Diss. ETH No. 16619

Strengthening Key Agreement using Hard-Core Sets

A dissertation submitted to

ETH ZURICH

for the degree of
Doctor of Sciences

presented by

Thomas Holenstein
Dipl. Inf. Ing. ETH

born June 14, 1976, in Zürich
citizen of Fischingen, TG

accepted on the recommendation of

Prof. Dr. Ueli Maurer, examiner

Prof. Dr. Johan Håstad, co-examiner

2006

Acknowledgments

First, I would like to thank Ueli Maurer, my advisor. Not only did he propose an excellent topic for reasearch, he also guided me through my time as a PhD student. He always asks excellent questions, and answering them often leads to many new insights. Also, I want to thank Johan Håstad many times for co-refereeing this thesis.

I also want to thank all the people which were part of the information security and cryptography research group or the quantum information group at ETH Zürich during my stay. This includes Georges Baatz, Zuzana Beerliova, Stefan Dziembowski, Thomas Dübendorfer, Marc Fischlin, Matthias Fitzi, Martin Hirt, Dominic Jost, Robert König, Lennart Meier, Jesper Buus Nielsen, Krzysztof Pietrzak, Bartosz Przydatek, Dominik Raub, Renato Renner, Johan Sjödin, Stefano Tessaro, Michèle Wigger, Douglas Wikström, Stefan Wolf, Jürg Wullschleger, and Vassilis Zikas.

Special thanks go to Renato Renner who was my officemate for most of my stay at ETH. Much of the research in this thesis originated from discussions with him or was done in collaboration with him.

I would also like to thank my diploma students Johan Sjödin, Pierre Bajgrowicz, and Georges Baatz. Working with them was a source of new inspirations.

Many thanks go to Beate Bernhard and Hans Dubach, who ensured that I needed almost no time for administrative tasks. Such help is always greatly welcome.

This research was partially supported by the Swiss National Science Foundation (SNF).

Abstract

Given an authentic communication channel, a key agreement protocol enables two parties to obtain a common bit string (the key), such that an eavesdropper does not have any information about it, even if he observes the whole communication. While no such protocol is secure in an information theoretic sense, it seems possible to give a key agreement protocol which is secure against eavesdroppers which do not have exceedingly large computational power. In fact, many protocols which promise to achieve such computational security are used in practice today. This holds even though no such protocol has been proven secure. Instead, the security of such a protocol is based on an unproven, but plausible assumption.

The goal of this thesis is to construct a computationally secure key agreement protocol whose security is based on an assumption which is as weak as possible. The assumption we use is the existence of a "weak key agreement protocol". Such a protocol works *partially*: in some executions the honest parties get the same key, but sometimes their respective keys differ. Furthermore, in some cases the resulting key is secret, while sometimes information about the key is leaked to an eavesdropper. We then strengthen such a protocol; i.e., we make it both secret and correct. In order to simplify the study, we restrict the given weak key agreement protocol to yield a single key bit.

To strengthen a weak key agreement protocol, we proceed in two steps. In a first step, we solve a related, completely information theoretic problem. More concretely we assume that some trusted source distributes random variables to the honest parties and to an eavesdropper according to a fixed and commonly known distribution. We then study whether the honest parties can use this randomness in order to obtain an information theoretically secure key. Such information theoretic key agreement from correlated information is a problem which has been studied before. It is interesting in its own right, and we look at it in some depth.

In a second step we show that certain protocols for the information theoretic setting we described can be used in the computational setting as well. Thus, we first use the weak key agreement protocol to obtain

random variables with certain computational security. We then use these random variables in a protocol designed for information theoretic security. We will see that for certain protocols the resulting key is computationally secure.

The two step process has many advantages. It greatly simplifies the constructions as well as the security proofs, since most of the work can be done in the easier information theoretic setting. It is also very intuitive, and it allows us to give constructions which work for optimal parameters.

In order to show that this two step process is possible, we use a powerful lemma about hard-core sets. Roughly speaking, the lemma shows that any computational problem which is mildly hard has a set of instances for which it is very hard. In our setting this implies that for a weak key agreement protocol as given, if the randomness of Alice and Bob is restricted to a certain subset, finding the key given the communication is a very hard problem. Such a lemma has been known before, and it has found various applications in theoretical computer science. In this thesis we improve on the known result in two ways. First, we increase the size of the hard set to the maximum possible. Further we give a variant which can be applied in the usual *uniform* setting (where the adversary is modeled as an algorithm). Previously, only a lemma applicable in the *non-uniform* setting (where the adversary is modeled using circuits) was known.

Zusammenfassung

Ein Schlüsselvereinbarungsverfahren erlaubt zwei Parteien eine geheime Bitfolge (einen sogenannten Schlüssel) zu erzeugen, falls ihnen ein authentischer Kanal zur Verfügung steht. Falls kein zusätzliches Hilfsmittel (wie zum Beispiel ein Kanal für einzelne Photonen) zur Verfügung steht, ist jedes Protokoll für diesen Zweck informationstheoretisch gesehen unsicher. Trotzdem scheint Schlüsselvereinbarung auch in diesem Fall möglich zu sein, falls Sicherheit nur gegen Gegner mit beschränkter Rechenzeit nötig ist. Tatsächlich werden in der Praxis verschiedene solche Verfahren verwendet. Unglücklicherweise kennt man für kein solches Verfahren einen Beweis für die Sicherheit. Diese beruht auf einer unbewiesenen (aber üblicherweise vernünftigen) Annahme.

Das Ziel dieser Arbeit ist die Konstruktion von einem Schlüsselvereinbarungsverfahren, basierend auf einer möglichst schwachen Annahme. Wir werden annehmen, dass ein solches Verfahren existiert, welches aber nur *teilweise* funktioniert: in einigen Ausführungen werden die ehrlichen Parteien unterschiedliche Schlüssel erhalten, und in anderen kann ein Gegner Information über den Schlüssel aus der Kommunikation folgern. Wir demonstrieren dann wie man aus einem solchen Protokoll ein sicheres erzeugt. Um dies zu erleichtern, beschränken wir uns auf den Fall in welchem das gegebene Verfahren einzelne Bits produziert.

Wir werden in zwei Schritten vorgehen. In einem ersten Schritt lösen wir ein verwandtes informationstheoretisches Problem. In diesem erhalten die ehrlichen Parteien und der Gegner korrelierte Information, wobei die Art der Korrelation allen bekannt ist. Wir studieren dann die Frage in welchen Fällen solche zusätzliche Information informationstheoretisch sichere Schlüsselvereinbarung erlaubt. Dieses Problem wurde schon in früheren Arbeiten intensiv studiert und ist auch ohne unsere ursprüngliche Motivation interessant; wir werden es deshalb eine Weile lang untersuchen.

In einem zweiten Teil werden wir zeigen dass wir gewisse Lösungen vom ersten Teil auch verwenden können, um unser ursprüngliches Problem zu lösen. Genauer: wir verwenden das gegebene teilweise funktionierende Protokoll und erzeugen damit Zufallsvariablen welche eine ge-

wisse Sicherheit gegen Gegner mit beschränkter Rechenzeit bieten. Wir verwenden diese dann in einem Protokoll für ähnliche, aber informationstheoretische sichere Zufallsvariablen, und zeigen dass der resultierende Schlüssel berechenmässige Sicherheit haben wird.

Dieser zweiteilige Beweis hat viele Vorteile. So ist er wesentlich einfacher als ähnliche bislang bekannte Beweise, und dazu auch noch recht intuitiv. Weiters erlaubt uns die Zweiteilung Konstruktionen welche für den grösstmöglichen Bereich von Parametern funktionieren.

Um zu zeigen dass diese Zweiteilung tatsächlich funktioniert verwenden wir ein mächtiges Lemma über harte Teilmengen. Salopp gesagt zeigt dieses Lemma dass jedes berechenmässig mittelschwere Problem einen "harten Kern" von Instanzen hat, auf welchem das Problem *sehr schwer* ist. In obigem Szenario impliziert es, dass in gewissen Fällen das Schlüsselbit von den ehrlichen Parteien sehr schwer vorherzusagen ist. Ein ähnliches Lemma war vor unserer Arbeit schon bekannt; wir verstärken dieses auf zwei Arten. Zum einen vergrössern wir den harten Kern auf das maximal mögliche. Zum anderen ist unser Lemma auch im üblicherweise verwendeten "uniformen" komplexitätstheoretischen Modell anwendbar. Das vorherige Lemma konnte dagegen nur im weniger verbreiteten "nicht-uniformen" Modell angewandt werden.

Contents

1. Introduction

1.1. Key Agreement

Alice and Bob, living in different places, would like to communicate privately. There are two security requirements which Alice and Bob have in such a setting. First, the communication should be *secret*, meaning that a potential eavesdropper, commonly called "Eve", will not get any information about their communication. Second, the communication should be *authentic*, which means that Eve cannot insert messages without being detected. If both these requirements are met we say that Alice and Bob can communicate *securely*.

In this thesis, we make the basic assumption that Alice and Bob share an authentic channel, i.e., the second goal is already achieved by physical means or some underlying protocol.

The term key agreement refers to the following task: given an authentic channel, Alice and Bob communicate and then agree on a bit string (called the *key*), about which Eve has no information. It is not so hard to see that key agreement is equivalent to achieving secret communication from an authentic channel, and our goal from now on will be to obtain a key.

For classical communication channels, key agreement is not possible unconditionally. But if we assume that Eve is *computationally bounded*, i.e., if the computing time which Eve has at her disposal is not very large, then this changes (or rather, it is commonly believed that this changes). Protocols which are belived to achieve key agreement in this case were first proposed by Merkle [Mer79] (in a limited sense) and by Diffie and Hellman [DH76], and are widely used in practice nowadays. However, we are currently unable to *prove* the security of any such protocol. Such a proof would imply a non-trivial lower bound for the computation of Eve, and to give such a lower bound is a notoriously hard problem in theoretical computer science (in particular, if Eve needs superpolynomial computation to break a protocol which runs in polynomial time, then $\mathsf{P} \neq \mathsf{NP}$). Consequently one makes assumptions under which such a protocol is secure. For example one assumes that computing discrete logarithms or factoring large numbers are intractable problems.

The goal of this thesis is to make this assumption as weak as possible. A very strong result of this form would be to base a key agreement protocol on an arbitrary one-way function (i.e., a function which is easy to evaluate but for which it is difficult to find a preimage of a given image). However, Impagliazzo and Rudich [IR89] showed that this is not possible using black-box reductions, which roughly means that it is beyond reach using the techniques we know.

We therefore use a different assumption, namely that a weak form of computationally secure key agreement is given: we assume that Alice and Bob have a protocol in which they end up with key bits X and Y, satisfying $\Pr[X = Y] \geq \frac{1+\alpha}{2}$ for a given parameter α (throughout the thesis we assume that the given protocol produces single key bits, which is to keep the studies simpler — extending the results to longer strings is an open problem). Furthermore, we assume that it is intractable for Eve, observing only the communication Z, to guess one of these bits with probability larger than $\frac{1+\beta}{2}$. The question studied is for what parameters α and β such a protocol is strong enough to provide key agreement.

A reader familiar with the concept of one-way functions will immediately agree that this question should be *much* simpler to answer than the question whether key agreement can be constructed from an arbitrary one-way function. Nevertheless, it is far from trivial.

1.2. Outline and Proof Sketch

Assume that a computationally secure protocol $\mathfrak{P}_C$ as described above is given: it produces key bits X and Y (Alice gets X and Bob gets Y) such that $\Pr[X = Y] \geq \frac{1+\alpha}{2}$. Furthermore, it has the property that it is intractable to predict (say) X from the communication Z with probability exceeding $\frac{1+\beta}{2}$ (one might also use a bound on the maximal probability in predicting Y from the communication; such differences will be discussed later).

Our idea is that this situation is analogous to a situation where a trusted third party distributes bits X and Y to Alice and Bob with $\Pr[X = Y] \geq \frac{1+\alpha}{2}$, as well as some information Z to Eve which satisfies

$$\Pr[f(Z) = X] < \frac{1+\beta}{2} \tag{1.1}$$

for *all* functions f (i.e., we remove the condition that f should be efficiently computable). For such random variables we can now try to

give a protocol to obtain an *information theoretically* secure key; i.e., a key whose security does not depend on the fact that Eve's computational power is bounded, but instead simply on the properties of the distribution of (X, Y, Z). This scenario has first been studied for general random variables by Maurer [Mau93], and we will use results from this line of research in our thesis. Afterwards we show that a protocol for this information theoretic setting can also be used in the computational setting.

The thesis is divided into two parts. In the first part, protocols for information theoretically secure key agreement are studied. In the second part we show that some of the resulting protocols can also be used in the computational setting.

Information Theoretic Part

After introducing basic concepts and definitions in Chapter 2, we study information theoretic key agreement in Chapters 3 and 4, when random variables X, Y, and Z are given. While the main goal is to obtain protocols which can be used in the computational setting, we believe that this study is interesting in its own right, and thus we do slightly more work than what is required for the goal of strengthening computational key agreement.

Chapter 3 studies the case where only a single message from Alice to Bob is allowed, but does so for arbitrary random variables. The one-message case is important for several reasons. First, it is much better understood than the general case. Second, protocols for the one-message case have applications in other areas (we will see two examples in this thesis: circuit polarization in Section 4.5 and the strengthening of public-key encryption schemes in Section 7.4). Third, all known protocols for the general case consist of two phases: in a first phase, the given random variables are "enhanced" using arbitrary communication to obtain instances which can be used in a one-message protocol, and in the second phase the one-message protocol is used.

In Chapter 4 we go closer to the computational setting and study random variables as explained above, i.e., we assume that the honest parties know bounds on $\Pr[X = Y]$ and $\max_f \Pr[f(Z) = X]$ (or some similar expression), but do not know the exact distribution of (X, Y, Z). For this case we study key agreement where only one message is allowed as well as key agreement where arbitrary messages are allowed.

Computational Part

In the second part of the thesis we show that if we use the given computational protocol $\mathfrak{P}_C$ from above to produce "computational instances" of random variables and then use them in an information theoretic protocol $\mathfrak{P}_{IT}$ from Chapter 4, we obtain a computationally secure key.

We first provide some definitions and some basic facts in Chapter 5, and then prove our lemma about "hard-core sets" in Chapter 6; a strengthening of a lemma previously given by Impagliazzo [Imp95]. Let us sketch this lemma quickly here: consider any process which generates random variables X and Z from some common source of uniform randomness, such that all efficient algorithms satisfy

$$\Pr[A(Z) = X] \leq \frac{1+\beta}{2}.$$

The hard-core lemma essentially states that we can distinguish two cases: with probability $1-\beta$, predicting X from Z is *very* hard for *all* algorithms, while with probability β it is easy. In other words, the randomness used to generate X and Z has a "hard subset" $\mathcal{S}$, such that if X and Z is generated using randomness from $\mathcal{S}$, then all efficient algorithms satisfy

$$\Pr[A(Z) = X] \approx \frac{1}{2}.$$

Additionally, uniform randomness will be in $\mathcal{S}$ with probability $1-\beta$. A similar result was previously known, and has found various applications in theoretical computer science. In this thesis, we improve on the known result in two ways. First, in our lemma the set size is maximal. Second, our lemma is also applicable if predicting X from Z is hard for algorithms, while the previous lemma could only be applied if the computational hardness is against circuits. In other words, our lemma can also be used in the more common "uniform" setting.

In Chapter 7 we show that the hard-core lemma implies that instances generated by $\mathfrak{P}_C$ can be used in the information theoretic protocol $\mathfrak{P}_{IT}$ to generate a computationally secure key. We can give the intuition of the proof here: imagine that we simulate a modification of the resulting protocol in which, before running $\mathfrak{P}_{IT}$, we check for each instance of the random variables produced by $\mathfrak{P}_C$ whether the randomness is from the hard set $\mathcal{S}$ (since the result of Chapter 6 holds for arbitrary processes generating (X, Z) it must hold as well for the instances generated by $\mathfrak{P}_C$). If the randomness is from the hard set, we replace the resulting key bits with

information theoretic key bits before running $\mathfrak{P}_{\mathrm{IT}}$. Now, by the properties of the information theoretic protocol, running $\mathfrak{P}_{\mathrm{IT}}$ with these random variables will produce a key which is information theoretically secure. However, the bits we replaced by random bits were indistinguishable to random bits, and we thus expect that the key of the simulation is indistinguishable from a key in the real protocol. This will imply that the key produced in the real protocol is computationally secure.

2. Preliminaries

Throughout this thesis, sets will be denoted by the calligraphic letters $\mathcal{A}$ to $\mathcal{Z}$. Exceptions are $\mathbb{R}$ for the reals and $\mathbb{N}$ for the non-negative integers.

For a finite set $\mathcal{X}$, a distribution P_X over $\mathcal{X}$ is a function $\mathsf{P}_X : \mathcal{X} \to [0,1]$ with $\sum_{x\in\mathcal{X}} \mathsf{P}_X(x) = 1$. We usually denote random variables using capital letters, and values with lower case letters. For example, if a probability distribution P_X is specified in the context, we write $\Pr[X > 1]$ for $\sum_{x>1} \mathsf{P}_X(x)$. If $\mathcal{X}$ is a subset of $\mathbb{R}$ the expected value of X is $\mathrm{E}[X] := \sum_{x\in\mathcal{X}} x\mathsf{P}_X(x)$.

If P is a probability distribution which is not specifically associated with X we analogously write $\Pr_{X\leftarrow\mathsf{P}}[X > 1]$ to denote the probability that a random variable chosen according to P is larger than 1; or $\mathrm{E}_{X\leftarrow\mathsf{P}}[X]$ for the expected value of a random variable chosen according to P. For a set $\mathcal{S}$ we also write $\Pr_{X\leftarrow\mathcal{S}}[X > 1]$ to denote that X is chosen according to the uniform distribution over $\mathcal{S}$, i.e., $\mathsf{P}_X(x) := \frac{1}{|\mathcal{S}|}$.

We often manipulate distributions freely. For example, if P_X is a distribution on $\mathcal{X}$ then $(\mathsf{P}_X)^2$ is understood to be the distribution on $\mathcal{X}^2$ where the first and second element are independent, i.e., $(\mathsf{P}_X)^2(x_0,x_1) := \mathsf{P}_X(x_0)\cdot\mathsf{P}_X(x_1)$. Furthermore, if a distribution P_{XY} over $\mathcal{X}\times\mathcal{Y}$ is given, then we write P_X and P_Y to denote marginal distributions, i.e., $\mathsf{P}_X(x) := \sum_{y\in\mathcal{Y}} \mathsf{P}_{XY}(x,y)$, analogous for P_Y. The conditional distribution $\mathsf{P}_{X|Y}$ is defined as $\mathsf{P}_{X|Y}(x|y) := \mathsf{P}_{XY}(x,y)/\mathsf{P}_Y(y)$.

We use superscripts to denote tuples, e.g., $X^n := (X_0,\ldots,X_{n-1})$ and $x^n := (x_0,\ldots,x_{n-1})$.

2.1. Prediction of Random Variables

Let P_{XY} be an arbitrary probability distribution, where X is a bit and Y is over an arbitrary finite alphabet $\mathcal{Y}$. A natural question is how good X can be predicted from Y.

Definition 2.1 (Prediction Advantage). *For a distribution P_{XY} over $\{0,1\}\times\mathcal{Y}$ and any function $f : \mathcal{Y} \to \{0,1\}$ the* advantage *of f in predicting X from Y*

is

$$\mathrm{Adv}^{f}(X|Y) := 2\Pr[f(Y) = X] - 1.$$

The maximal prediction advantage is

$$\mathrm{Adv}^{\max}(X|Y) := \max_{f:\mathcal{Y}\to\{0,1\}} \mathrm{Adv}^{f}(X|Y).$$

We could also allow randomized decisions in predicting X from Y. However, this would not change the maximal prediction advantage: every randomized strategy can be described by a distribution over functions, and the prediction advantage of a randomized strategy is just the expectation of the advantage of the chosen function.

The following lemma states that for any distribution P_{XY} we can distinguish two cases: with probability $1 - \mathrm{Adv}^{\max}(X|Y)$ *every* strategy has advantage 0, and with probability $\mathrm{Adv}^{\max}(X|Y)$ the best strategy has advantage 1.

Lemma 2.2. *Let P_{XY} be any distribution over $\{0,1\} \times \mathcal{Y}$. There exists a conditional distribution $\mathsf{P}_{B|XY}$ over $\{0,1\} \times \{0,1\} \times \mathcal{Y}$ such that*

$$\Pr[B{=}1] = \mathrm{Adv}^{\max}(X|Y), \tag{2.1}$$

for all functions $f : \mathcal{Y} \to \{0,1\}$:

$$\Pr[f(Y){=}X|B{=}0] = \frac{1}{2}, \tag{2.2}$$

and there exists a function $g : \mathcal{Y} \to \{0,1\}$ with

$$\Pr[g(Y){=}X|B{=}1] = 1. \tag{2.3}$$

Proof. We first note that (2.2) and (2.3) imply (2.1) because

$$\mathrm{Adv}^{f}(X|Y) \le \Pr[B{=}1] \cdot 1 + \Pr[B{=}0] \cdot 0 = \Pr[B{=}1]$$

holds for all functions f, and with equality for g.

We define

$$\Pr[B{=}0|X{=}x, Y{=}y] := \frac{\min(\mathsf{P}_{XY}(0,y), \mathsf{P}_{XY}(1,y))}{\mathsf{P}_{XY}(x,y)}.$$

First, we show that (2.2) holds: for any $y \in \mathcal{Y}$ we get (using Bayes' Theorem):

$$\begin{aligned}\Pr[X{=}0|B{=}0, Y{=}y] &= \frac{\Pr[B{=}0|X{=}0, Y{=}y]\ \Pr[X{=}0|Y{=}y]}{\Pr[B{=}0|Y{=}y]} \\ &= \frac{\Pr[B{=}0|X{=}1, Y{=}y]\ \Pr[X{=}1|Y{=}y]}{\Pr[B{=}0|Y{=}y]} \\ &= \Pr[X{=}1|B{=}0, Y{=}y],\end{aligned}$$

which implies $\Pr[f(Y){=}X|B{=}0, Y{=}y] = \frac{1}{2}$ for every fixed y and thus this must also hold overall.

Further, the function g defined as

$$g(y) := \begin{cases} 0 & \text{if } \mathsf{P}_{XY}(0, y) \geq \mathsf{P}_{XY}(1, y), \\ 1 & \text{otherwise,} \end{cases}$$

satisfies (2.3), since $B = 1$ implies $\mathsf{P}_{XY}(x, y) > \mathsf{P}_{XY}(1{-}x, y)$. □

We give a quick example how the above lemma can be useful (we will use this example in Section 4.3).

Lemma 2.3. *Let P_{XY} be an arbitrary distribution over $\{0, 1\} \times \mathcal{Y}$, and P_{XY}^n the n-wise direct product. Then,*

$$\mathrm{Adv}^{\max}(X_0 \oplus \cdots \oplus X_{n-1}|Y_0, \ldots, Y_{n-1}) = \left(\mathrm{Adv}^{\max}(X|Y)\right)^n.$$

Proof. For every i, let B_i be the random variable whose existence is guaranteed by Lemma 2.2. If $B_i = 0$ for *any* $0 \leq i < n$, then any function $f : \mathcal{Y}^n \to \{0, 1\}$ will output $X_0 \oplus \cdots \oplus X_{n-1}$ with probability $\frac{1}{2}$, and thus

$$\begin{aligned}&\mathrm{Adv}^{\max}(X_0 \oplus \cdots \oplus X_{n-1}|Y_0, \ldots, Y_{n-1}) \\ &\qquad \leq \left(\mathrm{Adv}^{\max}(X|Y)\right)^n \cdot 1 + \left(1 - \left(\mathrm{Adv}^{\max}(X|Y)\right)^n\right) \cdot 0 \\ &\qquad = \left(\mathrm{Adv}^{\max}(X|Y)\right)^n.\end{aligned}$$

Further, let g be the function which predicts X correctly from Y in case $B = 1$. Then $g'(y_0, \ldots, y_n) := g(y_0) \oplus \cdots \oplus g(y_{n-1})$ will be correct if

$B_i = 1$ for all i. Thus,

$$\begin{aligned}\mathrm{Adv}^{\max}(X_0 \oplus \cdots \oplus X_{n-1}|Y_0, \ldots, Y_{n-1}) \\ &\geq \mathrm{Adv}^{g'}(X_0 \oplus \cdots \oplus X_{n-1}|Y_0, \ldots, Y_{n-1}) \\ &= \big(\mathrm{Adv}^{g}(X|Y)\big)^n. \\ &= \big(\mathrm{Adv}^{\max}(X|Y)\big)^n. \end{aligned}$$

□

A concept related to $\mathrm{Adv}^{\max}(X|Y)$ is the statistical distance of two distributions. For two distributions P_{Y_0} and P_{Y_1} over the same set it is defined as follows:

Definition 2.4 (Statistical distance). *For probability distributions* P_{Y_0} *and* P_{Y_1} *over* $\mathcal{Y}$ *the statistical distance* $\|\mathsf{P}_{Y_0} - \mathsf{P}_{Y_1}\|$ *is*

$$\|\mathsf{P}_{Y_0} - \mathsf{P}_{Y_1}\| := \frac{1}{2} \sum_{y \in \mathcal{Y}} \big|\mathsf{P}_{Y_0}(y) - \mathsf{P}_{Y_1}(y)\big|.$$

If P_{XY} is a distribution over $\{0,1\} \times \mathcal{Y}$ for which X is a uniform bit, then the statistical distance of the two distributions $\mathsf{P}_{Y|X=0}$ and $\mathsf{P}_{Y|X=1}$ is exactly the maximal prediction advantage.

Lemma 2.5. *Let* P_{XY} *be any distribution with* $\mathsf{P}_X(0) = \mathsf{P}_X(1) = \frac{1}{2}$*. Then,*

$$\mathrm{Adv}^{\max}(X|Y) = \|\mathsf{P}_{Y|X=0} - \mathsf{P}_{Y|X=1}\|.$$

Proof. For an arbitrary function $g : \mathcal{Y} \to \{0,1\}$ we get

$$\begin{aligned}\Pr[g(Y) = X] &= \sum_{y \in \mathcal{Y}} \mathsf{P}_{XY}(g(y), y) \\ &\leq \sum_{y \in \mathcal{Y}} \max(\mathsf{P}_{XY}(0,y), \mathsf{P}_{XY}(1,y)) \\ &= \sum_{y \in \mathcal{Y}} \frac{\mathsf{P}_{XY}(0,y) + \mathsf{P}_{XY}(1,y)}{2} + \sum_{y \in \mathcal{Y}} \frac{|\mathsf{P}_{XY}(0,y) - \mathsf{P}_{XY}(1,y)|}{2} \\ &= \frac{1}{2} + \frac{1}{2}\|\mathsf{P}_{Y|X=0} - \mathsf{P}_{Y|X=1}\|.\end{aligned}$$

Also, the above holds with equality for the function g defined as

$$g(y) := \begin{cases} 0 & \text{if } \mathsf{P}_{XY}(0,y) \geq \mathsf{P}_{XY}(1,y), \\ 1 & \text{otherwise.} \end{cases}$$

□

2.2. Security of a Key

Let S_A be the random variable Alice wants to use as a key, S_B the random variable Bob wants to use as key, and T all the information accessible to an eavesdropper Eve. If S_A and S_B are a perfect key, then $S_A = S_B$, uniformly distributed over the keyspace $\mathcal{X}$, for every value of T.

Definition 2.6 (Perfect Key). *The random variables S_A and S_B are a perfect key over $|\mathcal{S}|$ with respect to information T if $\mathsf{P}_{S_A S_B T} = \mathsf{P}_{S_A S_B}\mathsf{P}_T$ (i.e., T is independent of S_A and S_B) and*

$$\mathsf{P}_{S_A S_B}(s_a, s_b) = \begin{cases} \frac{1}{|\mathcal{S}|} & \text{if } s_a = s_b, \\ 0 & \text{otherwise.} \end{cases}$$

In many scenarios we do not have the possibility of obtaining such a perfect key, but still we can obtain random variables S_A and S_B while Eve gets T such that $\mathsf{P}_{S_A S_B T}$ has statistical distance at most ε from a perfect key. Lemma 2.2 together with Lemma 2.5 emphasizes the usefulness of this: it is impossible to distinguish the imperfect from a perfect key better than with probability $\frac{1+\varepsilon}{2}$, even given the key and all the information of Eve.

We define the related concept that X is close to uniform with respect to Y.

Definition 2.7 (Closeness). *Let P_{XY} be a probability distribution over $\mathcal{X} \times \mathcal{Y}$. Let P_U be the uniform distribution over $\mathcal{X}$. Then, X is ε-close to uniform with respect to Y if*

$$\|\mathsf{P}_{XY} - \mathsf{P}_U\mathsf{P}_Y\| \leq \varepsilon.$$

Our goal of obtaining a key can then be formulated as follows: we want a protocol which produces random variables S_A for Alice and S_B for Bob, such that $\Pr[S_A = S_B] \geq 1 - \gamma$ and S_A is ε-close to uniform with respect to T:

Definition 2.8 (Soundness and secrecy). *Let $\mathsf{P}_{S_A S_B T}$ be a probability distribution. We say that (S_A, S_B) is a key with soundness $1 - \gamma$ if $\Pr[S_A = S_B] \geq 1 - \gamma$. Further (S_A, S_B) is a key with secrecy $1 - \varepsilon$ with respect to T if S_A is ε-close to uniform with respect to T.*

When no confusion can arise, we often omit T and say that a key has secrecy $1 - \varepsilon$. Then it is understood that the key should have secrecy $1 - \varepsilon$ with respect to the complete information accessible to an eavesdropper Eve.

2.3. Entropy Measures

The entropy of a random variable X is a measure of the *uncertainty* of X. We use several different flavors of entropy in this thesis:[1]

Definition 2.9 (Entropy measures). *Let P_{XYZ} be a probability distribution over $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. The* entropy *is*

$$H(X) := \sum_{x \in \mathcal{X}} \mathsf{P}_X(x) \log\Big(\frac{1}{\mathsf{P}_X(x)}\Big).$$

The conditional entropy *is*

$$H(X|Y) := \sum_{y \in \mathcal{Y}} \mathsf{P}_Y(y) \cdot H(X|Y{=}y),$$

the mutual information *is*

$$I(X;Y) := H(X) + H(Y) - H(XY),$$

and the conditioned mutual information *is*

$$I(X;Y|Z) := H(X|Z) + H(Y|Z) - H(XY|Z).$$

The min-entropy *of X is*

$$H_\infty(X) := \min_{x \in \mathcal{X}} \log\Big(\frac{1}{\mathsf{P}_X(x)}\Big),$$

and the min-entropy *of X conditioned on Y is*

$$H_\infty(X|Y) := \min_{y \in \mathcal{Y}} H_\infty(X|Y{=}y).$$

Further, we define

$$H_0(X) := \log\Big|\big\{x \in \mathcal{X} \mid \mathsf{P}_X(x) > 0\big\}\Big|$$

and

$$H_0(X|Y) := \max_{y \in \mathcal{Y}} H_0(X|Y{=}y).$$

[1] Throughout this thesis, we use $\log(\cdot)$ to denote the logarithm to base 2.

The entropies $H(X)$ and $H(X|Y)$, as well as the mutual information $I(X;Y)$ and $I(X;Y|Z)$ are standard quantities in information theory. We refer to [CT91] or any other textbook on information theory for a discussion of these quantities.

The interpretation of $H_\infty(X|Y)$ and $H_0(X|Y)$ is as follows: the conditional min-entropy $H_\infty(X|Y)$ is the *guaranteed* amount of randomness which is in X, even if Y is publicly known. Further, $2^{H_0(X|Y)}$ is an upper bound on the size of the set of possible values for X given Y; or, more intuitively (but not exactly correct because of rounding issues), $H_0(X|Y)$ is the number of bits needed if one wants to communicate X to someone who knows Y.

For the min-entropy it is often useful to have these quantities in a version tolerating errors (analogous for H_0, but we do not need this in this thesis). Thus, $H^\varepsilon_\infty(X|Y)$ is a lower bound on the randomness contained in X if Y is given, as long as an "error probability" ε is tolerated. This leads to the definition of smoothened min-entropy, as introduced in [RW05]:

Definition 2.10 (Smooth min-entropy). *For $1 > \varepsilon \geq 0$ the ε-smooth min-entropy is*

$$H^\varepsilon_\infty(X) := \max_{X' : \|\mathsf{P}_{X'} - \mathsf{P}_X\| \leq \varepsilon} H_\infty(X'),$$

where the maximization is over random variables X with potentially larger alphabets than $\mathcal{X}$ (it is easy to see that this maximum exists). Analogously the ε-smooth conditioned min-entropy $H^\varepsilon_\infty(X|Y)$ *is*

$$H^\varepsilon_\infty(X|Y) := \max_{X'Y' : \|\mathsf{P}_{X'Y'} - \mathsf{P}_{XY}\| \leq \varepsilon} H_\infty(X'|Y').$$

2.4. Independent Repetitions

Upper bounds on the probabilities of tails

Let any distribution P_X be given, and $\mathsf{P}_{X^r} := (\mathsf{P}_X)^r$ be the distribution of r independent copies of X. The asymptotic equipartition property (see for example [CT91]) states that, if r is large enough, the vast majority of the outcomes will have probability $2^{-r(H(X)\pm\varepsilon)}$. The following Proposition from [Ren05, Section 3.3] gives a quantitative bound on this (and also for the conditioned case), which is tight up to the constants which appear in it. A slightly weaker bound can be found in [ILL89].

Proposition 2.11. *Let P_{XY} be a probability distribution over $\mathcal{X} \times \mathcal{Y}$, $\mathsf{P}_{X^rY^r} := (\mathsf{P}_{XY})^r$ the r-wise direct product. Then, for any $\delta \in [0, \log(|\mathcal{X}|)]$ and (x^r, y^r) chosen according to $\mathsf{P}_{X^rY^r}$*

$$\Pr_{(U,V)\leftarrow \mathsf{P}_{X^rY^r}}\Big[\log\Big(\frac{1}{\mathsf{P}_{X^r|Y^r}(U|V)}\Big) \geq r\big(H(X|Y)+\delta\big)\Big] \leq 2^{-\frac{r\delta^2}{16\log^2(|\mathcal{X}|)}},$$

and, similarly,

$$\Pr_{(U,V)\leftarrow \mathsf{P}_{X^rY^r}}\Big[\log\Big(\frac{1}{\mathsf{P}_{X^r|Y^r}(U|V)}\Big) \leq r\big(H(X|Y)-\delta\big)\Big] \leq 2^{-\frac{r\delta^2}{16\log^2(|\mathcal{X}|)}}.$$

We immediately get bounds on the smooth min-entropy for this case.

Corollary 2.12. *Let P_{XY} be a probability distribution over $\mathcal{X} \times \mathcal{Y}$, $\mathsf{P}_{X^rY^r} := (\mathsf{P}_{XY})^r$ the r-wise direct product. Then, for any $\varepsilon > 0$:*

$$H_\infty^\varepsilon(X_0 \dots X_{r-1}|Y_0 \dots Y_{r-1}) \geq rH(X|Y) - 4\sqrt{r\log(1/\varepsilon)}\log(|\mathcal{X}|).$$

Proof. We set $\delta := 4\sqrt{r\log(\frac{1}{\varepsilon})}\log(|\mathcal{X}|)$ in the first bound in Proposition 2.11. □

A different (but very similar) situation is the following: Assume that X_0 to X_{r-1} are independently distributed in the interval $[0,1]$. Then, analogously to Proposition 2.11 one would expect that with high probability, the mean of the X_i is very close to the expected mean. The following bound by Hoeffding [Hoe63] states that this is indeed the case.

Proposition 2.13 (Hoeffding's bound). *Let $\mathsf{P}_{X_0X_1\cdots X_{r-1}} = \mathsf{P}_{X_0} \cdot \dots \cdot \mathsf{P}_{X_{r-1}}$ be a product distribution with $X_i \in [0,1]$. Let $\overline{X} := \frac{1}{r}\sum_{i=0}^{r-1} X_i$. Then, for any $\varepsilon > 0$,*

$$\Pr\big[\,\overline{X} \geq \mathrm{E}[\overline{X}] + \varepsilon\big] \leq e^{-r\varepsilon^2},$$

and,

$$\Pr\big[\,\overline{X} \leq \mathrm{E}[\overline{X}] - \varepsilon\big] \leq e^{-r\varepsilon^2}.$$

Lower bounds on the probabilities of tails

A random variable B over $\{0,1\}$ distributed according to the distribution

$$\mathsf{P}_B(b) := \begin{cases} 1-p & \text{if } b = 0, \\ p & \text{if } b = 1, \end{cases}$$

is said to be a *Bernoulli trial with success probability* p. The *binomial distribution* $\mathsf{P}_p(k|r)$ is defined as

$$\mathsf{P}_p(k|r) := \binom{r}{k} p^k (1-p)^k,$$

and $\mathsf{P}_p(k|r)$ is exactly the probability of obtaining k successes from r independent Bernoulli trials with probability p.

The Hoeffding bound (Proposition 2.13) shows that that if n is large enough the number of successes of r Bernoulli trials will be very close to rp. The following lemma from [HR05a] gives a bound in the opposite direction, i.e., it gives a lower bound on the probability that the number of successes is at least $r(p+\varepsilon)$ for some $\varepsilon > 0$. This will be useful when we want to show that our results are tight. We give a proof of this Lemma in Appendix A.

Lemma 2.14. *Let $p \geq \frac{1}{2}$, $r, s \in \mathbb{N}$ such that $pr + 3s \leq r$. Then,*

$$\sum_{k=\lceil pr \rceil + s}^{\lceil pr \rceil + 2s - 1} \mathsf{P}_p(k|r) > \frac{s}{2\sqrt{r}}\, e^{-\frac{2s^2}{rp(1-p)}}.$$

2.5. Randomness Extraction

Let a random variable X be given. An extractor is a randomized function which uses the randomness inherent in X to obtain a nearly uniform bit string. Allowing the function to be randomized means that it has access to additional uniform randomness, called *seed*, and we expect that the concatenation of the seed and the output of the extractor is close to uniform (this implies that the extractor cannot just output the seed).

The maximal probability occurring in P_X is $2^{-H_\infty(X)}$. Since in a uniform bit string of length r all occurring probabilities are 2^{-r} we only hope to extract r bits if $H_\infty(X) \geq r$; we call $H_\infty(X) - r$ the *entropy loss* of the extractor.

Definition 2.15 (Strong extractor). *A function $h : \mathcal{X} \times \mathcal{S} \to \{0,1\}^r$ is a strong extractor with closeness ε and entropy loss ℓ if, for any distribution P_X with $H_\infty(X) \geq r + \ell$, and S uniform over $\mathcal{S}$, the distribution of $h(X,S)$ is ε-close to uniform with respect to S.*

The term strong in the above definition comes from the requirement that $h(X,S)$ is ε-close to uniform *with respect to* S. In contrast, a non-strong extractor (a concept we do not need in this thesis) satisfies that the output is ε-close to uniform (without conditioning on $\mathcal{S}$). In this case one expects the output of f to be $\log(|\mathcal{S}|)$ bits longer.

In [BBR88, ILL89] it was proven that weak two-universal hash functions (as introduced in [CW79]) form one possibility of strong extractors. A weak two-universal hash function $h(x,s)$ satisfies that the collision probability of two different $x \neq x'$ is the same as for a completely random function. The term *weak* is used because we do not require that the output distribution of $h(x,S)$ for uniform random S is uniform.

Definition 2.16 (Weak two-universal hash function). *A function $h : \mathcal{X} \times \mathcal{S} \to \mathcal{Y}$ is weak two-universal if for all $x, x' \in \mathcal{X}$ with $x \neq x'$:*

$$\Pr_{S \leftarrow \mathcal{S}}[h(x,S) = h(x',S)] = \frac{1}{|\mathcal{Y}|}.$$

For example, let $\mathcal{X} = \mathcal{S} = \{0,1\}^r$, identify $\{0,1\}^r$ with $\mathrm{GF}(2^r)$ in an arbitrary way, and let $\odot$ be the multiplication over $\mathrm{GF}(2^r)$. Further, for $x \in \{0,1\}^r$ let $x|_{0\ldots i-1}$ be the first i bits of x. Because for any $x, x' \in \{0,1\}^r$ with $x \neq x'$

$$\begin{aligned}\Pr_{A \leftarrow \{0,1\}^r}&\left[(A \odot x)|_{0\ldots i-1} = (A \odot x')|_{0\ldots i-1}\right] \\ &= \Pr_{A \leftarrow \{0,1\}^r}\left[(A \odot (x - x'))|_{0\ldots i-1} = 0^i\right] = 2^{-i},\end{aligned}$$

the function $h : \{0,1\}^r \times \{0,1\}^r \to \{0,1\}^i$ which is given by $h(x,s) := (x \odot s)|_{0\ldots i-1}$ is a weak two-universal hash function.

The following theorem from [BBR88, ILL89] shows that any weak two-universal hash function is a strong extractor. Our proof is adapted from [LW95].

Theorem 2.17 (Left-over hash lemma). *Let $h : \mathcal{X} \times \mathcal{S} \to \{0,1\}^m$ be weak two-universal and $\varepsilon > 0$. Then, h is a strong extractor with closeness ε and entropy loss $2\log(1/\varepsilon)$.*

Proof. Let P_{VS} be the distribution of the output of the weak two-universal hash-function concatenated with the seed S, and P_U the uniform distri-

bution over $\{0,1\}^m$. We use the inequality $(\sum_{i=1}^n a_i)^2 \leq n \sum_{i=1}^n a_i^2$ which follows from Cauchy-Schwarz to obtain

$$\begin{aligned}
&\|\mathsf{P}_{VS} - \mathsf{P}_U\mathsf{P}_S\| \\
&\quad= \frac{1}{2} \sum_{v\in\{0,1\}^m, s\in\mathcal{S}} \Big|\mathsf{P}_{VS}(v,s) - \frac{1}{|\mathcal{S}|\,2^m}\Big| \\
&\quad\leq \frac{1}{2}\sqrt{|\mathcal{S}|\,2^m}\sqrt{\sum_{v,s}\mathsf{P}_{VS}^2(v,s) - 2\sum_{v,s}\frac{\mathsf{P}_{VS}(v,s)}{|\mathcal{S}|\,2^m} + \sum_{v,s}\Big(\frac{1}{|\mathcal{S}|\,2^m}\Big)^2} \\
&\quad= \frac{1}{2}\sqrt{|\mathcal{S}|\,2^m}\sqrt{\sum_{v,s}\mathsf{P}_{VS}^2(v,s) - \frac{1}{|\mathcal{S}|\,2^m}}\,. \qquad (2.4)
\end{aligned}$$

Let now X_0 and X_1 be independently distributed according to P_X, and S_0 and S_1 independently distributed according to P_S. The collision probability of $h(X,S)$ concatenated with S is

$$\Pr[h(X_0,S_0) = h(X_1,S_1) \wedge S_0 = S_1] = \sum_{v,s}\mathsf{P}_{VS}^2(v,s).$$

Thus, (2.4) gives an an upper bound on $\|\mathsf{P}_{VS} - \mathsf{P}_U\mathsf{P}_S\|$ from the collision probability of two independent invocations of the hash-function on two independent samples from the distribution P_X. We can estimate this collision probability as follows:

$$\begin{aligned}
&\Pr[h(X_0,S_0){=}h(X_1,S_1) \wedge S_0{=}S_1] \\
&\quad= \Pr[S_0{=}S_1]\Pr[h(X_0,S_0){=}h(X_1,S_0)] \\
&\quad\leq \Pr[S_0{=}S_1]\big(\Pr[X_0{=}X_1] + \Pr[h(X_0,S_0){=}h(X_1,S_0)|X_0{\neq}X_1]\big) \\
&\quad\leq \frac{1}{|\mathcal{S}|}\Big(\frac{1}{2^{m+2\log(1/\varepsilon)}} + \frac{1}{2^m}\Big) \\
&\quad= \frac{1+\varepsilon^2}{|\mathcal{S}|\,2^m}. \qquad (2.5)
\end{aligned}$$

Inserting (2.5) into (2.4) yields $\|\mathsf{P}_{VS} - \mathsf{P}_U\mathsf{P}_S\| \leq \frac{\varepsilon}{2}$. □

Extractors are a very well studied topic. In particular, the question how many bits the seed $\mathcal{S}$ needs to have has undergone much research. For us weak two-universal hash functions are sufficient, however. We refer to [Sha02] for an overview to the construction of other extractors.

