ, γ, k satisfy the following constraints (for constants d, f):

$$\begin{aligned} n_0 &< n-k \\ t &\leq d\log n\left(\frac{\gamma}{n}-2\right) \\ q &= n^f \\ \rho q &= 2n^{1/2+\sigma}\gamma \\ \gamma\log(2\rho q+1) &= k\log q \end{aligned}$$

Substituting $q = n^f$ and $\rho q = 2n^{1/2+\sigma}\gamma$ yields the following system of equations:

$$\begin{aligned} n_0 &< n - k \\ t &\leq d \log n \left(\frac{\gamma}{n} - 2\right) \\ \gamma \log(4n^{1/2+\sigma}\gamma + 1) &= k \log n^f \end{aligned}$$

This is the most general form of our construction, we can support any n, t, γ that satisfy these equations for constants d, f. However, the last equation may have no solution for f constant. Putting the last equation in terms of f one has:

$$\begin{aligned} n_0 &< n - k \\ t &\leq d \log n \left(\frac{\gamma}{n} - 2\right) \\ f &= \frac{\gamma}{k} \frac{\log 4n^{1/2+\sigma}\gamma + 1}{\log n} \end{aligned}$$

To ensure f is a constant, we set $t = c \log n$ for some constant c and that $k = n/g$ for some constant $g > 1$. Finally we assume that γ is the minimum value such that $t \leq d \log n(\gamma/n - 2)$ (that is, there are only as many dimensions as necessary for decoding using Lemma 6.1.6):

$$\begin{aligned} n_0 &< n - k \\ \gamma &= \frac{(c/d + 2)n \log n}{\log n} = (\frac{c}{d} + 2)n \\ f &= \frac{\gamma}{k} \frac{\log 4n^{1/2+\sigma}\gamma + 1}{\log n} = \frac{g(c + 2d)}{d} \frac{\log(\frac{4(c+2d)}{d} n^{3/2+\sigma} + 1)}{\log n} \end{aligned}$$

Note that f is at a constant in n. Assuming $n - k = n(1 - 1/g) > n_0$ and letting $t = c \log n$ we get the following setting:

$$\begin{aligned}
\gamma &= (\frac{c}{d} + 2)n \\
q &= n^f = n^{\frac{\gamma}{k} \frac{\log(4n^{1/2+\sigma}\gamma+1)}{\log n}} = \texttt{poly}(n) \\
\rho q &= 2n^{1/2+\sigma}\gamma = 2(\frac{c}{d} + 2)n^{3/2+\sigma}
\end{aligned}$$

Note, that $f > \frac{\gamma}{k} \geq \frac{\gamma}{n} \geq \frac{(c/d+2)n}{n} \geq 3$ as long as $d < c$ (this also ensures that $\gamma \geq 3n$, as required for Lemma 6.1.6 to hold). Since $\rho q = 2n^{1/2+\rho}\gamma = O(n^{5/2})$ in our setting $\rho = O(n^{-1/2})$. Thus, for large enough settings of parameters ρ is less than $1/10$ as required by Lemma 6.1.3.

Furthermore, we get decoding using $O(n^{4d+3})$ $\mathbb{F}_q$ operations. We can output a k fraction of X and the bits will be pseudorandom (conditioned on $\mathbf{A}, \mathbf{A}X + W$). The parameter g allows is a tradeoff between the number of dimensions needed for security and the size of the field q. In Theorem 6.1.8, we set $g = 2$ and output the first half of X. Setting $1 < g < 2$ achieves an increase in output length (over the input length of W). We also (arbitrarily) set $\sigma = 1/2$ to simplify the statement of Theorem 6.1.8, making $\rho q = 2(c/d + 2)n^2$.

6.4.1 Parameter Settings for Theorem 6.2.3

We repeat parameter settings for block fixing sources. We now have $\gamma + \alpha$ as the number of samples, while $n + \alpha + \omega(1)$ is the number of variables. We can support any setting where $t, n, q, \rho, \gamma, k, \alpha$ satisfy the following constraints (for $\beta = \omega(1)$ and constants d, f):

$$\begin{aligned} n_0 &< n - k - \alpha - \beta \\ t &\leq d \log n \left(\frac{\gamma}{n} - 2\right) \\ q &= n^f \\ \rho q &= 2n^{1/2+\sigma}\gamma \\ \gamma \log(2\rho q + 1) &= k \log q \end{aligned}$$

Substituting $q = n^f$ and $\rho q = 2n^{1/2+\sigma}\gamma$ yields the following system of equations:

$$\begin{aligned} n_0 &< n - k - \alpha - \beta \\ t &\leq d \log n \left(\frac{\gamma}{n} - 2\right) \\ \gamma \log(4n^{1/2+\sigma}\gamma + 1) &= k \log n^f \end{aligned}$$

As before we can support any setting any n, t, γ, α that satisfy these equations for $\beta = \omega(1)$ and constants d, f. However, the last equation may have no solution for f constant. Putting the last equation in terms of f one has:

$$\begin{aligned} n_0 &< n - k - \alpha - \beta \\ t &\leq d \log n \left(\frac{\gamma}{n} - 2\right) \\ f &= \frac{\gamma}{k} \frac{\log(4n^{1/2+\sigma}\gamma + 1)}{\log n} \end{aligned}$$

To ensure f is a constant, we set $t = c \log n$ for some constant c and that $k, \alpha = n/3$ and $\beta = \log n$. Finally we assume that γ is the minimum value such that $t \leq d \log n(\gamma/n - 2)$ (that is, there are only as many dimensions as necessary for decoding using Lemma 6.1.6):

$$
\begin{aligned}
n_0 &< n/3 - \log n \\
\gamma &= \frac{(c/d+2)n\log n}{\log n} = (\frac{c}{d}+2)n \\
f &= \frac{\gamma}{k}\frac{\log(4n^{1/2+\sigma}\gamma+1)}{\log n} = \Big(3(\frac{c}{d}+2)\Big)\frac{\log(4(\frac{c}{d}+2)n^{3/2+\sigma}+1)}{\log n} = O(1)
\end{aligned}
$$

Assuming $n/3 - \log(n) > n_0$ and letting $t = c\log n$ we get the following setting:

$$
\begin{aligned}
\gamma &= (\frac{c}{d}+2)n \\
q = n^f &= n^{\frac{\gamma}{n}\frac{\log(4n^{1/2+\sigma}\gamma+1)}{\log n}} = \texttt{poly}(n) \\
\rho q = 2n^{1/2+\sigma}\gamma &= 2(\frac{c}{d}+2)n^{3/2+\sigma}
\end{aligned}
$$

As before we arbitrarily set $\sigma = 1/2$, giving $\rho q = 2(\frac{c}{d}+2)n^2$. Also, if $c < d$ then we get efficient decoding and $\rho = o(1)$ satisfying the condition of Lemma 6.1.3.

Chapter 7

Computational Fuzzy Extractors from Point Obfuscation

In Chapter 6, we showed it was possible to construct a computational fuzzy extractor whose output key length was as long as the starting entropy. In this chapter, we show practical constructions of computational fuzzy extractors with additional properties. Both constructions are based on point obfuscation and can support more errors than entropy. In this section, we do not concentrate on the length of key as computational techniques we use can expand the key. We instead focus on supporting a wide classes of sources.

7.1 A reusable computational fuzzy extractor

In Construction 5.2.2, we showed a fuzzy extractor for a family of distributions with more errors than entropy. Using computational techniques we are able to retain many of the advantages of Construction 5.2.2 and achieve a reusable fuzzy extractor.

The construction samples a random subset of blocks $W_{j_1}, ..., W_{j_\eta}$ and obfuscates the concatenation of these blocks. Denote this concatenated value by V_1. This process is repeated to produce $V_1, ..., V_\ell$ where at least one V_i should be correct to "unlock" the correct key. Let $\mathsf{Sample}_{\gamma,\eta}(\cdot)$ be an algorithm that outputs a random subset of $\{1, ..., \gamma\}$ of size η given let r_{sam} bits of randomness.

Construction 7.1.1 (Sample-then-Obfuscate)**.** *Let $\mathcal{Z}$ be an alphabet, and let $W = W_1, ..., W_\gamma$ be a source where each W_j is over $\mathcal{Z}$. Let η be a parameter, and $\mathcal{O}$ be an obfuscator for the family of digital lockers with κ-bit outputs. Define* $\mathsf{Gen}, \mathsf{Rep}$ *as:*

```
Gen
  1. Input: $w = w_1, ..., w_\gamma$
  2. Sample $\mathsf{key} \xleftarrow{\$} \{0,1\}^\kappa$.
  3. For $i = 1, ..., \ell$:
       (i) Select $\lambda_i \xleftarrow{\$} \{0,1\}^{r_{sam}}$.
      (ii) Set $j_{i,1}, ..., j_{i,\eta} \leftarrow \mathsf{Sample}_{\gamma,\eta}(\lambda_i)$
     (iii) Set $v_i = w_{j_{i,1}}, ..., w_{j_{i,\eta}}$.
      (iv) Set $\rho_i = \mathcal{O}(I_{v_i,r})$.
       (v) Set $p_i = \rho_i, \lambda_i$.
  4. Output $(\mathsf{key}, p)$, where $p = p_1 \dots p_\ell$.
```

```
Rep
  1. Input: $(w' = w'_1, ..., w'_\gamma, p)$
  2. For $i = 1, ..., \ell$:
       (i) Parse $p_i$ as $\rho_i, \lambda_i$.
      (ii) $j_{i,1}, ..., j_{i,\eta} \leftarrow \mathsf{Sample}_{\gamma,\eta}(\lambda_i)$.
     (iii) Set $v'_i = w'_{j_{i,1}}, ..., w'_{j_{i,\eta}}$.
      (iv) Set $\rho_i(v'_i) = r_i$.
           If $\mathsf{key}_i \neq \perp$ output $\mathsf{key}_i$.
  3. Output $\perp$.
```

The use of a computational primitive (obfuscation of digital lockers) allows us to sample multiple times, because we need to argue only about individual entropy of V_i, as opposed to the information-theoretic setting, where it would be necessary to argue about the entropy of the joint variable V. This is the property that allows reusability.

This construction uses a naïve sampler that takes truly random samples, but the public randomness may be substantially decreased by using more sophisticated samplers. (See Goldreich [Gol97] for an introduction to samplers.)

