Diss. ETH No. 13138

Information-Theoretically and Computationally Secure Key Agreement in Cryptography

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZÜRICH

for the degree of
Doctor of Technical Sciences

presented by

STEFAN WOLF
Dipl. Math. ETH

born September 14, 1969
in Schaffhausen
citizen of Switzerland

accepted on the recommendation of

Prof. Dr. Ueli Maurer, examiner
Prof. Dr. Claude Crépeau, co-examiner

1999

to Eve

Acknowledgments

The research leading to this thesis was carried out under the supervision of my advisor and co-author Prof. Ueli Maurer. His door was always open to me, and it was a pleasure collaborating with him. I am particularly grateful for many inspiring discussions as well as for his permanent support and interest in my work.

I thank Prof. Claude Crépeau for being the co-examiner of this thesis.

I also thank all the previous and present members of the cryptography research group at ETH for the unique atmosphere through all the years. Particular thanks go to Christian Cachin, Jan Camenisch, and Martin Hirt for their patience in the execution of their duty as computer-system consultants, a burden that a mathematician's officemate must bear. With Reto Kohlas and Matthias Fitzi I also shared many non-professional interests. Many thanks go to Ulrike Stege, also for reading a preliminary version of some parts of this thesis.

An interesting research cooperation was carried out with Prof. Nicolas Gisin and his group at the University of Geneva. I thank them very much for their invitations and hospitality. I thank Igor Shparlinsky and Joachim von zur Gathen for inviting me to an exciting meeting in Oberwolfach, where I had the opportunity to present parts of my work.

Finally, I thank various people for interesting discussions on the subjects treated in this thesis, and for their helpful remarks. Among them are Daniel Bleichenbacher, Dan Boneh, Gilles Brassard, Ronald Cramer, Ivan Damgård, Marten van Dijk, Dima Grigoriev, Hendrik Lenstra, Shengli Liu, Jim Massey, Kevin McCurley, Victor Shoup, Douglas Stinson, and Salil Vadhan, as well as my diploma students Andrin Bachmann, Patrick Feisthammel, Clemens Holenstein, Olaf Keller, Christoph Lukesch, Markus Metzger, and Bernhard Weibel.

This research was partially supported by the Swiss National Science Foundation (SNF), grant No. 20–42105.94.

Abstract

One of the central problems in cryptography is secure message transmission over insecure channels. By the use of classical symmetric cryptosystems, this problem can be reduced to the one of generating a secret key by insecure communication. The security of such secret-key agreement can be based either on the computational hardness of certain number-theoretic problems, or on information theory. Examples of both types of key distribution are analyzed.

In the information-theoretic model, Maurer's setting of secret-key agreement by public discussion from common information and earlier scenarios, based on noisy communication channels, by Wyner and by Csiszár and Körner, are considered. It is shown that in the noisy-channel models, as also in Maurer's model, the generated secret key can have a much higher security level than previously believed and demanded. By using recent results from the theory of extractors, it is shown that such a stronger key can even be generated at the same rate as a key that is secure only with respect to the previous, weaker definition. Furthermore, conditions on the joint distribution of the legitimate partners' and the adversary's knowledge are given for the possibility and impossibility of information-theoretic secret-key agreement. In this context, a new information measure, the intrinsic conditional information, is introduced, and evidence is provided which suggests that this quantity is directly connected to the possibility of secret-key agreement. The case of key agreement secure even against active adversaries is also considered. An efficiently-verifiable criterion is given that allows for deciding whether such key agreement is possible at all in a specific situation. Finally, two protocols are described for the important special case of privacy amplification, i.e., transforming a common partially-secret string into a virtually-secret key, in the presence of an active adversary. The conditions for these protocols to work however are stricter than in

the passive-adversary case. Both protocols are based on novel interactive identification and message-authentication methods requiring only a partially-secret key.

In the computational model, we consider a number of question related to the security of the Diffie-Hellman key-agreement protocol. For instance, the relationship between the discrete-logarithm problem, the Diffie-Hellman problem, and the Diffie-Hellman decision problem is studied. The security of the Diffie-Hellman protocol is directly based on the hardness of the computational and decisional variants of the Diffie-Hellman problem. It is shown that with respect to generic algorithms, i.e., general-purpose algorithms that work in any group of a certain order, the Diffie-Hellman problem and the discrete-logarithm problem are computationally equivalent for certain, but not for all group orders. The Diffie-Hellman decision problem, which is to recognize a correct solution to the Diffie-Hellman problem, is not generically computationally equivalent to the Diffie-Hellman problem for any non-trivial group order. Finally, additional related problems are studied with respect to generic algorithms, such as the computation of roots in cyclic groups, or the improvement of a faulty Diffie-Hellman oracle. In all cases, the derived lower bounds turn out to nearly match the running times of the fastest generic algorithms. This leads to a complete classification of the problems of interest with respect to generic algorithms.

Zusammenfassung

Eines der zentralen Probleme der Kryptographie ist die sichere Übertragung von Nachrichten über unsichere Kanäle. Klassische symmetrische Verschlüsselungsmethoden erlauben es, dieses Problem auf die Erzeugung sicherer Schlüssel mittels unsicherer Kommunikation zu reduzieren. Die Sicherheit von Verfahren zur Schlüsselgenerierung kann einerseits auf der berechenmässigen Schwierigkeit gewisser zahlentheoretischer Probleme oder andererseits auf informationstheoretischen Argumenten beruhen. Wir analysieren Verfahren zur Schlüsselverteilung beiderlei Typs.

Im informationstheoretischen Modell betrachten wir das Szenario von Maurer, welches Schlüsselerzeugung mittels öffentlicher Kommunikation erlaubt unter der Voraussetzung, dass die beiden Parteien, wie aber auch der Gegner, gewisse korrelierte Information besitzen. Weitere Beispiele sind die Szenarien von Wyner sowie von Csiszár und Körner, die auf der Existenz verrauschter Kommunikationskanäle beruhen. Wir zeigen, dass in den letzteren Modellen, wie auch in Maurers Modell, die Bedingungen an die Sicherheit des erzeugten Schlüssels viel strenger sein können als bisher angenommen und gefordert. Neueste Resultate aus der Theorie der sogenannten Extraktoren lassen den Beweis zu, dass ein Schlüssel, der bezüglich der restriktiven Definition sicher ist, mit derselben Rate erzeugt werden kann wie ein Schlüssel, der nur sicher zu sein braucht im Sinne der herkömmlichen Definition. Ferner leiten wir Bedingungen her, unter denen informationstheoretisch sichere Schlüsselerzeugung möglich bzw. sicher nicht möglich ist. In diesem Zusammenhang wird ein neues Informationsmass, die bedingte intrinsische Information, definiert. Wir analysieren Beispiele, welche die Vermutung erhärten, dass dieses Mass direkt zusammenhängt mit der Möglichkeit beweisbar sicherer Schlüsselgenerierung. Wir betrachten auch Protokolle für Schlüsselverteilung, die nicht nur gegen passive, sondern auch gegen aktive Gegner,

welche die Kommunikation nicht nur abhören, sondern auch versuchen, Meldungen einzufügen, zu verändern oder sogar zu löschen, sicher sind. Wir beschreiben ein effizient verifizierbares Kriterium um zu entscheiden, ob in einer gewissen Situation solche Schlüsselerzeugung überhaupt möglich sein kann. Schliesslich werden zwei neue Protokolle beschrieben für sogenannte "Privacy Amplification" (Transformation eines teilweise sicheren in einen kürzeren, aber sicheren Schlüssel), die auch gegen einen aktiven Gegner sicher sind. Die Bedingungen an den anfänglich unsicheren Schlüssel sind allerdings stärker als im Falle eines nur passiven Gegners. In beiden Protokollen wird eine neue Identifikations- und Nachrichtenauthentisierungstechnik verwendet, die interaktiv ist und auch dann funktioniert, wenn der benützte Schlüssel einen nur niedrigen Grad an Sicherheit besitzt.

Im Modell berechenmässiger Sicherheit analysieren wir das Diffie-Hellman-Protokoll zur Schlüsselvereinbarung. Unter anderem vergleichen wir das sogenannte Problem des diskreten Logarithmus mit verschiedenen Varianten des Diffie-Hellman-Problems, welches ein Gegner lösen muss, um das erwähnte Protokoll zu brechen. Es wird gezeigt, dass in bezug auf generische Algorithmen, die in allen möglichen Gruppen einer gewissen Ordnung anwendbar sind, das Diffie-Hellman-Problem zwar für gewisse, sicher aber nicht für alle Gruppenordnungen berechenmässig äquivalent zum Problem ist, diskrete Logarithmen in der entsprechenden Gruppe zu berechnen. Ferner zeigen wir, dass die Aufgabe, eine korrekte Lösung des Diffie-Hellman-Problems nur zu erkennen (d.h. von einer falschen zu unterscheiden), nie in einem generischen Sinn berechenmässig äquivalent ist zum Diffie-Hellman-Problem. Schliesslich untersuchen wir weitere Berechnungs- und Entscheidungsprobleme in zyklischen Gruppen, wie z.B. das Ziehen von Wurzeln oder die Transformation eines fehlerhaften sogenannten Diffie-Hellman-Orakels in ein zuverlässiges. In den meisten Fällen sind die hergeleiteten theoretischen Komplexitätsuntergrenzen nahezu identisch mit der tatsächlichen Laufzeit der besten generischen Algorithmen. Dies führt zu einer kompletten Klassifikation der betrachteten Probleme im natürlichen Rechenmodell der generischen Algorithmen.

Contents

Part I

Introduction and Preliminaries

Chapter 1

Introduction

One of the basic problems in cryptography is to transmit a message securely from a sender, often called *Alice*, to a receiver, *Bob*, over an insecure channel overheard by an adversary, *Eve*, in such a way that Eve does not learn (any information about) the message. One possibility for solving this problem is that the parties initially generate a secret key by insecure communication, and then use this key as the private key of a classical cryptosystem (e.g., the one-time pad) for transmitting the message securely. This motivates the study of the problem of *secret-key agreement by insecure communication.* In the following, we distinguish between information-theoretically (Part I) and computationally (Part II) secure key agreement.

The security of an *information-theoretically-secure* scheme is based on probability-theoretic statements. Information theory, which was initiated some 50 years ago by *Claude Shannon*, is a mathematical theory of information and communication based on probability theory and statistics.

Computational security on the other hand is based on the hardness of certain (number-theoretic) problems. However, because no rigorously-proven general lower bounds exist in complexity theory, this type of security cannot be unconditional up to date. An important task however is to base the security of a particular scheme on a (number of) widely-accepted standard assumptions, such as the hardness of the integer-factoring, discrete-logarithm, Diffie-Hellman, or Diffie-Hellman decision problems. The relationship between these problems is considered with respect to generic, i.e., general-purpose, algorithms.

We study examples of secret-key agreement of both types. Unconditional security based on noisy channels or, equivalently, on correlated information, was studied by several authors, starting from, and qualifying, Shannon's pessimistic impracticality result. In many realistic models, secret-key agreement whose security can be rigorously proven was shown to be possible. We take a number of steps on this promising path.

The *Diffie-Hellman protocol* was presented by *Whitfield Diffie* and *Martin Hellman* in 1976 in a revolutionary paper that provided the foundation for later developments in public-key cryptography. We consider many questions more or less directly related to the security of this simple and elegant protocol.

Chapter 2

Notation and Information-Theoretic Preliminaries

In this chapter we introduce some notation and give a brief overview over certain basic concepts of discrete probability and information theory that are used in all parts of this thesis.

Unlike most other scientific disciplines, the foundations of information theory are mainly due to a single person, Claude Shannon. We only mention here his well-known papers introducing some of the main concepts of information theory such as entropy or mutual information [107], and of information-theoretic security in communication [106]. Detailed introductions into probability and information theory are the books by Feller [38], Cover and Thomas [29], or Blahut [7].

The algebraic and number-theoretic notation and preliminaries required in Part III are introduced there.

2.1 Notation

In the following, $\mathbf{N}$, $\mathbf{Z}$, $\mathbf{R}$, and $\mathbf{C}$ stand for the sets of natural numbers, integers, real and complex numbers, respectively, whereas $\mathbf{R}_{\geq 0}$ are the non-negative reals.

We denote random variables by capital letters $X, Y, Z, \ldots$, their ranges by the corresponding calligraphic letters $\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \ldots$, whereas $x, y, z, \ldots$ stand for particular elements of the ranges.

We write $a \in_R A$ if an element a is chosen randomly from a set A according to the uniform distribution. The cardinality of a set A is denoted by $|A|$, whereas 2^A is the power set of A. The length of a string S is denoted by $\mathrm{len}(S)$. The concatenation of strings S_1 and S_2 is written as $S_1||S_2$. By $\log(x)$ we denote the *binary* logarithm of x, and $\ln(x)$ stands for the *natural* logarithm, i.e., with respect to the basis e.

Let f, g be non-negative real-valued functions defined on an unbounded set $\mathcal{A} \subseteq \mathbf{R}_{\geq 0}$. Then we write

$$f(x) = O(g(x))$$

if there exist constants $R, C \geq 0$ such that $f(x) \leq C \cdot g(x)$ holds for all $x \in \mathcal{A}$ with $x \geq R$. Moreover,

$$f(x) = \Omega(g(x))$$

means that there exist constants $R \geq 0$ and $c > 0$ such that $f(x) \geq c \cdot g(x)$ holds for all $x \in \mathcal{A}$ with $x \geq R$. Furthermore,

$$f(x) = \Theta(g(x))$$

stands for $f(x) = O(g(x))$ and $f(x) = \Omega(g(x))$ together. Finally,

$$f(x) = o(g(x))$$

denotes the fact that $f(x)/g(x) \to 0$ for $x \to \infty$. Somewhat abusing the notation, we write for instance

$$f(x) = 2^{-\Omega(g(x))}$$

if $-\log(f(x)) = \Omega(g(x))$.

The "soft" O-Notation is defined as follows. The idea is to neglect factors which are polynomially bounded in the input length $\log x$.

. We write

$$f(x) = O_s(g(x))$$

if there exists a constant $c \geq 0$ such that

$$f(x) = O(g(x) \cdot (\log x)^c) \ .$$

Furthermore,

$$f(x) = \Omega_s(g(x))$$

is true if, for some fixed $c \geq 0$,

$$f(x) = \Omega(g(x)/(\log x)^c) \ .$$

Finally,

$$f(x) = \Theta_s(g(x))$$

means that $f(x) = O_s(g(x))$ and $f(x) = \Omega_s(g(x))$ both hold.

2.2 Discrete Probability Theory

2.2.1 Discrete Random Experiments

A *discrete random experiment* is a pair (Ω, P), where Ω is a finite or countably-infinite set, called *sample space*, and P is a function

$$\mathrm{P} \, : \, \Omega \longrightarrow [0,1] \ ,$$

called the *probability function*, such that

$$\sum_{\omega \in \Omega} \mathrm{P}(\omega) = 1 \ .$$

An element $\omega \in \Omega$ is called an *elementary event.* A set $\mathcal{A} \subseteq \Omega$ is an *event.* The mapping P is extended to the set 2^Ω of events by

$$\mathrm{P}[\mathcal{A}] := \sum_{\omega \in \mathcal{A}} \mathrm{P}(\omega) \ .$$

(Formally, the probability distribution $\mathrm{P}[\cdot]$ is not an extension of the probability function $\mathrm{P}(\cdot)$, but a new mapping satisfying $\mathrm{P}[\{\omega\}] = \mathrm{P}(\omega)$.) The probability distribution P satisfies the following properties:

$$\begin{array}{rcll} 0 \leq \mathrm{P}[\mathcal{A}] & \leq & 1 & \\ \mathrm{P}[\emptyset] & = & 0 & \\ \mathrm{P}[\Omega] & = & 1 & \\ \mathrm{P}[\mathcal{A}] & \leq & \mathrm{P}[\mathcal{B}] & \qquad (\text{if } \mathcal{A} \subseteq \mathcal{B}) \\ \mathrm{P}[A \cup B] & = & \mathrm{P}[\mathcal{A}] + \mathrm{P}[\mathcal{B}] - \mathrm{P}[\mathcal{A} \cap \mathcal{B}] & \\ & \leq & \mathrm{P}[\mathcal{A}] + \mathrm{P}[\mathcal{B}] \ . & \end{array}$$

The fourth property is called *monotonicity*, whereas the last inequality is the *union bound*. For two events $\mathcal{A}$ and $\mathcal{B}$ with $\mathrm{P}[\mathcal{B}] > 0$, the *conditional probability* of $\mathcal{A}$ given $\mathcal{B}$ is

$$\mathrm{P}[\mathcal{A}|\mathcal{B}] := \frac{\mathrm{P}[\mathcal{A} \cap \mathcal{B}]}{\mathrm{P}[\mathcal{B}]} .$$

Two events $\mathcal{A}$ and $\mathcal{B}$ are called *(statistically) independent* if

$$\mathrm{P}[\mathcal{A} \cap \mathcal{B}] = \mathrm{P}[\mathcal{A}] \cdot \mathrm{P}[\mathcal{B}] .$$

If $\mathcal{A}$ and $\mathcal{B}$ are independent then either $\mathrm{P}[\mathcal{B}] = 0$ or $\mathrm{P}[\mathcal{A}|\mathcal{B}] = \mathrm{P}[\mathcal{A}]$ holds.

2.2.2 Discrete Random Variables

Let a discrete random experiment (Ω, P) be given. A *discrete random variable* is a mapping

$$X \,:\, \Omega \longrightarrow \mathcal{X}$$

to some set $\mathcal{X}$. The probability distribution P_X of X is defined as

$$P_X(x) := \sum_{\omega : X(\omega) = x} \mathrm{P}(\omega) .$$

The *joint distribution* $P_{X_1 X_2 \cdots X_N}$ of N random variables is a probability distribution over the set $\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_N$. The random variables $X_1, X_2, \ldots, X_N$ are called *statistically independent* if

$$P_{X_1 X_2 \cdots X_N}(x_1, x_2, \ldots, x_N) = P_{X_1}(x_1) \cdot P_{X_2}(x_2) \cdots P_{X_N}(x_N)$$

holds for all $x_1, x_2, \ldots, x_N$, i.e., when the joint distribution equals the product of the *marginal distributions*, which are determined by the joint distribution by

$$P_{X_i}(x_i) = \sum_{j \neq i \,:\, x_j \in \mathcal{X}_j} P_{X_1 X_2 \cdots X_N}(x_1, x_2, \ldots, x_N) .$$

2.2.3 Conditional Distributions and Markov Chains

The *conditional distribution* of X, given that the event $\mathcal{A}$ (with $\mathrm{P}[\mathcal{A}] > 0$) occurs, is defined as

$$P_{X|\mathcal{A}}(x) := \mathrm{P}[\{X = x\}|\mathcal{A}] = \frac{\mathrm{P}[\{X = x\} \cap \mathcal{A}]}{\mathrm{P}[\mathcal{A}]} .$$

As a special case, a random variable can be conditioned on the event

$$\mathcal{A} := \{Y = y\}$$

that another random variable Y takes a particular value y. The resulting distribution

$$P_{X|Y}(x, y) := P_{X|Y=y}(x)$$

is called the *conditional distribution of X given Y*. Note that the function $P_{X|Y}(\cdot\,,\cdot)$ with two arguments is *not* a probability distribution on $\mathcal{X} \times \mathcal{Y}$, but for every $y \in \mathcal{Y}$, the function $P_{X|Y}(\cdot\,, y)$ is a distribution on $\mathcal{X}$.

A sequence of random variables $X_1, X_2, \ldots, X_N$ is called a *Markov chain*, denoted by

$$X_1 \to X_2 \to \cdots \to X_N \ ,$$

if for all $i > 1$,

$$P_{X_i|X_1 X_2 \cdots X_{i-1}} = P_{X_i|X_{i-1}} \ . \tag{2.1}$$

Equation (2.1) stands for the formally correct condition that

$$P_{X_i|X_1 X_2 \cdots X_{i-1}}(x_i, x_1, x_2, \ldots, x_{i-1}) = P_{X_i|X_{i-1}}(x_i, x_{i-1})$$

holds for all $x_1, x_2, \ldots, x_{i-1}, x_i$. Intuitively, $X \to Y \to Z$ means that given Y (the "present"), X (the "past") and Z (the "future") are independent.

2.2.4 The Expectation and Jensen's Inequality

Let X be a random variable with $\mathcal{X} \subseteq \mathbf{R}$. Then the *expectation* of X is defined as

$$\mathrm{E}[X] := \sum_{x \in \mathcal{X}} P_X(x) \cdot x$$

if either $\sum_{x \in \mathcal{X}} P_X(x) \cdot \max\{x, 0\} < \infty$ or $\sum_{x \in \mathcal{X}} P_X(x) \cdot \min\{x, 0\} > -\infty$. Note that $\mathrm{E}[X] \in \{-\infty, \infty\}$ is possible.

Lemma 2.1 states *Jensen's inequality*, which is of great importance in probability theory and information theory. Most of the basic inequalities in information theory follow directly from this inequality. Let X be a random variable, and let $f : \mathcal{X} \to \mathcal{Y}$ be a mapping. Then the new random variable $Y = f(X)$ is defined by $Y(\omega) := f(X(\omega))$. For $\mathcal{X} \subseteq \mathbf{R}$, a function $\mathcal{X} \to \mathbf{R}$ is called *convex* if for all $x_1, x_2 \in \mathcal{X}$ and $\lambda \in [0,1]$

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

holds. f is called *concave* if $-f$ is convex.

Lemma 2.1. *Let X be a random variable with $\mathcal{X} \subseteq \mathbf{R}$, and let $f : \mathcal{X} \to \mathbf{R}$ be a convex function. Then*

$$f(\mathrm{E}[X]) \leq \mathrm{E}[f(X)] . \tag{2.2}$$

If $g : \mathcal{X} \to \mathbf{R}$ is concave, then $g(\mathrm{E}[X]) \geq \mathrm{E}[g(X)]$ holds.

Finally, we define a metric, i.e., a distance function, on the set of distributions on a discrete set $\mathcal{X}$. Let P_X and P_Y be two probability distributions on the same set $\mathcal{X}$. The *variational distance* between P_X and P_Y is defined as

$$d(P_X, P_Y) := \frac{1}{2} \sum_{x \in \mathcal{X}} |P_X(x) - P_Y(x)| .$$

2.2.5 The Variance and the Standard Deviation

Let X be a random variable with $\mathcal{X} \subseteq \mathbf{R}$. The *variance* of X is defined as

$$\mathrm{Var}[X] := \mathrm{E}[(X - \mathrm{E}[X])^2] = \mathrm{E}[X^2] - \mathrm{E}[X]^2 .$$

Clearly, $\mathrm{Var}[X] \geq 0$ holds for every X (with equality if and only if $P_X(\mathrm{E}[X]) = 1$). The quantity

$$\sigma(X) := \sqrt{\mathrm{Var}[X]}$$

is called the *standard deviation* of X.

Let $X_1, \ldots, X_N$ be pairwise independent random variables with $\mathcal{X}_i \subseteq \mathbf{R}$ for $1 \leq i \leq N$. Then the variance of the sum equals the sum of the variances, i.e.,

$$\mathrm{Var}[X_1 + \cdots + X_N] = \mathrm{Var}[X_1] + \cdots + \mathrm{Var}[X_N] .$$

Furthermore, if $k \in \mathbf{R}$ is a constant and X is a random variable with $\mathcal{X} \subseteq \mathbf{R}$, then

$$\mathrm{Var}\left[k \cdot X\right] = k^2 \cdot \mathrm{Var}\left[X\right] .$$

A consequence of these facts is Lemma 2.2, which is a variant of the law of large numbers.

Lemma 2.2. *Let $X_1, X_2, X_3, \ldots$ be independent binary random variables with identical distribution P_{X_i} with $P_{X_i}(1) = p$. Let further $\alpha, \delta' > 0$. If t is the smallest number such that the event*

$$\frac{X_1 + \cdots + X_t}{t} \in [p - \delta', p + \delta']$$

has probability at least $1 - \alpha$, then

$$t = O\left(\frac{\log(1/\alpha)}{\delta'^2}\right)$$

holds.

Proof. Since the random variables X_i are independent, we have

$$\mathrm{Var}\left(\frac{X_1 + \cdots + X_t}{t}\right) = \frac{t \cdot \mathrm{Var}(X_i)}{t^2} = \Theta(1/t) .$$

Hence the number of standard deviations corresponding to δ' is of order $\Theta(\delta'\sqrt{t})$. The normal approximation of the binary distribution leads to

$$\delta'\sqrt{t} = \Theta\left(\sqrt{\log(1/\alpha)}\right)$$

or $t = \Theta(\log(1/\alpha)/\delta'^2)$. □

2.3 Uncertainty and Entropy

2.3.1 Bar Kochba, the Game of 20 Questions, and Binary Coding

The following story has been reported about *Bar Kochba* (the "Son of the Star"), leader of the Jews during their independence war in 135

B.C., who defended his fortress heroically against a superior number of Romans [98].

"It is also said that Bar Kochba sent out a scout to the Roman camp who was captured and tortured, having his tongue cut out. He escaped from captivity and reported back to Bar Kochba, but being unable to talk, he could not tell in words what he had seen. Bar Kochba accordingly asked him questions which he could answer by nodding or shaking his head. Thus he acquired from his mute scout the information he needed to defend the fortress. [...] It occurred to me that, if the story of Bar Kochba were true, then he would have been the forefather of information theory".

In the so-called *Bar-Kochba game*, one player has to find out, by asking yes/no-questions, what the second player has in mind. This game was extremely popular among writers in Budapest at the beginning of this century. Regardless of the (possibly adaptive) strategy of the questioner he cannot, with at most 20 questions, distinguish between more than 2^{20}, i.e., about one million, different objects (because there are only 2^{20} ways of answering the 20 questions differently). On the other hand, given that the object to be found comes from a set of size at most n, then $\lceil \log n \rceil$ questions are always sufficient if the following strategy is used. Let a fixed encoding of all the objects by binary strings of length 20 be defined. Then, the strategy is to ask whether the first, second, ... bit of the encoding is 1.

This example shows the close relationship between the Bar-Kochba game and binary coding. For a random variable X that takes one of $n = 2^k$ values with equal probabilities, the minimal average number of questions in the Bar-Kochba game, as well as the minimal average codeword length of a prefix-free binary code, is k. Note that this bound cannot be beaten even if a strategy is used with variable codeword lengths for the different outcomes. We call this quantity the uncertainty or *entropy* of X, denoted by $H(X)$.

If the size of the range $\mathcal{X}$ of X is not a power of 2, then the average number of questions required obviously lies between $\lfloor \log |\mathcal{X}| \rfloor$ and $\lceil \log |\mathcal{X}| \rceil$. When combining r independent realizations of the random variable X, the optimal average number of questions required to learn all the outcomes together lies between $\lfloor \log |\mathcal{X}|^r \rfloor$ and $\lceil \log |\mathcal{X}|^r \rceil$. Taking such combinations into account, we obtain for the entropy of X that

$$\log |\mathcal{X}| - \frac{1}{r} < \frac{\lfloor \log |\mathcal{X}|^r \rfloor}{r} \le H(X) \le \frac{\lceil \log |\mathcal{X}|^r \rceil}{r} < \log |\mathcal{X}| + \frac{1}{r}$$

for all $r \geq 1$, hence

$$H(X) = \log |\mathcal{X}| \ . \tag{2.3}$$

Equation (2.3) is called *Hartley's formula* and gives the entropy of a uniformly-distributed random variable.

We consider an example of a random variable Y that is *not* uniformly distributed. Let $\mathcal{Y} = \{a, b, c, d\}$, with $P_Y(a) = 1/2$, $P_Y(b) = 1/4$, $P_Y(c) = P_Y(d) = 1/8$. We conclude from the above that two questions are always sufficient, hence $H(Y) \leq 2$. However, there is a better strategy of asking questions or, equivalently, a prefix-free code with a shorter average codeword length, namely,

$$a \rightsquigarrow 0 \ , \quad b \rightsquigarrow 10 \ , \quad c \rightsquigarrow 110 \ , \quad d \rightsquigarrow 111 \ .$$

The average number of questions required when asking the bits of the codewords is

$$\frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = \frac{7}{4} \ \ (< 2) \ .$$

On the other hand, this code (or strategy of asking questions) is optimal. Note that when encoding a sequence of independent realizations of Y, the resulting codeword string is a sequence of independent symmetric bits. Observe further that in this example, the length of the codeword of a letter is $\log(1/p)$, where p is the probability of this letter. The quantity $\log(1/p)$ is sometimes called the *unexpectedness* of a elementary event with probability p.

A code is optimal if the length of every codeword is equal to the unexpectedness of the corresponding outcome. Hence, for a random variable Y for which each probability p_i is of the form $p_i = 2^{-s_i}$ for an integer s_i, we have

$$H(Y) = p_1 \log(1/p_1) + p_2 \log(1/p_2) + \cdots \ . \tag{2.4}$$

Equation (2.4) is called *Shannon formula*, and is a generalization of Hartley's formula (2.3). By combining independent realizations of the random variable for the encoding, one obtains that this formula gives the entropy of any discrete random variable.

2.3.2 Shannon Entropy

The following definition was given by Shannon in 1948 [107].

Definition 2.1. **[107]** The *(Shannon) entropy* $H(X)$ of a random variable X with distribution P_X is given by

$$H(X) := \sum_{x \in \mathcal{X}} -P_X(x) \cdot \log P_X(x) = \mathrm{E}[-\log P_X]$$

(where $0 \log 0 := 0$.) ○

Although the entropy of a random variable is a real number, the units are often called *bits* (basic information units).

The entropy of a *binary* random variable with probability distribution $[p, 1-p]$ is given by the *binary entropy function*

$$h(p) := -p \log p - (1-p) \log(1-p)$$

(see Figure 2.1).

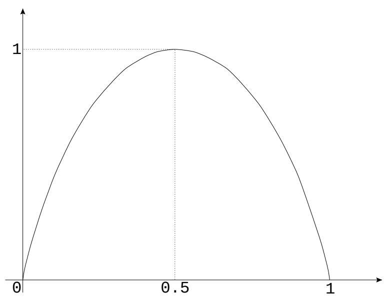


Figure 2.1: The Binary Entropy Function

Clearly, the Shannon entropy $H(X)$ is a function of the distribution P_X of the random variable X; we alternatively write $H(P_X)$ sometimes. The *joint entropy* of random variables $X_1, X_2, \ldots, X_N$ is the entropy of the joint distribution, i.e.,

$$H(X_1 X_2 \cdots X_N) := H(P_{X_1 X_2 \cdots X_N}) \ .$$

Moreover, Definition 2.1 also covers the case where the distribution is conditioned on an event $\mathcal{A}$. We write $H(X|\mathcal{A}) := H(P_{X|\mathcal{A}})$ or, if $\mathcal{A} = \{Y = y\}$,

$$H(X|Y = y) := H(P_{X|Y=y}) \ .$$

2.3.3 Properties of Shannon Entropy

The entropy of a random variable X is always non-negative and upper bounded by the binary logarithm of the cardinality of the range, i.e.,

$$0 \leq H(X) \leq \log |\mathcal{X}| \ . \tag{2.5}$$

The second inequality, which is intuitively clear from the discussion above, follows from Jensen's inequality for *concave* functions:

$$H(X) = \mathrm{E}[\log(1/P_X)] \leq \log(\mathrm{E}[1/P_X]) = \log |\mathcal{X}| \ .$$

Equality on the left-hand side of (2.5) holds if and only if there exists an element $x_0 \in \mathcal{X}$ with $P_X(x_0) = 1$, whereas equality on the right-hand side is equivalent to the fact that X is uniformly distributed over $\mathcal{X}$, i.e., that $P_X(x) = 1/|\mathcal{X}|$ holds for all $x \in \mathcal{X}$.

In the special case where the outcome of the random experiment is a binary string S of a certain length, the second inequality of (2.5) implies that the entropy of the random variable can be equal to, but not exceed the length of S. Equivalently, the entropy about the value of M *binary digits* can be at most M.

If the additional random variable U is a party's entire knowledge about a binary string S, we say that this party has

$$n - H(S|U = u)$$

bits of *Shannon information* about S. This definition can be analogously made for Rényi entropy and min-entropy (introduced in Section 2.6.2).

Lemma 2.3 quantifies the fact that if equality on the right-hand side of (2.5) does not hold, then X is not uniformly distributed. It states that if $H(X)$ is *not* close to the maximum $\log |\mathcal{X}|$, then X is not close-to-uniformly distributed.

Lemma 2.3. *Let Z be a random variable with range $\mathcal{Z} \subset \{0,1\}^k$. Then*

$$H(Z) \geq k \cdot (1 - d(U_k, P_Z) - 2^{-k}) \ , \tag{2.6}$$

where U_k stands for the uniform distribution over $\{0,1\}^k$.

Proof. Let $d := d(U_k, Z)$. We can assume that $d < 1 - 2^{-k}$ because otherwise the inequality is trivially satisfied. The distribution P_Z of Z can be thought of as obtained from the uniform distribution U_k by increasing some of the probabilities (by total amount d) and decreasing some others (by the same total amount). The function

$$\frac{\mathrm{d}}{\mathrm{d}p}(-p\log p) = -\frac{\ln p + 1}{\ln 2}$$

is monotonically decreasing, hence increasing [decreasing] a *smaller* probability increases [decreases] the entropy more than modifying a greater probability by the same amount. Hence a distribution with distance d from U_k with minimal entropy can be obtained by adding d to one of the probabilities, and by reducing as many probabilities as possible to 0, leaving the other probabilities unchanged. One of the probabilities of the new distribution equals $2^{-k} + d$, $\lfloor 2^k d \rfloor$ probabilities are equal to 0, one probability equals $2^{-k}(2^k d - \lfloor 2^k d \rfloor)$ (if this is not 0), and $\lfloor 2^k(1-d) \rfloor - 1$ probabilities are unchanged and hence equal to 2^{-k}. Thus the entropy of the new random variable Z can be bounded from below by

$$\begin{aligned} H(Z) &\geq 2^{-k}(2^k d - \lfloor 2^k d \rfloor) \cdot k + (\lfloor 2^k(1-d) \rfloor - 1) \cdot 2^{-k} \cdot k \\ &= k \cdot (1 - d - 2^{-k}) \ . \end{aligned}$$

□

For certain values of d equality can hold in (2.6). In particular $H(Z) \geq k \cdot (1 - d(U_k, P_Z))$ is false in general: $H(Z) = 0$ is equivalent to $d(U_k, P_Z) = 1 - 2^{-k} < 1$.

2.4 Conditional Entropy and Mutual Information

2.4.1 Conditional Entropy, Chain Rule, and Fano's Inequality

For random variables X and Y, we have

$$H(XY) \leq H(X) + H(Y) \ , \tag{2.7}$$

with equality if and only if X and Y are statistically independent.

When considering inequality (2.7) it appears natural to interpret the (non-negative) quantity $H(XY) - H(X)$ as the entropy of the random variable Y when X is given.

Definition 2.2. The *conditional entropy of* Y *when given* X is defined as

$$H(Y|X) := H(XY) - H(X) . \tag{2.8}$$

○

Note that in contrast to all previously introduced entropies such as $H(X) = H(P_X)$, $H(XY) = H(P_{XY})$, or $H(Y|X = x) = H(P_{Y|X=x})$, the conditional entropy $H(Y|X)$ is *not* the entropy of a specific probability distribution, but rather the expectation of the entropies $H(Y|X = x)$, i.e.,

$$H(Y|X) = \mathrm{E}_X[H(Y|X = x)] . \tag{2.9}$$

Remark. Often, the conditional entropy if defined by equation (2.9). We have chosen the maybe more natural and intuitive, and equivalent, equality (2.8) as the definition.

Equation (2.8) can be rewritten as

$$H(XY) = H(X) + H(Y|X) .$$

This *chain rule* can be generalized as follows. For random variables $X_1, \ldots, X_N$ and an event $\mathcal{A}$ we have

$$\begin{aligned} H(X_1X_2 \cdots X_N|\mathcal{A}) &= \\ &H(X_1|\mathcal{A}) + H(X_2|X_1, \mathcal{A}) + \cdots + H(X_N|X_1X_2 \cdots X_{N-1}, \mathcal{A}) . \end{aligned}$$

The conditional entropy satisfies

$$0 \leq H(Y|X) \leq H(Y) . \tag{2.10}$$

Equality on the left-hand side of (2.10) holds if and only if Y is determined by X, whereas equality on the right-hand side of (2.10) is equivalent to the fact that X and Y are statistically independent. Lemma 2.4 states *Fano's inequality* which quantifies the fact that if $H(Y|X) > 0$, then Y cannot be guessed from X without error: Fano's inequality (2.11) is a lower bound on the expected guessing error probability, depending on $H(Y|X)$ and $|\mathcal{Y}|$.

Lemma 2.4. *Let*

$$Y \to X \to Y'$$

be a Markov chain. Then

$$H(Y|X) \leq h(\mathrm{P}[Y' \neq Y]) + \mathrm{P}[Y' \neq Y] \cdot \log(|\mathcal{Y}| - 1) . \tag{2.11}$$

2.4.2 Mutual Information and Data Processing

The second inequality of (2.10) is a consequence of inequality (2.7). Note that $H(Y|X = x) > H(Y)$ is possible, as the following example illustrates. Let Y be 100 independent flips of an unfair coin with P["heads"] $= 99.9\%$, and let X be the number of "heads" in the sequence. Then, although of course $H(Y|X) < H(Y)$ holds, we have

$$\begin{aligned} H(Y) &= 100 \cdot h(0.999) \approx 1.14 \\ H(Y|X=50) &= \log\left(\binom{100}{50}\right) \approx 96.4 \ . \end{aligned}$$

Of course the event $\{X = 50\}$ is extremely unlikely.

Informally speaking, the second inequality of (2.10) can be interpreted as the fact that additional knowledge can never increase uncertainty. The quantity

$$I(Y;X) := H(Y) - H(Y|X) = H(X) + H(Y) - H(XY) \geq 0 \quad (2.12)$$

is the amount of information that X gives about Y. The last expression of (2.12) shows that $I(Y;X)$ is symmetric in its arguments, i.e., that

$$I(X;Y) = I(Y;X)$$

holds. The quantity $I(X;Y)$ is called the *mutual information between* X *and* Y. Analogously, one can define $I(X;Y|\mathcal{A}) := H(X|\mathcal{A}) - H(X|Y,\mathcal{A})$ and

$$I(X;Y|Z) := H(X|Z) - H(X|YZ) = \mathrm{E}_Z[I(X;Y|Z=z)] \ ,$$

the *conditional mutual information* between X and Y, given Z.