Part I.

Information Theoretically Secure Key Agreement

3. One-Message Key Agreement

In this chapter we study the following scenario: Alice, Bob, and Eve have access to many independent instances of random variables X, Y, and Z, respectively, distributed according to some fixed and commonly known distribution P_{XYZ}. Alice and Bob want to agree on a key with a single message from Alice to Bob, and using as few random variables as possible. In this chapter we will see for which distributions P_{XYZ} this is possible and give protocols for this task. Further, we give lower bounds on the number of random variables needed.

Overview of this chapter

In Section 3.1 we start by giving an intuition of the protocol and explain the three basic steps. In the first step, a *preprocessing* step, Alice manipulates her random variables individually to obtain random variables which are better suited for the subsequent protocol. In the next step, called *information reconciliation*, Alice sends Bob information which allows him (but not Eve) to find the random variables of Alice. This gives Alice and Bob a common string about which Eve has imperfect knowledge. In the third step, called *privacy amplification*, Alice and Bob use this common string to obtain a secure key. The discussion of Section 3.1 also leads to the definition of the *one-message key rate*, the rate at which Alice and Bob can use their random variables to obtain key bits.

In Section 3.2 we give a detailed study of the preprocessing step, and Section 3.3 contains a description of our information reconciliation protocol. As privacy amplification is simple to implement (we apply a strong extractor) we do not need a section in order to describe it. Section 3.4 then describes the complete protocol in detail.

Section 3.5 contains lower bounds on the number of random variables needed to obtain a key. We will see that the lower bounds basically match the usage in our protocols (up to constants in the non-dominating terms).

Related work

The problem of communicating securely in an information theoretic setting was first considered by Shannon [Sha49a]. He showed that, for one-

way communication, Alice and Bob cannot secretly communicate over a non-secret channel unless they have some previously shared information K. The result was later extended to two-way communication by Maurer [Mau93].

Wyner [Wyn75], and subsequently Csiszár and Körner [CK78] studied the question whether information theoretic secure communication is possible when Alice has a noisy channel to Bob on which Eve has only limited access. Maurer [Mau93] proposed to study key agreement in the more general scenario where a source distributes random variables to parties Alice, Bob, and Eve (for example a satellite broadcasting random bits). He showed that interaction between Alice and Bob can make key agreement possible even if one-way communication is not sufficient. Subsequently, Ahlswede and Csiszár [AC93] studied Maurer's scenario in the setting where only one-way communication is allowed.

The concepts of information reconciliation and privacy amplification evolved parallel with the above development ([BBR88, BBCM95, BS93]).

In a different line of research Juels and Wattenberg [JW99] constructed a "fuzzy commitment scheme". Based on this work, Dodis, Reyzin, and Smith [DRS04] introduced the concept of *fuzzy extractors*. It is easy to see that this concept is equivalent to the combination of information reconciliation and privacy amplification.

Contributions of this thesis

Almost all the contents of this chapter were previously known. We list the minor improvements over the previously known results which are made in this chapter:

- Theorem 3.3 concerns the preprocessing step, in which Alice uses X to obtain random variables U and V (as described later). The theorem states that the alphabet size of U and V need not be larger than the alphabet size of X. Previously, only slightly weaker bounds were known (the alphabet size of V was bounded by $|\mathcal{X}| + 3$, and the alphabet size of U by $|\mathcal{X}| + 1$ see [CK78, AC93]). A similar statement where two-way communication is considered can be found in [CRW03a] (with the same alphabet size as in our case). It is possible that our bounds are tight.

- We show how to do information reconciliation for any distribution P_{XYZ} using a code which has a rate close to the channel capacity, instead of using a two-universal hash-function. The method

Alice	1	1	1	0	0	1	0	1	0	0	0
Bob	1	$\perp$	1	0	0	1	$\perp$	1	0	0	0
Eve	$\perp$	$\perp$	1	$\perp$	$\perp$	$\perp$	0	$\perp$	0	$\perp$	$\perp$

Figure 3.1.: Possible sample of random variables in the example ($p = 0.2$, $q = 0.8$)

we use does not seem to appear anywhere in the literature, but it is safe to assume that the possibility of this was folklore. The advantage of this approach is the possibility of using concatenated codes, and this allows for a polynomial time implementation of Bob's algorithm (Theorem 3.15). This was previously published in [HR05b].

- We give more exact lower bounds on the number of random variables needed for one-message key agreement (Theorem 3.17). Previously, lower bounds were only stated in an asymptotic manner ([CK78, AC93, HR05b]). Also, we show that the number of random variables used must increase as the required security increases (Theorem 3.18).

3.1. Example and One-Message Key Rate

In order to illustrate the basic idea, we start with an example probability distribution P_{XYZ}, and give a simple protocol for this distribution. The protocol has the disadvantage that it requires a huge amount of communication.

The distribution is best described by the following random process: Alice gets a uniform random bit X, Bob gets the same bit after it has been sent through an erasure channel which outputs a special erasure symbol $\perp$ with probability p, and Eve also gets X but erased with probability $q > p$ (see Figure 3.1, one can imagine that these random variables are distributed by a trusted third party). Clearly, in such a scenario Alice and Bob have a certain advantage over Eve, and one hopes that this advantage can be exploited in order to get a secure key.

In this case, our protocol has two steps. First, *information reconciliation* makes sure Alice and Bob get a common string over which Eve has some uncertainty. Then, *privacy amplification* transforms this bit string into a secret key. We now describe these steps for this distribution in more detail.

3.1.1. Information Reconciliation

Alice, who obtained the bit string x_0 of length n from the trusted third party first chooses a very large number of uniform random n-bit strings $x_1, \ldots, x_t$ and sends them to Bob, whereas she hides x_0 in a random position in the strings (in other words, she sends a random permutation of $x_0, \ldots, x_t$ to Bob). We do not care about the amount of communication needed in this simple protocol.

A randomly chosen string matches Bob's information with probability $2^{-n(1-p)}$, while it matches Eve's information with probability $2^{-n(1-q)}$. Thus, if $q > p$ and n is large enough, we can choose t appropriately between $2^{n(1-q)}$ and $2^{n(1-p)}$, such that with high probability only the string x_0 matches Bob's information, while many strings will match Eve's information. In other words, Alice and Bob agree on a common string, while Eve still has large min-entropy about x_0.

This method to do information reconciliation requires a large amount of communication. A different method often used in the literature is that Alice sends Bob the output of a randomly chosen two-universal hash function applied on her input (including the information which two-universal hash function was chosen). The idea here is that the possible preimages of a two-universal hash function have similar properties as our randomly chosen strings. However, in this case it is not clear how Bob can recover the input of Alice *computationally* efficiently. For this reason we will use error correcting codes in our construction, as explained in Section 3.3.

3.1.2. Privacy Amplification

After information reconciliation, Alice and Bob both know x_0 but cannot use it as key, since Eve has some information about it (in fact, Eve knows some positions of x_0 with certainty in our setting). Alice and Bob rectify this situation in the next step, called *privacy amplification*.

The simple idea is that Alice and Bob can apply a strong extractor: Alice chooses a seed uniformly at random and sends it to Bob. Then, they both apply the extractor to x_0. Since for Eve x_0 has large min-entropy, this gives a bit string which is close to uniform with respect to Eve's information.

3.1.3. Preprocessing

The simple protocol explained above uses the fact that Bob knows more about the value of Alice than Eve knows. In fact, one can show that a

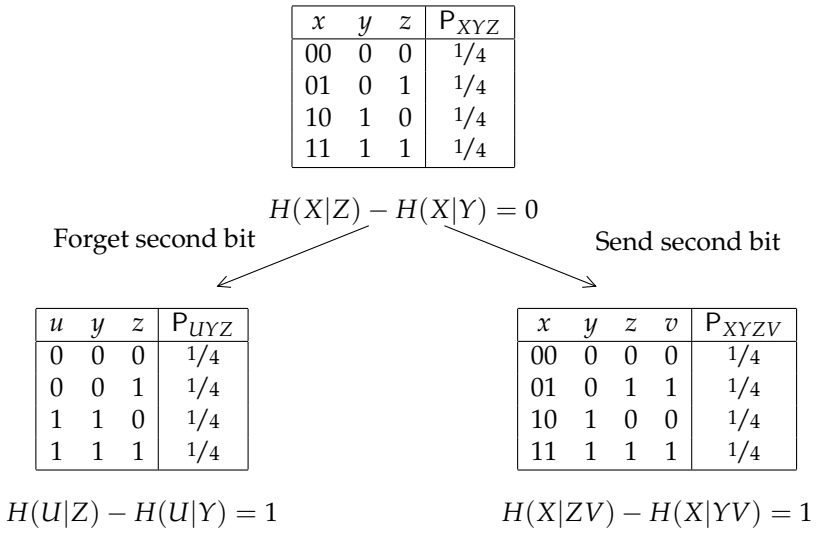


Figure 3.2.: For some random variables "forgetting information" or "sending information" helps.

protocol similar to the above works for any distribution P_{XYZ} as long as $H(X|Z) - H(X|Y)$ is positive. Even more, the protocol generates key bits at a rate of $H(X|Z) - H(X|Y)$, i.e., asymptotically Alice and Bob get that many key bits per random variable used.

In some cases it is possible to improve upon $H(X|Z) - H(X|Y)$. For example, assume that Alice gets two independent bits, Bob knows the first, and Eve the second (see Figure 3.2, topmost table). In this case, both conditioned entropies $H(X|Z)$ and $H(X|Y)$ are equal to one. But since Alice knows the distribution she can solve this problem easily: she "forgets" the second bit in every X. If U is the resulting random variable (i.e., the first bit of X) we have $H(U|Z) = 1$ and $H(U|Y) = 0$, and therefore the protocol sketched previously will work. Alternatively Alice can send the second bit to Bob. If we call the random variable sent V (i.e., the second bit of X), we then have $H(X|ZV) = 1$ and $H(X|YV) = 0$. We will see later that these two preprocessing steps are sufficient to make sure our protocols achieve an optimal rate, i.e., asymptotically no stronger results are possible.

3.1.4. The One-Message Key Rate

This previous discussions give rise to the following definition, which we can trace back to [AC93].[1]

Definition 3.1 (One-message key rate). *Let P_{XYZ} be any probability distribution over $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. The one-message key rate $S_{\rightarrow}(X;Y|Z)$ is*

$$S_{\rightarrow}(X;Y|Z) := \sup_{\mathsf{P}_{UV|X}} H(U|ZV) - H(U|YV),$$

where the supremum is over all conditional distributions $\mathsf{P}_{UV|X}$ with finite alphabets for U and V.

In our previous example we obtained U and V in a deterministic way from X. However, in general this is not possible (we will see an example in Chapter 4).

It will follow from Theorem 3.3 that there always exists a distribution $\mathsf{P}_{UV|X}$ with $S_{\rightarrow}(X;Y|Z) = H(U|ZV) - H(U|YV)$, i.e., the supremum can always be achieved and is finite.

The naming of $S_{\rightarrow}(X;Y|Z)$ as one-message key rate originates from Theorem 3.13, which states that for any rate $R < S_{\rightarrow}(X;Y|Z)$ it is possible to obtain nR key bits from n random variables, as long as n is large enough; as well as from Theorem 3.18, which states that no rate $R > S_{\rightarrow}(X;Y|Z)$ can be achieved by any protocol.

Finally, we note that the expression we maximize over in Definition 3.1 can be equivalently written as

$$\begin{aligned} H(U|ZV) - H(U|YV) & \\ &= H(UZV) - H(ZV) - H(UYV) + H(YV) \\ &= H(Z|UV) - H(Y|UV) - (H(Z|V) - H(Y|V)). \end{aligned} \tag{3.1}$$

3.2. Preprocessing: Alphabet Size

We first show that in Definition 3.1 it is sufficient to consider random variables U and V over $\mathcal{X}$, i.e., the alphabet of U and V need not be larger

[1] In [AC93], the expression $I(VY;U) - I(VZ;U)$ (which is easily seen to be equal to $H(U|ZV) - H(U|YV)$) is used. Further, $\mathsf{P}_{UV|X}$ is restricted to distributions of the form $\mathsf{P}_{U|X}\mathsf{P}_{V|U}$. This is possible because from $H(U|ZV) - H(U|YV) = H(UV|ZV) - H(UV|YV)$ we see that V can be encoded into U. We believe that our version is more natural.

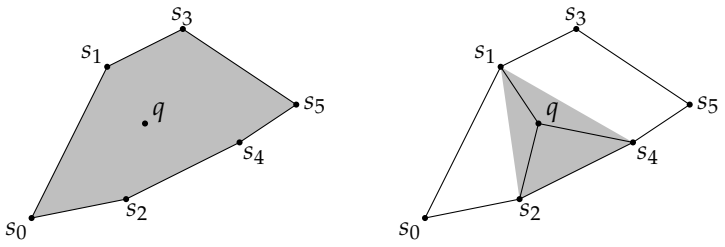


Figure 3.3.: Carathéodory's Theorem for $\mathbb{R}^2$: Every point in the convex hull of points $\{s_0, \ldots, s_{r-1}\}$ can be written as a convex combination of at most 3 of these points.

than $|\mathcal{X}|$. This means that the supremum in Definition 3.1 is over a compact set, which implies that it can be achieved and is finite. Further, it is convenient because we can limit the amount of communication and storage needed, and it may simplify the task of finding optimal variables U and V.

We will prove the theorem by showing that for any given probability distribution $\mathsf{P}_{UV|X}$ we can reduce the alphabet size of U and V without decreasing $H(U|ZV) - H(U|YV)$. We do this in two steps: first, we show that the alphabet size of $\mathcal{V}$ can be reduced, then we show that for every $v \in \mathcal{V}$ the alphabet size of $\mathcal{U}$ can be reduced as well.

For both steps, the following lemma is of key importance. It is a slight strengthening of Carathéodory's Theorem, and attributed to Fenchel and Eggleston in [CK78]. However, the proof referred to in [CK78] is for a different statement.

Carathéodory's Theorem is very intuitive: it states that any point in the convex hull of a finite number of points in $\mathbb{R}^d$ can be written as convex combination of $d+1$ points (see Figure 3.3). One can conveniently formulate this as follows: for any finite set $\mathcal{S} \subseteq \mathbb{R}^d$ and for any probability distribution P_S over $\mathcal{S}$, there exists a probability distribution $\mathsf{P}_{\overline{S}}$ over $\mathcal{S}$ such that $\mathrm{E}[\overline{S}] = \mathrm{E}[S]$ and $\mathsf{P}_{\overline{S}}(x)$ is zero for all but $d+1$ elements of $\mathcal{S}$. Our strengthening shows that this holds even if the expected value of a given function h on these points is not allowed to decrease.

Lemma 3.2. *Let $\mathcal{S} \subseteq \mathbb{R}^d$ be a finite set and let $h : \mathcal{S} \to \mathbb{R}$ be an arbitrary function. For any probability distribution $\mathsf{P}_S : \mathcal{S} \to [0,1]$ there exists a probability distribution $\mathsf{P}_{\overline{S}} : \mathcal{S} \to [0,1]$ such that $\mathrm{E}[\overline{S}] = \mathrm{E}[S]$, $\mathrm{E}[h(\overline{S})] \geq \mathrm{E}[h(S)]$ and the*

size of the support of $\overline{S}$ is at most $d+1$.

Proof. We show that we can remove one point from the support if it is greater than $d+1$. Induction then proves the lemma.

Let $\mathcal{S} := \{s_0, \ldots, s_{r-1}\}$ and assume without loss of generality that $\mathsf{P}_S(s) > 0$ for all $s \in \mathcal{S}$ and that $\mathrm{E}[S] = \mathbf{0}$. Since we assume $r > d+1$ we can extend all points in $\mathcal{S}$ with one coordinate which is equal to 1 (i.e., we map the points from $\mathbb{R}^d$ to $\mathbb{R}^{d+1}$ by placing them in the hyperplane with last coordinate 1) and the extended points will still be linearly dependent over $\mathbb{R}^{d+1}$. Therefore (as this is the definition of linear independence) we can find coefficients c_i which are not all zero and satisfy

$$\sum_{i=0}^{r-1} c_i s_i = \mathbf{0} \tag{3.2}$$

as well as (this follows from the linear dependence in the additional coordinate)

$$\sum_{i=0}^{r-1} c_i = 0. \tag{3.3}$$

For any $\lambda \in \mathbb{R}$ define the function $\mathsf{P}_\lambda : \mathcal{S} \to \mathbb{R}$ as

$$\mathsf{P}_\lambda(s_i) := \mathsf{P}_S(s_i) + \lambda c_i.$$

The following equations then follow from (3.2) and (3.3) by linearity:

$$\sum_{i=0}^{r-1} \mathsf{P}_\lambda(s_i) = 1, \tag{3.4}$$

$$\sum_{i=0}^{r-1} \mathsf{P}_\lambda(s_i) s_i = \mathbf{0}, \tag{3.5}$$

$$\sum_{i=0}^{r-1} \mathsf{P}_\lambda(s_i) h(s_i) = \mathrm{E}[h(S)] + \lambda c, \tag{3.6}$$

where $c \in \mathbb{R}$ is some constant. We now note that there must exists numbers $\lambda^- < 0 < \lambda^+$ such that P_λ is a probability distribution exactly if $\lambda \in [\lambda^-, \lambda^+]$, and P_{λ^-} as well as P_{λ^+} are probability distributions with smaller support than P_S (for this we first use that not all c_i are zero, which means that some values $\mathsf{P}_\lambda(s_i)$ will eventually leave the interval $[0,1]$ as

λ is increased or decreased, and because of (3.4) first one value $\mathsf{P}_\lambda(s_i)$ will drop to zero; we use the corresponding λ).

Equations (3.5) and (3.6) now imply that either $\mathsf{P}_{\overline{S}} = \mathsf{P}_{\lambda^-}$ or $\mathsf{P}_{\overline{S}} = \mathsf{P}_{\lambda^+}$ reduces the support size under the conditions of the lemma, and we can use induction. □

We are now ready to give the main result of this section, namely that in Definition 3.1 it is sufficient to maximize over probability distributions with support size $|\mathcal{X}|$ for U and V. This theorem appears (without explicit proof) with slightly weaker parameters in [AC93] (see also [CK78]). However, the proof method employed here does not differ significantly from the one used in [AC93].

Theorem 3.3. *Let P_{UVXYZ} be a probability distribution over $\mathcal{U} \times \mathcal{V} \times \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ such that $\mathsf{P}_{UVXYZ} = \mathsf{P}_{XYZ} \cdot \mathsf{P}_{UV|X}$ for all u, v, x, y, z.*

There exists a conditional distribution $\mathsf{P}_{U'V'|X}$ such that

$$H(U'|ZV') - H(U'|YV') \geq H(U|ZV) - H(U|YV),$$

and such that the support of U' and V' is at most of size $|\mathcal{X}|$.

Proof. To simplify notation let $\mathcal{X} = \{x_0, \ldots, x_{r-1}\}$, i.e., $|\mathcal{X}| = r$. First, we use

$$H(U|ZV) - H(U|YV) = H(Z|UV) - H(Y|UV) - (H(Z|V) - H(Y|V)), \tag{3.7}$$

as noted after Definition 3.1. We start by reducing the alphabet size of V. For this, we rewrite the probability distribution as

$$\mathsf{P}_{UVXYZ}(u, v, x, y, z) = \mathsf{P}_V(v)\ \mathsf{P}_{U|V}(u|v)\ \mathsf{P}_{X|UV}(x|u, v)\ \mathsf{P}_{YZ|X}(y, z|x). \tag{3.8}$$

We now change P_V in (3.8) to $\mathsf{P}_{V'}$ and leave the other terms on the right hand side constant such that the alphabet size of V is reduced, and the distribution of X, Y, and Z stays the same, while (3.7) does not decrease. For this, we define $f : \mathcal{V} \to \mathbb{R}^{r-1}$

$$f(v) := \big(\mathsf{P}_{X|V}(x_0|v), \mathsf{P}_{X|V}(x_1|v), \ldots, \mathsf{P}_{X|V}(x_{r-2}|v)\big)$$

(note that we omit $\mathsf{P}_{X|V}(x_{r-1}|v)$). Further, define the function $h : \mathcal{V} \to \mathbb{R}$ as

$$h(v) := H(Z|U, V{=}v) - H(Y|U, V{=}v) - H(Z|V{=}v) + H(Y|V{=}v)$$

(note that $h(v)$ does *not* depend on P_V, only on $\mathsf{P}_{U|V}$, $\mathsf{P}_{X|UV}$ and $\mathsf{P}_{YZ|X}$). We apply Lemma 3.2 (we identify the points of $\mathcal{V}$ with points in $\mathbb{R}^{r-1}$ as given by the function f and apply the lemma on those points) and get a distribution $\mathsf{P}_{V'}$ over a subset of size r of $\mathcal{V}$. We define the new probability distribution

$$\begin{aligned}&\mathsf{P}_{U'V'XYZ}(u,v,x,y,z)\\ &\qquad := \mathsf{P}_{V'}(v)\cdot\mathsf{P}_{U|V}(u|v)\cdot\mathsf{P}_{X|UV}(x|u,v)\cdot\mathsf{P}_{YZ|X}(y,z|x). \qquad (3.9)\end{aligned}$$

Note that we keep $\mathsf{P}_{U|V}$, $\mathsf{P}_{X|UV}$, and $\mathsf{P}_{YZ|X}$ constant. Nevertheless, in general the distribution of U changes. However, $\mathrm{E}[f(V')] = \mathrm{E}[f(V)]$ implies that the distributions P_X and thus also P_{XYZ} stay the same. Further $\mathrm{E}[h(V')] \geq \mathrm{E}[h(V)]$ implies that (3.7) does not decrease if we use U' and V' instead of U and V.

Starting from (3.9), we now reduce the alphabet size of U' without changing the alphabet size of V'. We can do this for every fixed $v \in \mathcal{V}$ separately (this is sufficient because we can later rename the symbols in $\mathcal{U}$ such that they are the same for every v, which does not change any quantity we are interested in). Similarly to before, in order to reduce the alpha be size, we change $\mathsf{P}_{U|V=v}$ to $\mathsf{P}_{U''|V=v}$ in (3.9), while leaving the other terms constant. Define

$$f'_v(u) := \big(\mathsf{P}_{X|UV}(x_0|u,v), \mathsf{P}_{X|UV}(x_1|u,v), \ldots, \mathsf{P}_{X|UV}(x_{r-2}|u,v)\big),$$

and the function h'_v as

$$h'_v(u) := H(Z|U{=}u, V{=}v) - H(Y|U{=}u, V{=}v).$$

Again applying Lemma 3.2 (using the points of f'_v and h'_v) we obtain $\mathsf{P}_{U''|V'=v}$ such that U'' has support size at most $|\mathcal{X}|$. We consider the probability distribution

$$\begin{aligned}&\mathsf{P}_{U''V'XYZ}(u,v,x,y,z)\\ &\qquad := \mathsf{P}_{V'}(v)\cdot\mathsf{P}_{U''|V'}(u|v)\cdot\mathsf{P}_{X|UV}(x|u,v)\cdot\mathsf{P}_{YZ|X}(y,z|x). \qquad (3.10)\end{aligned}$$

Because $\mathrm{E}[f'_v(U)]$ is constant, this does not change $\mathsf{P}_{X|V'=v}$ and thus P_{XYZ}

also stays constant. Also we get

$$\begin{aligned}
&H(Z|U'V') - H(Y|U'V') - (H(Z|V') - H(Y|V')) \\
&\quad = \sum_{v\in\mathcal{V}} \mathsf{P}_{V'}(v)\Big(\mathrm{E}[h'_v(U')] - (H(Z|V'{=}v) - H(Y|V'{=}v))\Big) \\
&\quad \leq \sum_{v\in\mathcal{V}} \mathsf{P}_{V'}(v)\Big(\mathrm{E}[h'_v(U'')] - (H(Z|V'{=}v) - H(Y|V'{=}v))\Big) \\
&\quad = H(Z|U''V') - H(Y|U''V') - (H(Z|V') - H(Y|V')). \qquad \square
\end{aligned}$$

3.3. Information Reconciliation

This section describes information reconciliation, i.e., it shows how Alice and Bob can obtain a common string over which Eve has large min-entropy. For this we assume that Alice and Bob have instances of random variables which are distributed according to a distribution P_{XYZ} which satisfies $H(X|Z) > H(X|Y)$ (i.e., we ignore the preprocessing in this section).

3.3.1. Overview

Our information reconciliation protocol works as follows: Alice chooses a word $(d_0, \ldots, d_{n-1}) \in \mathcal{X}^n$ of an appropriately chosen error correcting code which we describe later. For each position i, Alice uses her random variable x_i and sends $x_i \oplus d_i$ to Bob (we identify $\mathcal{X}$ with $\{0, \ldots, |\mathcal{X}| - 1\}$, and use $\oplus$ to denote the addition modulo $|\mathcal{X}|$). The properties of the error correcting code will then ensure that with high probability Bob can find $(d_0, \ldots, d_{n-1})$, while Eve has large min-entropy about $(d_0, \ldots, d_{n-1})$. Alice and Bob can then use privacy amplification to obtain a key.

The code we use is such that a codeword can be decoded after sending it through the channel $\mathfrak{C}$ which maps an input $d \in \mathcal{X}$ to a pair $(d \oplus X, Y)$, where X and Y are chosen according to P_{XY} (note that this is exactly the information Bob obtains for one invocation). We quickly compute the capacity $\mathsf{Cap}(\mathfrak{C})$ of this channel. Recall (see, e.g.[CT91]) that the capacity of a channel from $\mathcal{S}$ to $\mathcal{T}$ specified by $\mathsf{P}_{T|S}$ equals $\max_{\mathsf{P}_S}(H(T) - H(T|S))$. In our case $H(T|S) = H(XY)$ independently of the distribution on the input, and it is easy to see that $H(T)$ is maximized by the uniform distri-

bution on the input. This gives

$$\mathsf{Cap}(\mathfrak{C}) = \max_{\mathsf{P}_S}\Big(\underbrace{H(T)}_{\log(|\mathcal{X}|)+H(Y)} - \underbrace{H(T|S)}_{H(XY)}\Big) = \log(|\mathcal{X}|) - H(X|Y). \quad (3.11)$$

(This implies that the capacity of the channel can be artificially increased by enlarging the alphabet $\mathcal{X}$. However, this does not achieve anything, as it would give Eve more information as well.)

In Section 3.3.2 we give codes with rate close to the capacity. In Section 3.3.3 we show that Eve's min-entropy about the codeword is large after the protocol above, assuming the rate of the code used is close enough to the capacity.

3.3.2. Error Correcting Codes

We now show how to construct codes for an arbitrary memoryless channel with rate arbitrarily close to the capacity. Our constructions will work for any channel which achieves the capacity on the uniform input distribution, i.e., a channel from $\mathcal{S}$ to $\mathcal{T}$ for which $H(T) - H(T|S)$ is maximized for the uniform distribution over the input $\mathcal{S}$.

Given a channel $\mathfrak{C}$ from $\mathcal{S}$ to $\mathcal{T}$, we write $\mathfrak{C}(s)$ to denote the random variable over $\mathcal{T}$ given by $\mathsf{P}_{T|S=s}$ as specified by the channel. Further, if $s^n \in \mathcal{S}^n$ we write $\mathfrak{C}^{(n)}(s^n)$ to denote the corresponding random variable over $\mathcal{T}^n$, i.e., we apply the channel independently n times.

To transmit information reliably over a channel, redundancy is added in a systematic way, as described by an error correcting code. Such a code is given by a *encoding function* $\mathcal{C}$ and a *decoding function* $\mathcal{D}$. The encoding function takes as input a bit string of appropriate length and outputs an n-tuple s^n over $\mathcal{S}^n$. After applying the channel n times we obtain $\mathfrak{C}^{(n)}(s^n)$, and from this the decoding function can find the initial bit string with very large probability.

Definition 3.4 (Error correcting code and rate). *An $(n, 2^k)$-error correcting code with error probability p_{err} for a channel $\mathfrak{C}$ from $\mathcal{S}$ to $\mathcal{T}$ is a pair of functions $\mathcal{C} : [2^k] \to \mathcal{S}^n$ and $\mathcal{D} : \mathcal{T}^n \to [2^k]$ such that for all $c \in [2^k]$: $\Pr[\mathcal{D}(\mathfrak{C}^{(n)}(\mathcal{C}(c))) = c] \geq 1 - p_{\text{err}}$. The rate of the code is $\frac{k}{n}$.*

The codes we study will be chosen at random from a family of codes. In this case we are interested in the error probability over the randomness of channel *and* the choice of the code. This is not usual, since in most

applications a single code with good properties is needed, and good *expected* performance is not sufficient. In our setting this is different: we can choose an instance of the code at random every time we need one.

Definition 3.5 (Family of error correcting codes). *A family of $(n, 2^k)$-error correcting codes with index set $\mathcal{A}$ and expected error probability p_{err} for the channel $\mathfrak{C}$ from $\mathcal{S}$ to $\mathcal{T}$ is a pair of function families $\mathcal{C}_{\mathcal{A}} : [2^k] \to \mathcal{S}^n$ and $\mathcal{D}_{\mathcal{A}} : \mathcal{T}^n \to [2^k]$ such that for P_A the uniform distribution over $\mathcal{A}$ and all $c \in [2^k]$: $\Pr[\mathcal{D}_A(\mathfrak{C}^{(n)}(\mathcal{C}_A(c))) = c] \geq 1 - p_{\text{err}}$ (where the probability is over the choice of A and the randomness of the channel).*

Codes with rate arbitrarily close to the capacity can be constructed in several ways. In this thesis we use two constructions. First, we use a construction which resembles *random linear codes* (this is similar to the random codes used by Shannon [Sha48]). Second, we concatenate these codes with a *Reed-Solomon code* (this construction is from Forney [For66]). Both these constructions are well known and studied in the literature, but usually only the asymptotic behavior of the error is studied. We will be interested in concrete values, and we study these codes with this in mind in the following.

Random linear codes

We use the following form of a random linear code[2] $\mathcal{C} : [2^k] \to \mathcal{S}^n$: First, identify $\mathcal{S}$ with $\{0, \ldots, |\mathcal{S}| - 1\}$ and let $\oplus$ be the addition modulo $|\mathcal{S}|$. Also, identify $[2^k]$ in an arbitrary way with the vector space $\mathrm{GF}(2)^k$ with basis $e_1, \ldots, e_k$, and, for every basis vector e_i, choose an image $\mathcal{C}(e_i)$ from $\mathcal{S}^n$ uniformly at random. Additionally, choose a vector c_0 from $\mathcal{S}^n$ uniformly at random. The image $\mathcal{C}(c)$ of a vector $c = \sum_{i=1}^k \lambda_i e_i$ with $\lambda_i \in \{0, 1\}$ is then given by $c_0 \oplus \bigoplus_{i=1}^k \lambda_i \mathcal{C}(e_i)$ (we shift the code by the random vector c_0 in order to make sure that $\mathcal{C}(\mathbf{0})$ is also chosen uniformly at random). For these codes we obtain the following theorem:

Theorem 3.6. *Let $\mathfrak{C}$ be a channel from $\mathcal{S}$ to $\mathcal{T}$ which achieves the capacity on the uniform input distribution. For any n, k, there exists a family of codes*

[2] Our construction does *not* give a linear code in a strict sense: if s_0 and s_1 are codewords, $s_0 \oplus s_1$ may not be a codeword. The reason we use this construction is that linear codes with good properties can only be constructed if $|\mathcal{S}|$ is a prime power. Nevertheless, our construction resembles the usual construction of linear codes, and also the resulting codes share the most important properties with random linear codes, which is why we still use this name.

$\mathcal{C}_a : [2^k] \to \mathcal{S}^n$ *with expected error probability*

$$p_{\text{err}} \leq 2^{\left(3 - \frac{n\varepsilon^2}{256 \log^2(|\mathcal{S}|\,|\mathcal{T}|)}\right)},$$

where $\varepsilon := \mathsf{Cap}(\mathfrak{C}) - \frac{k}{n}$. *Further,* $\mathcal{C}_a(\cdot)$ *can be computed in time* $\mathcal{O}(n^2)$, $\mathcal{D}_a(\cdot)$ *can be computed in time* $\mathcal{O}(n^2) \cdot 2^k$, *and the index set satisfies* $|\mathcal{A}| = (|\mathcal{S}|)^{n(k+1)}$.

Proof. We use a family of random linear codes as described above.

Consider the joint distribution P_{ST} defined by the uniform distribution P_S on $\mathcal{S}$ and the conditional distribution $\mathsf{P}_{T|S}$ as given by the channel $\mathfrak{C}$. For this distribution and a parameter $\delta > 0$ (which we later set to $\varepsilon/4$), we define the typical set

$$\begin{aligned} A_\delta^{(n)} := \Bigg\{ (s^n, t^n) \in \mathcal{S}^n \times \mathcal{T}^n \;\Bigg|\; & \left| \frac{1}{n} \log\Big(\frac{1}{\mathsf{P}_{S^nT^n}(s^n, t^n)}\Big) - H(ST) \right| \leq \delta \wedge \\ & \left| \frac{1}{n} \log\Big(\frac{1}{\mathsf{P}_{S^n}(s^n)}\Big) - H(S) \right| \leq \delta \wedge \\ & \left| \frac{1}{n} \log\Big(\frac{1}{\mathsf{P}_{T^n}(t^n)}\Big) - H(T) \right| \leq \delta \Bigg\}. \end{aligned}$$

Let $\mathcal{C}_a$ be a randomly chosen linear code as described above. On received word $t^n \in \mathcal{T}^n$ the receiver enumerates all words $c \in [2^k]$ and checks whether $(\mathcal{C}_a(c), y^n) \in A_\delta^{(n)}$. In case there is a unique codeword c which satisfies this, the decoder outputs this word. Otherwise an arbitrary codeword is returned. The running time of the encoder is clearly bounded by $\mathcal{O}(n^2)$, and the running time of the decoder can be bounded by $\mathcal{O}(n^2) \cdot 2^k$.

Let now c be the codeword chosen by Alice. Since by construction $\mathcal{C}_A(c)$ is chosen according to P_{S^n} (i.e., the uniform distribution), Proposition 2.11 implies that the probability $\Pr[(S^n, T^n) \notin A_\delta^{(n)}]$ is at most $6 \cdot 2^{-\frac{n\delta^2}{16 \log^2(|\mathcal{S}|\,|\mathcal{T}|)}}$.

On the other hand, let now c' be a codeword which was *not* sent, i.e., $c' \neq c$. In this case, our choice of the code implies that $\mathcal{C}_A(c')$ and T^n are both chosen independently with the marginals of P_{ST}. The probability that $(\mathcal{C}_A(c'), T^n) \in A_\delta^{(n)}$ is at most $|A_\delta^{(n)}| \cdot 2^{-nH(S) - nH(T) + 2n\delta}$. From the definition of $A_\delta^{(n)}$ it is easy to see that $|A_\delta^{(n)}| \leq 2^{nH(ST)+\delta}$ and together we

obtain

$$\Pr[(\mathcal{C}_A(c'), T^n) \in A_\delta^{(n)}] \leq 2^{-n\mathsf{Cap}(\mathfrak{C})+3n\delta} = 2^{-n(\varepsilon-3\delta)-k}.$$

Setting $\delta := \varepsilon/4$ and using the union bound we obtain:

$$\begin{aligned}\Pr[(\mathcal{C}_A(c), T^n) \notin A_\delta^{(n)} \vee \exists c' \neq c : (\mathcal{C}_A(c'), T^n) \in A_\delta^{(n)}] \\ &\leq 6 \cdot 2^{-\frac{n\varepsilon^2}{256\log^2(|\mathcal{S}||\mathcal{T}|)}} + 2^{-\frac{n\varepsilon}{4}} \\ &\leq 7 \cdot 2^{-\frac{n\varepsilon^2}{256\log^2(|\mathcal{S}||\mathcal{T}|)}},\end{aligned}$$

where we used $\varepsilon \leq \log(|\mathcal{S}|\,|\mathcal{T}|)$ in the last step. Finally, by inspection of the construction above we get $|\mathcal{A}| = (|\mathcal{S}|)^{n(k+1)}$. □

Concatenated codes

Random linear codes have the disadvantage that we do not know how to decode a noisy codeword efficiently. Codes which do not have this drawback were first proposed by Forney [For66]. He combined random codes with algebraic codes and obtained codes which were efficiently decodable, while still achieving a rate arbitrarily close to the capacity. We can use this technique to get the following theorem:

Theorem 3.7. *Let $\mathfrak{C}$ be a channel from $\mathcal{S}$ to $\mathcal{T}$ which achieves the capacity on the uniform input distribution. Set $C := \mathsf{Cap}(\mathfrak{C})$ and $d := 2^{20}\log^2(|\mathcal{S}||\mathcal{T}|)$. If k and n, $k < Cn$ are such that for $\varepsilon := C - \frac{k}{n}$ the inequalities*

$$\frac{C}{\varepsilon} > 2 \quad \textit{and} \quad n \geq \left(\frac{C}{\varepsilon}\right)^{\frac{d}{\varepsilon^2}}$$

are satisfied, then there exists a family of codes $\mathcal{C} : [2^k] \to \mathcal{S}^n$ which can be encoded and decoded in time $\mathcal{O}(n^2)$ with expected error probability

$$p_{\text{err}} < 2^{-\frac{n\varepsilon^4}{dC^2}}$$

for the channel $\mathfrak{C}$. Further, the index set satisfies $|\mathcal{A}| < (|\mathcal{S}|)^{n^2}$.

The idea is that instead of using one random linear code, many small instances are used, in order to keep the task of decoding manageable. These blocks are then combined using an algebraic code which can be decoded efficiently. In this thesis we use a (shortened) Reed-Solomon

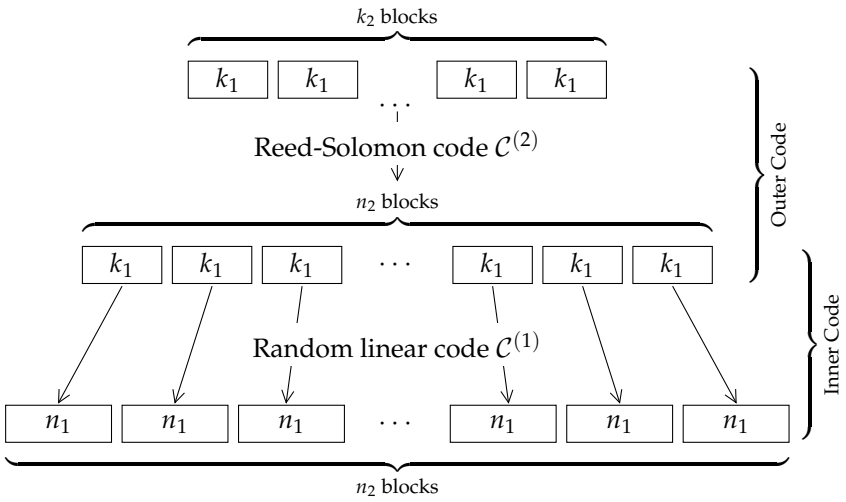


Figure 3.4.: Concatenation of a random linear code $\mathcal{C}^{(1)}$ with a Reed-Solomon code $\mathcal{C}^{(2)}$.

code, introduced in [RS60], for this. A Reed-Solomon code is defined as follows over any finite field $\mathrm{GF}(p)$: The function $\mathcal{C} : \mathrm{GF}(p)^k \to \mathrm{GF}(p)^n$ is given by interpreting the input as description of a polynomial over $\mathrm{GF}(p)$ of degree $k-1$, and then evaluating this polynomial at n points. There are highly developed algorithms to correct errors in such a code [Ber68, Mas69, Jus76, WB86]. We use the following proposition whose proof can be found in several of these references:

Proposition 3.8. *Let $n, k, p \in \mathbb{N}$ with $k < n \leq p$ and p a prime power be given. There exists a code $\mathcal{C} : [p^k] \to [p^n]$ which can be encoded in time $\mathcal{O}(p^2)$, such that for $\varepsilon := 1 - \frac{k}{n}$ a codeword with less than fraction $\varepsilon/2$ errors, can be decoded correctly in time $\mathcal{O}(p^2)$.*

In order to concatenate the Reed-Solomon code with a linear code, we need a stronger version of the Hoeffding bound. Compared with Proposition 2.13 an additional logarithmic term appears in the exponent. A proof is given in [Hoe63].

Proposition 3.9. *Let $\mathsf{P}_{X_0 X_1 \cdots X_{r-1}} = \mathsf{P}_{X_0} \mathsf{P}_{X_1} \cdots \mathsf{P}_{X_{r-1}}$ be a product distribu-*

tion with $X_i \in [0,1]$. Let $\overline{X} := \frac{1}{r}\sum_{i=0}^{r-1} X_i$. Then, for any $\varepsilon > 0$,

$$\Pr\left[\overline{X} \geq \mathrm{E}[\overline{X}] + \varepsilon\right] \leq e^{-r\varepsilon^2 \ln\left(\frac{1}{\mathrm{E}[\overline{X}]}\right)} < 2^{-r\varepsilon^2 \log\left(\frac{1}{\mathrm{E}[\overline{X}]}\right)}.$$

We can now concatenate random linear codes as inner codes with a Reed-Solomon code as outer code: the input is interpreted as bit string and split into blocks, then encoded using an appropriate Reed-Solomon code, and then every of the resulting blocks is encoded with a random linear code (see Figure 3.4). We choose every instance of the random linear code independently of the other instances.

The following Lemma describes when two such codes can be concatenated. In this Lemma, k_1, n_1 and ε_1 will denote the parameters of the (inner) random linear code $\mathcal{C}^{(1)}$, while conversely k_2, n_2 and ε_2 will denote the parameters of the (outer) Reed-Solomon code $\mathcal{C}^{(2)}$. The parameters have to be provided in order to use this lemma.

Lemma 3.10. *Let $\mathfrak{C}$ be any channel from $\mathcal{S}$ to $\mathcal{T}$ which achieves the capacity on the uniform input distribution. Let $k_1, n_1, k_2, n_2 \in \mathbb{N}$ be given. Set $\varepsilon_1 := \mathsf{Cap}(\mathfrak{C}) - \frac{k_1}{n_1}$ and $\varepsilon_2 := 1 - \frac{k_2}{n_2}$, $d := 8192 \log^2(|\mathcal{S}||\mathcal{T}|)$. If these parameters satisfy $k_1 < \mathsf{Cap}(\mathfrak{C})n_1$, $\frac{n_2}{2} < k_2 < n_2 \leq 2^{k_1}$, and $n_1 \geq \frac{d}{\varepsilon_1^2}\log(\frac{1}{\varepsilon_2})$, then there exists a family of codes $\mathcal{C}_A : [2^{k_1 k_2}] \to \mathcal{S}^{n_1 n_2}$, which can be encoded and decoded in time $\mathcal{O}(n_1^2 n_2 2^{k_1} + 2^{2k_1})$ and which has expected error probability*

$$p_{\mathrm{err}} < 2^{-\frac{n_1 n_2 \varepsilon_1^2 \varepsilon_2^2}{d}}$$

for the channel $\mathfrak{C}$. Further, the index set satisfies $|\mathcal{A}| = (|\mathcal{S}|)^{n_1 n_2 (k_1+1)}$.

Proof. Let $\mathcal{C}_A^{(1)} : [2^{k_1}] \to \mathcal{S}^{n_1}$ be a family of random linear codes. We concatenate this code with a Reed-Solomon code $\mathcal{C}^{(2)} : [(2^{k_1})^{k_2}] \to [(2^{k_1})^{n_2}]$ from Proposition 3.8. This means that the resulting code $\mathcal{C}_a$ first splits the input of length $k_1 k_2$ into k_2 blocks of length k_1, and then encodes these blocks with a Reed-Solomon code as given by Proposition 3.8, by interpreting the blocks as elements of $\mathrm{GF}(2^{k_1})$. We can use Proposition 3.8 in that way because $k_2 < n_2 \leq 2^{k_1}$. Subsequently, every block (i.e., every element of $\mathrm{GF}(2^{k_1})$) is encoded using a random linear code, where we take an independently chosen random code for every block.