Theorem 7.1.2. *Let $\mathcal{Z}$ be an alphabet. Let n be a security parameter. Let $\mathcal{W}$ be the family of $(\alpha = \Omega(1), \beta \leq \gamma(1 - \Theta(1)))$-partial block sources over $\mathcal{Z}^\gamma$ where $\gamma = \Omega(n)$. Let η be such that $\eta = \omega(\log n)$ and $\eta = o(\gamma)$, and let $c > 1$ be a constant and ℓ be such that $\ell = n^c$. Let $\mathcal{O}$ be an ℓ-composable VGB obfuscator for digital lockers (with κ bit outputs) with auxiliary inputs. Then for every $s_{sec} = \mathtt{poly}(n)$ there exists some $\epsilon_{sec} = \mathtt{ngl}(n)$ such that Construction 7.1.1 is a $(\mathcal{Z}^\gamma, \mathcal{W}, \kappa, t)$-computational fuzzy extractor that is $(\epsilon_{sec}, s_{sec})$-hard with error δ for*

$$t \leq -\frac{(c-1)}{2}\frac{(\gamma - \eta)\log n}{\eta} = o(\gamma)$$

$$\delta = e^{-n}$$

7.1.1 Security of Construction 7.1.1

In this section we show security of Construction 7.1.1. With overwhelming probability, at each of the ℓ iterations, the sampler will choose enough coordinates of W that have high entropy, making V_i have sufficient entropy. Once each of the $V_1, ..., V_\ell$ have high entropy the obfuscations are unlikely to return a value other than $\perp$ to an adversary. We begin by showing that each V_i is statistically close to a high entropy distribution. Let Λ represent the random variable of all the coins used by Sample and $\lambda = \lambda_1 \ldots \lambda_\ell$ be some particular outcome.

Lemma 7.1.3. *Let all variables be as in Theorem 7.1.2. There exists* $\epsilon_{sam} = O(e^{-\eta}) = \mathtt{ngl}(n)$ *and* $\alpha' = \alpha\eta(\gamma - \beta - \eta)/\gamma = \omega(\log n)$ *such that for each* i*,*

$$\Pr_{\lambda \leftarrow \Lambda}[\mathrm{H}_\infty(V_i | \Lambda = \lambda) \geq \alpha'] \geq 1 - \epsilon_{sam}.$$

Proof. Consider some fixed i. Recall that there a set J of size $\gamma - \beta = \Theta(\gamma)$ such that each w and block $j \in J$, $\mathrm{H}_\infty(W_j | W_1 = w_1, ..., W_{j-1} = w_{j-1}, W_{j+1} = w_{j+1}, ..., W_\gamma = w_\gamma) \geq \alpha$. Since this is a worst case guarantee, the entropy of V_i can be deduced from the number of symbols in V_i that come from J. Namely, Denote by $X = |\{j_{i,1}, ..., j_{i,\eta}\} \cap J|$.

Claim 7.1.4.

$$\mathrm{H}_\infty(V_i | \Lambda = \lambda) \geq \alpha X.$$

Proof. Denote by $j_1, ..., j_\eta$ the indices selected by the randomness λ_i. We begin by noting that

$$\begin{aligned} \mathrm{H}_\infty(V_i | \Lambda = \lambda) &= -\log \max_{v \in V_i} \Pr[V_i = v | \Lambda = \lambda] \\ &= -\log \max_{w_{j_1}, ..., w_{j_\eta}} \Pr[W_{j_1} = w_{j_1} \wedge \cdots \wedge W_{j_\eta} w_{j_\eta}]. \end{aligned}$$

Then

$$\begin{aligned}
&\max_{w_{j_1},\dots,w_{j_\eta}} \Pr[W_{j_1} = w_{j_1} \wedge \cdots \wedge W_{j_\eta} = w_{j_\eta}] \\
&\qquad = \max_{w_{j_1},\dots,w_{j_\eta}} \prod_{k=1}^{\eta} \Pr[W_{j_k} = w_{j_k} | W_{j_{k-1}} = w_{j_{k-1}} \wedge \dots \wedge W_{j_1} = w_{j_1}] \\
&\qquad \leq \prod_{k=1}^{\eta} \max_{w_{j_1},\dots,w_{j_\eta}} \Pr[W_{j_k} = w_{j_k} | W_{j_{k-1}} = w_{j_{k-1}} \wedge \dots \wedge W_{j_1} = w_{j_1}] \\
&\qquad \leq \prod_{k=1}^{\eta} \max_{w_1,\dots,w_\gamma} \Pr[W_{j_k} = w_{j_k} | W_1 = w_1 \wedge \dots \wedge W_{j_k-1} = w_{j_k-1}]
\end{aligned}$$

Taking the negative logarithm of both sides we have that

$$\begin{aligned}
\mathrm{H}_\infty(V_i|\Lambda = \lambda) &\geq \sum_{k=1}^{\eta} \min_{w_1,\dots,w_\gamma} \mathrm{H}_\infty(W_{j_k}|W_1 = w_1 \wedge \dots \wedge W_{j_k-1} = w_{j_k-1}) \\
&\geq \sum_{j_k \in J} \alpha = \alpha X
\end{aligned}$$

This completes the proof of Claim 7.1.4. □

We note that X is distributed according to the hypergeometric distribution, and that $\mathbb{E}[X] = \eta(\gamma - \beta)/\gamma$. Using the tail bounds from [Chv79, Ska13], we can conclude that $\Pr[X \leq \mathbb{E}[X]/2] \leq e^{-2((\gamma-\beta)/2\gamma)^2\eta} = O(e^{-\eta})$. Thus, setting $\alpha' = \frac{\alpha\eta(\gamma-\beta)}{2\gamma}$ and applying Claim 7.1.4, we conclude that

$$\Pr[\mathrm{H}_\infty(V_i) \geq \alpha'] \geq 1 - O(e^{-\eta}).$$

This completes the proof of Lemma 7.1.3. □

We can then argue that all V_i simultaneously have individual entropy with good probability (by union bound):

Corollary 7.1.5. *Let ϵ_{sam} and α' be as in Lemma 7.1.3, and all the other variables be as in Theorem 7.1.2. Then $\Pr_{\lambda \leftarrow \Lambda}[\forall i, \mathrm{H}_\infty(V_i|\Lambda = \lambda) \geq \alpha'] \geq 1 - \ell\epsilon_{sam}$.*

Once all V_i all simultaneously have good entropy, the adversary only sees $\perp$ as an output from the obfuscations (with overwhelming probability). If the adversary only sees $\perp$ from the obfuscations, they have no information about key. This is each output $V_1, ..., V_\ell$ is hard to guess. We call this type of distribution block unguessable source:[1]

Definition 7.1.6. *Let $I_v(\cdot, \cdot)$ be an oracle that returns*

$$I_v(j, v_j') = \begin{cases} 1 & v_j = v_j' \\ 0 & \textit{otherwise.} \end{cases}$$

*A source $V = V_1|...|V_\gamma$ is a (q, α, β)-*unguessable block source *if there exists a set $J \subset \{1, ..., \gamma\}$ of size at least $\gamma - \beta$ such that for any unbounded adversary S with oracle access to I_v making at most q queries*

$$\forall j \in J, \tilde{\mathrm{H}}_\infty(V_j | View(S^{I_V(\cdot,\cdot)})) \geq \alpha.$$

This is made formal in the following corollary:

Corollary 7.1.7. *Let ϵ_{sam}, α' be as in Lemma 7.1.3, and all the other variables be as in Theorem 7.1.2. Take any $q = \mathtt{poly}(n)$. For $\alpha'' = \alpha' - 1 - \log(q+1) = \omega(\log n)$, with probability $1 - \ell\epsilon_{sam}$ over the choice of $\Lambda = \lambda$, the distribution $V|\Lambda = \lambda$ is a $(q, \alpha'', 0)$-unguessable block source.*

Finally, we can show the construction is secure if the inputs form a unguessable block source.

Lemma 7.1.8. *Let all the variables be as in Theorem 7.1.2. For every $s_{sec} = \mathtt{poly}(n)$ there exists $\epsilon_{sec} = \mathtt{ngl}(n)$ such that $\delta^{\mathcal{D}_{s_{sec}}}((\mathsf{Key}, P), (U_\kappa, P)) < \epsilon_{sec}$.*

Proof. Let $\mathcal{O}$ be a ℓ-composable VGB obfuscator with auxiliary input for digital lockers over $\mathcal{Z}$[2] . Let V be a $(q, \alpha'' = \omega(\log n), 0)$-unguessable block source. Our goal is to show that for all $s_{sec} = \mathtt{poly}(n)$ there exists $\epsilon_{sec} = \mathtt{ngl}(n)$ such that $\delta^{\mathcal{D}_{s_{sec}}}((R, P), (U, P)) \leq \epsilon_{sec}$.

[1] In this definition we allow there to be a set of weak blocks. Construction 7.2.3 is secure for sources that satisfy this weaker definition.

[2] In this proof we only consider the case where the sampling has produced a block unguessable source. The negligible portion of the time when this does not happen in included in the security of Theorem 7.1.2

Suppose not, that is suppose there is some $s_{sec} = \mathtt{poly}(n)$ such that exists $\epsilon_{sec} = \mathtt{poly}(n)$ and $\delta^{\mathcal{D}_{s_{sec}}}((\mathsf{Key}, P), (U, P)) > \epsilon_{sec}$. Let D be such a distinguisher of size at most s_{sec}. That is,

$$|\,\mathbb{E}[D(\mathsf{Key}, P)] - \mathbb{E}[D(U, P)] > \epsilon_{sec} = 1/\mathtt{poly}(n).$$

Define the oracle $I_{v_1,...,v_\ell,r}(\cdot,\cdot)$ as follows:

$$I_{v_1,...,v_\ell,\mathsf{key}}(x, i) = \begin{cases} \mathsf{key} & v_i = x \\ \perp & \text{otherwise.} \end{cases}$$

By the security of obfuscation (Definition 2.4.1), there exists a unbounded time simulator S (making at most q queries) such that

$$|\,\mathbb{E}[D(\mathsf{Key}, P_1, ..., P_\ell)] - \mathbb{E}[S^{I_{v_1,...,v_\ell,r}(\cdot,\cdot)}(\mathsf{Key}, 1^{\ell \log |Z|})]| \leq \epsilon_{sec}/3. \tag{7.1}$$

We now prove S cannot distinguish between Key and U.

Lemma 7.1.9. $\mathbf{SD}(S^{I_{v_1,...,v_\ell,r}(\cdot,\cdot)}(\mathsf{Key}, 1^{\ell \log |Z|}), S^{I_{v_1,...,v_\ell,r}(\cdot,\cdot)}(U, 1^{\ell \log |Z|})) \leq \ell 2^{-\alpha''}$.