Let X, Y, and Z be three random variables. Then the fact that

$$X \rightarrow Y \rightarrow Z$$

is a Markov chain is equivalent to

$$I(X;Z|Y) = 0 \ .$$

In this case, we have

$$I(X;Z) \leq I(X;Y) \ . \quad (2.13)$$

Inequality (2.13) is called the *data-processing inequality*.

2.5 Graphical Representation of Information-Theoretic Quantities

Let X and Y be random variables. Then the quantities $H(XY)$, $H(X)$, $H(Y)$, $H(X|Y)$, $H(Y|X)$, and $I(X;Y)$ can be graphically represented as shown in Figure 2.2. The union of all inner regions corresponds to $H(XY)$. The representation has the property that the quantity corresponding to the disjoint union of some regions equals the sum of the quantities corresponding to these partial regions. For a detailed discussion of this measure-theoretic representation of information-theoretic quantities see Yeung's paper [123].

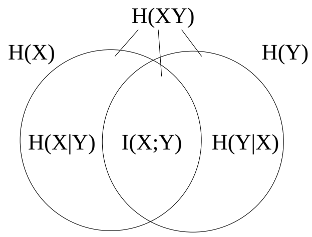


Figure 2.2: Two Random Variables

The case of three random variables is shown in Figure 2.3. Note that the quantity corresponding to the region in the middle,

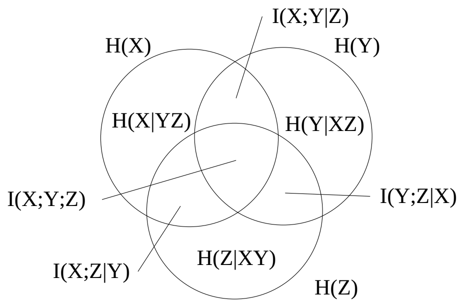


Figure 2.3: Three Random Variables

$$I(X;Y;Z) := I(X;Y) - I(X;Y|Z) ,$$

is symmetric in X, Y, and Z and can be negative. All the other regions represent information-theoretic quantities that are always non-negative.

Figure 2.4 illustrates the case of independent symmetric bits X and Y and $Z := X \oplus Y$. Figure 2.5 shows a Markov chain.

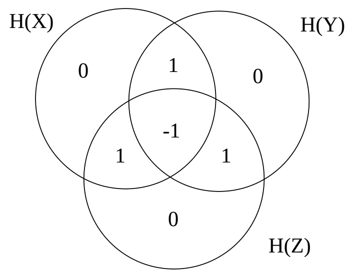


Figure 2.4: $Z = X \oplus Y$

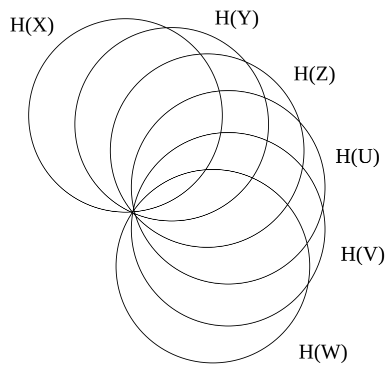


Figure 2.5: A Markov Chain $X \to Y \to Z \to U \to V \to W$

2.6 The Significance of Shannon Entropy and Alternative Entropy Measures

In the previous sections we have described Shannon entropy as the quantity that naturally arises for measuring the uncertainty of a random variable. Although this quantity is just one particular property of a distribution, it has turned out to be an appropriate measure for many problems and scenarios such as *source coding* or *channel coding*. However, Shannon information was shown to be less significant in certain different contexts, where other entropy measures can be more useful.

2.6.1 Source and Channel Coding

A first justification of Shannon's entropy definition is given by *source coding*, i.e., data compression. The outcome of a stochastic process can be compressed to a string of length arbitrarily close to the Shannon entropy of the data (in such a way that error-free decompression is possible), but any compression to a shorter string necessarily leads to decoding errors.

A second example is *channel coding* (error-control coding). Let a discrete memoryless channel (DMC) with input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$ be characterized by the conditional distribution $P_{Y|X}$ (see Figure 2.6). Then the *channel capacity* $C(P_{Y|X})$ is defined as the maxi-

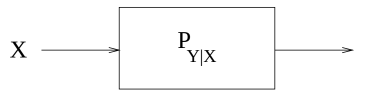


Figure 2.6: A Discrete Memoryless Channel

mal Shannon information that the output can provide about the input, taken over all input distributions or, more precisely,

$$C(P_{Y|X}) = \max_{P_X} (I(X;Y)) \ .$$

Shannon proved that information cannot be transmitted arbitrarily reliably over a channel at a rate exceeding the capacity of the channel, but that, even more surprisingly, data can be transmitted over the channel at a rate smaller than, but arbitrarily close to, the cannel capacity in such a way that the probability of a decoding error is arbitrarily small.

2.6.2 Rényi Entropy and Min-Entropy

In the following chapters, we will see that different entropy measures can be useful in certain contexts such as privacy amplification (i.e., shrinking a partially-random string to a shorter, highly-uniformly-distributed string) or authentication with a partially-secret key. One of the reasons is that although from a possible adversary's viewpoint a string has a substantial amount of uncertainty in terms of Shannon entropy, it is possible that the adversary is able to guess the string correctly with a non-negligible success probability, a fact that is generally unacceptable in the context of secrecy systems. We describe two entropy measures, *Rényi entropy* and *min-entropy*, that are both smaller than Shannon entropy, and that have the property that if the adversary's success probability of guessing the outcome of a random experiment is non-negligible, then its entropy is small.

Let X be a random variable. The *collision probability* $P_C(X)$ of X is defined by

$$P_C(X) := \sum_{x \in \mathcal{X}} P_X(x)^2 \ ,$$

and is the probability that two independent realizations of X show the same outcome. The *Rényi entropy* (or *collision entropy*) of X is defined as

$$H_2(X) := -\log(P_C(X)) \ .$$

The *min-entropy* of X is

$$H_\infty(X) := -\log \max_{x \in \mathcal{X}} (P_X(x)) \ ,$$

i.e., depends only on the maximal probability occurring in P_X. The entropy measures satisfy

$$\frac{H_2(X)}{2} \leq H_\infty(X) \leq H_2(X) \leq H(X)$$

for all random variables X.

It is obvious that a random variable with a substantial amount of Rényi entropy or min-entropy is hard to guess. More precisely, the probability of guessing X correctly is at most

$$2^{-H_\infty(X)}$$

or

$$\sqrt{2^{-H_2(X)}} \ ,$$

respectively. Furthermore, $2^{-H_2(X)}$ is the maximal success probability when guessing X if the probability distribution P_X of X is unknown, but if the outcome of a different realization, independent of the one to be guessed, of X is given.

The measures of Rényi entropy and min-entropy share many properties with Shannon entropy (such as the lower and upper bounds). Indeed, all the mentioned entropy measures can be characterized by the same natural properties one would expect such a measure to have. By varying one particular of these properties, the (uniquely determined) resulting uncertainty measure can be either Shannon entropy, Rényi entropy, or min-entropy, or, more generally, Rényi entropy H_α of order $\alpha \in [0, \infty]$, where H_1 corresponds to Shannon entropy (see [99] or [19] for an axiomatic introduction and a discussion of these general entropy measures, respectively).

However, these alternative entropy measures have some counterintuitive properties. For instance,

$$H_\alpha(X|Y) > H_\alpha(X) \tag{2.14}$$

is possible for $\alpha > 1$ (if the conditional Rényi entropy of order α is defined in analogy to the conditional Shannon entropy). The side information Y is sometimes called *spoiling knowledge* if (2.14) holds (see for example [5] or [19]).

Part II

Unconditionally-Secure Key Agreement

Chapter 3

Introduction and Previous Results

In this chapter we give an introduction to perfect secrecy and unconditionally-secure key agreement based on common information, e.g., from noisy channels. We describe the most important previous results, due to various authors, in a non-technical, introductory manner. Many of the concepts discussed in this chapter are made precise and studied extensively in the subsequent chapters.

Section 3.1 introduces Shannon's concept of perfect secrecy and his pessimistic impracticality theorem. In Sections 3.2 and 3.3 however it is demonstrated that information theory cannot be used only to prove such pessimistic results. It is somewhat surprising that when the models and security requirements are only slightly modified, then practical information-theoretic security can be achieved in many realistic scenarios. The settings of Wyner, Csiszár and Körner, and Maurer are described, in chronological order, which achieve almost-perfect secrecy under realistic conditions such as the existence of noisy channels or somehow correlated information. For Maurer's special "satellite scenario," the steps of a secret-key agreement protocol are informally described. (A detailed analysis of these protocol steps is given in Chapters 4 and 5.)

Parts of this chapter have been published in an overview article on unconditional security [120].

3.1 Perfect Secrecy and Shannon's Pessimistic Theorem

Let us consider the problem of information-theoretically secure key generation and message transmission over an insecure channel. This section contains Shannon's definition of perfect secrecy of a cipher and his well-known theorem which appears to imply that unconditional security is inherently completely impractical.

We start with the classical scenario of a symmetric cryptosystem with message M, key K, and ciphertext C (see Figure 3.1). The fol-

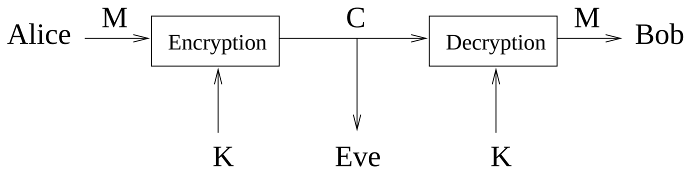


Figure 3.1: A Symmetric Cryptosystem

lowing security definition appears to be the strongest possible for such a cryptosystem.

Definition 3.1. [**106**] A cipher is called *perfectly secret* if the ciphertext reveals no information about the message, i.e., if $I(M;C) = 0$ holds. ○

Equivalent characterizations of this condition are that M and C are statistically independent, or that the best strategy of an eavesdropper who wants to obtain (information about) the message from the ciphertext is to use only the a priori knowledge about M and to discard C.

Perfect secrecy can even be achieved without any computation, as the example in Figure 3.2 shows. One can prove that the ciphertext reveals no information about the message at all in this example. (For more on "visual cryptography," see [89].)

This visual cipher is a graphical implementation of the *one-time pad* that was already proposed by Vernam in 1926 [116]. Here, the message is a string $M = [m_1, m_2, \ldots, m_N]$ of length N, and the key is a uniformly-distributed N-bit string $K = [k_1, k_2, \ldots, k_N]$ which is independent of M. The ciphertext C is computed from M and K by

$$C = [c_1, c_2, \ldots, c_N] = [m_1 \oplus k_1, m_2 \oplus k_2, \ldots, m_N \oplus k_N] =: M \oplus K \ .$$

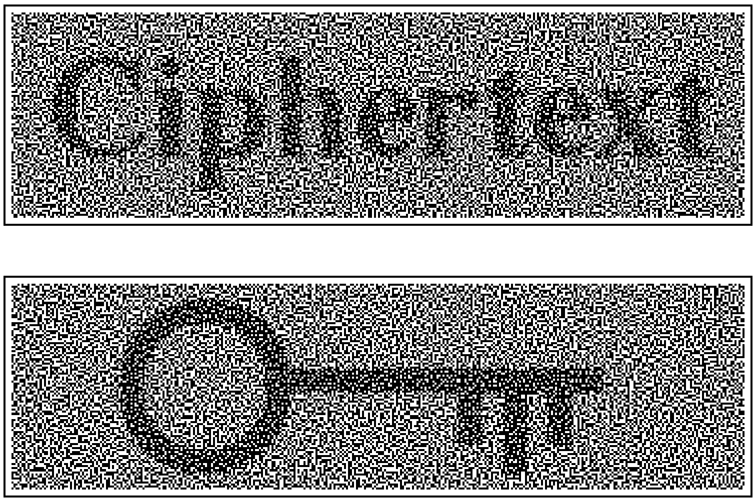


Figure 3.2: Visual Decryption

The one-time pad is perfectly secret. To see this, observe first that when given the cleartext and the ciphertext, then the key is uniquely determined, i.e., $H(K|MC) = 0$. Furthermore, $I(K;C|M) = N$ (remember than N is the block length) follows then from $H(K) = N$ and $I(M;K) = 0$. Finally, $I(M;C) = 0$ holds because $H(C) \leq \log|\mathcal{C}| = N$. A graphical representation of the quantities is given in Figure 3.3.

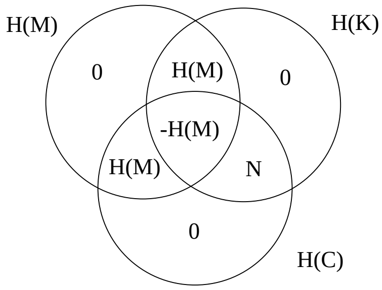


Figure 3.3: Perfect Secrecy of the One-Time Pad

Unfortunately, the price one has to pay here for perfect secrecy is that the communicating parties must share a secret key which is at least as long as the message (and can only be used once). In view of this property, the one-time pad appears to be quite impractical and can only offer an advantage in time: the key can be safely transmitted whenever this is possible, and the message can be secretly sent whenever this is needed.

However, Shannon showed that perfect secrecy cannot be obtained in a cheaper way, i.e., that the one-time pad is optimal with respect to key length.

Theorem 3.1. [**106**] *For every perfectly-secret uniquely-decodable cryptosystem we have*

$$H(K) \geq H(M) \ .$$

For a proof of Shannon's theorem, note first that unique decodability means $H(M|CK) = 0$. The graphic representation of the involved

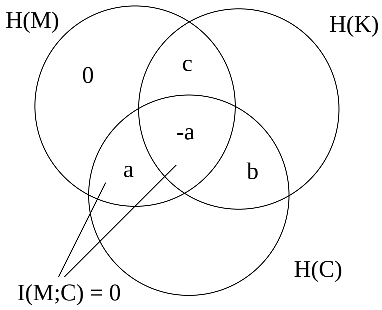


Figure 3.4: A Graphical Proof of Shannon's Theorem [123]

quantities is given in Figure 3.4. We have $b \geq a$ because $I(C;K) \geq 0$, and

$$H(K) \geq b - a + c \geq a - a + c = H(M) \ .$$

This concludes the proof.

3.2 Optimistic Results by Limiting the Adversary's Information

Unfortunately, Shannon's theorem implies that perfect secrecy is possible only between parties who share a secret key of length at least equal to the entropy of the message to be transmitted. Hence every perfectly-secret cipher is necessarily as impractical as the one-time pad. On the other hand, the assumption that the adversary has a *perfect* access to the ciphertext is overly pessimistic and unrealistic in general, since every transmission of a signal over a physical channel is subject to noise.

Motivated by this, a number of models have been presented and analyzed in which the information the adversary obtains is limited in some way, and which offer the possibility of information-theoretically-secure key agreement and, under the assumption that insecure channels are always available, secret message transmission (using the one-time pad with the generated secret key).

The condition that the opponent's knowledge is bounded can for instance be based on noise in communication channels [122], [34], [2], [64], on the fact that the adversary's memory capacity is limited [66], [21], or on the uncertainty principle of quantum mechanics [4], [43]. We describe three models that belong to the first category, starting with the most restrictive one.

3.2.1 Wyner's Wire-Tap Channel

Consider the following (simple but generally unrealistic) situation first. Assume that two parties Alice and Bob are connected by an authentic and noiseless binary channel, and that a wiretapper Eve receives the bits sent over the channel with some error probability $\varepsilon > 0$. In other words, her wire-tap channel is a *binary-symmetric channel (BSC)* with error probability ε (see Figure 3.5).

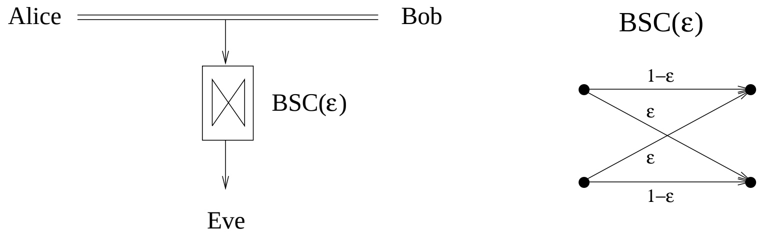


Figure 3.5: A Binary-Symmetric Wire-Tap Scenario

In this situation, Alice can send a message bit M to Bob by sending an N-bit block $[X_1, X_2, \ldots, X_N]$, where $X_1, X_2, \ldots, X_{N-1}$ are independent and symmetric bits and X_N is such that

$$X_1 \oplus X_2 \oplus \cdots \oplus X_N = M \ .$$

Eve's error probability when guessing the bit M with the optimal strategy is

$$p = \frac{1-(1-2\varepsilon)^N}{2} \ ,$$

and converges to $1/2$ exponentially fast in N. Moreover, the information that Eve obtains about M from the noisy versions of $X_1, X_2, \ldots, X_N$ does not exceed $1 - h(p)$. By repeating this process, Alice and Bob can agree on a highly-secret key of arbitrary length.

The following, more general scenario of the *wire-tap channel* (see Figure 3.6) was introduced and analyzed by Wyner [122] and simplified by Massey [58]. In this setting, Alice and Bob are connected by a

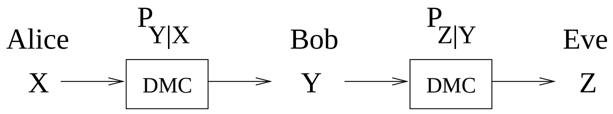


Figure 3.6: Wyner's Wire-Tap Channel

discrete memoryless channel (characterized by its conditional probability distribution $P_{Y|X}$), whereas Eve receives a noisy version Z of Bob's channel output Y (i.e., $X \to Y \to Z$ is a Markov chain). Alice chooses the input to the first channel according to some distribution P_X.

It was shown in [122] that in this scenario, Alice and Bob can agree on a highly-secret key at some rate in many situations (for instance in the case where all the random variables are binary and the channels are binary-symmetric with error probabilities not $1/2$ and not 0 nor 1, respectively). Exact definitions of the security requirements to such a key, as well as of the secret-key generation rate, are given in Chapter 4.

However, the assumption that the adversary only receives a degraded version of the legitimate receiver's information is unrealistic in general. This fact motivated the study of generalizations of Wyner's model.

3.2.2 Broadcast Channels

Csiszár and Körner [34] considered the situation where the sender Alice is connected to the receiver Bob by a discrete memoryless channel (with conditional distribution $P_{Y|X}$), and where also the adversary Eve receives a noisy version Z of X over a different channel (characterized by $P_{Z|XY}$, i.e., the channels are not necessarily independent). As before, Alice chooses the channels' input X according to some distribution P_X. The broadcast scenario is illustrated in Figure 3.7.

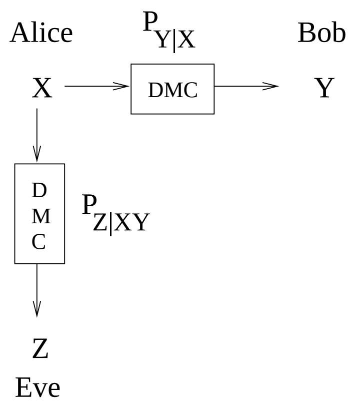


Figure 3.7: Csiszár and Körner's Broadcast-Channel Scenario

For this setting, the *secrecy capacity* $C_S(P_{YZ|X})$ has been defined as the maximal rate at which Alice and Bob can generate a virtually-secret key. For a precise definition and discussion of this quantity, see Section 4.3. We just remark here that both the size of the generated secret key as well as the amount of information leaked to the adversary are defined in terms of a *rate*, i.e., measured as average information *per channel use*.

In [34] it was shown that if the channel connecting the legitimate partners is less noisy than the adversary's channel (in some sense to be specified, see Section 4.3), then this advantage can be exploited for generating a secret key. However, if Alice and Bob have no such advantage to start with, then no secret-key agreement is possible in this model. Let us for instance consider the situation where the channels are independent and binary-symmetric with error probabilities ε and δ (see Figure 3.8). In this special scenario, the secrecy capacity was shown to be

$$C_S(\varepsilon, \delta) = \begin{cases} h(\delta) - h(\varepsilon) & \delta > \varepsilon \\ 0 & \text{otherwise} . \end{cases}$$

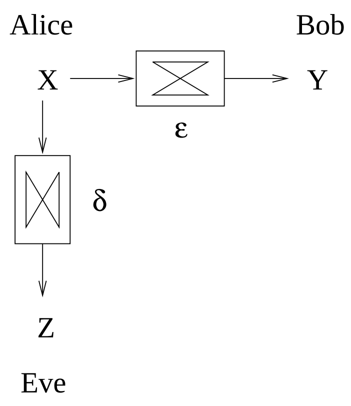


Figure 3.8: Independent Binary-Symmetric Broadcast Channels

In other words, secret-key agreement is impossible unless Bob's channel is superior to Eve's. Unfortunately, it may often be impossible to guarantee that the adversary's channel is noisier than the one of the legitimate partner.

3.2.3 The Power of Interaction

The following example, given in [64], illustrates how much more powerful interaction can be in contrast to one-way transmission for unconditionally-secure key agreement. This is a motivation for considering a more general model of secret-key agreement from common information by insecure noiseless two-way communication.

We start with the situation as shown in Figure 3.8, where $0 < \delta \leq \varepsilon < 1/2$. As mentioned above, no secret-key agreement is possible. However, let us assume an *interactive* variant of this model with an additional noiseless and insecure but authentic channel. (Note that channels with virtually these properties often exist in reality, e.g., telephone lines.) Surprisingly, the situation is now entirely different although the additional channel can be completely overheard by Eve.

Observe first that the additional public-discussion channel allows to invert the direction of the noisy channel between Alice and Bob by the following trick. First, Alice chooses a random bit X and sends it over the noisy channel(s). This bit is received by Bob as Y and by Eve as Z. Bob, who wants to send the message bit C to Alice, computes $C \oplus Y$ and sends this over the noiseless public channel. Alice computes $(C \oplus Y) \oplus X$, whereas Eve can compute $(C \oplus Y) \oplus Z$. This perfectly corresponds to the situation where the direction of the main channel is inverted (see Figure 3.9).

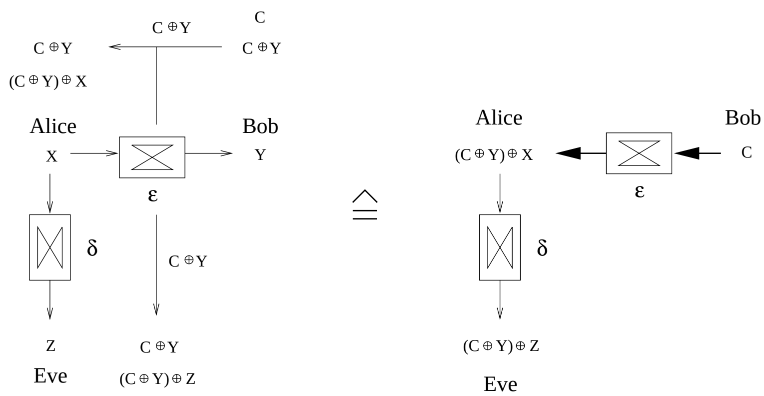


Figure 3.9: Inverting the Main Channel

The second crucial observation is that this is exactly the binary-symmetric setting of Wyner's wire-tap channel of Section 3.2.1, allowing secret-key agreement at some rate. We conclude from this example that the possibility of feedback from Bob to Alice can substantially improve the legitimate partners' situation towards a wire-tapping adversary.

3.3 Interactive Secret-Key Agreement from Correlated Randomness

3.3.1 The General Scenario

Maurer [64] and subsequently Ahlswede and Csiszár [2] have proposed the following interactive model of secret-key agreement by public discussion from common information. The parties Alice and Bob who want to establish a mutual secret key have access to realizations of random variables X and Y, respectively, whereas the adversary knows a random variable Z. Let P_{XYZ} be the joint distribution of the random variables. Furthermore, the legitimate partners are connected by an insecure but authentic channel, i.e., a channel that can be passively overheard by Eve but over which no undetected active attacks by the opponent, such as modifying or inserting messages, are possible (see Figure 3.10).

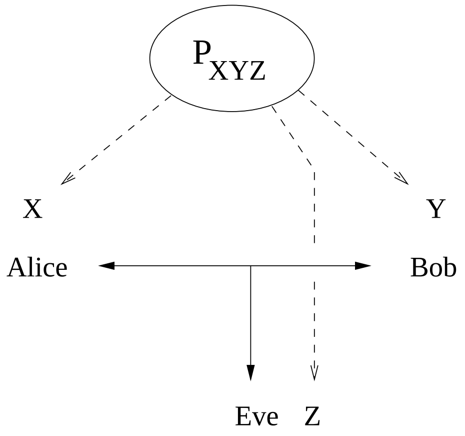


Figure 3.10: Secret-Key Agreement by Public Discussion from Common Information

Note that it is natural to consider this model for the following reasons. First, it is an interactive (i.e., two-way) generalization of Wyner's and Csiszár and Körner's models. It is not necessary to assume the existence of noisy communication channels in this interactive setting because equivalents of such channels can be obtained by the same trick as shown in Section 3.2.3 for inverting the binary-symmetric channel. Secondly, the assumption that the parties have access to correlated randomness appears to be realistic in many contexts. An example of a possible physical implementation is described in Section 3.3.2.

In analogy to the previous models, where the channels could be used many times independently, it can be assumed here that the parties have access to a number of independent realizations of the corresponding random variables. Consequently, the so-called *secret-key rate* has been defined in this model as the maximal rate at which Alice and Bob can generate a highly-secret key by communication over the insecure channel, where the required number of channel uses from the definition of the secrecy capacity is replaced by the amount of randomness (i.e., the number of realizations of X and Y) necessary for the generation of a key of some length.

The secret-key rate is a quite fundamental and mathematically interesting property of a distribution P_{XYZ}. One challenging problem in this context is to enlighten the exact relationship between P_{XYZ} and $S(X;Y||Z)$, i.e., to determine the secret-key rate of a given distribution, or at least to decide whether the rate is nonzero and secret-key agreement is possible in principle in a particular situation. We discuss the definition of the secret-key rate and the mentioned questions in Chapters 4 and 5.

3.3.2 The Satellite Scenario and Phases of Secret-Key-Agreement Protocols

The following realistic special scenario was proposed in [64] and completely analyzed in [77]. Assume that a satellite sends out random bits at very low signal power and that Alice, Bob, and Eve receive these bits over independent binary-symmetric channels with error probabilities α, β, and ε, respectively (see Figure 3.11).

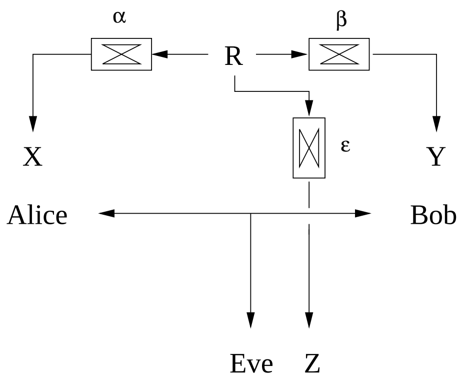


Figure 3.11: The Satellite Scenario

In general, we may have to assume that Eve has a better antenna than the legitimate partners, and hence a possibly substantially lower error rate. It is a somewhat surprising fact that secret-key agreement is always possible in this scenario (unless Eve has a noiseless access to the satellite bits or either Alice or Bob obtains no information at all about these bits).

In the following, we sketch a protocol for secret-key agreement in the satellite scenario. Such a protocol is often interpreted as consisting of three phases. As mentioned, Alice and Bob possibly start in a situation in which the adversary has an advantage over the legitimate partners with respect to the information about each other's random variables. The objective of the first phase, *advantage distillation*, is to generate an advantage over the opponent by exploiting the authenticity of the public channel. However, Alice and Bob do generally not share a mutual string after this phase. Hence, a (possibly interactive) error-correction phase, *information reconciliation*, is required. Finally, the resulting mutual but only partially-secret string must be transformed into a (shorter) highly-secret string. This final phase is called *privacy amplification*. In the illustration of the three phases in Figure 3.12, the relations between the amounts of information that Bob's and Eve's knowledge provide about Alice's string are shown. The protocol steps are sketched in the next three sections. (An interactive demonstration of the phases is provided on the Internet [15].)

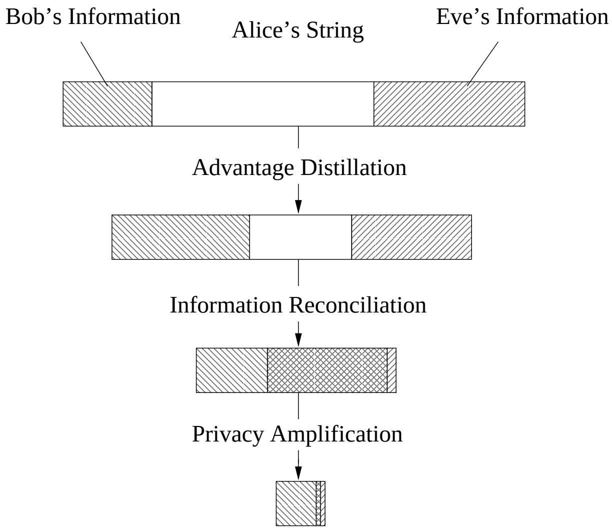


Figure 3.12: Phases of a Secret-Key-Agreement Protocol

3.3.3 Advantage Distillation

Let us assume the satellite scenario described in the previous section with error probabilities $0 < \alpha, \beta < 1/2$ and $0 < \varepsilon < \min\{\alpha, \beta\}$, i.e., the adversary has an initial advantage over the legitimate partners in terms of the error probabilities. We consider N independent realizations of the random variables. Then

$$\begin{aligned} I(X^N;Y^N) &= \sum_{i=1}^{N} I(X_i\,;\,Y_i) \;=\; N \cdot (1 - h(\alpha(1-\beta) + (1-\alpha)\beta))\ , \\ I(X^N;Z^N) &= \sum_{i=1}^{N} I(X_i\,;\,Z_i) \;=\; N \cdot (1 - h(\alpha(1-\varepsilon) + (1-\alpha)\varepsilon))\ , \\ I(Y^N;Z^N) &= \sum_{i=1}^{N} I(Y_i\,;\,Z_i) \;=\; N \cdot (1 - h(\beta(1-\varepsilon) + (1-\beta)\varepsilon)) \end{aligned}$$

hold, i.e.,

$$I(X^N;Y^N) < \min\{I(X^N;Z^N)\,,\,I(Y^N;Z^N)\}\ .$$

The basic idea of the advantage-distillation phase is that Alice and Bob use the noiseless discussion channel for exchanging information about their bits in an insecure but authentic way with the objective of identifying bits that are correct with a higher probability than others.

The following protocol is based on the exchange of parity-check bits and is more efficient than a simpler repeat-code protocol which we use in Chapter 5 for proving some existence results. The parity-check protocol was presented in [65] (see also [40]).

Parity-Check Protocol. Alice computes the parity bit $X_1 \oplus X_2$ and sends it over the public channel. Bob accepts if and only if

$$X_1 \oplus X_2 = Y_1 \oplus Y_2\ ,$$

i.e., if the parities of Alice's and Bob's first two bits are equal. In this case, the values X_1 and Y_1 are chosen by Alice and Bob, respectively, for the next protocol round (whereas otherwise, the bits are discarded). This step is repeated a number of times. After this first round it may be necessary, depending on the initial error probabilities, to carry out additional rounds (see Figure 3.13).

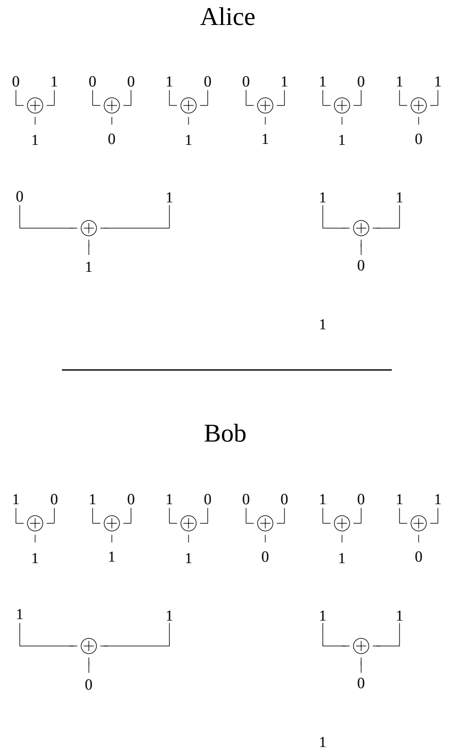


Figure 3.13: Two Rounds of the Parity-Check Protocol

3.3.4 Information Reconciliation

During advantage distillation, the partners Alice and Bob compute (possibly distinct) strings S_A and S_B, respectively, about which the adversary also has some information. At the end of the key-agreement protocol however, Alice's and Bob's strings must be equal and highly secure, both with overwhelming probability. The phase of information-reconciliation consists of interactive error correction and establishes the first of these two conditions.

After advantage distillation, Bob has more information about Alice's string than Eve has, and after information reconciliation, Bob should exactly know Alice's string. (A more general condition would be that after information reconciliation, Alice and Bob share a string that is equally long as S_A and S_B.) This leads to a lower bound on the amount of error-correction information E that must be exchanged. Namely, since Bob must know S_A completely with overwhelming probability when given S_B and E, we have

$$0 \approx H(S_A \,|\, S_B, E) \geq H(S_A \,|\, S_B) - H(E) \ ,$$

and hence

$$H(E) \gtrsim H(S_A \,|\, S_B) \ .$$

On the other hand, the uncertainty about S_A from Eve's viewpoint can as well be reduced by $H(E)$ in the worst case when Eve learns E (see Figure 3.14).

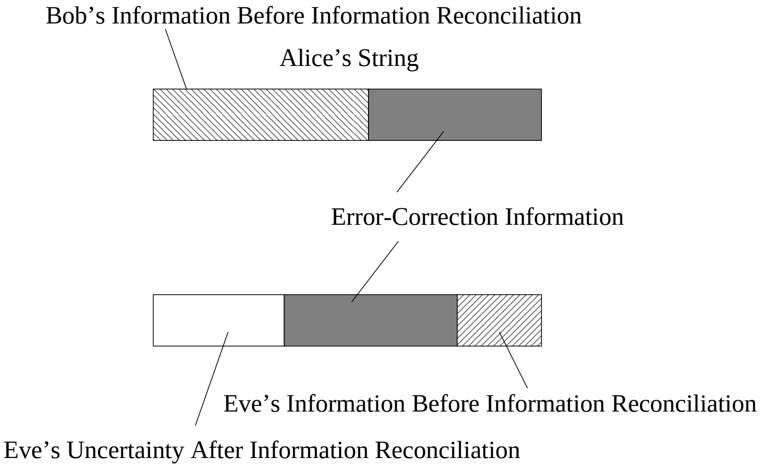


Figure 3.14: The Effect of Information Leaked During Information Reconciliation

A good protocol for information reconciliation should both minimize the amount of information leaked to the adversary and be efficient. Examples of protocols satisfying both conditions are given in [14]. We sketch two examples of these protocols. The first is optimal with respect to the information leaked but completely inefficient, whereas the second protocol leaks more information but is more efficient as well. An analysis of a variant of the first protocol is provided in Sections 4.2.1 and 4.2.2.

Let us assume that Alice and Bob have finished the advantage-distillation phase in the satellite model. This means that Bob's string is a (good) estimate about Alice's string, i.e., the same string with a (small) number of errors.

Random-Label Protocol. [**14**] The first, non-interactive, protocol works as follows. Alice randomly chooses a function f mapping $\{0,1\}^n \to \{0,1\}^m$ (where n is the length of S_A and S_B and m can be roughly equal to $H(S_A|S_B)$) among all such functions and sends (a description of) f together with $f(S_A)$ to Bob, who determines the string S'_A with minimal Hamming distance from S_B that satisfies $f(S'_A) = f(S_A)$. According to the discussion above, and because $m \approx H(S_A|S_B)$, this protocol is optimal with respect to the leaked information. However, it is completely inefficient, hence useless, for the following two reasons. First, the description of the random function f would require $m2^n$ bits. Furthermore, S'_A cannot be efficiently determined from f, $f(S_A)$, and S_B.

Binary-Search Protocol. [**14**] The idea of the second protocol is to interactively detect the positions where Alice's and Bob's strings differ and to correct these errors. Alice and Bob start by comparing the parity bit, i.e., the XOR-sum, of the bits in randomly but identically-chosen substrings S'_A and S'_B of S_A and S_B, respectively. If there are bit errors between the strings S_A and S_B, then the resulting parity bits differ with probability $1/2$ over the choice of the substrings.

If the parities are different, Alice and Bob have detected substrings containing an odd number of errors with respect to each other, and they can locate one of them by partitioning the substring into two subsets of equal size, one of which clearly contains an odd number of errors as well (and has different parity sums). This splitting procedure is continued until the error is localized and can be corrected by Bob (see Figure 3.15).

Alice and Bob repeat this procedure until all the errors are found and corrected. If n is the length of the strings S_A and S_B, this protocol requires the exchange of $\lceil \log n \rceil$ bits per error to be corrected. Hence it is efficient if the strings of Alice and Bob differ in only a few bit positions.

After information reconciliation, Alice and Bob have agreed on a mutual string S about which Eve has a possibly considerable amount of information consisting of both a priori knowledge but also information leaked during information reconciliation.

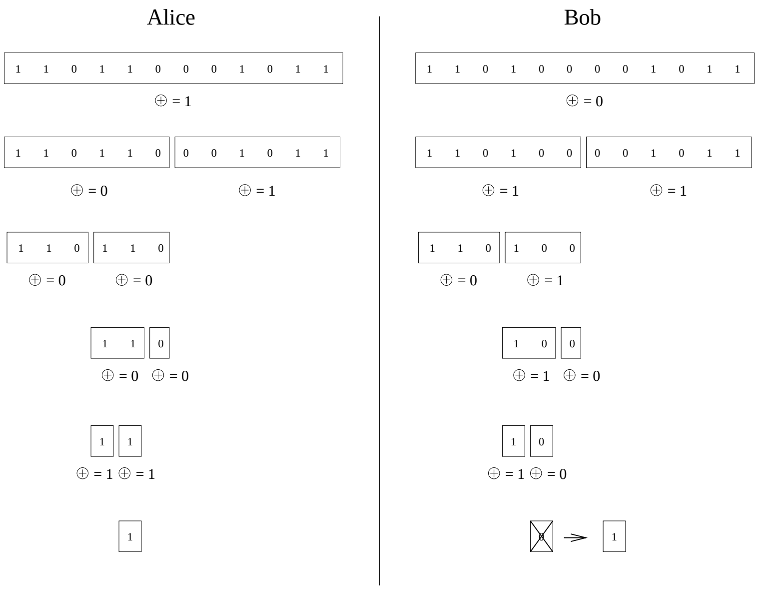


Figure 3.15: Finding and Correcting an Error by Comparing Parities

3.3.5 Privacy Amplification

Privacy amplification is the art of shrinking a partially-secure string S to a highly-secret string S' by public discussion. Hereby, the information of the adversary about S can consist of *physical bits* (i.e., particular binary digits of the string), of parities thereof, or other types of information (see Figure 3.16).