According to Theorem 3.6, the probability that a fixed block is not decoded correctly is at most (observe that $n_1 \geq \frac{d}{\varepsilon_1^2}\log(\frac{1}{\varepsilon_2})$ and $\varepsilon_2 \leq \frac{1}{2}$ imply

$\frac{n_1\varepsilon_1^2}{512\log^2(|\mathcal{S}||\mathcal{T}|)} \geq 16\log(\frac{1}{\varepsilon_2}) \geq 16$):

$$2^{3-\frac{n_1\varepsilon_1^2}{256\log^2(|\mathcal{S}||\mathcal{T}|)}} \leq 2^{-\frac{n_1\varepsilon_1^2}{512\log^2(|\mathcal{S}||\mathcal{T}|)}} \leq \frac{\varepsilon_2}{4}. \tag{3.12}$$

Since Proposition 3.8 implies that we can decode correctly if there are less than fraction $\frac{\varepsilon_2}{2}$ errors, we can only get a decoding error if the fraction of wrongly decoded blocks deviates by more than $\frac{\varepsilon_2}{4}$ from the expected value μ. According to Proposition 3.9, the probability of this event is at most (where we use $\mu \leq 2^{-\frac{n_1\varepsilon_1^2}{512\log^2(|\mathcal{S}||\mathcal{T}|)}}$, as implied by (3.12)):

$$p_{\text{err}} \leq 2^{-n_2\frac{\varepsilon^2}{16}\log(\frac{1}{\mu})} \leq 2^{-\frac{n_1 n_2\varepsilon_1^2\varepsilon_2^2}{8192\log^2(|\mathcal{S}||\mathcal{T}|)}}.$$

To encode a word first the encoding of a Reed Solomon code is needed, which takes time $\mathcal{O}(2^{2k_1})$. Then n_2 blocks are encoded in time $\mathcal{O}(n_1^2)$ each, which gives a total time of $\mathcal{O}(n_1^2 n_2 + 2^{2k_1})$. To decode a noisy word, first n_2 blocks are decoded in time $\mathcal{O}(n_1^2 2^{k_1})$ each, then the Reed Solomon code is decoded in time $\mathcal{O}(2^{2k_1})$, which gives a total time of $\mathcal{O}(n_1^2 n_2 2^{k_1} + 2^{2k_1})$ to decode. The size of the index set follows directly from the definition of the code. □

For appropriately chosen parameters, Lemma 3.10 provides what we need: an efficiently decodable code for any channel $\mathfrak{C}$ from $\mathcal{S}$ to $\mathcal{T}$ with rate arbitrarily close to the capacity. Unfortunately, the parameters in Lemma 3.10 are rather hard to handle, as everything needs to be provided properly. It is more convenient to have a theorem where one only n and k needs to be specified. Doing this we arrive at Theorem 3.7.

Proof (of Theorem 3.7). Given n, it is possible to choose numbers n_1 and n_2 such that

$$n_1 n_2 < n < n_1 n_2\big(1 + \frac{\varepsilon}{3C}\big) \text{ and}$$
$$\frac{3\log(n_2)}{C} < n_1 < \frac{4\log(n_2)}{C}$$

are satisfied. Further, we chose $k_1, k_2 \in \mathbb{N}$, and thus $\varepsilon_1 := C - \frac{k_1}{n_1}$ and

$\varepsilon_2 := 1 - \frac{k_2}{n_2}$, such that

$$\frac{\varepsilon}{4} < \varepsilon_1 < \frac{\varepsilon}{3},$$
$$\frac{\varepsilon}{4C} < \varepsilon_2 < \frac{\varepsilon}{3C},$$

which is always possible for the numbers we use. We will construct a code $[2^{k_1 k_2}] \to \mathcal{S}^{n_1 n_2}$, and since $n_1 n_2 < n$ and

$$\begin{aligned} k_1 k_2 &= (n_1 C - n_1 \varepsilon_1)(n_2 - n_2 \varepsilon_2) \\ &> n_1 n_2 C - n_1 n_2 \varepsilon_1 - n_1 n_2 \varepsilon_2 C \\ &> n_1 n_2 (C - \frac{2\varepsilon}{3}) \\ &> n(C - \frac{2\varepsilon}{3})/(1 + \varepsilon/3C) \\ &> n(C - \varepsilon) = nk, \end{aligned}$$

this is sufficient.

We now check that we can use Lemma 3.10 for these parameters. We first note that $n_2 > k_2$, $Cn_1 > k_1$ and also $\frac{n_2}{2} < k_2$ (the last one because $\varepsilon_2 < \frac{\varepsilon}{3C}$ and $\varepsilon < C/2$). Further, $\varepsilon < \frac{C}{2}$ also implies

$$2^{k_1} = 2^{n_1(C-\varepsilon_1)} > 2^{\frac{Cn_1}{2}} > 2^{\log n_2} = n_2.$$

Finally, the condition $n \geq (\frac{C}{\varepsilon})^{\frac{d}{\varepsilon^2}}$ implies $2n_1 n_2 \geq (\frac{C}{\varepsilon})^{\frac{d}{\varepsilon^2}}$ and thus, using $\frac{4\log(n_1)}{C} > n_2$, $2n_2 2^{\frac{4n_2}{C}} \geq (\frac{C}{\varepsilon})^{\frac{d}{\varepsilon^2}}$. Taking the logarithm on both sides gives $\frac{4n_2}{C} + \log(n_2) + 1 \geq \frac{d}{\varepsilon^2}\log(\frac{C}{\varepsilon})$ and this implies $n_2 \geq \frac{Cd}{8\varepsilon^2}\log(\frac{C}{\varepsilon}) \geq \frac{Cd}{128\varepsilon_2^2}\log(\frac{1}{4\varepsilon_1}) \geq \frac{8192\log^2(|\mathcal{S}|\,|\mathcal{T}|)}{\varepsilon_2^2}\log(\frac{1}{\varepsilon_1})$.

The error probability and the run time of the algorithms also follow from Lemma 3.10. □

3.3.3. A Bound on Eve's Knowledge

In the information reconciliation protocol we use, Alice chooses a codeword from an appropriate error correcting code and then sends the pointwise sum of her random variables and the codeword to Bob. We want to show that given Eve's random variables and this communication, the min-entropy over the codeword is still large with high probability. This will follow from the following lemma (adapted from [RW05, Cac97]).

Lemma 3.11. *Let P_{UVW} be a probability distribution over $\mathcal{U} \times \mathcal{V} \times \mathcal{W}$. For any $\varepsilon \geq 0$ and $\varepsilon' > 0$:*

$$H_\infty^{\varepsilon+\varepsilon'}(U|VW) \geq H_\infty(U|W) + H_\infty^{\varepsilon}(V|UW) - H_0(V|W) - \log\Big(\frac{1}{\varepsilon'}\Big).$$

The intuition is that this is similar to the common equality $H(U|VW) = H(U|W) + H(V|UW) - H(V|W)$ which holds for Shannon entropy.

In our application we can use this lemma as follows: we set U to be the codeword chosen by Alice, V to the communication produced by Alice, and W to the *a priori* information Eve has, i.e., $(Z_1 \dots Z_n)$. If the rate of the code used is R, the quantities on the right hand side can then be bounded as follows:

$$\begin{aligned} H_\infty(U|W) &\geq nR \\ H_\infty^{\varepsilon}(V|UW) &\gtrapprox nH(X|Z) \\ H_0(V|W) &\leq n\log(|\mathcal{X}|), \end{aligned}$$

where the second bound follows because we use independent repetitions (i.e., using Corollary 2.12). If the code has rate close enough to the capacity $\log(|\mathcal{X}|) - H(X|Y)$ (as given in equation (3.11) on page 36) of the corresponding channel, and if $H(X|Z) > H(X|Y)$, then the min-entropy of Eve will grow linearly in n.

Proof. We prove the following two inequalities. The lemma then follows by inserting (3.14) into (3.13).

$$H_\infty^{\varepsilon+\varepsilon'}(U|VW) \geq H_\infty^{\varepsilon}(UV|W) - H_0(V|W) - \log\Big(\frac{1}{\varepsilon'}\Big) \tag{3.13}$$

$$H_\infty^{\varepsilon}(UV|W) \geq H_\infty(U|W) + H_\infty^{\varepsilon}(V|UW). \tag{3.14}$$

Proof of (3.13): Let $\mathsf{P}_{U'V'W'}$ be a distribution over $\mathcal{U} \times \mathcal{V} \times \mathcal{W}$ with $\|\mathsf{P}_{UVW} - \mathsf{P}_{U'V'W'}\| \leq \varepsilon$ which satisfies $H_\infty(U'V'|W') = H_\infty^{\varepsilon}(UV|W)$ and $H_0(V'|W') \leq H_0(V|W)$ (note that $H_0(V'|W') \leq H_0(V|W)$ is not a restriction on the maximization, as it is always sufficient to enlarge the alphabet of U). We show

$$\begin{aligned} H_\infty^{\varepsilon+\varepsilon'}(U|VW) &\overset{(1)}{\geq} H_\infty^{\varepsilon'}(U'|V'W') \\ &\overset{(2)}{\geq} H_\infty(U'V'|W') - H_0(V'|W') - \log\Big(\frac{1}{\varepsilon'}\Big) \\ &\overset{(3)}{\geq} H_\infty^{\varepsilon}(UV|W) - H_0(V|W) - \log\Big(\frac{1}{\varepsilon'}\Big). \end{aligned}$$

Inequality (1) follows from the fact that any distribution within statistical distance ε' from $\mathsf{P}_{U'V'W'}$ is also within statistical distance $\varepsilon + \varepsilon'$ from P_{UVW}, and thus the maximization on the right hand side is over a subset of the maximization on the left hand side.

For inequality (2) let

$$\mathcal{S}_{\varepsilon'} := \{(v, w) \in \mathcal{V} \times \mathcal{W} | \mathsf{P}_{V'|W'}(v|w) \leq \varepsilon' 2^{-H_0(V'|W')}\},$$

and define the distribution $\mathsf{P}_{U''V''W''}$ as

$$\mathsf{P}_{U''V''W''}(u, v, w) := \begin{cases} \mathsf{P}_{U'V'W'}(u, v, w) & \text{if } (v, w) \notin \mathcal{S}_{\varepsilon'} \\ \frac{1}{|\mathcal{X}|}\mathsf{P}_{V'W'}(v, w) & \text{if } (v, w) \in \mathcal{S}_{\varepsilon'}. \end{cases}$$

Since

$$\begin{aligned} \Pr[(V', W') \in \mathcal{S}_{\varepsilon'}] &= \sum_{(v,w)\in\mathcal{S}_{\varepsilon'}} \mathsf{P}_{W'}(w)\, \mathsf{P}_{V'|W'}(v|w) \\ &\leq \sum_{(v,w)\in\mathcal{S}_{\varepsilon'}} \mathsf{P}_{W'}(w)\, \varepsilon'\, 2^{-H_0(V'|W')} \\ &\leq \sum_{w\in\mathcal{W}} \mathsf{P}_{W'}(w) 2^{H_0(V'|W'=w)}\, \varepsilon'\, 2^{-H_0(V'|W')} \\ &\leq \varepsilon', \end{aligned}$$

we get $\|\mathsf{P}_{U'V'W'} - \mathsf{P}_{U''V''W''}\| \leq \varepsilon'$. Thus, for inequality (2) it is sufficient to show that $H_\infty(U''|V''W'') \geq H_\infty(U'V'|W') - H_0(V'|W') - \log(1/\varepsilon')$, or, equivalently, for all $(u, v, w) \in \mathcal{U} \times \mathcal{V} \times \mathcal{W}$:

$$\mathsf{P}_{U''|V''W''}(u|v, w) \leq 2^{-H_\infty(U'V'|W')+H_0(V'|W')+\log(\frac{1}{\varepsilon'})}.$$

For $(v, w) \in \mathcal{S}_{\varepsilon'}$ this is clear since $H_\infty(U''|V''{=}v, W''{=}w)$ is maximal. For $(v, w) \notin \mathcal{S}_{\varepsilon'}$ we get

$$\begin{aligned} \mathsf{P}_{U''|V''W''}(u|v, w) = \frac{\mathsf{P}_{U''V''|W''}(u, v|w)}{\mathsf{P}_{V''|W''}(v|w)} &< \frac{2^{-H_\infty(U'V'|W')}}{\varepsilon'\, 2^{-H_0(V'|W')}} \\ &= 2^{-H_\infty(U'V'|W')+H_0(V'|W')+\log(\frac{1}{\varepsilon'})}. \end{aligned}$$

Finally, (3) follows from $H^\varepsilon_\infty(UV|W) = H_\infty(U'V'|W')$ and $H_0(V|W) \geq H_0(V'|W')$.

Proof of (3.14): Let $\mathsf{P}_{U'V'W'}$ be a distribution with $\|\mathsf{P}_{UVW} - \mathsf{P}_{U'V'W'}\| \leq \varepsilon$ and $H^\varepsilon_\infty(V|UW) = H_\infty(V'|U'W')$. Such a distribution can be found

even if we require that $H_\infty(U'|W') \geq H_\infty(U|W)$ is satisfied because the conditional probabilities $\mathsf{P}_{U|W}$ can be held constant.

We will show that

$$\begin{aligned} H_\infty^\varepsilon(UV|W) &\overset{(4)}{\geq} H_\infty(U'V'|W) \\ &\overset{(5)}{\geq} H_\infty(U'|W') + H_\infty(V'|U'W') \\ &\overset{(6)}{\geq} H_\infty(U|W) + H_\infty^\varepsilon(V|UW). \end{aligned}$$

Inequality (4) is immediate from $\|\mathsf{P}_{UVW} - \mathsf{P}_{U'V'W'}\| \leq \varepsilon$. Inequality (5) is equivalent to

$$\begin{aligned} 2^{-H_\infty(U'V'|W')} &= \max_{(u,v,w)\in\mathcal{U}\times\mathcal{V}\times\mathcal{W}} \mathsf{P}_{U'V'|W'}(u,v|w) \\ &= \max_{(u,v,w)\in\mathcal{U}\times\mathcal{V}\times\mathcal{W}} \Big(\mathsf{P}_{U'|W'}(u|w)\cdot \mathsf{P}_{V'|U'W'}(v|u,w)\Big) \\ &\leq \Big(\max_{(u,w)\in\mathcal{U}\times\mathcal{W}} \mathsf{P}_{U'|W'}(u|w)\Big)\times \\ &\qquad \Big(\max_{(u,v,w)\in(\mathcal{U}\times\mathcal{V}\times\mathcal{W})} \mathsf{P}_{V'|U'W'}(v|u,w)\Big) \\ &= 2^{-H_\infty(U'|W')-H_\infty(V'|U'W')}. \end{aligned}$$

Finally, (6) follows from the definition of $\mathsf{P}_{U'V'W'}$. □

3.4. The Protocol

We now combine the three steps (preprocessing, information reconciliation, and privacy amplification) into a single protocol. Depending on the code we use to do information reconciliation the protocol will have different properties. We first give the protocol where without specifying a specific code and later insert the different codes we have into that protocol. For the following theorem recall that $S_\rightarrow(X;Y|Z)$ is the one-message secret key rate (see Definition 3.1 on page 30).

3.4.1. The General Protocol

Lemma 3.12. *Let P_{XYZ} be a probability distribution over $\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}$ and $\mathsf{P}_{UV|X}$ a conditional probability distribution over $\mathcal{X}\times\mathcal{X}\times\mathcal{X}$ which maximizes $H(U|ZV) - H(U|YV)$.*

Let n, k, and a code $\mathcal{C} : [2^k] \to \mathcal{X}^n$ be given for the channel $\mathfrak{C}$ which, on input $x \in \mathcal{X}$, chooses U, V, and Y according to P_{UVY} and outputs $(x \oplus U, V, Y)$. Assume that $\mathcal{C}$ has rate $\frac{k}{n} = \log(|\mathcal{X}|) - H(U|YV) - \varepsilon$ and decoding error probability γ. Further, let $\kappa_2 > \frac{2}{n}$ be given.

There exists a one-message key agreement protocol which uses n random variables and yields a key with

$$\lfloor n\big(S_{\rightarrow}(X;Y|Z) - \varepsilon - 8\log(|\mathcal{X}|)\sqrt{\kappa_2}\big)\rfloor$$

bits, soundness $1-\gamma$ and secrecy $1-2^{-n\kappa_2}$. The protocol has the properties that Alice sends at most $3n\lceil\log(|\mathcal{X}|)\rceil$ bits, Alice encodes one word of $\mathcal{C}$, and Bob decodes one noisy word of $\mathcal{C}$.

The protocol needs inputs κ_2, n, and a description of $\mathsf{P}_{UV|X}$.

Proof. First, Alice does the preprocessing, i.e., using the supplied channel $\mathsf{P}_{UV|X}$ for every random variable X_i, $0 \le i < n$, she computes U_i and V_i from X_i, sends V_i to Bob and keeps U_i.

Next, Alice chooses a random word $D \in [2^k]$ from $\mathcal{C}$ and sends, for every i, $(\mathcal{C}(D))_i \oplus U_i$ to Bob. Bob now has $\big((\mathcal{C}(D))_i \oplus U_i, V_i, Y_i\big)$ for every i which is exactly the information he gets in an application of the channel $\mathfrak{C}$. This implies that he can decode the received word and finds the original codeword D with probability $1-\gamma$. Alice then sends a randomly chosen seed of a two-universal hash-function which maps the codeword to a string of length $\lfloor n\big(S_{\rightarrow}(X;Y|Z) - \varepsilon - 8\log(|\mathcal{X}|)\sqrt{\kappa_2}\big)\rfloor$. Both parties apply the hash-function (Alice to D, and Bob to the recovered version of D) and output S_A and S_B, respectively. It is clear that the protocol satisfies $\Pr[S_A \neq S_B] \le \gamma$.

We now show that the resulting key has secrecy $1-2^{-n\kappa_2}$. For this, we first note that Lemma 3.11, implies, for any $\delta > 0$:

$$\begin{aligned} H_\infty^{2\delta}(D|(U^n \oplus \mathcal{C}(D))Z^nV^n) &\ge \underbrace{H_\infty(D|Z^nV^n)}_{=k} + \underbrace{H_\infty^\delta(U^n \oplus \mathcal{C}(D)|DZ^nV^n)}_{=H_\infty^\delta(U^n|Z^nV^n)} \\ &\quad - \underbrace{H_0(U^n \oplus \mathcal{C}(D)|Z^nV^n)}_{\le n\log(|\mathcal{X}|)} - \log\Big(\frac{1}{\delta}\Big) \\ &\ge k + H_\infty^\delta(U^n|Z^nV^n) - n\log(|\mathcal{X}|) - \log\Big(\frac{1}{\delta}\Big). \end{aligned}$$

Furthermore, from Corollary 2.12 we get $H_\infty^\delta(U^n|Z^nV^n) \ge nH(U|ZV) - 4\sqrt{n\log(\frac{1}{\delta})}\log(|\mathcal{X}|)$. Together with $k = n(\log(|\mathcal{X}|) - H(U|YV) - \varepsilon)$ we

obtain

$$H_\infty^{2\delta}(D|(U^n \oplus \mathcal{C}(D))Z^nV^n) \geq n(H(U|ZV) - H(U|YV) - \varepsilon) \\ - 4\sqrt{n\log\Big(\frac{1}{\delta}\Big)}\log(|\mathcal{X}|) - \log\Big(\frac{1}{\delta}\Big).$$

Using Theorem 2.17 (for closeness δ, giving entropy loss $2\log(\frac{1}{\delta})$) we see that we can get a string of length

$$\lfloor n(S_\rightarrow(X;Y|Z) - \varepsilon) - 4\sqrt{n\log(\frac{1}{\delta})}\log(|\mathcal{X}|) - 3\log(\frac{1}{\delta})\rfloor$$

which is 3δ-close to uniform with respect to Z^n, V^n, and the communication. We can now set $\delta := 2^{-2n\kappa_2}$ (because $\kappa_2 > \frac{2}{n}$ this suffices to ensure that the output has secrecy $1 - 2^{-n\kappa_2} > 1 - 3\delta$) which implies that the key has length at least

$$\Big\lfloor n\Big(S_\rightarrow(X;Y|Z) - \varepsilon - 4\sqrt{2\kappa_2}\log(|\mathcal{X}|) - 6\kappa_2\Big)\Big\rfloor.$$

Because $S_\rightarrow(X;Y|Z) \leq \log(|\mathcal{X}|)$ we can see that if $\kappa_2 \geq \frac{1}{64}$ the statement only guarantees a key of length 0. We thus assume $\kappa_2 < \frac{1}{64}$, from which we see $\sqrt{\kappa_2} < \frac{1}{8}$ and thus $6\kappa_2 < \frac{3}{4}\sqrt{\kappa_2}$, which implies the lemma. □

3.4.2. The Protocol Using a Random Linear Code

We first use the random linear code from Theorem 3.6 in our protocol.

Theorem 3.13. *Let P_{XYZ} be a probability distribution over $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. Let parameters n, $\kappa_1 > \frac{2}{n}$ and $\kappa_2 > \frac{2}{n}$ be given. There exists a one-message key agreement protocol which uses n random variables and yields a key with at least*

$$\Big\lfloor n\Big(S_\rightarrow(X;Y|Z) - 90\log(|\mathcal{X}|\,|\mathcal{Y}|)(\sqrt{\kappa_1} + \sqrt{\kappa_2})\Big)\Big\rfloor$$

key bits, soundness $1 - 2^{-n\kappa_1}$, and secrecy $1 - 2^{-n\kappa_2}$. In the protocol, Alice sends $\mathcal{O}(n^2)$ bits to Bob, the computations of Alice can be done in time $\mathcal{O}(n^2)$, and the computations of Bob can be done in time $\mathcal{O}(n^2)|\mathcal{X}|^n$.

The protocol of Theorem 3.13 is the same for any distribution P_{XYZ} as long as a description of P_{XY} and a description of $\mathsf{P}_{UV|X}$ maximizing $S_\rightarrow(X;Y|Z)$ are supplied as input.

Proof. We combine the protocol from Lemma 3.12 with the random linear code from Theorem 3.6.

Let $\mathsf{P}_{UV|X}$ be a distribution over $\mathcal{X} \times \mathcal{X} \times \mathcal{X}$ for which $H(U|ZV) - H(U|YV)$ is maximal. Let $\mathfrak{C}$ be the channel which maps an input d to a triple $(d \oplus U, V, Y)$, where (U, V, Y) is chosen according to P_{UVY}. Let k be

$$k := \left\lfloor n\big(\mathsf{Cap}(\mathfrak{C}) - 28\sqrt{\kappa_1}\log(|\mathcal{X}|^3\,|\mathcal{Y}|)\big)\right\rfloor$$

(this implies $28\sqrt{\kappa_1}\log(|\mathcal{X}|^3\,|\mathcal{Y}|) \leq \varepsilon \leq 30\sqrt{\kappa_1}\log(|\mathcal{X}|^3\,|\mathcal{Y}|)$ for $\varepsilon = \mathsf{Cap}(\mathfrak{C}) - \frac{k}{n}$). For this k, Alice chooses a random linear code $\mathcal{C} : [2^k] \to \mathcal{X}^n$ for $\mathfrak{C}$ and sends a description of this code to Bob. Alice uses this code and parameter κ_2 in the protocol guaranteed by Lemma 3.12.

From Theorem 3.6 we get that the probability that the code is decoded incorrectly is less than $2^{3-3n\kappa_1} < 2^{-n\kappa_1}$ (where we used $2n\kappa_1 > 4$), and Lemma 3.12 implies that the probability that S_A is not equal to S_B is at most this value. Also, Lemma 3.12 states that the protocol has secrecy $1 - 2^{-n\kappa_2}$.

The length of the key is at least $\lfloor n(S_{\to}(X;Y|Z) - \varepsilon - 8\log(|\mathcal{X}|)\sqrt{\kappa_2})\rfloor \geq \lfloor n\big(S_{\to}(X;Y|Z) - 90\log(|\mathcal{X}|\,|\mathcal{Y}|)(\sqrt{\kappa_1} + \sqrt{\kappa_2})\big)\rfloor$. Finally, the runtime and amount of complexity are easy to check. □

For example, assume that a fixed distribution P_{XYZ} and a rate $R < S_{\to}(X;Y|Z)$ is given. We can then fix a constant κ depending on R such that the protocol uses n random variables and yields nR secret bits with soundness and secrecy $1 - 2^{-n\kappa}$. In other words, we can get a key with exponential security for every fixed rate smaller than $S_{\to}(X;Y|Z)$.

Corollary 3.14. *Let P_{XYZ} be a probability distribution and R a constant satisfying $R < S_{\to}(X;Y|Z)$. There exists a constant c (depending on P_{XYZ} and R) and a one-message key agreement protocol (also depending on P_{XYZ} and R) which, on input n, uses n random variables and yields a key with at least $\lfloor nR\rfloor$ bits, soundness $1 - 2^{-cn}$ and secrecy $1 - 2^{-cn}$.*

Proof. From Theorem 3.13, choosing κ_1 and κ_2 accordingly and hard coding the description of P_{XY} and $\mathsf{P}_{UV|X}$ into the protocol. □

3.4.3. The Protocol Using a Concatenated Code

Analogously as in the previous section with the random linear code, we can combine the concatenated code from Theorem 3.7 with the protocol from Lemma 3.12. This has the advantage that Bob's algorithm runs in

polynomial time, but the usage of random variables is slightly worse (and also n needs to be very large).

Theorem 3.15. *Let P_{XYZ} be a probability distribution over $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. Set $R := S_{\rightarrow}(X;Y|Z)$, $d := 2^{22} \cdot \log^2(|\mathcal{X}|\,|\mathcal{Y}|)$, and let parameters n, κ_1, and κ_2 be given. If these parameters satisfy $\kappa_1 < \frac{R^2}{16d}$, $\kappa_2 > \frac{2}{n}$, and $n \geq (\frac{\log^2(|\mathcal{X}|)}{\kappa_1 d})^{\frac{1}{4R}} \sqrt{\frac{d}{\kappa_1}}$, then there exists a one-message key agreement protocol which uses n random variables and yields a key with at least*

$$\lfloor n(R - \sqrt[4]{\kappa_1 d \log^2(|\mathcal{X}|)} - 8\log(|\mathcal{X}|)\sqrt{\kappa_2}) \rfloor$$

key bits, soundness $1 - 2^{-n\kappa_1}$ and secrecy $1 - 2^{-n\kappa_2}$. In the protocol, Alice sends $\mathcal{O}(n^2)$ bits to Bob, and the computations of Alice and Bob can be done in time $\mathcal{O}(n^2)$.

As before, the protocol of Theorem 3.15 is the same for any distribution P_{XYZ} as long as a description of P_{XY} and a description of $\mathsf{P}_{UV|X}$ maximizing $S_{\rightarrow}(X;Y|Z)$ are supplied as input.

Proof. Again we use the protocol from Lemma 3.12, this time combined with the concatenated code from Theorem 3.7.

Consider now the channel $\mathfrak{C}(x)$ mapping x to $(U \oplus x, Y, V)$. We know already that $C := \mathsf{Cap}(\mathfrak{C}) = \log(|\mathcal{X}|) - H(U|YV)$ (see remarks at the beginning of Section 3.3), and thus we see that $\log(|\mathcal{X}|) \geq C \geq H(U|ZV) - H(U|YV) = R$. Let now $\varepsilon := \sqrt[4]{\kappa_1 d C^2}$ and $k := n(C - \varepsilon)$. Alice chooses a code as described in Theorem 3.7 for the Channel $\mathfrak{C}$ mapping x to a triple $(U \oplus x, Y, V)$ with rate $C - \varepsilon$ and sends a description to Bob. The requirements for Theorem 3.7 are satisfied: first, using $\kappa_1 < \frac{R^2}{16d}$ we get $\frac{C}{\varepsilon} = \frac{\sqrt{C}}{\sqrt[4]{\kappa_1 d}} > \frac{2\sqrt{C}}{\sqrt{R}} > 2$. Second, $n \geq (\frac{\log^2(|\mathcal{X}|)}{\kappa_1 d})^{\frac{1}{4R}} \sqrt{\frac{d}{\kappa_1}} > (\frac{C^2}{\kappa_1 d})^{\frac{1}{4R}} \sqrt{\frac{d}{\kappa_1}} = (\frac{C}{\varepsilon})^{\frac{1}{R}} \sqrt{\frac{d}{\kappa_1}} = (\frac{C}{\varepsilon})^{\frac{1}{R}\frac{dC}{\varepsilon^2}} > (\frac{C}{\varepsilon})^{\frac{d}{\varepsilon^2}}$. Subsequently, Alice uses Lemma 3.12 together with this code, which directly implies the rest. □

Again, we can fix a rate $R < S_{\rightarrow}(X;Y|Z)$ to get a corollary which is simple to use.

Corollary 3.16. *Let P_{XYZ} be a probability distribution, $R < S_{\rightarrow}(X;Y|Z)$. There exists a constant c (depending on P_{XYZ} and R) and a one-message key agreement protocol which, on input n, uses n random variables and yields a key with at least nR key bits, soundness $1 - 2^{-cn}$ and secrecy $1 - 2^{-cn}$. Further, the protocol uses $\mathcal{O}(n^2)$ bits of communication, both the computations of Alice and Bob can be done in time $\mathcal{O}(n^2)$.*

Proof. From Theorem 3.15. □

3.5. Lower Bounds

In this section we give lower bounds on the number of random variables used in a one-message key agreement protocol. The lower bounds match the usage in our protocols in the dominating terms.

We first give a theorem (Theorem 3.17) which states that for *any distribution* P_{XYZ} no protocol for one-message key agreement has higher rate than $S_{\rightarrow}(X;Y|Z)$. This was already proven in [AC93] for the case where n goes to infinity; we use the same method to give a quantitative statement which holds for any (finite) n. Further, we show (Theorem 3.18) that there exist distributions P_{XYZ} for which it is impossible to obtain more than

$$n(S_{\rightarrow}(X;Y|Z) - \frac{\sqrt{1-\beta}}{3}\sqrt{\kappa})$$

key bits with secrecy $1 - 2^{-\kappa n}$ and equal soundness from n random variables. In other words, for some distributions the dependence of the number of random variables in terms of κ is optimal up to constant factors in our protocol which uses a random linear code (cf. Theorem 3.13).

Theorem 3.17. *Let a distribution P_{XYZ} and a one-message key agreement protocol for this distribution be given. If the protocol uses n random variables and yields m key bits with soundness $1-\gamma$ and secrecy $1-\varepsilon$, then these parameters satisfy $nS_{\rightarrow}(X;Y|Z) \geq m \cdot (1-\gamma-\varepsilon) - h(\gamma)$.*

Proof. Consider an arbitrary protocol which uses n instances of random variables: First, Alice sends one message M to Bob, and then they both compute their key S_A and S_B, respectively. Any such protocol is sufficiently described by the three conditional distributions $\mathsf{P}_{M|X^n}$, $\mathsf{P}_{S_A|X^nM}$, and $\mathsf{P}_{S_B|Y^nM}$. We can thus consider the joint distribution $\mathsf{P}_{X^nY^nZ^nMS_AS_B}$.

We prove the following chain of inequalities:

$$\begin{aligned} nS_{\rightarrow}(X;Y|Z) &\overset{(1)}{\geq} S_{\rightarrow}(X^n;Y^n|Z^n) \\ &\overset{(2)}{\geq} S_{\rightarrow}(X^nM;Y^nM|Z^nM) \\ &\overset{(3)}{\geq} H(S_A|MZ^n) - H(S_A|MY^n) \\ &\overset{(4)}{\geq} m(1-\gamma-\varepsilon) - h(\gamma). \end{aligned}$$

(It is possible to show that (1) holds with equality, but we do not need this here.)

For (1), we need to show that

$$n\big(\max_{\mathsf{P}_{UV|X}} H(U|ZV) - H(U|YV)\big) \geq \max_{\mathsf{P}_{\underline{U}\,\underline{V}|X^n}} H(\underline{U}|Z^n\underline{V}) - H(\underline{U}|Y^n\underline{V}). \tag{3.15}$$

To see this it is sufficient to construct from a given distribution $\mathsf{P}_{\underline{U}\,\underline{V}|X^n}$ a new distribution $\mathsf{P}_{UV|X}$ with

$$H(U|ZV) - H(U|YV) \geq \frac{1}{n}\Big(H(\underline{U}|Z^n\underline{V}) - H(\underline{U}|Y^n\underline{V})\Big). \tag{3.16}$$

For this, we write

$$\begin{aligned} &H(\underline{U}|Z_1\dots Z_n\underline{V}) - H(\underline{U}|Y_1\dots Y_n\underline{V}) \\ &\qquad = \sum_{i=1}^{n} H(\underline{U}|Z_1\dots Z_iY_{i+1}\dots Y_n\underline{V}) - H(\underline{U}|Z_1\dots Z_{i-1}Y_i\dots Y_n\underline{V}). \end{aligned} \tag{3.17}$$

For every i the expression of the second line can be written equivalently as $H(\underline{U}_i|Z_i\underline{V}_i) - H(\underline{U}_i|Y_i\underline{V}_i)$ where $\underline{U}_i$ and $\underline{V}_i$ are obtained from a channel from X_i (this can be done as follows: for all $j \neq i$ choose (X_j, Y_j, Z_j) according to P_{XYZ}, then choose $(\underline{U}, \underline{V})$ according to $\mathsf{P}_{\underline{U}\,\underline{V}|X_1,\dots,X_n}$, and set $\underline{U}_i := \underline{U}$ and $\underline{V}_i := (Z_1, \dots, Z_{i-1}, Y_{i+1}, \dots, Y_n, \underline{V})$). Let i be the value for which the difference on the right hand side in (3.17) is maximal. For this i we thus find a channel $\mathsf{P}_{\underline{U}_i\,\underline{V}_i|X}$ which satisfies (3.16).

Inequality (2) holds because

$$\begin{aligned} &\max_{\mathsf{P}_{\underline{U}\,\underline{V}|X^n}} H(\underline{U}|Z^n\underline{V}) - H(\underline{U}|Y^n\underline{V}) \\ &\qquad \geq \max_{\mathsf{P}_{\underline{U}\,\underline{V}|MX^n}} H(\underline{U}|MZ^n\underline{V}) - H(\underline{U}|MY^n\underline{V}), \end{aligned}$$

which follows from the fact that in the maximum on the left hand side M can be encoded in $\underline{V}$.

Inequality (3) is equivalent to

$$\max_{\mathsf{P}_{\underline{U}\,\underline{V}|MX^n}} H(\underline{U}|MZ^n\underline{V}) - H(\underline{U}|MY^n\underline{V}) \geq H(S_A|MZ^n) - H(S_A|MY^n),$$

but this holds because the maximum on the left hand side ranges over the values used on the right hand side.

For (4) first note that according to the specification of the protocol it is possible to compute S_B from MY^n such that $\Pr[S_A = S_B] \geq 1 - \gamma$. This implies, using Fano's inequality (see [CT91, Theorem 2.11.1])

$$H(S_A|MY^n) \leq h(\gamma) + m\gamma.$$

Also with probability $1 - \varepsilon$ Eve has no information about S_A, which implies

$$H(S_A|MZ^n) \geq m(1 - \varepsilon).$$

Together this gives (4). □

The above Theorem states that the number of random variables needed must grow basically as $\frac{m}{S_{\rightarrow}(X;Y|Z)}$, which matches the protocol given in Theorem 3.13 in terms of the dependence on m. However, Theorem 3.13 also needs a term depending on κ. It is obvious that this term is not necessary for all distributions: if $X = Y$ is a uniform bit such that Eve already has no information (i.e., X and Y are already perfect key bits), Alice and Bob can obtain a secure key of length m using only $n = m = m/S_{\rightarrow}(X;Y|Z)$ random variables *for any* security parameter κ. We next show that for more general distributions (where information is leaked to Eve) a dependence on κ is inherent. We will consider the distribution where Alice and Bob always get the same uniform bit, and Eve either also gets this bit, or a special symbol $\perp$ signaling that she gets no information. (The distributions used below are a special case of class of distributions which we will study intensively in Chapter 4.)

Theorem 3.18. *Let β with $1 > \beta \geq \frac{1}{2}$ be given. Let P_{XYZ} be the distribution over $\{0,1\} \times \{0,1\} \times \{0,1,\perp\}$ given by*

$$\begin{aligned}\mathsf{P}_{XY}(0,0) &:= \mathsf{P}_{XY}(1,1) := 1/2,\\ \mathsf{P}_{Z|XY}(0|0,0) &:= \mathsf{P}_{Z|XY}(1|1,1) := \beta, \textit{and}\\ \mathsf{P}_{Z|XY}(\perp|0,0) &:= \mathsf{P}_{Z|XY}(\perp|1,1) := 1 - \beta.\end{aligned}$$

Assume there exists a one-message key agreement protocol for this distribution which uses n random variables and yields m key bits with secrecy and soundness $1 - 2^{-n\kappa}$, where $1 - \beta > \kappa > \frac{144}{n(1-\beta)}$. Then, these quantities satisfy

$$m < n(S_{\rightarrow}(X;Y|Z) - \frac{\sqrt{1-\beta}}{3}\sqrt{\kappa}).$$

Proof. We first show that for any one-message protocol with message M, for $\varepsilon := 2^{-n\kappa}$

$$H_\infty^{2\varepsilon}(X^n|Z^nM) \geq m. \tag{3.18}$$

To see this first note that the security of a one-message protocol requires

$$H_\infty^{\varepsilon}(S_A|Z^nM) \geq m,$$

Further there must be a (deterministic) function f mapping X^n and M to a string with $\Pr[f(X^n, M) = S_A] \geq 1 - \varepsilon$, since otherwise $\Pr[S_A = S_B] < 1 - \varepsilon$. Thus, we get $H_\infty^{2\varepsilon}(X^nM|Z^nM) \geq m$, which is equivalent to (3.18).

Next, consider the probability that Eve gets information in k of the n random variables (i.e., Eve obtains the symbol $\perp$ exactly $n - k$ times). This probability is exactly $\binom{n}{k}\beta^k(1-\beta)^{n-k}$. Setting $s := n\sqrt{(1-\beta)\kappa}/3$ Lemma 2.14 states (note that we can apply the lemma since $s < \frac{n(1-\beta)}{3}$) that the probability that Eve gets information in more than $n\beta + s$ random variables is at least

$$\frac{s}{2\sqrt{n}} e^{-\frac{2s^2}{n(1-\beta)\beta}} \geq \frac{\sqrt{(1-\beta)\kappa n}}{6} e^{-\frac{\kappa n}{4\beta}} \geq 2e^{-\frac{\kappa n}{2}} > 2 \cdot 2^{-\kappa n} = 2\varepsilon.$$

Therefore, $n - (n\beta + s)$ must be larger than m, because otherwise (3.18) is violated. Together we get

$$\begin{aligned} m &< n(1-\beta) - s \\ &= n \underbrace{\left(H(X|Z) - H(X|Y)\right)}_{=1-\beta} - s \\ &\leq n(S_{\rightarrow}(X;Y|Z) - \frac{\sqrt{1-\beta}}{3}\sqrt{\kappa}). \end{aligned}$$

□

4. Bounded Distributions

In this chapter we study key agreement for probability distributions P_{XYZ} over $\{0,1\} \times \{0,1\} \times \mathcal{Z}$ if only a lower bound on $\Pr[X{=}Y]$ and upper bounds on the maximal advantage of predicting X or Y from Z are given. There are at least two reasons for such a study: first, such bounds form an information theoretic analog of the computational bounds we have after a weak bit agreement protocol. Second, there is is a connection to the study of statistical zero knowledge (more concretely circuit polarization), and our results will allow insights in this area.

Overview of this chapter

In Section 4.2 we study the one-message key rate for a distribution if only bounds as described above are given. The results will specify what the bounds must satisfy in order to guarantee the existence of a one-message key agreement protocol.

Section 4.3 gives an alternative one-message protocol, which is due to Sahai and Vadhan. If Bob's computation needs to be efficient, this protocol usually needs fewer random variables than the protocols from Chapter 3. An overview of the different protocols will be given in Section 4.4.

In Section 4.5 we study the relationship of two seemingly unrelated tasks. Namely, we show that one-message key agreement as studied in Sections 4.3 and 4.4 is roughly the same as polarizing circuits (we will define this later), which was introduced by Sahai and Vadhan in the context of statistical zero knowledge. This result is not needed to understand the remainder of the thesis.

In Section 4.6 key agreement protocols are studied when arbitrary communication between Alice and Bob is allowed. We will see that this case is less understood than the one-message case. Section 4.7 compares the different protocols.

Related work

As far as I know, key agreement for this particular constraints on the random variables has not been studied before. However, Sahai and Vadhan

[SV97, SV99] have studied the problem of *polarizing circuits* which is related to one-message key agreement for the constraints we use (see Section 4.5).

Dwork, Naor and Reingold [DNR04] study the task of strengthening public-key encryption schemes, a task which is quite directly connected to the problem studied in this chapter. While they note a connection between this and the problem of polarizing circuits, they do not study this connection in detail.

In Section 4.6 key agreement protocols with two-way communication are studied. It was first shown in [Mau93] that distributions exists for which one message is not sufficient to obtain a secure key, but two-way communication suffices. Some of the distributions we study will also have this property. More such cases are studied in [Wol99].

Contributions of this thesis

The results of Sections 4.2 and 4.5 are joint work with Renato Renner [HR05b]; the theorems contained therein are thus original to this thesis. Section 4.3 gives a one-message key agreement protocol, which is based on a method to polarize circuits by Sahai and Vadhan [SV97, SV99]. Thus, while Theorem 4.14 may look novel, it not a new result in a strict sense. Section 4.6 also contains new results; in part these were previously published in [Hol05].

4.1. Definitions and Overview

We first define the correlation α and the leakage β of a distribution P_{XYZ} over $\{0,1\} \times \{0,1\} \times \mathcal{Z}$. In our protocols we will then require a lower bound on the correlation and an upper bound on the leakage.

The correlation is the probability that X equals Y normalized in the interval $[-1,1]$.

Definition 4.1 (Correlation). *The correlation of a probability distribution* P_{XY} *over* $\{0,1\} \times \{0,1\}$ *is* $\alpha := 2\Pr[X = Y] - 1$.

We assume that the correlation satisfies $\alpha \geq 0$ (otherwise, Bob can swap his bit to get positive correlation). Further we note that the correlation only depends on P_{XY}, but not on $\mathsf{P}_{Z|XY}$.

The leakage is the maximal advantage of predicting X from Z.

Definition 4.2 (Leakage). *The leakage of a distribution* P_{XZ} *over* $\{0,1\} \times \mathcal{Z}$ *is* $\beta := \mathrm{Adv}^{\max}(X|Z)$.

The leakage depends only on the joint distribution of X and Z (but not of the distribution of Y). This is what one expects in the one-message case which we study in most of this chapter: in this case the joint distribution of Y and Z does not matter. We will study parameters which are relevant if arbitrary messages are allowed in Sections 4.6 and 4.7.

The *characteristic distribution* for correlation α and leakage β is, intuitively, the worst case distribution for a fixed α and β (i.e., one of the distributions where one-message key agreement should be hardest).

The idea of the characteristic distribution is that random variables distributed according to it can be obtained as follows. First, X is chosen as a uniform random bit. Then, Y is obtained by sending X through a binary symmetric channel with bit flip probability $\frac{1-\alpha}{2}$, and Z is obtained by sending X through a binary erasure channel with erasure probability $1-\beta$.

Definition 4.3 (Characteristic distribution). *For fixed α, β, the characteristic distribution P_{XYZ} over $\{0,1\} \times \{0,1\} \times \{0,1,\perp\}$ is defined as*

$$\mathsf{P}_{XYZ}(x,y,z) := \mathsf{P}_X(x) \cdot \mathsf{P}_{Y|X}(y|x) \cdot \mathsf{P}_{Z|X}(x|z),$$

where

$$\mathsf{P}_X(0) := \mathsf{P}_X(1) := \frac{1}{2},$$

$$\mathsf{P}_{Y|X}(y,x) := \begin{cases} \frac{1+\alpha}{2} & \text{if } x = y, \\ \frac{1-\alpha}{2} & \text{otherwise,} \end{cases}$$

$$\mathsf{P}_{Z|X}(z,x) := \begin{cases} \beta & \text{if } z = x, \\ 1-\beta & \text{if } z = \perp, \\ 0 & \text{otherwise.} \end{cases}$$

4.2. The One-Message Key Rate

In this section we show two statements (given in Theorem 4.9). First, the one-message key rate of any distribution P_{XYZ} over $\{0,1\} \times \{0,1\} \times \mathcal{Z}$ with correlation α and leakage $\beta < \alpha^2$ is at least $\frac{(\alpha^2-\beta)^2}{7}$. Second, if $\alpha^2 \leq \beta$, then there exists a distribution with correlation α, leakage β, and one-message key rate zero.