Proof. It suffices to show that for any two values in $\{0,1\}^\kappa$, the statistical distance is at most $\ell 2^{-\alpha''}$.

Lemma 7.1.10. *Let* key *be true value encoded in* I *and let* $u \in \{0,1\}^\kappa$. *Then,*

$$\mathbf{SD}(S^{I_{v_1,...,v_\ell,\mathsf{key}}(\cdot,\cdot)}(\mathsf{key}, 1^{\ell \log |Z|}), S^{I_{v_1,...,v_\ell,r}(\cdot,\cdot)}(u, 1^{\ell \log |Z|})) \leq \ell 2^{-\alpha''}.$$

Proof. Recall that for all j, $\tilde{\mathrm{H}}_\infty(V_j | View(S)) \geq \alpha''$. The only information about the correct value of r is contained in the query responses. When all responses are $\perp$ the view of S is identical when presented with key or u. We now show that for any value of key all queries return $\perp$ with probability $1 - 2^{-\alpha''}$. Suppose not, that is suppose, the probability of at least one nonzero response is $> 2^{-(\alpha'')}$.

When there is a response other than $\perp$ for some j this means that there is no remaining min-entropy in V_j. If this occurs with over $2^{-\alpha''}$ probability this violates the block unguessability of V (Definition 7.1.6). By the union bound over the indices j the total probability of a response other than $\perp$ is at most $\ell 2^{-\alpha''}$. Thus, for all key, u the statistical distance is at most $\ell 2^{-\alpha''}$. This concludes the proof of Lemma 7.1.10. □

By averaging over all points in $\{0,1\}^\kappa$ we conclude that

$$\mathbf{SD}(S^{I_{v_1,\ldots,v_\ell,r}X(\cdot,\cdot)}(\mathsf{Key}, 1^{\ell\log|Z|}), S^{I_{v_1,\ldots,v_\ell,r}(\cdot,\cdot)}(U, 1^{\ell\log|Z|})) < \ell 2^{-\alpha''}.$$

This completes the proof of Lemma 7.1.9. □

Now by the security of obfuscation we have that

$$|\,\mathbb{E}[D(\mathsf{Key}, P_1, \ldots, P_\ell)] - \mathbb{E}[S^{I_{v_1,\ldots,v_\ell,r}(\cdot,\cdot)}(\mathsf{Key}, 1^{\ell\log|Z|})]| \le \epsilon_{sec}/3. \tag{7.2}$$

Combining Equations 7.1 and 7.2 and Lemma 7.1.9, we have

$$\begin{aligned}\delta^D((\mathsf{Key}, P), (U, P)) &\le |\,\mathbb{E}[D(\mathsf{Key}, P_1, \ldots, P_\ell)] - \mathbb{E}[S^{I_{v_1,\ldots,v_\ell,r}(\cdot,\cdot)}(\mathsf{Key}, 1^{\ell\log|Z|})]| \\ &+ |\,\mathbb{E}[S^{I_{v_1,\ldots,v_\ell,r}(\cdot,\cdot)}(\mathsf{Key}, 1^{\ell\log|Z|})] - \mathbb{E}[S^{I_{v_1,\ldots,v_\ell,r}(\cdot,\cdot)}(U, 1^{\ell\log|Z|})]| \\ &+ |\,\mathbb{E}[S^{I_{v_1,\ldots,v_\ell,r}(\cdot,\cdot)}(U, 1^{\ell\log|Z|})] - \mathbb{E}[D(U, P_1, \ldots, P_\ell)]| \\ &\le \epsilon_{sec}/3 + \ell 2^{-\alpha''} + \epsilon_{sec}/3 \\ &\le 2\epsilon_{sec}/3 + \mathtt{ngl}(n) < \epsilon_{sec}.\end{aligned}$$

This is a contradiction and completes the proof of Lemma 7.1.8. □

7.1.2 Correctness of Construction 7.1.1

We encode the entire key in each obfuscation. For correctness, at least one of the repeated readings must be correct with overwhelming probability. Let V_i represent one of the initial readings and V_i' represent a repeated reading. For showing correctness we must show that $\Pr[\forall i, V_i \neq V_i'] < \mathtt{ngl}(n)$.

Lemma 7.1.11. *Let all the variables be as in Theorem 7.1.2. Then* $\Pr[\forall i, v_i \neq v_i'] < \mathtt{ngl}(n)$*, where the probability is over the coins of* Gen*.*

Proof. Recall that $\mathsf{dis}(w, w') \le t$ and that the locations of the errors is independent of the selected locations. Denote by $\mu = -\frac{(c-1)\log n}{2}$. Since $\eta = \omega(\log n)$, we will assume

$\eta \geq 2\mu$. We begin by computing the probability that a single $v_i = v_i'$.

$$\begin{aligned}
\Pr[v_i = v_i'] &= \Pr[w \text{ and } w' \text{ agree on positions } j_{i,1}, ..., j_{i,\eta}] \\
&\geq \prod_{j=0}^{\eta-1} \left(1 - \frac{t}{\gamma - j}\right) \geq \prod_{j=0}^{\eta-1} \left(1 - \frac{\mu(\gamma-\eta)/\eta}{\eta - j}\right) \\
&\geq \prod_{j=0}^{\eta-1} \left(1 - \frac{\mu}{\eta}\left(\frac{\gamma-\eta}{\gamma-j}\right)\right) \geq \prod_{j=0}^{\eta-1} \left(1 - \frac{\mu}{\eta}\right) \\
&= \left(1 - \frac{\mu}{\eta}\right)^{\eta} = \left(\left(1 - \frac{\mu}{\eta}\right)^{\eta/\mu}\right)^{\mu} \geq \left(\frac{1}{2}\right)^{2\mu} \\
&\geq \left(\frac{1}{2}\right)^{(c-1)\log n} = \frac{1}{n^{c-1}}.
\end{aligned}$$

We then have the probability that all $v_i \neq v_i'$ as:

$$\begin{aligned}
\Pr[\forall i, v_i \neq v_i'] &= (1 - \Pr[v_i = v_i'])^{\ell} \\
&= \left(1 - \frac{1}{n^{c-1}}\right)^{\ell} = \left(\left(1 - \frac{1}{n^{c-1}}\right)^{n^{c-1}}\right)^{\ell/n^{c-1}} \\
&\leq \left(\frac{1}{e}\right)^{n^c/n^{c-1}} = \frac{1}{e^n}.
\end{aligned}$$

This completes the proof of Lemma 7.1.11. □

7.1.3 Reusability of Construction 7.1.1

The reusability of Construction 7.1.1 follows from the security of the VGB obfuscator with auxiliary input. We consider a bounded $q = \texttt{poly}(n)$ number of reuses. For some fixed $i \in \{1, ..., q\}$ we will treat the remaining keys as auxiliary input to the adversary, and the simulator still performs comparably to a distinguisher with access to the obfuscations. Thus, given sufficiently strong reusability we achieve the following result:

Theorem 7.1.12. *Let $q = \texttt{poly}(n)$, and let all the variables be as in Theorem 7.1.2, except that $\mathcal{O}$ be an $\ell \times q$-composable VGB obfuscator for digital lockers (with κ bit outputs) with auxiliary inputs. For any admissible $f_2, ..., f_q$, for all $s_{sec} = \texttt{poly}(n)$*

there exists some $\epsilon_{sec} = \mathtt{ngl}(n)$ *such that* $(\mathsf{Gen}, \mathsf{Rep})$ *is* $(q, \epsilon_{sec}, s_{sec}, f_2, ..., f_q)$*-reusable fuzzy extractor.*

Proof. The only modification to the proof is in Lemma 7.1.8 with the other keys $\mathsf{Key}_1, ..., \mathsf{Key}_{i-1}, \mathsf{Key}_{i+1}, ..., \mathsf{Key}_q$ treated as additional auxiliary input to the adversary/simulator. The simulator in the definition of composable obfuscation is required to function for arbitrary circuits in the family even if the choice of these circuits depends on the previous obfuscations. Thus allows reading w_i to be chosen depending on public values $p_1, ..., p_{i-1}$. □

More errors than entropy? We now show Construction 7.1.1 supports partial block sources with more errors than entropy. The structure of the partial block source implies that $\mathrm{H}_\infty(W) \geq \alpha(\gamma - \beta) = \Theta(\gamma)$. We assume that $\mathrm{H}_\infty(W) = \Theta(\gamma)$. We are able to correct $o(\gamma)$ errors. This yields:

$$\#\,\text{Errors} - \text{Entropy} = \log |B_t| - \mathrm{H}_\infty(W) \geq t \log |\mathcal{Z}| - \Theta(\gamma) = o(\gamma) \log |\mathcal{Z}| - \Theta(\gamma)$$

That is, there exists a super-constant alphabet size for which Construction 7.1.1 is secure with more errors than entropy.

Notes: Construction 7.1.1 works for an arbitrary size alphabet; however, for a constant size alphabet, the required entropy is greater than the number of corrected error patterns. However, Construction 7.1.1 is reusability for an arbitrary size alphabet.

In the analysis of Construction 7.1.1 we restricted our attention to partial block sources, to allow for an easy comparison with Construction 5.2.2. However, in fact Construction 7.1.1 is secure for any source where sampling produces a high entropy string (entropy $\omega(\log n)$) with overwhelming probability. For example, it is secure for sources with symbols that are $\omega(\log n)/\log |\mathcal{Z}|$-wise independent.

7.2 Allowing Correlated Symbols

In the previous section, we presented a reusable computational fuzzy extractor that supported sources with more errors than entropy. Unfortunately, both Constructions 5.2.2 and 7.1.1 required each symbol to contribute "fresh" entropy. In this section, we present a computational construction that allows for correlation between symbols while still supporting more errors than entropy and correcting a constant fraction of errors. This construction is inspired by the construction of digital lockers from point obfuscation by Canetti and Dakdouk [CD08]. Instead of having large parts of the string w unlock key, we have individual symbols unlock bits of the output. The construction that follows is a computational fuzzy conductor (Definition 3.3.7) not a computational fuzzy extractor (Definition 3.3.6, so we call its output c to distinguish from key.