The following questions related to privacy amplification were studied and answered in [6], [5]:

- What is a good technique of computing S' from S?
- What is the possible length of S', depending on the shrinking technique and on the adversary's (type and amount of) information about S?

It is quite clear that the best technique would be to compute S' (of length r) from the n-bit string S by applying a random function

$$f : \{0,1\}^n \to \{0,1\}^r .$$

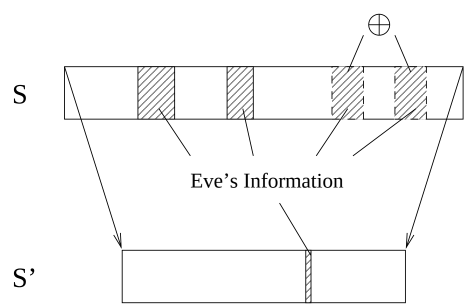


Figure 3.16: Eliminating Eve's Knowledge by Privacy Amplification

However, Alice and Bob would have to exchange $r2^n$ bits of information to agree on such a function. On the other hand, there exist relatively small classes of functions with "random-like" properties. Examples are so-called *universal classes* of hash functions or *extractors* (see Sections 4.2.1 and 4.3.2).

All the proposed protocols for privacy amplification are of the form that Alice chooses a suitable number of random bits and sends them publicly to Bob. From these bits, both Alice and Bob then compute the compressed string. The central question in this context is how long the virtually-secure string S' can be, depending on the hashing technique and on the type and amount of Eve's knowledge about S. Note first that the fact that Eve has some information about a string S is another way of saying that given Eve's entire knowledge $U = u$ about S, the random variable S is *not* uniformly distributed, i.e.,

$$H(S \,|\, U = u) < n \ ,$$

and Eve has $n - H(S|U = u)$ bits of (Shannon-) information about S. Because the resulting string S' must satisfy

$$H(S' \,|\, C, U = u) \approx r$$

(where r is the length of S' and C is the communication held over the public channel), privacy amplification can be interpreted as "distribution smoothing."

Intuitively, one might think that if Eve has t bits of information about S, then the length r of the resulting string S' can be roughly $n - t$ (see Figure 3.17). This fact was shown to be correct if Eve has *deterministic*

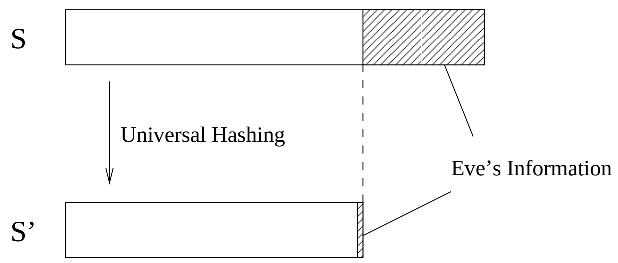


Figure 3.17: Can Eve's Knowledge Be Simply Cut Away by Universal Hashing?

information about S, i.e., if Eve knows the value $g(S)$ for some fixed function g [6]. However, if Eve's information is not deterministic, it is *not* true in general that $n-t$ secure bits can be extracted when Eve has t bits of Shannon information about S, as the following example shows. Let $P_{S|U=u}(s_0) = 1/2$ for some $s_0 \in \{0,1\}^n$, and $P_{S|U=u}(s) = 1/(2\cdot(2^n-1))$ for all n-bit strings $s \neq s_0$. Then, we have $H(S|U = u) \approx n/2$, but no secure string S' (of any length, let alone $n/2$) can be extracted because Eve precisely knows S, hence also $S' = h(S)$, with probability $1/2$ (where h is the randomly-chosen hash function). This means that S' cannot be highly secure.

It was shown in [5] (see also [46]) that when universal hashing is used for privacy amplification, then the Rényi entropy $H_2(S|U = u)$ is the right quantity for measuring the adversary's information or, equivalently, uncertainty (see Section 4.2.1), whereas in the context of extractors, the (generally smaller) min-entropy $H_\infty(S|U = u)$ must be used (see Section 4.3.2).

Chapter 4

The Model of Information-Theoretic Secret-Key Agreement

The notion of a key-agreement protocol and of the secret-key rate in the model of secret-key agreement by public discussion from common information, described already in the previous chapter, were defined by Maurer [64]. In Section 4.1 we state these definitions as we will use them in the subsequent chapters. In particular, a stronger notion of secret-key rate than the one of [64] is used that corresponds to the definition given in [63]. We show that the secret-key rates with respect to the weaker and stronger definitions are always equal. A part of the given proof is a simplified and completed version of the proof sketch of a weaker, but related result in [63]. In addition, a similar statement is proved in Section 4.3 for the secrecy capacity in Wyner's [122] and Csiszár and Körner's [34] models. It is shown, by using recent results on extractors, that a secret key can be generated at the same rate in such a way that the adversary obtains, in total, only an arbitrarily-small amount of information about the key.

4.1 Definitions of Secret-Key Agreement and Secret-Key Rate

Assume the following situation. Two parties Alice and Bob have access to a realization of the random variables X and Y, respectively, whereas an adversary Eve knows the outcome of a random variable Z. Let P_{XYZ} be the joint distribution of the three random variables. Furthermore, the legitimate partners Alice and Bob are connected by a noiseless and authentic but otherwise insecure channel. Alternatively, the adversary Eve can be assumed to be a *passive* wire-tapper. It is the objective of Alice and Bob to generate a provably-virtually-secret key in this situation. The scenario is illustrated in Figure 3.10.

Definition 4.1. A $(P_{XYZ}, r, \varepsilon)$*-protocol for secret-key agreement* consists of two phases: a *communication phase* and a *key-generation phase.*

During the *communication phase*, Alice and Bob exchange messages $C_1, C_2, \ldots$ over the public channel. We assume here that Alice sends the messages $(C_1, C_3, \ldots)$, whereas Bob sends $(C_2, C_4, \ldots)$. A message C_i sent at some point in the protocol can (only) depend on the sender's knowledge when sending the message and possibly some (locally-generated) random bits R_A or R_B, respectively. More precisely, we have for odd i that

$$H(C_i \,|\, X, C_1 \cdots C_{i-1}, R_A) = 0 \ , \tag{4.1}$$

whereas for even i

$$H(C_i \,|\, Y, C_1 \cdots C_{i-1}, R_B) = 0 \tag{4.2}$$

holds. Assume that t messages are exchanged during the communication phase, and let

$$C := [C_1, C_2, \ldots, C_t]$$

be the collection of all messages sent over the public channel.

In the subsequent *key-generation phase* of the protocol, Alice and Bob compute keys S and S', respectively. It is their goal to minimize both the probability that $S \neq S'$ holds and the information (in terms of Shannon entropy) that the adversary obtains about S. More precisely, we have

$$\begin{aligned} H(S \,|\, CXR_A) &= 0 \,, \\ H(S' \,|\, CYR_B) &= 0 \,, \\ H(S) &\geq r \,, \\ \mathrm{P}[S \neq S'] &\leq \varepsilon \,, \\ I(S; ZC) &\leq \varepsilon \end{aligned}$$

for some independent random bit strings R_A and R_B. ○

Privacy amplification, which was described as the final step of an agreement protocol, can alternatively be seen as the important special case of secret-key agreement for which $X = Y$ holds with probability 1. Informal and technical descriptions of privacy amplification can be found in Sections 3.3.5 and 4.2.1, respectively.

A quantitative definition of the amount of secret key that can be generated by Alice and Bob has been given for the special scenario where all the parties have access to a large number of *independent repetitions* of the same random experiment, i.e., realizations of the random variables X, Y, and Z. In other words, Alice, Bob, and Eve receive, for some (large) N, $X^N = [X_1, X_2, \ldots, X_N]$, $Y^N = [Y_1, Y_2, \ldots, Y_N]$, and $Z^N = [Z_1, Z_2, \ldots, Z_N]$, respectively, with

$$P_{X^N Y^N Z^N} = P_{XYZ}^N := \prod_{i=1}^{N} P_{X_i Y_i Z_i} \ ,$$

where $P_{X_i Y_i Z_i} = P_{XYZ}$ for all $1 \leq i \leq N$. In this scenario it is natural to define and quantify the ability of generating a secret key asymptotically in terms of a *rate*, the so-called secret-key rate.

Definition 4.2. Let P_{XYZ} be the joint distribution of the random variables X, Y, and Z. The *secret-key rate* $S(X;Y||Z)$ of X and Y with respect to Z is the least upper bound of the set of real numbers $R \geq 0$ such that for every $\varepsilon > 0$ and sufficiently large $N = N(\varepsilon)$, there exists a $(P_{XYZ}^N, (R-\varepsilon)N, \varepsilon)$-protocol for secret-key agreement such that the key additionally satisfies a uniformity condition. More precisely, Alice and Bob can compute strings S and S', respectively, such that

$$\begin{aligned} H(S)/N &\geq R - \varepsilon \, , \\ \mathrm{P}[S \neq S'] &\leq \varepsilon \, , \\ I(S; Z^N C) &\leq \varepsilon \, , && (4.3) \\ H(S) &\geq \log |\mathcal{S}| - \varepsilon && (4.4) \end{aligned}$$

hold. ○

Remark. In Definition 4.2 one could alternatively demand that the highly-secret key S is even *perfectly*-uniformly distributed, and that Alice and Bob have strings S_A and S_B which are equal to S with high probability. An advantage of this definition would be that it is symmetric (with respect to Alice's and Bob's key). Such a definition was used in Csiszár and Körner's model [34] for the secrecy capacity C_S (see Definition 4.5). It can be shown by a trick similar to the one used in the proof of Theorem 4.12 that the two definitions are equivalent, i.e., that the resulting rates are equal.

The value of the secret-key rate is an intrinsic property of the joint distribution of three random variables. The precise connection between $S(X;Y||Z)$ and P_{XYZ} has been studied intensively, and will be an important subject in this and the subsequent chapter. The following upper bound on $S(X;Y||Z)$ in terms of P_{XYZ} is very intuitive, and was first shown in [64]. It states that it is not possible to generate a greater amount of secret key from X and Y than the mutual information between X and Y (given Z):

$$S(X;Y||Z) \leq \min \{ \, I(X;Y) \, , \, I(X;Y|Z) \, \} \, . \tag{4.5}$$

This bound will be discussed and improved in Chapter 5.

4.2 Discussion of the Definition of the Secret-Key Rate

In contrast to earlier definitions of the secret-key rate (e.g., in [64]) and of the secrecy capacity C_S in Wyner's [122] and Csiszár and Körner's [34] models, Definition 4.2 is more restrictive in the sense that not only the *rate* at which Eve obtains information about the secret key, but the *total amount* of this information must be arbitrarily small. Clearly, such a definition is much more satisfactory since the opponent's knowledge about a key that is secret only according to the weaker definition is not necessarily bounded, let alone negligibly small. Additionally, and also in contrast to the definition given in [64], inequality (4.4) of Definition 4.2 requires that the resulting key is almost-uniformly distributed. In this section we show, extending and simplifying results by Maurer [63], that these stronger conditions can be added to or removed from the definition of the secret-key rate without effect on its value. Some of the proofs have already been given or sketched by Maurer [63].

We first define the "weak" secret-key rate $S_w(X;Y||Z)$ in analogy to $S(X;Y||Z)$, but where inequality (4.3) is replaced by the weaker condition

$$\frac{1}{N}I(S;Z^N C) \leq \varepsilon ,$$

and where the uniformity condition (4.4) is dropped completely. We show that

$$S_w(X;Y||Z) = S(X;Y||Z) .$$

We first give technical descriptions and analyses of certain realizations of the protocol phases information reconciliation and privacy amplification, which allow secret-key agreement by one-way communication in the case where the random variable Y provides more information about Alice's information X than Eve's random variable Z (Section 4.2.1). Note that advantage distillation is not needed here because Alice and Bob already start with an advantage over the opponent in terms of information about each other's knowledge. The analysis of the steps of this particular protocol for secret-key agreement leads to a lower bound on $S(X;Y||Z)$ in terms of P_{XYZ} (Section 4.2.3). Finally, the mentioned results and techniques are used in Section 4.2.4 to prove that the weak and strong rates are equal. Most of the proofs in this section are based on so-called *typical sequences*.

4.2.1 Information Reconciliation and Privacy Amplification

In this section we analyze the two steps of a protocol allowing secret-key agreement whenever $I(X;Y) - I(X;Z) > 0$ or $I(Y;X) - I(Y;Z) > 0$

holds. In particular, we show that

$$S(X;Y||Z) \geq \max\{\, I(X;Y) - I(X;Z)\,,\, I(Y;X) - I(Y;Z)\,\}\ . \quad (4.6)$$

It should be pointed out again, and will be extensively studied in Chapter 5, that secret-key agreement can also be possible in situations for which the right-hand side of inequality (4.6) is zero or negative. However, the advantage distillation step, requiring feedback instead of only one-way communication, must be used in this case (see Section 3.3).

Assume now that $I(X;Y) > I(X;Z)$. The information-reconciliation phase consists of the following step (which is similar to the random-label protocol described in Section 3.3.4). Alice sends $h(X^N)$ to Bob, where $h : \mathcal{X}^N \to \{0,1\}^L$ is supposed to provide Bob (who knows Y^N) with a sufficient amount of redundant information about X^N, allowing him to reconstruct X^N with high probability. The existence of such a function for L on the order of $N \cdot H(X|Y)$ is stated in Theorem 4.1. The proof of Theorem 4.1 (which was stated already in [63] without a proof) will be given in Section 4.2.2 as a first application of the typical-sequences arguments.

Theorem 4.1. *Let X and Y be random variables with joint distribution P_{XY}, and let $[(X_1,Y_1),\ldots,(X_N,Y_N)]$ be a block of N independent realizations of X and Y. Then for every $\varepsilon > 0$ and $\varepsilon' > 0$, for sufficiently large N, and for every L satisfying $L/N > (1+\varepsilon)H(X|Y)$, there exists a function $h : \mathcal{X}^N \to \{0,1\}^L$ such that $[X_1,\ldots,X_N]$ can be decoded from $[Y_1,\ldots,Y_N]$ and $h(X^N)$ with error probability at most ε'.*

In the second protocol phase, privacy amplification, Alice and Bob compress the now shared string X^N, in a manner known to Eve, to result in a shorter M-bit string S with virtually-uniform distribution, and about which Eve has essentially no information. Of course, this privacy-amplification step must take into account Eve's total information about X^N consisting of Z^N and $h(X^N)$.

The privacy-amplification phase was informally described already in Section 3.3.5. Assume Alice and Bob share an N-bit string w (e.g., the specific value of the random variable X^N) about which an eavesdropper Eve has incomplete information characterized by a probability distribution P_W over the N-bit strings, and that Alice and Bob have some knowledge of this distribution P_W, but that they do not know exactly what is compromised about the secrecy of their string. Using the public-discussion channel they wish to agree on a function

$g : \{0,1\}^N \to \{0,1\}^M$ (for some suitable M) such that Eve, despite her partial knowledge about w and complete knowledge of g, almost certainly knows nearly nothing about $g(w)$. This process transforms a partially-secret N-bit string w into a highly secret but shorter M-bit string $g(w)$.

As mentioned in Section 3.3.5, the two natural questions in this context are what a good technique is for computing the shorter from the initial string, and how long the virtually-secret string can be, depending on this technique and on P_W. Bennett, Brassard, and Robert [6] considered the case where Eve receives *deterministic* information, i.e., where the key is, in Eve's view, uniformly distributed over a subset of the set of all possible keys. They used universal hashing as the technique for compressing the string.

Definition 4.3. [26] A class G of functions $g : \mathcal{A} \longrightarrow \mathcal{B}$ is *universal*$_2$ ("universal" for short) if, for any distinct x_1 and x_2 in $\mathcal{A}$, the probability that $g(x_1) = g(x_2)$ holds is at most $1/|\mathcal{B}|$ when g is chosen at random from G according to the uniform distribution. ○

The following is an example of a universal class of functions from $\{0,1\}^N$ to $\{0,1\}^M$, for $M \leq N$, with 2^N elements.

Example 4.1. Let a be an element of $GF(2^N)$, and interpret $x \in \{0,1\}^N$ as an element of $GF(2^N)$ with respect to a fixed basis of the extension field over the prime field $GF(2)$. Consider the function $h_a : \{0,1\}^N \to \{0,1\}^M$ assigning to an argument x the first M bits (with respect to this basis representation) of the element ax of $GF(2^N)$, i.e.,

$$h_a(x) := \mathrm{LSB}_M(a \cdot x) \ .$$

The class

$$\{h_a \,|\, a \in GF(2^N)\}$$

is a universal class of functions for $1 \leq M \leq N$. ◇

The special results of [6] were extended by Bennett, Brassard, Crépeau, and Maurer [5] to scenarios in which Eve's information about w is not deterministic, but where the probability distribution P_W satisfies a constraint in terms of Rényi entropy. The main result of [5] is the following theorem (see also Figure 4.1). A closely related result was shown in [46].

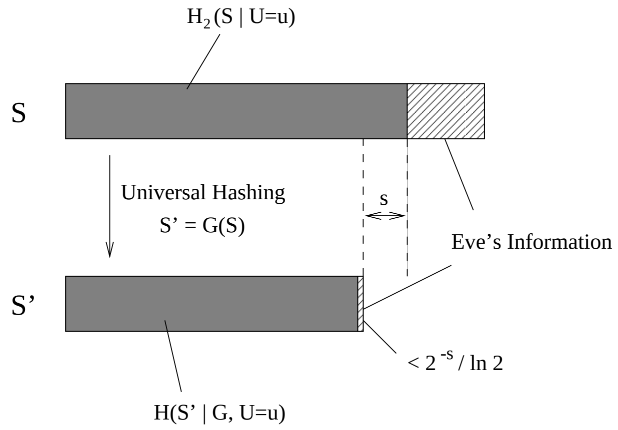


Figure 4.1: Universal Hashing Allows to Extract Rényi Entropy

Theorem 4.2. [**5**] *Let P_W be a probability distribution over $\mathcal{W}$ with Rényi entropy $H_2(W)$, and let G be the random variable corresponding to the random choice, with respect to the uniform distribution, of an element of a universal class of functions mapping $\mathcal{W}$ to $\{0,1\}^M$. Then*

$$H(G(W)|G) \geq H_2(G(W)|G) \geq M - \frac{2^{M-H_2(W)}}{\ln 2} \; .$$

Theorem 4.2 states that if Alice and Bob share a particular string S and Eve's information about S corresponds to the distribution $P_{S|Z=z}$ (where z denotes the particular value of her information Z) about which Alice and Bob know nothing except a lower bound R on the Rényi entropy, i.e., $H_2(S|Z=z) \geq R$, then Alice and Bob can generate a secret key S' of roughly R bits. More precisely, if Alice and Bob compress S slightly more to an $(R-s)$-bit key for some security parameter $s > 0$, then Eve's total information about this key decreases exponentially in s.

A natural problem that arises when combining information reconciliation and privacy amplification with universal hashing is to determine the effect of the error-correction information (leaked also to the adversary) on the Rényi entropy of the partially-secret string, given Eve's information. The following result, which was shown by Cachin [19] as an improvement of an earlier result by Cachin and Maurer [22], states that leaking t physical bits of arbitrary side information about a random variable cannot reduce its Rényi entropy by substantially more than t but with exponentially-small probability. A generalized version of this result is stated and proved in Chapter 6 as Lemma 6.15.

Lemma 4.3. [19] *Let X and Q be random variables, and let $s > 0$. Then with probability at least $1 - 2^{-(s/2-1)}$ (taken over $q \in \mathcal{Q}$), we have*

$$H_2(X) - H_2(X|Q = q) \leq \log |\mathcal{Q}| + s \ .$$

4.2.2 Typical Sequences

In the following, we will make use of so-called *typical-sequences arguments*. Such arguments are based on the fact that if a large number of independent realizations of a random variable U is considered, then the actual probability of the particular outcome sequence is, with overwhelming probability, close to a certain "typical probability." There exist various definitions of typical sequences. The definition given below corresponds to a weak notion of typicality, dealing only with probabilities and not with the number of occurrences of certain outcome symbols of the original random variable U in the sequence.

Definition 4.4. Let U be a random variable with probability distribution P_U and range $\mathcal{U}$. For $N \geq 0$, let $P_{U^N} = P_U^N$ be the distribution of the random variable U^N corresponding to N independent realizations of U. Then a sequence $u = (u_1, u_2, \ldots, u_N) \in \mathcal{U}^N$ is called (weakly) δ-typical if

$$2^{-N(H(U)+\delta)} \leq P_{U^N}(u) \leq 2^{-N(H(U)-\delta)} \ .$$

○

Lemma 4.4 states that if N is large enough, then U^N is δ-typical with high probability. More precisely, the probability of the "non-typicality" event tends to zero faster than $1/N^2$. The lemma follows immediately from Theorem 12.69 in [29].

Lemma 4.4. *For every $\delta > 0$ and $\varepsilon > 0$, we have*

$$N \cdot (\mathrm{P}[U^N \textit{ is } \mathrm{not} \ \delta\textit{-typical}])^{1/2} < \varepsilon$$

for sufficiently large N.

As a first application of typical sequences and of Lemma 4.4, we can now give a proof of Theorem 4.1 from the previous section. The argument given partly follows the lines of the analysis of the random-label protocol in [14].

Proof of Theorem 4.1. It is sufficient to show that the statement is true if the function h is chosen at random from the set $\mathcal{H}$ of all functions mapping $\mathcal{X}^N$ to $\{0,1\}^L$. Then we can conclude from

$$\mathrm{P}_{h\in_r\mathcal{H}\,,\,[x_1,\ldots,x_N]\in\mathcal{X}^N}[\mathcal{E}] = \mathrm{E}_H\left[\mathrm{P}_{[x_1,\ldots,x_N]\in\mathcal{X}^N|H=h}[\mathcal{E}]\right] \leq \varepsilon'$$

(where $\mathcal{E}$ stands for the event of a decoding error when using the optimal strategy) that there exists a particular function $h_0 \in \mathcal{H}$ such that the probability of a decoding error is at most ε', given $H = h_0$.

We can assume $H(X|Y) > 0$ because the statement is trivial otherwise. Let $0 < \alpha < \varepsilon'/2$ and $0 < \delta < \varepsilon H(X|Y)/2$ be constants. For sufficiently large N, we have with probability at least $1 - 2\alpha$ that both

$$2^{-N(H(XY)+\delta)} \leq P_{X^N Y^N}(x^N, y^N) \leq 2^{-N(H(XY)-\delta)} \tag{4.7}$$

and

$$2^{-N(H(Y)+\delta)} \leq P_{Y^N}(y^N) \leq 2^{-N(H(Y)-\delta)} \tag{4.8}$$

hold. From (4.7) and (4.8) one can conclude

$$2^{-N(H(X|Y)+2\delta)} \leq P_{X^N|Y^N}(x^N, y^N) \leq 2^{-N(H(X|Y)-2\delta)} \ . \tag{4.9}$$

For a particular fixed y^N, there are at most

$$2^{N(H(X|Y)+2\delta)}$$

different x^N for which (4.9) can hold. Let $L > (1+\varepsilon)NH(X|Y)$. The probability that for a randomly-chosen function $h : \mathcal{X}^N \to \{0,1\}^L$, and for given y^N and $h(x^N)$, there exists $(x')^N$, different from x^N, satisfying (4.9) and with $h(x^N) = h((x')^N)$, is at most

$$2^{-L} \cdot 2^{N(H(X|Y)+2\delta)} < 2^{-N(\varepsilon H(X|Y)-2\delta)}$$

because of the union bound. This expression is arbitrarily small for sufficiently large N because $\delta < \varepsilon H(X|Y)/2$. From the above we conclude that the decoding-error probability is upper bounded by

$$\mathrm{P}[\mathcal{E}] \leq 2\alpha + 2^{-N(\varepsilon H(X|Y)-2\delta)} \ .$$

This expression is smaller than ε' for sufficiently large N by the definition of α and δ. □

As another application of the typical-sequences technique, and as a further step towards proving equality of the secret-key rates with respect to the different definitions, we show that the weak definition can be extended by an additional condition requiring that the resulting key be close-to-uniformly distributed. More precisely, Lemma 4.5 states that the condition

$$\frac{1}{N}H(S) \geq \frac{1}{N}\log|\mathcal{S}| - \varepsilon \tag{4.10}$$

can be included into the definition of $S_w(X;Y||Z)$ without effect on its value. Note that the condition (4.10) is much weaker than the uniformity condition in the definition of $S(X;Y||Z)$.

Lemma 4.5. *Let the* uniform *weak secret-key rate $S_{w,u}(X;Y||Z)$ be defined similarly to the rate $S_w(X;Y||Z)$, but including the additional condition (4.10). Then*

$$S_{w,u}(X;Y||Z) = S_w(X;Y||Z) \ .$$

Proof. The idea is to carry out the key-generation procedure independently many times and to apply data compression. More precisely, secret-key agreement with respect to the definition of $S_w(X;Y||Z)$ is repeated M times. Clearly, we can assume that the resulting triples $[S_i, S_i', (Z^N C)_i]$ are independent for different values of i and can be considered as the random variables in a new random experiment. When repeating this experiment for a sufficient number of times and applying data compression to the resulting sequence of keys, thereby using that with high probability both $[S_1, S_2, \ldots]$ and $[S_1', S_2', \ldots]$ are typical sequences, one finally obtains key agreement that ends up in a highly-uniformly-distributed key.

Let $R := S_w(X;Y||Z)$. We show that for any $\varepsilon > 0$ (and for a sufficiently large number of realizations of the random variables) secret-key agreement at a rate at least $R - \varepsilon$ is possible even with respect to the stronger definition, including the uniformity condition (4.10).

For parameters $\varepsilon' > 0$ and $N > 0$, both to be determined later, let secret-key agreement (not necessarily satisfying the new condition) be carried out M times independently. Let S_i and S_i', $i = 1, \ldots, M$, be the generated keys, and let C_i and $(Z^N)_i$ be the corresponding collection of messages sent over the public channel and the realizations of Z that Eve obtains, respectively. Then the triples $[S_i, S_i', (Z^N C)_i]$, $i = 1, \ldots, M$, are statistically independent and identically distributed. According to the definition of $S_w(X;Y||Z)$, we can achieve for every i

$$\begin{aligned} H(S_i)/N &\geq R - \varepsilon' , \\ \mathrm{P}[S_i \neq S_i'] &< \tilde{\varepsilon} , \\ I(S_i;(Z^N C)_i)/N &< \varepsilon' , \end{aligned} \tag{4.11}$$

where the constant $\tilde{\varepsilon}$ will be specified later. (Note that in order to make only $\tilde{\varepsilon}$ smaller and to leave ε' unchanged, it is not necessary to increase N because the condition (4.11) is stricter for larger N: The key can be subdivided into smaller pieces at the end, and for every such piece, the error probability is at most $\tilde{\varepsilon}$.)

Using the fact that for all $\alpha > 0$ and $\delta > 0$, the event $\mathcal{E}(\delta)$ that the sequence $[S_1, S_2, \ldots, S_M]$ is δ-typical has probability at least $1 - \alpha$ for sufficiently large M, we can transform the key vector $[S_1, \ldots, S_M]$ into an almost-uniformly-distributed key T as follows. If $\mathcal{E}(\delta)$ occurs, then let $T := [S_1, \ldots, S_M]$, otherwise $T := \Delta$ for some failure symbol Δ. The key T' is computed from $[S_1', \ldots, S_M']$ analogously. Then, T and T' have the following properties. First,

$$\log|\mathcal{T}| \leq M(H(S) + \delta) + 1$$

and

$$H(T) \geq (1 - \alpha)M(H(S) - \delta)$$

follow from the definitions of T and of δ-typical sequences. For the quantities occurring in the definition of $S_{w,u}(X;Y||Z)$, we hence obtain

$$H(T)/MN \geq (1 - \alpha)(R - \varepsilon' - \delta/N) , \tag{4.12}$$

$$\mathrm{P}[T \neq T'] < M\tilde{\varepsilon} , \tag{4.13}$$

$$I(T;(Z^N C)_{i=1,\ldots,M})/MN < \varepsilon' , \tag{4.14}$$

$$(\log|\mathcal{T}| - H(T))/MN \leq \alpha R + 2\delta/N . \tag{4.15}$$

Because of Lemma 4.4 one can choose, for every sufficiently large N, constants α, δ, and ε' such that $\mathrm{P}[\overline{\mathcal{E}(\delta)}] < \alpha$ (where $\overline{\mathcal{E}(\delta)}$ stands for the complementary event of $\mathcal{E}(\delta)$) for this choice of M, and that the expressions on the right-hand sides of (4.14) and (4.15) are smaller than

ε, whereas the right-hand side of (4.12) is greater than $R-\varepsilon$. Finally, $\tilde{\varepsilon}$ can be chosen as ε/M, such that the condition (4.13) is also satisfied.

We conclude that the uniform weak secret-key rate $S_{w,u}(X;Y||Z)$ is at least $R = S_w(X;Y||Z)$. Hence we have

$$S_{w,u}(X;Y||Z) = S_w(X;Y||Z) ,$$

and this concludes the proof. □

Lemma 4.6 links typicality of sequences (and hence Shannon entropy) with Rényi entropy of the corresponding random variables. More precisely, the conditional Rényi entropy of a sequence is close to the length of the sequence times the conditional Shannon entropy of the original random variables, given a certain typicality event occurring with high probability.

Lemma 4.6. *Let P_{XZ} be the joint distribution of two random variables X and Z, let $0 < \delta \leq 1/2$, and let N be an integer. The event $\mathcal{F}(\delta)$ is defined as follows: First, the sequences x^N and $(x,z)^N$ must be δ-typical, and second, z^N must be such that the probability, taken over $(x')^N$ according to the distribution $P_{X^N|Z^N=z^N}$, that $(x',z)^N$ is δ-typical is at least $1-\delta$. Then we have*

$$N \cdot \mathrm{P}[\overline{\mathcal{F}(\delta)}] \longrightarrow 0$$

for $N \to \infty$, and

$$H_2(X^N|Z^N = z^N, \mathcal{F}(\delta)) \geq N(H(X|Z) - 2\delta) + \log(1-\delta) .$$

Proof. Because of Lemma 4.4, the event, denoted by $\mathcal{E}(\delta)$, that both x^N and $(x,z)^N$ are δ-typical has probability at least $1-\delta^2$ for some $N = N(\delta)$ with $N(\delta)\cdot\delta \to 0$. For this value of N, z^N has with probability at least $1-\sqrt{\delta^2} = 1-\delta$ the property that $(x',z)^N$ is with probability at least $1-\sqrt{\delta^2} = 1-\delta$, taken over $(x')^N$ according to $P_{X^N|Z^N=z^N}$, δ-typical. Hence the probability of the complementary event $\overline{\mathcal{F}(\delta)}$ of $\mathcal{F}(\delta)$ is at most $\delta^2+\delta$, thus $N \cdot \mathrm{P}[\overline{\mathcal{F}(\delta)}] \to 0$.

On the other hand, given that z^N and $(x',z)^N$ are δ-typical, we can conclude as in the proof of Theorem 4.1 that

$$2^{-N(H(X|Z)+2\delta)} \leq P_{X^N|Z^N}((x')^N, z^N) \leq 2^{-N(H(X|Z)-2\delta)}$$

holds. For a fixed value z^N, the Rényi entropy of X^N, given the events $Z^N = z^N$ and $\mathcal{F}(\delta)$, is lower bounded by the Rényi entropy of a uniform distribution over a set with $(1-\delta) \cdot 2^{N(H(X|Z)-2\delta)}$ elements:

$$H_2(X^N|Z^N = z^N, \mathcal{F}(\delta)) \geq N(H(X|Z) - 2\delta) + \log(1-\delta) .$$

□

4.2.3 A General Lower Bound on the Secret-Key Rate

In this section we prove the lower bound (4.6) on the secret-key rate. Note that this bound is a consequence of the results by Csiszár and Körner [34] and of Section 4.3, but that the proof given here is constructive and much simpler (a fact that is not surprising since in this model, in contrast to the broadcast-channel model of [34], Alice and Bob can use a noiseless public channel for free).

Theorem 4.7 was proved as the main result in [63]. The proof given below is a simplified version of the proof given there.

Theorem 4.7. *For every distribution P_{XYZ},*

$$S(X;Y||Z) \geq \max \{ I(X;Y) - I(X;Z) \, , \, I(Y;X) - I(Y;Z) \} .$$

Proof. We only prove that $I(X;Y) - I(X;Z) = H(X|Z) - H(X|Y)$ is an achievable rate. The statement then follows by symmetry.

Let $\varepsilon > 0$. We show that for sufficiently large N, there exists a protocol which achieves the above rate. Let $\delta < \varepsilon/4$ and $\alpha < \varepsilon/(2H(X))$ be constants, and let $\mathcal{F}(\delta)$ be the event defined in Lemma 4.6. Because of Lemma 4.6 we have for sufficiently large N that $N \cdot \mathrm{P}[\overline{\mathcal{F}(\delta)}] < \alpha$. On the other hand

$$H_2(X^N|Z^N = z^N, \mathcal{F}(\delta)) \geq N \cdot (H(X|Z) - 2\delta) + \log(1-\delta) .$$

The protocol now consists of two messages sent from Alice to Bob, one for information reconciliation and the other one for privacy amplification (see Section 4.2.1). Let $\beta < \varepsilon/(2H(X|Y))$ be a positive constant. According to Theorem 4.1 there exists for sufficiently large N a function $h : \mathcal{X}^N \to \{0,1\}^L$, where

$$L := \lceil (1+\beta) N H(X|Y) \rceil ,$$

such that X^N can be determined from Y^N and $h(X^N)$ with probability at least $1-\varepsilon$ (using the optimal strategy). Clearly, the value $h(X^N)$ reduces Eve's uncertainty in terms of Rényi entropy about X^N. We conclude from Lemma 4.3 for $s := 2\log(2NH(X)/\varepsilon)+2$ that with probability at least $1-2^{-(s/2-1)}$,

$$\begin{aligned} H_2(X^N|Z^N = z^N, h(X^N) = h(x^N), \mathcal{F}(\delta)) \\ &\geq N \cdot (H(X|Z) - 2\delta) + \log(1-\delta) \\ &\quad -[(1+\beta)\cdot N \cdot H(X|Y) + 1 + s] \\ &= N \cdot (H(X|Z) - H(X|Y)) - 2\delta N \\ &\quad -\beta N H(X|Y) - 1 - s + \log(1-\delta) \\ &=: Q \ . \end{aligned} \tag{4.16}$$

Finally, Alice and Bob transform their mutual information X^N into a highly-secret string S by privacy amplification. Let $r := \lceil \log N \rceil$, and let $M := Q - r$ be the length of the resulting string S. If G is the random variable corresponding to the random choice of a universal hash function mapping $\mathcal{X}^N \to \{0,1\}^M$, and if $S := G(X^N)$, then we have

$$H(S|Z^N = z^N, h(X^N) = h(x^N), G, \mathcal{F}(\delta)) \geq M - 2^{-r}/\ln 2$$

under the condition that inequality (4.16) holds. Hence we get for sufficiently large N

$$\begin{aligned} H(S|Z^N, h(X^N), G) \\ &\geq (\mathrm{P}[\mathcal{F}(\delta)] - 2^{-(s/2-1)})(M - 2^{-r}/\ln 2) \\ &\geq M - 2^{-r}/\ln 2 - (\mathrm{P}[\overline{\mathcal{F}(\delta)}] + 2^{-(s/2-1)}) \cdot N \cdot H(X) \\ &> \log|\mathcal{S}| - \varepsilon \end{aligned}$$

by definition of r, α, and s. For $C = [h(X^N), G]$, this implies both

$$I(S; Z^N C) < \varepsilon$$

and

$$H(S) > \log|\mathcal{S}| - \varepsilon$$

because

$$H(S) \geq H(S|Z^N, h(X^N), G) \ .$$

Finally, note that the probability that Bob's estimate S' of S differs from S is upper bounded by ε. The achievable key-generation rate with this protocol is hence at least

$$H(X|Z) - H(X|Y) - 2\delta - \beta H(X|Y) \geq I(X;Y) - I(X;Z) - \varepsilon \ .$$

Thus we obtain

$$S(X;Y||Z) \geq I(X;Y) - I(X;Z) \ ,$$

which concludes the proof. □

4.2.4 Equality of Weakly- and Strongly-Defined Rates

Theorem 4.8 states that the secret-key rate is always equal to the weak secret-key rate. Hence the attention can be restricted completely to the stronger version of secret-key agreement with respect to the more satisfactory definition actually given in Section 4.1.

Theorem 4.8. *For all distributions P_{XYZ}, we have*

$$S(X;Y||Z) = S_w(X;Y||Z) \ .$$

Proof. Clearly, $S(X;Y||Z) \leq S_w(X;Y||Z)$. Let $R := S_w(X;Y||Z)$, and let $\varepsilon > 0$. According to the definition of the secret-key rate $S_w(X;Y||Z)$ (and because of Lemma 4.5), there exists, for sufficiently large N, a protocol with the following properties: Alice and Bob know, at the end of the protocol, strings S and S' such that

$$\begin{aligned} H(S) &\geq NR - N\varepsilon \ , \\ \mathrm{P}[S \neq S'] &< \varepsilon \ , \\ I(S;Z^N C) &\leq N\varepsilon \ , \\ H(S) &\geq \log|\mathcal{S}| - N\varepsilon \ . \end{aligned}$$

From these equations, we can conclude by Fano's inequality (2.11) that

$$\begin{aligned} I(S;S') &= H(S) - H(S|S') \\ &\geq H(S) - h(\mathrm{P}[S \neq S']) - \mathrm{P}[S \neq S'](H(S) + N\varepsilon) \\ &> H(S)(1-\varepsilon) - h(\varepsilon) - N\varepsilon^2 \\ &\geq NR - NR\varepsilon - N\varepsilon - h(\varepsilon) \ , \end{aligned}$$

hence

$$I(S;S') - I(S;Z^N C) \geq NR - NR\varepsilon - 2N\varepsilon - h(\varepsilon) \ .$$

Let us now consider the random experiment $[S, S', Z^N C]$ (where we assume that the realizations are independent). By applying Theorem 4.7 to the new distribution, we get

$$\begin{aligned} S(X;Y||Z) &\geq S(S;S'||Z^N C)/N \\ &\geq (I(S;S') - I(S;Z^N C))/N \\ &\geq R - R\varepsilon - 2\varepsilon - h(\varepsilon)/N \end{aligned}$$

for every $\varepsilon > 0$, thus

$$S(X;Y||Z) \geq S_w(X;Y||Z) \ .$$

□

4.3 Strengthening the Definition of the Secrecy Capacity in Wyner's and Csiszár and Körner's Models

This section is concerned with the model introduced by Wyner [122] and the generalization thereof by Csiszár and Körner [34], which served as a motivation for Maurer's [64] and Ahlswede and Csiszár's [2] scenarios of key agreement by public discussion from common information, as described in Chapter 3. In Csiszár and Körner's setting, Alice sends information to Bob (and Eve) over a noisy broadcast channel. (Wyner's model corresponds to the special case where the sequence formed by Alice's, Bob's, and Eve's information forms a Markov chain.) The secrecy capacity has been defined as the maximal rate at which Alice and Bob can generate a mutual secret string by communication only over the given noisy (one-way) broadcast channel such that the rate at which the eavesdropper receives information about the key is arbitrarily small. In analogy to the weak definition of the secret-key rate, this definition is not entirely satisfactory because the total amount of information about the resulting key that the adversary obtains can be unbounded. Following the lines of the arguments used in Section 4.2, we show that the definition of the secrecy capacity can be strengthened, without any effect on the actual value of this quantity, in the sense that the total amount of information the adversary obtains about the secret key is negligibly small. However, an additional problem that arises here is that no communication is "for free" since there is no public discussion channel. However, the use of extractors (see Section 4.3.2) instead of universal hashing for privacy amplification allows to keep the fraction of channel uses for communicating the error-correction and privacy-amplification messages arbitrarily small.