For this, Lemma 4.4 shows that both statements above hold for the characteristic distribution. Lemma 4.8 then shows that the characteristic dis-

tribution is the distribution which has the lowest one-message key rate among all distributions with correlation α and leakage β.

Together with the results of Chapter 3 this gives protocols for one-message key agreement for any distribution with correlation α and leakage β, as long as $\alpha^2 > \beta$.

Lemma 4.4. *Let P_{XYZ} be the characteristic distribution with correlation α and leakage β. Then,*

$$S_{\rightarrow}(X;Y|Z) = \max_{\lambda} H(X_\lambda|Z) - H(X_\lambda|Y), \tag{4.1}$$

where X_λ is the random variable over $\{0,1\}$ obtained by sending X through a binary symmetric channel with bit flip probability $\frac{1-\lambda}{2}$, i.e.,

$$\mathsf{P}_{X_\lambda|X}(\overline{x}|x) := \begin{cases} \frac{1+\lambda}{2} & \text{if } \overline{x} = x, \\ \frac{1-\lambda}{2} & \text{otherwise.} \end{cases}$$

Moreover, $\alpha^2 > \beta$ implies $S_{\rightarrow}(X;Y|Z) \geq \frac{1}{7}(\alpha^2-\beta)^2$ while $\alpha^2 \leq \beta$ implies $S_{\rightarrow}(X;Y|Z) = 0$.

Since the term in the maximum of (4.1) only involves random variables whose distribution is explicitly known (cf. Definition 4.3) we get the following form of it (where $h(x) := -x\log(x) - (1-x)\log(1-x)$ is the binary entropy function):

$$\begin{aligned} g_{\alpha,\beta}(\lambda) &:= H(X_\lambda|Z) - H(X_\lambda|Y) \\ &= (1-\beta) + \beta h\Big(\frac{1+\lambda}{2}\Big) - h\Big(\frac{1+\alpha\lambda}{2}\Big). \end{aligned} \tag{4.2}$$

See Figure 4.1 for a plot of the function $g_{\alpha,\beta}$ for specific parameters of α and β.[1]

We first give a few properties of the function $g_{\alpha,\beta}$. For this we need the following estimate on the binary entropy function.

Lemma 4.5. *For any $-1 \leq x \leq 1$:*

$$1 - \frac{x^2}{2\ln(2)} - \Big(1 - \frac{1}{2\ln(2)}\Big)x^4 \leq h\Big(\frac{1+x}{2}\Big) \leq 1 - \frac{x^2}{2\ln(2)}.$$

[1] We can see in Figure 4.1 that these random variables are a case where "forgetting helps": $H(X|Z) - H(X|Y)$ is negative in Figure 4.1, while $H(X_\lambda|Z) - H(X_\lambda|Y)$ is positive for some $\lambda \in [-1,1]$.

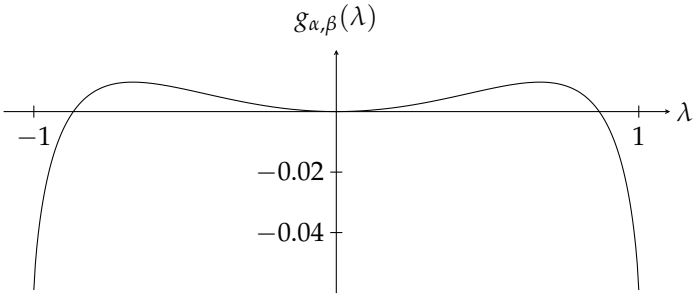


Figure 4.1.: Plot of $g_{\alpha,\beta}(\lambda)$ with $\alpha = 0.8$ and $\beta = 0.59$.

Proof. Using standard tools of calculus we obtain for $-1 \leq x \leq 1$:

$$h\Big(\frac{1+x}{2}\Big) = 1 - \frac{1}{2\ln(2)} \sum_{n=1}^{\infty} \frac{x^{2n}}{2n^2 - n}.$$

The lower bound can be obtained by using $x^{2n} \leq x^4$ for $n \geq 2$. The upper bound is obtained by stopping the summation after the first term. □

We now prove several properties of $g_{\alpha,\beta}$.

Lemma 4.6. *Let the function $g_{\alpha,\beta} : [-1,1] \to \mathbb{R}$ be as in (4.2). If $\alpha^2 \leq \beta$, then $g_{\alpha,\beta}(\lambda) \leq 0$ for all $\lambda \in [-1,1]$ and $g_{\alpha,\beta}$ is concave. If $\alpha^2 > \beta$, then $g_{\alpha,\beta}$ has one local minimum at $\lambda = 0$ with $g_{\alpha,\beta}(0) = 0$ and two local maxima at $-\lambda^+$ and λ^+, $\lambda^+ \in (0,1]$ with $g_{\alpha,\beta}(-\lambda^+) = g_{\alpha,\beta}(\lambda^+) \geq \frac{1}{7}(\alpha^2 - \beta)^2$. Furthermore, $g_{\alpha,\beta}$ is concave in $[-1,-\lambda^+]$ and $[\lambda^+,1]$.*

Proof. First, note that $g_{\alpha,\beta}$ is symmetric with respect to the vertical axis. Further, $g_{\alpha,\beta}(0) = 0$ and $g'_{\alpha,\beta}(0) = 0$. The second derivation $g''_{\alpha,\beta}$ of $g_{\alpha,\beta}$ has the simple form

$$g''_{\alpha,\beta}(\lambda) = \frac{1}{\ln(2)}\Big(\frac{\alpha^2}{1-\alpha^2\lambda^2} - \frac{\beta}{1-\lambda^2}\Big),$$

which is never positive if $\alpha^2 \leq \beta$, implying the statements for this case. If $\alpha^2 > \beta$, this implies $g''_{\alpha,\beta}(0) > 0$, and thus $g_{\alpha,\beta}$ has a local minimum for $\lambda = 0$.

Furthermore we see that $g''_{\alpha,\beta}(\lambda) = 0$ if and only if

$$\alpha^2 - \beta = \alpha^2\lambda^2(1-\beta)$$

Since this is quadratic in λ, the second derivation $g''_{\alpha,\beta}$ has at most two zeros. Hence, the function $g_{\alpha,\beta}$ has at most two inflection points and thus at most two local maxima. Since the function has a local minimum at 0, and $[-1,1]$ is closed, it must have at least 2 local maxima (possibly at the endpoints -1 and 1), and we denote these by $-\lambda^+$ and λ^+. Also, the function must be concave outside of the inflection points.

To see that $g_{\alpha,\beta}(\lambda^+) \geq \frac{(\alpha^2-\beta)^2}{7}$, we first use Lemma 4.5 to obtain

$$g_{\alpha,\beta}(\lambda) \geq \frac{\lambda^2}{2\ln(2)}(\alpha^2 - \beta - \lambda^2(2\beta\ln(2) - \beta))$$

and assume first that $0 \leq \alpha^2 - \beta \leq 4\beta\ln(2) - 2\beta$. Then, setting $\lambda^2 := \frac{\alpha^2-\beta}{4\beta\ln(2)-2\beta}$ yields

$$\max_\lambda g_{\alpha,\beta}(\lambda) \geq \frac{(\alpha^2-\beta)^2}{8\beta\ln(2)(2\ln(2)-1)} > \frac{(\alpha^2-\beta)^2}{3}.$$

If, on the other hand, $\alpha^2 - \beta > 4\beta\ln(2) - 2\beta$ (i.e., $\beta < \alpha^2/(4\ln(2)-1)$), then setting $\lambda := 1$ gives

$$\max_\lambda g_{\alpha,\beta}(\lambda) \geq \frac{\alpha^2}{2\ln(2)} - \beta > \frac{\alpha^2}{7} \geq \frac{(\alpha^2-\beta)^2}{7},$$

since $\alpha \leq 1$. □

We can now show that V is not needed, i.e., we prove $S_\rightarrow(X;Y|Z) = \max_\lambda g_{\alpha,\beta}(\lambda)$. Per definition, $S_\rightarrow(X;Y|Z) \geq \max_\lambda g_{\alpha,\beta}(\lambda)$, so we only need to show that $\max_\lambda g_{\alpha,\beta}(\lambda)$ is also a lower bound.

Lemma 4.7. *Let P_{XYZ} be the characteristic distribution with correlation α and leakage β. Then, $S_\rightarrow(X;Y|Z) \leq \max_\lambda g_{\alpha,\beta}(\lambda)$, where $g_{\alpha,\beta}$ is defined by* (4.2)*.*

Proof. Let $\mathsf{P}_{UV|X}$ be a fixed channels. We will show that $H(U|ZV) - H(U|YV) \leq \max_\lambda g_{\alpha,\beta}(\lambda)$.

As in (3.1) (see page 30) we rewrite $H(U|ZV) - H(U|YV)$ as

$$H(U|ZV) - H(U|YV) = H(Z|UV) - H(Y|UV) - (H(Z|V) - H(Y|V)). \tag{4.3}$$

Consider now a fixed pair (u, v). Setting $\frac{1+\lambda_{u,v}}{2} := \Pr[X{=}0|U{=}u, V{=}v]$ and $\frac{1+\lambda_v}{2} := \Pr[X{=}0|V{=}v]$, a straightforward computation yields:

$$\begin{aligned} H(Z|U{=}u, V{=}v) - H(Y|U{=}u, V{=}v) \\ = h(\beta) + \beta h\Big(\frac{1+\lambda_{u,v}}{2}\Big) - h\Big(\frac{1+\alpha\lambda_{u,v}}{2}\Big) \\ H(Z|V{=}v) - H(Y|V{=}v) = h(\beta) + \beta h\Big(\frac{1+\lambda_v}{2}\Big) - h\Big(\frac{1+\alpha\lambda_v}{2}\Big). \end{aligned}$$

Because $g_{\alpha,\beta}$ differs from these expressions only by a constant, together with (4.3) this gives

$$H(U|ZV) - H(U|YV) = \mathop{\mathrm{E}}_{\mathsf{P}_{UV}}[g_{\alpha,\beta}(\lambda_{U,V})] - \mathop{\mathrm{E}}_{\mathsf{P}_V}[g_{\alpha,\beta}(\lambda_V)].$$

Using $\mathop{\mathrm{E}}_{\mathsf{P}_{U|V=v}}[\lambda_{U,v}] = \lambda_v$ we thus obtain

$$H(U|ZV) - H(U|YV) = \mathop{\mathrm{E}}_{\mathsf{P}_V}\Big[\mathop{\mathrm{E}}_{\mathsf{P}_{U|V=v}}[g_{\alpha,\beta}(\lambda_{U,V})] - g_{\alpha,\beta}(\mathop{\mathrm{E}}_{\mathsf{P}_{U|V=v}}[\lambda_{U,V}])\Big]. \tag{4.4}$$

For every fixed v, we can use Lemma 4.6 to obtain the following upper bound on the term in the expectation:

$$\begin{aligned} \mathop{\mathrm{E}}_{\mathsf{P}_{U|V=v}}[g_{\alpha,\beta}(\lambda_{U,v})] - g_{\alpha,\beta}(\mathop{\mathrm{E}}_{\mathsf{P}_{U|V=v}}[\lambda_{U,V}]) &\le \max_\lambda g_{\alpha,\beta}(\lambda) - g_{\alpha,\beta}(0) \\ &= \max_\lambda g_{\alpha,\beta}(\lambda), \end{aligned}$$

which can now be inserted in (4.4). □

These Lemmas immediately imply Lemma 4.4:

Proof (of Lemma 4.4). From Lemmas 4.6 and 4.7. □

The following lemma shows that if we have any distribution P_{XYZ} with correlation α and leakage β, it is safe to assume it is the characteristic distribution.

Lemma 4.8. *Let α, β and λ be given, and let P_{XYZ} be any distribution with correlation at least α and leakage at most β. Let further $\mathsf{P}_{\overline{X}\,\overline{Y}\,\overline{Z}}$ be the characteristic distribution with correlation α and leakage β.*

Let X_λ be the random variable obtained from X by sending it through an binary symmetric channel with bit flip probability $\frac{1-\lambda}{2}$. Analogously, let $\overline{X}_\lambda$ be

obtained from $\overline{X}$ by sending it through an binary symmetric channel with bit flip probability $\frac{1-\lambda}{2}$. Then,

$$H(X_\lambda|Z) - H(X_\lambda|Y) \geq H(\overline{X}_\lambda|\overline{Z}) - H(\overline{X}_\lambda|\overline{Y}).$$

Proof. For the distribution P_{XZ} let $\mathsf{P}_{B|XZ}$ be the distribution as guaranteed by Lemma 2.2. We get $H(X_\lambda|Z) \geq H(X_\lambda|ZB) \geq (1-\beta) + \beta h(\frac{1+\lambda}{2}) = H(\overline{X}_\lambda|\overline{Z})$.

Thus, it is sufficient to show that $H(X_\lambda|Y) \leq H(\overline{X}_\lambda|\overline{Y})$. For this, let U be a uniform random bit, which is independent of X and Y. We get $H(X_\lambda|Y) = H((X_\lambda \oplus U)|Y \oplus U, U) \leq H((X_\lambda \oplus U)|Y \oplus U) \leq H(\overline{X}_\lambda|\overline{Y})$ where the last inequality can be seen by noting that instead of computing $(X_\lambda \oplus U)$ one can equivalently send $X \oplus U$ through a binary symmetric channel (in which case we have exactly the same random experiment as for the characteristic distribution). □

Theorem 4.9. *Let $\alpha \in [0,1]$ and $\beta \in [0,1]$ be constants. If $\alpha^2 > \beta$ then any distribution P_{XYZ} with correlation α and leakage β has $S_{\rightarrow}(X;Y|Z) \geq \frac{(\alpha^2-\beta)^2}{7}$. If $\alpha^2 \leq \beta$ then there exists a distribution P_{XYZ} with correlation α, leakage β, and $S_{\rightarrow}(X;Y|Z) = 0$.*

Proof. From Lemma 4.4 and Lemma 4.8. □

This implies that we can combine Theorem 4.9 with the protocols from Section 3.4 and obtain protocols which can be used without knowing more of a given distribution than the correlation and leakage it has.[2]

4.3. The Sahai-Vadhan Protocol

In [SV97], Sahai and Vadhan give a method to polarize circuits, a task which occurs in the study of statistical zero knowledge. In Section 4.5 we will show that circuit polarization is essentially the same as one-message key agreement for α-correlated variables with leakage β. This implies that Sahai and Vadhan implicitly present such a one-message key agreement protocol in their paper. This protocol usually uses fewer random variables than the protocol which uses a concatenated code given in Theorem 3.15, and thus we present it here (the protocol from Theorem 3.13 is

[2] Strictly speaking, one must also check that the protocols given in Section 3.4 work in case the correlation is bigger than expected. However, it is easy to see that one can modify the error correcting codes used slightly in order to make sure that this is not a problem.

our most efficient protocol in terms of random variables used, but lacks polynomial time decoding).

In the protocol, Alice and Bob start by taking the XOR of s random variables they have (this requires no communication). Then, r instances of these new random variables are used in a repeat code to correct errors: Alice chooses a bit X and sends $(X_0 \oplus X, \ldots, X_{r-1} \oplus X)$ to Bob, who uses his information to guess X. If $\alpha^2 > \beta$ and (r, s) is appropriately chosen, this two stage process can be used to transform α-correlated random variables with leakage β to $\frac{2}{3}$-correlated random variables with leakage $\frac{1}{3}$. To get a key from such random variables we then use the standard protocol using a concatenated code as given in Theorem 3.15 (Sahai and Vadhan use a different, slightly less efficient approach).

First, we need a technical lemma:

Lemma 4.10. *Let X_1 and X_2 be independent random variables over $\{0,1\}$ with $\Pr[X_1 = 0] = \frac{1+\alpha_1}{2}$ and $\Pr[X_2 = 0] = \frac{1+\alpha_2}{2}$. Then,*

$$\Pr[X_1 \oplus X_2 = 0] = \frac{1 + \alpha_1\alpha_2}{2}.$$

Proof. $\Pr[X_1 \oplus X_2 = 0] = \dfrac{1+\alpha_1}{2} \cdot \dfrac{1+\alpha_2}{2} + \dfrac{1-\alpha_1}{2} \cdot \dfrac{1-\alpha_2}{2} = \dfrac{1+\alpha_1\alpha_2}{2}.$ □

Using this, we can analyze the first step in the protocol (i.e., taking the XOR of independent random variables).

Lemma 4.11. *Let $(X_0, Y_0, Z_0), \ldots, (X_{s-1}, Y_{s-1}, Z_{s-1})$ be independent α-correlated random variables with leakage β. Then $(X_0 \oplus \cdots \oplus X_{s-1}, Y_0 \oplus \cdots \oplus Y_{s-1}, Z_0 \ldots Z_{s-1})$ is α^s-correlated and has leakage β^s.*

Proof. Since $(X_0 \oplus Y_0) \oplus \cdots \oplus (X_{s-1} \oplus Y_{s-1}) = 0$ exactly if the resulting random random variables of Alice and Bob are equal, the fact that the resulting random variables are α^s-correlated follows directly from Lemma 4.10 using induction. Further, Lemma 2.3 states

$$\mathrm{Adv}^{\max}(X_0 \oplus \cdots \oplus X_{s-1} | Z_0, \ldots, Z_{s-1}) = \big(\mathrm{Adv}^{\max}(X|Z)\big)^s. \qquad \square$$

We now analyze the used repeat code:

Lemma 4.12. *Let $(X_0, Y_0, Z_0), \ldots, (X_{r-1}, Y_{r-1}, Z_{r-1})$ be independent α-correlated random variables with leakage β. Then, Alice can send one message M of length r to Bob such that Alice and Bob get output X and Y, respectively*

and $(X, Y, Z_0 \dots Z_{r-1}M)$ *is* $1 - 2e^{-r\frac{\alpha^2}{4}}$*-correlated and has leakage at most* $r\beta$*. Furthermore, both Alice's and Bob's algorithms run in time* $\mathcal{O}(r)$*.*

Proof. Alice chooses a uniform random bit $X \in \{0,1\}$ and sends the values $X_1 \oplus X, \dots, X_r \oplus X$ to Bob, who sets Y to the majority of what $X \oplus X_i \oplus Y_i$ yields. Using the Hoeffding bound (Proposition 2.13) we see that $\Pr[X = Y] \geq 1 - \exp(-r\frac{\alpha^2}{4})$.

To see that $\mathrm{Adv}^{\max}(X|Z_0 \dots Z_{s-1}M) \leq r\beta$ we use Lemma 2.2 and associate a random variable B_i with each pair (X_i, Z_i) such that $\Pr[B_i = 0] = 1 - \beta$ and $\Pr[f(Z_i) = X_i | B_i = 0] = \frac{1}{2}$. If B is the random variable which is zero if $B_i = 0$ for all i, then, for all functions f

$$\begin{aligned}\Pr[f(Z_0,\dots,&Z_{r-1}, M) = X] \\ &= \Pr[B = 0]\Pr[f(Z_0, \dots, Z_{r-1}, M) = X | B = 0] \\ &\quad + \Pr[B = 1]\Pr[f(Z_0, \dots, Z_{r-1}, M) = X | B = 1] \\ &\leq \Pr[B = 0]\frac{1}{2} + \Pr[B = 1].\end{aligned}$$

The union bound implies $\Pr[B = 0] \geq 1 - r\beta$, and thus we get

$$\Pr[f(Z_0, \dots, Z_{r-1}, M) = X] \leq \frac{1 + r\beta}{2}. \qquad \square$$

The following lemma combines these protocols to get $\frac{2}{3}$-correlated random variables with leakage $\frac{1}{3}$ from α-correlated random variables with leakage β, satisfying $\alpha^2 > \beta$.

Lemma 4.13. *Let* P_{XYZ} *be a probability distribution with correlation at least* α *and leakage at most* $\beta < \alpha^2$*. For* $\gamma := \max(1, \frac{1}{\log(\alpha^2/\beta)})$ *there exists a one-message key agreement protocol which uses* $n = \lfloor \frac{128\gamma}{\alpha^{(12\gamma)}} \rfloor$ *random variables and outputs one random variable with correlation at least* $\frac{2}{3}$ *and leakage at most* $\frac{1}{3}$*. The protocol needs* α *and* β *as input. Further, the computations of both Alice and Bob can be done in time* $\mathcal{O}(n)$*.*

Proof. We assume $\alpha^2 \leq 2\beta$ (otherwise, we increase β to $\alpha^2/2$; this will not make the protocol insecure and does not change γ). We first apply Lemma 4.11 for $s := \lceil 5/\log(\alpha^2/\beta) \rceil$. Then, we use the resulting random variables in the protocol of Lemma 4.12 with $r := \lceil \frac{1}{4\beta^s} \rceil$. We first note

(using $2\beta \geq \alpha^2$) that

$$s < \frac{5}{\log(\frac{\alpha^2}{\beta})} + 1 = \frac{5 + \log(\frac{\alpha^2}{\beta})}{\log(\frac{\alpha^2}{\beta})} \leq \frac{6}{\log(\frac{\alpha^2}{\beta})}.$$

Further (using $s > 5/\log(\alpha^2/\beta) > 5/\log(1/\beta) = \log_\beta(1/32)$) we get

$$r < \frac{1}{4\beta^s} + 1 = \frac{1 + 4\beta^s}{4\beta^s} < \frac{1 + \frac{4}{32}}{4\beta^s} < \frac{1}{3\beta^s}.$$

Using Lemmas 4.11 and 4.12, we see that this gives us random variables with correlation

$$\begin{aligned}
1 - 2\exp\Big(-r\frac{\alpha^{2s}}{4}\Big) &\geq 1 - 2\exp\Big(-\frac{\alpha^{2s}}{16\beta^s}\Big) \\
&= 1 - 2\exp\Big(-\frac{1}{16}\Big(\frac{\alpha^2}{\beta}\Big)^s\Big) \\
&\geq 1 - 2\exp\Big(-\frac{1}{16}2^{\log(\alpha^2/\beta)\frac{5}{\log(\alpha^2/\beta)}}\Big) \\
&= 1 - 2\exp(2^{-32/16}) > \frac{2}{3}.
\end{aligned}$$

Using $r < \frac{1}{3\beta^s}$, we get that the leakage is at most $r\beta^s < \frac{1}{3}$. Finally, the number of random variables used is $s \cdot r$ which is at most

$$\frac{6}{\log(\frac{\alpha^2}{\beta})} \cdot \frac{1}{3\beta^{\frac{6}{\log(\frac{\alpha^2}{\beta})}}} = \frac{2}{\log(\frac{\alpha^2}{\beta})\beta^{\frac{6}{\log(\frac{\alpha^2}{\beta})}}} = \frac{2\gamma}{\beta^{6\gamma}}.$$

Since $2^{1/\gamma} = \frac{\alpha^2}{\beta}$ (and thus $\beta = \frac{\alpha^2}{2^{1/\gamma}}$) we get $\beta^{6\gamma} = \alpha^{12\gamma}/64$, and thus

$$\frac{2\gamma}{\beta^{6\gamma}} = \frac{128\gamma}{\alpha^{12\gamma}}.$$

Finally, the bound on the runtime of Alice and Bob follows directly from Lemmas 4.11 and 4.12. □

The random variables produced by Lemma 4.13 are now used in a standard key agreement protocol which uses a concatenated code for information reconciliation (Theorem 3.15). Together this gives a complete one-message key agreement protocol for α-correlated random variables with leakage β.

In the following theorem, we assume that we want m bits with soundness and secrecy $1 - 2^{-m}$ (i.e., we do not use a separate security parameter).[3]

Theorem 4.14. *Let P_{XYZ} be a probability distribution with correlation at least α and leakage at most $\beta < \alpha^2$. There exists an absolute constant $d > 0$ and a one-message key agreement protocol which uses n random variables and yields a key with at least*

$$m = \left\lfloor \frac{dn\alpha^{12\gamma}}{\gamma} \right\rfloor$$

bits, secrecy and soundness $1 - 2^{-m}$, where $\gamma := \max(1, \frac{1}{\log(\alpha^2/\beta)})$. For this, the protocol needs inputs α, β, and n. Further, Alice's and Bob's algorithms run in time $\mathcal{O}(n^2)$.

Proof. We first use the protocol from Lemma 4.13 to get $\frac{2}{3}$-correlated random variables with leakage $\frac{1}{3}$. These random variables are then used in the protocol of Theorem 3.15. Let $n_0 := \lfloor \frac{n\alpha^{12\gamma}}{128\gamma} \rfloor$ be the number of random variables used in Theorem 3.15. Clearly, the one-message key rate of $\frac{2}{3}$-correlated random variables with leakage $\frac{1}{3}$ is a positive constant, and we choose constants $\kappa_1 = \kappa_2 = \kappa$ small enough that Theorem 3.15 yields secret bits at a constant rate (say at rate d_0), for which only $n_0 > d_1$ for some large constant d_1 is required. As the key produced has both secrecy and soundness $1 - 2^{-\kappa n_0}$, we can now possibly use only a constant fraction of the key output, in order to make sure that the key of length m has secrecy and soundness $1 - 2^{-m}$. Thus, for an appropriate constant d all the statements hold. □

4.4. Summary of One-Message Protocols

If random variables distributed according to a distribution P_{XYZ} with correlation at least α and leakage at most $\beta < \alpha^2$ are given, our results up to now give several one-message key agreement protocols. Namely, we can

[3] This is in contrast to Theorems 3.13 and 3.15, where we studied how the number of random variable grows depending on the number of key bits as well as the security separately. The main reason for this is that in Theorems 3.13 and 3.15 it is possible to achieve a rate arbitrarily close to the one-message key rate. In Theorem 4.14 the use of Lemma 4.13 prevents us from achieving the rate in the first place. We thus give the simpler version of Theorem 4.14 which costs only constant factors.

Protocol	Rate achieved	Remarks
Theorem 3.13	$S_{\rightarrow}(X;Y\|Z)-\varepsilon$	Exponential computation for Bob
Theorem 3.15	$S_{\rightarrow}(X;Y\|Z)-\varepsilon$	$n > (\frac{1}{\alpha^2-\beta})^{\mathrm{poly}(1/(\alpha^2-\beta))}$
Theorem 4.14	$\Theta(\frac{\alpha^{12\gamma}}{\gamma})$	$\gamma = \max(1, \frac{1}{\log(\alpha^2/\beta)})$

Table 4.1.: One-message key agreement protocols for distributions with correlation α and leakage β; ε can be any small constant larger than 0.

use either Theorem 3.13 (our basic protocol where we use a random linear code to do information reconciliation), Theorem 3.15 (the protocol with a concatenated code for information reconciliation) or Theorem 4.14. We will now compare these approaches, an overview is given in Table 4.1.

The protocol of Theorem 3.13 has the advantage that it requires the least number of random variables for given security parameters and given number of key bits required. On the other hand, it does not allow efficient decoding.

Comparing the protocols of Theorem 3.15 and Theorem 4.14, we see that the former has the advantage that it can give key bits at a rate arbitrary close to the one-message key rate. The disadvantage of this protocol is that n needs to be very large, namely

$$n > \Big(\frac{1}{\alpha^2-\beta}\Big)^{\mathrm{poly}(\frac{1}{\alpha^2-\beta})}, \tag{4.5}$$

(we used the fact that κ_1 has to be smaller than the one-message key rate). We compare this with the number of random variables needed for a single key bit when using Theorem 4.14 which is

$$\Theta\Big(\frac{\gamma}{\alpha^{12\gamma}}\Big). \tag{4.6}$$

We can show that (4.5) usually grows much faster than (4.6): in case $\alpha^2 \geq 2\beta$, (4.5) has asymptotic behavior $(\frac{1}{\alpha})^{\mathrm{poly}(\frac{1}{\alpha})}$, while (4.6) grows approximately as $\Theta(\frac{1}{\alpha^{12}})$, i.e., Theorem 4.14 uses fewer random variables in this case. If $\alpha^2 < 2\beta$ we can use $1-x \leq \ln(\frac{1}{x})$ (which holds for $0 < x \leq 1$) and get $\alpha^2-\beta = \alpha^2(1-\frac{\beta}{\alpha^2}) \leq \alpha^2 \ln(\frac{\alpha^2}{\beta}) \leq \ln(\frac{\alpha^2}{\beta}) < \frac{2}{\gamma}$ (the last inequality holds per definition of γ because $\alpha^2 < 2\beta$), and thus $\frac{1}{\alpha^2-\beta} > \frac{\gamma}{2}$. Therefore,

(4.5) has a larger exponent and will still grow asymptotically faster than (4.6). On the other hand, if the number of key bits required is exceedingly large, the protocol from Theorem 3.15 will use fewer random variables (since it achieves a higher rate).

4.5. One-Message Protocols and Circuit Polarization

Let two circuits be given, which on uniform random input yield output distributions P_{W_0} and P_{W_1} over the same set. For two given parameters α and β with $\alpha > \beta$, the distributions are guaranteed to satisfy either

$$\|\mathsf{P}_{W_0} - \mathsf{P}_{W_1}\| \geq \alpha, \tag{4.7}$$

or

$$\|\mathsf{P}_{W_0} - \mathsf{P}_{W_1}\| \leq \beta. \tag{4.8}$$

In this section we look for an efficient method to *polarize* the circuits: if (4.7) holds, the method should output two new circuits which produce near disjoint output distributions. If (4.8) holds, then the method should output two circuits which produce near identical distributions.[4]

This problem arises in the study of statistical zero knowledge (for honest verifiers). Assume that Vic (the verifier) has given two circuits for which the corresponding distributions satisfies (4.7) or (4.8). Peggy, a powerful prover who has also given the circuits, would like to convince Vic that (4.7) holds. The following protocol achieves this (but is not zero knowledge): Vic chooses a random sample of either P_{W_0} or P_{W_1} and sends it to Peggy. Peggy replies with a guess which distribution was chosen. By repeating this, Vic can check that Peggy succeeds with probability $\frac{1+\alpha}{2}$, which is only possible if $\|\mathsf{P}_{W_0} - \mathsf{P}_{W_1}\| \geq \alpha$. The protocol is *not* zero knowledge, even if Vic does not deviate from the protocol, because Vic learns on which instances Peggy errs. If an efficient polarization method can be applied before the protocol, Vic will always be able to anticipate the answer of Peggy which implies that the protocol is zero knowledge (in case Vic does not deviate from the protocol).

[4] Deciding which of (4.7) or (4.8) holds is believed to be a computationally hard problem (an example where it seems difficult for $\alpha = 1$ and $\beta = 0$ can be constructed based on the conjectured difficulty of deciding whether two given graphs are isomorphic). This means that we cannot polarize by first measuring the statistical distance.

4.5.1. Polarization and Oblivious Polarization

In general, a method to polarize two circuits can use the description of the circuits given. Here, we focus on methods with several restrictions. First, they use the given circuits with uniform and independent random input in a black-box manner only. Second, they output all the samples obtained, but do not do any other computation with them.

The following process can be used to describe such a method completely: on input $b \in \{0,1\}$ (this denotes which distribution should be produced) and k (a "security" parameter), algorithm A outputs a list of query bits $(Q_b^0, Q_b^1, \ldots, Q_b^{n-1})$ and some side information R_b. The output of the algorithm is then the concatenation of samples $W_{Q_b^0}$ to $W_{Q_b^{n-1}}$ and the string R_b.

Definition 4.15 (Oblivious polarization). *Let an algorithm be given which, for parameters α and β, and inputs k and b outputs bits $Q_b^0, \ldots, Q_b^{n-1}$ and a string R_b. For two distributions P_{W_0} and P_{W_1}, let $\mathsf{P}_{\overline{W}_b}$ be the distribution obtained by concatenating independent samples $W_{Q_b^0}$ to $W_{Q_b^{n-1}}$ and R_b. The algorithm is an* oblivious polarization method *if it satisfies*

$$\|\mathsf{P}_{W_0} - \mathsf{P}_{W_1}\| \geq \alpha \Longrightarrow \|\mathsf{P}_{\overline{W}_0} - \mathsf{P}_{\overline{W}_1}\| \geq 1 - 2^{-k}$$
$$\|\mathsf{P}_{W_0} - \mathsf{P}_{W_1}\| \leq \beta \Longrightarrow \|\mathsf{P}_{\overline{W}_0} - \mathsf{P}_{\overline{W}_1}\| \leq 2^{-k}.$$

The method is efficient if the algorithm runs in time polynomial in k.

The method given in [SV99] to polarize circuits is oblivious in this sense.[5] We note here that in [SV99] a second, non-oblivious method is given which *inverts* the statistical distance of given (but already polarized) distributions.

4.5.2. Equivalence of Polarization and Key Agreement

An oblivious polarization method for parameters α and β is basically equivalent to a one-message key agreement protocol for probability distributions with correlation α and leakage β.

Theorem 4.16. *There exists an oblivious polarization method for parameters α and β if and only if there exists a one message key agreement protocol secure on distributions with correlation α and leakage β. Moreover, there exists an* efficient *oblivious polarization method if and only if there exists a protocol where Alice's algorithm runs in polynomial time.*

[5] In fact, R_b is the empty string in their method.

We prove Theorem 4.16 in both directions separately. We first show that a polarization method implies the existence of a one-message key agreement protocol:

Lemma 4.17. *Let an oblivious polarization method for parameters α and β be given. Then there exists a one-message key agreement protocol which is secure for any distribution P_{XYZ} with correlation at least α and leakage at most β. Furthermore, if the polarization method is efficient, then Alice's algorithm runs in polynomial time.*

Proof. We give a one-message key agreement protocol which yields a single key bit. This is clearly sufficient.

The protocol, which uses as many random variables as the number of query bits produced by the polarization method, works as follows: Alice first simulates the polarization method with input k and a uniform random bit B. This yields R_B and $Q_B^0, \ldots, Q_B^{n-1}$. Then, Alice sends R_B as well as $(X_0 \oplus Q_B^0, \ldots, X_{n-1} \oplus Q_B^{n-1})$ as communication to Bob, and outputs B as secret bit.

Bob can find B with high probability from the communication and Y^n (it may not be possible to implement this efficiently): Since P_{XYZ} has correlation at least α the random variables $\mathsf{P}_{W_0} := (X, Y)$ and $\mathsf{P}_{W_1} := (1 \oplus X, Y)$ satisfy $\|\mathsf{P}_{W_0} - \mathsf{P}_{W_1}\| \geq \alpha$. Furthermore, $Y_1, \ldots, Y_n$ and the communication gives Bob a sample of the distribution produced by the polarization method with input P_{W_0} and P_{W_1}. The polarization property of the method now implies that a statistical test can find B while making only an exponentially small error in k.

Also the protocol is secure against Eve: consider the random variables $W_0' := (Z, X)$ and $W_1' := (Z, X \oplus 1)$. Since P_{XYZ} has leakage at most β we see that $\|P_{W_0'} - \mathsf{P}_{W_1'}\| \leq \beta$. Further, Eve gets exactly a sample of of the distribution produced by the polarization method with input $\mathsf{P}_{W_0'}$ and $\mathsf{P}_{W_1'}$, which means that it is independent of B except with probability exponentially small in k. □

On the other hand, a one-message key agreement protocol yields a polarization method:

Lemma 4.18. *Let a one-message key agreement protocol secure for any distribution P_{XYZ} with correlation at least α and leakage at most β be given. Then, there exists an oblivious polarization method for parameters α and β. Furthermore, if Alice's algorithm runs in polynomial time, then the polarization method is efficient.*

Proof. Throughout the proof we only need key agreement for one key bit. The polarization method works as follows: on input b and k, the polarization method first chooses random (uniform and independent) query bits $Q_b^0, \ldots, Q_b^{n-1}$. Then Alice is simulated with given random variables $X_0 := Q_b^0, \ldots, X_{n-1} := Q_b^{n-1}$, which yields communication Γ, and a secret bit S. The string R_b is then defined as $R_b := (\Gamma, S \oplus b)$.

We first show that $\|\mathsf{P}_{W_0} - \mathsf{P}_{W_1}\| \geq \alpha$ implies that this polarization method produces distribution with statistical distance exponentially close to 1. From Lemma 2.5 we see that we can assume that B (i.e., the random variable which takes values b) is chosen uniformly at random, and then show how to find B from the produced distribution $\mathsf{P}_{\overline{W}_B}$ with error probability exponentially close 1. Again using Lemma 2.5 we see that $\|\mathsf{P}_{W_0} - \mathsf{P}_{W_1}\| \geq \alpha$ implies that there exists a function y such that setting $Y_i := y(W_{Q_b^i})$ gives $\Pr[Y_i = Q_b^i] \geq \frac{1+\alpha}{2}$. Thus we can use the decoding algorithm of Bob which is needed by the key agreement protocol to reconstruct S with probability exponentially close to one. Since $S \oplus B$ is also given we can find B.

Now assume that $\|\mathsf{P}_{W_0} - \mathsf{P}_{W_1}\| \leq \beta$. Consider the tripartite probability distribution P_{XYZ} where $X = Y$ is a uniform random bit, and $Z = W_X$. The distribution P_{XYZ} has correlation 1 and leakage at most β. If the one-message key agreement protocol is run with this distribution, Eve will see a sample of $(W_{Q_b^0}, \ldots, W_{Q_b^{n-1}}, R_b)$. The properties of the protocol imply that this distribution is statistically independent (with high probability) of S. Furthermore, in the output of the polarization method we get exactly the same distribution, plus the value of $S \oplus b$, which is thus independent of the rest, and the resulting distributions must be exponentially close to each other. □

Proof (of Theorem 4.16). Follows from Lemmas 4.17 and 4.18. □

Since we know for which parameters α and β a one-message key agreement protocol exists, we get:

Corollary 4.19. *For constant parameters α and β, there exists an (efficient) oblivious black-box polarization method if and only if $\alpha^2 > \beta$.*

Proof. Using Theorem 4.9, Theorem 4.16, and Theorem 3.17. □

In particular this shows that no oblivious polarization method exists in case $\alpha^2 \leq \beta$, which answers an open question posed in [Vad99].

4.6. Multi Message Key Agreement

In this section we remove the restriction that Alice sends only a single message to Bob. Instead, we assume that Alice and Bob share an authentic channel from Alice to Bob, as well as an authentic channel from Bob to Alice; and they may communicate arbitrarily over these channels.

This scenario is much less understood than the one-message case: no expression for the achievable rate in terms of entropies is known (as opposed to the one-message key rate $S_{\rightarrow}(X;Y|Z)$), and it can be very hard to even decide if for a given distribution P_{XYZ} key agreement with arbitrary communication is possible. The usual upper bound on the rate in terms of entropies is given by the *intrinsic information* $I(X;Y\downarrow Z)$, introduced in [MW99]. It is defined as

$$I(X;Y\downarrow Z) := \min_{\mathsf{P}_{\overline{Z}|Z}} I(X;Y|\overline{Z}),$$

i.e., the mutual entropy of X and Y conditioned on the minimizing random variable $\overline{Z}$ Eve can obtain from Z. While it is known that this bound is not tight [RW03], it is an open problem whether $I(X;Y\downarrow Z) > 0$ implies that key agreement is possible (the usual conjecture is that this is not the case).

Because of these facts, our study of general key agreement is much shorter than the study of one-message key agreement, and we only study random variables where $\mathcal{X} = \mathcal{Y} = \{0,1\}$, and the information of $\mathcal{Z}$ is bounded in a similar way as the leakage.

In the case where arbitrary messages are allowed, most key agreement protocols proceed in two phases: first, in a process called *advantage distillation*, Alice and Bob communicate to obtain random random variables X' and Y' (while giving Eve information Z') which have positive one-message key rate, i.e., $S_{\rightarrow}(X';Y'|Z') > 0$ or $S_{\rightarrow}(Y';X'|Z') > 0$. Then, Alice and Bob apply a one-message key agreement protocol. The intuitive reason why interaction can help is that for some distributions Alice and Bob can first find "good instances" of their random variables without leaking too much information about those.

4.6.1. Considered Bounds

In the one-message case, we bounded the information which is given to Eve by $\mathrm{Adv}^{\max}(X|Z)$; this was possible because the marginal P_{YZ} was not of interest. If arbitrary communication is allowed this is not the case

anymore, and at least both $\mathrm{Adv}^{\max}(X|Z)$ and $\mathrm{Adv}^{\max}(Y|Z)$ will be of interest in this case. However, one must also consider the best advantage possible in predicting X in case $X = Y$ and in case $X \neq Y$. This comes from the fact that a protocol can treat instances with $X = Y$ differently than instances with $X \neq Y$.

We will make the simplifying assumption that Z contains the information whether $X = Y$, i.e., $\mathrm{Adv}^{\max}(X \oplus Y|Z) = 1$.[6] In this case, we only have to consider the amount of information leaked about X in case $X = Y$, and in case $X \neq Y$. As it turns out, with this assumption the quantity of interest is the leakage in case $X = Y$.

Definition 4.20 (Equality leakage). *Let P_{XYZ} be a tripartite probability distribution over $\{0,1\} \times \{0,1\} \times \mathcal{Z}$, and define a new probability distribution $\mathsf{P}_{X'Z'}$ by*

$$\mathsf{P}_{X'Z'}(x,z) := \Pr[X = x, Z = z | X = Y] = \mathsf{P}_{XYZ}(x,x,z)/\mathsf{P}_{XY}(x,x).$$

The equality-leakage β_{eq} of P_{XYZ} is

$$\beta_{\mathrm{eq}} := \mathrm{Adv}^{\max}(X'|Z').$$

We will see later that for a distribution P_{XYZ} key agreement is possible if the equality leakage β_{eq} and the correlation α satisfy $\beta_{\mathrm{eq}} < \frac{2\alpha}{1+\alpha}$. On the other hand, we will see that for any α there exists a distribution P_{XYZ} with equality leakage $\beta_{\mathrm{eq}} = \frac{2\alpha}{1+\alpha}$ (i.e., exactly meeting the bound) for which key agreement is impossible. This distribution has the additional property that $X \neq Y$ implies that X cannot be predicted with advantage exceeding 0 from Z. Thus, our protocol is tight in this sense.

4.6.2. Advantage Distillation

The basic idea of the protocol is that Alice and Bob use the authentic channel to discard the positions where their random variables disagree, without revealing much information about X or Y. This is done using the following advantage distillation protocol: Alice sends Bob the XOR of two random variables X_0 and X_1, and Bob replies whether $Y_0 \oplus Y_1$ is the same value. If the values are the same, Alice and Bob keep their first random variable as output, otherwise they discard both and start the protocol again.

[6] Our protocols will not become insecure in case Z does not contain this information, but they might not be optimal in this case. See the discussion in Section 4.7.

In this context it is convenient to write the correlation α as $\alpha = \frac{1-\vartheta}{1+\vartheta}$ (the parameters will get simpler in that way). Since α is in the interval $[0,1]$, ϑ will be in the interval $[0,1]$ as well, with a larger value denoting a smaller α. For reference, we get the following conversion formulas:

$$\vartheta = \frac{1-\alpha}{1+\alpha}$$
$$\alpha = \frac{1-\vartheta}{1+\vartheta}$$
$$\Pr[X=Y] = \frac{1}{1+\vartheta} = \frac{1+\alpha}{2}.$$

Lemma 4.21. *Let P_{XYZ} be a probability distribution with correlation $\frac{1-\vartheta}{1+\vartheta}$ and equality leakage β_{eq}. There exists a protocol which uses an expected number of $n = 2(1+\vartheta)^2/(1+\vartheta^2)$ random variables and yields random variables with correlation $\frac{1-\vartheta^2}{1+\vartheta^2}$ and equality leakage $1-(1-\beta_{\text{eq}})^2$. Further, the computations of Alice and Bob can be done in time $\mathcal{O}(n)$.*

Proof. Alice and Bob use two instances (X_0, Y_0) and (X_1, Y_1) of the random variables. Alice sends $X_0 \oplus X_1$ to Bob, who checks if this is equal to $Y_0 \oplus Y_1$. If this is the case he notifies Alice that the protocol was successful, and they output $X = X_0$ and $Y = Y_0$, respectively. Otherwise they discard the bits and start over again. Note that Eve will know at which point Alice and Bob accepted.