Before presenting the construction we provide some definitions from error correcting codes. We use error-correct codes over $\{0,1\}^\gamma$ which correct up to t bit flips from 0 to 1 but no bit flips from 1 to 0 (this is the Hamming analog of the Z-channel [TABB02]).[3]

Definition 7.2.1. *Let $e, c \in \{0,1\}^\gamma$ be vectors. Let $x = \mathsf{Err}(c, e)$ be defined as follows*

$$x_i = \begin{cases} 1 & c_i = 1 \vee e_i = 1 \\ 0 & \textit{otherwise.} \end{cases}$$

Definition 7.2.2. *A set C (over $\{0,1\}^\gamma$) is a (t, δ_{code})-Z code if there exists an effi-*

[3] Any code that corrects t Hamming errors also corrects t $0 \to 1$ errors, but more efficient codes exist for this type of error [TABB02]. Codes with $2^{\Theta(\gamma)}$ codewords and $t = \Theta(\gamma)$ over the binary alphabet exist for Hamming errors and suffice for our purposes (first constructed by Justensen [Jus72]). These codes also yield a constant error tolerance for $0 \to 1$ bit flips. The class of errors we support in our source (t Hamming errors over a large alphabet) and the class of errors for which we need codes (t $0 \to 1$ errors) are different. Use of a code that corrects t Hamming errors gives the construction perfect correctness.

cient procedure Decode *such that*

$$\forall e \in \{0,1\}^\gamma | \mathsf{Wgt}(e) \leq t, \Pr_{c\in C}[\mathsf{Decode}(\mathsf{Err}(c,e)) \neq c] \leq \delta_{code}.$$

Construction 7.2.3. *Let $\mathcal{Z}$ be an alphabet and let $W = W_1, ..., W_\gamma$ be a distribution over $\mathcal{Z}^\gamma$. Let $\mathcal{O}$ be an obfuscator for point functions with points from $\mathcal{Z}$. Let $C \subset \{0,1\}^\gamma$ be an error-correcting code. We describe* Gen, Rep *as follows:*

```
Gen
  1. Input: $w = w_1, ..., w_\gamma$
  2. Sample $c \leftarrow C$.
  3. For $j = 1, ..., \gamma$:
       (i) If $c_j = 0$: $p_j = \mathcal{O}(I_{w_j})$.
       (ii) Else: $r_j \xleftarrow{\$} \mathcal{Z}$.
            Let $p_j = \mathcal{O}(I_{r_j})$.
  4. Output $(c, p)$, where $p = p_1 \dots p_\gamma$.
```

```
Rep
  1. Input: $(w', p)$
  2. For $j = 1, ..., \gamma$:
       (i) If $p_j(w'_j) = 1$: set $c'_j = 0$.
       (ii) Else: set $c'_j = 1$.
  3. Set $c = \mathsf{Decode}(c')$.
  4. Output $c$.
```

Construction 7.2.3 is secure if no distinguisher can tell whether it is working with random obfuscations or obfuscations of W_j. By the security of point obfuscation, anything learnable from the obfuscation is learnable from oracle access to the function. Therefore, our construction is secure as long as enough blocks are unpredictable even after adaptive queries to equality oracles for individual symbols. Definition 7.1.6 formalizes this intuition.

We show some examples of unguessable block sources in Appendix B. In particular, any source W where for all j, $\mathrm{H}_\infty(W_j) \geq \omega(\log n)$ (but all blocks may arbitrarily correlated) is an unguessable block source (Claim B.1.3).

Construction 7.2.3 is not a computational fuzzy extractor. The codewords c are not uniformly distributed and it is possible to learn some bits of c (for the symbols of W without much entropy). However, Construction 7.2.3 a computational fuzzy conductor (Definition 3.3.7). Computational fuzzy conductors can be converted to computational fuzzy extractors using standard techniques (Lemma 3.3.8).

Theorem 7.2.4. *Let n be a security parameter. Let $\mathcal{Z}$ be an alphabet where $|\mathcal{Z}| \geq 2^{\omega(\log(n))}$. Let $\mathcal{W}$ be a family of $(q, \alpha = \omega(\log n), \beta)$-unguessable block sources over $\mathcal{Z}^\gamma$, for any $q = \texttt{poly}(n)$. Furthermore, let C be a $(\mathsf{Neigh}_t, \delta_{code})$-code over $\mathcal{Z}^\gamma$. Let $\mathcal{O}$ be an γ-composable VGB obfuscator for point functions with auxiliary inputs. Then for any $s_{sec} = \texttt{poly}(n)$ there exists some $\epsilon_{sec} = \texttt{ngl}(n)$ such that Construction 7.2.3 is a $(\mathcal{Z}^\gamma, \mathcal{W}, \tilde{m} = H_0(C) - \beta, t)$-computational fuzzy conductor that is $(\epsilon_{sec}, s_{sec})$-hard with error $\delta_{code} + \gamma/|\mathcal{Z}|$.*

7.2.1 Security of Construction 7.2.3

Security of Construction 7.2.3 is similar to the security of Construction 7.1.1. However, security is more complicated, the main difficulty is that the definition of block unguessable sources (Definition 7.1.6) allows for weak blocks that can easily be guessed. This means we must limit our indistinguishable distribution to blocks that are difficult to guess. Security is proved via the following lemma:

Lemma 7.2.5. *Let all variables be as in Theorem 7.2.4. For every $s_{sec} = \texttt{poly}(n)$ there exists some $\epsilon_{sec} = \texttt{ngl}(n)$ such that $H^{\texttt{HILL}}_{\epsilon_{sec}, s_{sec}}(C|P) \geq H_0(C) - \beta$.*

We give a brief outline of the proof, followed by the proof. It is sufficient to show that there exists a distribution C' with conditional min-entropy and

$$\delta^{\mathcal{D}_{s_{sec}}}((C, P), (C', P)) \leq \texttt{ngl}(n).$$

Let J be the set of indices that exists according to Definition 7.1.6. Define the distribution C' as a uniform codeword conditioned on the values of C and C' being equal on all indices outside of J. We first note that C' has sufficient entropy, because $\tilde{\mathrm{H}}_\infty(C'|P) = \tilde{\mathrm{H}}_\infty(C'|C_{J^c}) \geq \mathrm{H}_\infty(C', C_{J^c}) - H_0(C_{J^c}) = H_0(C) - |J^c|$ (the second step is by [DORS08, Lemma 2.2b]). It is left to show $\delta^{\mathcal{D}_{s_{sec}}}((C, P), (C', P)) \leq \texttt{ngl}(n)$. The outline for the rest of the proof is as follows:

- Let D be a distinguisher between (C, P) and (C', P). Since P is a collection of obfuscated programs, there exists a simulator S (outputting a single bit), such that $\Pr[D(C, P) = 1]$ is close to $\Pr[S^{\mathcal{O}}(C) = 1]$.

- Show that even an unbounded S making a polynomial number of queries to the stored points cannot distinguish between C and C'. That is, $\mathbf{SD}(S^{\mathcal{O}}(C), S^{\mathcal{O}}(C'))$ is small.

- By the security of obfuscation, $\Pr[S^{\mathcal{O}}(C') = 1]$ is close to $\Pr[D(C', P) = 1]$.

Proof of Lemma 7.2.5. Let $\mathcal{O}$ be a γ-composable VGB obfuscator with auxiliary input for point programs over $\mathcal{Z}$. Let W be a $(q, \alpha = \omega(\log n), \beta)$-unguessable block source. Our goal is to show that for all $s_{sec} = \texttt{poly}(n)$ there exists $\epsilon_{sec} = \texttt{ngl}(n)$ such that $H^{\texttt{HILL}}_{\epsilon_{sec}, s_{sec}}(C|P) \geq H_0(C) - \beta$. Suppose not, that is suppose there is some $s_{sec} = \texttt{poly}(n)$ such that exists $\epsilon_{sec} = \texttt{poly}(n)$ and $H^{\texttt{HILL}}_{\epsilon_{sec}, s_{sec}}(C|P) < H_0(C) - \beta$. By Definition 7.1.6 there exists a set of indices J such that all blocks within J are unguessable. Define by C' the distribution of sampling a uniform codeword where all locations outside J are fixed. Then $\tilde{\mathrm{H}}_\infty(C'|C_{J^c}) \geq \mathrm{H}_\infty(C', C_{J^c}) - H_0(C_{J^c}) = H_0(C) - \beta$ (by [DORS08, Lemma 2.2b]).

Let D a distinguisher of size at most s_{sec} such that

$$|\,\mathbb{E}[D(C, P)] - \mathbb{E}[D(C', P)] > \epsilon_{sec} = 1/\texttt{poly}(n).$$

Define the distribution X as follows:

$$X_j = \begin{cases} W_j & C_j = 0 \\ R_j & C_j = 1. \end{cases}$$

By the security of obfuscation (Definition 2.4.1), there exists a unbounded time simulator S (making at most q queries) such that

$$|\,\mathbb{E}[D(P_1, ..., P_\gamma, C)] - \mathbb{E}[S^{I_X(\cdot,\cdot)}(C, 1^{\gamma \log |Z|})]| \leq \epsilon_{sec}/3. \tag{7.3}$$

We now prove S cannot distinguish between C and C'.

Lemma 7.2.6. $\mathbf{SD}(S^{I_X(\cdot,\cdot)}(C, 1^{\gamma \log |Z|}), S^{I_X(\cdot,\cdot)}(C', 1^{\gamma \log |Z|})) \leq (\gamma - \beta)2^{-(\alpha+1)}$.

Proof. It suffices to show that for any two codewords that agree on J^c, the statistical distance is at most $(\gamma-\beta)2^{-(\alpha+1)}$.