4.3.1 The Secrecy Capacity $C_S(P_{YZ|X})$

Assume that the parties Alice and Bob, and the adversary Eve, are connected by a noisy broadcast channel with conditional output distribution $P_{YZ|X}$ [34] (see Figure 3.7). Wyner's wire-tap channel (see Figure 3.6) corresponds to the special case where

$$P_{YZ|X} = P_{Y|X} \cdot P_{Z|Y}$$

holds. The ability of generating mutual secret information was quantified as follows.

Definition 4.5. [122], [34] Consider a broadcast channel characterized by the conditional joint distribution $P_{YZ|X}$. The *secrecy capacity*

$$C_S(P_{YZ|X})$$

of the channel is the maximal real number $R \geq 0$ such that for every $\varepsilon > 0$ and for $K := \lfloor (R-\varepsilon)N \rfloor$, there exists a possibly probabilistic (i.e., additionally depending on some random bits) encoding function

$$e \;:\; \{0,1\}^K \longrightarrow \mathcal{X}^N$$

together with a decoding function

$$d \;:\; \mathcal{Y}^N \longrightarrow \{0,1\}^K$$

such that if S is uniformly distributed over $\{0,1\}^K$, we have for $X^N = e(S)$, $S' := d(Y^N)$, and $P_{Y^N|X^N} = P_{Y|X}^N$ (where $P_{Y|X}$ is the marginal distribution of $P_{YZ|X}$)

$$\mathrm{P}[S' \neq S] < \varepsilon$$

and

$$\frac{1}{K} H(S|Z^N) > 1 - \varepsilon \ . \tag{4.17}$$

○

In [34] it was shown that

$$C_S(P_{YZ|X}) \geq \max_{P_X} \left(I(X;Y) - I(X;Z)\right) \tag{4.18}$$

(where the maximum is taken over all distributions P_X), and that equality holds in (4.18) whenever $I(X;Y) - I(X;Z)$ is non-negative for all distributions P_X. On the other hand, it is clear that inequality (4.18) also implies that if $U \to X \to YZ$ is a Markov chain, then

$$C_S(P_{YZ|X}) \geq I(U;Y) - I(U;Z) \ .$$

If the maximization is extended this way, then equality always holds:

$$C_S(P_{YZ|X}) = \max_{P_{UX}} \left(I(U;Y) - I(U;Z)\right) \tag{4.19}$$

(where $P_{YZ|UX} = P_{YZ|X}$).

It is a consequence of the last equation that Alice and Bob can generate a secret key only in scenarios that provide some kind of advantage of the legitimate partners over the opponent in terms of the broadcast channel's conditional distribution $P_{YZ|X}$.

4.3.2 Privacy Amplification with Extractors

In order to show that the notion of secrecy used in the definition of C_S can be significantly strengthened without reduction of this quantity, we need a different technique for privacy amplification, requiring less information to be transmitted, namely only an asymptotically arbitrarily small fraction of the number of bits of the partially-secure string to be compressed. (Otherwise, the channel uses needed for sending this message would reduce the achievable key-generation rate.) We show that such a technique is given by so-called *extractors*. Roughly speaking, an extractor allows to efficiently isolate the randomness of some source into virtually-random bits, using a small additional number of random bits as a catalyst, i.e., in such a way that these bits reappear as a part of the almost-uniformly-distributed output. Extractors are of great importance in theoretical computer science, where randomness is often regarded as a resource. They have been studied intensively in the past years by many authors. For an introduction to the subject and some constructions, see for example [91] or [114], and the references therein.

Recent results, described below, show that such functions allow, using only a small amount of true randomness, to distill (almost) the entire randomness, measured in terms of H_∞, of some string into an almost-uniformly-distributed string. A disadvantage of using extractors instead of universal hashing is that a string of length only roughly equal to the *min*-entropy instead of the generally greater *Rényi* entropy of the original random variable can be extracted. However, this drawback has no effect in connection with typical sequences, i.e., almost-uniform distributions. (Notice that for perfectly-uniform distributions, all the mentioned entropy measures are equal.) In Chapter 6, we again make use of privacy amplification with extractors, namely in Protocol EX.

Definition 4.6. A function $E : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^r$ is called a (δ', ε')-extractor if for any distribution P on $\{0,1\}^N$ with min-entropy $H_\infty(P) \geq \delta' N$, the variational distance of the distribution of

$$[V, E(X,V)]$$

to the uniform distribution over $\{0,1\}^{d+r}$ is at most ε' when choosing X according to P and V independently according to the uniform distribution over $\{0,1\}^d$. ○

The following theorem was proved in [114]. It states that there exist extractors which distill virtually all the min-entropy out of a weakly-random source, thereby requiring only a small (i.e., "poly-logarithmic") number of truly-random bits. Note that Theorem 4.9 is formally slightly stronger than the corresponding theorem in [114] because it not only states that the length of the extractor output is roughly equal to the min-entropy of the source plus the number of random bits, but even that these bits reappear as a part of the output. Although not explicitly stated, it is not difficult to see that the extractors described in [114] have this property.

Theorem 4.9. **[114]** *For every choice of the parameters N, $0 < \delta' < 1$, and $\varepsilon' > 0$, there exists a (δ', ε')-extractor*

$$E : \{0,1\}^N \times \{0,1\}^d \longrightarrow \{0,1\}^{\delta' N - 2\log(1/\varepsilon') - O(1)} ,$$

where $d = O((\log(N/\varepsilon'))^2 \log(\delta' N))$.

Corollary 4.10, which is a consequence of Theorem 4.9, is what we need in the proof of Theorem 4.12. The statement of Corollary 4.10 is related to Theorem 4.2, where universal hashing is replaced by extractors, and min-entropy must be used instead of Rényi entropy (see Figure 4.2).

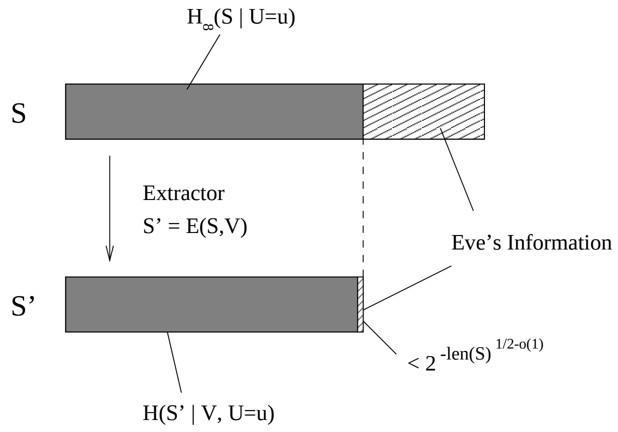


Figure 4.2: Privacy Amplification with Extractors

Corollary 4.10. *Let* $\delta', \Delta_1, \Delta_2 > 0$ *be constants. Then there exists, for all sufficiently large* N*, a function*

$$E \,:\, \{0,1\}^N \times \{0,1\}^d \longrightarrow \{0,1\}^r \ ,$$

where $d \leq \Delta_1 N$ *and* $r \geq (\delta' - \Delta_2)N$*, such that for all random variables* T *with* $\mathcal{T} \subseteq \{0,1\}^N$ *and*

$$H_\infty(T) > \delta' N \ ,$$

we have

$$H(E(T,V)|V) \geq r - 2^{-N^{1/2-o(1)}} \ . \tag{4.20}$$

Proof. Let $\varepsilon'(N) := 2^{-\sqrt{N}/\log N}$. Then there exists N_0 such that for all $N \geq N_0$ we have a (δ', ε')-extractor E, mapping $\{0,1\}^{N+d}$ to $\{0,1\}^r$, where $d \leq \Delta_1 N$ (note that $d = O(N/\log N)$ holds for this choice of ε')

and $r \geq (\delta' - \Delta_2)N$. By definition, this means that for a uniformly-distributed d-bit string V and if $H_\infty(T) \geq \delta' N$, the distance of the distribution of $[V, E(T,V)]$ to the uniform distribution U_{d+r} over $\{0,1\}^{d+r}$ is at most $\varepsilon' = 2^{-\sqrt{N}/\log N}$. Because

$$d([V, E(T,V)], U_{d+r}) = \mathrm{E}_V[d(E(T,V), U_r)] \leq \varepsilon'$$

for uniformly distributed V, the distance of the distribution of $E(T,v)$ to the uniform distribution U_r (over $\{0,1\}^r$) is at most $\sqrt{\varepsilon'}$ with probability at least $1 - \sqrt{\varepsilon'}$ over v, i.e.,

$$P_V\left[d\left(E(T,V), U_r\right) \leq 2^{-\sqrt{N}/2\log N}\right] \geq 1 - 2^{-\sqrt{N}/2\log N} \ .$$

Inequality (4.20) now follows from Lemma 2.3. □

In analogy to Lemma 4.3, which gives an upper bound on the effect of side information on the Rényi entropy of a random variable (hereby linking information reconciliation and privacy amplification with universal hashing) we now need such a result with respect to min-entropy H_∞. Lemma 4.11 is an immediate consequence of Lemma 6.15.

Lemma 4.11. *Let X and Q be random variables, and let $s > 0$. Then with probability at least $1 - 2^{-s}$ (taken over $q \in \mathcal{Q}$), we have*

$$H_\infty(X) - H_\infty(X|Q = q) \leq \log|\mathcal{Q}| + s \ .$$

4.3.3 The Strong Secrecy Capacity $\overline{C_S}(P_{YZ|X})$

In this section we show that the definition of the secrecy capacity in Csiszár and Körner's, hence also in Wyner's, model can be strengthened similarly to the weak and strong notions of secret-key rate: Not the rate, but the total amount of leaked information is small. Note that an additional uniformity condition is not necessary here since the definition of C_S even requires the key to be perfectly-uniformly distributed.

Definition 4.7. For a distribution $P_{YZ|X}$, the *strong secrecy capacity* $\overline{C_S}(P_{YZ|X})$ is defined similarly to $C_S(P_{YZ|X})$, but where the secrecy condition (4.17) is replaced by the stronger requirement

$$H(S|Z^N) > K - \varepsilon \ . \tag{4.21}$$

○

Theorem 4.12. *For all conditional distributions $P_{YZ|X}$, we have*

$$\overline{C_S}(P_{YZ|X}) = C_S(P_{YZ|X}) \ .$$

Proof. The idea of the proof is to repeat the (weak) key generation a number of times and to compute from the block of resulting weak keys a secure string satisfying the stronger definition of secrecy capacity. More precisely, this is done by information reconciliation as described in Section 4.2.1, and by privacy amplification with extractors. Since the parties have, in contrast to the public-discussion model, no access to a noiseless public channel, all the error-correction and privacy-amplification information must be sent over the noisy channel characterized by $P_{Y|X}$. However, the use of extractors instead of universal hashing for privacy amplification allows to keep the fraction of channel uses required for this communication negligibly small. This is precisely what is needed for showing equality of C_S and $\overline{C_S}$.

Let $R := C_S(P_{YZ|X})$. For a constant $\varepsilon' > 0$ and integers M and N to be determined later, assume that the key-generation procedure, with respect to the (weak) secrecy capacity C_S and parameters ε' and N, is repeated independently M times. Let $S^M := [S_1, \ldots , S_M]$ and $(S')^M := [S'_1, \ldots , S'_M]$ be the generated keys of Alice and Bob, respectively, and let $K = \lfloor (R-\varepsilon')N \rfloor$ be the length of the binary strings S_i and S'_i. From the fact that $\mathrm{P}[S_i \neq S'_i] < \varepsilon'$ holds we conclude, by Fano's inequality (2.11),

$$H(S_i|S'_i) \leq \varepsilon' K + 1$$

for all i, hence

$$H(S^M|(S')^M) \leq M(\varepsilon' K + 1) \ .$$

For constants $\Delta_1, \Delta_2 > 0$, we conclude from Theorem 4.1 that there exists an error-correction-information function

$$h \ : \ (\{0,1\}^K)^M \longrightarrow \{0,1\}^{\lceil (1+\Delta_1)M(\varepsilon' K+1)\rceil}$$

such that S^M can be determined from $(S')^M$ and $h(S^M)$ with probability at least $1 - \Delta_2$ for sufficiently large M. Hence $\lceil (1 + \Delta_1)M(\varepsilon' K + 1)\rceil$ message bits have to be transmitted over the channel $P_{Y|X}$ for error correction (see below).

According to the definition of the (weak) secrecy capacity C_S, we have

$$H(S_i|Z_i^N) \geq K(1 - \varepsilon') \ .$$

For $\delta > 0$, let the event $\mathcal{F}(\delta)$, with respect to the random variables S and Z^N, be defined as in Lemma 4.6. For every $\alpha > 0$ we can achieve, for arbitrarily large (fixed) N and M,

$$MK \cdot \mathrm{P}[\overline{\mathcal{F}(\delta)}] < \alpha$$

and

$$H_\infty(S^M|(Z^N)^M = (z^N)^M, \mathcal{F}(\delta)) \geq M(K(1-\varepsilon') - 2\delta) + \log(1-\delta) \ .$$

The reason is that the statement of Lemma 4.6 also holds for the min-entropy H_∞ instead of H_2. The proof of this variant is exactly the same because it is ultimately based on uniform distributions, for which H_2 and H_∞ (and also H) are equal.

Let us now consider the effect of the error-correction information (partially) leaked to the adversary. According to Lemma 4.11, we have for $s > 0$ with probability at least $1 - 2^{-s}$

$$\begin{aligned} & H_\infty(S^M|(Z^N)^M = (z^N)^M, h(S^M) = h(s^M), \mathcal{F}(\delta)) \\ \geq \ & M(K(1-\varepsilon') - 2\delta) + \log(1-\delta) \\ & \quad - \lceil (1+\Delta_1) M(\varepsilon' K + 1) \rceil - s \\ \geq \ & MK(1-\Delta_3) \end{aligned} \tag{4.22}$$

for some constant Δ_3 that can be made arbitrarily small by choosing N large enough, $s := \lceil \log M \rceil$, and Δ_1 as well as ε' small enough.

Let now for constants $\Delta_4, \Delta_5 > 0$ and sufficiently large M an extractor function E be given according to Corollary 4.10, i.e.,

$$E \ : \ \{0,1\}^{MK} \times \{0,1\}^d \longrightarrow \{0,1\}^r$$

with $d \leq \Delta_4 MK$ and $r \geq MK(1 - \Delta_3 - \Delta_5)$ such that, for

$$\tilde{S} := E(S^M, V) \ ,$$

the inequality

$$H(\tilde{S}|(Z^N)^M = (z^N)^M, h(S^M) = h(s^M), V, \mathcal{F}(\delta)) \geq r - 2^{-(MK)^{1/2 - o(1)}}$$

holds for V uniformly distributed in $\{0,1\}^d$. Let S' be the key computed in the same way by Bob (where the random bits V are sent to him by Alice using the channel $P_{Y|X}$ with an appropriate error-correcting code).

The resulting key $\tilde{S}$ of Alice is now close-to-uniformly, but not perfectly-uniformly distributed. Given the events $\mathcal{F}(\delta)$ and that inequality (4.22) holds, we have

$$H(\tilde{S}) \geq r - 2^{-(MK)^{1/2-o(1)}} .$$

Let now S be the "uniformization" of $\tilde{S}$, i.e., the random variable which is uniformly distributed in $\{0,1\}^r$ and jointly distributed with $\tilde{S}$ in such a way that $\mathrm{P}[S \neq \tilde{S}]$ is minimized. It is clear that for any $\Delta_6 > 0$,

$$\mathrm{P}[S \neq \tilde{S}] < \Delta_6$$

can be achieved for sufficiently large M.

Let us finally consider the number of channel uses necessary for communicating the information for information reconciliation and privacy amplification. The number of bits to be transmitted is, according to the above, at most

$$\lceil (1+\Delta_1)M(\varepsilon' K + 1) \rceil + \Delta_4 MK .$$

It is an immediate consequence of Shannon's channel-coding theorem (see Section 2.6.1 or, for example, [29]) that for arbitrary $\Delta_7, \Delta_8 > 0$ and sufficiently large M, the number of channel uses for transmitting these messages can be at most

$$\frac{MK((1+\Delta_1)\varepsilon' + \Delta_4) + (1+\Delta_1)M + 1}{C(P_{Y|X}) - \Delta_7}$$

(where $C(P_{Y|X})$ is the capacity of the channel $P_{Y|X}$ from Alice to Bob), keeping the probability of a decoding error below Δ_8. Note that $C(P_{Y|X}) > 0$ clearly holds when $C_S(P_{YZ|X}) > 0$. (If $C(P_{Y|X}) = 0$, the statement of the theorem is hence trivially satisfied.) Thus the total number of channel uses for the entire key generation can be made smaller than

$$MN(1+\Delta_9)$$

for arbitrarily small $\Delta_9 > 0$ and sufficiently large N.

From the above we can now conclude that S is a perfectly-uniformly-distributed string of length

$$r = (1 - o(1))RL ,$$

where $L = (1 + o(1))MN$ is the total number of channel uses. Furthermore, we have by construction

$$\mathrm{P}[S' \neq S] = o(1)$$

and finally

$$\begin{aligned} H(S|Z^L) &= H(S) - I(S; Z^L) \\ &\geq H(S) - I(\tilde{S}; Z^L) \qquad (4.23) \\ &= r - 2^{-(MK)^{1/2-o(1)}} - r \cdot (2^{-s} + \mathrm{P}[\overline{\mathcal{F}(\delta)}]) \\ &= r - o(1) \; . \end{aligned}$$

Here, inequality (4.23) holds because $Z^L \to \tilde{S} \to S$ is a Markov chain. Hence the achievable rate with respect to the strong secrecy-capacity definition is of order $(1 - o(1))R = (1 - o(1))C_S(P_{YZ|X})$, thus

$$\overline{C_S}(P_{YZ|X}) = C_S(P_{YZ|X})$$

holds. □

4.4 Discussion

The main results of this chapter suggest that the original requirements to unconditionally-secure key agreement, in all the scenarios considered, were too weak. As often done in information theory, the involved quantities, including the information that an opponent is allowed to obtain about the key, were bounded in terms of information *rates*. However, the security of the resulting string is not always satisfactory because only the fraction of Eve's information about the (possibly long) key is small, whereas the total amount of the adversary's knowledge may be unbounded. Fortunately, the definition of secret-key agreement can be strengthened, both in Csiszár and Körner's (hence Wyner's) and Maurer's models (in the latter this stronger security definition was already proposed in [63]), without effect on the achievable key-generation rates, as can be shown using recent results from the theory of extractors. Concludingly, there is no longer any reason to study secret-key agreement with respect to the weaker definitions.

Chapter 5

The Possibility of Secret-Key Agreement and Intrinsic Information

In this chapter we derive criteria for the (im-)possibility of secret-key agreement by public discussion according to the model motivated and described in Chapter 3 and defined in Chapter 4. A new information measure, called the intrinsic conditional information, is defined, and it is shown that this new quantity is not only an upper bound on the secret-key rate, but also often useful for distinguishing between cases where secret-key agreement is possible and impossible in principle. It is conjectured, supported by some evidence, but unknown in general, that the intrinsic information leads to a complete characterization of the possibility of unconditional secret-key agreement in every scenario.

In Section 5.1 we define the intrinsic conditional information and show that this measure gives an improved upper bound on the secret-key rate. In Section 5.2 we formulate some fundamental properties of the secret-key rate. Sections 5.3 and 5.4 address the problem whether secret-key agreement is always possible when this new upper bound is strictly positive. We consider this for the cases where X, Y, and Z are generated by sending a binary random variable over independent channels (Scenario BS in Section 5.3), and where Z is generated by sending X and Y over erasure channels (Scenarios EC1 and EC2 in Section 5.4). For Scenarios BS and EC1 it is shown that secret-key agreement is possible if the intrinsic conditional information is positive. For a generalized version of Scenario EC1, where Eve obtains both Alice's and Bob's information with a certain probability $1-\delta$, the new information measure is shown to be closely related to δ and to a new, natural quantity measuring the deviation of Alice's and Bob's information from statistically-independent information. For Scenario EC2 finally, we show that a new protocol is more powerful than the previously-known protocols.

Many of the results of this chapter have already been published in [77], [75], [76], and [69].

5.1 The Intrinsic Conditional Mutual Information

Let us start with the upper bound (4.5) on the secret-key rate, stating that

$$S(X;Y||Z) \leq \min\{I(X;Y), I(X;Y|Z)\} \ .$$

Trying to reduce the quantity $I(X;Y|Z)$ in this bound, the adversary Eve can send the random variable Z over a channel, characterized by $P_{\overline{Z}|Z}$, in order to generate the random variable $\overline{Z}$. Obviously,

$$S(X;Y||Z) \leq S(X;Y||\overline{Z}) \leq I(X;Y|\overline{Z}) \tag{5.1}$$

holds for every such $\overline{Z}$. A similar bound also appeared in [2]. Inequality (5.1) motivates the following definition of the intrinsic conditional mutual information between X and Y when given Z, which is the infimum of $I(X;Y|\overline{Z})$, taken over all discrete random variables $\overline{Z}$ that can be obtained by sending Z over a channel, characterized by $P_{\overline{Z}|Z}$. Alternatively speaking, $\overline{Z}$ must be such that

$$XY \rightarrow Z \rightarrow \overline{Z}$$

is a Markov chain.

Definition 5.1. For a distribution P_{XYZ}, the *intrinsic conditional mutual information between* X *and* Y *when given* Z, denoted by $I(X;Y{\downarrow}Z)$, is

$$I(X;Y{\downarrow}Z) := \inf \left\{ I(X;Y|\overline{Z}) \ : \ P_{XY\overline{Z}} = \sum_{z\in\mathcal{Z}} P_{XYZ} \cdot P_{\overline{Z}|Z} \right\} \ ,$$

where the infimum is taken over all possible conditional distributions $P_{\overline{Z}|Z}$. ○

The intrinsic conditional information has the following properties:

$$\begin{array}{rcl} 0 \ \leq \ I(X;Y{\downarrow}Z) & \leq & I(X;Y) \ , \\ I(X;Y{\downarrow}Z) & \leq & I(X;Y|Z) \ , \\ I(X;Y{\downarrow}Z) & \leq & I(X;Y{\downarrow}\overline{Z}) \ , \end{array}$$

In the last inequality, $\overline{Z}$ is generated by sending Z over an arbitrary channel (i.e., $XY \to Z \to \overline{Z}$ is a Markov chain).

Theorem 5.1. *For arbitrary random variables X, Y, and Z, we have*

$$S(X;Y||Z) \leq I(X;Y{\downarrow}Z) \ . \tag{5.2}$$

Proof. Bound (5.2) follows from the definition of the intrinsic information and from inequality (5.1). □

Theorem 5.1 implies in particular that secret-key agreement can be possible only if $I(X;Y{\downarrow}Z) > 0$. The following simple example shows that the intrinsic conditional information can be equal to 0, and secret-key agreement is hence impossible, even when both $I(X;Y|Z) > 0$ and $I(X;Y) > 0$ hold. Thus the bound (5.2) is a strict improvement of bound (4.5). Let $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \{0,1,2,3\}$,

$$P_{XYZ}(0,0,0) = P_{XYZ}(0,1,1) = P_{XYZ}(1,0,1) = P_{XYZ}(1,1,0) = \frac{1}{8} \ ,$$

and

$$P_{XYZ}(2,2,2) = P_{XYZ}(3,3,3) = \frac{1}{4} \ .$$

Then $I(X;Y) = 3/2$ and $I(X;Y|Z) = 1/2$ (note that $Z = X \oplus Y$ if $X, Y \in \{0,1\}$), but $I(X;Y{\downarrow}Z) = 0$. To see this, consider the random variable $\overline{Z}$, generated by sending Z over the channel characterized by

$$P_{\overline{Z}|Z}(0,0) = P_{\overline{Z}|Z}(0,1) = P_{\overline{Z}|Z}(1,0) = P_{\overline{Z}|Z}(1,1) = \frac{1}{2}$$

and

$$P_{\overline{Z}|Z}(2,2) = P_{\overline{Z}|Z}(3,3) = 1 \ .$$

Intuitively, giving the side information Z "destroys" all the information between X and Y, but generates new conditional mutual information (that cannot be used to generate a secret key). In contrast to $I(X;Y|Z)$, the intrinsic information $I(X;Y{\downarrow}Z)$ measures only the *remaining* conditional mutual information between X and Y (possibly reduced by giving Z), but *not* the *additional* information between X and Y brought in by Z.

We give a graphical representation of $I(X;Y{\downarrow}Z)$. When given arbitrary X, Y, and Z (i.e., even when $I(X;Y;Z) < 0$), we consider all the random variables $\overline{Z}$ that can be generated by sending Z over a channel $P_{\overline{Z}|Z}$. Note that $I(X;\overline{Z}) \leq I(X;Z)$ and $I(Y;\overline{Z}) \leq I(Y;Z)$ hold for such random variables $\overline{Z}$ because of the data-processing inequality (2.13). The particular $\overline{Z}$ which minimizes $I(X;Y|\overline{Z})$ fulfills $I(X;Y;\overline{Z}) \geq 0$. (This means that there are no negative regions in the graphical representation of X, Y, and $\overline{Z}$, although this may be the case for X, Y, and Z). The quantity $I(X;Y{\downarrow}Z)$ can be directly associated with one of the regions (see Figure 5.1). The random variable $\overline{Z}$ is the one that maximally reduces the size of this region.

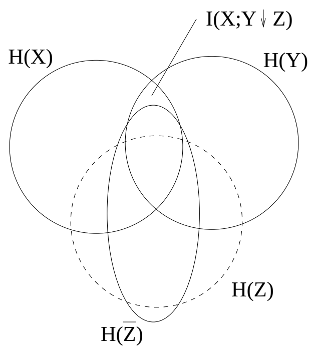


Figure 5.1: Visualization of $I(X;Y{\downarrow}Z)$

5.2 Secret-Key Agreement with General Random Variables

As stated above, the intrinsic conditional mutual information $I(X;Y{\downarrow}Z)$ defined above gives a new upper bound on the secret-key rate, and in particular, secret-key agreement is impossible unless $I(X;Y\downarrow Z) > 0$. It appears plausible that this condition is also sufficient for a positive secret-key rate.

Conjecture 5.1. *Let P_{XYZ} be such that $I(X;Y\downarrow Z) > 0$. Then*

$$S(X;Y||Z) > 0 \ .$$

An even stronger conjecture would be that $S(X;Y||Z) = I(X;Y\downarrow Z)$ holds for every distribution P_{XYZ}. It is a fundamental open problem to prove or disprove the conjecture for the general case. In Section 5.3 we show the validity of Conjecture 5.1 in the scenarios where the random variables X, Y, and Z are noisy versions of a common binary signal (Scenario BS), or when Eve's information Z results from sending the pair $[X,Y]$ over an erasure channel (Scenario EC1).

Another example of a realistic scenario for which the conjecture has been proved is when Alice and Bob use the so-called 4-state or 6-state protocol for quantum key agreement. Here, Eve learns whether the bit sent by Alice over the quantum channel is received by Bob correctly or not, and a noisy version of Bob's bit, where the error probability depends on whether Bob has received his bit correctly or not. It was shown in [43] that, for the case of incoherent eavesdropping strategies, Conjecture 5.1 holds with respect to classical protocols, and that quantum privacy amplification does not work neither in scenarios for which classical key agreement fails.

On the other hand, a scenario is described in Section 5.4 for which it is an open problem whether Conjecture 5.1 holds (Scenario EC2).

As a preparation for the analysis in the following sections we prove some important and basic properties of the secret-key rate. The three lemmas below are intuitive and follow quite directly from the definition of the secret-key rate. Lemma 5.2 states that Alice and Bob cannot increase the secret-key rate by ignoring certain realizations of the random variables X and Y. We say that Alice and Bob obtain new random variables by *restriction of the ranges* if they discard realizations that do not lie in certain subsets $\hat{\mathcal{X}}$ and $\hat{\mathcal{Y}}$ of $\mathcal{X}$ and $\mathcal{Y}$. Lemma 5.3 states that processing X and Y cannot increase the secret-key rate.

Lemma 5.2. *Let X, Y, and Z be random variables with ranges $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ and joint distribution P_{XYZ}. For $\hat{\mathcal{X}} \subseteq \mathcal{X}$ and $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$, we define a new random experiment with random variables $\hat{X}$ and $\hat{Y}$ (with ranges $\hat{\mathcal{X}}$ and $\hat{\mathcal{Y}}$, respectively). If Ω is the event that $X \in \hat{\mathcal{X}}$ and $Y \in \hat{\mathcal{Y}}$ hold,* *and if* $\mathrm{P}[\Omega]$ *is its probability, then the joint distribution of* $\hat{X}$ *and* $\hat{Y}$ *with* Z *is defined as follows:*

$$P_{\hat{X}\hat{Y}Z}(x,y,z) := \frac{P_{XYZ}(x,y,z)}{\mathrm{P}[\Omega]}$$

for all $(x,y,z) \in \hat{\mathcal{X}} \times \hat{\mathcal{Y}} \times \mathcal{Z}$. *(This is a probability distribution for* $\hat{\mathcal{X}} \times \hat{\mathcal{Y}} \times \mathcal{Z}$.*) Then*

$$S(X;Y||Z) \geq \mathrm{P}[\Omega] \cdot S(\hat{X};\hat{Y}||Z) . \tag{5.3}$$

In other words, the secret-key rate cannot be increased by restricting the ranges of X *and* Y.

Proof. The secret-key rate $S(X;Y||Z)$ is the maximum key-generation rate, taken over all possible protocols between Alice and Bob. One possible strategy is to restrict the ranges of their random variables. With probability $\mathrm{P}[\Omega]$, they both receive random variables $\hat{X}$ and $\hat{Y}$, respectively, and inequality (5.3) follows. □

Lemma 5.3. *Let* X, Y, Z, $\overline{X}$, *and* $\overline{Y}$ *be random variables with distribution*

$$P_{XYZ\overline{X}\,\overline{Y}} = P_{XYZ} \cdot P_{\overline{X}|X} \cdot P_{\overline{Y}|Y} ,$$

where $P_{\overline{X}|X}$ *and* $P_{\overline{Y}|Y}$ *are arbitrary conditional probability distributions. (Equivalently,* $YZ \to X \to \overline{X}$ *and* $XZ \to Y \to \overline{Y}$ *are Markov chains.) Then*

$$S(\overline{X};\overline{Y}||Z) \leq S(X;Y||Z) .$$

Proof. As in the proof of Lemma 5.2, the statement follows because it is one of the possible strategies for Alice and Bob to send X and Y over two channels, and because the secret-key rate is the maximum key generation rate taken over all possible protocols. □

Lemma 5.4 states that if Eve has access to a random variable U (in addition to Z) that can be interpreted as side information provided by an oracle, then the secret-key rate is not greater than in the original situation.

Lemma 5.4. *Let X, Y, Z, and U be arbitrary random variables. Then*

$$S(X;Y||[Z,U]) \leq S(X;Y||Z) \ .$$

Proof. Obviously, only condition (4.3) in the definition of the secret key rate is affected. Because $I(S;[Z,U]^N C) \leq \varepsilon$ implies $I(S;Z^N C) \leq \varepsilon$, the lemma follows. □

Theorem 5.5 is an immediate consequence of the three Lemmas 5.2, 5.3, and 5.4.

Definition 5.2. We say that $\overline{X}$ *and* $\overline{Y}$ *are generated from* X *and* Y *with positive probability* if one can obtain from X and Y random variables $\hat{X}$ and $\hat{Y}$ by restriction of the ranges (see above), and the random variables $\overline{X}$ and $\overline{Y}$ by sending $\hat{X}$ and $\hat{Y}$ over two channels, specified by $P_{\overline{X}|\hat{X}}$ and $P_{\overline{Y}|\hat{Y}}$. ○

Theorem 5.5. *Let X, Y, Z, and U be arbitrary random variables, and let $\overline{X}$ and $\overline{Y}$ be generated from X and Y with positive probability. Then $S(\overline{X};\overline{Y}||[Z,U]) > 0$ implies $S(X;Y||Z) > 0$.*

5.3 Noisy Versions of a Binary Signal

5.3.1 Discrete Random Variables

The first special scenario, which we analyze completely in this section, is defined as follows. It is a generalization of the "satellite model" described in Section 3.3.2.

Scenario BS (Binary Signal). Let R be an arbitrary binary random variable, and let X, Y, and Z be arbitrary discrete random variables, generated by sending R over independent channels C_A, C_B, and C_E, i.e.,

$$P_{XYZ|R} = P_{X|R} \cdot P_{Y|R} \cdot P_{Z|R} \ . \tag{5.4}$$

In other words, X, Y, and Z are statistically independent when given R. The following is a different but equivalent characterization of Scenario BS. There exist $0 \leq \lambda \leq 1$ and probability distributions $P_X^{(1)}$, $P_X^{(2)}$, $P_Y^{(1)}$, $P_Y^{(2)}$, $P_Z^{(1)}$, and $P_Z^{(2)}$ such that

$$P_{XYZ} = \lambda \cdot P_X^{(1)} \cdot P_Y^{(1)} \cdot P_Z^{(1)} + (1-\lambda) \cdot P_X^{(2)} \cdot P_Y^{(2)} \cdot P_Z^{(2)},$$

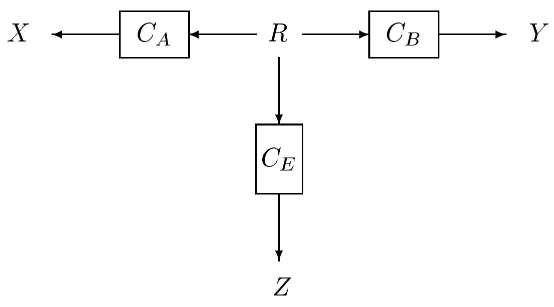


Figure 5.2: Scenario BS.

i.e., P_{XYZ} is a weighted sum of two different distributions over the set $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ that both correspond to independent random variables with ranges $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$.

The main result of this section is the following theorem, completely characterizing the cases for which $S(X;Y||Z) > 0$ holds (i.e., secret-key agreement is possible in principle) in Scenario BS, and implying that Conjecture 5.1 is true in this case.

Theorem 5.6. *In Scenario BS, the following conditions are equivalent:*

(A) $I(X;Y|Z) > 0$,

(B) $S(X;Y||Z) > 0$,

(C) $I(X;Y{\downarrow}Z) > 0$.

The proof that *(A)* implies *(B)* is subdivided into several steps stated below as lemmas. We begin with the special case where R is a symmetric binary random variable and all three channels are binary symmetric. This special result is not necessary for the proof of Theorem 5.6, but we show it in order to present the protocol and some estimates that will be useful later. Finally, we show a similar result also for *continuous* instead of discrete random variables (see Section 5.3.4).

5.3.2 Binary-Symmetric Channels

Let us first consider the following special case of Scenario BS. Let

$$P_R(0) = P_R(1) = 1/2$$

and consider three independent binary symmetric channels C_A, C_B, and C_E with bit-error probabilities α, β, and ε, respectively, i.e., we have

$$P_{X|R}(0,0) = 1-\alpha \ , \quad P_{Y|R}(0,0) = 1-\beta \ , \quad P_{Z|R}(0,0) = 1-\varepsilon \ ,$$

where $0 \leq \alpha < 1/2$, $0 \leq \beta < 1/2$, and $0 < \varepsilon \leq 1/2$. We can assume here that $\alpha = \beta$, i.e., that Alice's and Bob's channels are identical. If for example $\alpha < \beta$, Alice can cascade her channel with another binary symmetric channel with error probability $(\beta-\alpha)/(1-2\alpha)$ to obtain error probability β. (In the particular case of binary symmetric channels it is not even necessary to assume $\alpha = \beta$. The statement of Lemma 5.7 also holds if $\alpha \neq \beta$ when the party with the greater error probability is the sender and the other party is the receiver in Protocol RC described below.)

Scenario BS(symm) (Binary Signal, Symmetric). The distribution P_{RXYZ} is defined by the fact that all the random variables are binary and symmetric, by

$$P_{X|R}(0,0) = P_{Y|R}(0,0) = 1-\alpha \ \text{ and } \ P_{Z|R}(0,0) = 1-\varepsilon \ ,$$

where $0 \leq \alpha < 1/2$ and $0 < \varepsilon \leq 1/2$, and by condition (5.4).

In Scenario BS(symm), Alice can send a randomly-chosen bit C to Bob by the following protocol, which was already described in [64].

Protocol RC (Repeat Code). Let N be fixed. Alice sends

$$[C \oplus X_1, C \oplus X_2, \ldots , C \oplus X_N]$$

over the public channel. Bob computes

$$[(C \oplus X_1) \oplus Y_1, \ldots , (C \oplus X_N) \oplus Y_N]$$

and accepts exactly if this is equal to either $[0,0,\ldots,0]$ or $[1,1,\ldots,1]$. In other words, Alice and Bob make use of a repeat code of length N with the only two codewords $[0,0,\ldots,0]$ and $[1,1,\ldots,1]$. (See Figure 5.3 for an illustration of Protocol RC.)