That probability that Alice and Bob accept is $p_{\text{acc}} := \frac{1}{(1+\vartheta)^2} + \frac{\vartheta^2}{(1+\vartheta)^2} = \frac{1+\vartheta^2}{(1+\vartheta)^2}$, thus the expected number of repetitions of the protocol is $\frac{(1+\vartheta)^2}{1+\vartheta^2}$, and the expected number or random variables used $2\frac{(1+\vartheta)^2}{1+\vartheta^2}$. Further, the probability that $X_0 = Y_0$ and $X_1 = Y_1$ is $\frac{1}{(1+\vartheta)^2}$, which implies that the probability that $X = Y$ holds after the protocol is $\frac{\Pr[(X_1=Y_1)\wedge(X_2=Y_2)]}{p_{\text{acc}}} = \frac{1}{1+\vartheta^2}$, and the correlation is thus $\frac{2}{1+\vartheta^2} - 1 = \frac{1-\vartheta^2}{1+\vartheta^2}$.

We now show that the equality leakage of the new random variables is $1-(1-\beta_{\text{eq}})^2$. Note first that $X = Y$ exactly if $X_0 = Y_0$ and $X_1 = Y_1$. Let B_0 and B_1 be the random variables guaranteed by Lemma 2.2 for the distributions conditioned on $X_0 = Y_0$ and $X_1 = Y_1$. Conditioned on $X = Y$, if both $B_0 = 0$ and $B_1 = 0$, then all functions have advantage 0 in predicting X from (Z_1, Z_2) and the communication. Since this happens with conditional probability $(1-\beta_{\text{eq}})^2$, we obtain the lemma. □

Lemma 4.22 shows how to repeat the steps of Lemma 4.21 multiple times until Alice and Bob have random variables usable in a one-message protocol (i.e., it produces random variables with correlation α and leakage β such that $\alpha^2 > \beta$). It can be applied in case

$$\vartheta < 1 - \beta_{\text{eq}}. \tag{4.9}$$

It is easy to see that (4.9) is equivalent to $\beta_{\text{eq}} < \frac{2\alpha}{1+\alpha}$, and as we mentioned before, we will see that this is optimal.

Lemma 4.22. *Let P_{XYZ} be a probability distribution with correlation $\frac{1-\vartheta}{1+\vartheta}$ and equality leakage β_{eq} such that $\vartheta < 1 - \beta_{\text{eq}}$.*

For $\varphi := \max\left(2, \frac{8}{\log(\frac{1-\beta_{\text{eq}}}{\vartheta})}\right)$, there exists a protocol which uses an expected number of at most $n = \varphi \cdot \frac{1+\vartheta}{1-\vartheta}$ random variables and yields random variables with correlation at least $\alpha \geq \frac{7}{8}$ and leakage at most β such that $\frac{\alpha^2}{\beta} > 1 + \vartheta^\varphi$. The computations of Alice and Bob can be done in time $\mathcal{O}(n)$.

Proof. We assume that $1 - \beta_{\text{eq}} \leq 16\vartheta$ (otherwise we increase β_{eq} accordingly; this can not make the protocol insecure, and φ will not change). We set $\delta := 1 - \beta_{\text{eq}}$ and then choose the parameter $r \in \mathbb{N}$ such that

$$\frac{4}{\log(\frac{\delta}{\vartheta})} \leq 2^r < \frac{8}{\log(\frac{\delta}{\vartheta})} = \varphi. \tag{4.10}$$

To keep the notation simple we set $R := 2^r$. We note for later that (4.10) implies $\frac{\delta}{\vartheta} \geq 2^{4/R}$ and thus $\delta^R \geq 16\vartheta^R$.

We use Lemma 4.21 recursively r times (i.e., the output of one application of Lemma 4.21 is used as input in the next step). From this, we get random variables with correlation $\alpha = \frac{1-\vartheta^R}{1+\vartheta^R} \geq (1-\vartheta^R)^2 \geq 1 - 2\vartheta^R$ (this is at least $7/8$ since $\vartheta^R \leq \frac{1}{16}\delta^R$), and equality leakage at most $1 - \delta^R$. In case $X \neq Y$ the leakage can be at most 1 and we can thus upper bound β as

$$\begin{aligned}\beta &\leq \underbrace{\Pr[X = Y]}_{=\frac{1}{1+\vartheta^R}}(1-\delta^R) + \underbrace{\Pr[X \neq Y]}_{=\frac{\vartheta^R}{1+\vartheta^R}} \\ &= \frac{1-\delta^R+\vartheta^R}{1+\vartheta^R} \leq \frac{1-15\vartheta^R}{1+\vartheta^R} \leq \frac{1}{1+16\vartheta^R}.\end{aligned}$$

Thus, we get for $\frac{\alpha^2}{\beta}$:

$$\begin{aligned}\frac{\alpha^2}{\beta} &\geq (1-4\vartheta^R)(1+16\vartheta^R)\\ &= 1+12\vartheta^R-64\vartheta^{2R}\\ &\geq 1+\vartheta^R+11\vartheta^R(1-6\vartheta^R)\\ &\geq 1+\vartheta^R,\end{aligned}$$

where the last inequality follows from $\vartheta^R \leq \frac{1}{16}\delta^R \leq \frac{1}{16}$. The expected number of random variables used is upper bounded by

$$\begin{aligned}&\frac{2(1+\vartheta)^2}{1+\vartheta^2}\cdot\frac{2(1+\vartheta^2)^2}{1+\vartheta^4}\cdot\frac{2(1+\vartheta^4)^2}{1+\vartheta^8}\cdots\frac{2(1+\vartheta^{2^{r-1}})^2}{1+\vartheta^{2^r}}\\ &\quad= 2^r\frac{(1+\vartheta)^2(1+\vartheta^2)(1+\vartheta^4)\cdots(1+\vartheta^{2^{r-1}})}{1+\vartheta^{2^r}}\\ &\quad\leq 2^r(1+\vartheta)\Big((1+\vartheta)(1+\vartheta^2)(1+\vartheta^4)\cdots(1+\vartheta^{2^{r-1}})\Big)\\ &\quad= 2^r(1+\vartheta)(1+\vartheta+\vartheta^2+\vartheta^3+\ldots+\vartheta^{2^r-1})\\ &\quad= 2^r(1+\vartheta)\frac{1-\vartheta^{2^r}}{1-\vartheta}\\ &\quad\leq 2^r\frac{1+\vartheta}{1-\vartheta}.\end{aligned}$$

It follows directly from Lemma 4.21 that the computation can be done in linear time. □

4.6.3. Combining the Protocols

We now combine the advantage distillation from Lemma 4.22 with the one-message key agreement protocol from Theorem 4.14. This will give our protocol for key agreement with arbitrary messages.

Theorem 4.23. *Let P_{XYZ} be a distribution with correlation at least $\frac{1+\vartheta}{1-\vartheta}$ and equality leakage at most β_{eq}. If $\vartheta < 1-\beta_{\mathrm{eq}}$, there exists a key agreement protocol and an absolute constant $d > 0$, such that the protocol uses n random variables and yields*

$$m = \left\lfloor \frac{dn}{\varphi 2^{4\gamma}}\frac{1-\vartheta}{1+\vartheta}\right\rfloor$$

key bits with secrecy and soundness $1 - 2^{-m}$*, where* $\varphi := \max\left(2, \frac{8}{\log(\frac{1-\beta_{\text{eq}}}{\vartheta})}\right)$*, and* $\gamma := \frac{1}{\log(1+\vartheta^{\varphi})}$*. For this, the protocol needs inputs* ϑ*,* β_{eq}*, and* n*. Further, Alice's and Bob's algorithms run in time* $\mathcal{O}(n^2)$*.*

Proof. We use Lemma 4.22 to get random variables which we can use in Theorem 4.14.

From Lemma 4.22, we see that for $\varphi\frac{1+\vartheta}{1-\vartheta}$ instances of the original random variables we get one instance of a α-correlated random variable with leakage β, such that $\alpha > \frac{7}{8}$ and $\frac{\alpha^2}{\beta} > 1 + \vartheta^{\varphi}$, and thus $\log(\alpha^2/\beta) > \frac{1}{\gamma}$. Applying Theorem 4.14 (note that $\gamma \geq 1$) we see that from n' of those new random variables we get

$$m = \left\lfloor \frac{dn'\alpha^{12\gamma}}{\gamma} \right\rfloor > \left\lfloor \frac{dn'}{\gamma 2^{3\gamma}} \right\rfloor > \left\lfloor \frac{dn'}{2^{4\gamma}} \right\rfloor$$

random variables with soundness and secrecy $1 - 2^{-m}$. Thus, from n of the initial random variables we can obtain

$$\left\lfloor \frac{dn}{\varphi 2^{4\gamma}} \frac{1-\vartheta}{1+\vartheta} \right\rfloor$$

key bits with the required security. □

4.6.4. Impossibility

In this section, we show that our bound is tight. For this we present, for every $\vartheta \in [0,1]$, a distribution P_{XYZ} which has correlation $\frac{1-\vartheta}{1+\vartheta}$, equality leakage $1 - \vartheta$, secrecy in case $X \neq Y$, and which Alice and Bob can obtain by a protocol even if they have no shared randomness before. Clearly such random variables cannot help in obtaining a key since otherwise there would be a key agreement protocol which does not need a priori information.

Alice and Bob obtain these random variables by mixing two strategies: either, they both pick a bit uniformly and independently of each other as random variable, or Alice sends a uniform bit to Bob which they both use.

Theorem 4.24. *For any* $\vartheta \in [0,1]$*, there exists a distribution* P_{XYZ} *such that Alice and Bob can produce random variables* X *and* Y *using two bits of communication* Z *from Alice to Bob. Further,* P_{XYZ} *has correlation* $\frac{1-\vartheta}{1+\vartheta}$*, equality leakage* $\beta_{\text{eq}} = 1 - \vartheta$*, and satisfies* $\mathsf{P}_{Z|X=0,Y=1} = \mathsf{P}_{Z|X=1,Y=0}$*.*

Proof. Alice chooses a bit b_0 such that $\Pr[b_0 = 0] = \frac{2\vartheta}{1+\vartheta}$ and an independent uniform bit b_1. She then sends these bits to Bob. If $b_0 = 0$ then Alice outputs another uniform random bit as X, and Bob outputs a uniform random bit as Y. In case $b_0 = 1$ they output $X = Y = b_1$.

The probability that they output the same bit is $\frac{\vartheta}{1+\vartheta} + \frac{1-\vartheta}{1+\vartheta} = \frac{1}{1+\vartheta}$ (and thus the correlation is $\frac{1-\vartheta}{1+\vartheta}$). Further, predicting X from Z in case $X = Y$ is possible with advantage 1 if $b_0 = 1$, and with advantage 0 otherwise. Conditioned on $X = Y$ we have $b_0 = 1$ with probability $1 - \vartheta$, and thus the equality leakage is $1 - \vartheta$.

Finally, if Alice and Bob do not have the same bit then $b_0 = 0$ and thus the distribution of the communication is independent of the bits of Alice and Bob. □

Thus, we have an exact characterization of when key agreement is possible in this case. This is stated in the following corollary.

Corollary 4.25. *Let $\vartheta \in [0,1]$, $\beta_{\text{eq}} \in [0,1]$ be constants. If $\vartheta < 1 - \beta_{\text{eq}}$, then there exists a key agreement protocol for random variables distributed according to any distribution P_{XYZ} over $\{0,1\} \times \{0,1\} \times \mathcal{Z}$ with correlation at least $\frac{1-\vartheta}{1+\vartheta}$ and equality leakage at most β_{eq}. If $\vartheta \geq 1 - \beta_{\text{eq}}$ then there exists a distribution P_{XYZ} with correlation $\frac{1-\vartheta}{1+\vartheta}$ and equality leakage β_{eq} such that for this distribution no key agreement is possible.*

Proof. The first part follows from Theorem 4.23. The second part follows from Theorem 4.24, and the fact that no key agreement is possible using an authentic channel and arbitrary communication only [Mau93]. □

4.7. Discussion of the Results

Let P_{XYZ} be a probability distribution over $\{0,1\} \times \{0,1\} \times \mathcal{Z}$, with correlation α, leakage β and equality-leakage β_{eq}. If

$$\alpha^2 > \beta, \tag{4.11}$$

then key agreement is possible with a single message from Alice to Bob (Theorem 4.14). If

$$\beta_{\text{eq}} < \frac{2\alpha}{1+\alpha}, \tag{4.12}$$

then key agreement is possible with two-way communication. We cannot compare (4.11) and (4.12) directly, as one deals with the leakage while the

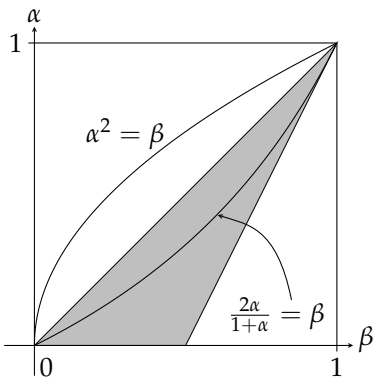


Figure 4.2.: Feasible regions for key agreement for distributions with correlation α and leakage β.

other deals with the equality leakage. In the following we distinguish the case where Z contains the information whether $X = Y$ from the case where the honest parties have (partial) secrecy about this information.

Distributions which leak complete information about the equality

First, we only consider distributions P_{XYZ} which have the property that the information whether X equals Y can be inferred from Z. If we define the inequality leakage as $\beta_{\text{neq}} := \text{Adv}^{\max}(X'|Z')$ where $\mathsf{P}_{X'Z'}(x,z) := \Pr[X{=}x, Z{=}z|X{\neq}Y]$, then we get for such distributions

$$\beta = \Pr[X{=}Y]\beta_{\text{eq}} + \Pr[X{\neq}Y]\beta_{\text{neq}} = \frac{1+\alpha}{2}\beta_{\text{eq}} + \frac{1-\alpha}{2}\beta_{\text{neq}}. \tag{4.13}$$

Now, consider the case $\beta_{\text{eq}} := \frac{2\alpha}{1+\alpha}$, i.e., β_{eq} is exactly such that key agreement is not possible anymore. If we insert this expression for β_{eq} into (4.13) and use $\beta_{\text{neq}} \in [0,1]$ we get the following inequalities which β must satisfy in case key agreement is *exactly not* possible anymore:

$$\alpha \leq \beta \leq \frac{1+\alpha}{2}.$$

This is illustrated in Figure 4.2. In the area on the top left, (above $\alpha^2 = \beta$), one-message key agreement is possible, below this line one-message key agreement is impossible. Between this line and the shaded

area, key agreement is possible, and in the shaded area, key agreement *may* be possible depending on the (in)equality leakage. In case $\beta = \beta_{\text{eq}}$, the second solid line is relevant ($\frac{2\alpha}{1+\alpha} = \beta$).

Distributions which do not leak complete information about the equality

From Figure 4.2 it seems clear that whenever our one-message protocol (i.e., Theorem 4.14) works, the protocol using arbitrary messages (i.e., Theorem 4.23) could also be used. However, this is not the case for distributions which do not satisfy (4.13) (which is the case if the honest parties have secrecy about the information whether their random variables are equal).

As an example, consider the distribution P_{XYZ} over $\{0,1\} \times \{0,1\} \times \{0,1,\perp\}$, parameterized by the correlation $\alpha \in [0,1]$ and a second parameter $\varepsilon \in [0,1]$, which is defined as follows:

$$\mathsf{P}_X(0) := \mathsf{P}_X(1) := \frac{1}{2},$$

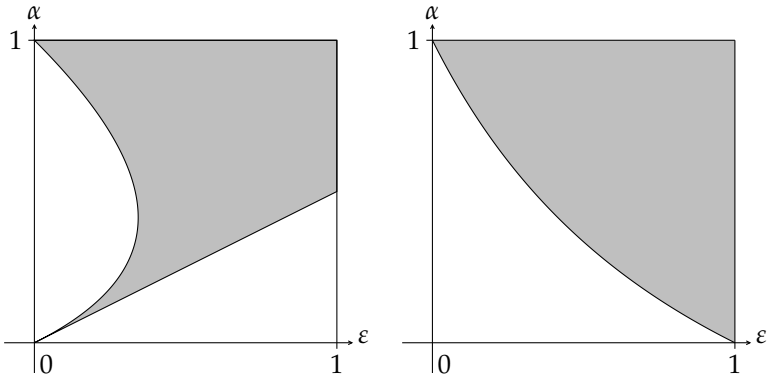
$$\mathsf{P}_{Y|X}(y|x) := \begin{cases} \frac{1+\alpha}{2} & \text{if } x = y, \\ \frac{1-\alpha}{2} & \text{otherwise,} \end{cases}$$

$$\mathsf{P}_{Z|XY}(z|x,y) := \begin{cases} \varepsilon & \text{if } z = \perp \wedge\ x = y, \\ 1-\varepsilon & \text{if } z = y \wedge\ x = y, \\ 1 & \text{if } z = y \wedge\ x \neq y, \\ 0 & \text{otherwise.} \end{cases}$$

First, we note that for any $\varepsilon > 0$, one-message key agreement from Alice to Bob is possible since $H(X|Z) > H(X|Y)$. But this does not mean that we can infer that key agreement is possible if we only know the correlation α and the leakage β (or the correlation and the equality leakage β_{eq}).

The correlation of P_{XYZ} is α, and the equality-leakage β_{eq} is $1 - \varepsilon$. To compute the leakage β we first find

$$\mathsf{P}_{Z|X=0}(z) := \begin{cases} (1-\varepsilon)\left(\frac{1+\alpha}{2}\right) & \text{if } z = 0, \\ \frac{1-\alpha}{2} & \text{if } z = 1, \\ \varepsilon\left(\frac{1+\alpha}{2}\right) & \text{if } z = \perp, \end{cases} \tag{4.14}$$



(a) One-message key agreement.

(b) Key agreement with arbitrary messages.

Figure 4.3.: Shaded are the regions for which our protocols achieve key agreement on the distribution P_{XYZ} given in the text.

and

$$\mathsf{P}_{Z|X=1}(z) := \begin{cases} \frac{1-\alpha}{2} & \text{if } z = 0, \\ (1-\varepsilon)\left(\frac{1+\alpha}{2}\right) & \text{if } z = 1, \\ \varepsilon\left(\frac{1+\alpha}{2}\right) & \text{if } z = \perp, \end{cases} \tag{4.15}$$

which gives (using Lemma 2.5) $\beta = \left|(1-\varepsilon)(\frac{1+\alpha}{2}) - \frac{1-\alpha}{2}\right| = |\alpha - \frac{\varepsilon}{2} - \frac{\alpha\varepsilon}{2}|$.

From the above we can find the values for α and ε for which $\alpha^2 > \beta$, and thus for which our one-message protocol works. The region is shaded dark in Figure 4.3 (a). The protocol with arbitrary messages works if $\beta_{\text{eq}} < \frac{2\alpha}{1+\alpha}$, and the corresponding region is shown in Figure 4.3 (b). Note that for some parameters one protocol works while for other parameters the other protocol works. The reason for this paradoxical situation is that in the above distribution, Z does *not* contain the information whether $X = Y$. In this case our protocols are not optimal. In particular, (4.13) does not hold for such a distribution. It is an open problem to understand these cases better.

Part II.

Computationally Secure Key Agreement

5. Computational Security

This chapter introduces basic concepts needed for our studies of computationally secure key agreement. It also contains an example of a simple black-box reduction.

Overview of this chapter

Section 5.1 explains the computational models we use in this part of the thesis. In particular, we define circuits, oracle circuits, and oracle algorithms. We also give two basic theorems about these concepts we need later (we count the number of predicates which can be computed by circuits of fixed size and we show that circuits can simulate Turing machines).

Section 5.2 contains an example of a computational security proof. We use this example to explain the basic concept of a black-box security proof and illustrate the difference between uniform and non-uniform security.

5.1. Computational Models

Usual models of computation include random access machines, single and multi-tape Turing machines, and others. We assume that the reader is familiar with at least one of these models. Since in most cases the exact model at hand is irrelevant, we are not specific about it, but simply speak of algorithms. In this subsection we quickly explain oracle algorithms, where an arbitrary model from above gets enhanced by one or more oracles. Also, we discuss circuits as a computational model.

5.1.1. Oracle Algorithms

For an arbitrary function $f : \{0,1\}^* \to \{0,1\}^*$, an oracle algorithm $A^{(f)}$ is an algorithm which can call f as a subroutine. For such algorithms, the computation required to evaluate f is neglected; instead calling f requires just one single step in the computation. However, we often count the number of oracle calls separately if we want a more exact analysis.

If multiple functions $f_0, \ldots, f_{t-1}$ are given we use $A^{(f_0,\ldots,f_{t-1})}$ in a similar way: the algorithm may call any of the t functions in arbitrary order. In this case we may count the calls the algorithm does to the different oracles separately.

5.1.2. Circuits

Circuits are a computational model which are slightly less known than Turing machines or random access machines. Since we use circuits extensively in this thesis we discuss this model here in some detail. The main difference between circuits and a more usual model (like Turing machines) is that a circuit is restricted to a single problem size: a circuit can only compute a function $f : \{0,1\}^{\ell_{\text{in}}} \to \{0,1\}^{\ell_{\text{out}}}$ for fixed ℓ_{in} and fixed ℓ_{out}, while a Turing machine has well defined behavior for all input sizes. A model which treats all input sizes with one object (e.g., Turing machines) is often called *uniform*, while a model in which each input size is treated separately is *non-uniform* (e.g., circuits).

We define a circuit as sequence of gates; the first ℓ_{in} gates are input gates (and will be directly associated with the respective input) and the last ℓ_{out} gates are output gates. Except for the input gates, a gate is described by a triple (j, k, f), where f is a function and j and k are integers which designate the inputs of this function.

Definition 5.1 (Circuit). *A circuit C with ℓ_{in} inputs and ℓ_{out} outputs is a finite tuple $(g_{\ell_{\text{in}}}, \ldots, g_{s-1})$ of gates (where $s > \ell_{\text{in}}$ and $s > \ell_{\text{out}}$). A gate g_i is a triple $g_i := (j_i, k_i, f_i)$ such that $0 \leq j_i \leq k_i < i$, and f_i is a function $f_i : \{0,1\}^2 \to \{0,1\}$. The size of a circuit C is $\text{Size}(C) := s$.*

Every circuit computes a function $f : \{0,1\}^{\ell_{\text{in}}} \to \{0,1\}^{\ell_{\text{out}}}$ which is defined recursively as follows. First, for $0 \leq i < \ell_{\text{in}}$ let the values v_i be the i-th bit of the input x, i.e., $v_i := x|_i$. For $\ell_{\text{in}} \leq i < s$ where $g_i = (j_i, k_i, f_i)$ let $v_i := f_i(v_{j_i}, v_{k_i})$. The output y of the function is the bit string with bits $v_{s-\ell_{\text{out}}}$ to v_{s-1}, i.e., $y|_i := v_{s-\ell_{\text{out}}+i}$.

Given an arbitrary function $f : \{0,1\}^{\ell_{\text{in}}} \to \{0,1\}^{\ell_{\text{out}}}$ we can construct a circuit C which computes this function (for example by writing the boolean formula for every output bit in disjunctive normal form). This is in harsh contrast to uniform computation, where some functions are not computable.

Similar to Turing machines, we can consider *oracle circuits*. For a function $f : \{0,1\}^r \to \{0,1\}$ an oracle circuit C^f is defined like a usual circuit, but some of the gates may be oracle gates $(f, j_{i,0}, j_{i,1}, \ldots, j_{i,r-1})$. Again we

can count the number of function calls to f separately (but the size is still the number of total gates). For a set $\mathcal{F}$ of functions an oracle circuit $C^{\mathcal{F}}$ is defined such that it can call any function from $\mathcal{F}$.

Circuit complexity

We first get a well known fact about circuit complexity (originally due to Shannon [Sha49b]):

Lemma 5.2. *The set of circuits of size at most s with ℓ inputs and one output compute less than $(45s)^s$ functions $f : \{0,1\}^\ell \to \{0,1\}$.*

Proof. Let $\mathfrak{C}$ be the set of all tuples $\big((j_\ell, k_\ell, f_\ell), \ldots, (j_{s-1}, k_{s-1}, f_{s-1})\big)$ where $0 \leq j_i < s$, $0 \leq k_i < s$, and $f_i : \{0,1\}^2 \to \{0,1\}$ is arbitrary (i.e., like circuits, but the restrictions on j_i and k_i are relaxed). There are $(16s^2)^{s-\ell}$ such $(s-\ell)$-tuples. Let $C \in \mathfrak{C}$ be such an $(s-\ell)$-tuple which additionally satisfies $0 \leq j_i \leq k_i < i$ for all i (i.e., C is a circuit), and which also satisfies that no triple (j_i, k_i, f_i) occurs twice. We can associate $(s-\ell-1)!$ different tuples in $\mathfrak{C}$ with C by permuting all but the last entry (The idea is that we permute the labelling of the gates. Formally we choose a permutation $\pi : \{0, \ldots, s-1\} \to \{0, \ldots, s-1\}$ which satisfies $\pi(j) = j$ for $0 \leq j < \ell - 1$ and then consider the tuple

$$\big((\pi^{-1}(j_{\pi(\ell)}), \pi^{-1}(k_{\pi(\ell)}), f_{\pi(\ell)}), \ldots, (\pi^{-1}(j_{\pi(s-1)}), \pi^{-1}(k_{\pi(s-1)}), f_{\pi(s-1)})\big),$$

which can easily be seen to correspond to this intuition of permuting labellings.) All these tuples are different (because given the initial circuit and the obtained tuple it is possible to find the permutation when starting from the inputs — here we use that no two entries in the tuple are the same), all are in the initial set $\mathfrak{C}$, and for no two circuits which compute different functions we will obtain the same tuple.

For every circuit of size at most s we can give a circuit of size s which computes the same function and satisfies that no triple (j_i, k_i, f_i) occurs twice, which means that for every function computed by a circuit of size at most s we identified $(s-\ell-1)!$ distinct entries in $\mathfrak{C}$.

Thus, the number of functions computed by circuits of size s is at most (where we use $n! > n^n e^{-n}$ which follows from Stirling's formula)

$$\begin{aligned}\frac{(16s^2)^{s-\ell}}{(s-\ell-1)!} &= (s-\ell)\frac{(16s^2)^{s-\ell}}{(s-\ell)!} \\ &< (s-\ell)(16e)^{s-\ell}s^s\frac{s^{s-2\ell}}{(s-\ell)^{s-\ell}} \quad < s\cdot(44s)^s\frac{s^{s-2\ell}}{(s-\ell)^{s-\ell}}.\end{aligned}$$

For natural numbers $a > b$ one can show (by expanding) that $(a+1)^b \leq a^{b+1}$. Thus $\frac{s^{s-2\ell}}{(s-\ell)^{s-\ell}}$ is at most 1, which proves the lemma. □

From the above lemma, we immediately get that some functions need circuits of size at least $2^\ell/\ell$:

Corollary 5.3. *For every $\ell > 45$, there exists a function $f : \{0,1\}^\ell \to \{0,1\}$ such that the smallest circuit which computes f has size at least $2^\ell/\ell$.*

Proof. According to Lemma 5.2, circuits of size $2^\ell/\ell$ compute at most

$$\left(45\frac{2^\ell}{\ell}\right)^{\frac{2^\ell}{\ell}} = \underbrace{\left(\frac{45}{\ell}\right)^{\frac{2^\ell}{\ell}}}_{<1}\left(2^\ell\right)^{\frac{2^\ell}{\ell}} < 2^{2^\ell}$$

different functions. Since there are 2^{2^ℓ} functions $f : \{0,1\}^\ell \to \{0,1\}$ at least one of them cannot be computed by a circuit of size $2^\ell/\ell$. □

Lupanov [Lup58] showed (see also [Weg87, Section 4.2]) that every function $f : \{0,1\}^\ell \to \{0,1\}$ can be computed by a circuit C of $\text{Size}(C) = \frac{2^\ell}{\ell}(1+o(1))$. Thus, Corollary 5.3 is tight in this sense.

Constructing circuits from algorithms

Assume that there is a polynomial time algorithm A which computes a function family $f_k : \{0,1\}^k \to \{0,1\}$ for every $k \in \mathbb{N}$. Is it true that for every k there exists a small circuit which computes f_k? The answer is yes, since circuits can simulate algorithms. This is given in the following well known theorem. In it, the exact computational model is relevant; we use (single tape) Turing machines to model algorithms here.

Theorem 5.4. *Let $f_k : \{0,1\}^k \to \{0,1\}$ be an infinite function family which can be evaluated by a Turing machine in time $g(n)$. Then, there exists an infinite family of circuits $\{C_k\}$ of size $\text{Size}(C_k) \in \mathcal{O}(g^2(n))$ such that C_k computes f_k (i.e., the function f restricted on inputs of length k).*

Proof. Let M be a Turing machine which decides f in time $g(n)$ (i.e., after $g(n)$ steps it ends either in a special accepting or special rejecting state, depending on $f(x)$). Clearly, M cannot use more than $2g(n)+1$ cells on the tape (namely, the $g(n)$ on the left hand side of the starting position, the $g(n)$ on the right hand side of the starting position, and the one at the starting position). For every time step, we now use gates as follows:

- A constant number of gates which compute the state of the Turing Machine in this time step,
- $\lceil \log(2g(n)+1) \rceil$ gates which compute an encoding of the position of the head at this time step,
- a constant number of gates for each of the $2g(n)+1$ possibly used cells, which compute the contents of this cell at this time step.

For every time step we use $\mathcal{O}(g(n))$ gates. Further, it is easy to see that for every time step it is possible to compute this information with the described amount of gates, given the information of the previous time step. This also holds at the beginning, if the input is given. Finally, a constant number of gates suffice to output 1 if the machine ends in the accepting state and 0 otherwise. □

5.2. Black-Box Security Proofs

5.2.1. Introduction

An important task in cryptography is to base primitives on other, usually seemingly simpler primitives. For example, a pseudorandom generator (a function which expands the input and whose output is indistinguishable from uniform random bits) can be built from an arbitrary one-way function, as shown by Håstad, Impagliazzo, Levin, and Luby [HILL99] (see also [ILL89, Hås90]). Such constructions can be complicated, but up to very few examples they always have the following simple structure (see also Figure 5.1): in a first part, the more complicated primitive g is built based on the given primitive f, i.e., an algorithm $g^{(\cdot)}$ is given which uses oracle access to f. In a second part, it is shown how every adversary A_g which breaks g can be used to get an adversary A_f which breaks f. For this one gives an algorithm $A_f^{(\cdot,\cdot)}$ which uses oracle access to A_g and f such that whenever A_g breaks g, $A_f^{(A_g,f)}$ breaks f. Clearly this implies that no efficient A_g exists, as otherwise we would obtain an algorithm A_f which breaks f efficiently. Such proofs usually work for *all* functions f and A_g, and not only for efficiently implementable ones. In this case, such a proof is called a black-box reduction, since it only considers the input and output behavior of the functions f and A_g; it treats these functions as black box.

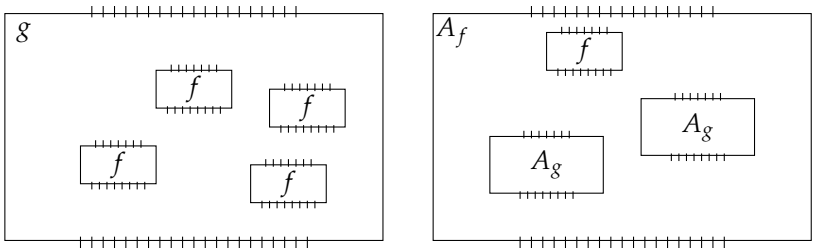


Figure 5.1.: A cryptographic construction of a primitive g from a primitive f. In (a), a way to build g from f is given. In (b), an adversary A_g for g is used to build an adversary $A_f^{(A_g,f)}$ for f.

To discuss parameters of such black-box reductions we give a simple example of such a reduction. It shows how to strengthen one-way functions (i.e., functions which are easy to evaluate but hard-to invert).

5.2.2. Example: Strengthening One-way Functions

In the following definition of (weak) one-way functions, algorithm A may use some randomness $\mathfrak{R}_A$, which we assume to be a uniformly chosen bit string of appropriate length.

Definition 5.5 (One-way function). *For a function $\delta(k) : \mathbb{N} \to [0,1]$, computable in time $\mathrm{poly}(k)$, a function $f : \{0,1\}^* \to \{0,1\}^*$ is a δ-weak one-way function if*

- *there is an algorithm running in time $\mathrm{poly}(k)$ which outputs $f(x)$ on input $x \in \{0,1\}^k$, and*
- *any algorithm $A(\cdot,\cdot,\cdot)$ running in time $\mathrm{poly}(k)$ satisfies*

$$\Pr_{\substack{X \leftarrow \{0,1\}^k \\ \mathfrak{R}_A}} [f(A(k, f(X), \mathfrak{R}_A)) = f(X)] < \delta(k)$$

for all but finitely many k.

A function $f : \{0,1\}^* \to \{0,1\}^*$ is a strong one-way function if it is a $\frac{1}{p(k)}$-weak one-way function for every polynomial $p(k)$.

We use the following conventions to simplify notation: first, we write $f : \{0,1\}^k \to \{0,1\}^\ell$, where we understand that the function f is defined for all $k \in \mathbb{N}$ and assume that the output length for all $x \in \{0,1\}^k$ is exactly $\ell(k)$ (this is no loss of generality; otherwise it is possible to pad the output appropriately). Second, we omit the parameter k as an input to the algorithm A and to δ from now on. Finally, we omit $\mathfrak{R}_A$ as an argument to A (however, we still make explicit which probability is over this randomness, see the following theorem for an example).

The next theorem (this can be traced back to [Yao82]) shows how to increase the difficulty of inverting one-way functions.

Theorem 5.6. *Let $\varepsilon : \mathbb{N} \to [0,1]$, $\delta : \mathbb{N} \to [0,1]$, and $s : \mathbb{N} \to \mathbb{N}$ be functions computable in time* $\mathrm{poly}(k)$*. Define*

$$g^{(f)}(x^s) := f(x_0) \parallel \cdots \parallel f(x_{s-1}).$$

There is an algorithm $A_f^{(\cdot,\cdot)}(\cdot)$, running in time $\mathrm{poly}(\frac{1}{\varepsilon}, s, k)$ *such that for any function A_g with*

$$\Pr_{\substack{X^s \leftarrow \{0,1\}^{ks} \\ \mathfrak{R}_{A_g}}}[g(A_g(g(X^s))) = g(X^s)] > \delta^s + \varepsilon,$$

$A_f^{(A_g,f)}$ *makes* $\lceil \frac{2s^2}{\varepsilon} \ln(\frac{4s}{\varepsilon}) \rceil$ *calls to A_g and satisfies:*

$$\Pr_{\substack{X \leftarrow \{0,1\}^k \\ \mathfrak{R}_{A_f}}}[f(A_f(f(X))) = f(X)] > \delta.$$

Before we go to the proof of Theorem 5.6 we show how it can be used to obtain a strong one-way function from a weak one-way function:

Corollary 5.7. *If there exists a δ-weak one-way function with $1 - \delta(k) > \frac{1}{p(k)}$ for some polynomial $p(k)$ then there exists a strong one-way function.*

Proof. In Theorem 5.6 choose $s(k) := kp(k)$. Note that there is an algorithm which computes $s(k)$ efficiently (since $p(k)$ is a polynomial). Since $\delta^{s(k)} = (1 - \frac{1}{p(k)})^{s(k)} \leq e^{-\frac{s(k)}{p(k)}} = e^{-k}$ any polynomial time algorithm which inverts g with non-negligible probability can be used to invert f with probability exceeding δ infinitely often and in polynomial time. □

It is a question of style whether one prefers formulations as in Theorem 5.6 or as in Corollary 5.7; in the literature it is more common to state such results solely in a form similar to Corollary 5.7.

Comparing the statements, the advantage of Corollary 5.7 is obvious: it is easier to state and easier to understand. However Theorem 5.6 also has some advantages. First, the parameters of the reduction are explicit; clearly, a construction of g from f which uses fewer calls is preferable to one which uses more calls; analogously we would like A_g to be called as few times as possible by A_f (the most important resource such a construction should minimize is probably the amount of randomness needed in order to use the construction). Second, Theorem 5.6 is independent of the definition of (weak) one-way functions, and thus arbitrary choices we made when defining these terms are irrelevant when only considering Theorem 5.6. Finally, it is possible that Theorem 5.6 is interesting in an information theoretic setting, because it does not require A_g or f to be efficiently computable.

Proof (of Theorem 5.6). Algorithm A_f, which gets input $f(x)$, first chooses a position $i \in \{0, \ldots, s-1\}$ at random, then chooses values $x_0, \ldots, x_{i-1}$, $x_{i+1}, \ldots, x_{s-1}$ from $\{0,1\}^k$ at random, and then runs

$$A_g(f(x_0)\| \ldots \|f(x_{i-1})\|f(x)\|f(x_{i+1})\| \ldots \|f(x_{s-1})).$$

It then checks whether the potential preimage of $f(x)$ is indeed a preimage of $f(x)$ (using the oracle f). If so it returns it, otherwise it repeats the above $r := \lceil \frac{2s^2}{\varepsilon} \ln(\frac{4s}{\varepsilon}) \rceil$ times. The bounds on the run time and the number of oracle calls of the algorithm are immediate.

We now analyze the success probability of this algorithm. Since the inputs to the subsequent calls to A_g are not independent this is not completely straightforward. First note that for a fixed $f(x)$ every call to A_g has the same distribution. Thus, consider the probability p_x that one call to A_g inverts conditioned on that the input to A_f is $f(x)$:

$$p_x := \Pr_{\substack{X^s \leftarrow \{0,1\}^{ks} \\ I \leftarrow \{0,\ldots,s-1\}, \Re_{A_g}}} \Big[g(A_g(g(X_0, \ldots, X_{I-1}, x, X_{I+1}, \ldots, X_{s-1}))) = g(X_0, \ldots, X_{I-1}, x, X_{I+1}, \ldots, X_{s-1}) \Big]$$

Let $\mathcal{S}$ be the set for which $p_x > \frac{\varepsilon}{2s^2}$, i.e.,

$$\mathcal{S} := \Big\{ x \in \{0,1\}^k \;\Big|\; p_x > \frac{\varepsilon}{2s^2} \Big\},$$

and (for ease of notation) set $\mu(\mathcal{S}) := |\mathcal{S}|\, 2^{-k} = \Pr_{X \leftarrow \{0,1\}^k}[X \in \mathcal{S}]$. Then, the probability that A_f inverts f is at least:

$$\begin{aligned}
&\Pr_{\substack{X \leftarrow \{0,1\}^k \\ \mathfrak{R}_{A_f}}}[f(A_f(f(X))) = f(X)] \\
&\qquad \geq \Pr_{X \leftarrow \{0,1\}^k}[X \in \mathcal{S}] \cdot \left(1 - \left(1 - \frac{\varepsilon}{2s^2}\right)^{\frac{2s^2}{\varepsilon}\ln(4s/\varepsilon)}\right) \\
&\qquad \geq \mu(\mathcal{S}) \cdot \left(1 - \left(e^{-\frac{\varepsilon}{2s^2}}\right)^{\frac{2s^2}{\varepsilon}\ln(4s/\varepsilon)}\right) \\
&\qquad = \mu(\mathcal{S})\left(1 - \frac{\varepsilon}{4s}\right). \qquad (5.1)
\end{aligned}$$

We now show that $\mu(\mathcal{S})$ is big because otherwise A_g would not have the required success probability. For this, we have to shorten the notation somewhat. In the following all probabilities are over the random choices of X_0 to X_{s-1} and the randomness of A_g. Further, let $\mathcal{I}$ be the event (depending on X_0 to X_{s-1} and $\mathfrak{R}_{A_g}$) that $A_g(g(X_0, \ldots, X_{s-1}))$ finds a preimage. With these conventions we can write the probability that A_g inverts g as follows:

$$\begin{aligned}
&\Pr[\mathcal{I}] = \\
&\quad \Pr[\mathcal{I} | X_0 \notin \mathcal{S}] \Pr[X_0 \notin \mathcal{S}] \\
&+ \Pr[\mathcal{I} | X_0 \in \mathcal{S} \wedge X_1 \notin \mathcal{S}] \Pr[X_0 \in \mathcal{S} \wedge X_1 \notin \mathcal{S}] \\
&+ \Pr[\mathcal{I} | X_0 \in \mathcal{S} \wedge X_1 \in \mathcal{S} \wedge X_2 \notin \mathcal{S}] \Pr[X_0 \in \mathcal{S} \wedge X_1 \in \mathcal{S} \wedge X_2 \notin \mathcal{S}] \\
&+ \qquad \ldots \\
&+ \Pr[\mathcal{I} | X_0 \in \mathcal{S} \wedge \cdots \wedge X_{s-2} \in \mathcal{S} \wedge X_{s-1} \notin \mathcal{S}] \times \\
&\qquad \Pr[X_0 \in \mathcal{S} \wedge \cdots \wedge X_{s-2} \in \mathcal{S} \wedge X_{s-1} \notin \mathcal{S}] \\
&+ \Pr[\mathcal{I} | X_0 \in \mathcal{S} \wedge \cdots \wedge X_{s-1} \in \mathcal{S}] \Pr[X_0 \in \mathcal{S} \wedge \cdots \wedge X_{s-1} \in \mathcal{S}]. \qquad (5.2)
\end{aligned}$$

We now upper bound the summands on the right hand side of (5.2): first, for the last summand we get

$$\begin{aligned}
&\Pr[\mathcal{I} | X_0 \in \mathcal{S} \wedge X_1 \in \mathcal{S} \wedge \cdots \wedge X_{s-1} \in \mathcal{S}] \times \\
&\qquad \Pr[X_0 \in \mathcal{S} \wedge X_1 \in \mathcal{S} \wedge \cdots \wedge X_{s-1} \in \mathcal{S}] \\
&\qquad\quad \leq 1 \cdot \Pr[X_0 \in \mathcal{S} \wedge X_1 \in \mathcal{S} \wedge \cdots \wedge X_{s-1} \in \mathcal{S}] = \mu(\mathcal{S})^s. \qquad (5.3)
\end{aligned}$$

Further for any i we have $\frac{\varepsilon}{2s} \geq \Pr[\mathcal{I} | X_i \notin \mathcal{S}]$. (To see this, consider any fixed value $x \notin \mathcal{S}$. If $\Pr[\mathcal{I} | X_i = x] > \frac{\varepsilon}{2s}$ one iteration of our algorithm A_f

has at least probability $\frac{\varepsilon}{2s^2}$ in inverting on input $f(x)$, and thus $x \in \mathcal{S}$.) We thus get for any event $\mathcal{E}$

$$\begin{aligned}\frac{\varepsilon}{2s} &\geq \Pr[\mathcal{I}|X_i \notin \mathcal{S}] \\ &= \Pr[\mathcal{I}|(X_i \notin \mathcal{S}) \wedge \mathcal{E}] \Pr[\mathcal{E}|X_i \notin \mathcal{S}] + \Pr[\mathcal{I}|(X_i \notin \mathcal{S}) \wedge \overline{\mathcal{E}}] \Pr[\overline{\mathcal{E}}|X_i \notin \mathcal{S}] \\ &\geq \Pr[\mathcal{I}|(X_i \notin \mathcal{S}) \wedge \mathcal{E}] \Pr[\mathcal{E}|X_i \notin \mathcal{S}],\end{aligned}$$

and thus for $\mathcal{E} := X_0 \in \mathcal{S} \wedge X_1 \in \mathcal{S} \wedge \cdots \wedge X_{i-1} \in \mathcal{S}$

$$\begin{aligned}&\Pr[X_0 \in \mathcal{S} \wedge \cdots \wedge X_{i-1} \in \mathcal{S} \wedge X_i \notin \mathcal{S}] \times \\ &\quad \Pr[\mathcal{I}|X_0 \in \mathcal{S} \wedge \cdots \wedge X_{i-1} \in \mathcal{S} \wedge X_i \notin \mathcal{S}] \\ &\qquad = \Pr[X_i \notin \mathcal{S}] \Pr[\mathcal{E}|X_i \notin \mathcal{S}] \Pr[\mathcal{I}|\mathcal{E} \wedge (X_i \notin \mathcal{S})] \\ &\qquad \leq \Pr[X_i \notin \mathcal{S}] \frac{\varepsilon}{2s} < \frac{\varepsilon}{2s}, \qquad (5.4)\end{aligned}$$

which upper bounds every summand in (5.2) except the last one. We combine the requirement on A_g in the theorem, (5.2), (5.3), and (5.4), and get

$$\delta^s + \varepsilon \leq \Pr[\mathcal{I}] \leq s \cdot \frac{\varepsilon}{2s} + \mu(\mathcal{S})^s$$

which implies

$$\delta^s + \frac{\varepsilon}{2} \leq \mu(\mathcal{S})^s.$$

Using (5.1) we thus get for the probability that A_f inverts f

$$\begin{aligned}&\Big(\Pr_{\substack{X \leftarrow \{0,1\}^k \\ \mathfrak{R}_{A_f}}}[f(A_f(f(X))) = f(X)]\Big)^s \\ &\qquad \geq \mu(S)^s \Big(1 - \frac{\varepsilon}{4s}\Big)^s \geq \Big(\delta^s + \frac{\varepsilon}{2}\Big)\Big(1 - \frac{\varepsilon}{4s}\Big)^s \\ &\qquad > \Big(\delta^s + \frac{\varepsilon}{2}\Big)\Big(1 - \frac{\varepsilon}{4}\Big) \geq \delta^s,\end{aligned}$$

which proves the theorem. □

5.2.3. Non-uniform Security

In this section we discuss the difference between non-uniform and uniform security as well as non-uniform and uniform reductions.