Lemma 7.2.7. *Let c^* be true value encoded in X and let c' a codeword in C'. Then,*

$$\mathbf{SD}(S^{I_X(\cdot,\cdot)}(c^*, 1^{\gamma\log|Z|}), S^{I_X(\cdot,\cdot)}(c', 1^{\gamma\log|Z|})) \leq (\gamma-\beta)2^{-(\alpha+1)}.$$

Proof. Recall that for all $j\in J$, $\tilde{\mathrm{H}}_\infty(W_j|View(S))\geq\alpha$. The only information about the correct value of c_j^* is contained in the query responses. When all responses are 0 the view of S is identical when presented with c^* or c'. We now show that for any value of c^* all queries on $j\in J$ return 0 with probability $1-2^{-\alpha+1}$. Suppose not, that is suppose, the probability of at least one nonzero response on index j is $>2^{-(\alpha+1)}$. Since w, w' are independent of r_j, the probability of this happening when $c_j^*=1$ is at most $q/\mathcal{Z}$ or equivalently $2^{-\log|\mathcal{Z}|+\log q}$. Thus, it must occur with probability:

$$\begin{aligned}2^{-\alpha+1} &< \Pr[\text{non zero response location } j]\\ &= \Pr[c_j^*=1]\Pr[\text{non zero response location } j\wedge c_j^*=1]\\ &+ \Pr[c_j^*=0]\Pr[\text{non zero response location } j\wedge c_j^*=0]\\ &\leq 1\times 2^{-\log|\mathcal{Z}|+\log q}+1\times\Pr[\text{non zero response location } j\wedge c_j^*=0] \qquad (7.4)\end{aligned}$$

We now show that for an unguessable block source the remaining entropy $\alpha\leq\log|\mathcal{Z}|-\log q$:

Claim 7.2.8. *If W is a (q,α,β)-block unguessable source over $\mathcal{Z}$ then $\alpha\leq\log|\mathcal{Z}|-\log q$.*

Proof. Let W be a (q,α,β)-block unguessable source. Let $J\subset\{1,...,\gamma\}$ the set of good indices. It suffices to show that there exists an S making q queries such that for some $j\in J, \tilde{\mathrm{H}}_\infty(W_j|S^{I_W(\cdot,\cdot)})\leq\log|\mathcal{Z}|-\log q$. Let $j\in J$ be some arbitrary element of J and denote by $w_{j,1},...,w_{j,q}$ the q most likely outcomes of W_j (breaking ties arbitrarily). Then $\sum_{i=1}^q\Pr[W_j=w_{j,i}]\geq q/|\mathcal{Z}|$. Suppose not, this means that there is some $w_{j,i}$ with probability $\Pr[W_j=w_{j,i}]<1/|\mathcal{Z}|$. Since there are $\mathcal{Z}-q$ remaining possible values of W_j for their total probability to be at least $1-q/|\mathcal{Z}|$ at least of these values has probability at least $1/\mathcal{Z}$. This contradicts the statement $w_{j,1},...,w_{j,q}$ are the most likely values. Consider S that queries its oracle on $(j,w_{j,1}),..,(j,w_{j,q})$. Denote by Bd the random variable when $W_j\in\{w_{j,1},..,w_{j,q}\}$ After these queries the remaining min-entropy is at most:

$$\begin{aligned}\tilde{\mathrm{H}}_\infty(W_j|S^{J_W(\cdot,\cdot)}) &= -\log\left(\Pr[Bd=1]\times 1 + \Pr[Bd=0]\times\max_w \Pr[W_j = w|Bd=0]\right)\\ &\le -\log\left(\Pr[Bd=1]\times 1\right)\\ &= -\log\left(\frac{q}{|\mathcal{Z}|}\right) = \log|\mathcal{Z}| - \log q\end{aligned}$$

This completes the proof of Claim 7.2.8. □

Rearranging terms in Equation 7.4, we have:

$$\Pr[\text{non zero response location } j \wedge c_j = 0] > 2^{-\alpha+1} - 2^{-(\log|\mathcal{Z}|-\log q)} = 2^{-\alpha}$$

When there is a 1 response and $c_j = 0$ this means that there is no remaining min-entropy. If this occurs with over $2^{-\alpha}$ probability this violates the block unguessability of W (Definition 7.1.6). By the union bound over the indices $j \in J$ the total probability of a 1 in J is at most $(\gamma-\beta)2^{-\alpha+1}$. Recall that c^*, c' match on all indices outside of J. Thus, for all c^*, c' the statistical distance is at most $(\gamma-\beta)2^{-\alpha+1}$. This concludes the proof of Lemma 7.2.7. □

By averaging over all points in C' we conclude that

$$\mathbf{SD}(S^{I_X(\cdot,\cdot)}(C, 1^{\gamma\log|Z|}), S^{I_X(\cdot,\cdot)}(C', 1^{\gamma\log|Z|})) < (\gamma-\beta)2^{-(\alpha+1)}.$$

This completes the proof of Lemma 7.2.6. □

Now by the security of obfuscation we have that

$$|\,\mathbb{E}[D(P_1,...,P_\gamma,C')] - \mathbb{E}[S^{I_X(\cdot,\cdot)}(C', 1^{\gamma\log|Z|})]| \le \epsilon_{sec}/3. \tag{7.5}$$

Combining Equations 7.3 and 7.5 and Lemma 7.2.6, we have

$$\begin{aligned}\delta^D((P,C),(P,C')) &\le |\,\mathbb{E}[D(P_1,...,P_\gamma,C)] - \mathbb{E}[S^{I_X(\cdot,\cdot)}(C,1^{\gamma\log|Z|})]|\\ &\quad + |\,\mathbb{E}[S^{I_X(\cdot,\cdot)}(C,1^{\gamma\log|Z|})] - \mathbb{E}[S^{I_X(\cdot,\cdot)}(C',1^{\gamma\log|Z|})]|\\ &\quad + |\,\mathbb{E}[S^{I_X(\cdot,\cdot)}(C',1^{\gamma\log|Z|})] - \mathbb{E}[D(P_1,...,P_\gamma,C')]|\\ &\le \epsilon_{sec}/3 + (\gamma-\beta)2^{-(\alpha-1)} + \epsilon_{sec}/3\\ &\le 2\epsilon_{sec}/3 + \mathtt{ngl}(n) < \epsilon_{sec}.\end{aligned}$$

This is a contradiction and completes the proof of Lemma 7.2.5. □

7.2.2 Correctness of Construction 7.2.3

We now argue correctness of Construction 7.2.3. We begin by showing that the probability of a single $1 \to 0$ bit flip in c is negligible.

Lemma 7.2.9. *Let all variables be as in Theorem 7.2.4. The probability of at least one $1 \to 0$ bit flip (an obfuscation of a random block being interpreted as the obfuscation of the point) is $\leq \gamma/|\mathcal{Z}| = \mathtt{ngl}(n)$.*

Proof. Consider a coordinate j for which $c_j = 1$. Since w' is chosen independently of the points r_j, and r_j is uniform, $\Pr[r_j = w'_j] = 1/|\mathcal{Z}|$. The lemma follows by the union bound, since there are at most γ such coordinates. □

Since there are most t locations for which $w_j \neq w'_j$ there are at most t $0 \to 1$ bit flips in c, which the code will correct with probability $1 - \delta_{code}$, because c is chosen independently of w'. Therefore, Construction 7.2.3 is correct with error at most $\gamma/|\mathcal{Z}|$.

More errors than entropy? In this section, we show that Construction 7.2.3 can support distributions with more errors than entropy. We first calculate the size of the Hamming ball.

$$\log|B_t| = \log\sum_{i=0}^{t}\binom{\gamma}{i}(|\mathcal{Z}|-1)^i > \log\binom{\gamma}{t}(|\mathcal{Z}|-1)^t = \Theta(t\log|\mathcal{Z}|) + \log\binom{\gamma}{t}$$

The simplest type of unguessable block source is where each block is independent and has super-logarithmic entropy (Claim B.1.1). For this type of source the entropy is $\mathrm{H}_\infty(W) = \gamma\omega(\log n)$. This yields:

$$\#\ \text{errors} - \text{entropy} = \log|B_t| - \mathrm{H}_\infty(W) > \left(\Theta(t\log|\mathcal{Z}|) + \log\binom{\gamma}{t}\right) - \gamma\omega(\log n).$$

When $t = \Theta(\gamma)$ and the entropy of each block is $o(\log|\mathcal{Z}|)$, then the construction supports more errors than entropy. Furthermore, the output entropy is $H_0(C) - \beta$ (if C is a constant rate code, this is $\Theta(\gamma)$).

Improvements If most codewords have Hamming weight close to $1/2$, we can decrease the error tolerance needed from the code from t to about $t/2$, because roughly half of the mismatches between w and w' occur where $c_j = 1$.

If γ is not long enough to get a sufficiently long output, the construction can be run multiple times with the same input and independent randomness.

Appendix A

A Definitional Equivalence

As described in Section 4.3, our negative results rule out security for an average member of $\mathcal{W}$. It may be possible to significantly improve parameters by only ruling out security for a single member W.

Recall the security game of a fuzzy extractor: 1) the challenger specifies $(\mathsf{SS}, \mathsf{Rec})$, 2) the adversary specifies a source $W \in \mathcal{W}$ 3) The challenger wins if $\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) \geq \tilde{m}$. Instead of just thinking of the uniform distribution over $\mathcal{W}$, consider an arbitrary distribution V over elements of $\mathcal{W}$. The minimax theorem says we can reverse which of these actions is announced first [VN28] if $\mathcal{A}$ announces V instead of a single element W. That is, the following two player games have the same equilibrium:

Experiment $\mathbf{Exp}_1^{\mathcal{W}}(\mathcal{A}, \mathcal{C}, \tilde{m})$:
$(\mathsf{SS}, \mathsf{Rec}) \leftarrow \mathcal{C}(\mathcal{W})$
$W \leftarrow \mathcal{A}(\mathcal{W}, \mathsf{SS}, \mathsf{Rec})$
If $W \notin \mathcal{W}$, $\mathcal{C}$ wins.
If $\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) \geq \tilde{m}$, $\mathcal{C}$ wins.
Else $\mathcal{A}$ wins.

Experiment $\mathbf{Exp}_2^{\mathcal{W}}(\mathcal{A}, \mathcal{C}, \tilde{m})$:
$V \leftarrow \mathcal{A}(\mathcal{W})$
$(\mathsf{SS}, \mathsf{Rec}) \leftarrow \mathcal{C}(V, \mathcal{W})$
$W \leftarrow V$
If $W \notin \mathcal{W}$, $\mathcal{C}$ wins.
If $\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) \geq \tilde{m}$, $\mathcal{C}$ wins.
Else $\mathcal{A}$ wins.

This means that showing security for a family of distributions $\mathcal{W}$ is equivalent to showing security for all distributions V when the distribution is known to the algorithms V. In our negative results, the adversary uses the uniform distribution V over $\mathcal{W}$. However, it may be possible to improve parameters by using a different V. This would just rule out some member of $\mathcal{W}$ not an average member. This is true for fuzzy extractors as well and is resilient to changes in parameters including imperfect correctness.

Appendix B

Characterizing unguessable block sources

Definition 7.1.6 is an inherently adaptive definition and a little unwieldy. In this section, we partially characterize sources that satisfy Definition 7.1.6. The majority of the difficulty in characterizing Definition 7.1.6 is that different blocks may be dependent, so an equality query on block i may reshape the distribution of block j. In the examples that follow we denote the adversary by S as we consider security against computationally unbounded adversaries defined in VGB obfuscation (Definition 2.4.1). We first show some sources that are unguessable block sources (Section B.1) and then show distributions with high overall entropy that are not unguessable block sources (Section B.2).