Note that Bob accepts exactly if Alice and Bob have either the same bit in all positions, i.e., $X^N = Y^N$, or opposite values in each position, i.e., $Y^N = X^N \oplus 1^N$, where $1^N := [1,1,\ldots,1]$. It is intuitively clear

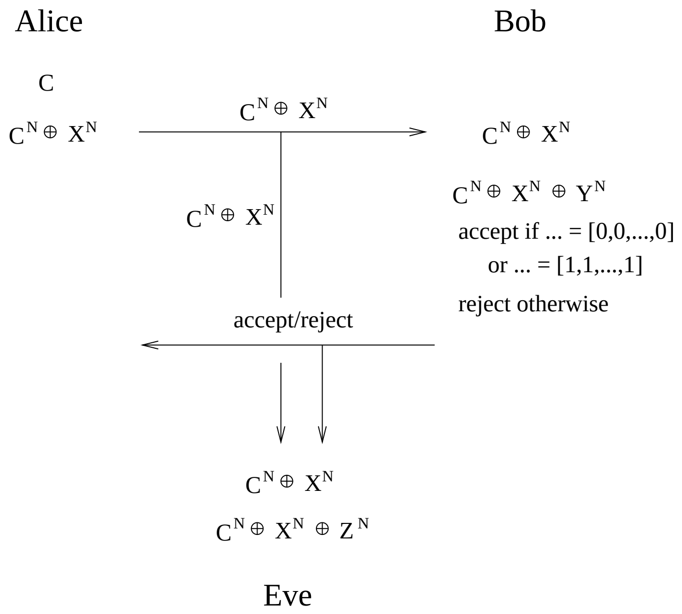


Figure 5.3: Protocol RC

that Alice and Bob not only obtain an arbitrarily low probability of the event that $C' \neq C$ for large N this way, but also that they improve their position compared to the opponent's by accepting only in situations of highly-reliable transmission. However, also the adversary Eve, who can compute $(C^N \oplus X^N) \oplus Z^N$, takes advantage of a greater value of N. It is obvious that Eve's optimal strategy for guessing C is to compute the block $[(C \oplus X_1) \oplus Z_1, \ldots, (C \oplus X_N) \oplus Z_N]$ and guess C as 0 if at least half of the bits in this block are 0, and as 1 otherwise. Note that although it is not the adversary's ultimate goal to guess the bits C sent by Alice, Lemma 5.9 below leads, from Eve's error probability when guessing these bits, to a lower bound on the secret-key rate.

Protocol RC is computationally efficient, but it is not efficient in terms of the size of the generated secret key. As shown in Section 3.3.3, there exist protocols that use parity checks instead of repeat codes and which are much more efficient in terms of the achievable key-generation rate [40].

We show first that for all possible choices of α and ε, in particular even if Eve's channel is superior to both Alice's and Bob's channel, Bob's error probability β_N about the bit sent by Alice decreases, for $N \to \infty$, asymptotically faster than Eve's error probability γ_N when she uses the optimal strategy for guessing this bit. (Note that γ_N is an *average* error probability, and that for a particular realization, Eve's actual error probability will typically be smaller or greater than γ_N.)

Lemma 5.7. *If Protocol RC is used in Scenario BS(symm), there exist real-valued positive constants b and c with $b < c$ such that $\beta_N \leq b^N$ and $\gamma_N \geq c^N$ for sufficiently large N.*

For the proof of Lemma 5.7, we need the following fact on binomial coefficients.

Lemma 5.8. *For sufficiently large even N, we have*

$$\binom{N}{N/2} \geq \frac{1}{\sqrt{2\pi N}} \cdot 2^N \ .$$

Proof. Stirling's formula states that $n!/((n/e)^n \cdot \sqrt{2\pi n}) \to 1$ for $n \to \infty$, and thus we have for sufficiently large even N

$$\binom{N}{N/2} = \frac{N!}{((N/2)!)^2} \geq \frac{1}{2} \cdot \frac{N^N \cdot \sqrt{2\pi N} \cdot e^N}{e^N \cdot (N/2)^N \cdot \pi N} = \frac{1}{\sqrt{2\pi N}} \cdot 2^N \ ,$$

and this concludes the proof. □

Proof of Lemma 5.7. Let α_{rs} $(r, s \in \{0,1\})$ be the probability that the single bit 0 sent by Alice is received by Bob as r and by Eve as s. Then

$$\begin{aligned} \alpha_{00} &= (1-\alpha)^2(1-\varepsilon) + \alpha^2\varepsilon \ , \\ \alpha_{01} &= (1-\alpha)^2\varepsilon + \alpha^2(1-\varepsilon) \ , \\ \alpha_{10} &= \alpha_{11} \ = \ \alpha(1-\alpha) \ . \end{aligned}$$

Let $p_{a,N}$ be the probability that Bob accepts the message sent by Alice. If we assume (without loss of generality) that N is even, then

$$\beta_N = \frac{1}{p_{a,N}} \cdot (\alpha_{10} + \alpha_{11})^N = \frac{1}{p_{a,N}} \cdot (2\alpha - 2\alpha^2)^N \ , \tag{5.5}$$

$$\gamma_N \geq \frac{1}{2} \cdot \frac{1}{p_{a,N}} \cdot \binom{N}{N/2} \alpha_{00}^{N/2} \alpha_{01}^{N/2} \ . \tag{5.6}$$

The last expression is half of the probability that Bob receives the correct codeword, and that Eve receives the same number of 0's and 1's, given that Bob accepts. Note that (5.6) gives a lower bound on Eve's average error probability when guessing C for all possible strategies because in this symmetric case, Eve obtains no information about the bit C, and half of the guesses will be incorrect. From Lemma 5.8 we conclude that

$$\gamma_N \geq \frac{1}{2} \cdot \frac{1}{p_{a,N}} \cdot \frac{1}{\sqrt{2\pi N}} \cdot 2^N \cdot \sqrt{\alpha_{00}\alpha_{01}}^N = \frac{C}{\sqrt{N}} \cdot \frac{\left(2\sqrt{\alpha_{00}\alpha_{01}}\right)^N}{p_{a,N}}$$

for some constant C, and for sufficiently large N. For $0 < \varepsilon \leq 1/2$ we have

$$\sqrt{\alpha_{00}\alpha_{01}} = \sqrt{(1-2\alpha+\alpha^2-\varepsilon+2\alpha\varepsilon)(\alpha^2-2\alpha\varepsilon+\varepsilon)} > \alpha - \alpha^2 . \quad (5.7)$$

For $\varepsilon = 0$ equality holds in (5.7), and for $\varepsilon > 0$ the greater factor of the product under the square root is decreased by the same value by which the smaller factor is increased. Hence the square root of this product is greater than $\alpha - \alpha^2$. (For $\varepsilon = 1/2$ the factors are equal, and the left-hand side of (5.7) is maximal, as expected.) From

$$\begin{aligned} (1-2\alpha+2\alpha^2)^N \;\leq\; p_{a,N} &= (1-2\alpha+2\alpha^2)^N + (2\alpha-2\alpha^2)^N \\ &< 2\cdot(1-2\alpha+2\alpha^2)^N \ , \end{aligned} \quad (5.8)$$

we conclude that $\beta_N \leq b^N$ and $\gamma_N \geq c^N$ for sufficiently large N and for

$$\begin{aligned} b &= (2\alpha-2\alpha^2)/(1-2\alpha+2\alpha^2) \\ c &= 2\sqrt{\alpha_{00}\alpha_{01}}/(1-2\alpha+2\alpha^2) - \delta \end{aligned}$$

(where δ can be made arbitrarily small for sufficiently large N). From the above, we conclude that $c > b$ holds for sufficiently small δ. □

The fact that Eve has a greater error probability than Bob when guessing C does not automatically imply that Eve has a greater uncertainty about this bit in terms of Shannon entropy, and hence that $S(X;Y||Z) > 0$. The next lemma, together with Lemma 5.7, nevertheless implies that the secret-key rate is positive in Scenario BS(symm).

Lemma 5.9. *Let X, Y, and Z be arbitrary random variables, and let C be a bit, randomly chosen by Alice. Assume that for all N, Alice can generate a message M from X^N and C (and possibly some random bits)* *such that with some probability $p_{a,N} > 0$, Bob (who knows M and Y^N) publicly accepts and can compute a bit C' such that* $\mathrm{P}[C \neq C'] \leq b^N$ *for some $b \geq 0$. If in addition, given that Bob accepts, for every strategy for guessing C when given M and Z^N, the average error probability γ_N is at least c^N for some $c > b$ and for sufficiently large N, then $S(X;Y||Z) > 0$.*

Proof. According to Theorem 4.7 is suffices to show that Alice and Bob can, for some arbitrarily large but fixed N, construct random variables $\hat{X}$ and $\hat{Y}$ from X^N and Y^N by exchanging messages over an insecure, but authenticated channel, such that

$$I(\hat{X};\hat{Y}) - I(\hat{X};\hat{Z}) > 0 \tag{5.9}$$

with $\hat{Z} = [Z^N, V]$, where V is the collection of all messages sent over the public channel.

Let $\hat{X}$ and $\hat{Y}$ be defined as follows. Let C be a bit, randomly chosen by Alice, and transmitted to Bob as described in the lemma. If Bob accepts, let $\hat{X} = C$ and $\hat{Y} = C'$, and if Bob (publicly) rejects, let $\hat{X} = \hat{Y} =$ `reject`. We show that (5.9) holds for sufficiently large N. If Bob accepts then

$$H(C|C') = h(b^N) \leq 2b^N \cdot \log(1/b^N) = 2b^N \cdot N \cdot \log(1/b) < c^N$$

for sufficiently large N (where $h(p)$ is the binary entropy function, the first inequality follows from Jensen's inequality (Lemma 2.1), and the reason for the second inequality is that $-p \log p \geq -(1-p)\log(1-p)$ for $p \leq 1/2$). Moreover

$$\begin{aligned} H(C|\hat{Z}) &= \sum_{\hat{z} \in \mathcal{Z}^N \times \mathcal{V}} P_{\hat{Z}}(\hat{z}) \cdot H(C|\hat{Z} = \hat{z}) \; = \; \mathrm{E}_{\hat{Z}}[h(p_{E,\hat{Z}})] \\ &\geq \mathrm{E}_{\hat{Z}}[p_{E,\hat{Z}}] \; = \; \gamma_N \; \geq \; c^N, \end{aligned}$$

where $p_{E,\hat{z}}$ is the probability of guessing C incorrectly with the optimal strategy given that $\hat{Z} = \hat{z}$. Note that $p_{E,\hat{z}} \leq 1/2$, hence $h(p_{E,\hat{z}}) \geq p_{E,\hat{z}}$, for all $\hat{z}$. Given that Bob publicly *rejects*, we have $H(\hat{X}|\hat{Y}) = H(\hat{X}|\hat{Z}) = H(\hat{X}|V) = 0$. From $p_{a,N} > 0$ we conclude that $I(\hat{X};\hat{Y}) - I(\hat{X};\hat{Z}) > 0$. □

5.3.3 General Binary-Input Channels and the Proof of Theorem 5.6

First we show that the above results hold even when Eve knows R *precisely* with a certain probability smaller than 1. This is the case if Z is generated by sending R over a binary *erasure* channel instead of a binary-symmetric channel, i.e., if Z is either equal to a special erasure symbol Δ, or else $Z = R$.

Scenario BS(ec) (Binary Signal, Erasure Channel). Let R, X, and Y be as in Scenario BS(symm), but let Z be generated from R by a (possibly asymmetric) binary erasure channel (with erasure symbol Δ) C_E^*, independent of the pair (C_A,C_B), and with transition probabilities $P_{Z|R}(\Delta,0) = \delta > 0$, $P_{Z|R}(0,0) = 1-\delta$, $P_{Z|R}(\Delta,1) = \delta' > 0$, and $P_{Z|R}(1,1) = 1-\delta'$.

Lemma 5.10. *If Protocol RC is used in Scenario BS(ec), there exist real-valued positive constants b and c with $b < c$ such that $\beta_N \leq b^N$ and $\gamma_N \geq c^N$ for sufficiently large N.*

Proof. We show first that we can assume without loss of generality that C_E^* is symmetric with erasure probability $\delta > 0$. Let $\delta < \delta'$, and let an oracle be given that tells Eve the correct bit R with probability $(\delta'-\delta)/\delta'$ if $R = 1$ and $Z = \Delta$. According to Lemma 5.4, the additional information U provided by this oracle cannot increase the secret-key rate. The pair of random variables $[Z,U]$ is equivalent to a random variable generated from R by a symmetric binary erasure channel with erasure probability δ, and which is independent of the pair (C_A,C_B).

If $\delta = 1$, the lemma is trivial. Let $\delta < 1$, and let $0 < \rho < \min\{\delta, 1-\delta\}$. For sufficiently large N, the probability that the number of bits (out of N bits) known to Eve is even and lies between $(1-\delta-\rho)N$ and $(1-\delta+\rho)N$ is at least $1/3$. We can assume without loss of generality that N and $(1-\delta-\rho)N$ are even integers. (Otherwise, ρ can be chosen slightly smaller in order to fulfill this.) We give a lower bound on Eve's average error probability γ_N about the bit sent by Alice, given that Bob accepts. As in the proof of Lemma 5.7 we obtain a lower bound on Eve's error probability γ_N by taking a (small) part of all positive terms adding up to γ_N. We have

$$\begin{aligned}
\gamma_N &\geq \frac{1}{2} \cdot \frac{(1-2\alpha+2\alpha^2)^N}{p_{a,N}} \cdot \frac{1}{3} \cdot \binom{(1-\delta-\rho)N}{(1-\delta-\rho)N/2} \\
&\quad \cdot \left[\frac{(1-\alpha)^2}{(1-\alpha)^2+\alpha^2}\right]^{(1-\delta+\rho)N/2} \left[\frac{\alpha^2}{(1-\alpha)^2+\alpha^2}\right]^{(1-\delta+\rho)N/2} \\
&\geq \frac{1}{2} \cdot \frac{1}{p_{a,N}} \cdot \frac{1}{3} \cdot \frac{1}{\sqrt{2\pi(1-\delta-\rho)N}} \\
&\quad \cdot \left[(1-2\alpha+2\alpha^2)^{\delta-\rho} 2^{1-\delta-\rho} (\alpha-\alpha^2)^{1-\delta+\rho}\right]^N
\end{aligned}$$

for sufficiently large N. Here we have made use of Lemma 5.8. The first expression is $1/2$ times a lower bound on the probability that Bob receives the correct codeword, that Eve knows an even number of bits which lies between $(1-\delta-\rho)N$ and $(1-\delta+\rho)N$, and that she receives the same number of 0's and 1's in her reliable bits, given that Bob accepts. In this case, Eve obtains no information about the bit sent by Alice. The expressions $(1-\alpha)^2/((1-\alpha)^2+\alpha^2)$ and $\alpha^2/((1-\alpha)^2+\alpha^2)$ are the probabilities that $R = X$ and $R \neq X$ hold, respectively, given that $X = Y$. Bob's error probability, given that he accepts, is $\beta_N = (2\alpha - 2\alpha^2)^N / p_{a,N}$. For sufficiently small (positive) ρ we have

$$(1-2\alpha+2\alpha^2)^{\delta-\rho} 2^{1-\delta-\rho} (\alpha-\alpha^2)^{1-\delta+\rho} > 2\alpha - 2\alpha^2$$

because $\delta > 0$ and $1 - 2\alpha + 2\alpha^2 > 2\alpha - 2\alpha^2$. The lemma is proved because of inequality (5.8). □

We now consider the general Scenario BS. The following lemma states equivalent characterizations of the condition $I(X;Y|Z) > 0$.

Lemma 5.11. *In Scenario BS, the following three conditions are equivalent.*

(i) $I(X;Y|Z) > 0$.

(ii) $I(X;R) > 0$, $I(Y;R) > 0$, *and* $H(R|Z) > 0$.

(iii) There exist $x, x' \in \mathcal{X}$ *such that*

$$P_{X|R}(x,0) > P_{X|R}(x,1) \quad \textit{and} \quad P_{X|R}(x',0) < P_{X|R}(x',1) \ , \tag{5.10}$$

there exist $y, y' \in \mathcal{Y}$ *such that*

$$P_{Y|R}(y,0) > P_{Y|R}(y,1) \quad \textit{and} \quad P_{Y|R}(y',0) < P_{Y|R}(y',1) \ , \tag{5.11}$$

and there exists $z \in \mathcal{Z}$ *such that*

$$P_Z(z) > 0 \quad \textit{and} \quad 0 < P_{R|Z}(0,z) < 1 \ . \tag{5.12}$$

Proof. First we give an alternative characterization of the independence of the three channels, i.e., of $P_{XYZ|R} = P_{X|R} \cdot P_{Y|R} \cdot P_{Z|R}$. (We sometimes omit all the arguments of the probability distribution functions. In this case the statements hold for all possible choices of arguments. For example, $P_{X|Y} = P_X$ stands for the fact that $P_{X|Y}(x,y) = P_X(x)$ holds for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.) From

$$P_{YZ|R} = \sum_{x \in \mathcal{X}} P_{XYZ|R} = \sum_{x \in \mathcal{X}} P_{X|R} \cdot P_{Y|R} \cdot P_{Z|R} = P_{Y|R} \cdot P_{Z|R}$$

and

$$P_R \cdot P_{YZ|R} \cdot P_{X|YZR} = P_{XYZR} = P_R \cdot P_{X|R} \cdot P_{Y|R} \cdot P_{Z|R}$$

we conclude that $P_{X|YZR} = P_{X|R}$ and, analogously, that $P_{Y|XZR} = P_{Y|R}$ and $P_{Z|XYR} = P_{Z|R}$.

(i) implies (ii). Let $I(X;Y|Z) > 0$. Assume $I(X;R) = 0$. Then $P_{X|YZR} = P_{X|R} = P_X$, and X is also independent of YZ (and hence of Z). Thus

$$I(X;Y|Z) = H(X|Z) - H(X|YZ) = H(X) - H(X) = 0 \ ,$$

which is a contradiction. We conclude that $I(X;R) > 0$ and by a symmetric argument that $I(Y;R) > 0$. Finally, assume $H(R|Z) = 0$. Then

$$\begin{aligned} I(X;Y|Z) &= H(X|Z) + \underbrace{H(R|XZ)}_{0} - H(X|YZ) - \underbrace{H(R|XYZ)}_{0} \\ &= H(XR|Z) - H(XR|YZ) \\ &= \underbrace{H(R|Z)}_{0} + H(X|RZ) - \underbrace{H(R|YZ)}_{0} - H(X|RYZ) \\ &= H(X|R) - H(X|R) = 0 \ , \end{aligned}$$

which is a contradiction. Hence $H(R|Z) > 0$.

(ii) implies (iii). Let $I(X;R) > 0$, that is X and R are not statistically independent, which implies that there exists $\overline{x}$ such that $P_{X|R}(\overline{x},0) \neq P_{X|R}(\overline{x},1)$, i.e., such that one of the inequalities of (5.10) holds. Because

$$\sum_{x \in \mathcal{X}} P_{X|R}(x,0) = \sum_{x \in \mathcal{X}} P_{X|R}(x,1) = 1 \ ,$$

there must as well exist an element of $\mathcal{X}$ satisfying the other inequality of (5.10). Similarly, we conclude the existence of appropriate y and y' from $I(Y;R) > 0$. Finally, $P_{R|Z}(0,z) \in \{0,1\}$ for all $z \in \mathcal{Z}$ with $P_Z(z) > 0$ would imply that $H(R|Z) = 0$. Hence (5.12) holds for some $z \in \mathcal{Z}$.

(iii) implies (i). Let x, x', y, y', and z be as in *(iii)*. It suffices to prove that $I(X;Y|Z=z) > 0$ because $P_Z(z) > 0$. This is equivalent to the statement that X and Y are not statistically independent, given $Z = z$. We show that

$$P_{X|YZ}(x,y,z) > P_{X|YZ}(x,y',z) \ . \tag{5.13}$$

For both $\overline{y} = y$ and $\overline{y} = y'$ we have

$$\begin{aligned} P_{X|YZ}(x,\overline{y},z) &= P_{X|R=0}(x) \cdot P_{R|YZ}(0,\overline{y},z) \\ &\qquad + P_{X|R=1}(x) \cdot P_{R|YZ}(1,\overline{y},z) \ . \end{aligned}$$

Because $P_{X|R=0}(x) > P_{X|R=1}(x)$ we must, in order to prove (5.13), to

$$P_{R|YZ}(0,y,z) > P_{R|YZ}(0,y',z) \ . \tag{5.14}$$

Because of $P_{R|YZ} = P_{Y|R} \cdot P_{RZ}/(P_{Y|Z} \cdot P_Z)$, inequality (5.14) is equivalent to

$$\frac{P_{Y|R}(y,0)}{P_{Y|Z}(y,z)} > \frac{P_{Y|R}(y',0)}{P_{Y|Z}(y',z)} \ ,$$

which follows from

$$\begin{aligned} & P_{Y|R=0}(y) \cdot [P_{Y|R=0}(y') \cdot P_{R|Z=z}(0) + P_{Y|R=1}(y') \cdot P_{R|Z=z}(1)] \\ & \quad > \ P_{Y|R=0}(y) \cdot P_{Y|R=0}(y') \qquad\qquad (5.15) \\ & \quad > \ [P_{Y|R=0}(y) \cdot P_{R|Z=z}(0) + P_{Y|R=1}(y) \cdot P_{R|Z=z}(1)] \\ & \qquad \cdot P_{Y|R=0}(y') \ . \end{aligned}$$

Both inequalities in (5.15) follow from the fact that $0 < P_{R|Z=z}(0) < 1$, and because of (5.11). □

We are now ready to prove Theorem 5.6.

Proof of Theorem 5.6. Clearly, *(B)* implies *(C)* by Theorem 5.1, and *(C)* implies *(A)* by the definition of the intrinsic information. We show that *(A)* implies *(B)*. Given that $I(X;Y|Z) > 0$, we construct, from R, X, Y, and Z, random variables $\tilde{R}$, $\tilde{X}$, $\tilde{Y}$, and U with the following properties (see also Figure 5.4):

1. $\tilde{X}$ and $\tilde{Y}$ are generated from X and Y, respectively, with positive probability.
2. $\tilde{R}$ is binary and symmetric, and $\tilde{X}$ and $\tilde{Y}$ can be interpreted as being generated by sending $\tilde{R}$ over two independent binary symmetric channels with identical error probability $\alpha < 1/2$.
3. $\tilde{Z} := [Z, U]$ contains exactly the same information about $\tilde{R}$ as a random variable generated by sending $\tilde{R}$ over a binary erasure channel (which is independent of the channels from (3)) with positive erasure probabilities $\delta > 0$ and $\delta' > 0$.

For such random variables $\tilde{X}$, $\tilde{Y}$, and U, we have by Lemma 5.10 that $S(\tilde{X};\tilde{Y}||[Z,U]) > 0$, and with Theorem 5.5 we conclude that

$$S(X;Y||Z) > 0 \ .$$

Hence it remains to show that suitable random variables $\tilde{R}$, $\tilde{X}$, $\tilde{Y}$, and U exist.

According to Lemma 5.11, there exist $x, x' \in \mathcal{X}$ and $y, y' \in \mathcal{Y}$ such that (5.10) and (5.11) hold. Let $\tilde{X}$ and $\tilde{Y}$ be obtained from X and Y as follows. First, the ranges of X and Y are restricted to $\{x, x'\}$ and $\{y, y'\}$, respectively, and secondly, the resulting random variables X' and Y' (which correspond to a new random experiment) are made symmetric. This is done by sending X' over the following channel to obtain $\tilde{X}$ (we assume $P_X(x) \geq P_X(x')$ without loss of generality):

$$P_{\tilde{X}|X'}(0,x) = \frac{1}{2 \cdot P_{X'}(x)} \ , \quad P_{\tilde{X}|X'}(1,x) = 1 - \frac{1}{2 \cdot P_{X'}(x)} \ ,$$

$$P_{\tilde{X}|X'}(1,x') = 1 \ , \quad P_{\tilde{X}|X'}(0,x') = 0 \ .$$

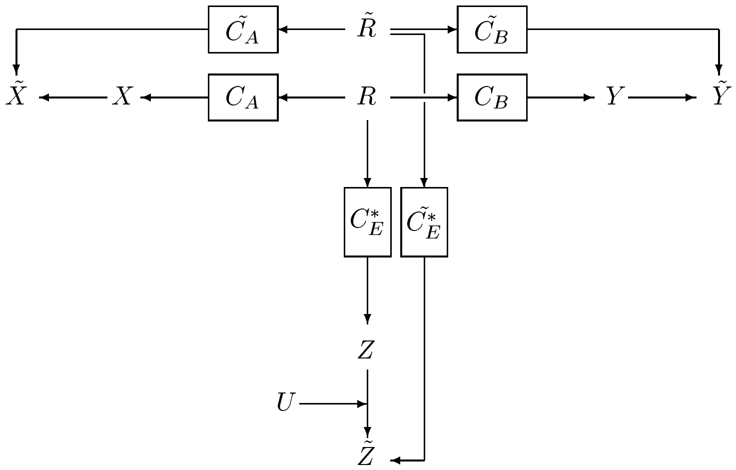


Figure 5.4: The Random Variables in the Proof of Theorem 5.6

In an analogous way, $\tilde{Y}$ and $\tilde{R}$ are obtained from Y' and R, respectively.

According to Lemma 5.11 there exists $z \in \mathcal{Z}$ such that $P_Z(z) > 0$ and $0 < P_{R|Z}(0, z) < 1$. Let the random variable U be defined as follows. If $Z \neq z$, let $U = \tilde{R}$, and if $Z = z$, let $U = \Delta$. Intuitively, the information U can be thought as being provided by an oracle that tells Eve the correct $\tilde{R}$ if $Z \neq z$. Such an oracle can only decrease Eve's average error probability and, according to Lemma 5.4, the secret-key rate.

It remains to show that $\tilde{R}$, $\tilde{X}$, $\tilde{Y}$, and U have all the stated properties. The properties 1 and 3 are satisfied by definition of the random variables. We prove 2. First, it is clear that $\tilde{R}$, $\tilde{X}$, and $\tilde{Y}$ are binary and symmetric. For the rest, we consider the case $P_R(0) \geq 1/2$ and $P_{X'}(x) \geq 1/2$. The other cases are analogous. We have to show $P_{\tilde{X}\tilde{R}}(0,0) > P_{\tilde{X}}(0) \cdot P_{\tilde{R}}(0) = 1/4$, which is sufficient because $\tilde{R}$ and $\tilde{X}$ are symmetric binary random variables.

$$\begin{aligned}
P_{\tilde{X}\tilde{R}}(0,0) &= P_{\tilde{X}\tilde{R}X'R}(0,0,x,0) \\
&= P_R(0) \cdot P_{\tilde{R}|R}(0,0) \cdot P_{X'|R}(x,0) \cdot P_{\tilde{X}|X'}(0,x) \\
&= \frac{1}{4} \cdot \frac{P_{X'|R}(x,0)}{P_{X'}(x)} > \frac{1}{4}
\end{aligned}$$

because $P_{X'|R}(x,0) > P_{X'|R}(x,1)$ and $0 < P_R(0) < 1$. An analogous result can be proved for $\tilde{Y}$. As in the proof of Lemma 5.7, the error probabilities of the two channels can be made identical, and we have proved property 2. The theorem now follows from Lemma 5.10, Theorem 5.5, and Lemma 5.9. Note that in this application of Lemma 5.9 the event that Bob accepts means that Alice and Bob both accept a sufficiently large number N of consecutive realizations of X and Y (if Alice does not accept, she sends $M = \texttt{reject}$ over the public channel), and that Bob accepts the received message sent by Alice. (Of course this would be a very wasteful and inefficient way of generating a secret key in practice. For example it is not necessary that the N realizations of X and Y accepted by Alice and Bob are consecutive.) This concludes the proof. □

The condition that R is a *binary* random variable is crucial in Theorem 5.6. To see this, consider the following example.

Example 5.1. Let R be uniformly distributed over the set

$$\mathcal{R} := \{r_{00}, r_{01}, r_{10}, r_{11}\} ,$$

and let X, Y, and Z be binary random variables, generated from R by the following independent channels (let δ be the Kronecker symbol, i.e., $\delta_{ij} = 1$ if $i = j$, and otherwise $\delta_{ij} = 0$):

$$P_{X|R}(x, r_{ij}) = \delta_{xi} \ , \quad P_{Y|R}(y, r_{ij}) = \delta_{yj} \ , \quad P_{Z|R}(z, r_{ij}) = \delta_{z,i \oplus j} \ .$$

Note that for all $r \in \mathcal{R}$, $Z = X \oplus Y$ holds with probability 1, that is $I(X;Y|Z) = 1$. On the other hand $I(X;Y) = 0$ hence, hence also $S(X;Y||Z) = 0$. ◊

In fact, any distribution P_{XYZ} can be seen as generated by sending a random variable R over three independent channels for some R with $|\mathcal{R}| \leq |\mathcal{X}| \cdot |\mathcal{Y}| \cdot |\mathcal{Z}|$. Such a random variable R can be defined as follows. Let $\mathcal{R} := \{r_{xyz} \mid (x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}\}$ and $P_{X|R}(\overline{x}, r_{xyz}) = \delta_{\overline{x}x}$, $P_{Y|R}(\overline{y}, r_{xyz}) = \delta_{\overline{y}y}$, and $P_{Z|R}(\overline{z}, r_{xyz}) = \delta_{\overline{z}z}$.

5.3.4 Continuous Random Variables

Let us consider the modification of Scenario BS, which we call Scenario BS(cont) (Binary Signal, Continuous), for which the random variables that are generated from R are not necessarily discrete. For example, this is the case if Eve receives her information about R from a Gaussian channel.

Let X, Y, and Z be continuous random variables, and let f_{XYZ}, $f_{X|Y}$, $\ldots$ be the probability density functions (we assume that such functions exist). The differential entropy of X, the conditional differential entropy of X when given Y, and the mutual information between X and Y are defined as follows (see for example [29]):

$$\begin{aligned} h(X) &= -\int f_X \cdot \log f_X \, dx \\ h(X|Y) &= -\int f_{XY} \cdot \log f_{X|Y} \, dx \, dy \\ I(X;Y) &= h(X) - h(X|Y) = \int f_{XY} \cdot \log \frac{f_{XY}}{f_X \cdot f_Y} \, dx \, dy \end{aligned}$$

The conditional information between X and Y when given Z can be defined in analogy to the case of discrete random variables as follows:

$$\begin{aligned} I(X;Y|Z) &= h(X|Z) - h(X|YZ) \\ &= \int f_{XYZ} \cdot \log \frac{f_{XY|Z}}{f_{X|Z} \cdot f_{Y|Z}} \, dx \, dy \, dz \\ &= \int I(X;Y|Z=z) \cdot f_Z(z) \, dz \ . \end{aligned}$$

As in Section 5.3 we assume that X, Y, and Z are generated by sending a binary random variable R over independent channels, i.e.,

$$f_{XYZ|R} = f_{X|R} \cdot f_{Y|R} \cdot f_{Z|R} \ , \tag{5.16}$$

or equivalently $f_{X|RYZ} = f_{X|R}$, $f_{Y|RXZ} = f_{Y|R}$, and $f_{Z|RXY} = f_{Z|R}$.

Theorem 5.12. *Let R be a binary random variable, and let X, Y, and Z be (real-valued) random variables with probability density function f_{XYZ} and conditional density $f_{XY|Z}$. Assume that the joint distribution of XYZ with R satisfies (5.16). Then secret-key agreement is possible, i.e., $S(X;Y||Z) > 0$, if $I(X;Y|Z) > 0$.*

Proof. We assume $I(X;Y|Z) > 0$, and conclude the following two statements:

Fact 1. We have $0 < P_R(0) < 1$, and Alice and Bob can generate binary random variables $\overline{X}$ and $\overline{Y}$ from X and Y with positive probability such that

$$P_{\overline{X}|R}(0,0) > P_{\overline{X}|R}(1,0) \tag{5.17}$$

and

$$P_{\overline{X}|R}(0,1) < P_{\overline{X}|R}(1,1) \tag{5.18}$$

(as well as the corresponding inequalities when replacing $\overline{X}$ by $\overline{Y}$) hold.

Fact 2. The random variable Z, together with some specific additional information U, corresponds to a random variable $\overline{Z}$ obtained by sending R over a symmetric binary erasure channel with positive erasure probability.

Because of Theorems 5.5 and 5.6, Facts 1 and 2 together imply

$$S(X;Y||Z) > 0 \ .$$

Proof of Fact 1. Obviously, $0 < P_R(0) < 1$ holds. We show that

$$\mathrm{P}_X[f_{X|R=0}(x) \neq f_{X|R=1}(x)] > 0 \ . \tag{5.19}$$

Otherwise, if $f_{X|R=0}(x) = f_{X|R=1}(x)$ with probability 1, then

$$\begin{aligned}
f_{XY|Z=z} &= f_{XY|R=0} \cdot P_{R|Z=z}(0) + f_{XY|R=1} \cdot P_{R|Z=z}(1) \\
&= f_{X|R=0} \cdot f_{Y|R=0} \cdot P_{R|Z=z}(0) \\
&\quad + f_{X|R=1} \cdot f_{Y|R=1} \cdot P_{R|Z=z}(1) \\
&= f_{X|R=0} \cdot (f_{Y|R=0} \cdot P_{R|Z=z}(0) + f_{Y|R=1} \cdot P_{R|Z=z}(1)) \\
&= f_{X|Z=z} \cdot f_{Y|Z=z}
\end{aligned}$$

with probability 1. Hence $I(X;Y|Z=z) = 0$ for all z, and $I(X;Y|Z) = 0$, which is a contradiction. Therefore (5.19) holds. We define

$$A_0 := \{x \mid f_{X|R=0}(x) > f_{X|R=1}(x)\}$$

and

$$A_1 := \{x \mid f_{X|R=0}(x) < f_{X|R=1}(x)\} \ .$$

Then A_0 and A_1 are disjoint measurable sets, with

$$P_{X|R=0}(A_0) > P_{X|R=1}(A_0) \tag{5.20}$$

and

$$P_{X|R=0}(A_1) < P_{X|R=1}(A_1)$$

(where $P_{X|R=0}(A_0)$ stands for $\int_{A_0} f_{X|R=0}\,dx$). Inequality (5.20) holds because if $P_{X|R=0}(A_0) = P_{X|R=1}(A_0)$, then

$$\int\limits_{A_0} f_{X|R=0}(x) - f_{X|R=1}(x)\,dx = 0$$

(and the same holds for A_1, because the $f_{X|R=i}$ are densities of normed probability measures). It is a well-known fact from measure theory that the integral of a strictly-positive function on a set with non-vanishing measure is also strictly positive, and hence A_0 and A_1 would be null sets, which is a contradiction to (5.19). For the random variable Y, two sets B_0 and B_1 can be defined similarly.

In analogy to the case of discrete random variables (see Section 3), Alice and Bob can obtain new random variables $\hat{X}$ and $\hat{Y}$ by restriction of the ranges of X and Y to $A_0 \cup A_1$ and $B_0 \cup B_1$, respectively, and send these random variables $\hat{X}$ and $\hat{Y}$ over two channels in order to generate binary random variables $\overline{X}$ and $\overline{Y}$ such that $\overline{X} = 0$ if $\hat{X} \in A_0$ and $\overline{X} = 1$ if $\hat{X} \in A_1$ (and analogously for $\overline{Y}$). It is obvious that (5.17) and (5.18) hold, as well as the corresponding inequalities for $\overline{Y}$.

Proof of Fact 2. From

$$I(X;Y|Z) = \int I(X;Y|Z=z) \cdot f_Z(z)\,dz > 0$$

we conclude that there is a measurable set D with $\mu(D) > 0$ (where μ denotes the Lebesgue measure of $\mathbf{R}$) and

$$I(X;Y|Z=z) > 0 \quad \text{for all} \quad z \in D\ . \tag{5.21}$$

Because of (5.21) we have both $f_{R|Z=z}(0) > 0$ and $f_{R|Z=z}(1) > 0$ for all $z \in D$. (If for example $f_{R|Z=z}(0) = 0$, then

$$\begin{aligned} f_{XY|Z=z} &= f_{XY|R=0} \cdot P_{R|Z=z}(0) + f_{XY|R=1} \cdot P_{R|Z=z}(1) \\ &= f_{XY|R=1} = f_{X|R=1} \cdot f_{Y|R=1} = f_{X|Z=z} \cdot f_{Y|Z=z} \end{aligned}$$

and $I(X;Y|Z=z) = 0$.) For every n, let D_n be the (measurable) set of all z in D such that $f_{R|Z=z}(0) \geq P_R(0)/n$ and $f_{R|Z=z}(1) \geq P_R(1)/n$. Then $D = \cup D_n$, and $\mu(D) > 0$ implies

$$0 < \mu(D) = \mu(\cup D_n) \leq \sum_n \mu(D_n)\ .$$

We conclude that there exists n_0 such that $\mu(D_{n_0}) > 0$.

Let U be a random variable such that $U = R$ with probability 1 if $z \notin D_{n_0}$, and with probability

$$\frac{f_{R|Z=z}(i) - P_R(i)/n_0}{f_{R|Z=z}(i)}$$

if $z \in D_{n_0}$ and $R = i$ (and such that otherwise, U gives no information about R). The random variable Z, together with this side information U, corresponds to a random variable $\overline{Z}$, generated from R by a symmetric binary erasure channel with erasure probability $\mu(D_{n_0})/n_0 > 0$. □

5.4 Towards the General Case: Protocol RC is not Optimal

In this section we assume that X and Y are completely general random variables, and that Eve obtains her information from a random variable that is generated by sending X and Y over erasure channels. The advantage of considering such a scenario is that it is less difficult to analyze than the completely general situation. Additionally, more general situations can be reduced, by the methods of Theorem 5.5, to such a scenario (with respect to the question whether secret-key agreement is possible).

We approach the general situation by studying two extremal cases. For Scenario EC1 below, the statement of Conjecture 5.1 is shown to be true, whereas for Scenario EC2, this problem remains open. Also for Scenario EC2, we prove that Protocol RC is not optimal. A new protocol is shown to be strictly stronger with respect to the possibility in principle of secret-key agreement.

Scenario EC1 (Erasure Channel I). The random variables X and Y are binary and distributed according to

$$P_{XY}(0,0) = P_{XY}(1,1) = \frac{1-\alpha}{2} \ , \quad P_{XY}(0,1) = P_{XY}(1,0) = \frac{\alpha}{2} \quad (5.22)$$

for some $\alpha < 1/2$. The random variable Z is generated by sending $[X, Y]$ over an erasure channel with positive erasure probability $1 - r$.