Non-uniform security definitions

Let $f : \{0,1\}^k \to \{0,1\}^\ell$ be a strong one-way function with uniform security (i.e., as in Definition 5.5). Then, it may happen that there exists a small polynomial $p(\cdot)$ (for example $p(k) = k^2$) such that for every k there is a circuit C_k of size $p(k)$ which inverts f with probability $\frac{1}{2}$. If this is the case then it must be hard to compute the circuit C_k for a given k (as otherwise we immediately get an algorithm contradicting the hardness: we compute the circuit and then evaluate it).

If one uses a one-way function in practice, one probably hopes that this is not the case, since even if the circuit is hard to find for a computer, it would be more comforting if the security of a one-way function guarantees that no such circuit exists. For example, one could hope that the smallest circuit inverting f with probability more than $2^{-k/3}$ has size at least $2^{k/3}$. This gives rise to non-uniform security.

Traditionally, a *non-uniformly secure one-way function* has the property that every polynomial sized family of circuits has negligible success probability in inverting (however, in the case of circuits the security requirement can also be formulated in an non-asymptotic way). We remark that the function itself should still be computable by a polynomial time algorithm.

Non-uniform reductions

Similar to the uniform model, one often wants to construct stronger primitives from seemingly weaker ones. Clearly, if the weaker primitive f has non-uniform security, it is desirable that the constructed primitive g has non-uniform security as well.

However, if a uniform reduction is given, as in Theorem 5.6, then this is automatically implied. The reason is simply that Theorem 5.6 does not assume that A_g is a Turing machine, any function (in particular one computed by a small circuit) which inverts g can also be used. In this case Theorem 5.6 (together with Theorem 5.4) guarantees that it is possible to obtain a small circuit which inverts f.

On the other hand, a black-box security proof tailored for the non-uniform model may not be applicable in the uniform model; it is possible that only the existence of a circuit is shown but not *how* to find it for every k. An example of this will be given in Section 6.1. Because of this, security proofs for the uniform model are preferable.

6. Hard-Core Sets

Let P_{XY} be an arbitrary probability distribution over $\{0,1\} \times \mathcal{Y}$, and assume that for all functions g from $\mathcal{Y}$ to $\{0,1\}$

$$\Pr[g(Y)=X] \leq 1 - \frac{\delta}{2} \tag{6.1}$$

holds. This implies (see Lemma 2.2) that there exists a conditional probability distribution $\mathsf{P}_{B|XY}$ over $\{0,1\} \times \{0,1\} \times \mathcal{Y}$ for which $\Pr[B=0] = \delta$ and

$$\Pr[g'(Y)=X|B=0] = \frac{1}{2} \tag{6.2}$$

for all functions g'.

The *non-uniform hard-core lemma* we prove in this section shows that this implication also holds for functions with small circuit complexity. More concretely, assume that (6.1) only holds for all functions g which can be computed by a circuit of size at most s. Then, there exists a distribution $\mathsf{P}_{B|XY}$ with $\Pr[B=0] = \delta$ such that

$$\Pr[g'(Y)=|B=0] \approx \frac{1}{2} \tag{6.3}$$

for all functions g' with can be computed by circuits of a size which is slightly smaller than s.

This non-uniform hard-core lemma can only be applied if functions g with small *circuit complexity* are considered, but not if bounds on the maximal run time of algorithms are given. For this case, we also prove a *uniform* version of the hard-core lemma, which has a slightly different formalization, but achieves the same in applications.

In the computational setting it is usual to model the process of drawing random variables X and Y with functions P and f, i.e., $X = P(W)$ and $Y = f(W)$, where W is a uniform random bit string. The hard-core lemma then states there is a large subset $\mathcal{S}$ — a hard-core set — of the possible randomness such that if w is drawn from $\mathcal{S}$, then $P(w)$ becomes very hard to predict given $f(w)$.

Overview of this chapter

The chapter is divided into two sections. We first give a proof of the *non-uniform* hard-core lemma in Section 6.1. Section 6.2 contains the *uniform* version of the hard-core lemma and it's proof. The non-uniform version is easier to understand than the uniform version, which is why we give both lemmas (the uniform lemma implies the non-uniform one with slightly weaker parameters).

Related work

The first hard-core lemma proven by Impagliazzo [Imp95] was exclusively for the non-uniform setting, and the size of the hard set was only half as big as it is in our version (but, as Impagliazzo notes, the lemma can be applied repeatedly in order to get arbitrarily close to the our set size). In [Imp95] two proofs of the lemma were given: a constructive one (we will change it and then use it in the proof of the uniform hard-core lemma), and a non-constructive one due to Nisan (we will slightly change this proof to get twice the set size in the non-uniform version).

In [KS03], Klivans and Servedio give a connection of hard-core sets to boosting algorithms in computational learning theory (i.e., any algorithm used in a proof of the hard-core lemma is a boosting algorithm, and any boosting algorithm which has an additional smoothness property can be used to prove the hard-core lemma). Boosting algorithms are usually uniform, which makes the existence of a uniform hard-core lemma less surprising (in fact, [KS03] motivated us to prove a uniform hard-core lemma).

Previously to our work, Trevisan proved a variant of the hard-core lemma for the uniform setting [Tre03]. The main difference between the two versions is that Trevisan does not assume that the predicate P is efficiently computable. Consequently he arrives at a weaker conclusion.[1] In general, the simple guideline is that our version should be applied if P (and f) can be computed efficiently, otherwise Trevisan's version can be used.

[1] Described in the words of Theorem 6.9, in Trevisan's lemma algorithm B has only a small probability to produce a circuit which performs well (while in our version it does so with probability almost 1). The net effect is that his lemma can only be applied as long as the relative size of the hard set and the advantage of algorithms on the hard set are relatively large (usually, larger than about $\frac{1}{\log(k)}$), while in our case these quantities only need to be noticeable.

Contributions of this thesis

The contributions of this chapter are Theorems 6.1 and 6.9. Theorem 6.1 gives a non-uniform hard-core lemma. Similar lemmas were known previously, but our variant is the first with a tight set size. Theorem 6.9, the uniform hard-core lemma gives a variant applicable in the uniform setting as well; previously no such lemma was known. These results were previously published in [Hol05], the description here is more detailed.

6.1. The Non-Uniform Case

We first consider the simpler case, where we have computational hardness for non-uniform circuits, as the hard-core lemma is more intuitive in this case.

We start in Section 6.1.1 by giving a formal version of the hard-core lemma. Section 6.1.2 gives an example of how the lemma is used in applications, it can be skipped it if no such example is desired. Then we prove the hard-core lemma in two steps. First, in Section 6.1.3 we show that every mildly hard predicate has a hard-core "measure" (a measure is a fuzzy set which can contain some elements more than others — see Definition 6.3). This is the main part of the proof. Then, in Section 6.1.4 we show that a predicate which has a hard-core measure also has a hard-core set.

6.1.1. The Non-Uniform Hard-Core Lemma

We now give the exact statement of the non-uniform hard-core lemma.

Theorem 6.1 (Non-uniform hard-core lemma — set version). *Let functions $f : \{0,1\}^k \to \{0,1\}^\ell$ and $P : \{0,1\}^k \to \{0,1\}$, and constants $\gamma \in (0,1)$, $\delta \in (0,1)$, and $s' \leq 2^k \frac{\delta^4}{60}$ be given. If all circuits C' of size s' satisfy*

$$\Pr_{W \leftarrow \{0,1\}^k}\left[C'\big(f(W)\big) = P(W)\right] \leq 1 - \frac{\delta}{2} + \frac{\gamma\delta}{8}, \tag{6.4}$$

then there exists a set $\mathcal{S} \subseteq \{0,1\}^k$ with size $|\mathcal{S}| \geq \delta 2^k$ such that all circuits C of size at most $s = \frac{\gamma^2}{40k} s'$ satisfy

$$\Pr_{W \leftarrow \mathcal{S}}[C(f(W)) = P(W)] < \frac{1+\gamma}{2}. \tag{6.5}$$

The upper bound $2^k \frac{\delta^4}{60}$ on the size of the circuits considered is not a restriction in most applications, since all predicates with ℓ bits input can be computed by circuits of size $\frac{2^\ell}{\ell}(1+o(1))$.[2] The term $\frac{\gamma\delta}{8}$ in (6.4) makes the theorem a bit (but not significantly) stronger than what one would expect.

Alternatively we could formulate this result using the notation introduced in Section 2.1: if all circuits C' of size s' satisfy

$$\mathop{\mathrm{Adv}}_{W\leftarrow\{0,1\}^k}^{C'}(P(W)|f(W)) \leq 1-\delta+\frac{\gamma\delta}{4},$$

then all circuits C of size s satisfy

$$\mathop{\mathrm{Adv}}_{W\leftarrow\mathcal{S}}^{C}(P(W)|f(W)) < \gamma$$

for an appropriate set $\mathcal{S}$. We believe the notation in the theorem is more intuitive.

It is interesting to note that this result is tight in the set size $\delta 2^k$ (it may not be tight in the circuit size s', but this is secondary). Assume that there is a set of size $\delta 2^n$ for which no circuit of size s does better than a random guess in finding $P(w)$ from $f(w)$. Then, no circuit of the same size can predict $P(w)$ from $f(w)$ with probability larger than $1-\frac{\delta}{2}$ overall: any circuit contradicting the latter would also contradict the first statement.

As mentioned before, Theorem 6.1 is similar to the information theoretic Lemma 2.2, which says that if a distribution P_{XY} over $\{0,1\}\times\mathcal{Y}$ has the property that no function predicts X from Y with probability larger that $1-\frac{\delta}{2}$, then there is an event with probability δ conditioned on which no function predicts X from Y with probability exceeding $\frac{1}{2}$.

6.1.2. An Application

Before we go to the proof of Theorem 6.1 we provide an application of it. For this, assume that a function $f:\{0,1\}^\ell\to\{0,1\}^k$ and a predicate $P:\{0,1\}^\ell\to\{0,1\}$ are given for which it is mildly hard to predict $P(w)$ from $f(w)$. We define $f^{(n)}:\{0,1\}^{nk}\to\{0,1\}^{n\ell}$ as the concatenation of $f(w_0)$ to $f(w_{n-1})$:

$$f^{(n)}(w_0,\ldots,w_{n-1}) := f(w_0)\|\ldots\|f(w_{n-1}),$$

[2] In case this might be a problem it might still be possible to use Theorem 6.4, where no such upper bound is needed.

and $P^{(\oplus n)} : \{0,1\}^{nk} \to \{0,1\}$ as the XOR of $P(w_0)$ to $P(w_{n-1})$:

$$P^{(\oplus n)}(w_0, \ldots, w_{n-1}) := P(w_0) \oplus \cdots \oplus P(w_{n-1}).$$

The following theorem is the computational analog to Lemma 2.3 and commonly known as "Yao's XOR Lemma". It states that it is harder to predict $P^{(\oplus n)}(w_0, \ldots, w_{n-1})$ from $f^{(n)}(w_0, \ldots, w_{n-1})$ than it is to predict $P(w)$ from $f(w)$. It appeared first implicitly in [Yao82], the first proof is by Levin [Lev87]. For an overview see [GNW95].

Theorem 6.2. *Let $f : \{0,1\}^k \to \{0,1\}^\ell$, $P : \{0,1\}^k \to \{0,1\}$, $\gamma \in (0,1)$, $\beta \in (0,1)$ and $s' \leq 2^k \frac{(1-\beta)^4}{60}$ be given. If all circuits C' of size s' satisfy*

$$\Pr_{W \leftarrow \{0,1\}^k}[C'(f(W) = P(W))] \leq \frac{1+\beta}{2}, \tag{6.6}$$

then, all circuits C of size at most $s = \frac{\gamma^2}{40k} s'$ satisfy

$$\Pr_{W^n \leftarrow \{0,1\}^{nk}}[C(f^{(n)}(W^n)) = P^{(\oplus n)}(W^n)] < \frac{1+\beta^n+\gamma}{2}. \tag{6.7}$$

Proof. Using Theorem 6.1 for $\delta := 1 - \beta$ we see that there exists a set $\mathcal{S}$ with $|\mathcal{S}| \geq \delta 2^k$ such that all circuits C of size at most s satisfy

$$\Pr_{W \leftarrow \mathcal{S}}[C(f(W)) = P(W)] < \frac{1+\gamma}{2}. \tag{6.8}$$

We assume that a circuit $\widetilde{C}$ of size at most s is given which does not satisfy (6.7), i.e.,

$$\Pr_{W^n \leftarrow \{0,1\}^{nk}}[\widetilde{C}(f^{(n)}(W^n)) = P^{(\oplus n)}(W^n)] \geq \frac{1+\beta^n+\gamma}{2}.$$

From $\widetilde{C}$ and the given set $\mathcal{S}$ we will, using some randomness, construct a circuit C which contradicts (6.8) on average over the randomness used in our construction. This is enough to give a contradiction, and thus it proves the theorem.

For this, we first choose n pairs $(w_i, P(w_i))$ for $0 \leq i < n$ and check which of the w_i are elements of $\mathcal{S}$. If there is no coordinate i for which $w_i \in \mathcal{S}$, then our construction chooses randomly either the circuit which outputs the constant 0 or the circuit which outputs the constant 1. In this case, the expectation of $\Pr[C(f(W)) = P(W)]$ is $\frac{1}{2}$.

If there is at least one i with $w_i \in \mathcal{S}$ we construct C as follows: first, we pick a position j such that $w_j \in \mathcal{S}$ uniformly at random. Then, on input w, circuit C runs the given circuit $\widetilde{C}$ with input

$$(f(w_0), \ldots, f(w_{j-1}), f(w), f(w_{j+1}), \ldots, f(w_{n-1})).$$

If this yields bit b, circuit C returns the bit

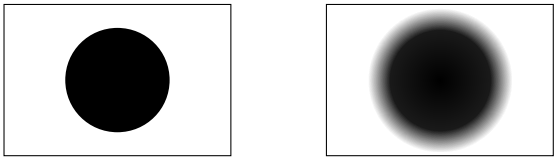
$$b \oplus P(w_0) \oplus \cdots \oplus P(w_{j-1}) \oplus P(w_{j+1}) \oplus \cdots \oplus P(w_{n-1})$$

(note that C can be constructed from $\widetilde{C}$ without increasing the size).

To see why this yields the claimed advantage, let $\mathcal{E}$ be the event that in the construction above $w_i \in \mathcal{S}$ holds for at least one i. Then we get (the expected values in the following are over the random choices done to generate the circuit C, and the set $\overline{\mathcal{S}}$ is defined as $\overline{\mathcal{S}} := \{0,1\}^k \setminus \mathcal{S}$):

$$\begin{aligned}
&\mathrm{E}\Big[\Pr_{W \leftarrow \mathcal{S}}[C(f(W)) = P(W)]\Big] \\
&\quad = \Pr[\mathcal{E}]\,\mathrm{E}\Big[\Pr_{W \leftarrow \mathcal{S}}[C(f(W)) = P(W)] \,\Big|\, \mathcal{E}\Big] \\
&\qquad\quad + \Pr[\overline{\mathcal{E}}]\,\mathrm{E}\Big[\Pr_{W \leftarrow \mathcal{S}}[C(f(W)) = P(W)] \,\Big|\, \overline{\mathcal{E}}\Big] \\
&\quad = \Pr[\mathcal{E}]\,\mathrm{E}\Big[\Pr_{W \leftarrow \mathcal{S}}[C(f(W)) = P(W)] \,\Big|\, \mathcal{E}\Big] + \frac{\Pr[\overline{\mathcal{E}}]}{2} \\
&\quad \overset{(1)}{=} \Pr[\mathcal{E}] \Pr_{W^n \leftarrow \{0,1\}^{nk} \setminus \overline{\mathcal{S}}^n}[\widetilde{C}(f(W)) = P(W)] + \frac{\Pr[\overline{\mathcal{E}}]}{2} \\
&\quad \geq \Pr[\mathcal{E}] \Pr_{W^n \leftarrow \{0,1\}^{nk} \setminus \overline{\mathcal{S}}^n}[\widetilde{C}(f(W)) = P(W)] \\
&\qquad\quad + \Pr[\overline{\mathcal{E}}]\Big(\Pr_{W^n \leftarrow \overline{\mathcal{S}}^n}[\widetilde{C}(f(W)) = P(W)] - 1\Big) + \frac{\Pr[\overline{\mathcal{E}}]}{2} \\
&\quad = \Pr_{W^n \leftarrow \{0,1\}^{nk}}[\widetilde{C}(f(W)) = P(W)] - \frac{\Pr[\overline{\mathcal{E}}]}{2} \\
&\quad \geq \frac{1+\gamma}{2}.
\end{aligned}$$

Equality (1) is easiest seen as follows: instead of choosing w_i for each coordinate, first choose an appropriately biased bit which signalizes whether $w_i \in \mathcal{S}$, and then choose w_i either uniformly from $\mathcal{S}$ or $\overline{\mathcal{S}}$. For one coordinate where w_i is chosen from $\mathcal{S}$ we then use the input, which does not change the distribution.



(a) A set of relative size $\frac{1}{4}$ (b) A measure of density $\frac{1}{4}$

Figure 6.1.: Sets and Measures

From the above, we see that a circuit C which contradicts (6.8) must exist, which finishes the proof. □

6.1.3. Hard-Core Measures

The proof of the hard-core lemma is by contradiction: if for every set $\mathcal{S}$ of size at least $\delta 2^k$ there exists a circuit which contradicts (6.5), then we can combine these circuits to get one which contradicts (6.4). This process will not be done directly on sets, but rather on measures.

Definition 6.3 (Measure). *A measure is a function* $\mathcal{M} : \{0,1\}^k \to [0,1]$. *The size of a measure is* $|\mathcal{M}| := \sum_{w \in \{0,1\}^k} \mathcal{M}(w)$. *The density of a measure* M *is* $\mu(\mathcal{M}) := |\mathcal{M}| \, 2^{-k}$.

Measures should be thought of as fuzzy sets (see Figure 6.1): an element w with $\mathcal{M}(w) = 1$ is in the measure, an element w with $\mathcal{M}(w) = 0$ is not in the measure, and an element w with $\mathcal{M}(w) = \frac{1}{2}$ is "half" in the measure. A measure induces the probability distribution $\mathsf{P}_{\mathcal{M}}(w) := \frac{\mathcal{M}(w)}{|\mathcal{M}|}$, which we often associate implicitly with the measure.[3] Further we will choose a set $\mathcal{S}$ randomly according to the measure (we write $\mathcal{S} \Leftarrow \mathcal{M}$ for this): every element $w \in \{0,1\}$ is in $\mathcal{S}$ independently of the other elements with probability $\mathcal{M}(w)$. Clearly the expectation of the size of the set is just the size of the measure: $\mathrm{E}_{\mathcal{S} \Leftarrow \mathcal{M}}\left[|\mathcal{S}|\right] = |\mathcal{M}|$ (we also use the letter $\mathcal{S}$ to denote the random variable over sets).

The following version of the hard-core lemma states that for a mildly hard predicate we can find a measure $\mathcal{M}$ on which the predicate is hard. We state the contrapositive version.

[3] The connection between measures and distributions over $\{0,1\}^k$ is as follows: the min-entropy of the distribution induced by a measure of density δ is at least $k - \log(\frac{1}{\delta})$, and for any distribution with min-entropy $k - \log(\frac{1}{\delta})$ there exists a measure of density δ which induces it.

In applications, one can usually use the measure version instead of the set version (in fact, we have never seen an example where this is not the case). But since it is simpler to think in terms of sets than in terms of measures we also included the set version.

Theorem 6.4 (Non-uniform hard-core lemma — measure version). *Let $f : \{0,1\}^k \to \{0,1\}^\ell$, $P : \{0,1\}^k \to \{0,1\}$, $\gamma \in (0,1)$, and $\delta \in (0,1)$ be given. Let $\mathfrak{C}$ be a set of functions such that for every measure $\mathcal{M} : \{0,1\}^k \to [0,1]$ with density $\mu(\mathcal{M}) \geq \delta$ there exists a function $C_\mathcal{M} : \{0,1\}^\ell \to \{0,1\}$ in $\mathfrak{C}$ such that*

$$\Pr_{W \leftarrow \mathcal{M}}[C_\mathcal{M}(f(W)) = P(W)] \geq \frac{1+\gamma}{2}. \tag{6.9}$$

Then there exists an oracle circuit $D^\mathfrak{C}$ with $\frac{16k}{\gamma^2}$ oracle gates and $11 \cdot \frac{16k}{\gamma^2}$ non-oracle gates such that

$$\Pr_{W \leftarrow \{0,1\}^k}\left[D^\mathfrak{C}\big(f(W)\big) = P(W)\right] > 1 - \frac{\delta}{2} + \frac{\delta\gamma}{4}. \tag{6.10}$$

The proof of Theorem 6.4 is in two steps. Lemma 6.5 (originally due to Nisan, see [Imp95]) assumes that a collection $\mathfrak{C}$ as in the theorem exists, and shows that in this case a small collection $\mathfrak{C}' \subseteq \mathfrak{C}$ exists, such that a function chosen uniformly from $\mathfrak{C}'$ has slightly higher probability than $\frac{1}{2}$ in predicting $P(w)$ correctly from $f(w)$ for *every* set $\mathcal{S}$ of size $\delta 2^k$ (note the change in quantifiers: the collection $\mathfrak{C}'$ is the same for every set). Lemma 6.6 shows that such a collection $\mathfrak{C}'$ is sufficient to obtain an oracle circuit satisfying equation (6.10).

Lemma 6.5. *Let $f : \{0,1\}^k \to \{0,1\}^\ell$, $P : \{0,1\}^k \to \{0,1\}$, $\gamma \in (0,1)$, and $\delta \in (0,1)$ be given. Let $\mathfrak{C}$ be a set of functions such that for every measure $\mathcal{M} : \{0,1\}^k \to [0,1]$ with density $\mu(\mathcal{M}) \geq \delta$ there exists a function $C_\mathcal{M} : \{0,1\}^\ell \to \{0,1\}$ in $\mathfrak{C}$ such that*

$$\Pr_{W \leftarrow \mathcal{M}}[C_\mathcal{M}(f(W)) = P(W)] \geq \frac{1+\gamma}{2}. \tag{6.11}$$

Then there exists a collection $\mathfrak{C}' \subseteq \mathfrak{C}$ with $|\mathfrak{C}'| \leq \frac{16k}{\gamma^2}$ such that for every set $\mathcal{S}$ of size $|\mathcal{S}| \geq \delta 2^k$

$$\Pr_{C \leftarrow \mathfrak{C}', W \leftarrow \mathcal{S}}[C(f(W)) = P(W)] > \frac{1+\frac{\gamma}{2}}{2}. \tag{6.12}$$

Proof. Consider the following zero-sum game of two players Alice and Bob: Alice chooses a function $C \in \mathfrak{C}$ and simultaneously Bob chooses a set $\mathcal{S} \subseteq \{0,1\}^k$ with $|\mathcal{S}| \geq \delta 2^k$. The payoff for Alice is

$$\Pr_{W \leftarrow \mathcal{S}}[C(f(W)) = P(W)].$$

A randomized strategy for Bob is thus a distribution on sets of size at least $\delta 2^k$, and corresponds to a measure $\mathcal{M}$ on $\{0,1\}^k$ with $\mu(\mathcal{M}) \geq \delta$. For any such strategy, the assumption of the lemma implies that Alice has a strategy to obtain a value of at least $\frac{1+\gamma}{2}$. According to von Neumann's min-max Theorem [vN28] this means that there exists a strategy (i.e., a distribution on functions) for Alice, such that for no strategy of Bob the payoff is lower than $\frac{1+\gamma}{2}$. Let $\mathfrak{C}'$ be this distribution on the functions. Thus, for every set S with $|S| \geq \delta 2^k$, we get

$$\Pr_{C \leftarrow \mathfrak{C}', W \leftarrow \mathcal{S}}[C(f(W)) = P(W)] \geq \frac{1+\gamma}{2}. \tag{6.13}$$

Fix now $w \in \{0,1\}^k$. If $\mathfrak{C}''$ is obtained by sampling $\frac{16k}{\gamma^2}$ functions independently from the distribution $\mathfrak{C}'$, then

$$\left| \Pr_{C \leftarrow \mathfrak{C}'}[C(f(w)) = P(w)] - \Pr_{C \leftarrow \mathfrak{C}''}[C(f(w)) = P(w)] \right| < \frac{\gamma}{4} \tag{6.14}$$

with probability $1 - 2e^{-k}$, according to Proposition 2.13. Thus, assuming $k \geq 2$ and using the union bound, the probability that (6.14) holds *for all* $w \in \{0,1\}^k$ is at least $1 - 2^k \cdot 2 \cdot e^{-k} > 0$, which implies that there exists a collection $\mathfrak{C}''$ which satisfies (6.14) for all w. Together with (6.13) this implies that (6.12) is satisfied for this collection. □

The key observation to improve over [Imp95] in the set size is given in the following lemma, which states that to do so, a collection of functions which does well on average for every set is sufficient. The proof uses a trick very similar as one used by Levin in order to give a tight proof the XOR-Lemma (see [GNW95]), namely it does a randomized decision instead of taking the majority (which is a more usual but inferior strategy).

Lemma 6.6. *Let $f : \{0,1\}^k \to \{0,1\}^\ell$, $P : \{0,1\}^k \to \{0,1\}$, $\delta \in (0,1)$ and $\gamma \in (0,1)$ be given. Let $\mathfrak{C}$ a collection of functions such that for every $\mathcal{S} \subseteq \{0,1\}^k$ of size $|\mathcal{S}| \geq \delta 2^k$*

$$\Pr_{\substack{C \leftarrow \mathfrak{C} \\ W \leftarrow \mathcal{S}}}[C(f(W)) = P(W)] > \frac{1+\gamma}{2}. \tag{6.15}$$

Then there is an oracle circuit $D^{\mathfrak{C}}$ with $|\mathfrak{C}|$ oracle gates and $11|\mathfrak{C}|$ usual gates such that

$$\Pr_{W \leftarrow \{0,1\}^k}[D(f(W)) = P(W)] \geq 1 - \frac{\delta}{2} + \frac{\gamma\delta}{2}. \tag{6.16}$$

The main idea of the proof is as follows: consider the set $\mathcal{S}$ on which the collection $\mathfrak{C}$ performs worst. On input x, our circuit first invokes all circuits in $\mathfrak{C}$ and then does a decision depending on the number of positive answers. This decision is always correct on elements outside of $\mathcal{S}$, and is not worse on elements from $\mathcal{S}$ than what a randomly chosen circuit from $\mathfrak{C}$ would achieve.

Proof. Let

$$\alpha_{\text{corr}}(w) := 2 \Pr_{C \leftarrow \mathfrak{C}}[C(f(w)) = P(w)] - 1$$

be the expected advantage of a function from $\mathfrak{C}$ on w. Analogous, let

$$\alpha_1(w) := 2 \Pr_{C \leftarrow \mathfrak{C}}[C(f(w)) = 1] - 1.$$

Consider a subset $\mathcal{S} \subseteq \{0,1\}^k$ of size $|\mathcal{S}| \geq \delta 2^k$ for which the sum $\sum_{w \in \mathcal{S}} \alpha_{\text{corr}}(w)$ is minimal, and let $\varphi > 0$ be the maximum of $\alpha_{\text{corr}}(w)$ for $w \in \mathcal{S}$.

We first describe a randomized circuit which satisfies (6.16) (i.e., a circuit which has an additional input $R \in \mathcal{R}$ which is chosen at random from some appropriate set $\mathcal{R}$; the circuit can do randomized decisions in that way). On input $f(w)$, circuit $D^{\mathfrak{C}}$ first uses oracle gates to evaluate all functions in the collection $\mathfrak{C}$ and finds $\alpha_1(w)$. It then outputs 1 with probability

$$\Pr_{R \leftarrow \mathcal{R}}[D(f(w), R) = 1] = \begin{cases} 0 & \text{if } \alpha_1(w) \leq -\varphi, \\ \frac{1}{2} + \frac{\alpha_1(w)}{2\varphi} & \text{if } -\varphi < \alpha_1(w) < \varphi, \\ 1 & \text{if } \varphi \leq \alpha_1(w). \end{cases}$$

The probability that $D(f(w), R)$ equals $P(w)$ is then $\frac{1}{2} + \frac{\alpha_{\text{corr}}(w)}{2\varphi}$ truncated at 0 and 1. Therefore for $w \notin \mathcal{S}$, the circuit will always be correct.

On the other hand, since $|\mathcal{S}| \geq \delta 2^k$, (6.15) implies

$$\Pr_{C \leftarrow \mathfrak{C}, W \leftarrow \mathcal{S}}[C(f(W)) = P(W)] > \frac{1}{2} + \frac{\gamma}{2},$$

and thus $\mathrm{E}_{W \leftarrow \mathcal{S}}[\alpha_{\mathrm{corr}}(W)] > \gamma$. For a fixed $w \in \mathcal{S}$ we obtain

$$\Pr_{R \leftarrow \mathcal{R}}[D(f(w), R) = P(w)] = \max\Big(0, \frac{1}{2} + \frac{\alpha_{\mathrm{corr}}(w)}{2\varphi}\Big) \geq \frac{1}{2} + \frac{\alpha_{\mathrm{corr}}(w)}{2\varphi}$$

and thus $\Pr_{W \leftarrow \mathcal{S}, R \leftarrow \mathcal{R}}[D(f(W), R) = P(W)] \geq \frac{1+\gamma}{2}$. In total we obtain

$$\begin{aligned}\Pr_{\substack{W \leftarrow \{0,1\}^k \\ R \leftarrow \mathcal{R}}}[D(f(W), R) = P(W)] &= \delta \Pr_{\substack{W \leftarrow \mathcal{S} \\ R \leftarrow \mathcal{R}}}[D(f(W), R) = P(W)] + (1 - \delta) \\ &\geq 1 - \frac{\delta}{2} + \frac{\delta\gamma}{2}.\end{aligned}$$

Now, fix the randomness used to the value on which the circuit performs best. It is easy to see that in this case we only need to compute a threshold function (i.e., check if more than a fixed number of outputs equals one). This can be done with $11|\mathfrak{C}|$ gates (see [Weg87, Section 3.4]). □

We can use Lemmas 6.5 and 6.6 to proof Theorem 6.4.

Proof (of Theorem 6.4). We know that for every measure $\mathcal{M}$ which has density $\mu(\mathcal{M}) \geq \delta$ there exists a function $C_{\mathcal{M}} \in \mathfrak{C}$

$$\Pr_{W \leftarrow \mathcal{M}}[C_{\mathcal{M}}(f(W)) = P(W)] \geq \frac{1+\gamma}{2}.$$

Lemma 6.5 then implies that there exists a collection $\mathfrak{C}' \subseteq \mathfrak{C}$ of size at most $|\mathfrak{C}'| \leq 16\,k\gamma^{-2}$ with

$$\Pr_{\substack{C \leftarrow \mathfrak{C}' \\ W \leftarrow \mathcal{S}}}[C(f(W)) = P(W)] > \frac{1 + \frac{\gamma}{2}}{2}.$$

Lemma 6.6 states that we can combine these circuits to obtain one circuit $D^{\mathfrak{C}'}$ with $16\,k\gamma^{-2}$ oracle gates and $11 \cdot 16\,k\gamma^{-2}$ usual gates such that

$$\Pr_{W \leftarrow \{0,1\}^k}[D^{\mathfrak{C}'}(f(W)) = P(W)] > 1 - \frac{\delta}{2} + \frac{\gamma\delta}{4}.$$

Since $\mathfrak{C}' \subseteq \mathfrak{C}$ we get the theorem. □

6.1.4. From Measures to Sets

The reason why the hard-core lemma for measures implies the hard-core lemma for sets is simple: if we choose a set according to a measure $\mathcal{M}$ (recall that this means that every element $w \in \{0,1\}^k$ is in the set independently of the others with probability $\mathcal{M}(w)$) then no circuit which is not too large will distinguish the set from the measure. Thus, if the measure is a hard core, a set chosen according to it will also be a hard core. For this, we only need the fact that there are not so many functions computed by circuits of size $2^k \frac{\gamma^2\delta^4}{64k}$.

The following lemma is used for this. It will be reused in Section 6.2, and, because of this, it is slightly stronger than what we need here (it shows that for almost all sets all small circuits behave almost the same; here we would only need that there exists a set for which all small circuits behave almost the same).

Lemma 6.7. *Let $f : \{0,1\}^k \to \{0,1\}^\ell$, $P : \{0,1\}^k \to \{0,1\}$, $\gamma \in (0, \frac{1}{2})$, and $\delta \in (0,1)$ be given. Let further $\mathcal{M} : \{0,1\}^k \to [0,1]$ be a measure with density $\mu(\mathcal{M}) \geq \delta$. The probability that for a random set $\mathcal{S}$ chosen according to $\mathcal{M}$ there exists a circuit C with $\mathrm{Size}(C) \leq 2^k \frac{\gamma^2\delta^4}{64k}$ satisfying*

$$\left| \Pr_{W \leftarrow \mathcal{M}}\left[C(f(W)) = P(W)\right] - \Pr_{W \leftarrow \mathcal{S}}\left[C(f(W)) = P(W)\right] \right| \geq \gamma \qquad (6.17)$$

is less than $2^{-2^k\gamma^2\delta^4/64}$.

Proof. First, the Hoeffding bound (Proposition 2.13), shows that

$$\left(1 - \frac{\gamma\delta}{4}\right)\delta 2^k \leq |\mathcal{S}| \leq \left(1 + \frac{\gamma\delta}{4}\right)\delta 2^k \qquad (6.18)$$

with probability at least $1 - 2\exp(-2^k\gamma^2\delta^4/16)$ (since $|\mathcal{S}|$ is the sum of 2^k independent random variables with range $[0,1]$, namely the indicator variables for $w \in \mathcal{S}$).

Fix any function $g : \{0,1\}^\ell \to \{0,1\}$, and assume w.l.o.g. that $\mathcal{T} := \{w \in \{0,1\}^k \mid g(f(w)) = P(w)\}$ satisfies $\mathrm{E}[|\mathcal{S} \cap \mathcal{T}|] \geq \delta 2^k/2$ (otherwise apply the following argument to the negation of the output of g). The probability that

$$\left(1 - \frac{\gamma\delta}{4}\right)\mathrm{E}\left[|\mathcal{S} \cap \mathcal{T}|\right] \leq |\mathcal{S} \cap \mathcal{T}| \leq \left(1 + \frac{\gamma\delta}{4}\right)\mathrm{E}\left[|\mathcal{S} \cap \mathcal{T}|\right] \qquad (6.19)$$

holds is at least $1 - 2\exp(-2^k\gamma^2\delta^4/32)$ (by the same Hoeffding bound as before). If both (6.18) and (6.19) is satisfied, a straightforward calculation shows that for this function g:

$$\begin{aligned}\Big|\Pr_{W\leftarrow\mathcal{M}}\big[g(f(W)) = P(W)\big] - \Pr_{W\leftarrow\mathcal{S}}\big[g(f(W)) = P(W)\big]\Big| \\ = \Big|\frac{\mathrm{E}[|\mathcal{S}\cap\mathcal{T}|]}{\mathrm{E}[|\mathcal{S}|]} - \frac{|\mathcal{S}\cap\mathcal{T}|}{|\mathcal{S}|}\Big| \leq \gamma. \qquad (6.20)\end{aligned}$$

The probability that (6.20) holds is thus at least $1 - 4\exp(-2^k\gamma^2\delta^4/32)$.

According to Lemma 5.2, for fixed size s, there are less than $(45s)^s$ functions computed by circuits of size s, and thus for maximum size $s := 2^k\frac{\gamma^2\delta^4}{64k}$ there are less than (assuming $k > 2$)

$$\Big(2^k\frac{\gamma^2\delta^4}{k}\Big)^{2^k\frac{\gamma^2\delta^4}{64k}} = 2^{2^k\frac{\gamma^2\delta^4}{64}}\underbrace{\Big(\frac{\gamma^2\delta^4}{k}\Big)^{2^k\frac{\gamma^2\delta^4}{64k}}}_{<\frac{1}{4}} < \frac{1}{4}2^{2^k\frac{\gamma^2\delta^4}{64}}$$

functions computed. Using the union bound, this implies that the probability that a circuit of size $2^k\frac{\gamma^2\delta^2}{64k}$ which contradicts (6.17) exists is bounded by $2^{-2^k\gamma^2\delta^2/64}$. □

We can now show Theorem 6.1.

Proof (of Theorem 6.1). Let $\gamma' := \frac{2\gamma}{3}$ and $\delta' := \delta(1 + \frac{\gamma}{12})$. Assume for a contradiction that for every set $\mathcal{S}$ of size $\delta 2^k$ we have a circuit C of size s contradicting (6.5), i.e.,

$$\Pr_{W\leftarrow\mathcal{S}}[C(f(W)) = P(W)] \geq \frac{1+\gamma}{2}. \qquad (6.21)$$

We show that for any measure $\mathcal{M}$ of density $\mu(M) \geq \delta'$ there is a circuit C of size s such that

$$\Pr_{W\leftarrow\mathcal{M}}[C(f(W)) = P(W)] \geq \frac{1+\gamma'}{2}. \qquad (6.22)$$

For this, let $\mathcal{M}$ be such a measure, and let the set $\mathcal{S}$ be chosen according to $\mathcal{M}$. Since $s < 2^k\frac{\gamma^2\delta^4}{2400k}$, Lemma 6.7 implies that with positive probability we get a set which has size at least $\delta 2^k$ and on which no circuit of size s

differs more than $\frac{\gamma}{6}$ in the probability of guessing $P(w)$ from $f(w)$, which implies (6.22).

We now use Theorem 6.4 with the set $\mathfrak{C}$ of functions computed by circuits of size at most s. This gives an oracle circuit $D^{\mathfrak{C}}$ which satisfies

$$\begin{aligned}\Pr_{W \leftarrow \{0,1\}^k}\left[D^{\mathfrak{C}}(f(W)) = P(W)\right] &\geq 1 - \frac{\delta'}{2} + \frac{\delta'\gamma'}{4} \\ &\geq 1 - \frac{\delta}{2} - \frac{\delta\gamma}{24} + \frac{\delta\gamma}{6} \\ &= 1 - \frac{\delta}{2} + \frac{\delta\gamma}{8}.\end{aligned}$$

We can now replace all the oracle gates in D with the respective circuits. This gives a contradicting (non-oracle) circuit of size at most $\frac{16k}{(\gamma')^2}s + 11\frac{16k}{(\gamma')^2} = \frac{36k}{\gamma^2}s + 11\frac{36k}{\gamma^2} < \frac{40k}{\gamma^2}s$ (we silently used $s \geq 99$). □

6.2. The Uniform Case

Theorems 6.1 and 6.4 are only applicable in the non-uniform setting, i.e., where the hardness of predicting $P(w)$ given $f(w)$ is for non-uniform circuits. It is not immediately clear how to translate these theorems into the uniform setting (where one algorithm is used for all k). For example, the following naive idea does not seem to work: argue that for any pair (f, P) which is mildly hard there exists an infinite sequence of sets $\mathcal{S}_1, \mathcal{S}_2, \ldots$ such that it is very hard to predict $P(w)$ from $f(w)$ if w is chosen from the set $\mathcal{S}_k$ for the respective problem size k. In fact, it is both unclear how to prove this statement and how to use it in applications.

In this section, we will develop a uniform version of the hard-core lemma. First, in Section 6.2.1, we give the hard-core lemma for the uniform setting. In Section 6.2.2 we describe the basic algorithm which gives a measure version of the hard-core lemma (this is the main work). In Section 6.2.3 we use the measure version to prove the set version of the lemma.

6.2.1. The Uniform Hard-Core Lemma

The uniform hard-core lemma states that if the family (f, P) of functions (now defined for every $k \in \mathbb{N}$) is mildly hard for every efficient algorithm, then no efficient algorithm can perform even slightly good on every large enough subset $\mathcal{S}$ of inputs, *even* if it can access the characteristic

function[4] $\chi_{\mathcal{S}}$ of $\mathcal{S}$ (but is restricted to query χ_S independently of the input[5]).

For the following theorem, recall that $\mathfrak{R}_A$ is the randomness which algorithm A uses, even though we omit it as argument to A. Analogously, $\mathfrak{R}_B$ is the randomness used by algorithm B.

Theorem 6.8 (Uniform hard-core lemma — set version). *Let the functions $f:\{0,1\}^k \to \{0,1\}^\ell$, $P:\{0,1\}^k \to \{0,1\}$, $\delta:\mathbb{N} \to (0,1)$, and $\gamma:\mathbb{N} \to (0,1)$, computable in time* $\mathrm{poly}(k)$ *be given, such that γ and δ are noticeable.*

Assume that there is no polynomial time algorithm B such that

$$\Pr_{\substack{W\leftarrow\{0,1\}^k \\ \mathfrak{R}_B}}[B(f(W)) = P(W)] \geq 1 - \frac{\delta}{2} + \frac{\gamma^2\delta^5}{8192} \tag{6.23}$$

for infinitely many k. Then there is no polynomial time oracle algorithm $A^{(\cdot)}$ such that for infinitely many k the following holds: for any set $\mathcal{S} \subseteq \{0,1\}^k$ with $|\mathcal{S}| \geq \delta 2^k$,

$$\Pr_{\substack{W\leftarrow\mathcal{S} \\ \mathfrak{R}_A}}[A^{\chi_{\mathcal{S}}}(f(W)) = P(W)] \geq \frac{1+\gamma}{2}, \tag{6.24}$$

and the queries of A to $\chi_{\mathcal{S}}$ are computed independently of the input $f(w)$.

We give a quick example how this theorem is used, again based on Yao's XOR-Lemma. In the non-uniform proof of it (Theorem 6.2) we used the hard-core lemma as follows: assume a circuit $\widetilde{C}$ contradicts the conclusion that $P^{(\oplus n)}(w^n)$ is very hard-to predict from $f^{(n)}(w^n)$. Then, for any large enough set $\mathcal{S}$ we used the circuit $\widetilde{C}$ to get a circuit C which satisfied

$$\Pr_{W\leftarrow\mathcal{S}}[C(f(W)) = P(W)] \geq \frac{1+\gamma}{2}.$$

This was done as follows: on input $f(w)$ first n random samples w_i were chosen. Then, $\widetilde{C}$ was called with input $f(w_0), \ldots, f(w_{n-1})$, but one of the entries with $w_i \in \mathcal{S}$ was replaced with the input $f(w)$ to C (by doing it that way we ensured that the distribution with which we called $\widetilde{C}$ is not changed). This way of constructing C from $\widetilde{C}$ can be implemented

[4]The characteristic function $\chi_{\mathcal{S}}$ of a set $\mathcal{S}$ is defined as $\chi_{\mathcal{S}}(w) := 1$ if $w \in \mathcal{S}$ and $\chi_{\mathcal{S}}(w) := 0$ otherwise.