B.1 Positive Examples

We begin with the case of independent blocks.

Claim B.1.1. *Let $W = W_1, ..., W_\gamma$ be a source in which all blocks W_j are mutually independent. Let α be a parameter. Let $J \subset \{1, ..., \gamma\}$ be a set of indices such that for all $j \in J$, $\mathrm{H}_\infty(W_j) = \alpha$. Then for any q, W is a $(q, \alpha - \log(q + 1), \gamma - |J|)$-unguessable block source. In particular, when $\alpha = \omega(\log n)$ and $q = \mathtt{poly}(n)$, then W is a $(q, \omega(\log n), \gamma - |J|)$-unguessable block source.*

Proof. It suffices to show that for all $j \in J, \tilde{\mathrm{H}}_\infty(W_j|View(S^{I_W(\cdot,\cdot)}) = \alpha - \log(q + 1)$. We can ignore queries for all blocks but the j-th, as the blocks are independent. Furthermore, without loss of generality, we can assume that no duplicate queries are asked, and that the adversary is deterministic (S can calculate the best coins). Let $A_1, A_2, \ldots A_q$ be the random variables representing the oracle answers for an adversary S making q queries about the ith block. Each A_k is just a bit, and at most one of them is equal to 1 (because duplicate queries are disallowed). Thus, the total number of possible responses is $q+1$. Thus, we have the following,

$$\begin{aligned}\tilde{\mathrm{H}}_\infty(W_j|View(S^{\mathcal{O}_W(\cdot,\cdot)}) &= \tilde{\mathrm{H}}_\infty(W_j|A_1,\ldots,A_q)\\ &= \mathrm{H}_\infty(W_j) - |A_1,\ldots,A_q|\\ &= \alpha - \log(q+1)\,,\end{aligned}$$

where the second line follows from the first by [DORS08, Lemma 2.2]. □

Construction 6.1.1 is a computational fuzzy extractor for block fixing sources. Claim B.1.1 shows that unguessable block distributions are a superset of block fixing sources. We now consider more complicated distributions where blocks are not independent.

Claim B.1.2. *Let $f : \{0,1\}^e \rightarrow \mathcal{Z}^\gamma$ be a function. Furthermore, let f_j denote the restriction of f's output to its jth coordinate. If for all j, f_j is injective then $W = f(U_e)$ is a $(q, e-\log(q+1), 0)$-unguessable block source.*

Proof. Since f is injective on each block,

$$\tilde{\mathrm{H}}_\infty(W_j|View(S^{I_W(\cdot,\cdot)})) = \tilde{\mathrm{H}}_\infty(U_e|View(S^{I_W(\cdot,\cdot)})).$$

Consider a query q_k on block j. There are two possibilities: either q_k is not in the image of f_j, or q_k can be considered a query on the preimage $f_j^{-1}(q_k)$. Then (by assuming S knows f) we can eliminate queries which correspond to the same value of U_e. Then the possible responses are strings with Hamming weight at most 1 (like in the proof of Claim B.1.1), and by [DORS08, Lemma 2.2] we have for all j, $\tilde{\mathrm{H}}_\infty(W_j|View(S^{I_W(\cdot,\cdot)})) \geq \mathrm{H}_\infty(W_j) - \log(q+1)$. □

Note the total entropy of a source in Claim B.1.2 is e, so there is a family of distributions with total entropy $\omega(\log n)$ for which Construction 7.2.3 is secure. For these distributions, all the coordinates are as dependent as possible: one determines all others. We can prove a slightly weaker claim when the correlation between the coordinates W_j is arbitrary:

Claim B.1.3. *Let $W = W_1, ..., W_\gamma$ be a source. Suppose that for all j, $\mathrm{H}_\infty(W_j) \geq \alpha$, and that $q \leq 2^\alpha/4$ (this holds asymptotically, in particular, if q is polynomial and α is super-logarithmic). Then W is a $(q, \alpha-1-\log(q+1), 0)$-unguessable block source.*

Proof. Intuitively, the claim is true because the oracle is not likely to return 1 on any query. Formally, we proceed by induction on oracle queries, using the same notation as in the proof of Claim B.1.1. Our inductive hypothesis is that $\Pr[A_1 \neq 0 \vee \cdots \vee A_{k-1} \neq 0] \leq (k-1)2^{1-\alpha}$. If the inductive hypothesis holds, then, for each j,

$$\mathrm{H}_\infty(W_j | A_1 = \cdots = A_{k-1} = 0) \geq \alpha - 1\,. \tag{B.1}$$

This is true for $k = 1$ by the condition of the theorem. It is true for $k > 1$ because, as a consequence of the definition of H_∞, for any random variable X and event E, $\mathrm{H}_\infty(X|E) \geq \mathrm{H}_\infty(X) + \log \Pr[E]$; and $(k-1)2^{1-\alpha} \leq 2q2^{-\alpha} \leq 1/2$.

We now show that $\Pr[A_1 \neq 0 \vee \cdots \vee A_k \neq 0] \leq k2^{1-\alpha}$, assuming that $\Pr[A_1 \neq 0 \vee \cdots \vee A_{k-1} \neq 0] \leq (k-1)2^{1-\alpha}$.

$$\begin{aligned}
&\Pr[A_1 \neq 0 \vee \cdots \vee A_{k-1} \neq 0 \vee A_k \neq 0] \\
&\qquad = \Pr[A_1 \neq 0 \vee \cdots \vee A_{k-1} \neq 0] + \Pr[A_1 = \cdots = A_{k-1} = 0 \wedge A_k = 1] \\
&\qquad \leq (k-1)2^{1-\alpha} + \Pr[A_k = 1 \,|\, A_1 = \cdots = A_{k-1} = 0] \\
&\qquad \leq (k-1)2^{1-\alpha} + \max_j 2^{-\mathrm{H}_\infty(W_j | A_1 = \cdots = A_{k-1} = 0)} \\
&\qquad \leq (k-1)2^{1-\alpha} + 2^{1-\alpha} \\
&\qquad = k2^{1-\alpha}
\end{aligned}$$

(where the third line follows by considering that to get $A_k = 1$, the adversary needs to guess some W_j, and the fourth line follows by (B.1)). Thus, using $k = q + 1$ in (B.1), we know $\mathrm{H}_\infty(W_j | A_1 = \cdots = A_q = 0) \geq \alpha - 1$. Finally this means that

$$\begin{aligned}
\tilde{\mathrm{H}}_\infty(W_j | A_1, \ldots, A_q) &\geq -\log(2^{-\mathrm{H}_\infty(W_j | A_1 = \cdots = A_q = 0)} \Pr[A_1 = \cdots = A_q = 0] \\
&\qquad + 1 \cdot \Pr[A_1 \neq 0 \vee \cdots \vee A_q \neq 0]) \\
&\geq -\log\left(2^{-\mathrm{H}_\infty(W_j | A_1 = \cdots = A_q = 0)} + q2^{1-\alpha}\right) \\
&\geq -\log\left((q+1)2^{1-\alpha}\right) = \alpha - 1 - \log(q+1)\,.
\end{aligned}$$

□

B.2 Negative Examples

Claims B.1.2 and B.1.3 rest on there being no easy "entry" point to the distribution. This is not always the case. Indeed it is possible for some blocks to have very high entropy but lose all of it after equality queries.

Claim B.2.1. *Let* $p = (\texttt{poly}(n))$ *and let* $f_1, ..., f_\gamma$ *be injective functions where* $f_j : \{0,1\}^{j\times \log p} \rightarrow \{0,1\}^n$.[1] *Then define the distributions*

$$\begin{aligned} W_1 &= f_1(U_{1,...,\gamma}),\\ W_2 &= f_2(U_{1,...,2\gamma})\\ &,...,\\ W_\gamma &= f_\gamma(U). \end{aligned}$$

There is an adversary making $p \times \gamma = \texttt{poly}(n)$ *queries such that*

$$\tilde{\mathrm{H}}_\infty(W|View(S^{I_W(\cdot,\cdot)})) = 0.$$

Proof. Let x be the true value for $U_{p\times\gamma}$. We present an adversary S that completely determines x. S computes $y_1^1 = f_1(x_1^1), ..., y_1^p = f(x_1^p)$. Then S queries on $(1, y_1), ..., (1, y_p)$, exactly one answer returns 1. Let this value be y_1^* and its preimage x_1^*. Then S computes $y_2^1 = f_2(x_1^*, x_2^1), ..., y_2^p = f_2(x_1^*, x_2^p)$ and queries $y_2^1, ..., y_2^p$. Again, exactly one of these queries returns 1. This process is repeated until all of x is recovered (and thus w). □

The previous example relies on an adversaries ability to determine a block from the previous blocks. We formalize this notion next. We define the entropy jump of a block source as the remaining entropy when other blocks are known:

Definition B.2.2. *Let* $W = W_1, ..., W_\gamma$ *be a source under ordering* $i_1, ..., i_\gamma$. *The* jump *of a block* i_j *is* $\texttt{Jump}(i_j) = \max_{w_{i_1},...,w_{i_{j-1}}} H_0(W_{i_j}|W_{i_1} = w_{i_1}, ..., W_{i_{j-1}} = w_{i_{j-1}})$.

If an adversary can learn blocks in succession they can eventually recover the entire secret. In order for a source to be block unguessable the adversary must get "stuck"

[1] Here we assume that $n \geq \gamma \times \log p$, that is the source has a small number of blocks.

early enough in their recovery process. This translates to having a super-logarithmic jump early enough.

Claim B.2.3. *Let W be a distribution and let q be a parameter, if there exists an ordering $i_1, ..., i_\gamma$ such that for all $j \leq \gamma - \beta + 1$, $\texttt{Jump}(i_j) = \log q/(\gamma - \beta + 1)$, then W is not $(q, 0, \beta)$-unguessable block source.*

Proof. For convenience relabel the ordering that violates the condition as $1, ..., \gamma$. We describe an unbounded adversary that determines $W_1, ..., W_{\gamma-\beta+1}$. As before S queries the q/γ possible values for W_1 and determines W_1. Then S queries the (at most) $q/(\gamma - \beta + 1)$ possible values for $W_2|W_1$. This process is repeated until $W_{\gamma-\beta+1}$ is learned. □

Presenting a sufficient condition for security is more difficult as S may interleave queries to different blocks. It seems like the optimum strategy is to focus on a single block at a time but it is unclear how to formalize this intuition.