Scenario EC2 (Erasure Channel II). The random variables X and

Y are distributed according to (5.22), and $Z = [Z_X, Z_Y]$, where Z_X and Z_Y are generated by sending X and Y over two independent erasure channels with positive erasure probabilities $1 - r_X$ and $1 - r_Y$, respectively.

The two scenarios are illustrated in Figure 5.5.

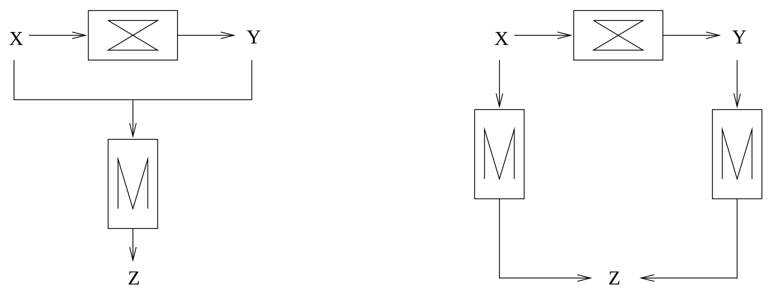


Figure 5.5: Scenarios EC1 and EC2. The boxes stand for binary symmetric and for erasure channels, respectively. (Note that the erasure channel in Scenario EC1 is not binary.)

Prior to the analysis of Scenarios EC1 and EC2 we show that under a condition which appears to be satisfied most likely if X, Y, and Z are general random variables with $I(X;Y{\downarrow}Z) > 0$, there exist random variables $\overline{X}$ and $\overline{Y}$, which can be generated from X and Y with positive probability, and side information U such that $\overline{X}$, $\overline{Y}$, and $[Z,U]$ correspond to one of the Scenarios EC1 or EC2. Theorem 5.5 then implies that $S(X;Y||Z) > 0$ if $S(\overline{X};\overline{Y}||[Z,U]) > 0$. The proof of the following lemma is related to one of the arguments in the proof of Theorem 5.6.

Lemma 5.13. *Let $I(X;Y) > 0$. Then Alice and Bob can generate symmetric binary random variables $\overline{X}$ and $\overline{Y}$ from X and Y with positive probability such that $\overline{X}$ and $\overline{Y}$ have a symmetric joint distribution as given in (5.22) for some $\alpha < 1/2$.*

Proof. Because X and Y are not statistically independent there exist $x, x' \in \mathcal{X}$ and $y, y' \in \mathcal{Y}$ satisfying

$$P_{X|Y}(x,y) > P_X(x) > P_{X|Y}(x,y') \tag{5.23}$$

and

$$P_{Y|X}(y,x) > P_Y(y) > P_{Y|X}(y,x') \ . \tag{5.24}$$

Alice and Bob can generate random variables $\hat{X}$ and $\hat{Y}$ by restricting the ranges of X and Y to $\{x,x'\}$ and $\{y,y'\}$, respectively. Then $\hat{X}$ is sent over the following channel (we can assume without loss of generality that $P_{\hat{X}}(x) \geq 1/2$):

$$P_{\overline{X}|\hat{X}}(0,x) = \frac{1}{2 \cdot P_{\hat{X}}(x)} \ , \quad P_{\overline{X}|\hat{X}}(1,x) = 1 - P_{\overline{X}|\hat{X}}(0,x) \ ,$$

$$P_{\overline{X}|\hat{X}}(1,x') = 1 \ , \quad P_{\overline{X}|\hat{X}}(0,x') = 0 \ .$$

It is obvious that $P_{\overline{X}}(0) = P_{\overline{X}}(1) = 1/2$. The symmetrically-distributed random variable $\overline{Y}$ can be obtained from $\hat{Y}$ in an analogous way. Then $\overline{X}$ and $\overline{Y}$ are distributed according to (5.22) with

$$\alpha = 1 - P_{\hat{X}\hat{Y}}(x,y)/(2 \cdot P_{\hat{X}}(x) \cdot P_{\hat{Y}}(y))$$

if $P_{\hat{Y}}(y) \geq 1/2$, and

$$\alpha = P_{\hat{X}\hat{Y}}(x,y')/(2 \cdot P_{\hat{X}}(x) \cdot P_{\hat{Y}}(y'))$$

if $P_{\hat{Y}}(y) < 1/2$. In both cases we have $\alpha < 1/2$ because of (5.23) and (5.24). □

If, for $\overline{X}$ and $\overline{Y}$ obtained as in Lemma 5.13, there exists $z \in \mathcal{Z}$ such that the conditional probabilities $P_{\overline{X}\,\overline{Y}|Z=z}(i,j)$ are positive for all $(i,j) \in \{0,1\}^2$, then there exists side information U such that U equals $[\overline{X},\overline{Y}]$ with some probability (that depends on $[\overline{X},\overline{Y}]$), but where U contains no information about $\overline{X}$ or $\overline{Y}$ otherwise, i.e., the pair $[Z,U]$ can be interpreted as being generated by sending $[\overline{X},\overline{Y}]$ over an erasure channel with positive erasure probability.

We conclude that very general situations can be reduced to Scenario EC1, in which Z is obtained by sending $[X,Y]$ over an erasure channel. In an analogous way, general distributions can be reduced to Scenario EC2. However, it appears to be difficult to decide in general which reduction leads to the strongest results.

5.4.1 Analysis of Scenario EC1

Scenario EC1 has been defined above as the symmetric situation where X and Y are distributed according to (5.22) with $\alpha < 1/2$, and where Z is obtained by sending $[X,Y]$ over an erasure channel with erasure probability $1-r$. We derive a condition for when Protocol RC (with parameter N) allows secret-key agreement. Bob's conditional error probability when guessing the bit sent by Alice, given that he accepts, is

$$\beta_N = \frac{1}{p_{a,N}} \cdot \alpha^N \leq \left(\frac{\alpha}{1-\alpha}\right)^N ,$$

where $p_{a,N} = \alpha^N + (1-\alpha)^N$ is the probability that Bob accepts the received block. Given that Bob accepts, Eve (using the optimal strategy) guesses the bit sent by Alice correctly unless she receives N times the erasure symbol Δ. In the latter case her error probability is $1/2$, independently of her strategy. Hence Eve's error probability, given that Bob accepts, is

$$\gamma_N = \frac{1}{2} \cdot (1-r)^N .$$

Using Lemma 5.9, we conclude that Protocol RC works and allows the generation of a secret key if

$$1 - r > \frac{\alpha}{1-\alpha} .$$

Theorem 5.14 shows that Protocol RC is optimal in Scenario EC1 in the sense that if *some* protocol allows secret-key agreement in principle, then Protocol RC also does. Another consequence of Theorem 5.14 is that Conjecture 5.1 is true in Scenario EC1.

Theorem 5.14. *In Scenario EC1, the following four conditions are equivalent.*

(i) $I(X;Y{\downarrow}Z) > 0$,

(ii) $1 - r > \alpha/(1-\alpha)$,

(iii) Protocol RC allows secret-key agreement,

(iv) $S(X;Y||Z) > 0$.

Proof. It remains to show that *(i)* implies *(ii)*, i.e., that $I(X;Y{\downarrow}Z) = 0$ holds whenever $1 - r \leq \alpha/(1-\alpha)$. Let the random variable $\overline{Z}$ be generated from Z by the following channel: if $Z = \Delta$, $Z = [0,1]$, or $Z = [1,0]$, then $\overline{Z} = \overline{\Delta}$. When $Z = [0,0]$ or $Z = [1,1]$, then $\overline{Z}$ is defined to be equal to Z with probability

$$\frac{1-2\alpha}{r(1-\alpha)} \quad (\leq 1) ,$$

and equal to $\overline{\Delta}$ otherwise.

We show that $I(X;Y|\overline{Z}) = 0$. It is obvious that

$$I(X;Y|\overline{Z} = [0,0]) = I(X;Y|\overline{Z} = [1,1]) = 0 .$$

Because of

$$P_{XY\overline{Z}}(0,1,\overline{\Delta}) = P_{XY\overline{Z}}(1,0,\overline{\Delta}) = \frac{\alpha}{2}$$

and

$$\begin{aligned} P_{XY\overline{Z}}(0,0,\overline{\Delta}) &= P_{XY\overline{Z}}(1,1,\overline{\Delta}) \\ &= \frac{1-\alpha}{2} \cdot (1-r) + \frac{1-\alpha}{2} \cdot r \cdot \left(1 - \frac{1-2\alpha}{r(1-\alpha)}\right) = \frac{\alpha}{2} , \end{aligned}$$

X and Y also are independent when given $\overline{Z} = \overline{\Delta}$, i.e.,

$$I(X;Y|\overline{Z} = \overline{\Delta}) = 0 .$$

□

Note that in contrast to Scenario BS, the condition $I(X;Y|Z) > 0$ is (necessary, but) *not* sufficient for $S(X;Y||Z) > 0$ in Scenario EC1, a fact that is a justification of the definition of intrinsic information.

5.4.2 Erasure Probability, Deviation from Independence, and Intrinsic Information

In this section we show that a property of the intrinsic information which is similar to the implication from *(i)* to *(ii)* in Theorem 5.14 can be proved also in the case of general random variables X and Y. Namely, we show that $I(X;Y{\downarrow}Z) = 0$ if Eve knows X and Y precisely with some positive probability, and if the joint distribution of X and Y is "too close" to an "independent distribution". We have to define an appropriate measure for the "deviation from independence" of the joint distribution P_{XY} of the two random variables X and Y.

Definition 5.3. Let X and Y be random variables with ranges $\mathcal{X}$ and $\mathcal{Y}$, respectively, and joint distribution P_{XY}. Let

$$F(P_{XY}) := \min_{Q_{XY}} \left(\max_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \left(\frac{P_{XY}(x,y)}{Q_{XY}(x,y)} \right) \cdot \max_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \left(\frac{Q_{XY}(x,y)}{P_{XY}(x,y)} \right) \right) ,$$

where the minimum is taken over all probability distributions Q_{XY} for which X and Y are statistically independent, and where we set $0/0 := 1$ and $c/0 := \infty$ for $c > 0$. The *deviation* $d_{ind}(P_{XY})$ *of* P_{XY} *from independence* is defined as

$$d_{ind}(P_{XY}) := 1 - \frac{1}{F(P_{XY})} ,$$

where we set $1/\infty := 0$. ○

For every P_{XY} we have

$$0 \leq d_{ind}(P_{XY}) \leq 1 , \tag{5.25}$$

with equality on the left-hand side of (5.25) if and only if the random variables X and Y are independent, and with equality on the right hand side of (5.25) if and only if there exist x and y such that $P_{XY}(x,y) = 0$ although $P_X(x) \cdot P_Y(y) \neq 0$. Furthermore,

$$d_{ind}(P_{XY}) \leq 1 - \frac{\min_{(x,y)\in\mathcal{X}\times\mathcal{Y}} P_{XY}(x,y)}{\max_{(x,y)\in\mathcal{X}\times\mathcal{Y}} P_{XY}(x,y)} .$$

This can be seen by taking the uniform distribution for Q_{XY}. When X and Y are distributed according to (5.22), then

$$d_{ind}(P_{XY}) = 1 - \frac{\alpha}{1-\alpha} = \frac{1-2\alpha}{1-\alpha} .$$

Theorem 5.15 implies that secret-key agreement is impossible under the surprisingly simple and intuitive condition that the probability that Eve reliably knows X and Y equals or exceeds $d_{ind}(P_{XY})$.

Theorem 5.15. *Let X and Y be arbitrary discrete random variables with joint distribution P_{XY}, and let Z be generated by sending $[X, Y]$ over an erasure channel with erasure probability $1-r$. Then*

$$r \geq d_{ind}(P_{XY})$$

implies $I(X;Y{\downarrow}Z) = 0$.

Proof. Let $r \geq d_{ind}(P_{XY}) = 1 - 1/F(P_{XY})$, i.e., $F(P_{XY}) \leq 1/(1-r)$. Then, from the definition of $F(P_{XY})$, we conclude that there exists a distribution Q_{XY}, corresponding to an independent distribution of X and Y, such that

$$(1-r) \cdot P_{XY}(x,y) \leq \lambda \cdot Q_{XY}(x,y) \leq P_{XY}(x,y) \tag{5.26}$$

for all $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and for some $0 \leq \lambda \leq 1$. We define the random variable $\overline{Z}$, which can be obtained from Z, as follows. If $Z = \Delta$, then $\overline{Z} = \overline{\Delta}$. Because $P_{XYZ}(x,y,\Delta) = (1-r) \cdot P_{XY}(x,y)$, and because of (5.26), $\overline{Z}$ can be defined to be equal to $\overline{\Delta}$ with some conditional probability when given $Z = [x,y]$, and $\overline{Z} = Z$ otherwise, such that

$$P_{XY\overline{Z}}(x,y,\overline{\Delta}) = \lambda \cdot Q_{XY}(x,y) \ . \tag{5.27}$$

This can be done for all pairs (x,y), and (5.27) implies $P_{XY|\overline{Z}=\overline{\Delta}} = Q_{XY}$, i.e., that X and Y are independent when given $\overline{Z} = \overline{\Delta}$. Hence

$$\begin{aligned} I(X;Y{\downarrow}Z) &\leq I(X;Y|\overline{Z}) = \sum_{\overline{z}} P_{\overline{Z}}(\overline{z}) \cdot I(X;Y|\overline{Z} = \overline{z}) \\ &= P_{\overline{Z}}(\overline{\Delta}) \cdot I(X;Y|\overline{Z} = \overline{\Delta}) = 0 \ , \end{aligned}$$

and the theorem is proved. □

5.4.3 Analysis of Scenario EC2

In this section we analyze Scenario EC2. Let α be the probability that $X \neq Y$, and let r_X and r_Y be the probabilities that Eve does *not* receive the erasure symbol from her (independent) channels. We assume here that $r_Y \geq r_X$, i.e., that Alice's information is better protected that Bob's. For fixed α and r_Y, we prove three different upper bounds on r_X with the property that secret-key agreement is possible if r_X is smaller than at least one of these bounds. Moreover, a new protocol is presented that is applicable for a larger class of distributions P_{XYZ} than Protocol RC, hence proving that Protocol RC is not optimal (with respect to the possibility or impossibility of secret-key agreement) for Scenario EC2.

The first upper bound on r_X comes from Theorem 4.7, according to which the secret-key rate is positive if $I(X;Y) > I(X;Z)$. This condition is equivalent to

$$H(X|Y) < H(X|Z) \; , \tag{5.28}$$

where

$$\begin{aligned} H(X|Y) &= h(\alpha) \\ H(X|Z) &= (1-r_X)(1-r_Y) + (1-r_X)r_Y h(\alpha) \; . \end{aligned}$$

Lemma 5.16. *In Scenario EC2, $S(X;Y||Z)$ is strictly positive if*

$$r_X < 1 - \frac{h(\alpha)}{1 - r_Y + r_Y h(\alpha)} \; . \tag{5.29}$$

If Lemma 5.16 does not apply, in some cases one can prove that secret-key agreement is nevertheless possible by using Protocol RC. When the block length is N, the probability p_{10} that Bob accepts and receives the bit sent by Alice incorrectly, and that Eve receives this bit correctly, is upper bounded by α^N (and of order $\Theta(\alpha^N)$). On the other hand, the probability p_{01} that Bob accepts and receives the correct bit, and that Eve guesses the bit incorrectly with the optimal strategy, equals

$$p_{01} = \frac{1}{2}(1-\alpha)^N(1-r_X)^N(1-r_Y)^N \; .$$

The reason for this is that if Eve receives only erasure symbols, her error probability about the bit sent by Alice is, independently of her strategy, equal to $1/2$ (and 0 otherwise). Finally, the probability p_{11} that Bob accepts, and that both Bob and Eve receive the bit incorrectly satisfies

$$p_{11} \le \alpha^N(1-r_X)^N \; .$$

Hence Bob's error probability is of order $\Theta(\alpha^N)$, whereas Eve's error probability is of order $\Theta(((1-\alpha)(1-r_X)(1-r_Y))^N + (\alpha(1-r_X))^N)$.

From this and from Lemma 5.9 we can conclude that Protocol RC works if and only if

$$\alpha < (1-\alpha)(1-r_X)(1-r_Y) , \tag{5.30}$$

and the following lemma is proved.

Lemma 5.17. *In Scenario EC2, Protocol RC allows secret-key agreement, and $S(X;Y||Z)$ is positive, if and only if*

$$r_X < 1 - \frac{\alpha}{(1-\alpha)(1-r_Y)} . \tag{5.31}$$

Note that this bound is strictly positive only if $1-r_Y > \alpha/(1-\alpha)$. This is the same condition as in Theorem 5.14 of the previous section.

We remark that each of the expressions in (5.29) and (5.31) can be greater than the other. If r_Y is constant and $\alpha \to 0$, the expression of (5.31) is greater, whereas if $r_Y = \alpha/(1-\alpha)$, the expression of (5.31) equals 0, and the expression of (5.29) is greater than 0 for all $\alpha < 1/2$.

Intuitively, the repeat-code protocol does not appear to be very appropriate in a situation where Eve has perfect access to X or Y with some positive probability, because revealing one bit of a repeat code block means revealing the entire block. It is therefore conceivable that a protocol using blocks which contain a certain fraction (less than half) of incorrect bits is better here, although the effect that Alice's and Bob's bits become more reliable is weaker in this case. The advantage is that if Eve has reliable access to one bit (or a small number of bits) of a block, then she does not automatically know the whole block. We will show that in Scenario EC2 the following protocol is superior to Protocol RC.

Protocol RCE (Repeat Code with Errors). Bob randomly chooses a bit C and a random N-bit block $[C_1, \ldots , C_N]$ such that tN of the bits are equal to C, and $(1-t)N$ of the bits are equal to $\overline{C} := C \oplus 1$ (where $t > 1/2$ is a parameter, and tN is an integer). As in Protocol RC, Bob computes $[C_1 \oplus Y_1, \ldots , C_N \oplus Y_N]$ and sends this block over the public channel. Alice computes $[(C_1 \oplus Y_1) \oplus X_1, \ldots , (C_N \oplus Y_N) \oplus X_N]$ and accepts only if this equals $[0, 0, \ldots , 0]$ or $[1, 1, \ldots , 1]$.

The analysis of the protocol shows that it is advantageous for Alice and Bob when Bob, and not Alice, is the sender of the bit in Protocol RCE if $r_Y \geq r_X$. Note that Protocol RCE corresponds to Protocol RC for the choice $t = 1$. Protocol RCE is, as Protocol RC, efficient in terms of computation but wasteful with respect to the achievable rate of generated secret key. An efficiency improvement similar to the parity-check version of Protocol RC [40] exists also for Protocol RCE.

Theorem 5.18 gives a sufficient condition for protocol RCE to work. The analysis of Protocol RCE, leading to the proof of Theorem 5.18, is quite technical. It is a surprising fact that the parameter t must typically be chosen only slightly greater than $1/2$ (whereas it is obvious that the choice $t = 1/2$ is completely useless).

Theorem 5.18. *In Scenario EC2, Protocol RCE allows for secret-key agreement, and $S(X;Y||Z)$ is hence positive, if*

$$r_X < \frac{2\left(1 - \frac{\alpha}{1-\alpha}\right)^2 (1 - r_Y)}{5 - 4r_Y} \tag{5.32}$$

(when $1 - \alpha/(1-\alpha) \leq 5/4 - r_Y$), or if

$$r_X < (1 - r_Y)\left(1 - \frac{\alpha}{1-\alpha} - \frac{1 - r_Y}{2} - \frac{1}{8}\right) \tag{5.33}$$

(when $1 - \alpha/(1-\alpha) > 5/4 - r_Y$), respectively.

Theorem 5.18 shows that in Scenario EC2, Protocol RCE is strictly better than Protocol RC, which is therefore not optimal. It is easy to see that the upper bounds of (5.32) and (5.33) are greater than the bounds given by (5.29) and (5.31) in many cases. We consider two examples.

If r_Y is constant and $\alpha \to 1/2$, then the bound given in (5.29) tends to 0 much faster than (5.32) (which applies in this situation). The bound of (5.31) is even negative. On the other hand, if $\alpha = 3/7$, and $r_Y \to 1$, then (5.29) is smaller than (5.33) (which applies here). The bound (5.31) is negative again.

We now analyze the performance of Protocol RCE in Scenario EC2 and prove Theorem 5.18. Let the protocol parameter t be fixed, and let

$$K = K(t) := \frac{1}{4t - 2} .$$

We first compute the conditional probability β_N that Alice receives the bit sent by Bob incorrectly, given that she accepts:

$$\beta_N = \frac{\alpha^{tN}(1-\alpha)^{(1-t)N}}{(1-\alpha)^{tN}\alpha^{(1-t)N} + \alpha^{tN}(1-\alpha)^{(1-t)N}} \leq \left(\frac{\alpha}{1-\alpha}\right)^{N/(2K)} . \tag{5.34}$$

Eve's conditional error probability γ_N, given that Alice accepts, is lower bounded by $1/2$ times the probability that Eve receives exactly sN of the tN correct bits of Bob's block (more precisely, that she receives the corresponding realizations of Y from the erasure channel, and erasure symbols for the other $(t-s)N$ realizations of Y that also correspond to correct bits in Bob's block), and exactly the same number of incorrect bits, and that she learns nothing about Alice's block (i.e., about all the realizations of X) because she receives only erasure symbols from that channel. This is a lower bound on γ_N because in this case, Eve's error probability for guessing Bob's bit is equal to $1/2$, and is independent of her strategy. This holds for all possible s, and hence the maximum of this probability, taken over all $0 \leq s \leq 1-t$, gives also a lower bound.

$$\begin{aligned} \gamma_N \quad \geq \quad & \frac{1}{2} \cdot \max_{0 \leq s \leq (1-t)} \left\{ \binom{tN}{sN} (r_Y)^{sN} (1-r_Y)^{(t-s)N} \right. & (5.35) \\ & \left. \cdot \binom{(1-t)N}{sN} (r_Y)^{sN} (1-r_Y)^{(1-t-s)N} \right\} \cdot (1-r_X)^N . \end{aligned}$$

The next lemma gives a simpler lower bound that can be derived from the bound (5.35) by determining its asymptotic behavior.

Lemma 5.19. *The lower bound (5.35) implies that*

$$\gamma_N^{2K/N} \geq 1 - \frac{1}{4K} - \frac{1}{16(1-r_Y)K} - 2Kr_X \tag{5.36}$$

if $r_Y/2 \leq 1-t$ holds, and if N is sufficiently large.

Proof. First note that $r_Y/2 \leq 1-t$ means that $s := r_Y/2$ is a possible choice (in fact, and not surprisingly, this is the optimal choice). From Stirling's formula we can conclude that

$$\binom{aN}{bN} \geq \frac{C}{\sqrt{N}} \cdot \left(\frac{a^a}{b^b (a-b)^{a-b}} \right)^N$$

for some constant C. The binomial coefficients in (5.35) can be replaced by the corresponding expressions, and a straight-forward computation leads to the following asymptotic behavior of the lower bound on γ_N.

$$\begin{aligned}
\gamma_N^{2K/N} &\geq \left(1 - \frac{1}{2K}\right)^K \cdot \left(1 + \frac{1}{2K}\right)^K \\
&\quad \cdot \left(1 + \frac{1}{4(1-r_Y)K}\right)^{(1-r_Y)K} \\
&\quad \cdot \left(1 - \frac{1}{4(1-r_Y)K}\right)^{(1-r_Y)K} \cdot (1-r_X)^{2K} \\
&\geq \left(1 - \frac{1}{4K}\right) \cdot \left(1 - \frac{1}{16(1-r_Y)K}\right) \cdot (1 - 2Kr_X) \\
&\geq 1 - \frac{1}{4K} - \frac{1}{16(1-r_Y)K} - 2Kr_X
\end{aligned}$$

for sufficiently large N. □

The bound (5.36) in the above lemma holds for all K that correspond to a protocol parameter t which satisfies $r_Y/2 \leq 1 - t$. This condition is equivalent to

$$\frac{1}{2K} \leq 1 - r_Y \ . \tag{5.37}$$

The idea of the proof of Theorem 5.32 is to find the best choice for K (i.e., the best choice of t in Protocol RCE) with respect to the fixed parameters α and r_Y, and such that (5.37) holds. This optimal choice of K leads to an upper bound on r_X, such that if r_X is smaller than this bound, then Protocol RCE works for secret-key agreement. This is exactly the upper bound stated in the theorem.

Proof of Theorem 5.18. According to (5.34) and (5.36), Protocol RCE (with parameter t) works for secret-key agreement if

$$\gamma_N^{2K/N} \geq 1 - \frac{1}{4K} - \frac{1}{16(1-r_Y)K} - 2Kr_X > \frac{\alpha}{1-\alpha} \geq \beta_N^{2K/N} \ , \tag{5.38}$$

and if the condition (5.37) also holds. The reason is that (5.38) implies that Eve's error probability about the bit sent by Bob is asymptotically greater than Alice's error probability for $N \to \infty$. Lemma 5.9 states that this is sufficient for the possibility of secret-key agreement by public discussion. Let $\delta := 1 - \alpha/(1-\alpha)$. Then (5.38) is satisfied if

$$r_X < \frac{\delta}{2K} - \frac{1}{8K^2}\left(1 + \frac{1}{4(1-r_Y)}\right) . \tag{5.39}$$

This bound depends on K, and from (5.39) we can determine the optimal choice for K (and hence the optimal choice of the protocol parameter t). The only restriction is that the choice must be compatible with (5.37). It is easy to see that the expression on the right of (5.39) is maximal for

$$K = K_0 := \frac{1}{\delta} \cdot \left(\frac{1}{2} + \frac{1}{8(1-r_Y)}\right) .$$

It is somewhat surprising that if δ is small and $r_Y \approx 1$ (i.e., in a situation which is not advantageous to Alice and Bob) K must be large, and this means that t is only slightly greater than $1/2$ (whereas the choice $t = 1/2$ is obviously the worst possible choice). Choosing $K = K_0$ is compatible with (5.37) if $\delta \geq 5/4 - r_Y$. Then the condition (5.39) is

$$r_X < \frac{2\delta^2(1-r_Y)}{5-4r_Y} .$$

If $\delta > 5/4 - r_Y$, the condition (5.37) is not fulfilled for $K = K_0$. For $K = K_0' := 1/(2-2r_Y)$ (the smallest choice for K that satisfies (5.37)) the right-hand side of (5.39) equals

$$(1-r_Y)\left(\delta - \frac{1-r_Y}{2} - \frac{1}{8}\right) .$$

This proves Theorem 5.18. □

Note that the main objective of the above analysis of Protocol RCE is to show that it leads to a strict improvement of Protocol RC, rather than to characterize the performance of the protocol completely. In particular, the bounds of Theorem 5.18 are not tight for two reasons. First, it is not necessary to choose t such that $r_Y/2$ is a possible choice for s, as done in the proof of Lemma 5.19. Secondly, we have compared Alice's error probability with Eve's conditional error probability, given that Alice's bit is correct. Eve's error probability, given that *Alice accepts*, is greater, because, given that *Alice does not receive the correct bit*, it is more likely that Eve's bit is also incorrect. However, it appears to be difficult to determine Eve's optimal strategy of guessing the bit, and hence to compute the *exact* error probability of her guess. Note that with an optimal analysis, Protocol RCE would clearly turn out to be at least as good as Protocol RC in *any* situation, because Protocol RC is a special case of Protocol RCE and corresponds to the choice $t = 1$. It is finally conceivable that the above results can be improved when a block protocol is used in which both Alice and Bob (and not only Bob) have a block that is not composed by N times the same bit. However, such a protocol appears to be much more difficult to analyze.

Finally, we give a pessimistic bound on r_X for Scenario EC2. As in the previous section we derive a condition here for the fact that $I(X;Y{\downarrow}Z) = 0$.

Theorem 5.20. *In Scenario EC2, $I(X;Y{\downarrow}Z) = 0$ holds whenever*

$$r_X \geq \frac{(1-r_Y)(1-2\alpha)}{r_Y\alpha + 2(1-r_Y)((1-\alpha)\sqrt{1-2\alpha} - (1-2\alpha)) + (1-r_Y)(1-2\alpha)} . \tag{5.40}$$

Remark. Of course the bound on r_X given in (5.40) is greater than the bounds (5.29), (5.31), and (5.32) (or (5.33), respectively) for all possible choices of α and r_Y.

Proof of Theorem 5.20. We show that if (5.40) is satisfied, then a channel, characterized by $P_{\overline{Z}|Z}$, can be constructed such that $I(X;Y|\overline{Z}) = 0$. The only $z \in \mathcal{Z}$ with $I(X;Y|Z = z) > 0$ is $z = [\Delta, \Delta]$, and the event $Z = [\Delta, \Delta]$ has probability $(1 - r_X)(1 - r_Y)$. The idea of the proof is to split this into three events $\overline{Z} = \Delta_1$, $\overline{Z} = \Delta_2$, and $\overline{Z} = \Delta_3$ (where $\overline{Z} = \Delta_i$ can also occur if $Z \neq [\Delta, \Delta]$) such that $I(X;Y|\overline{Z} = \Delta_i) = 0$ for $i = 1, 2, 3$. More precisely, the random variable $\overline{Z}$ will be defined such that $\overline{Z} = \Delta_1$ is possible not only if $Z = [\Delta, \Delta]$, but also if $Z = [0, 1]$ and $Z = [1, 0]$, whereas $\overline{Z} = \Delta_2$ is also possible if $Z = [0, \Delta]$ and $Z = [\Delta, 0]$, and finally $\overline{Z} = \Delta_3$ also if $Z = [1, \Delta]$ and $Z = [\Delta, 1]$. We determine the maximal possible probability of $Z = [\Delta, \Delta]$ which allows that this event can completely be split.

We define the random variable $\overline{Z}$ as follows, by giving the joint distribution with Z:

$$\begin{aligned} P_{\overline{Z}Z}(\Delta_1, [\Delta, \Delta]) &= \mu \cdot \frac{r_X r_Y \alpha}{1 - 2\alpha} \\ P_{\overline{Z}Z}(\Delta_1, [0, 1]) &= P_{\overline{Z}Z}(\Delta_1, [1, 0]) \;=\; \mu \cdot P_Z([0, 1]) \\ &= \mu \cdot P_Z([1, 0]) \;=\; \mu \cdot \frac{r_X r_Y \alpha}{2} \end{aligned}$$

$$
\begin{aligned}
P_{\overline{Z}Z}(\Delta_2,[\Delta,\Delta]) &= P_{\overline{Z}Z}(\Delta_3,[\Delta,\Delta]) \\
&= \mu \cdot r_X(1-r_Y)\left(\frac{1-\alpha}{\sqrt{1-2\alpha}}-1\right) \\
P_{\overline{Z}Z}(\Delta_2,[\Delta,0]) &= P_{\overline{Z}Z}(\Delta_2,[0,\Delta]) \;=\; P_{\overline{Z}Z}(\Delta_3,[\Delta,1]) \\
&= P_{\overline{Z}Z}(\Delta_3,[1,\Delta] \;=\; \mu \cdot P_Z([0,\Delta]) \\
&= \mu \cdot \frac{r_X(1-r_Y)}{2}
\end{aligned}
$$

and $\overline{Z}=Z$ otherwise. The parameter $0 \leq \mu \leq 1$ is such that

$$
\sum_{i=1}^{3} P_{\overline{Z}Z}(\Delta_i,[\Delta,\Delta]) = P_Z([\Delta,\Delta]) \ .
$$

Note that $\mu > 1$ is not possible. It is easy to see that the random variable $\overline{Z}$ can be obtained by sending Z over a channel specified by some conditional probability distribution $P_{\overline{Z}|Z}$. We show that $I(X;Y|\overline{Z}=\Delta_i)=0$ for $i=1,2,3$. For $i=1$ this follows from

$$
P_{XY\overline{Z}}(0,0,\Delta_1) = P_{XY\overline{Z}}(1,1,\Delta_1) = \mu \cdot \frac{r_X r_Y \alpha(1-\alpha)}{2(1-2\alpha)}
$$

and

$$
\begin{aligned}
P_{XY\overline{Z}}(0,1,\Delta_1) &= P_{XY\overline{Z}}(1,0,\Delta_1) \;=\; \mu \cdot \frac{r_X r_Y \alpha}{1-2\alpha} \cdot \frac{\alpha}{2} + \mu \cdot \frac{r_X r_Y \alpha}{2} \\
&= \mu \cdot \frac{r_X r_Y \alpha(1-\alpha)}{2(1-2\alpha)} \ .
\end{aligned}
$$

For $i=2$ and $i=3$ one can easily verify that

$$
P_{XY\overline{Z}}(0,0,\Delta_i) \cdot P_{XY\overline{Z}}(1,1,\Delta_i) = P_{XY\overline{Z}}(0,1,\Delta_i) \cdot P_{XY\overline{Z}}(1,0,\Delta_i)
$$

holds, which implies that X and Y are statistically independent, given that $\overline{Z}=\Delta_i$. If $\overline{z} \notin \{\Delta_1,\Delta_2,\Delta_3\}$ then $I(X;Y|\overline{Z}=\overline{z})=0$ obviously holds, and we conclude $I(X;Y|\overline{Z})=0$ and $I(X;Y{\downarrow}Z)=0$.

The maximal probability $P_Z([\Delta,\Delta])$ such that the event $Z=[\Delta,\Delta]$ can be completely split into $\overline{Z}=\Delta_i$ as above is the sum of the probabilities $P_{\overline{Z}Z}(\Delta_i,[\Delta,\Delta])$ $(i=1,2,3)$ with $\mu=1$. Thus the described construction of $\overline{Z}$ works if

$$
\begin{aligned}
&\frac{r_X r_Y \alpha}{1-2\alpha} + 2r_X(1-r_Y)\cdot\left(\frac{1-\alpha}{\sqrt{1-2\alpha}}-1\right) \\
&\quad\geq\; P_Z([\Delta,\Delta]) \\
&\quad=\; (1-r_X)(1-r_Y) \ , \qquad (5.41)
\end{aligned}
$$

and this is equivalent to (5.40). □

Remark. Note that the condition given in the lemma is sufficient, but not necessary for $I(X;Y\downarrow Z) = 0$. If $r_X \neq r_Y$, a better bound can be achieved when $Z = [0, \Delta]$ and $Z = [\Delta, 0]$ (as well as $Z = [1, \Delta]$ and $Z = [\Delta, 1]$) are not transformed symmetrically to $\overline{Z} = \Delta_2$ ($\overline{Z} = \Delta_3$), but each with the maximal possible probability, i.e., $r_X(1-r_Y)/2$ and $(1-r_X)r_Y/2$, respectively. The condition (5.20) for $I(X;Y\downarrow Z) = 0$ can then be replaced by the better, but more complicated condition

$$\frac{r_X r_Y \alpha}{1-2\alpha} - r_X(1-r_Y) - r_Y(1-r_X) + \sqrt{T} \geq (1-r_X)(1-r_Y) \ ,$$

where

$$\begin{aligned} T \quad := \quad & r_X^2(1-r_Y)^2 + r_Y^2(1-r_X)^2 \\ & + \left(2 + \frac{4\alpha^2}{1-2\alpha}\right) r_X(1-r_X) r_Y(1-r_Y) \ . \end{aligned}$$

5.5 Discussion

We have investigated the question whether the secret-key rate is positive or zero or, equivalently, whether secret-key agreement is possible in principle, in certain special but natural settings. By completely or partially analyzing a number of scenarios, we have found evidence suggesting that the possibility of secret-key agreement is equivalent to the positivity of a new information-theoretic quantity, the intrinsic conditional information. However, the problem of completely characterizing the distributions P_{XYZ} for which $S(X;Y||Z) > 0$ holds, i.e., for which secret-key agreement is possible in principle, has not been fully solved yet (see Figure 5.6).

For Scenario EC2 discussed above for instance, the resulting (sufficient but not necessary) conditions for $I(X;Y\downarrow Z) = 0$ and for the presented protocols for secret-key agreement to be successful are not exactly complementary. (This is true although both the optimistic and pessimistic bounds of Theorems 5.18 and 5.20 can be slightly improved by a better but more complicated analysis.) It is an open problem to derive *necessary and sufficient* conditions for either $I(X;Y\downarrow Z) = 0$ and $S(X;Y||Z) > 0$, and to decide whether Conjecture 5.1 also holds in Scenario EC2, and in general.

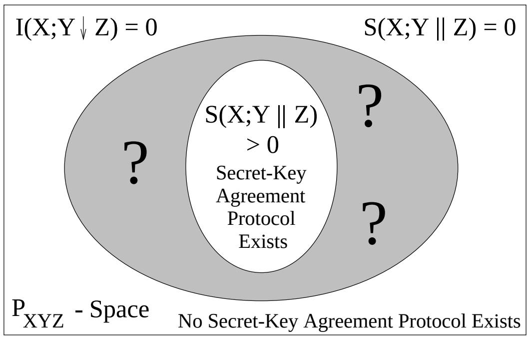


Figure 5.6: Characterizing when Secret-Key Agreement is Possible

Chapter 6

Key Agreement Secure Against Active Adversaries

In this chapter we study secret-key agreement secure not only against passive wire-tappers, but also against active opponents trying to insert, modify, or delete messages sent over the public channel. Equivalently, the model studied in the previous chapters is modified by dropping the condition that the noiseless channel connecting the legitimate partners must be authentic.

In Section 6.1, the requirements to a protocol for secret-key agreement secure against active adversaries are defined. Section 6.2 contains some impossibility results, e.g., the intuitive fact is made precise that in case of a completely insecure channel, the legitimate partners need some different kind of advantage, in terms of the distribution P_{XYZ}, for the possibility of secret-key agreement. This advantage can be formulated in terms of a certain "non-simulatability" condition, an intuitive and efficiently-verifiable version of which is given in Section 6.2.4. In Section 6.3, the important special case of privacy amplification secure against active adversaries is investigated. In this context, two problems are studied that are of proper interest. The first one is to give lower bounds on the Rényi entropy or min-entropy of parts of a string, depending on the entropy of the entire string. The derived results partially depend on upper bounds on the reduction of Rényi entropy or min-entropy when side information is given (see also [19], [22]). The second problem is identification and message authentication by using an only partially-secret key. A new, interactive way of authentication is presented that enjoys better properties in this case than the known authentication methods. An alternative method for interactive message authentication, for which the required size of the secret key is substantially smaller, was given by Gemmell and Naor [42]. Our method however achieves different goals. More precisely, our technique is such that a partially-secret key is sufficient, and that only a small number of communication bits, hence information leaked to the adversary, is required. This enables us to use the same key repeatedly many times.

The model analyzed in this chapter has first been presented and studied by Maurer [59], who considered an apparently significantly but in fact only slightly weaker type of security against active attacks. Some of the results of [59] are extended or corrected (see Sections 6.2 and 6.4). The parts of this chapter dealing with privacy amplification are based on, and extend, work already published in part in [74] and [119].