[5]There is some subtlety here: we require that the queries are computed *before even looking* at the input. This is stronger than requiring that the queries are *distributed* independently of the input.

by an algorithm, as long as it is possible to decide whether a randomly chosen element of $\{0,1\}^k$ is in $\mathcal{S}$, i.e., if $\chi_{\mathcal{S}}$ is given as oracle. Thus, if an algorithm which has high advantage in predicting the XOR is given, it is easy to obtain an algorithm contradicting (6.24) (note that the queries to $\chi_{\mathcal{S}}$ will be independent of the input w, as required by Theorem 6.8).[6]

6.2.2. The Basic Algorithm

Again we will first prove a measure version of our theorem (this means now that algorithm A has oracle access to a measure[7] instead of the characteristic function of a set).

We assume here that A does not get an input w distributed according to the measure $\mathcal{M}$; instead A has oracle access to the measure $\mathcal{M}$ to *produce* a circuit C which performs well on $\mathcal{M}$ (here C does not have access to $\mathcal{M}$ anymore). This is basically equivalent to the requirement in Theorem 6.8 that the queries of A to the oracle must be computed independently of the input (we explain this in more detail in the proof of Theorem 6.8).

We formulate the contrapositive lemma (i.e., we describe the properties of the algorithm we use in the lemma). Note that the algorithm B we describe in the proof will have to use algorithm A. Thus, B will to supply A with measures; these measures in turn are dependent of the circuits returned by previous calls to A.

Theorem 6.9 (Uniform hard-core lemma — measure version). *Let the functions $f : \{0,1\}^k \to \{0,1\}^\ell$, $P : \{0,1\}^k \to \{0,1\}$, $\delta : \mathbb{N} \to (0,1)$ and $\gamma : \mathbb{N} \to (0,1)$, computable in time $\mathrm{poly}(k)$ be given.*

There is an oracle algorithm $B^{(\cdot)}$ such that:

- *If, for every measure $\mathcal{M}$ with $\mu(\mathcal{M}) \geq \delta$, $A^{\mathcal{M}}$ returns a circuit $C_{\mathcal{M}}$ satisfying*

$$\Pr_{W \leftarrow \mathcal{M}}[C_{\mathcal{M}}(f(W)) = P(W)] \geq \frac{1+\gamma}{2}$$

then, with probability at least $1 - 2^{-k}$ (over the randomness $\mathfrak{R}_B$ of B), B^A returns a circuit C' satisfying

$$\Pr_{W \leftarrow \{0,1\}^k}[C'(f(W)) = P(W)] \geq 1 - \frac{\delta}{2} + \frac{\gamma^2\delta^5}{2048}. \tag{6.25}$$

[6] A formal version of the uniform XOR-Lemma can be derived from Theorem 7.3.

[7] Consistent with the notation introduced in Section 5.1.1, we expect that the oracle returns $\mathcal{M}(w)$ encoded appropriately in $\{0,1\}^*$; for example as pair (u,v) denoting the rational number u/v.

- *B^A does $\mathcal{O}(\gamma^{-2}\delta^{-3})$ calls to A.*
- *B^A evaluates f and P at most $\mathcal{O}(k\gamma^{-4}\delta^{-7})$ times.*
- *B^A does $\mathcal{O}(k\gamma^{-6}\delta^{-10})$ simulations of circuits returned by A.*
- *Additionally, for every call which A does to $\mathcal{M}$, $\mathcal{O}(\gamma^{-2}\delta^{-3})$ simulations of circuits returned by A in previous calls are done, and f and P are evaluated once.*
- *Besides the simulations, B runs in time* $\mathrm{poly}(\gamma^{-1}, \delta^{-1}, k)$.

Algorithm B starts with an empty collection[8] $\mathfrak{C}$ of circuits, and adds circuits one by one to $\mathfrak{C}$. In every step, the collection $\mathfrak{C}$ is used to define a measure $\mathcal{M}$ with $\mu(\mathcal{M}) \geq \delta$. The measure is then used with A to obtain another circuit, which is then added to the collection. This is repeated until for the collection either the majority of the circuits answers correctly on slightly more than fraction $1 - \frac{\delta}{2}$, or else for every set $\mathcal{S}$ of size $\delta 2^k$, a random circuit of $\mathfrak{C}$ has slightly larger probability than $1/2$ of being correct on $\mathcal{S}$ (a similar condition as for Lemma 6.6). We then show that in both cases we can obtain a circuit satisfying (6.25). A graphical overview of the process is given in Figure 6.2.

The idealized algorithm

We first describe an idealized version of the algorithm B. The idealized version assumes that some characteristics of a given collection $\mathfrak{C}$ of circuits (for example the density of the measure $\mathcal{M}_{\mathfrak{C},s}$ defined by C, see below) can be estimated up to some error margin, but the probability of a larger error is zero. We will later show (Claim 1) that we can estimate these quantities while the probability that we make a larger mistake is at most 2^{-2k}.

For the collection $\mathfrak{C}$ of circuits let

$$N_{\mathfrak{C}}(w) := \big|\{C \in \mathfrak{C} \mid C(f(w)) = P(w)\}\big| - \big|\{C \in \mathfrak{C} \mid C(f(w)) \neq P(w)\}\big|. \tag{6.26}$$

The measure $\mathcal{M}_{\mathfrak{C},s}(w)$ used to request the next circuit depends on $N_{\mathfrak{C}}$ and additionally on a number s (which is initially 0 but will be increased

[8] Formally, $\mathfrak{C}$ should be a multiset, because we may want to insert the same circuit more then once.

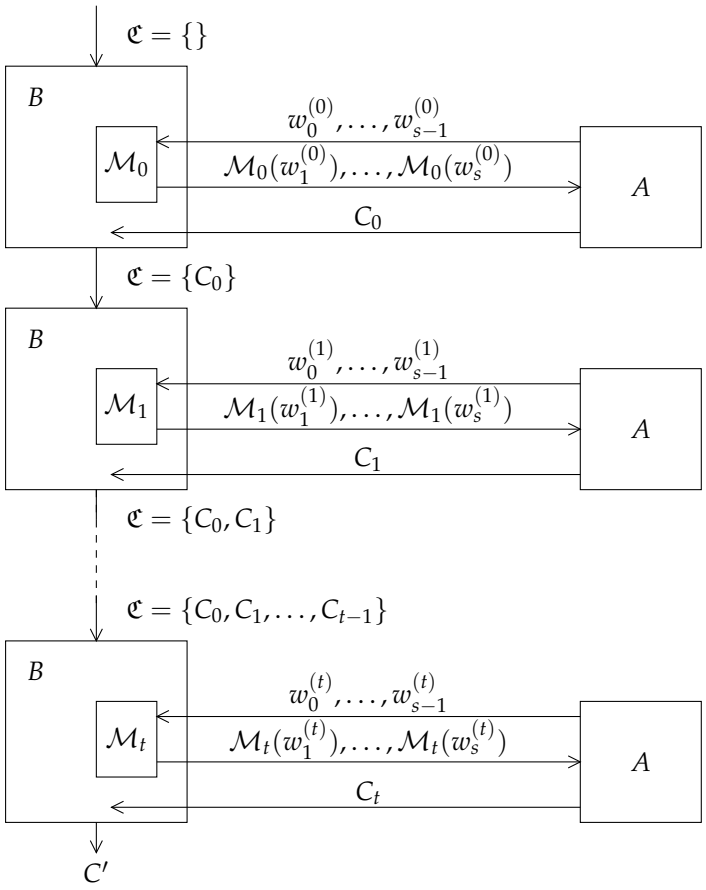


Figure 6.2.: Uniform hard-core lemma: an algorithm A which produces good circuits for every measure $\mathcal{M}$ can be used to get a circuit which is good overall. The algorithm B uses A several $(t+1)$ times, always using different measures $\mathcal{M}_i$. In the end, the circuits C_i are combined into a single circuit C'.

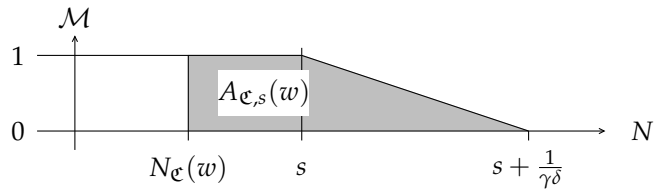


Figure 6.3.: The advantage $N_{\mathfrak{C}}(w)$, measure $\mathcal{M}_{\mathfrak{C},s}(w)$, area $A_{\mathfrak{C},s}(w)$ for one fixed w.

while the collection is growing). It is defined as

$$\mathcal{M}_{\mathfrak{C},s}(w) := \begin{cases} 1 & \text{if } N_{\mathfrak{C}}(w) \leq s, \\ 1 - (N_{\mathfrak{C}}(w) - s)\gamma\delta & \text{if } s < N_{\mathfrak{C}}(w) < s + \frac{1}{\gamma\delta}, \\ 0 & \text{if } N_{\mathfrak{C}}(w) \geq s + \frac{1}{\gamma\delta} \end{cases} \tag{6.27}$$

(cf. Figure 6.3). We note that given s, a collection $\mathfrak{C}$ of circuits, and an element $w \in \{0,1\}^k$ we can compute $\mathcal{M}_{\mathfrak{C},s}(w)$ if we simulate all circuits from $\mathfrak{C}$ on input w and evaluate $f(w)$ and $P(w)$.

In order to prove that our algorithm will stop we consider the area under the curve in Figure 6.3, starting from $N_{\mathfrak{C}}(w)$. Formally, $A_{\mathfrak{C},s}(w)$ is defined as

$$A_{\mathfrak{C},s}(w) := \begin{cases} s - N_{\mathfrak{C}(w)} + \frac{1}{2\gamma\delta} & \text{if } N_{\mathfrak{C}}(w) \leq s, \\ 0 & \text{if } N_{\mathfrak{C}}(w) \geq s + \frac{1}{\gamma\delta}, \\ \frac{\mathcal{M}_{\mathfrak{C},s}(w)}{2}\left(s + \frac{1}{\gamma\delta} - N_{\mathfrak{C}}(w)\right) & \text{otherwise.} \end{cases} \tag{6.28}$$

The total area is also important, and it is thus natural to define

$$A(\mathfrak{C}, s) := \frac{1}{2^k} \sum_{w \in \{0,1\}^k} A_{\mathfrak{C},s}(w).$$

The idealized version of the algorithm is shown in Figure 6.1. We use the following notation: The skip-statement does nothing. The semantics of the if statement as in

```
1  procedure GoodEnough(Collection 𝔈):
2    p := min_{S:|S|≥δ2^k} Pr_{W←S,C←𝔈}[C(f(W)) = P(W)]
3    r := (1/2^k) |{w | N_𝔈(w) ≤ 0}|
4    if p ≥ 1/2 + γδ²/32 ∨ r ≤ 7δ/16 → return true
5     [] p ≤ 1/2 + γδ²/16 ∧ r ≥ 3δ/8 → return false
6    fi
7  end GoodEnough.
8
9  procedure HardCore:
10   s := 0, 𝔈 := ∅
11   while not GoodEnough(𝔈) do
12        if μ(M_{𝔈,s}) ≤ δ(1 + γδ/16) → s := s + 1
13         [] μ(M_{𝔈,s}) ≥ δ → skip
14        fi
15        𝔈 := 𝔈 ∪ {C_M}, where C_M satisfies
16                Pr_{W←M_{𝔈,s}}[C_M(f(W)) = P(W)] > (1+γ)/2.
17   od
18   return 𝔈
19 end HardCore.
```

Listing 6.1: Algorithm for the proof of Theorem 6.9. The statement "skip" does nothing. In an if-statement, any line may be executed for which the guard evaluates to true.

if $C_1 \to S_1$
$[\!]$ $C_2 \to S_2$
fi

for conditions C_1 and C_2, and statements S_1 and S_2 is that one condition which holds is chosen in an arbitrary way, and the corresponding statement is executed. It is important that we do not make any assumption which statement is executed if both conditions hold.

The algorithm, given completely in Figure 6.1, is very simple: it adds the circuit which performs well on $\mathcal{M}_{\mathfrak{C},s}$ to the collection $\mathfrak{C}$ as long as the measure has density at least δ. If the density is too small, i.e., if $\mu(\mathcal{M}_{\mathfrak{C},s}) \leq \delta$, then s is increased before obtaining the circuit. This is repeated until the resulting collection is good enough to prove Theorem 6.9.

The if-statements in the idealized algorithm require sampling, and thus it is not possible to give an efficient implementation of the algorithm in Figure 6.1 without making mistakes with very small probability. We will give a bound on the probability of these mistakes.

We show the correctness of the algorithm in several steps. First, we show that there exists an efficient randomized implementation for the loop. Then we show that in the idealized version the loop terminates after at most $4\gamma^{-2}\delta^{-3}$ iterations. Finally, we prove that a collection as returned by the idealized version is sufficient to prove Theorem 6.9.

Efficient implementation of one loop

To implement the algorithm in Figure 6.1, some knowledge about the quantities $\mu(\mathcal{M}_{\mathfrak{C},s})$, p (line 2), and r (line 3) is required. Also we need to make sure that we can obtain circuits for the measures as required in line 15.

Claim 1. *There exists an implementation of the conditional statements in lines 4 and 12, which does $\mathcal{O}(\frac{k}{\gamma^2\delta^4})$ simulations of all the circuits in $\mathfrak{C}$, $\mathcal{O}(\frac{k}{\gamma^2\delta^4})$ calls to both f and P, and has error probability at most 2^{-2k} for every call.*

Proof. First, consider line 12: we can implement it if we can estimate $\mu(\mathcal{M}_{\mathfrak{C},s})$ up to an error of $\frac{\gamma\delta^2}{32}$. For this, we choose $\mathcal{O}(\frac{k}{\gamma^2\delta^4})$ times a uniform $w \in \{0,1\}^k$ and compute $\mathcal{M}_{\mathfrak{C},s}(w)$: we query $f(w)$ and simulate all the circuits in $\mathfrak{C}$ on input $f(w)$, then compare the result with $P(w)$. Since $\mu(\mathcal{M}_{\mathfrak{C},s})$ is just the expectation of this, the Hoeffding bound (Proposition 2.13) states that the probability of an error can be made smaller than 2^{-2k}.

Now consider line 4. In order to estimate r within a margin of $\frac{\delta}{32}$ it is sufficient to do $\mathcal{O}(\frac{k}{\delta^2})$ simulations of circuits in $\mathfrak{C}$ at random points and that many oracle queries to f and P, again the probability that we are off too much can be bounded by 2^{-2k}.

It is a bit more tricky to see that we can efficiently estimate p such that with probability 2^{-2k} the estimate is not more than $\gamma\delta^2/64$ off. For this, let $\preceq$ be a total order on $\{0,1\}^k$ satisfying

$$N_{\mathfrak{C}}(w) < N_{\mathfrak{C}}(w') \Rightarrow w \preceq w',$$

and such that $\preceq$ is simple to compute given w and w'. We set w^δ to be the element at the δ-quantile according to this order, and define the set $\mathcal{T} := \{w \in \{0,1\}^k \mid w \preceq w^\delta\}$. With this notation we want to estimate $p = \Pr_{W \leftarrow \mathcal{T}}[C(f(W)) = P(W)]$. After drawing elements $w_0, \ldots, w_{s-1}$, computing $f(w_i)$, $P(w_i)$ and simulating $C(f(w_i))$ for all circuits $C \in \mathfrak{C}$ and all i we can compute $N_{\mathfrak{C}}(w_i)$ for all w_i. Considering only the sampled elements w_i, let $\overline{w}^\delta$ be the element at the δ-quantile according to $\preceq$ and define $\overline{\mathcal{T}} := \{w \in \{0,1\}^k \mid w \preceq \overline{w}^\delta\}$. Now, $\overline{p} := \Pr_{W \leftarrow \overline{\mathcal{T}}}[C(f(W)) = P(W)]$ is a good estimate for p (note that $\overline{w}^\delta$ must be close to w^δ; this can be shown by applying the Hoeffding bound once for a slightly smaller set than $\mathcal{T}$ and once for a slightly bigger one). Further, the average of the respective probabilities of the sampled elements smaller than $\overline{w}^\delta$ gives a good estimate of $\overline{p}$ because of the Hoeffding bound. In total, we see that $s \in \mathcal{O}(\frac{k}{\gamma^2\delta^4})$ is sufficient. ◊

We also need to show that in the idealized version of the algorithm the measure satisfies $\mu(\mathcal{M}_{\mathfrak{C},s}) \geq \delta$ whenever a circuit for this measure is requested.

Claim 2. *In every iteration $\mu(\mathcal{M}_{\mathfrak{C},s}) \geq \delta$ holds after line 14.*

Proof. The claim holds in the first round. Furthermore, the claim can only be wrong if s is increased in line 12. In this case the measure cannot have decreased for any w when compared with the iteration before. This implies that the total density is at least as big as one iteration earlier, which implies the claim by induction. ◊

Termination

We now show that the algorithm stops after at most $4\gamma^{-2}\delta^{-3}$ iterations. For this, we show that $A(\mathfrak{C}, s) - \delta s$ decreases by at least $\gamma\delta^2/8$ in every

iteration, and that the algorithm must stop if this expression gets smaller than 0. Note that initially $A(\emptyset, 0) = \frac{1}{2\gamma\delta}$.

First, we show that adding a circuit to $\mathfrak{C}$ (while leaving s constant) decreases $A(\mathfrak{C}, s)$ by at least $\frac{\gamma\delta}{2}$.

Claim 3. *If $C_{\mathcal{M}}$ satisfies $\Pr_{W \leftarrow \mathcal{M}_{\mathfrak{C},s}}[C_{\mathcal{M}}(f(W)) = P(W)] \geq \frac{1+\gamma}{2}$ as well as $\mu(\mathcal{M}_{\mathfrak{C},s}) \geq \delta$, then $A(\mathfrak{C} \cup \{C_{\mathcal{M}}\}, s) \leq A(\mathfrak{C}, s) - \frac{\gamma\delta}{2}$.*

Proof. Let $\mathcal{S}^+ := \{w \mid C_{\mathcal{M}}(g(w)) = f(w)\}$ (i.e., the w for which $C_{\mathcal{M}}$ is correct), $\mathcal{S}^- := \{w \mid C_{\mathcal{M}}(g(w)) \neq f(w)\}$, and $\mathfrak{C}' := \mathfrak{C} \cup \{C_{\mathcal{M}}\}$.

Consider a fixed w. If $w \in S^+$, then $A_{\mathfrak{C}',s}(w) \leq A_{\mathfrak{C},s}(w) - \mathcal{M}_{\mathfrak{C},s}(w) + \frac{\gamma\delta}{2}$ (note that $N_{\mathfrak{C}'}(w) = N_{\mathfrak{C}}(w) + 1$, and with Figure 6.3 it is easy to see that the area decreases by at least $\mathcal{M}_{\mathfrak{C},s}(w)$ minus the small triangle which is cut off in case $\mathcal{M}_{\mathfrak{C},s}(w)$ does not stay constant when adding $C_{\mathcal{M}}$). Also, if $w \in S^-$ then $A_{\mathfrak{C}',s}(w) \leq A_{\mathfrak{C},s}(w) + \mathcal{M}_{\mathfrak{C},s}(w) + \frac{\gamma\delta}{2}$, using a similar argument. Thus,

$$A(\mathfrak{C}', s) \leq A(\mathfrak{C}, s) + \frac{\gamma\delta}{2} + \frac{1}{2^k}\Big(\sum_{w \in \mathcal{S}^-} \mathcal{M}_{\mathfrak{C},s}(w) - \sum_{w \in \mathcal{S}^+} \mathcal{M}_{\mathfrak{C},s}(w)\Big).$$

Now, $\Pr_{W \leftarrow \mathcal{M}_{\mathfrak{C},s}}[C_{\mathcal{M}}(f(W)) = P(W)] \geq \frac{1+\gamma}{2}$ is equivalent to

$$\sum_{w \in \mathcal{S}^+} \mathcal{M}_{\mathfrak{C},s}(w) - \sum_{w \in \mathcal{S}^-} \mathcal{M}_{\mathfrak{C},s}(w) \geq \gamma \sum_{w} \mathcal{M}_{\mathfrak{C},s}(w),$$

and using $\mu(\mathcal{M}_{\mathfrak{C},s}) \geq \delta$ we see that

$$A(\mathfrak{C}', s) \leq A(\mathfrak{C}, s) + \frac{\gamma\delta}{2} - \gamma\delta = A(\mathfrak{C}, s) - \frac{\gamma\delta}{2}\,. \qquad \Diamond$$

Of course, if s is increased in line 12, then the area $A(\mathfrak{C}, s)$ will grow. We can give an upper bound on this:

Claim 4. *If s is increased in line 12 then $A(\mathfrak{C}, s+1) \leq A(\mathfrak{C}, s) + \delta + \frac{\gamma\delta}{2} - \frac{\gamma\delta^2}{8}$.*

Proof. First we note that for any w, $A_{\mathfrak{C},s+1}(w) \leq A_{\mathfrak{C},s}(w) + \mathcal{M}_{\mathfrak{C},s}(w) + \gamma\delta/2$, and if $N_{\mathfrak{C}}(w) \leq 0 \leq s$ then $A_{\mathfrak{C},s+1}(w) \leq A_{\mathfrak{C},s}(w) + \mathcal{M}_{\mathfrak{C},s}(w)$. Since the loop would have stopped if $\mathcal{S} := \{w \mid N_{\mathfrak{C}}(w) \leq 0\}$ was smaller than

$(3\delta/8)2^k$, we get

$$\begin{aligned} A(\mathfrak{C}, s+1) &\leq A(\mathfrak{C}, s) + \frac{1}{2^k}\Big(\sum_{w\in\mathcal{S}} \mathcal{M}_{\mathfrak{C},s}(w) + \sum_{w\notin\mathcal{S}} \Big(\mathcal{M}_{\mathfrak{C},s}(w) + \frac{\gamma\delta}{2}\Big)\Big) \\ &\leq A(\mathfrak{C}, s) + \underbrace{\mu(\mathcal{M}_{\mathfrak{C},s})}_{\leq\delta(1+\gamma\delta/16)} + \Big(1 - \frac{3\delta}{8}\Big)\frac{\gamma\delta}{2} \\ &\leq A(\mathfrak{C}, s) + \delta + \frac{\gamma\delta}{2} - \frac{\gamma\delta^2}{8}. \end{aligned}$$ ◊

Claim 5. *In every iteration of the loop, $A(\mathfrak{C}, s) - s\delta$ decreases by at least $\frac{\gamma\delta^2}{8}$.*

Proof. Combine Claim 3 and 4. ◊

Claim 6. *If $A(\mathfrak{C}, s) - s\delta < 0$, then $\mathfrak{C}$ is a collection which satisfies*

$$\Pr_{C\leftarrow\mathfrak{C}, W\leftarrow\mathcal{S}}[C(f(W)) = P(W)] > \frac{1}{2} + \frac{1}{4\gamma\delta|\mathfrak{C}|}$$

for every $\mathcal{S} \subseteq \{0,1\}^k$ of size $|\mathcal{S}| \geq \delta 2^k$.

Proof. Let $\mathcal{H} \subseteq \{0,1\}^k$ be a set of size $\delta 2^k$ for which

$$\Pr_{C\leftarrow\mathfrak{C}, W\leftarrow\mathcal{H}}[C(f(W)) = P(W)]$$

is minimized. Since

$$\Pr_{C\leftarrow\mathfrak{C}, W\leftarrow\mathcal{H}}[C(f(W)) = P(W)] = \frac{1}{2} + \frac{\sum_{w\in\mathcal{H}} N_{\mathfrak{C}}(w)}{2|\mathfrak{C}||\mathcal{H}|}$$

it is enough to show that $\sum_{w\in\mathcal{H}} N_{\mathfrak{C}}(w) > \frac{2^k}{2\gamma}$. We see that $A_{\mathfrak{C},s}(w) \geq \frac{1}{2\gamma\delta} + s - N_{\mathfrak{C}}(w)$ (this is easiest seen as follows: $\frac{1}{2\gamma\delta} + s - N_{\mathfrak{C}}(w)$ can be thought of as $\int_a^b 1\,\mathrm{d}w$ for $a := N_{\mathfrak{C}}(w)$ and $b := s + \frac{1}{2\gamma\delta}$; comparing the corresponding area with Figure 6.3 yields the claim), and this implies

$$\begin{aligned} \sum_{w\in\mathcal{H}} N_{\mathfrak{C}}(w) &\geq \sum_{w\in\mathcal{H}} \frac{1}{2\gamma\delta} + s - A_{\mathfrak{C},s}(w) \\ &\geq \delta 2^k\Big(\frac{1}{2\gamma\delta} + s\Big) - \sum_{w\in\{0,1\}^k} A_{\mathfrak{C},s}(w) \\ &= \frac{2^k}{2\gamma} + \delta 2^k s - 2^k A(\mathfrak{C}, s) > \frac{2^k}{2\gamma}. \end{aligned}$$ ◊

Lemma 6.10. *The loop of algorithm* HardCore *is traversed at most* $4\gamma^{-2}\delta^{-3}$ *times.*

Proof. Initially the collection is empty, and thus $A(\mathfrak{C}, s) = A(\emptyset, 0) = \frac{1}{2\gamma\delta}$. Since in every iteration $A(\mathfrak{C}, s) - s\delta$ decreases by at least $\frac{\gamma\delta^2}{8}$, this means that after at most $4\gamma^{-2}\delta^{-3}$ iterations $A(\mathfrak{C}, s) - s\delta < 0$, in which case Claim 6 implies that

$$\Pr_{C \leftarrow \mathfrak{C}, W \leftarrow \mathcal{S}}[C(f(W)) = P(W)] > \frac{1}{2} + \frac{\gamma\delta^2}{16}$$

(note that $|\mathfrak{C}| \leq 4\gamma^{-2}\delta^{-3}$). Thus, the if statement in line 4 of the algorithm *must* return true (since the guard of line 5 is wrong), and the algorithm terminates. □

The collection yields a circuit

Claim 7. *Let* $\gamma : \mathbb{N} \to (0,1)$, $\delta : \mathbb{N} \to (0,1)$ *be given, and* $\mathfrak{C}$ *be a collection of circuits such that for every set* $\mathcal{S}$ *of size* $|\mathcal{S}| \geq \delta 2^k$

$$\Pr_{W \leftarrow \mathcal{S}, C \leftarrow \mathfrak{C}}[C(f(W)) = P(W)] > \frac{1}{2} + \frac{\gamma\delta^2}{16}.$$

Then there exists a randomized circuit C' *of size* $11|\mathfrak{C}| + \sum_{C \in \mathfrak{C}} \mathrm{Size}(C)$ *for which*

$$\Pr_{W \leftarrow \{0,1\}^k}[C'(f(W)) = P(W)] > 1 - \frac{\delta}{2} + \frac{\gamma^2\delta^5}{2048}$$

with probability $1 - 2^{-k}$. *Furthermore, such a circuit* C' *can be found by an algorithm from* $\mathfrak{C}$ *which performs* $\mathcal{O}(k\gamma^{-2}\delta^{-6})$ *simulations of all circuits from* $\mathfrak{C}$ *and does* $\mathcal{O}(k\gamma^{-2}\delta^{-6})$ *computations of* f *and* P. *The algorithm runs in time polynomial in the total size of the circuits in* $\mathfrak{C}$.

The proof is analogous to the proof of Lemma 6.6, but we need to make sure that we can find C' efficiently.

Proof. Let $\preceq$ be a total order on $\{0,1\}^k$ satisfying

$$N_{\mathfrak{C}}(w) < N_{\mathfrak{C}}(w') \Rightarrow w \preceq w',$$

and we assume that $\preceq$ is simple to compute given w and w'.

First, for $\varphi := \frac{\gamma\delta^2}{32}$, we find a triple $(\tilde{w}, f(\tilde{w}), P(\tilde{w}))$ such that

$$\frac{1}{2^k}\big|\{w \mid w \preceq \tilde{w}\}\big| \in [\delta, \delta(1+\varphi)] \tag{6.29}$$

with probability $1 - 2^{-2k}$. This can be done efficiently as follows: sample $\mathcal{O}(k\varphi^{-2}\delta^{-2})$ many triples $(w, f(w), P(w))$, order them according to $\preceq$, and select the element $\tilde{w}$ at relative position $\delta(1+\varphi/2)$. The Hoeffding bound implies that we can do this such that $\tilde{w}$ satisfies (6.29) with probability $1 - 2^{-2k}$. We compute $N_{\mathfrak{C}}(\tilde{w}) > 0$, and return a circuit C' which, on input $f(w)$ first computes the number

$$N_{\mathfrak{C},1}(w) := \big|\{C \in \mathfrak{C} \mid C(f(w)) = 1\}\big| - \big|\{C \in \mathfrak{C} \mid C(f(w)) = 0\}\big|,$$

and then outputs one with probability

$$\Pr_{\mathfrak{R}_{C'}}[C'(f(w)) = 1] = \begin{cases} 0 & \text{if } N_{\mathfrak{C},1}(w) \le -N_{\mathfrak{C}}(\tilde{w}), \\ \frac{1}{2} + \frac{N_{\mathfrak{C},1}(w)}{2N_{\mathfrak{C}}(\tilde{w})} & \text{if } -N_{\mathfrak{C}}(\tilde{w}) < N_{\mathfrak{C},1}(w) < N_{\mathfrak{C}}(\tilde{w}), \\ 1 & \text{if } N_{\mathfrak{C}}(\tilde{w}) \le N_{\mathfrak{C},1}(w). \end{cases}$$

Note that for a w with $w \not\preceq \tilde{w}$ the circuit is always correct. For a w satisfying $w \preceq \tilde{w}$, we have

$$\Pr_{\mathfrak{R}_{C'}}[C'(f(w)) = P(w)] \ge \frac{1}{2} + \frac{N_{\mathfrak{C}}(w)}{2N_{\mathfrak{C}}(\tilde{w})},$$

and thus

$$\begin{aligned} \Pr_{\substack{W \leftarrow \{w \mid w \preceq \tilde{w}\} \\ \mathfrak{R}_{C'}}}[C'(f(W)) = P(W)] &\ge \frac{1}{2} + \frac{\sum_{w \preceq \tilde{w}} N_{\mathfrak{C}}(w)}{2N_{\mathfrak{C}}(\tilde{w})\ |\{w \mid w \preceq \tilde{w}\}|} \\ &\ge \frac{1}{2} + \frac{\sum_{w \preceq \tilde{w}} N_{\mathfrak{C}}(w)}{2|\mathfrak{C}| \cdot |\{w | w \preceq \tilde{w}\}|} \\ &= \Pr_{C \leftarrow \mathfrak{C}, W \leftarrow \{w | w \preceq \tilde{w}\}}[C(f(W)) = P(w)] \\ &\ge \frac{1+\varphi}{2}. \end{aligned}$$

In total, we obtain

$$\begin{aligned}
&\Pr_{W\leftarrow\{0,1\}^k}[C'(f(W)) = P(W)] \\
&\qquad = \Pr_{W\leftarrow\{0,1\}^k}[W \preceq w']\frac{1+\varphi}{2} + \Pr_{W\leftarrow\{0,1\}^k}[W \not\preceq w'] \cdot 1 \\
&\qquad \geq \delta(1+\varphi)\frac{1+\varphi}{2} + (1-\delta(1+\varphi)) \\
&\qquad > 1 - \frac{\delta}{2} + \frac{\delta\varphi^2}{2}. \qquad \lozenge
\end{aligned}$$

Claim 8. *Let $\mathfrak{C}$ be a collection of circuits that $\frac{1}{2^k}\left|\{w \mid N_{\mathfrak{C}}(w) \leq 0\}\right| \leq \frac{7\delta}{16}$. Then there is a circuit C' of size $11|\mathfrak{C}| + \sum_{C\in\mathfrak{C}} \mathrm{Size}(C)$ for which*

$$\Pr_{W\leftarrow\{0,1\}^k}[C'(f(W)) = P(W)] > 1 - \frac{7\delta}{16}.$$

Furthermore, C' can be found efficiently from $\mathfrak{C}$.

Proof. The majority function applied to the output of all the circuits in the collection satisfies the desired properties. $\lozenge$

Finishing the proof

We can now finish the proof of Theorem 6.9.

Proof (of Theorem 6.9). We use the algorithm in Figure 6.1. For the if statements we use the observations in Claim 1. After running the algorithm we get a single circuit using either Claim 7 or Claim 8.

We check statements of the theorem one by one.

- Claim 2 makes sure that we only call A with measures which satisfy $\mu(\mathcal{M}) \geq \delta$. Further, because the algorithm stops (Lemma 6.10), and because of Claims 7 and 8, we see that B returns a circuit for which (6.25) holds.
- First, Lemma 6.10 states that the loop is traversed at most $4\gamma^{-2}\delta^{-3}$ times. Thus the algorithm does only that many calls to A. For later we note that this also implies $|\mathfrak{C}| \leq 4\gamma^{-2}\delta^{-3}$.
- The number of evaluations of f and P is $\mathcal{O}(k\gamma^{-4}\delta^{-7})$: first, f and P are evaluated $\mathcal{O}(k\gamma^{-2}\delta^{-4})$ times for every if statement (Claim 1), totaling to $\mathcal{O}(k\gamma^{-4}\delta^{-7})$. Further, they are called $\mathcal{O}(k\gamma^{-2}\delta^{-6})$ times to get a circuit from the collection $\mathfrak{C}$ (Claims 7 and 8).

- In every iteration of the loop, algorithm B does $\mathcal{O}(k\gamma^{-4}\delta^{-7})$ simulations of circuits returned by A (Claim 1), totaling to at most $\mathcal{O}(k\gamma^{-6}\delta^{-10})$ simulations. To get a circuit from the collection we do another $\mathcal{O}(k\gamma^{-2}\delta^{-6})$ simulations, which totals to $\mathcal{O}(k\gamma^{-6}\delta^{-10})$ simulations.

- For a call of $\mathcal{M}$ from A we have to simulate $C(f(w))$ for all circuits C in $\mathfrak{C}$; thus at most $\mathcal{O}(\gamma^{-2}\delta^{-3})$ simulations need to be performed. Also one call to f and P is needed.

- Clearly, B runs in time polynomial in γ^{-1}, δ^{-1} and k. □

6.2.3. Measures and Sets

As in the non-uniform case we can get a version of the hard-core lemma which uses sets. Again, the observation is that any algorithm which produces circuits for any set of size $\delta 2^k$ must also work for any measure of density δ. We only consider polynomial time algorithms and noticeable γ and δ here because it's the most usual case. Recall that for a set $\mathcal{S}$ we use $\chi_{\mathcal{S}}$ to denote its characteristic function.

Lemma 6.11. *Let the functions $f : \{0,1\}^k \to \{0,1\}^\ell$, $P : \{0,1\}^k \to \{0,1\}$, $\delta : \mathbb{N} \to (0,1)$ and $\gamma : \mathbb{N} \to (0,1)$, computable in time $\mathrm{poly}(k)$ be given, such that γ and δ are noticeable.*

Let $A^{(\cdot)}$ be a polynomial time algorithm such that for any set $\mathcal{S} \subseteq \{0,1\}^k$ with $|\mathcal{S}| \geq \delta 2^k$, $A^{\chi_{\mathcal{S}}}$ outputs a circuit C satisfying

$$\mathop{\mathrm{E}}_{\mathfrak{R}_A}\left[\Pr_{W \leftarrow \mathcal{S}}[C(f(W)) = P(W)]\right] \geq \frac{1+\gamma}{2}.$$

If $\mathrm{Size}(\mathfrak{C}) \in \mathrm{poly}(k)$, then there exists an algorithm $\overline{A}$ running in time $\mathrm{poly}(k)$ such that for any measure $\mathcal{M}$ with $|\mathcal{M}| \geq \delta$ algorithm $\overline{A}^{\mathcal{M}}$ outputs a circuit such that with probability $1 - 2^{-2k}$ (over $\mathfrak{R}_{\overline{A}}$):

$$\Pr_{W \leftarrow \mathcal{M}}[C(f(W)) = P(W)]] \geq \frac{1+\gamma/2}{2}. \tag{6.30}$$

Proof. Note that given oracle access to $\mathcal{M}(w)$, it is possible to efficiently simulate an oracle $\chi_{\mathcal{S}}$ for a set $\mathcal{S}$ chosen according to $\mathcal{M}$ (the answers must be cached). We thus run algorithm A^{χ_S} for a set $\mathcal{S}$ chosen according to $\mathcal{M}$. Because of Lemma 6.7 and Markov's inequality, the probability

that a circuit C is returned for which

$$q := \Pr_{W \leftarrow \mathcal{M}}[C(f(W)) = P(W)] \geq \frac{1 + \frac{3\gamma}{4}}{2}. \tag{6.31}$$

is noticeable (for infinitely many k). Using the Hoeffding bound it is easy to see that we can now make sure that (6.30) holds by estimating q (we accept if the estimate $\tilde{q}$ satisfies $\tilde{q} > \frac{1+5\gamma/8}{2}$), and such that the probability that we accept if (6.30) does not hold is at most 2^{-2k}. □

Lemma 6.11 and Theorem 6.9 together can be used to prove Theorem 6.8.

Proof (of Theorem 6.8). Assume otherwise, i.e., let A be a polynomial time oracle algorithm, which satisfies

$$\Pr_{\substack{W \leftarrow \mathcal{S} \\ \mathfrak{R}_A}}[A^{\chi_{\mathcal{S}}}(f(W)) = P(W)] \geq \frac{1+\gamma}{2},$$

and for which the queries of A to $\chi_{\mathcal{S}}$ are computed independently of the input $f(w)$. We construct A_1 as follows: first, it chooses the randomness for A and then computes the queries A makes to $\chi_{\mathcal{S}}$ and then outputs a circuit which performs the same computation as A on this randomness (this is only possible if the queries to $\chi_{\mathcal{S}}$ are computed independently of the input $f(w)$). The circuit has expected probability $\frac{1+\gamma}{2}$ of being correct, and thus we can use Lemma 6.11 and get that there exists an efficient algorithm A_2 such that for any measure $\mathcal{M}$ with density at least δ algorithm A_2 outputs a circuit which satisfies

$$\Pr_{W \leftarrow \mathcal{M}}[C(f(W)) = P(W)] \geq \frac{1+\gamma/2}{2}.$$

with probability $1 - 2^{-k}$ over $\mathfrak{R}_{A_2}$. We use this algorithm together with the algorithm B from Theorem 6.9. Since A_2 is efficient it only does a polynomial number of queries to $\mathcal{M}$ for any call, and since γ and δ are noticeable, the whole algorithm will be efficient. Thus, B^{A_2} yields a circuit C which satisfies

$$\Pr_{\substack{W \leftarrow \{0,1\}^k \\ \mathfrak{R}_B}}[C(f(W)) = P(W)] \geq 1 - \frac{\delta}{2} + \frac{\gamma^2\delta^5}{8192}$$

with probability $1 - 2^{-k-1}$. We can then simulate this circuit on the given input and output the result which gives a contradiction. □

7. Strengthening Key Agreement

This final chapter of this thesis connects the results up to now. Assume that a weak bit agreement protocol is given, i.e., Alice and Bob have a protocol where they end up with bits X and Y such that $\Pr[X = Y] \geq \frac{1+\alpha}{2}$. Further, assume that conditioned on the event $X = Y$ no polynomial time algorithm can predict these bits with probability exceeding $\frac{1+\beta_{\mathrm{eq}}}{2}$ from the communication Z and all but finitely many k. We show that if Alice and Bob insert the resulting bits of this protocol into an efficient information theoretic protocol for α-correlated random variables with equality leakage β_{eq}, they obtain a *computationally secure key*.

An analogous lemma holds for one-way key agreement; this case is interesting as it allows us to strengthen key agreement protocols without adding rounds, which is important when we want to strengthen public-key encryption schemes. The key step in the proof of these statements uses the lemma about hard-core sets from the previous chapter.

Overview of this chapter

This chapter is organized as follows: in Section 7.1, we start with a few preparations: first, we give a slight strengthening of Theorem 6.8 (which is for technical reasons — basically it states that a mildly hard predicate P which is only defined on a subset of $\{0,1\}^k$ also has a hard-core set). Second, we show that predicting $P(w)$ from $f(w)$ is essentially equivalent to distinguishing $P(w)$ from a uniform random bit given $f(w)$; this is a well known equivalence and we need it later. In Section 7.2 we use the hard-core lemma to give a general method of extracting pseudorandomness from a mildly hard predicate. Sections 7.3 and 7.4 then apply this general method to our two computational settings. Thus in these sections it is shown how to strengthen both general key agreement protocols and public-key cryptosystems. Both sections also contain theorems which show that for black-box reductions our results are tight (in the second case this holds only in case additional restrictions are placed on the reduction).

Related work

The question whether public-key encryption schemes can be strengthened has first been considered by Dwork, Naor, and Reingold [DNR04]. The construction they present is weaker than our construction in that it starts from a stronger public-key encryption scheme.

A result similar to Theorem 7.3 is given implicitly in [HILL99] (see also [Hås90]), but the proof uses a different technique. Independently of this work, Harnik, Haitner, and Reingold [HHR05] use our hard-core lemma to present another similar theorem. Our version is slightly stronger than these theorems in that it allows the possibility of side information (Theorem 7.3 allows a non-trivial function Leak).

Contributions of this thesis

The result of Section 7.3 is original to this thesis. It was previously published in [Hol05] in a slightly weaker form. The result of Section 7.4 is joint work with Renato Renner, and a slightly weaker form appears in [HR05b]. Theorem 7.3 in Section 7.2 is a novel abstraction and unifies a key transition step from the information theoretic to the computational setting, which is implicitly used in both of the above references.

7.1. Preparations

We first perform two technical preparations. First, a slight strengthening of the hard-core lemma is given. Second, we give a well known lemma which states that predicting a bit is essentially the same as distinguishing it from a uniform random bit.

7.1.1. Strengthening the Hard-Core Lemma

Assume that Alice and Bob have a weak key agreement protocol which produces random variables X and Y as well as communication Z such that $\Pr[X = Y] = \frac{1+\alpha}{2}$ and for all polynomial time algorithms A

$$\Pr_{XYZ,\mathfrak{R}_A}[A(Z) = X | X = Y] \leq \frac{1+\beta_{\text{eq}}}{2}.$$

We would like to use Theorem 6.8 to show that there exists a set of the randomness Alice and Bob use such that on this set it is very hard to

predict the key bit. However, because we have only a guarantee on the hardness of predicting the key in case $X = Y$ there is a minor technical problem: Theorem 6.8 only gives an assertion for predicates which are slightly hard on the whole domain $\{0,1\}^k$.

In this section, we remove this assumption. For this, we assume that we have an additional function $q : \{0,1\}^k \to \{0,1\}$ (in our application q will indicate whether $X = Y$, i.e., whether the output is equal), and we assume that it is slightly hard to predict $P(w)$ given $f(w)$ in case $q(w) = 1$. We show that the usual hard-core lemma implies a version for this case as well. The idea is to extend the predicate $P(w)$ in case $q(w) \neq 1$ such that it can easily be predicted in this case, and then to apply Theorem 6.8.

Lemma 7.1. *Let the functions $f : \{0,1\}^k \to \{0,1\}^\ell$, $P : \{0,1\}^k \to \{0,1\}$, $q : \{0,1\}^k \to \{0,1\}$, $\delta : \mathbb{N} \to (0,1)$, and $\gamma : \mathbb{N} \to (0,1)$, computable in time* poly(k) *be given, such that γ, δ, as well as the function given by $\Pr_{W \leftarrow \{0,1\}^k}[q(W) = 1]$ are noticeable. Let $\mathcal{Q} := \{w \in \{0,1\}^k | q(w) = 1\}$.*

If there is no polynomial time algorithm B such that

$$\Pr_{\substack{W \leftarrow \mathcal{Q} \\ \mathfrak{R}_B}}[B(f(W)) = P(W)] \geq 1 - \frac{\delta}{2} \tag{7.1}$$

for infinitely many k, then there is no polynomial time oracle algorithm $A^{(\cdot)}$ such that for infinitely many k the following holds: for any set $\mathcal{S} \subseteq \mathcal{Q}$ with $|\mathcal{S}| \geq \delta|\mathcal{Q}|$,

$$\Pr_{\substack{W \leftarrow \mathcal{S} \\ \mathfrak{R}_A}}[A^{\chi_{\mathcal{S}}}(f(W)) = P(W)] \geq \frac{1+\gamma}{2}, \tag{7.2}$$

and the queries of A to χ_S are computed independently of the input $f(w)$.