References

[AC93] Rudolf Ahlswede and Imre Csiszar. Common randomness in information theory and cryptography. part I: secret sharing. *IEEE Transactions on Information Theory*, 39(4), 1993.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *Theory of Cryptography*, pages 474–495. Springer Berlin Heidelberg, 2009.

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. On pseudorandom generators with linear stretch in NC 0. *Approximation, Randomization, and Combinatorial Optimization*, pages 260–271, 2006.

[BA12] Marina Blanton and Mehrdad Aliasgari. On the (non-) reusability of fuzzy sketches and extractors and security improvements in the computational setting. *IACR Cryptology ePrint Archive*, 2012:608, 2012.

[BA13] Marina Blanton and Mehrdad Aliasgari. Analysis of reusability of secure sketches and fuzzy extractors. *IEEE Transactions on Information Forensics and Security*, 8(9-10):1433–1445, 2013.

[BBR88] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[BC10] Nir Bitansky and Ran Canetti. On strong simulation and composable point obfuscation. In *Advances in Cryptology–CRYPTO*, pages 520–537. Springer, 2010.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[BCKP14] Nir Bitansky, Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On virtual grey box obfuscation for general circuits. In *Advances in Cryptology–CRYPTO*, pages 108–125. Springer, 2014.

[BCL$^+$11] Boaz Barak, Ran Canetti, Yehuda Lindell, Rafael Pass, and Tal Rabin. Secure computation without authentication. *Journal of Cryptology*, pages 720–760, 2011.

[BDH$^+$10] Ileana Buhan, Jeroen Doumen, Pieter Hartel, Qian Tang, and Raymond Veldhuis. Embedding renewable cryptographic keys into noisy data. *International Journal of Information Security*, 9(3):193–208, 2010.

[BDK$^+$05] Xavier Boyen, Yevgeniy Dodis, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Secure remote authentication using biometric data. In *Advances in Cryptology–EUROCRYPT*, pages 147–163. Springer, 2005.

[BDK$^+$11] Boaz Barak, Yevgeniy Dodis, Hugo Krawczyk, Olivier Pereira, Krzysztof Pietrzak, François-Xavier Standaert, and Yu Yu. Leftover hash lemma, revisited. In *Advances in Cryptology–CRYPTO*, pages 1–20. Springer, 2011.

[BGI$^+$01] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. In *Advances in Cryptology–CRYPTO*, pages 1–18. Springer, 2001.

[BH09] Marina Blanton and William MP Hudelson. Biometric-based non-transferable anonymous credentials. In *Information and Communications Security*, pages 165–180. Springer, 2009.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003.

[BLP$^+$13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Symposium on Theory of Computing*, pages 575–584. ACM, 2013.

[BM14] Christina Brzuska and Arno Mittelbach. Indistinguishability obfuscation versus multi-bit point obfuscation with auxiliary input. In *Advances in Cryptology–ASIACRYPT*, pages 142–161. Springer, 2014.

[BMvT78] Elwyn Berlekamp, Robert McEliece, and Henk van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384 – 386, May 1978.

[Boy04] Xavier Boyen. Reusable cryptographic fuzzy extractors. In *ACM Conference on Computer and Communications Security*, pages 82–91, New York, NY, USA, 2004. ACM.

[BS94] Gilles Brassard and Louis Salvail. Secret-key reconciliation by public discussion. In *Advances in Cryptology–EUROCRYPT*, pages 410–423. Springer, 1994.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *Advances in Cryptology–CRYPTO*, pages 455–469. Springer, 1997.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Obfuscating point functions with multibit output. In *Advances in Cryptology–EUROCRYPT*, pages 489–508. Springer, 2008.

[CFP$^+$14] Ran Canetti, Benjamin Fuller, Omer Paneth, Leonid Reyzin, and Adam Smith. Key derivation from noisy sources with more errors than entropy. *IACR Cryptology ePrint Archive*, 2014:243, 2014.

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

[Chv79] Vašek Chvátal. The tail of the hypergeometric distribution. *Discrete Mathematics*, 25(3):285–287, 1979.

[CK78] Imre Csiszár and Janos Korner. Broadcast channels with confidential messages. *IEEE Transactions on Information Theory*, 24(3):339–348, 1978.

[CK03] L. Csirmaz and G.O.H. Katona. Geometrical cryptography. In *International Workshop on Coding and Cryptography*, pages 101–109, 2003.

[CLRS01] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to Algorithms*, volume 2. MIT press Cambridge, 2001.

[Coo00] Colin Cooper. On the rank of random matrices. *Random Structures & Algorithms*, 16(2):209–232, 2000.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley-InterScience, 2nd edition, 2006.

[CW79] Larry Carter and Mark N. Wegman. Universal classes of hash functions. *Computer and System Sciences*, 18(2):143–154, 1979.

[CZC04] Yao-Jen Chang, Wende Zhang, and Tsuhan Chen. Biometrics-based cryptographic key generation. In *IEEE International Conference on Multimedia and Expo*, volume 3, pages 2203–2206. IEEE, 2004.

[Dau04] John Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21 – 30, January 2004.

[Dau06] John Daugman. Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparisons. *Proceedings of the IEEE*, 94(11):1927–1935, 2006.

[DKRS06] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In *Advances in Cryptology–CRYPTO*, pages 232–250. Springer, 2006.

[DMQ13] Nico Döttling and Jörn Müller-Quade. Lossy codes and a new variant of the learning-with-errors problem. In *Advances in Cryptology–EUROCRYPT*, pages 18–34, 2013.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

[DPW14] Yevgeniy Dodis, Krzysztof Pietrzak, and Daniel Wichs. Key derivation without entropy waste. In *Advances in Cryptology–EUROCRYPT*, pages 93–110. Springer, 2014.

[DSGKMk12] Dana Dachman-Soled, Rosario Gennaro, Hugo Krawczyk, and Tal Malkin. Computational extractors and pseudorandomness. In *Theory of Cryptography*, pages 383–403. Springer, 2012.

[FF81] Peter Frankl and Zoltán Füredi. A short proof for a theorem of Harper about Hamming-spheres. *Discrete Mathematics*, 34(3):311–313, 1981.

[FMR13] Benjamin Fuller, Xianrui Meng, and Leonid Reyzin. Computational fuzzy extractors. In *Advances in Cryptology–ASIACRYPT*, pages 174–193. Springer, 2013.

[FR12] Benjamin Fuller and Leonid Reyzin. Computational entropy and information leakage. *IACR Cryptology ePrint Archive*, 2012:466, 2012.

[FRS14] Benjamin Fuller, Leonid Reyzin, and Adam Smith. When are fuzzy extractors possible? *IACR Cryptology ePrint Archive*, 2014:961, 2014.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Symposium on Theory of Computing*, pages 25–32. ACM, 1989.

[Gol97] Oded Goldreich. A sample of samplers: A computational perspective on sampling. In *Electronic Colloquium on Computational Complexity*. Springer, 1997.

[GRGB$^+$12] Javier Galbally, Arun Ross, Marta Gomez-Barrero, Julian Fierrez, and Javier Ortega-Garcia. From the iriscode to the iris: A new vulnerability of iris recognition systems. *Black Hat*, 2012. Available at `https://media.blackhat.com/bh-us-12/Briefings/Galbally/BH_US_12_Galbally_Iris_Reconstruction_WP.pdf`.

[Gur10] Venkatesan Guruswami. Introduction to coding theory - lecture 2: Gilbert-Varshamov bound. University Lecture, 2010.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *Symposium on Theory of Computation*, pages 99–108. ACM, ACM, 2011.

[HAD06] Feng Hao, Ross Anderson, and John Daugman. Combining crypto with biometrics effectively. *IEEE Transactions on Computers*, 55(9):1081–1088, 2006.

[Har66] Lawrence H Harper. Optimal numberings and isoperimetric problems on graphs. *Journal of Combinatorial Theory*, 1(3):385–393, 1966.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HLR07] Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In *Advances in Cryptology–EUROCRYPT*, pages 169–186, 2007.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[HRvD$^+$14] Charles Herder, Ling Ren, Marten van Dijk, Meng-Day Mandel Yu, and Srinivas Devadas. Trapdoor computational fuzzy extractors. *IACR Cryptology ePrint Archive*, 2014:938, 2014.

[IW12] Tanya Ignatenko and Frans M.J. Willems. Biometric security from an information-theoretical perspective. *Foundations and Trends in Communications and Information Theory*, 7(2–3):135–316, 2012.

[Jus72] Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18(5):652–656, 1972.

[JW99] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Sixth ACM Conference on Computer and Communication Security*, pages 28–36. ACM, November 1999.

[KH11] Jae-Jung Kim and Seng-Phil Hong. A method of risk assessment for multi-factor authentication. *Journal of Information Processing Systems*, 7(1):187–198, 2011.

[KR09] Bhavana Kanukurthi and Leonid Reyzin. Key agreement from close secrets over unsecured channels. In *Advances in Cryptology–EUROCRYPT*, pages 206–223. Springer, 2009.

[Kra10] Hugo Krawczyk. Cryptographic extraction and key derivation: The HKDF scheme. In *Advances in Cryptology–CRYPTO*, pages 631–648. Springer, 2010.

[KZ07] Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM Journal on Computing*, 36(5):1231–1247, 2007.

[LC06] Qiming Li and Ee-Chien Chang. Robust, short and sensitive authentication tags using secure sketch. In *ACM Multimedia Security Workshop*, 2006.

[LT03] Jean-Paul Linnartz and Pim Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *Audio- and Video-Based Biometric Person Authentication*, pages 393–402. Springer, 2003.

[Lys] Alexander Lystad. The password project. Available at `http://thepassword-project.com/leaked_password_lists_and_dictionaries`.

[Mau93] Ueli M Maurer. Secret key agreement by public discussion from common information. *IEEE Transactions on Information Theory*, 39(3):733–742, 1993.

[MP13] Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with small parameters. In *Advances in Cryptology–CRYPTO*, pages 21–39. Springer, 2013.

[MRW02] Fabian Monrose, Michael K Reiter, and Susanne Wetzel. Password hardening based on keystroke dynamics. *International Journal of Information Security*, 1(2):69–83, 2002.

[NZ93] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52:43–52, 1993.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Symposium on Theory of Computing*, pages 333–342, New York, NY, USA, 2009. ACM.