6.1 Protocol Definition

In this section we define what it means for a secret-key agreement protocol to be secure against active opponents. Clearly, one cannot expect that such a protocol is always successful if the adversary has full control over the public channel and can possibly block it completely, preventing any communication between Alice and Bob. Hence the best that can be achieved by such a protocol is that key agreement is successful when the adversary is passive, and that the parties realize failure due to an active attack and reject the outcome (see Figure 6.1).

To make things even worse, we cannot even expect that a malicious active attack is always detected by both partners: Because Eve can always block the last (significant) message sent (the one that would make the second party accept), she can leave Alice and Bob in opposite acceptance states if this is her objective. Theorem 6.1 and the proof thereof make this fact precise. However, we show how nearly as strong robustness can be defined and achieved. It can be required that (with high probability) either both Alice and Bob reject, or secret-key agreement is successful. Note that we cannot expect that both Alice and Bob accept in the latter case, but that they both compute the correct and secure key nonetheless. It seems that this is the strongest possible security one

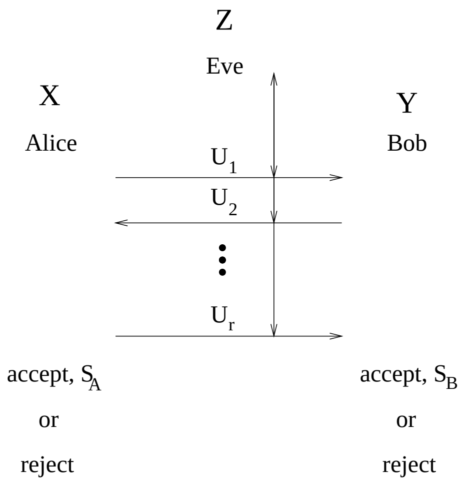


Figure 6.1: Unconditional Security Against Active Opponents

can achieve against active attackers, and that such protocols are what one actually has in mind when speaking about security against active adversaries in secret-key agreement. They have the property that no party can be misled by Eve: whenever a party accepts, then key agreement has been successful. One possible attempt to further strengthen this definition appears to be to introduce additional, "higher-level" acceptance states. However, the resulting robustness is not truly stronger than the one given.

Note that in [59], the security definition was, motivated by the mentioned impossibility result, weaker in the sense that only one party was requested to detect an active attack, i.e., to reject in case of failure. This definition is clearly too weak because the other party is not at all protected against active (e.g., impersonation) attacks. In contrast to this pessimistic definition, the protocols presented in [59] do in fact fulfill a stronger robustness: The key of an accepting party must be secure, though possibly not shared by the other party. We have chosen the even more restrictive yet more satisfactory definition here. However, every protocol for the weaker definition can be modified to satisfy our stronger requirements (see Section 6.4, where the results of [59] are corrected and adapted to our robustness definition).

An additional difference between the definitions given here and in [59] is that in case of an only passive adversary, it is not required that the parties accept *in all cases*, but *with high probability*. The protocols given in [59] for instance only satisfy this relaxed condition in general (see Section 6.4).

Definition 6.1. Assume the model of Definition 4.1, where the channel connecting Alice and Bob is not assumed to be authentic. The *communication phase* of the key agreement is defined as in Definition 4.1, where in the equalities (4.1) and (4.2) the C_i stand for the messages actually sent and received by the corresponding party (and possibly modified by the active opponent). After this phase, both Alice and Bob either accept or reject the outcome, and decide whether to compute a string S'_A or S'_B, respectively. An accepting party always computes a string. However, a party may also compute a string when rejecting the outcome of the protocol. The above decisions and the strings S'_A and S'_B are determined by X or Y, respectively, and by the messages sent and received.

Let r be an integer, and let $\varepsilon, \delta > 0$. A $(P_{XYZ}, r, \varepsilon, \delta)$*-protocol for secret-key agreement by communication over an insecure and non-authenticated channel* (or *robust* $(P_{XYZ}, r, \varepsilon, \delta)$*-protocol* for short) has the following properties. (Here and in the rest of this chapter, all the probabilities are taken over the random variables X, Y, and Z, and the additional coin tosses by Alice and Bob.)

1. Correctness and privacy. If Eve is a passive wire-tapper, then the probability that both Alice and Bob accept at the end of the protocol and secret-key agreement has been successful must be at least $1-\delta$. The latter is the event that S'_A and S'_B are r-bit strings satisfying

$$S'_A = S'_B \qquad \text{and} \qquad H(S'_A \,|\, CZ) \geq r - \varepsilon \ , \tag{6.1}$$

where $C := (C_1, C_2, \dots)$ summarizes the entire communication over the public channel.

2. Robustness. For every possible strategy of Eve, the probability that either *both* Alice and Bob reject the outcome of the protocol, or secret-key agreement has been successful, must be at least $1 - \delta$. ○

Remark. For simplicity, we have assumed in Definition 6.1 that a party's decisions of accepting or rejecting and whether to compute a string S'_A (S'_B), as well as these strings themselves, are determined by the final knowledge of the party, i.e., by X (or Y, respectively) and C, and do not depend on additional coin tosses. All the following results are valid (in the same or a closely-related form) also for protocols for which these decisions are probabilistic.

6.2 Impossibility Results

6.2.1 Perfect Synchronization is Impossible

Of course it is most desirable to use protocols for which Alice and Bob either both accept (and secret-key agreement is successful) or both reject with high probability. However, the following theorem states that such a synchronization cannot be achieved.

Theorem 6.1. *Assume that a robust $(P_{XYZ}, r, \varepsilon, \delta)$-protocol exists satisfying even the modified robustness property that with probability at least $1-\delta$, either both Alice and Bob reject, or both parties accept and secret-key agreement has been successful. Then suitable strings can be computed even without communication, i.e., there exist two functions f and g, mapping $\mathcal{X}$ and $\mathcal{Y}$ to $\{0,1\}^r$, respectively, such that $S'_A := f(X)$ and $S'_B := g(Y)$ satisfy (6.1) with probability at least $1-3\delta$.*

Proof. We assume that such a protocol exists although suitable S'_A and S'_B cannot be directly obtained from X and Y, respectively, with probability $1-3\delta$. Let us consider a particular execution of the protocol (i.e., fixed values of X, Y, and Z, and a particular choice of the coin tosses of Alice and Bob), where we assume first that Eve is passive. Let $C_1, C_2, \ldots, C_l$ be the messages sent during this execution. Let for all $2i+1 < l$ [$2j < l$] $\text{state}_A(2i+1)$ [$\text{state}_B(2j)$] be the state (`accept` or `reject`) of Alice [Bob] after sending the message C_{2i+1} [C_{2j}]. More precisely, this is the final state of the parties, provided that no further messages are received. Let analogously $\text{state}_A(l)$ and $\text{state}_B(l)$ be the final states of Alice and Bob in this execution. Because Eve is passive, we must have that $\text{state}_A(l) = \text{state}_B(l) =$ `accept` with probability at least $1-\delta$ by the protocol definition. Because suitable S'_A and S'_B cannot be computed directly from X and Y, respectively, except with probability smaller than $1-3\delta$, at least one message *must* be sent in the protocol or in other words, $\text{state}_B(0) =$ `reject`, with probability greater than $1-\delta-(1-3\delta) = 2\delta$.

From $\text{state}_B(0) = \texttt{reject}$ and $\text{state}_B(l) = \texttt{accept}$, an event that occurs with probability greater than

$$1 - (1 - 2\delta) - \delta = \delta$$

(by the union bound for the complementary event) we conclude that there exists j such that $\text{state}_B(2j) = \texttt{reject}$ and $\text{state}_B(2j + 2) = \texttt{accept}$ (or $\text{state}_B(l - 1) = \texttt{reject}$ and $\text{state}_B(l) = \texttt{accept}$). Hence we have either $\text{state}_B(2j) \neq \text{state}_A(2j+1)$ or $\text{state}_B(2j+2) \neq \text{state}_A(2j+1)$. If Eve blocks every message sent over the channel after C_{2j} (or C_{2j+1} in the second case), then Alice and Bob end up in opposite states. (In the case where $\text{state}_B(l-1) = \texttt{reject}$ and $\text{state}_B(l) = \texttt{accept}$, Eve can block the last message C_l.) We conclude that Eve can achieve disagreement in Alice's and Bob's states with probability greater than δ, which is a contradiction to the protocol definition. □

6.2.2 The Necessity of Interaction

In many cases, secret-key agreement protocols secure only against passive adversaries can be one-way. An example is secret-key agreement in cases where Alice and Bob have an initial advantage over Eve, in particular privacy amplification. In this section, however, we show that protocols secure against active adversaries must be interactive. This is not surprising since the party sending the first message must be protected from accepting when this message is deleted or modified.

Theorem 6.2. *Assume that a*

robust $(P_{XYZ}, r, \varepsilon, \delta)$-one-way-transmission protocol

exists. Then suitable strings can be computed even without communication with probability at least $1 - 2\delta$.

Proof. Note first that because Alice never receives a message, her decision is independent of Eve's strategy. The probability that Alice does not accept at the end is at most δ, even if Eve deletes the message sent. On the other hand, assume that the probability that key agreement is successful without any communication is smaller than $1 - 2\delta$. Then the probability that Alice accepts and no secure key can be computed nevertheless is greater than δ (by the union bound), which is a contradiction to the protocol definition. □

6.2.3 The Simulatability Condition

Clearly, secret-key agreement secure against active adversaries can only be possible if Alice and Bob have some advantage over Eve in terms of the distribution P_{XYZ}. More precisely, this advantage must be such that Eve cannot generate from Z a random variable $\overline{X}$ which Bob, knowing Y, is unable to distinguish from X (and vice versa). In [59], the following property of a distribution P_{XYZ} was defined.

Definition 6.2. [59] Let X, Y, and Z be random variables. Then X *is simulatable by* Z *with respect to* Y, denoted by

$$\mathrm{sim}_Y(Z \to X) ,$$

if there exists a conditional distribution $P_{\overline{X}|Z}$ such that $P_{\overline{X}Y} = P_{XY}$. ○

Another way of stating that $\mathrm{sim}_Y(Z \to X)$ holds is that there exists a random variable $\overline{X}$ such that $I(\overline{X}; Z|XY) = 0$, i.e., $XY \to Z \to \overline{X}$ is a Markov chain, and $P_{\overline{X}Y} = P_{XY}$. Later in this section we will describe simple criteria for (non-)simulatability in terms of the probabilities $P_{XYZ}(x, y, z)$. The following theorem states that a robust $(P_{XYZ}, r, \varepsilon, \delta)$-protocol can only exist if both X and Y are not simulatable by Z with respect to each other. In the scenario in which the parties obtain repeated realizations of the same random experiment, this condition is also sufficient (see Section 6.4).

Theorem 6.3. *Let X, Y, and Z be random variables with distribution P_{XYZ}. If either $\mathrm{sim}_Y(Z \to X)$ and $r > 5\delta r + h(4\delta) + \varepsilon$ hold, or $\mathrm{sim}_X(Z \to Y)$ and $r > 3\delta r + h(2\delta) + \varepsilon$ hold, then there exists no robust $(P_{XYZ}, r, \varepsilon, \delta)$-protocol.*

Remark. Notice that the asymmetry in the conditions with respect to Alice and Bob comes from the asymmetry in the definition of the secret-key rate (Definition 4.2; see also the subsequent remark).

Proof of Theorem 6.3. We assume that $\mathrm{sim}_Y(Z \to X)$ and $r > 5\delta r + h(4\delta) + \varepsilon$ hold, and that a robust $(P_{XYZ}, r, \varepsilon, \delta)$-protocol exists nevertheless. Because Bob cannot distinguish between Alice and an impersonating Eve, the probability that he accepts and that there was no interaction with Alice is at least $1 - \delta$ (i.e., the same as for the passive-adversary case). On the other hand, the probability that Bob accepts and that key agreement was not successful must be upper bounded by δ.

Hence, by the union bound, the probability that Bob accepts and that secret-key agreement has been successful even without interaction with Alice, is at least $1 - 2\delta$. We conclude that Alice (hence also Eve, who can simulate Alice towards Bob) and Bob can compute strings S'_A, $\overline{S'_A}$, and S'_B from X, Z, and Y, respectively, such that

$$\mathrm{P}[S'_A \neq S'_B] \leq 2\delta$$

and

$$\mathrm{P}[\overline{S'_A} \neq S'_B] \leq 2\delta \ ,$$

hence

$$\mathrm{P}[S'_A \neq \overline{S'_A}] \leq 4\delta \ . \tag{6.2}$$

On the other hand, the condition

$$(1-\delta)r - \varepsilon \leq H(S'_A \,|\, Z) \leq H(S'_A \,|\, \overline{S'_A}),$$

which is a consequence of the privacy condition in Definition 6.1, implies, by Fano's inequality (2.4) and inequality (6.2), that

$$r - \delta r - \varepsilon \leq h(4\delta) + 4\delta r \ ,$$

which is a contradiction to the second assumption. The argument for the case where $\mathrm{sim}_X(Z \to Y)$ holds is analogous. □

In the special scenario where all the parties obtain noisy versions of a binary signal over some independent channels (Scenario BS in Chapter 5), the condition $\mathrm{sim}_Y(Z \to X)$ is *equivalent* to the fact that $I(Y;Z) \geq I(Y;X)$. One of these two implications is always true, as Lemma 6.4 shows.

Lemma 6.4. *Let P_{XYZ} be a distribution such that $\mathrm{sim}_Y(Z \to X)$ holds. Then*

$$I(Y;Z) \geq I(Y;X) \ .$$

Proof. There exists a conditional distribution $P_{\overline{X}|Z}$ such that $P_{\overline{X}Y} = P_{XY}$. Hence we have

$$\begin{aligned} I(Y;Z) &= H(Y) - H(Y|Z) \\ &= H(Y) - H(Y|Z\overline{X}) \\ &\geq H(Y) - H(Y|\overline{X}) \\ &= I(Y;\overline{X}) \\ &= I(Y;X) \ , \end{aligned}$$

where the second equality holds since $I(Y;\overline{X}|Z) = 0$. □

However, the inverse implication is not true in general. To see this, consider the distribution P_{XYZ} corresponding to Scenario EC1 (see Chapter 5). Clearly, both $\mathrm{sim}_Y(Z \to X)$ and $\mathrm{sim}_X(Z \to Y)$ hold in this scenario if and only if

$$r + \frac{1-r}{2} \geq 1 - \alpha \ ,$$

i.e., $r \geq 1 - 2\alpha$. On the other hand, $I(Y;Z) \geq I(Y;X)$ holds exactly if

$$r \geq 1 - h(\alpha) \ .$$

It may be somewhat surprising that for $1-h(\alpha) \leq r < 1-\alpha$, Eve cannot simulate X with respect to Y although she has more information about Y than X provides.

An even stronger statement than Lemma 6.4 is the following connection between simulatability and the possibility of secret-key agreement by one-way communication: Whenever one of the random variables X or Y is simulatable, i.e., communication into one direction cannot be authenticated, then communication into this direction is necessary for secret-key agreement because the Csiszár-Körner protocol, from Bob to Alice, does not work in this case. Note that this is not a new result, but a direct conclusion from Theorem 6.3 (or, alternatively, from equation (4.19)). According to equation (4.19), another way of stating this result is that $\mathrm{sim}_Y(Z \to X)$ implies that there exists no random variable V such that $XZ \to Y \to V$ is a Markov chain and $I(V;X) - I(V;Z) > 0$ holds. For a further discussion of these connections, see Section 7.2.

As the example of Scenario BS shows, the fact that X and Y are simulatable does *not* imply that secret-key agreement against *passive* adversaries is not possible. However, the following statement, closely related to the inverse direction of this implication, is true.

Lemma 6.5. *Let P_{XYZ} be such that $I(X;Y\downarrow Z) = 0$. Then both $\mathrm{sim}_Y(Z \to X)$ and $\mathrm{sim}_X(Z \to Y)$ hold.*

Proof. Let $P_{\overline{Z}|Z}$ be a conditional distribution with

$$I(X;Y|\overline{Z}) = 0 \ . \tag{6.3}$$

Equivalently, $XY \to Z \to \overline{Z}$ is a Markov chain such that (6.3) holds, i.e., $X \to \overline{Z} \to Y$ is also a Markov chain. We show $\mathrm{sim}_Y(Z \to X)$. The proof that $\text{sim}_X(Z \to Y)$ holds is analogous. Let $\overline{X}$ be generated from $\overline{Z}$ by the channel $P_{\overline{X}|\overline{Z}} := P_{X|\overline{Z}}$. (This extends the Markov chain to $XY \to Z \to \overline{Z} \to \overline{X}$, hence $XY \to Z \to \overline{X}$ is also a Markov chain.) We show that $P_{\overline{X}Y} = P_{XY}$ holds. To see this, note that

$$P_{\overline{X}Y\overline{Z}} = P_{\overline{X}\overline{Z}} \cdot P_{Y|\overline{X}\overline{Z}} = P_{\overline{X}\overline{Z}} \cdot P_{Y|\overline{Z}}$$

and

$$P_{XY\overline{Z}} = P_{X\overline{Z}} \cdot P_{Y|X\overline{Z}} = P_{X\overline{Z}} \cdot P_{Y|\overline{Z}}$$

because $I(\overline{X}; Y|\overline{Z}) = I(X; Y|\overline{Z}) = 0$. From $P_{\overline{X}\overline{Z}} = P_{X\overline{Z}}$ (which is true by construction of $\overline{X}$) we conclude $P_{\overline{X}Y\overline{Z}} = P_{XY\overline{Z}}$, hence $P_{\overline{X}Y} = P_{XY}$. □

Theorem 6.6 is an immediate consequence of Lemma 6.5.

Theorem 6.6. *If Conjecture 5.1 is true, then* $S(X; Y||Z) > 0$ *holds whenever* $\text{sim}_Y(Z \to X)$ *and* $\text{sim}_X(Z \to Y)$ *are* not both true.

6.2.4 Characterizing Simulatability

According to the Theorems 6.3, 6.6, 6.23, and to Corollary 6.24, the simulatability condition is very useful in separating the cases where secret-key agreement is possible or impossible in the presence of (passive and) active adversaries. However, the characterization is a priori not practical at all because it depends on the existence of a particular channel (with certain properties) among the (uncountably-infinite) set of all discrete channels with given input and output alphabets. In this section, we consider the following questions:

- Is it, for a given distribution P_{XYZ}, possible to efficiently decide whether $\text{sim}_Y(Z \to X)$ holds?
- Furthermore, if the answer is "yes," is it possible to efficiently find a channel $P_{\overline{X}|Z}$ such that $P_{\overline{X}Y} = P_{XY}$ holds?

We start by analyzing an example.

Example 6.1. Let the distribution P_{XYZ} of the random variables X, Y, and Z with ranges $\mathcal{X} = \{x_1, x_2\}$, $\mathcal{Y} = \{y_1, y_2\}$, and $\mathcal{Z} = \{z_1, z_2, z_3\}$ be as follows:

$$\begin{array}{ll}
P_{XYZ}(x_1,y_1,z_1)=6/100\ , & P_{XYZ}(x_2,y_1,z_1)=4/100\ , \\
P_{XYZ}(x_1,y_1,z_2)=9/100\ , & P_{XYZ}(x_2,y_1,z_2)=6/100\ , \\
P_{XYZ}(x_1,y_1,z_3)=15/100\ , & P_{XYZ}(x_2,y_1,z_3)=10/100\ , \\
P_{XYZ}(x_1,y_2,z_1)=36/100\ , & P_{XYZ}(x_2,y_2,z_1)=4/100\ , \\
P_{XYZ}(x_1,y_2,z_2)=9/100\ , & P_{XYZ}(x_2,y_2,z_2)=1/100\ , \\
P_{XYZ}(x_1,y_2,z_3)=0\ , & P_{XYZ}(x_2,y_2,z_3)=0\ .
\end{array}$$

In order to decide whether $\text{sim}_Y(Z \to X)$ holds, we first consider the following representation of the (conditional) probabilities. We mark every symbol $x_i \in \mathcal{X}$ and every $z_j \in \mathcal{Z}$ with an empty or filled circle, where the size (or mass) of the circle corresponds to the probability $P_X(x_i)$ or $P_Z(z_j)$, and the position in the interval $[0,1]$ is given by the probability $P_{Y|X=x_i}(y_1)$ or $P_{Y|Z=z_j}(y_1)$, respectively (see Figure 6.2).

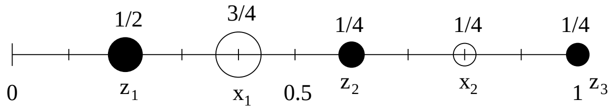


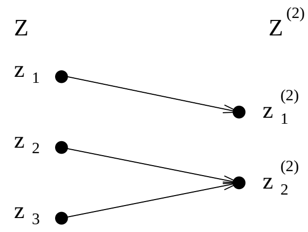
Figure 6.2: Representation of P_{XYZ}

Note first that not the entire information about P_{XYZ} is contained in this representation: only the distributions P_{XY} and P_{YZ}, but not P_{XYZ}, can be reconstructed from the quantities represented in the picture. We will see however that the fact whether or not X is simulatable by Z with respect to Y depends, not surprisingly, only on P_{XY} and P_{YZ}, as Theorem 6.7 shows. Second, not every representation corresponds to a distribution P_{XYZ}. This is only true if the total mass of each point set is 1, and if the marginal distribution P_Y is equal for both distributions P_{XY} and P_{YZ}. The last condition is equivalent to the fact that the sets of full and empty circles have the same center of gravity when interpreted as point masses.

Let now $Z^{(2)}$ with $\mathcal{Z}^{(2)} = \{z_1^{(2)}, z_2^{(2)}\}$ be generated by sending Z over the channel

$$\begin{aligned}
P_{Z^{(2)}|Z}(z_1^{(2)}, z_1) &= 1\ , \\
P_{Z^{(2)}|Z}(z_2^{(2)}, z_2) &= 1\ , \\
P_{Z^{(2)}|Z}(z_2^{(2)}, z_3) &= 1\ .
\end{aligned}$$

For the new distribution $P_{XYZ^{(2)}}$, the above representation is as shown in Figure 6.3: Two masses have been united in their center of gravity.



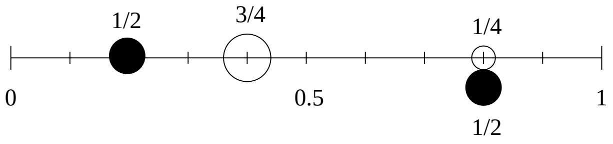


Figure 6.3: The Channel $P_{Z^{(2)}|Z}$, and $P_{XYZ^{(2)}}$

Let then $Z^{(2)}$ be sent over the additional channel $P_{Z^{(3)}|Z^{(2)}}$, where $\mathcal{Z}^{(3)} = \{z_1^{(3)}, z_2^{(3)}, z_3^{(3)}\}$, with

$$\begin{aligned} P_{Z^{(3)}|Z^{(2)}}(z_1^{(3)}, z_1^{(2)}) &= 1 \ , \\ P_{Z^{(3)}|Z^{(2)}}(z_2^{(3)}, z_2^{(2)}) &= 1/2 \ , \\ P_{Z^{(3)}|Z^{(2)}}(z_3^{(3)}, z_2^{(2)}) &= 1/2 \ . \end{aligned}$$

This corresponds to splitting one of the masses into two (equal) parts (see Figure 6.4).

Finally, let $P_{\overline{X}|Z^{(3)}}$, with $\overline{\mathcal{X}} = \{\overline{x}_1, \overline{x}_2\}$, be given by

$$\begin{aligned} P_{\overline{X}|Z^{(3)}}(\overline{x}_1, z_1^{(3)}) &= 1 \ , \\ P_{\overline{X}|Z^{(3)}}(\overline{x}_1, z_2^{(3)}) &= 1 \ , \\ P_{\overline{X}|Z^{(3)}}(\overline{x}_2, z_3^{(3)}) &= 1 \ . \end{aligned}$$

The use of this channel again corresponds to uniting two masses in their center of gravity. The constellation of the masses with respect to X and

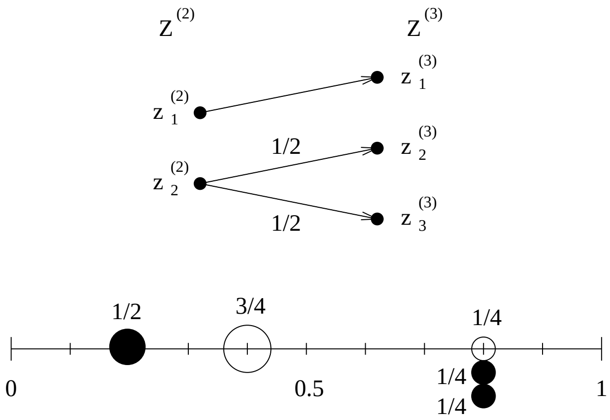


Figure 6.4: The Channel $P_{Z^{(3)}|Z^{(2)}}$, and $P_{XYZ^{(3)}}$

$\overline{X}$ are now equal (see Figure 6.5), which means that $P_{\overline{X}Y} = P_{XY}$ holds. Hence $\mathrm{sim}_Y(Z \to X)$ is true, and the corresponding channel $P_{\overline{X}|Z}$ is the cascade of the three channels above:

$$\begin{aligned} P_{\overline{X}|Z}(\overline{x}_1, z_1) &= 1 , \\ P_{\overline{X}|Z}(\overline{x}_1, z_2) &= P_{\overline{X}|Z}(\overline{x}_2, z_2) = 1/2 , \\ P_{\overline{X}|Z}(\overline{x}_1, z_3) &= P_{\overline{X}|Z}(\overline{x}_2, z_3) = 1/2 \end{aligned}$$

(see Figure 6.6). ◇

We can now make this representation in the mechanical model more precise and exploit the direct connection between distributions and channels on one side and mass constellations as well as -operations on the other, in order to give a simple characterization of (non-)simulatability. The purpose of this physical model is to give more intuitive deductions and formulations of results that could as well be stated and proved in terms of distributions and channels. Theorem 6.7 below makes a direct link between the two formalisms and justifies the point of view we take. In the following, particular emphasis lies on an intuitive presentation.

Definition 6.3. For an integer $N \geq 1$, an *N-dimensional (normed) mass constellation* $M := (m_i, a_i)_{i=1,\dots,\ell}$ is a family of pairs with $m_i \in (0,1]$ and $a_i \in [0,1]^N$ for all i such that $\sum_i m_i = 1$. We additionally

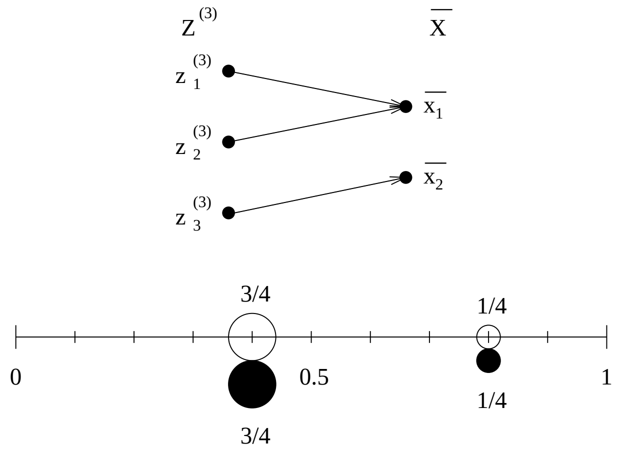


Figure 6.5: The Channel $P_{\overline{X}|Z^{(3)}}$, and $P_{XY\overline{X}}$

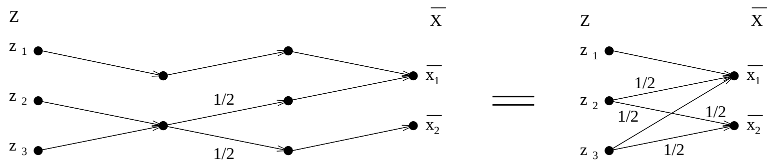


Figure 6.6: The Cascaded Channel $P_{\overline{X}|Z}$

assume that the pairs are ordered with respect to the lexicographic ordering of the vectors a_i. The *center of gravity* (*center* for short) $c(M)$ of such a constellation is given by

$$c(M) := \sum_{i=1}^{\ell} m_i a_i \ .$$

Two constellations are *equicentered* if they have the same center of gravity. A constellation $M' = (m_i', a_i')_{i=1,\ldots,\ell'}$ is *derived* from $M = (m_i, a_i)_{i=1,\ldots,\ell}$ by *mass splitting* if $\ell' = \ell+1$, and if there exist $0 < p < 1$,

$1 \leq i_0 \leq \ell$, such that

$$(m_i', a_i') = \begin{cases} (m_i, a_i) & 1 \leq i < i_0 \\ (pm_{i_0}, a_{i_0}) & i = i_0 \\ ((1-p)m_{i_0}, a_{i_0}) & i = i_0 + 1 \\ (m_{i-1}, a_{i-1}) & i_0 + 1 < i \leq \ell + 1 \ . \end{cases}$$

Furthermore, M' is *derived* from M by *mass union* if $\ell' = \ell - 1$, and if there exist $i_1 < i_2$, $i_1 \leq i_u \leq i_2$, such that

$$(m_i', a_i') = \begin{cases} (m_i, a_i) & 1 \leq i < i_1 \\ (m_{i+1}, a_{i+1}) & i_1 \leq i < i_u \\ \left(m_{i_1} + m_{i_2}, \frac{m_{i_1} a_{i_1} + m_{i_2} a_{i_2}}{m_{i_1} + m_{i_2}}\right) & i = i_u \\ (m_i, a_i) & i_u < i < i_2 \\ (m_{i+1}, a_{i+1}) & i_2 \leq i \leq \ell - 1 \ . \end{cases}$$

We call mass splitting and union *basic mass operations.* A constellation M is called *stronger* than M', an event denoted by $M \rightsquigarrow M'$, if there exists a finite sequence of basic operations that transforms M into M'.

Let P_{UV} be the joint distribution of two random variables U and V with ranges $\mathcal{U}$ and $\mathcal{V} = \{v_1, \ldots, v_{N+1}\}$. Then the N-dimensional constellation $M_{U \leftarrow V}$ is defined by

$$M_{U \leftarrow V} = (P_U(u), (P_{V|U=u}(v_1), \ldots, P_{V|U=u}(v_N)))_{u \in \mathcal{U}} \ .$$

○

Note that the definition of $M_{U \leftarrow V}$ leads to a one-to-one correspondence between distributions P_{UV}, where $|\mathcal{V}| = N + 1$, and N-dimensional normed mass constellations $(m_i, a_i)_{i=1,\ldots,\ell}$ contained in the simplex characterized by $(a)_j \geq 0$ and $\sum_{j=1}^{N} (a)_j \leq 1$.

It is clear that if $M \rightsquigarrow M'$, then the two constellations M and M' are equicentered. On the other hand, there exist equicentered constellations, none of which is stronger than the other (see Figure 6.7).

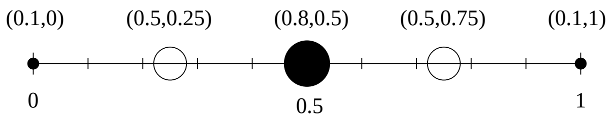


Figure 6.7: Incomparable Constellations: None is Stronger

Theorem 6.7 links simulatability and mass constellations. In this context, note first that for every distribution P_{XYZ}, $M_{X\leftarrow Y}$ and $M_{Z\leftarrow Y}$ are equicentered.

Theorem 6.7. *Let P_{XYZ} be the joint distribution of X, Y, and Z. Then X is simulatable by Z with respect to Y if and only if $M_{X\leftarrow Y}$ is stronger than $M_{Z\leftarrow Y}$:*

$$\mathrm{sim}_Y(Z \to X) \iff M_{Z\leftarrow Y} \rightsquigarrow M_{X\leftarrow Y} \ .$$

Proof. Let P_{U_1V} and P_{U_2V} be the joint distributions of random variables U_1 and V, and U_2 and V, respectively. Clearly, $M_{U_2\leftarrow V}$ can be obtained from $M_{U_1\leftarrow V}$ by a mass splitting or union operation if and only if there exists a "splitting channel" (as in Figure 6.3) or a "union channel" (see Figure 6.4) $P_{\overline{U}_2|U_1}$, respectively, such that

$$P_{\overline{U}_2V} = P_{U_1V} \cdot P_{\overline{U}_2|U_1} = P_{U_2V} \ .$$

The statement now follows from the facts that every discrete channel (with m output symbols) can be represented as a cascade of splitting and union channels, and that a cascade of channels is equivalent to the sequence of the corresponding mass operations. The first of these two facts can be shown as follows. First, all the letters of the input alphabet can be split, one after the other, to m symbols each (by $m-1$ splitting channels with certain probabilities for each symbol), and they can be united by union channels to the output symbols of the discrete channel. □

Remark. Note again that both conditions only depend on P_{XY} and P_{YZ}, but not otherwise on P_{XYZ}. Clearly, the condition given in Theorem 6.7 is a priori not more than a new formulation of simulatability, and is not obviously verifiable more efficiently. However, it leads to an efficiently-checkable criterion as Corollary 6.10 and Theorem 6.12 show.

As a preparation for the proof of Theorem 6.9, we describe a special mass operation, called *mass approach*, that can be composed by four basic operations (see Figure 6.8).

Lemma 6.8. *Let a constellation $M = (m_i, a_i)_{i=1,\ldots,\ell}$ be given, and let $i \neq i'$, $1 \leq i, i' \leq \ell$. We denote by*

$$c_{i,i'} := (m_i a_i + m_{i'} a_{i'})/(m_i + m_{i'})$$

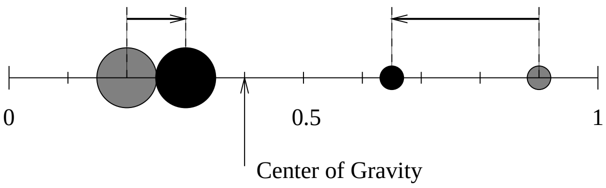


Figure 6.8: A Mass Approach

the center of the ith and i'th masses. Then for every $\lambda \in [0,1]$, there exist a sequence of four basic mass operations transforming M into the constellation that one obtains when the ith and i'th pairs are replaced by the pairs

$$(m_i, a_i + \lambda(c_{i,i'} - a_i)) \quad \textit{and} \quad (m_{i'}, a_{i'} + \lambda(c_{i,i'} - a_{i'}))$$

(which must be correctly put into the ordering).

Proof. The idea is that the masses m_i and $m_{i'}$ "exchange" a suitable mass $0 \leq m_e \leq \min\{m_i, m_{i'}\}$, i.e., that both masses are split into two parts, one of which is equal to m_e in both cases, and the union operation is applied twice to the remaining mass with the m_e-part of the other mass. Hence four basic operations are required. Because the choice $m_e = 0$ leaves m_i and $m_{i'}$ unchanged, whereas $m_e = \min\{m_i, m_{i'}\}$ corresponds to mass union, and since the result depends linearly on m_e, every position of m_i and $m_{i'}$ on their connecting line such that the masses are closer to each other, and such that the center of gravity remains unchanged, can be achieved this way. More explicitly, the mass m_e must be chosen as $\lambda \cdot \min\{m_i, m_{i'}\}$. □

We have now established the mechanical model and the necessary techniques for our characterizations of simulatability. In Corollary 6.10 we give a simple and efficiently verifiable, both necessary and sufficient condition for simulatability with respect to a *binary* random variable Y. Furthermore, the proof of Theorem 6.9 additionally shows that the corresponding channel $P_{\overline{X}|Z}$ can even be computed efficiently.

We first define what it means that a one-dimensional mass constellation is "more centered" than another. This relation leads to the characterization we are looking for. Note that this relation is not a linear ordering: When considering two random mass constellations, typically no one will be more centered than the other (see Figure 6.7).

Definition 6.4. For a one-dimensional mass constellation M and for $0 < t \leq 1$, we denote by $\ell_t(M)$ the leftmost masses of M of total amount d. (Typically, of one of the masses in M, only a part will be in $\ell_t(M)$.) A constellation M' is called *more centered* than M, denoted by

$$M' \prec M ,$$

if for all t,

$$c(\ell_t(M')) \geq c(\ell_t(M)) ,$$

where $c(S)$ stands for the center of gravity of a set S of masses. ○

Note first that this is a symmetric notion, i.e., that "left" and "$\geq$" could be replaced by "right" and "$\leq$" without changing the definition. Given two (finite) mass constellations, this quantity can be efficiently (i.e., in time linear in the total number of masses) checked. To see this, note that $M' = (m'_j, a'_j)_{j=1,\ldots,\ell'} \prec M$ is equivalent to the fact that for every $1 \leq k < \ell'$, the center of the set of masses $m'_1, \ldots, m'_k$ is not left of (i.e., smaller than) the center of $\ell_{m'_1+\cdots+m'_k}(M)$.

Theorem 6.9. *Let two equicentered one-dimensional mass constellations M and M' be given. Then M is stronger than M' if and only if M' is more centered than M:*

$$M \rightsquigarrow M' \iff M' \prec M .$$

Clearly, Corollary 6.10 follows immediately from Theorems 6.7 and 6.9.

Corollary 6.10. *Let P_{XYZ} be the joint distribution of random variables X, Y, and Z, where Y is binary. Then X is simulatable by Z with respect to Y if and only if $M_{X\leftarrow Y}$ is more centered than $M_{Z\leftarrow Y}$, i.e.,*

$$\mathrm{sim}_Y(Z \rightarrow X) \iff M_{X\leftarrow Y} \prec M_{Z\leftarrow Y} .$$

Proof of Theorem 6.9. We assume first that

$$M' = (m'_j, a'_j)_{j=1,\ldots,\ell'} \prec M = (m_i, a_i)_{i=1,\ldots,\ell}$$

holds. We show by induction that for every $0 \leq j_0 \leq \ell'$, there exists a sequence of basic mass operations that transforms M into a constellation $M_{j_0} = (\overline{m}_k, \overline{a}_k)_{k=1,\ldots,\overline{\ell}}$ such that for every $j \leq j_0$, there exists $k(j)$ (where $k(j) \neq k(j')$ if $j \neq j'$) with $\overline{m}_{k(j)} = m'_j$ and $\overline{a}_{k(j)} \leq a'_j$, and such that the center of the masses $\overline{m}_1, \ldots, \overline{m}_{j_0}$ of M_{j_0} is equal to $c(\ell_{\overline{m}_1+\cdots+\overline{m}_{j_0}}(M))$.