Proof. Define $\overline{P}(w) := q(w) \wedge P(w)$, i.e., $\overline{P}(w) = 1$ if both $P(w) = 1$ and $q(w) = 1$. Further, define $\overline{f}(w) := f(w) \| q(w)$ as the concatenation of $f(w)$ and $q(w)$.

The assumption (7.1) then implies that no polynomial time algorithm $\overline{B}$ satisfies (for $\mu(\mathcal{Q}) := |\mathcal{Q}|\, 2^{-k}$)

$$\begin{aligned}
&\Pr_{W \leftarrow \{0,1\}^k}[\overline{B}(\overline{f}(W)) = \overline{P}(W)] \\
&\qquad \geq \Pr_{W \leftarrow \{0,1\}^k}[q(W) = 0] + \Pr_{W \leftarrow \{0,1\}^k}[q(W) = 1]\Big(1 - \frac{\delta}{2}\Big) \\
&\qquad = 1 - \frac{\delta\mu(\mathcal{Q})}{2}.
\end{aligned}$$

Applying Theorem 6.8 this means that there exists no polynomial time algorithm $\overline{A}^{(\cdot)}$ such that, for any $\mathcal{S} \subseteq \{0,1\}^k$ with $|\mathcal{S}| \geq |\mathcal{Q}|\delta$, $\overline{A}^{\chi_{\mathcal{S}}}$ satisfies

$$\Pr_{W \leftarrow \mathcal{S}}[\overline{A}^{\chi_{\mathcal{S}}}(\overline{f}(W)) = \overline{P}(W)]] \geq \frac{1+\gamma}{2}.$$

and does queries to $\chi_{\mathcal{S}}$ which are independent of $\mathcal{S}$. However, any algorithm $A^{\chi_{\mathcal{S}}}$ which contradicts (7.2) gives such an algorithm $\overline{A}^{(\cdot)}$ (because in case $q(w) = 0$ it is easy to predict $\overline{P}$ from $\overline{f}$). □

7.1.2. Predicting and Distinguishing Single Bits

We will need a simple (well known) lemma which states that if an algorithm which gets $f(w)$ can distinguish $P(w)$ from a uniform bit slightly better than guessing uniformly at random, then it can be used to *predict* $P(w)$ from $f(w)$ slightly better than a uniform random guess.

Lemma 7.2. *Let functions $f : \{0,1\}^k \to \{0,1\}^\ell$, $P : \{0,1\}^k \to \{0,1\}$, and a distribution P_W over $\{0,1\}^k$ be given. There is an oracle algorithm $B^{(\cdot)}$ such that, for any algorithm A, setting*

$$\varepsilon := \Pr_{\substack{W \leftarrow \mathsf{P}_W \\ \mathfrak{R}_A}}[A(f(W), P(W)) = 1] - \Pr_{\substack{W \leftarrow \mathsf{P}_W \\ U \leftarrow \{0,1\}, \mathfrak{R}_A}}[A(f(W), U) = 1], \tag{7.3}$$

algorithm B^A satisfies

$$\Pr_{\substack{W \leftarrow \mathsf{P}_W \\ \mathfrak{R}_B}}[B^A(f(W)) = P(W)] = \frac{1}{2} + \varepsilon,$$

does one oracle call to A, and computes one XOR.

Note that in this lemma ε cannot be larger than $\frac{1}{2}$, since a uniform bit is equal to $P(w)$ with probability $\frac{1}{2}$. This explains why the probability that B^A is correct can be $\frac{1}{2} + \varepsilon$ and not only $\frac{1+\varepsilon}{2}$.

Proof. On input $f(w)$, Algorithm B chooses a bit u uniformly at random and simulates $A(f(w), u)$. Assume that A answers with the bit b. Then B outputs $b \oplus u \oplus 1$.

For $u \in \{0,1\}$ and $w \in \{0,1\}^k$ let

$$p_{w,u} := \Pr_{\mathfrak{R}_A}[A(f(w), u) = 1].$$

With this notation we can rewrite (7.3) as follows:

$$\begin{aligned}\varepsilon &= \mathop{\mathrm{E}}_{W\leftarrow \mathsf{P}_W}[p_{W,P(W)}] - \mathop{\mathrm{E}}_{\substack{W\leftarrow \mathsf{P}_W\\ U\leftarrow\{0,1\}}}[p_{W,U}] \\ &= \mathop{\mathrm{E}}_{W\leftarrow \mathsf{P}_W}\Big[p_{W,P(W)} - \frac{p_{W,P(W)} + p_{W,1-P(W)}}{2}\Big] \\ &= \mathop{\mathrm{E}}_{W\leftarrow \mathsf{P}_W}\Big[\frac{p_{W,P(W)} - p_{W,1-P(W)}}{2}\Big]. \end{aligned} \tag{7.4}$$

The output of Algorithm B is correct in two cases: if $u = P(w)$ and the output of A is 1, or if $u \neq P(w)$ and the output of A is 0. Thus, the probability that the output of B is correct is (where we use (7.4)):

$$\begin{aligned}\Pr_{W\leftarrow \mathsf{P}_W}[B^A(f(W)) = P(W)] &= \mathop{\mathrm{E}}_{W\leftarrow \mathsf{P}_W}\left[\frac{1}{2}\big(p_{W,P(W)} + 1 - p_{W,1-P(W)}\big)\right] \\ &= \mathop{\mathrm{E}}_{W\leftarrow \mathsf{P}_W}\left[\frac{1 + p_{W,P(W)} - p_{W,1-P(W)}}{2}\right] \\ &= \frac{1}{2} + \varepsilon. \end{aligned}$$

□

7.2. Extraction of Pseudorandomness

Assume that functions $f : \{0,1\}^k \to \{0,1\}^\ell$ and $P : \{0,1\}^k \to \{0,1\}$ are given such that every algorithm A running in polynomial time satisfies

$$\Pr_{\substack{W\leftarrow\{0,1\}^k\\ \mathfrak{R}_A}}[A(f(W)) = P(W)] \leq \frac{1+\beta}{2}$$

for a given function $\beta : \mathbb{N} \to [0,1]$ and all but finitely many k. Choose values $w^n = (w_0, \ldots, w_{n-1})$ independently and uniform from $\{0,1\}^k$. If n is large enough, the hard-core lemma suggest that a polynomial time algorithm which has given the concatenation

$$f^{(n)}(w^n) := f(w_0)\| \ldots \|f(w_{n-1})$$

of all $f(w_i)$ can not predict $P(w_i)$ better than a random guess for about $(1-\beta)n$ of the instances. While we cannot tell which instances are hard, we should be able to use an extractor (see Section 2.5) to get roughly $(1-\beta)n$ (minus some entropy loss) uniform looking bits out of

$$P^{(n)}(w^n) := P(w_0)\| \ldots \|P(w_{n-1}),$$

even conditioned on $f^{(n)}(w^n)$.

In fact, let Ext be a function such that for any distribution P_{XZ} with $\mathrm{Adv}^{\max}(X|Z) \leq \beta$ (recall Definition 2.1, page 11) the outcome of $\mathrm{Ext}(X^n)$ is ε-close to uniform with respect to Z^n (recall Definition 2.7, page 15) for some negligible ε. We believe that applying Ext on $P^{(n)}(w^n)$ yields bits which are computationally indistinguishable from uniform, even if a distinguisher also gets $f^{(n)}(w^n)$. The theorem below states that this is indeed the case, but it asserts even more. Namely, assume that additionally a function Leak is given. If, for any distribution as above, the output of Ext is ε-close to uniform with respect to the concatenation of Z^n and $\mathrm{Leak}(X^n)$, then $\mathrm{Ext}(P^{(n)}(w^n))$ yields bits which are pseudorandom even if both $f^{(n)}(w^n)$ and $\mathrm{Leak}(P^{(n)}(w^n))$ are given.

Actually, we need to make the theorem a bit more complicated: we additionally assume the existence of a function $q : \{0,1\}^k \to \{0,1\}$ which signals whether $(f(w), P(w))$ are a *valid* pair (in our application with two-way key agreement, $P(w)$ will be the key bit of the weak scheme; it will be valid if $X = Y$; in the one-message case it will always be valid).

Theorem 7.3. *Let the functions*

$$f : \{0,1\}^k \to \{0,1\}^\ell, \qquad P : \{0,1\}^k \to \{0,1\},$$
$$q : \{0,1\}^k \to \{0,1\}, \qquad \beta : \mathbb{N} \to [0,1],$$

computable in time $\mathrm{poly}(k)$ *be given, and define the set*

$$\mathcal{Q} := \{w \in \{0,1\}^k | q(w) = 1\}.$$

Assume that every polynomial time algorithm satisfies

$$\Pr_{\substack{W \leftarrow \mathcal{Q} \\ \mathfrak{R}_B}}[B(f(W)) = P(W)] \leq \frac{1+\beta}{2} \tag{7.5}$$

for all but finitely many k.

Further, let also functions $n(k)$, $s(k)$,

$$\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^s \to \{0,1\}^t,$$
$$\mathrm{Leak} : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^s \to \{0,1\}^{t'},$$

be given which are evaluable in time $\mathrm{poly}(k)$*, and satisfy the following: for any distribution P_{QXZ} over $\{0,1\} \times \{0,1\} \times \mathcal{Z}$ with*

$$\mathsf{P}_Q(1) = \Pr_{W \leftarrow \{0,1\}^k}[q(W) = 1]$$

and for which $\mathsf{P}_{XZ|Q=1}$ *has leakage at most* β*, the output of* $\mathrm{Ext}(Q^n, X^n, R)$ *is* $\varepsilon(k)$*-close to uniform with respect to* $(Z^n, \mathrm{Leak}(Q^n, X^n, R))$ *(where* R *is a uniform bit string of length* s*).*

Then, no polynomial time algorithm A*, which gets as input* $f^{(n)}(w^n)$ *and* $\mathrm{Leak}(q^{(n)}(w^n), P^{(n)}(w^n), R)$*, distinguishes*

$$\mathrm{Ext}(q^{(n)}(w^n), P^{(n)}(w^n), R)$$

from a uniform random string of length t *with advantage* $\varepsilon + \gamma$*, for any non-negligible function* γ*.*

Proof. We assume a contradicting algorithm A exists, and show that this implies the existence of a polynomial time algorithm B contradicting (7.5). Lemma 7.1 states that for this it is sufficient to give an oracle algorithm algorithm $\overline{A}^{\chi_{\mathcal{S}}}$ which, for any set $\mathcal{S} \subseteq \mathcal{Q}$ with $|\mathcal{S}| \geq (1-\beta)|\mathcal{Q}|$ satisfies

$$\Pr_{\substack{W \leftarrow \mathcal{S} \\ \Re_{\overline{A}}}} [\overline{A}^{\chi_{\mathcal{S}}}(f(W)) = P(W)] \geq \frac{1+\gamma'}{2},$$

for some non-negligible function γ', and calls $\chi_{\mathcal{S}}$ only with queries which are computed independently of the input.

First consider the following random experiment, defined for any fixed $j \in \{0, \ldots, n\}$ and any fixed set $\mathcal{S} \subseteq \mathcal{Q}$: start by choosing independent uniform bit strings $w_i \in \{0,1\}^k$ as well as uniform bits $u_i \in \{0,1\}$ for all $i \in \{0, \ldots, n-1\}$. Then set values y_i, as

$$y_i := \begin{cases} P(w_i) & \text{if } i \geq j \text{ or } w_i \notin \mathcal{S}, \\ u_i & \text{otherwise.} \end{cases} \tag{7.6}$$

Continue by choosing $r_0 \in \{0,1\}^s$ uniformly at random and set

$$\begin{aligned} e_j &:= \mathrm{Ext}(q^{(n)}(w^n), y^n, r_0) \quad \text{and} \\ \ell_j &:= \mathrm{Leak}(q^{(n)}(w^n), y^{(n)}, r_0). \end{aligned} \tag{7.7}$$

Let P_{E_j} be the distribution of e_j, P_{L_j} the distribution of ℓ_j, and P_F be the distribution of $f^{(n)}(w^n)$.

We can see that $\mathsf{P}_{FL_0E_0}$ is one of the distributions for which A is usually called (the one which is not uniform). On the other hand, the information

theoretic requirement on the functions Ext and Leak imply that E_n is ε-close to uniform conditioned on (L_n, F). With these facts we get

$$\Big| \Pr_{\mathfrak{R}_A, E_0, L_0, F}[A(E_0, L_0, F) = 1] - \Pr_{\mathfrak{R}_A, E_n, L_n, F}[A(E_n, L_n, F) = 1] \Big| \geq \gamma. \tag{7.8}$$

This implies

$$\mathop{\mathrm{E}}_{J \leftarrow \{0,\ldots,n-1\}} \Big[\Big| \Pr_{\mathfrak{R}_A, E_J, L_J, F}[A(E_J, L_J, F)=1] - \Pr_{\mathfrak{R}_A, E_{J+1}, L_{J+1}, F}[A(E_{J+1}, L_{J+1}, F)=1] \Big| \Big] \geq \frac{\gamma}{n}. \tag{7.9}$$

Given this we now show how to implement a distinguisher which distinguishes $(f(w), P(w))$ from $(f(w), U)$ with advantage γ/n, if w is chosen uniformly from $\mathcal{S}$ and U a uniform random bit, as long as oracle access to $\chi_{\mathcal{S}}$ is given. On input $(f(w), b)$, the distinguisher first picks $w_i \in \{0,1\}^k$, $u_i \in \{0,1\}$ as well as $j \in \{0, \ldots, n-1\}$ uniformly at random, computes $f(w_i)$, $P(w_i)$, and y_i as in (7.6). If $w_j \in \mathcal{S}$, he replaces $f(w_j)$ with the input $f(w)$, and y_i with b. Then, he evaluates e_j and ℓ_j as in (7.7). If b is a uniform bit, then this process gives random variables distributed according to $\mathsf{P}_{E_{j+1} L_{j+1} F}$, otherwise it gives random variables distributed according to $\mathsf{P}_{E_j L_j F}$. Thus, running A gives a distinguisher which can be used in Lemma 7.2, and from the result we can apply Lemma 7.1 to get the theorem. □

7.3. Key Agreement

Let a computational bit agreement protocol which has only weak security be given. We consider the distribution P_{XYZ} over $\{0,1\} \times \{0,1\} \times \{0,1\}^\ell$ which is generated by this protocol, where x is the key bit of Alice, y is the key bit of Bob, and z is the communication produced.

Theorem 7.4. *Let functions $\alpha : \mathbb{N} \to [0,1]$ and $\beta_{\mathrm{eq}} : \mathbb{N} \to [0,1]$, both efficiently computable, be given. Let a computational bit agreement protocol be given which produces a distribution P_{XYZ}, such that*

$$\Pr_{\mathsf{P}_{XY}}[X=Y] \geq \frac{1+\alpha}{2} \tag{7.10}$$

and for which all polynomial time algorithms A satisfy

$$\Pr_{\mathfrak{R}_A, XYZ}[A(Z)=X|X=Y] \leq \frac{1+\beta_{\text{eq}}}{2} \tag{7.11}$$

for all but finitely many k. Further, let an information theoretic protocol be given which takes α-correlated random variables with equality leakage β_{eq} as as well as k as input, produces a key of length $m(k)$ with soundness and secrecy $1-2^{-k}$ and runs in time $\text{poly}(k)$. Then, using the information theoretic protocol where every instance of the random variables is replaced by an independent outcome of the computational protocol gives a computationally secure key agreement protocol.

Proof. First we note that it is clear that the resulting key has soundness $1-2^{-k}$, since P_{XY} has correlation α.

In order to prove the secrecy, we would like to use Theorem 7.3. For this we describe the computational protocol by the following functions: $f : \{0,1\}^k \to \{0,1\}^\ell$, mapping the concatenation of the randomness of Alice and Bob to the communication, $P_A : \{0,1\}^k \to \{0,1\}$ which maps the randomness to the key bit x of Alice, and $q : \{0,1\}^k \to \{0,1\}$ which is defined as

$$q(w) := \begin{cases} 1 & \text{if } x = y \text{ for randomness } w \\ 0 & \text{otherwise.} \end{cases}$$

The functions Ext and Leak are defined by the information theoretic key agreement protocol; Ext produces the key of Alice while Leak produces the communication of the protocol. The input is given by X^n (i.e., the random variables), $Q^n := X^n \oplus Y^n$, and the joint randomness R which Alice and Bob may use in the protocol.

Now Theorem 7.3 can be applied. Note that all assumptions hold: first, it is easy to check that all functions can be evaluated in time $\text{poly}(k)$. Second, the requirement of the ε-closeness of $\text{Ext}(Q^n, Y^n, R)$ with respect to $(Z^n, \text{Leak}(Q^n, Y^n, R))$ translates directly to the security requirement of the information theoretic protocol. □

From Theorem 4.23 we know for which parameters α and β_{eq} information theoretic key agreement exists, we can get the following corollary.

Corollary 7.5. *Let efficiently computable functions $\alpha(k)$, $\beta_{\text{eq}}(k)$, be given such that*

$$\frac{1-\alpha}{1+\alpha} < 1 - \beta_{\text{eq}}. \tag{7.12}$$

Let $\varphi := \max\left(2, \frac{8}{\log(\frac{(1-\beta_{\text{eq}})(1+\alpha)}{1-\alpha})}\right)$ *and* $\gamma := \frac{1}{\log(1+((1-\alpha)/(1+\alpha))^{\varphi})}$*, and assume that* $\frac{\varphi 2^{4\gamma}}{\alpha} \in \text{poly}(k)$*. If there exists a weak bit agreement protocol which generates a distribution* P_{XYZ} *for which*

$$\Pr_{\mathsf{P}_{XY}}[X = Y] \geq \frac{1+\alpha}{2}$$

and for which all polynomial time algorithms A satisfy

$$\Pr_{\mathfrak{R}_A, W}[A(Z){=}X | X{=}Y] \leq \frac{1+\beta_{\text{eq}}}{2}.$$

for all but finitely many k, then there exists a computationally secure key agreement protocol.

Proof. Combine Theorems 4.23 and 7.4. □

We note that the corollary is tight in the following sense: if α and β_{eq} do not satisfy (7.12), i.e., $\frac{1-\alpha}{1+\alpha} \geq 1-\beta_{\text{eq}}$, then Theorem 4.24 shows that there exists a bit agreement protocol which yields random variables with correlation α and equality leakage β_{eq}. Clearly, showing that such a protocol implies key agreement is equivalent to proving that key agreement exists unconditionally. Since random variables with equality leakage β_{eq} also have the analogous computational hardness, strengthening Corollary 7.5 in that way would imply is impossible.

Theorem 7.6. *Let constants* $\alpha \in [0,1]$ *and* $\beta_{\text{eq}} \in [0,1]$ *be given. Using a black-box reduction, it is possible to base key agreement on a weak bit agreement protocol which generates a distribution* P_{XYZ} *with*

$$\Pr_{\mathsf{P}_{XY}}[X = Y] \geq \frac{1+\alpha}{2}$$

and for which all polynomial time algorithms A satisfy

$$\Pr_{\mathfrak{R}_A, \mathsf{P}_{XYZ}}[A(Z){=}X | X{=}Y] \leq \frac{1+\beta_{\text{eq}}}{2}.$$

for all but finitely many k, if and only if

$$\frac{1-\alpha}{1+\alpha} < 1-\beta_{\text{eq}}.$$

Proof. Directly from Corollary 7.5 and Theorem 4.24. □

7.4. Public-Key Encryption

A public-key encryption scheme is a key agreement protocol which has only two rounds. The name stems from the fact that in such a protocol the first message can be distributed to everybody, i.e., made public. Using this message, anyone can then finish the key agreement protocol by computing the second message and sending it to the initial sender. If this initial sender remembers the randomness used to generate the first message, the reply is sufficient for the two parties to agree on a shared key, which can then be used to encrypt a message. The party which replied usually does so immediately and concatenates the encryption to the second message of the key agreement protocol.

In this setting, we assume that Bob sends the first message.

Strengthening public-key encryption

We now show how to strengthen such a public-key encryption scheme.

Theorem 7.7. *Let functions $\alpha : \mathbb{N} \to [0,1]$ and $\beta : \mathbb{N} \to [0,1]$, both efficiently computable, be given. Let a public-key encryption scheme be given where Bob sends the first message which produces a distribution P_{XYZ} such that*

$$\Pr_{\mathsf{P}_{XY}}[X{=}Y] \geq \frac{1+\alpha}{2},$$

and for which all polynomial time algorithms A satisfy

$$\Pr_{\mathfrak{R}_A, \mathsf{P}_{XZ}}[A(Z){=}X] \leq \frac{1+\beta}{2}.$$

Further, let an information theoretic one-message protocol be given which takes as input k as well as α-correlated random variables with leakage β, produces a key of length $m(k)$ with soundness $1-2^{-k}$ and secrecy $1-2^{-k}$ and runs in time $\mathrm{poly}(k)$. Then, using the information theoretic protocol where every instance of the random variables is replaced by an independent outcome of the public-key encryption scheme gives a computationally secure public-key encryption scheme.

Proof. Again we can apply Theorem 7.3. This is done exactly as in the proof of Theorem 7.4, but in this case we define $q(w) := 1$ for all w.

The fact that we obtain a public-key encryption scheme is immediate, since the number of rounds in the protocol is not increased. □

Corollary 7.8. *Let efficiently computable functions α and β be given, as well as a β-secure public bit encryption scheme with correlation α, such that $\alpha^2 > \beta$. Further, let $\gamma := \max(1, \frac{1}{\log(\alpha^2/\beta)})$. If $\gamma\alpha^{-12\gamma} \in \mathrm{poly}(k)$, then there exists a public key encryption scheme.*

Proof. Combine Theorems 4.14 and 7.7. □

Tightness

We show that our results are tight, as long as one is restricted to a certain class of reductions. In order to argue about this class, we first characterize a public-key encryption scheme by the three functions, one for each round.

In the context of public-key encryption schemes, usually the following naming conventions are used. The method to obtain the first message is called a *key-generation algorithm* G, and the first message is the *public key* pk. The randomness which is used as input to the key-generation algorithm is then the *secret key* sk. The algorithm to generate the second message is called *encryption algorithm* Enc (usually it takes an additional input besides randomness which is then encrypted, i.e., the two tasks of key agreement and encryption are done at the same time; we do not follow this convention however), and the last algorithm is then analogously called *decryption algorithm* Dec.

We assume that these functions have the following form:

- $\mathsf{G}(\mathsf{sk}) : \{0,1\}^\ell \to \{0,1\}^\ell$,
- $\mathsf{Enc}(\mathsf{pk}, \rho) : \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}^{2\ell} \times \{0,1\}$,
- $\mathsf{Dec}(c, \mathsf{sk}) : \{0,1\}^{2\ell} \times \{0,1\}^\ell \to \{0,1\}$

The function Enc (for *encrypt*) takes as input the communication produced by Bob, and the randomness ρ of Alice, it outputs communication c and a key bit X. The function Dec (for *decrypt*) takes as input the communication c from Alice, and the randomness sk from Bob, and outputs a key bit Y.

We prove that our reduction is tight for a class of reductions which we call "restricted black-box reductions".

Definition 7.9 (Restricted Black-box Reductions). *A restricted black-box reduction for an α-correlated public-key encryption scheme with computational leakage β consists consists of three oracle algorithms $\mathsf{G}^{(\cdot)}$, $\mathsf{Enc}^{(\cdot)}$ and $\mathsf{Dec}^{(\cdot)}$,*

which have oracle access to the three functions G', Enc' *and* Dec' *and adhere to the following limitations:*

- G *has access to* G'*. Further, it queries* G' *only on independent, uniformly random chosen inputs. It outputs all the results, and does no further computation.*
- Enc *has access to* Enc'*. It queries* Enc' *with every output by* G' *exactly once, and the results appear in the output of* Enc*. It may additionally do any other computation.*
- Dec *has access to* Dec'*.*

Additionally, there is a black-box security proof of the reduction, i.e., an oracle algorithm which, for any oracle breaking the resulting scheme, breaks the original scheme with advantage exceeding β*, assuming that the original scheme produces random variables with correlation at least* α*.*

The restrictions in Definition 7.9 are very strong, but our method to strengthen weak public-key encryption satisfies them. The restrictions we make could be relaxed slightly at some points and the proof of the following lemma still works. However, we believe that such slight changes provide no new insight.

Lemma 7.10. *Let parameters* $\alpha, \beta \in [0,1]$ *be given. If there is a restricted black-box reduction which constructs a public-key encryption scheme from a weak public-key encryption scheme which satisfies*

$$\Pr_{\mathsf{P}_{XY}}[X = Y] \geq \frac{1+\alpha}{2}$$

and

$$\Pr_{\mathsf{P}_{XZ},\mathfrak{R}_A}[A(Z) = X] < \frac{1+\beta}{2}.$$

Then, there exists an information theoretic one-message key agreement scheme for α*-correlated random variables with leakage* β*.*

Proof. We can assume without loss of generality that the resulting scheme generates a single key bit.

Let oracle algorithms G, Enc and Dec be given which adhere to the limitations of Definition 7.9. From these algorithms we construct an information theoretic one message key agreement protocol which produces a secure key for α-correlated random variables with leakage β. Thus, let a

distribution P_{XYZ} with correlation α and leakage β be given. We can assume that the distribution P_{XYZ} satisfies $\mathsf{P}_X(0) = \mathsf{P}_X(1) = \frac{1}{2}$ (otherwise, Alice can XOR X with a uniform random bit and send the result to Bob to obtain such a distribution).

In a first step, Alice runs G, and whenever the algorithm calls G' on a uniform random input sk_i, Alice simulates G' by returning a uniform random value $\mathsf{pk}_i \in \{0,1\}^\ell$. Then, Alice remembers the pair $(\mathsf{sk}_i, \mathsf{pk}_i)$. In a second step, Alice runs the encryption procedure Enc, where, on every call to the oracle Enc', she obtains a new random variable X_i which she treats as the weak key bit Enc' agreed upon, and then outputs a random communication value $c_i \in \{0,1\}^{2\ell}$. She then stores (c_i, i). Alice continues the simulation until she obtains a message generated by Enc and a secret bit S_A.

Next, Alice sends the message generated by Enc as well as all tuples $(\mathsf{sk}_i, \mathsf{pk}_i, c_i)$ to Bob. These tuples are sent sorted, so Bob knows which entry belongs to which random variable.

Bob can find the S_B by simulating the algorithm Dec. Any query Dec makes to Dec' with a query in the list and the correct secret key will be answered by Y_i. Otherwise, a random bit is returned. It is easy to see that Alice and Bob will obtain agreement on the secret bit with overwhelming probability: otherwise, oracles $(\mathsf{G}'', \mathsf{Enc}'', \mathsf{Dec}'')$ for which the reduction fails can be constructed.

We next show that the protocol is information theoretically secure. We will prove this by contradiction. Thus, assume above protocol is insecure. We see that Eve gets a sample of

$$\Big(Z^n, m, (\mathsf{sk}_1, \mathsf{pk}_1, c_1), \ldots, (\mathsf{sk}_n, \mathsf{pk}_n, c_n)\Big),$$

where m is the communication produced by Enc. As a first observation we see that the secret keys sk_i are independent of the rest, and thus Eve can fill those with her own randomness. Thus, from now on we assume that Eve sees a a sample of

$$\Big(Z^n, m, (\mathsf{pk}_1, c_1), \ldots, (\mathsf{pk}_n, c_n)\Big). \tag{7.13}$$

Now, if the protocol is information theoretically insecure, we can assume that the statistical distance φ of the distribution in case $S_A = 0$ from the distribution in case $S_A = 1$ is non-negligible (otherwise a secure protocol can be constructed). Thus, by trying all possible values for X_i and the randomness used, we can, in polynomial space, find S_A with advantage φ.

Let now G' be a randomly chosen permutation. Further, let Enc' also be a random permutation if restricted to the first 2ℓ bits of the output, and choose the last one (the key bit) uniformly at random. Then, choose a function Dec' such that it outputs the same bit as Enc' with the corresponding input with probability $(1+\alpha)/2$. Finally, we define an additional oracle $\mathsf{Break}'(c) : \{0,1\}^{2\ell} \to \mathcal{Z}$ using the distribution P_{XZ} for which the above protocol fails; i.e., from the communication c the oracle finds the key bit $S_A = X$ of Alice, and then chooses Z according to $\mathsf{P}_{Z|X}$. Giving Eve access to the oracle Break' does not violate the security requirement of the initial protocol: still no efficient algorithm can find X probability exceeding $\frac{1+\beta}{2}$.

If the protocol is now run, Eve sees an instance of

$$\Big(m, \mathsf{pk}_1, \ldots, \mathsf{pk}_n, c_1, \ldots, c_n\Big).$$

Using the algorithm Break', Eve can obtain $Z_1, \ldots, Z_n$ to get an element of the same distribution as in in (7.13):

$$\Big(Z^n, m, (\mathsf{pk}_1, c_1), \ldots, (\mathsf{pk}_n, c_n)\Big).$$

Together with a PSPACE-algorithm Eve can break the scheme with advantage φ. Since breaking a single instance better than with probability $1-\beta$ is impossible using the security property of the reduction we arrive at a contradiction. □

Now we know exactly for what parameters α and β a restricted black-box reduction exists. We remark that we do not know how to prove that the following theorem also holds for *arbitrary* black-box reduction (even though we think that this could be true).

Theorem 7.11. *Let α, β be constants. There exists a restricted black-box reduction for α-correlated public-key encryption with leakage β to public-key encryption if and only if $\alpha^2 > \beta$.*

Proof. From Corollary 7.8 (noting by inspection that our proof gives a restricted black-box reduction) and from Lemma 7.10. □

A. On the Binomial Distribution

In this section we prove Lemma 2.14. For this we first introduce the binary Kullback-Leibler distance:

Definition A.1 (Kullback-Leibler distance). *For $p, q \in [0,1]$, the binary Kullback-Leibler distance $D(q\|p)$ is defined as*

$$D(q\|p) := q\log\Big(\frac{q}{p}\Big) + (1-q)\log\Big(\frac{1-q}{1-p}\Big).$$

The following lemma describes the asymptotic behavior of $D(p+\varepsilon\|p)$ for small ε.

Lemma A.2. *For $p \geq \frac{1}{2}$, $\varepsilon \geq 0$, $p+\varepsilon < 1$*

$$D(p+\varepsilon\|p) \leq \frac{\varepsilon^2}{2\ln(2)p(1-p)}.$$

Proof. Define the function $f_p(\varepsilon) := \mathcal{D}(p+\varepsilon\|p)$. Taylor's Theorem states that there exists a $\delta \in [0,\varepsilon]$ such that

$$D(p+\varepsilon\|p) = f_p(\varepsilon) = f_p(0) + f_p'(0)\varepsilon + f_p''(0)\frac{\varepsilon^2}{2} + f_p'''(\delta)\frac{\varepsilon^3}{6}. \qquad \text{(A.1)}$$

A simple calculation yields $f_p(0) = f_p'(0) = 0$ and $f_p''(0) = \frac{1}{\ln(2)p(1-p)}$. Also we get

$$f_p'''(\varepsilon) = \frac{2p+2\varepsilon-1}{(p+\varepsilon)^2(p+\varepsilon-1)^2\ln(2)},$$

which is positive for our parameters. Together with (A.1) this gives the lemma. □

We further use the following well known approximation of $n!$ by Stirling.

Proposition A.3. *For any $n > 0$*

$$\sqrt{2\pi}\, n^{n+\frac{1}{2}} e^{-n+1/(12n+1)} < n! < \sqrt{2\pi}\, n^{n+\frac{1}{2}} e^{-n+1/(12n)}, \tag{A.2}$$

or, equivalently,

$$e^{\frac{1}{12n+1}} < \frac{n!\, e^n}{\sqrt{2\pi n}\, n^n} < e^{\frac{1}{12n}}. \tag{A.3}$$

Using Proposition A.3 we can give a bounds on $\mathsf{P}_p(k|n)$; we are interested in the lower bound, the upper bound is given only for completeness.

Lemma A.4. *For $0 < k < n$, let $\mathsf{P}_p(k|n) := \binom{n}{k} p^k (1-p)^{n-k}$. Then*

$$e^{-\frac{1}{12k} - \frac{1}{12(n-k)}} < \mathsf{P}_p(k|n) \sqrt{2\pi \tfrac{k(n-k)}{n}}\, 2^{nD(\frac{k}{n}\|p)} < 1.$$

Proof. We get

$$\begin{aligned}
&\mathsf{P}_p(k|n) \sqrt{2\pi \tfrac{k(n-k)}{n}} 2^{nD(\frac{k}{n}\|p)} \\
&\quad = p^k (1-p)^{n-k} \frac{n!}{k!\,(n-k)!} \cdot \sqrt{2\pi \tfrac{k(n-k)}{n}} \cdot 2^{(k \log(\frac{k}{pn}) + (n-k)\log(\frac{n-k}{(1-p)n}))} \\
&\quad = p^k (1-p)^{n-k} \frac{n!}{k!\,(n-k)!} \cdot \sqrt{2\pi \tfrac{k(n-k)}{n}} \cdot \Big(\frac{k}{pn}\Big)^k \cdot \Big(\frac{n-k}{(1-p)n}\Big)^{n-k} \\
&\quad = \sqrt{2\pi \tfrac{k(n-k)}{n}} \cdot \frac{n!}{k!\,(n-k)!} \cdot \Big(\frac{k}{n}\Big)^k \cdot \Big(\frac{n-k}{n}\Big)^{n-k} \\
&\quad = \frac{n!\, e^n}{\sqrt{2\pi n}\, n^n} \cdot \frac{\sqrt{2\pi k}\, k^k}{k!\, e^k} \cdot \frac{\sqrt{2\pi(n-k)}\, (n-k)^{n-k}}{(n-k)!\, e^{n-k}}.
\end{aligned} \tag{A.4}$$

Using (A.3) three times we obtain

$$\mathsf{P}_p(k|n) \sqrt{2\pi \tfrac{k(n-k)}{n}} 2^{nD(\frac{k}{n}\|p)} > e^{\frac{1}{12n+1}} e^{-\frac{1}{12k}} e^{-\frac{1}{12(n-k)}} > e^{-\frac{1}{12k} - \frac{1}{12(n-k)}}.$$

Analogously (and since either $\frac{1}{12n} < \frac{1}{12k+1}$ or $\frac{1}{12n} < \frac{1}{12(n-k)+1}$)

$$\mathsf{P}_p(k|n) \sqrt{2\pi \tfrac{k(n-k)}{n}} 2^{nD(\frac{k}{n}\|p)} < e^{\frac{1}{12n}} e^{-\frac{1}{12k+1}} e^{-\frac{1}{12(n-k)+1}} < 1. \qquad \square$$

Corollary A.5. *Let $p \geq \frac{1}{2}$, $pn \leq k < n$. Let $\mathsf{P}_p(k|n) := \binom{n}{k} p^k (1-p)^{n-k}$. Then*

$$\mathsf{P}_p(k|n) > e^{-\frac{1}{6(n-k)}} \sqrt{\frac{n}{2\pi k(n-k)}}\, e^{-n\frac{(\frac{k}{n}-p)^2}{2p(1-p)}}.$$

Proof. From Lemma A.4 we get

$$\begin{aligned}\mathsf{P}_p(k|n) &> e^{-\frac{1}{12k}-\frac{1}{12(n-k)}}\sqrt{\frac{n}{2\pi k(n-k)}}\,2^{-nD(\frac{k}{n}\|p)}\\ &\geq e^{-\frac{1}{6(n-k)}}\sqrt{\frac{n}{2\pi k(n-k)}}\,2^{-nD(\frac{k}{n}\|p)},\end{aligned}$$

where we used $k \geq \frac{n}{2}$. Using the estimate in Lemma A.2 concludes the proof. □

We can now prove Lemma 2.14 (reproduced here for convenience).

Lemma 2.14. *Let $p \geq \frac{1}{2}$, $r, s \in \mathbb{N}$ such that $pr + 3s \leq r$. Then,*

$$\sum_{k=\lceil pr\rceil+s}^{\lceil pr\rceil+2s-1} \mathsf{P}_p(k|r) > \frac{s}{2\sqrt{r}}\,e^{-\frac{2s^2}{rp(1-p)}}.$$

Proof. Clearly, $\frac{r}{k(r-k)} \geq \frac{4}{r}$, for all values of r and k in the sum. Since $k < pr + 2s$ for all values in the above sum we get $\frac{k}{r} - p < \frac{2s}{r}$, and also we see that $r - k \geq s$. Using this together with Corollary A.5 thus implies for all k of interest

$$\begin{aligned}\mathsf{P}_p(k|r) &> e^{-\frac{1}{6s}}\sqrt{\frac{2}{\pi r}}\,e^{-r\frac{(\frac{2s}{r})^2}{2p(1-p)}}\\ &> \frac{1}{2\sqrt{r}}\,e^{-\frac{2s^2}{rp(1-p)}}.\end{aligned}$$

Since there are s summands we get the lemma. □

Bibliography

[AC93] Rudolph Ahlswede and Imre Csiszár. Common randomness in information theory and cryptography—part I: Secret sharing. *IEEE Transactions on Information Theory*, 39(4):1121–1132, 1993.

[BBCM95] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Ueli Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory*, 41(6):1915–1923, 1995.

[BBR88] Charles Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SICOMP*, 17(2):210–229, 1988.

[Ber68] Elwyn Berlekamp. Nonbinary BCH decoding. *IEEE Transactions on Information Theory*, 14(2):242, 1968.

[BS93] Gilles Brassard and Louis Salvail. Secret-key reconciliation by public discussion. In Tor Helleseth, editor, *Advances in Cryptology — EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 410–423, 1993.

[Cac97] Christian Cachin. *Entropy Measures and Unconditional Security in Cryptography*. PhD thesis, ETH Zurich, 1997. ISBN 3-89649-185-7.

[CK78] Imre Csiszár and János Körner. Broadcast channels with confidential messages. *IEEE Transactions on Information Theory*, 22(6):644–654, 1978.

[CRW03a] Matthias Christandl, Renato Renner, and Stefan Wolf. A property of the intrinisic mutual information. Long Version of [CRW03b], 2003.

[CRW03b] Matthias Christandl, Renato Renner, and Stefan Wolf. A property of the intrinsic mutual information. In *Proceedings of 2003 IEEE International Symposium on Information Theory*, page 258. IEEE, June 2003.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., first edition, 1991. ISBN 0-471-06259-6.

[CW79] Larry Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[DNR04] Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 342–360, 2004.

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540, 2004.

[For66] G. David Forney. *Concatenated Codes*. PhD thesis, Massachusetts Institute of Technology, 1966. ISBN 0-262-06015-9.

[GNW95] Oded Goldreich, Noam Nisan, and Avi Wigderson. On Yao's XOR-lemma. Technical Report TR95-050, Electronic Colloquium on Computational Complexity (ECCC), 1995.

[Hås90] Johan Håstad. Pseudo-random generators under uniform assumptions. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*, pages 395–404, 1990.

[HHR05] Iftach Haitner, Danny Harnik, and Omer Reingold. On the power of the randomized iterate. Technical Report TR05-135, Electronic Colloquium on Computational Complexity (ECCC), 2005.

[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.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[Hol05] Thomas Holenstein. Key agreement from weak bit agreement. In *Proceedings of the Thirty Seventh Annual ACM Symposium on Theory of Computing*, pages 664–673, 2005.

[HR05a] Thomas Holenstein and Renato Renner. On the smooth Rényi entropy of independently repeated random experiments. In preparation, 2005.

[HR05b] Thomas Holenstein and Renato Renner. One-way secret-key agreement and applications to circuit polarization and immunization of public-key encryption. In Victor Shoup, editor, *Advances in Cryptology — CRYPTO 2005*, Lecture Notes in Computer Science, 2005.

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstract). In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 12–24, 1989.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *The 36th Annual Symposium on Foundations of Computer Science*, pages 538–545, 1995.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 44–61, 1989.

[Jus76] Jørn Justesen. On the complexity of decoding Reed-Solomon codes (corresp.). *IEEE Transactions on Information Theory*, 22(2):237–238, 1976.

[JW99] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *ACM Conference on Computer and Communications Security 1999*, pages 28–36, 1999.

[KS03] Adam R. Klivans and Rocco A. Servedio. Boosting and hard-core set construction. *Machine Learning*, 51(3):217–238, 2003.

[Lev87] Leonid A. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, 1987.

[Lup58] Oleg B. Lupanov. A method of circuit synthesis. *Izvestiya VUZ, Radiofizika*, 1(1):120–140, 1958. In Russian.

[LW95] Michael Luby and Avi Wigderson. Pairwise independence and derandomization. Technical Report ICSI TR-95-035, International Computer Science Institute, Berkeley, CA, 1995.

[Mas69] James L. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, 15(1):122–127, 1969.

[Mau93] Ueli Maurer. Secret key agreement by public discussion. *IEEE Transactions on Information Theory*, 39(3):733–742, 1993.

[Mer79] Ralph C. Merkle. Secure communications over insecure channels. *Communications of the ACM*, 21(4):294–299, 1979.

[MW99] Ueli Maurer and Stefan Wolf. Unconditionally secure key agreement and the intrinsic conditional information. *IEEE Transaction on Information Theory*, 45(2):499–514, 1999.

[Ren05] Renato Renner. *Security of Quantum Key Distribution*. PhD thesis, Swiss Federal Institute of Technology, Zürich, 2005. `http://arxiv.org/abs/quant-ph/0512258`.

[RS60] Irving S. Reed and Gustave Solomon. Polynomial codes over certain finite fields. *SIAM Journal of Applied Mathematics*, 8(2):300–304, 1960.

[RW03] Renato Renner and Stefan Wolf. New bounds in secret-key agreement: The gap between formation and secrecy extraction. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 562–577, 2003.

[RW05] Renato Renner and Stefan Wolf. Simple and tight bounds for information reconciliation and privacy amplification. In Bimal Roy, editor, *Advances in Cryptology — ASIACRYPT 2005*, volume 3788 of *Lecture Notes in Computer Science*, pages 199–216, 2005.

[Sha48] Claude E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:373–423 and 27:623–656, 1948.

[Sha49a] Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28:656–715, 1949.

[Sha49b] Claude E. Shannon. The synthesis of two-terminal switching circuits. *Bell Systems Technical Journal*, 28:59–98, 1949.

[Sha02] Ronen Shaltiel. Recent developments in extractors, 2002. Manuscript.

[SV97] Amit Sahai and Salil Vadhan. A complete promise problem for statistical zero-knowledge. In *The 38th Annual Symposium on Foundations of Computer Science*, pages 448–457, 1997.

[SV99] Amit Sahai and Salil Vadhan. Manipulating statistical difference. In Panos Pardalos, Sanguthevar Rajasekaran, and José Rolim, editors, *Randomization Methods in Algorithm Design (DIMACS Workshop, December 1997)*, volume 43 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 251–270, 1999.

[Tre03] Luca Trevisan. List-decoding using the XOR lemma. In *The 44th Annual Symposium on Foundations of Computer Science*, pages 126–135, 2003.

[Vad99] Salil Vadhan. *A Study of Statistical Zero-Knowledge Proofs*. PhD thesis, Massachusets Institute of Technology, 1999.

[vN28] John von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928.

[WB86] Lloyd R. Welch and Elwyn R. Berlekamp. Error correction of algebraic block codes. U.S. patent, number 4,633,470, 1986.

[Weg87] Ingo Wegener. *The Complexity of Boolean Functions*. John Wiley & Sons Ltd, and B. G. Teubner, Stuttgart, 1987. ISBN 0-471-91555-6, available on ECCC.

[Wol99] Stefan Wolf. *Information-Theoretically and Computationally Secure Key Agreement in Cryptography*. PhD thesis, Swiss Federal Institute of Technology, Zürich, 1999.

[Wyn75] Aaron D. Wyner. The wire-tap channel. *Bell Systems Technical Journal*, 54:1355–1387, 1975.

[Yao82] Andrew C. Yao. Theory and applications of trapdoor functions (extended abstract). In *The 23rd Annual Symposium on Foundations of Computer Science*, pages 80–91, 1982.