[PRTG02] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.

[PST14] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multi-linear encodings. In *Advances in Cryptology–CRYPTO*, pages 500–517. Springer, 2014.

[RCB03] Nalini K Ratha, Jonathan H Connell, and Ruud M Bolle. Biometrics break-ins and band-aids. *Pattern Recognition Letters*, 24(13):2105–2113, 2003.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Symposium on Theory of Computing*, pages 84–93, New York, NY, USA, 2005. ACM.

[Reg10] Oded Regev. The learning with errors problem (invited survey). *Annual IEEE Conference on Computational Complexity*, pages 191–204, 2010.

[Rey11] Leonid Reyzin. Some notions of entropy for cryptography. In *Information Theoretic Security*, pages 138–142. Springer, 2011.

[RW05] Renato Renner and Stefan Wolf. Simple and tight bounds for information reconciliation and privacy amplification. In *Advances in Cryptology–ASIACRYPT*, pages 199–216. Springer, 2005.

[SD07] G Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th Annual Design Automation Conference*, pages 9–14. ACM, 2007.

[Sip12] Michael Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 2012.

[Ska13] Matthew Skala. Hypergeometric tail inequalities: ending the insanity. *arXiv*, 1311.5939, 2013.

[ŠTGP09] Boris Škorić, Pim Tuyls, Jorge Guajardo, and Bart Preneel. An efficient fuzzy extractor for limited noise. *IACR Cryptology ePrint Archive*, 2009:30, 2009.

[STP09] Koen Simoens, Pim Tuyls, and Bart Preneel. Privacy weaknesses in biometric sketches. In *IEEE Symposium on Security and Privacy*, pages 188–203. IEEE, 2009.

[SWBH49] Claude E. Shannon, Warren Weaver, Richard E. Blahut, and Bruce Hajek. *The Mathematical Theory of Communication*, volume 117. University of Illinois press Urbana, 1949.

[TABB02] Luca G Tallini, Sulaiman Al-Bassam, and Bella Bose. On the capacity and codes for the Z-channel. In *IEEE International Symposium on Information Theory*, page 422, 2002.

[TG04] Pim Tuyls and Jasper Goseling. Capacity and examples of template-protecting biometric authentication systems. In *Biometric Authentication*, pages 158–170, 2004.

[TSŠ$^+$06] Pim Tuyls, Geert-Jan Schrijen, Boris Škorić, Jan Van Geloven, Nynke Verhaegh, and Rob Wolters. Read-proof hardware from protective coatings. In *Cryptographic Hardware and Embedded Systems*, pages 369–383. Springer, 2006.

[TW14] Himanshu Tyagi and Shun Watanabe. A bound for multiparty secret key agreement and implications for a problem of secure computing. In *Advances in Cryptology–EUROCRYPT*, pages 369–386. Springer, 2014.

[Vad03] Salil P Vadhan. On constructing locally computable extractors and cryptosystems in the bounded storage model. In *Advances in Cryptology–CRYPTO*, pages 61–77. Springer, 2003.

[VN28] J. Von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928.

[VTO$^+$10] Evgeny A. Verbitskiy, Pim Tuyls, Chibuzo Obi, Berry Schoenmakers, and Boris Škorić. Key extraction from general nondiscrete signals. *IEEE Transactions on Information Forensics and Security*, 5(2):269–279, 2010.

[WACS10] Matt Weir, Sudhir Aggarwal, Michael Collins, and Henry Stern. Testing metrics for password creation policies by attacking large sets of revealed passwords. In *ACM Conference on Computer and Communications Security*, pages 162–175. ACM, 2010.

[WRDI12] Ye Wang, Shantanu Rane, Stark C Draper, and Prakash Ishwar. A theoretical analysis of authentication, privacy, and reusability across secure biometric systems. *IEEE Transactions on Information Forensics and Security*, 7(6):1825–1840, 2012.

[Wyn75] Aaron D Wyner. The wire-tap channel. *The Bell System Technical Journal*, 54(8):1355–1387, 1975.

[YBAG04] Jeff Jianxin Yan, Alan F Blackwell, Ross J Anderson, and Alasdair Grant. Password memorability and security: Empirical results. *IEEE Symposium on Security and Privacy*, 2(5):25–31, 2004.

Curriculum Vitae

Personal Information

email	bfuller@cs.bu.edu
website	http:/cs-people.bu.edu/bfuller
phone	(M) +1 (802) 999 0763

address
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA, 02421

Work Experience

2007- Research Scientist, MIT Lincoln Laboratory

Performed research at the intersection of cryptography and secure systems.

- **Secure and Resilient Cloud**. Important computations increasingly occur in a cloud environment. It is imprudent to assume that all cloud resources operate honestly. We evaluated the applicability of multi-party computation to the cloud environment. In particular, we considered how to build multi-party computation techniques that allow a sparse communication network.
- **Secure Cloud Authentication**. User's data is increasingly pushed to resources they do not control. Strong authentication is even more important in the cloud environment. The human iris is a potential authentication source. We researched image processing techniques and key derivation techniques to improve iris authentication.
- **Physical Unclonable Functions**. A strong root-of-trust is critical to securing hardware devices. Physical unclonable functions are one source for a root-of-trust. We developed a optical physical unclonable function and accompanying algorithms.
- **Dynamic Group Key Management**. Key management is a challenge in real-world cryptographic applications. Standard approaches use static keys and assume a fixed set of participants. We developed and deployed a dynamic key management system.

2006 Intern, National Security Agency

Applied mathematical principles to real-life cryptographic problems and protocols.

2005 Intern, International Business Machines

Collected worldwide inventory aging information, and automated process to make business recommendations about assets and reserves.

Education

2011- Boston University

Doctor of Philosophy: GPA: 3.96 · Department: Computer Science
Advisor: Assoc. Prof. Leonid Reyzin
Dissertation: Strong Key Derivation from Noisy Sources
Readers: Assoc. Prof. Leonid Reyzin & Prof. Ran Canetti & Assist. Prof. Daniel Wichs
Description: A shared cryptographic key enables strong authentication. Candidate sources for creating such a shared key include biometrics and physically unclonable functions. However, these sources come with a substantial problem: noise in repeated readings. A fuzzy extractor produces a stable key from a noisy source. For many sources of practical importance, traditional fuzzy extractors provide no meaningful security guarantee. This dissertation improves fuzzy extractors. First, we show how to incorporate structural information about the physical source to facilitate key derivation. Second, most fuzzy extractors work by first recovering the initial reading from the noisy reading. We improve key derivation by producing a consistent key without recovering the original reading. Third, traditional fuzzy extractors provide information-theoretic security. We build fuzzy extractors achieving new properties by only providing security against computational bounded adversaries.
Awards: Computer Science Research Excellence Award 2014

2009-2010 Boston University

Masters of Arts: GPA: 3.93 · Department: Computer Science
Thesis: *Computational Entropy and Information Leakage*
Readers: Assoc. Prof. Leonid Reyzin & Prof. Peter Gacs
Description: We investigate how information leakage reduces computational entropy of a random variable. We prove an intuitively natural result: conditioning on an event of probability p reduces the quality of computational entropy by a factor of p and the quantity of metric entropy by log 1/p. Our result improves previous bounds of Dziembowski and Pietrzak (FOCS 2008), where the loss in the quantity of entropy was related to its original quality. Our result also simplifies the result of Reingold et al. (FOCS 2008).

2003-2006 Rensselaer Polytechnic Institute

Bachelor of Science: GPA: 4.00 · Departments: Mathematics and Computer Science
Description: Dual degree in mathematics and computer science with a focus on pure mathematics and number theory.
Awards: Rensselaer Medal Winner · Computer Science Scholars Award

Technical Papers

Papers in Preparation

Unifying Leakage Classes: Simulatable Leakage and Pseudoentropy. Benjamin Fuller and Ariel Hamlin. Submitted November 2014.

When are Fuzzy Extractors Possible? Benjamin Fuller, Leonid Reyzin, and Adam Smith. Submitted October 2014.

Key Derivation from Noisy Sources with More Errors than Entropy. Ran Canetti, Benjamin Fuller, Omer Paneth, Leonid Reyzin, and Adam Smith. Submitted September 2014.

Cryptography Publications

Computational Fuzzy Extractors. *Advances in Cryptology – Asiacrypt.* Benjamin Fuller, Leonid Reyzin, and Xianrui Meng. December 2013.

A Unified Approach to Deterministic Encryption – New Constructions and a Connection to Computational Entropy. *Journal of Cryptology.* Benjamin Fuller, Adam O'Neil, and Leonid Reyzin. December 2013.
An earlier version of this work appeared at Theory of Cryptography 2012. The journal version contains significant new material.

Systems Security Publications

Robust Keys from Physical Unclonable Functions. *IEEE Symposium on Hardware Oriented Security and Trust 2014.* Merrielle Spain, Benjamin Fuller, Kyle Ingols, and Robert Cunningham

DSKE: Dynamic Set Key Encryption. *LCN Workshop on Security in Communication Networks 2012.* Galen Pickard, Roger Khazan, Benjamin Fuller, and Joseph Cooley

ASE: Authenticated Statement Exchange. *IEEE Network Computing and Applications 2010.* Benjamin Fuller, Roger Khazan, Joseph Cooley, and Galen Pickard. **Best Paper Award.**

GROK: A Practical System for Securing Group Communications. *IEEE Network Computing and Applications 2010.* Joseph Cooley, Roger Khazan, Benjamin Fuller, and Galen Pickard. **Best Paper Nominee.**

GROK Secure Multi-User Chat at Red Flag 2007-03. *Military Communications Conference 2008.* Roger Khazan, Joseph Cooley, Galen Pickard, and Benjamin Fuller.

Integrated Environment Management for Informaiton Operations Testbeds.. *Workshop on Visualization for Computer Security 2007.* Tamara Yu, Benjamin Fuller, John Bannick, Lee Rossey, and Robert Cunningham.

Unpublished Works

Computational Entropy and Information Leakage, 2011. Benjamin Fuller and Leonid Reyzin

Teaching

Teaching Assistant for Introduction to Network Security 2013.
Professor: Ran Canetti.

Teaching Assistant for Introduction to Cryptography 2012.
Professor: Leonid Reyzin.

Teaching Assistant for Computer Architecture 2006.
Professor: Franklin Luk.

Basic Skills Assistant for Calculus 2006.
Professor: Bruce Piper.

Teaching Assistant for Computer Organization 2006.
Professor: Franklin Luk.