Clearly, this holds for $j_0 = 0$. We assume that the statement is true for $0 \leq j_0 < \ell'$ and show its validity also for $j_0 + 1$. Let $M_{j_0} = (\overline{m}_k, \overline{a}_k)_{k=1,\ldots,\overline{\ell}}$ be defined as above.

We transform M_{j_0} into M_{j_0+1} as follows. First, the leftmost among the masses $\overline{m}_{j_0+1}, \overline{m}_{j_0+2}, \ldots,$ of total amount m'_{j_0+1}, are united into their center of gravity. Let $(\overline{m}_{j_0+1}, \overline{a}_{j_0+1}) = (m'_{j_0+1}, \overline{a}_{j_0+1})$ be the resulting mass. Then, because of $M' \prec M$ and by the induction assumption, the center of the masses $(\overline{m}_1, \overline{a}_1), \ldots, (\overline{m}_{j_0+1}, \overline{a}_{j_0+1})$ is not on the right-hand side of the center of gravity of $\ell_{m'_1+\ldots+m'_{j_0+1}}(M')$. Hence there exists a sequence of mass approaches, applied only to masses among $\overline{m}_1, \ldots, \overline{m}_{j_0+1}$, such that each of the the resulting masses (still of the same sizes) is on the left-hand side of (or at the same position as) the corresponding mass of M' (see Figure 6.9). Hence this new constellation satisfies the induction assumption for $j_0 + 1$, and this concludes the induction argument.

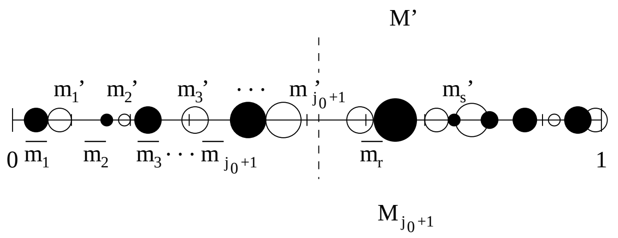


Figure 6.9: M' and M_{j_0+1}

Therefore M is stronger than some $\overline{M}$ satisfying the above property, with respect to M', for $j_0 = \ell'$. However, because $\overline{M}$ and M' are both equicentered to M, and because all masses of $\overline{M}$ lie, roughly speaking, on the left of (or at the same place as) the corresponding masses of M', we must have that $\overline{M} = M'$, hence $M \rightsquigarrow M'$.

We show the necessity of the condition. Assume for M and M' and for some t that

$$c(\ell_t(M')) < c(\ell_t(M)).$$

Then $M \not\rightsquigarrow M'$ holds because the basic mass operations, i.e., mass union (mass splitting leaves all the centers unchanged), can only move the center of the set $\ell_t(M)$ to the right (union of two masses, one in the set

$\ell_t(M)$, and one in the complement) or leave it at the same place (union within $\ell_t(M)$ or within the complement). □

Note that the criterion for simulatability of Corollary 6.10 is not only very simple and efficiently verifiable, but that the proof of Theorem 6.9 also shows how a channel $P_{\overline{X}|Z}$ for simulating X with respect to Y can be constructed efficiently.

Let us now, after the complete analysis of the case of a binary random variable Y, consider the general case again. In Definition 6.5, we give a straight-forward, and also efficiently checkable, generalization of the notion that a constellation is more centered than another. This leads to a *necessary* criterion for simulatability (Theorem 6.12). However, although it appears to be sufficient as well in many cases, we give an example for which non-simulatability is not detected by the criterion.

Definition 6.5. Let M and M' be two N-dimensional mass constellations. Let furthermore a line L, passing through the origin, be given. We now consider the orthogonal projections of all the masses in the N-dimensional space onto L. This yields two one-dimensional equicentered mass constellations M_L and M'_L. We say that M' is *more centered* than M, $M' \prec M$, if $M'_L \prec M_L$ for every line L. ○

It is not difficult to see that also in N dimensions $M \rightsquigarrow M'$ can only hold if $M' \prec M$ holds. The reason is that $M_L \rightsquigarrow M'_L$ follows from $M \rightsquigarrow M'$: Projections of mass operations are mass operations again.

Theorem 6.11. *Let M and M' be N-dimensional equicentered mass constellations. If M is stronger than M', then M' must be more centered than M:*

$$M \rightsquigarrow M' \implies M' \prec M \ .$$

Corollary 6.12. *Let P_{XYZ} be the joint distribution of X, Y, and Z. If $M_{X \leftarrow Y}$ is* not *more centered than $M_{Z \leftarrow Y}$, then X is* not *simulatable by Z with respect to Y, i.e.,*

$$M_{X \leftarrow Y} \not\prec M_{Z \leftarrow Y} \implies (\,\mathrm{sim}_Y(Z \to X) \ \ \textit{does}\ \text{not}\ \textit{hold}\,) \ .$$

Note that this condition is, despite the fact that the number of lines through the origin is infinite, efficiently verifiable since the number of points is finite. First, not every direction, i.e., every line, has to be checked separately. There are only at most $\binom{\ell+\ell'}{2}$ directions for which the mass constellations are different (with respect to the order of the masses), where ℓ and ℓ' are the numbers of masses in M and M', respectively. Equal orders means that, in the N-dimensional space, the same masses are "leftmost." Hence, all these directions can be treated simultaneously by looking at extremal directions. Furthermore, only the values t corresponding to a subset of the masses in M' have to be considered (as in the one-dimensional case).

Unfortunately, the given condition is not sufficient for simulatability (i.e., for a mass constellation being stronger than another) in the $N(\geq 2)$-dimensional case (although it appears to be a "good" condition failing to detect non-simulatability only in a small fraction of all cases). The following is a counterexample.

Example 6.2. Consider the following two-dimensional mass constellations M and M'.

$$\begin{aligned} M &= (0.2,(0,0)),(0.2,(0,0.5)),(0.2,(0.5,0)), \\ &\quad (0.2,(0.5,0.5)),(0.2,(0.25,0.25)), \\ \\ M' &= (0.2,(0.1,0)),(0.2,(0.1,0.5)),(0.2,(0.4,0)), \\ &\quad (0.2,(0.4,0.5)),(0.1,(0.15,0.25)),(0.1,(0.35,0.25)) \end{aligned}$$

(see Figure 6.10).

It is not difficult to see that $M' \prec M$ holds. First, it clearly holds for the horizontal line and, because the (horizontal) distances between neighboring masses change in the same ratios, for all lines except the vertical line, for which the projected constellations are identical however.

On the other hand, M cannot be transformed into M' by basic operations. This is true because when considering the projection to the vertical line, it is clear that no union operation can be made except between masses with the same y-coordinate. However, with such operations only, M can clearly not be transformed to M' because of the masses with y-coordinate $1/2$. Hence $M \not\rightarrow M'$ holds. ◊

6.3 Privacy Amplification

In this section we study privacy amplification secure against active adversaries, i.e., the special case of secret-key agreement in which the parties already have a mutual (but generally partially-insecure) key at the

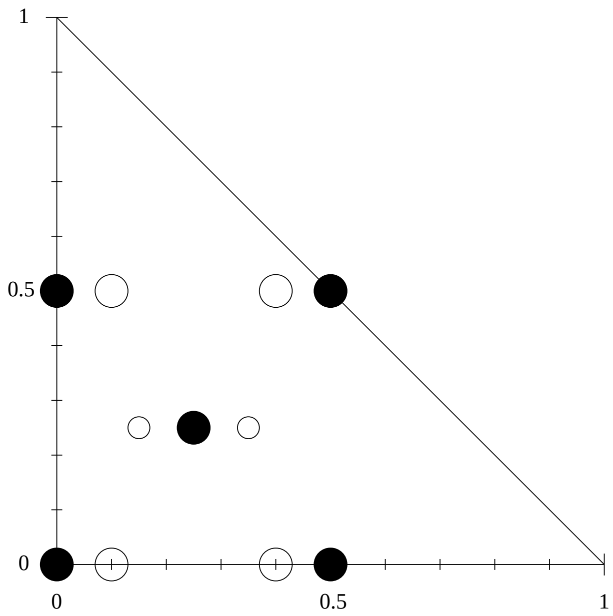


Figure 6.10: A 2-Dimensional Counterexample

beginning, i.e., where $X = Y =: S$ holds with probability 1. The protocol definition (which is a strengthened version of the general definition of Section 6.1) is given in Section 6.3.1, and some techniques are described (in Sections 6.3.2, 6.3.3, 6.3.4, and 6.3.6) as a preparation for the analysis of the Protocols UH and EX for privacy amplification in this model (presented in Sections 6.3.5 and 6.3.7).

6.3.1 Protocol Definition

The protocol definition for privacy amplification secure against active adversaries can be strengthened in two respects as compared to the definition in the general case. First, it is required that Alice and Bob both accept and end up with the same string with probability 1 if Eve is passive. Moreover, the protocols work for an entire class of distributions P_{XYZ} instead of only one distribution. (The latter would make no sense in the context of privacy amplification.) More precisely, Eve's knowledge about the mutual n-bit string S is bounded by assuming that $P_{S|Z=z}$

is, for all $z \in \mathcal{Z}$, contained in some subset $\mathcal{D}$ of all possible distributions over the set $\{0,1\}^n$. Typically, $\mathcal{D}$ consists of the distributions satisfying a certain condition in terms of Rényi entropy or min-entropy. We denote for every t and $\alpha \in \{2,\infty\}$ by $\mathcal{D}_{n,\alpha,t}$ the subset $\{P_X \mid H_\alpha(X) \geq t\}$ of distributions over n-bit strings.

Definition 6.6. Assume that Alice and Bob both know a mutual n-bit random variable S, and that the random variable Z summarizes Eve's entire knowledge about S. Let $\mathcal{D}$ be a subset of all probability distributions on the set of n-bit strings, let r be an integer, and let $\varepsilon, \delta > 0$. A $(n, \mathcal{D}, r, \varepsilon, \delta)$*-protocol for privacy amplification by communication over an insecure and non-authentic channel* (a *robust* $(n, \mathcal{D}, r, \varepsilon, \delta)$*-PA-protocol* for short) is a protocol according to Definition 6.1, but with the following modified correctness and privacy property. Assume that $P_{S|Z=z} \in \mathcal{D}$ for all $z \in \mathcal{Z}$.

1'. Correctness and privacy. If Eve is a passive wire-tapper receiving the particular value $Z = z$, then both Alice and Bob must accept at the end of the protocol, and there must exist an r-bit string S' such that $S' = S'_A = S'_B$ and $H(S'|C, Z = z) \geq r - \varepsilon$. ○

Clearly, the impossibility results of Section 6.2 immediately carry over to privacy amplification secure against active opponents (where the non-simulatability is obviously fulfilled in all non-trivial cases). There exists neither a protocol with perfect synchronization of the accepting states (i.e., equality of the final states in every case), nor a one-way-transmission protocol satisfying the required properties.

Theorem 6.13. *Let* $\alpha \in \{2,\infty\}$*. Assume that a*

$$\textit{robust } (n, \mathcal{D}_{n,\alpha,t}, r, \varepsilon, \delta)\textit{-PA-protocol}$$

either with perfect synchronization or using only one-way transmission exists. Then $\varepsilon \geq r - 1$*,*

$$\varepsilon > n - t + \log(1 - 2^{\varepsilon - r + 1}) ,$$

or $\delta = 1$ *holds.*

Proof. Assume $\varepsilon < r - 1$ and $\varepsilon \leq n - t + \log(1 - 2^{\varepsilon - r + 1})$. We show that there exists, for every fixed function $f : \{0,1\}^n \to \{0,1\}^r$ (on which

Alice and Bob could agree without any communication) a distribution $P_{S|Z=z} \in \mathcal{D}$ such that

$$H(f(S)|Z=z) < r - \varepsilon \ .$$

Let $\mathcal{A} \subseteq \{0,1\}^r$ be the particular set of size

$$2^r - \lfloor 2^{r-\varepsilon} \rfloor + 1$$

which minimizes the cardinality of the set

$$f^{-1}(\mathcal{A}) := \{s \in \{0,1\}^n \ : \ f(s) \in \mathcal{A}\}$$

Then

$$\begin{aligned} |f^{-1}(\mathcal{A})| &\leq 2^{n-r}(2^r(1-2^{-\varepsilon})+2) \\ &= 2^n(1-2^{-\varepsilon}+2^{-r+1}) \ . \end{aligned}$$

For $\mathcal{B} := \{0,1\}^n \setminus f^{-1}(\mathcal{A})$ we have hence

$$|\mathcal{B}| \geq 2^{n-\varepsilon}(1-2^{\varepsilon-r+1}) \ .$$

Let $P_{S|Z=z}$ be the uniform distribution on $\mathcal{B} \subseteq \{0,1\}^n$. Then we have

$$H_\alpha(S|Z=z) = \log|\mathcal{B}| \geq n - \varepsilon + \log(1-2^{\varepsilon-r+1}) \geq t$$

and

$$H(f(S)|Z=z) \leq \log(2^r - |\mathcal{A}|) < r - \varepsilon$$

by construction and by the assumption. This contradicts the protocol definition, hence at least one message must be sent in the protocol. The rest of the argument is as in the proofs of Theorems 6.1 and 6.2. □

6.3.2 The Effect of Side Information and Knowledge About Partial Strings

In this section we provide some facts necessary for the analysis of Protocols UH and EX for privacy amplification described below. We derive bounds on the amount of knowledge (e.g., of an adversary) in terms of Rényi entropy and min-entropy about a partial string, depending on the amount of knowledge about the entire string. This is done both for the cases where the adversary does (Corollary 6.16) or does not (Lemma 6.14) obtain information about the remaining part of the string. In both cases, the result is roughly the intuitive fact that [with high probability] one cannot know [substantially] more about a part than about the whole (see Figure 6.11). In the case where the adversary

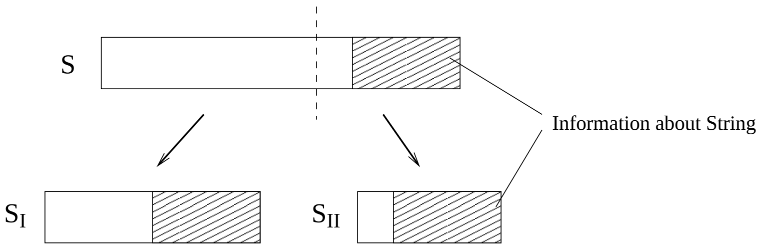


Figure 6.11: Information About Partial Strings

obtains information about the remaining part of the string, the result follows from a general upper bound on the reduction of the Rényi entropy and min-entropy of a random variable when side information is given (Lemma 6.15).

Lemma 6.14. *Let $S = (S_1, S_2, \ldots, S_n)$ be a random variable consisting of n binary random variables. For any k-tuple $\underline{i} = (i_1, i_2 \ldots, i_k)$, where $1 \leq i_1 < i_2 < \cdots < i_k \leq n$, let $S_{\underline{i}}$ be the string $(S_{i_1}, S_{i_2}, \ldots, S_{i_k})$. Then*

$$H_\alpha(S_{\underline{i}}) \geq H_\alpha(S) - (n-k)$$

holds for $\alpha \in \{2, \infty\}$.

Proof. Let first $\alpha = 2$. Consider a fixed string $(s_{i_1}, \ldots, s_{i_k})$. This particular value of the random variable $S_{\underline{i}}$ corresponds to exactly 2^{n-k} values $(s_1, \ldots, s_n)$ of the random variable S. Let $p_1, \ldots, p_{2^{n-k}}$ be the probabilities of these strings, and let $p_0 := \sum_{i=1}^{2^{n-k}} p_i$. Now, we have

$$\begin{aligned} \log\left(\sum_{i=1}^{2^{n-k}} \left(\frac{p_i}{p_0}\right)^2\right) &= \log\left(\mathrm{E}\left[\frac{p_i}{p_0}\right]\right) \\ &\geq \mathrm{E}\left[\log\left(\frac{p_i}{p_0}\right)\right] \\ &= -\mathrm{E}\left[\log\left(\frac{p_0}{p_i}\right)\right] \end{aligned}$$

$$\begin{aligned}
&\geq -\log\left(\mathrm{E}\left[\frac{p_0}{p_i}\right]\right)\\
&= -\log\left(\sum_{i=1}^{2^{n-k}} \frac{p_i}{p_0}\cdot\frac{p_0}{p_i}\right)\\
&= -\log(2^{n-k}) \;=\; k-n\;.
\end{aligned}$$

Here, the expectation is with respect to the probability distribution p_i/p_0 over the 2^{n-k} strings. We have twice made use of the fact that the logarithm is a concave function, and of Jensen's inequality. We conclude

$$\sum_{i=1}^{2^{n-k}} p_i^2 \geq \frac{p_0^2}{2^{n-k}}\;. \tag{6.4}$$

Because inequality (6.4) holds for every particular string $(s_{i_1},\ldots,s_{i_k})$, we have for the collision probabilities of the random variables S and $S_{\underline{i}}$.

$$\begin{aligned}
P_C(S_{\underline{i}}) &= \sum_{\{0,1\}^k} P_{S_{\underline{i}}}((s_{i_1},\ldots,s_{i_k}))^2\\
&\leq 2^{n-k}\cdot\sum_{\{0,1\}^n} P_S((s_1,\ldots,s_n))^2\\
&= 2^{n-k}\cdot P_C(S)\;.
\end{aligned}$$

Hence

$$H_2(S_{\underline{i}}) \geq H_2(S)-(n-k)\;.$$

For the case $\alpha=\infty$, the inequality follows directly from the fact that the maximal probability of a k-bit string is at most 2^{n-k} times the maximal probability of a string in S. □

Remark. Note that when the string S is split into two parts S_l and S_r, then the bounds of Lemma 6.14 applied to S_l and S_r are tight *simultaneously.* For example let $\alpha=\infty$ and $s=(s_l,s_r)$ be a particular n-bit string, and let s_l and s_r be the first and second half of s. Define (for some $v\leq n/2-1$) $P_S((s_l,\overline{s}))=P_S((\overline{s},s_r)):=2^{v-n}$ for all $n/2$-bit strings $\overline{s}$ (and a uniform distribution for the remaining n-bit strings), i.e., $H_\infty(S)=n-v$. Then

$$H_\infty(S_l)=H_\infty(S_r)=n/2-v=H_\infty(S)-n/2\;.$$

Intuitively speaking but counter to intuition, Eve's information about S in terms of min-entropy appears entirely in both substrings S_l and S_r.

Lemma 6.15, which is a modified version of Lemma 4.3, gives an upper bound on the reduction of the Rényi entropy and min-entropy $H_2(P)$ and $H_\infty(P)$ of a random variable P when side information $[Q, R]$ (consisting of a pair of random variables) is given, where $I(P;R) = 0$. It states that this reduction exceeds $\log|\mathcal{Q}|$ substantially only with small probability in both cases. (Note that it is not trivial that no additional reduction is induced by R if $I(P;R) = 0$. For instance, $I(P;Q) = 0$ and $I(P;R) = 0$ together do *not* imply that $H_2(P|Q = q, R = r) = H_2(P)$, as the example $P = Q \oplus R$ shows.)

Lemma 6.15. *Let P, Q, and R be random variables with $I(P;R) = 0$. Then*

$$\mathrm{P}_{QR}\left[H_2(P \mid Q = q, R = r) \geq H_2(P) - \log|\mathcal{Q}| - s\right] \geq 1 - 2^{-(s/2-1)}$$

for all $s > 2$, and

$$\mathrm{P}_{QR}\left[H_\infty(P \mid Q = q, R = r) \geq H_\infty(P) - \log|\mathcal{Q}| - s\right] \geq 1 - 2^{-s}$$

for $s > 0$.

Proof. We prove the statement concerning Rényi entropy first. The argument is a generalization of the proof of a related theorem in [19]. Let $r \in \mathcal{R}$ be fixed. It is straight-forward that

$$2^{-H_2(PQ|R=r)} = \mathrm{E}_Q\left[2^{\log P_{Q|R=r} - H_2(P \mid Q=q, R=r)}\right] .$$

Hence the probability that $\log P_{Q|R=r} - H_2(P|Q = q, R = r)$ exceeds $-H_2(PQ|R = r)$ by more than $s/2$ is at most $2^{-s/2}$, i.e.,

$$\mathrm{P}_Q\left[H_2(P \mid Q = q, R = r) \leq H_2(PQ \mid R = r) + \log P_{Q|R=r} - s/2\right] \leq 2^{-s/2} . \tag{6.5}$$

Furthermore,

$$\mathrm{P}_Q\left[\log P_{Q|R=r} \leq -\log|\mathcal{Q}| - s/2\right] \leq 2^{-s/2} . \tag{6.6}$$

The inequalities (6.5) and (6.6) together imply

$$\mathrm{P}_Q\left[H_2(P \mid Q = q, R = r) \leq H_2(PQ \mid R = r) - \log|\mathcal{Q}| - s\right] \leq 2^{-(s/2-1)} .$$

Finally, $H_2(PQ|R=r) \geq H_2(P|R=r) = H_2(P)$ holds because

$$\begin{aligned}
\sum_{p\in\mathcal{P},\, q\in\mathcal{Q}} P_{PQ|R=r}(p,q)^2 &= \sum_{p\in\mathcal{P}} \left(P_{P|R=r}(p)^2 \cdot \sum_{q\in\mathcal{Q}} P_{Q|P,R=r}(q,p)^2 \right) \\
&\leq \sum_{p\in\mathcal{P}} P_{P|R=r}(p)^2 \\
&= \sum_{p\in\mathcal{P}} P_P(p)^2 \ .
\end{aligned}$$

We have used for the last equality that P and R are statistically independent, i.e., that $P_{P|R=r} = P_P$. We conclude that

$$\mathrm{P}_Q[H_2(P\,|\,Q=q, R=r) \leq H_2(P) - \log|\mathcal{Q}| - s] \leq 2^{-(s/2-1)}$$

holds for all $r \in \mathcal{R}$, and the first statement of the lemma follows.

We give a proof of the second statement. Let $p_0 := 2^{-s}/|\mathcal{Q}|$. Then we have for all $r \in \mathcal{R}$

$$P_{Q|R=r}[\{q \,:\, P_{Q|R=r} < p_0\}] < 2^{-s} \ ,$$

and hence

$$P_{QR}\left[\left\{(q,r) \in \mathcal{Q}\times\mathcal{R} \ :\ P_{Q|R}(q,r) < p_0\right\}\right] < 2^{-s} \ .$$

This inequality implies that

$$\begin{aligned}
P_{P|QR}(p,q,r) &= \frac{P_{PQR}(p,q,r)}{P_{QR}(q,r)} = \frac{P_P(p)\cdot P_R(r)\cdot P_{Q|PR}(q,p,r)}{P_R(r)\cdot P_{Q|R}(q,r)} \\
&\leq \frac{P_P(p)}{P_{Q|R}(q,r)} \leq \frac{P_P(p)}{p_0} = P_P(p)\cdot|\mathcal{Q}|\cdot 2^s \ ,
\end{aligned}$$

holds with probability greater than $1-2^{-s}$ (taken over Q and R). The statement of the lemma follows by maximizing over all $p \in \mathcal{P}$, and by taking negative logarithms. □

The following corollary is a direct consequence of Lemma 6.15. It states that a formally slightly weaker result than that of Lemma 6.14, concerning the knowledge (in terms of H_2 and of H_∞) of a partial string, even holds when the rest of the string is made public.

Corollary 6.16. *Let S be an n-bit string, and let a partition of S be given into two strings S' and S'' of lengths l and $n-l$, respectively. Let $s > 2$ be a security parameter. Then the probability, taken over s'', that*

$$H_2(S' \,|\, S'' = s'') \geq H_2(S) - (n - l) - s$$

holds is at least $1 - 2^{-(s/2-1)}$. Furthermore, for $s > 0$, the probability, taken over s'', that

$$H_\infty(S' \,|\, S'' = s'') \geq H_\infty(S) - (n - l) - s$$

holds is at least $1 - 2^{-s}$.

6.3.3 Message Authentication with a Partially-Secret Key I: Strongly-Universal Hashing

All previous results on unconditionally-secure authentication require a key that is completely secret, i.e., the probability distribution of the key is uniform from the opponent's viewpoint. In this section we prove a result on authentication where the opponent is allowed to have some partial information about the key. These techniques are used in the protocols described in the following sections.

There exists a variety of possibility and impossibility results on information-theoretically-secure authentication (see for example [111], [60], or [113]). The following two types of attacks are possible. In an *impersonation attack*, the opponent tries to generate a (correctly authenticated) message, and in a *substitution attack*, the adversary observes a correctly authenticated message and tries to replace it by a different correctly authenticated message. The success probabilities are denoted by p_{imp} and p_{sub}, respectively. (General lower bounds on these probabilities are given in [60].)

One possibility for realizing information-theoretically-secure authentication is by using strongly-universal classes of hash functions (see for example [113]).

Definition 6.7. A class $\mathcal{H}$ of (hash) functions $\mathcal{A} \to \mathcal{B}$ is called *strongly-universal* (or SU_2 for short) if for all distinct $a_1, a_2 \in \mathcal{A}$ and for all $b_1, b_2 \in \mathcal{B}$, the number of functions $h \in \mathcal{H}$ for which both $h(a_1) = b_1$ and $h(a_2) = b_2$ hold is $|\mathcal{H}|/|\mathcal{B}|^2$. ○

Remark. Note that a strongly-universal class has in particular the following property. For every $a \in \mathcal{A}$ and $b \in \mathcal{B}$, the number of functions

$h \in \mathcal{H}$ such that $h(a) = b$ holds is $|\mathcal{H}|/|\mathcal{B}|$. This is true because for all $a, a' \in \mathcal{A}$, $a' \neq a$, and $b \in \mathcal{B}$, we have

$$\begin{aligned} |\{h \in \mathcal{H} : h(a) = b\}| &= \left| \bigcup_{b' \in \mathcal{B}} \{h \in \mathcal{H} : h(a) = b\,,\; h(a') = b'\} \right| \\ &= \sum_{b' \in \mathcal{B}} |\{h \in \mathcal{H} : h(a) = b\,,\; h(a') = b'\}| \\ &= |\mathcal{B}| \cdot \frac{|\mathcal{H}|}{|\mathcal{B}|^2} \;=\; \frac{|\mathcal{H}|}{|\mathcal{B}|}\;. \end{aligned}$$

By roughly the same argument one can also show that a strongly-universal class is in particular universal (a fact that is already suggested by the names of the properties).

A strongly-universal class of hash functions can immediately be used for authentication: the secret key determines a hash function of the class, and the message is authenticated by its hash value. The authentication code corresponding to an SU_2 class of hash function satisfies

$$p_{imp} = 1/|\mathcal{B}|$$

and

$$p_{sub} = 1/|\mathcal{B}|\;.$$

An SU_2 class of functions mapping N-bit strings to N-bit strings can be constructed similarly to the universal class described in Section 4.2.1. Namely, the class

$$\mathcal{H} = \{h_{ab} : (a, b) \in (GF(2^N))^2\}\;, \tag{6.7}$$

where

$$h_{ab}(x) := a \cdot x + b\;,$$

is an SU_2 class of hash functions $\{0,1\}^N \to \{0,1\}^N$ with 2^{2N} elements, i.e., with a key $S = a||b$ of length $2N$.

Let us now investigate the scenario in which the key is not entirely secret, i.e., where the opponent Eve has a certain amount of information about this key. The following result states that information-theoretically-secure authentication is possible under the condition that the Rényi entropy of the key from the adversary's viewpoint is greater than half the length of the key.

Theorem 6.17. *Let S be a binary string of (even) length n. Assume that S is used by two parties as the key in the authentication scheme based on strongly-universal hashing with respect to the class (6.7), that an adversary knows a random variable Z, jointly distributed with S according to some probability distribution, and that the opponent has no further information about S. Let*

$$H_2(S|Z=z) \geq (1/2+R) \cdot N$$

for a particular $z \in \mathcal{Z}$. Then the probabilities of successful impersonation and substitution attacks, given $Z = z$, are upper bounded by

$$p_{imp} \leq 2^{-RN/2}$$

and

$$p_{sub} \leq 3 \cdot 2^{-RN/4} \ , \tag{6.8}$$

respectively.

Remark. Note that in Theorem 6.17 it need not be assumed that the message observed by Eve be independent of S (but independent of S *given* $Z = z$). For example (6.8) holds also when the message is selected by Eve herself.

Proof. First we prove the upper bound on the success probability p_{imp} of the impersonation attack. For every possible message $m \in GF(2^{N/2})$ and for every authenticator $y \in GF(2^{N/2})$ there exist exactly $2^{N/2}$ possible keys such that y is the correct authenticator for m. The probability of such a set of keys, given that $Z = z$, can be upper bounded as follows. In the worst case (i.e., the best case for the impersonating attacker) the $2^{N/2}$ keys have all the same probability, say p. Then p must satisfy

$$2^{N/2} \cdot p^2 \leq P_C(S) \leq 2^{-(1/2+R)N} \ ,$$

i.e.,

$$p \leq 2^{-(1/2+R/2)N} \ .$$

Hence

$$p_{imp} \leq 2^{N/2} \cdot 2^{-(1/2+R/2)N} = 2^{-RN/2} \ .$$

Let us now consider the substitution attack. The crucial argument is that the key s is uniquely determined by $(m, h_s(m))$ and $(m', h_s(m'))$ if $m \neq m'$. Hence the probability of a successful substitution attack is not greater than the probability of guessing S correctly when given $(M, h_S(M))$. From Lemma 6.15, and because $I(S; M|Z = z) = 0$, we can conclude that

$$H_2(S \,|\, M = m, h_S(M) = h_s(m), Z = z) \geq RN/2 \tag{6.9}$$

holds with probability at least $1 - 2^{-(RN/4-1)}$. On the other hand, if inequality (6.9) holds, then the maximal probability of a particular key s is at most

$$\sqrt{2^{-H_2(S \,|\, M=m, h_S(M)=h_s(m), Z=z)}} \leq 2^{-RN/4} \ .$$

Thus we have

$$p_{sub} \leq 2^{-(RN/4-1)} + 2^{-RN/4} = 3 \cdot 2^{-RN/4} \ .$$

□

Remark. It has been proposed to use smaller but "weaker" classes of functions, so-called ε-almost strongly-universal (ε-ASU) hash functions, instead of strongly-universal hashing for authentication [113]. Such classes allow for authentication with a substantially smaller secret key at the price of a somewhat greater success probability of a substitution attack. However, for the purpose of authentication with a *partially*-secret key, these classes of functions do not lead to better results. Whenever the Rényi entropy of the partially-secret key is smaller than half the length of the key, then no unconditionally-secure authentication is possible with this key by using (ε-A)SU hashing because one correct message-authenticator pair can reveal the remaining information necessary to uniquely determine the key.

6.3.4 Challenge-Response Identification with a Highly-Insecure Key

In the previous sections we have shown that message authentication is possible with a partially-secret key, or more precisely, with a key the Rényi entropy of which from the adversary's viewpoint is more than half its length. In this section on the other hand we prove that a certain type of security against active attacks can even be achieved when the key shared by the legitimate partners is highly insecure (e.g., in terms of Rényi entropy). A challenge-response scheme is described that can successfully be attacked only by an adversary having almost complete knowledge about the secret key. This method is used as the final step in both the Protocols UH (Section 6.3.5) and EX (Section 6.3.7). The purpose of this step is to prevent the party sending the final message that is needed for successful secret-key agreement from accepting although key agreement has failed. In Section 6.3.6, a related result is proved that shows how the same scheme can be used for authenticating short messages.

Lemma 6.18. *Let N and ℓ be integers such that ℓ divides N and $2^\ell \geq N/\ell$ holds, and let K be a random variable with range $\mathcal{K} \subseteq GF(2^N)$. Let further for any $d \in GF(2^\ell)$ the function $f_d : \{0,1\}^N \rightarrow \{0,1\}^\ell$ be defined as*

$$f_d(x) := \sum_{i=0}^{N/\ell-1} d^i x_i \ ,$$

where $(x_0, \ldots, x_{N/\ell-1}) \in (GF(2^\ell))^{N/\ell}$ is a representation of $x \in GF(2^N)$ with respect to a fixed basis of $GF(2^N)$ over $GF(2^\ell)$, where the computations are carried out in the field $GF(2^\ell)$, and where the elements of $GF(2^\ell)$ are represented as ℓ-bit strings with respect to a fixed basis of $GF(2^\ell)$ over $GF(2)$. Assume that for $d \in_R GF(2^\ell)$, the value $f_d(K)$ can be guessed correctly (with some strategy) with probability $\alpha \geq (N/\ell)/2^\ell$, taken over the distribution of K, the choice of d, and the coin tosses of the guessing strategy. Then

$$H_2(K) \leq -\frac{2N}{\ell} \cdot \log\left(\alpha - \frac{N/\ell}{2^\ell}\right) \tag{6.10}$$

or, equivalently,

$$\alpha \leq 2^{-(\ell/2N)\cdot H_2(K)} + \frac{N/\ell}{2^\ell} \ .$$

Proof. First, we can assume without loss of generality that the strategy of guessing $f_d(k)$ is deterministic, since for every possible strategy there exists a deterministic strategy that is at least as good (since every randomized strategy can be seen as a combination of deterministic strategies, of which the optimal one can be chosen).

We give a lower bound on the probability β that for randomly- and independently-chosen distinct elements $d_1, \ldots, d_{N/\ell}$ of $GF(2^\ell)$, all the values $f_{d_i}(k)$ are guessed correctly. Let $g(x)$ be the function

$$g(x) := \begin{cases} x \cdot \left(x - \frac{1}{2^\ell}\right) \cdot \left(x - \frac{2}{2^\ell}\right) \cdots \left(x - \frac{N/\ell - 1}{2^\ell}\right) & \text{if } x \geq \frac{N/\ell - 1}{2^\ell} \\ 0 & \text{otherwise .} \end{cases}$$

Let, for every $k \in \mathcal{K}$, n_k denote the number of distinct $d \in GF(2^\ell)$ for which $f_d(k)$ is guessed correctly by the (deterministic) guessing strategy. Then we have

$$\beta = \mathrm{E}_K \left[g\left(\frac{n_k}{2^\ell}\right)\right] \geq g\left(\mathrm{E}_K\left[\frac{n_k}{2^\ell}\right]\right) \geq \left(\alpha - \frac{N/\ell}{2^\ell}\right)^{N/\ell} .$$

(Here, we have made use of Jensen's inequality. Note that g is a convex function.) Hence there exist N/ℓ distinct $d_1, \ldots, d_{N/\ell}$ such that the values $f_{d_i}(k)$ are simultaneously guessed correctly with probability at least $(\alpha - (N/\ell)/2^\ell)^{N/\ell}$, taken over k.

On the other hand, k is uniquely determined by the correct values $f_{d_i}(k)$, $k = 1, \ldots, N/\ell$. In order to see this, note first that

$$\begin{pmatrix} f_{d_1}(k) \\ f_{d_2}(k) \\ \vdots \\ f_{d_{N/\ell}}(k) \end{pmatrix} = \begin{pmatrix} d_1^0 & d_1^1 & \cdots & d_1^{N/\ell-1} \\ d_2^0 & d_2^1 & \cdots & d_2^{N/\ell-1} \\ \vdots & \vdots & & \vdots \\ d_{N/\ell}^0 & d_{N/\ell}^1 & \cdots & d_{N/\ell}^{N/\ell-1} \end{pmatrix} \cdot \begin{pmatrix} k_0 \\ k_1 \\ \vdots \\ k_{N/\ell-1} \end{pmatrix} ,$$

and second, that the determinant of the matrix, called *Vandermonde determinant*, is equal to

$$\prod_{1 \leq i < j \leq N/\ell} (d_j - d_i) \neq 0 ,$$

hence the matrix is invertible, and $k = (k_0, k_1, \ldots, k_{N/\ell-1})$ is uniquely determined by the $f_{d_i}(k)$'s. An alternative way to see this fact is by interpreting $f_d(x)$ as a polynomial $P_x(d)$ of degree $N/\ell - 1$ over $GF(2^\ell)$, which is uniquely determined, thus also x is, by its evaluation at N/ℓ distinct points $d_1, \ldots, d_{N/\ell}$. Hence there must be an element $k_0 \in \mathcal{K}$ with

$$P_K(k_0) \geq \left(\alpha - \frac{N/\ell}{2^\ell}\right)^{N/\ell} .$$

Because of $P_C(K) \geq P_K(k_0)^2$, we can conclude that (6.10) holds. □

6.3.5 A Protocol Using Universal Hashing

We are now ready to give a first protocol for privacy amplification secure against active adversaries. The ingredients of this protocol are universal hashing (for privacy amplification), strongly-universal hashing (for the authentication of the message, i.e., the random bits determining the hash function), and the challenge-response scheme of Section 6.3.4.

For parameters n and ℓ, where 3 divides n and ℓ divides $2n/3$, Protocol UH is defined as follows. (Here, as well as in Protocol EX, the `reject` states are the default states, and are valid initially and until "`accept`" appears in the protocol specification.)

Protocol UH (Universal Hashing)

Alice		**Bob**
$S = S_I \vert\vert S_{II} \vert\vert S_{III}$		$S = S_I \vert\vert S_{II} \vert\vert S_{III}$
$S_I \vert\vert S_{II} =$		$S_I \vert\vert S_{II} =$
$S_0 \vert\vert S_1 \vert\vert \cdots \vert\vert S_{2n/3\ell-1}$		$S_0 \vert\vert S_1 \vert\vert \cdots \vert\vert S_{2n/3\ell-1}$
$h \in_R GF(2^{n/3})$		
$S'_A := \mathrm{LSB}_r(h \cdot S_{III})$		
$a := h \cdot S_I + S_{II}$		
$u \in_R GF(2^\ell)$		
	$\xrightarrow{(h,a,u)}$	
		`accept` if
		$a = h \cdot S_I + S_{II}$
		`stop` otherwise
		$S'_B := \mathrm{LSB}_r(h \cdot S_{III})$
		$v := f_u(S_I \vert\vert S_{II})$
	$\xleftarrow{v}$	
`accept` if		
$v = f_u(S_I \vert\vert S_{II})$		

Here, S_I, S_{II}, and S_{III} are $(n/3)$-bit strings, whereas $S_0, \ldots, S_{N/\ell-1}$

are ℓ-bit strings. Recall that $h \in_R GF(2^{n/3})$ means that h is chosen randomly from $GF(2^{n/3})$ according to the uniform distribution. All the computations are carried out in the fields $GF(2^{n/3})$ and $GF(2^{\ell})$, respectively.

Theorem 6.19 states that Protocol UH allows for privacy amplification secure against active adversaries whenever the Rényi entropy, from Eve's point of view, of S is greater than two thirds of the length of S. Moreover, the length of the resulting secret key S' can be roughly equal to the excess, i.e., to

$$H_2(S) - (2/3) \cdot \mathrm{len}(S)$$

(see Figure 6.12).

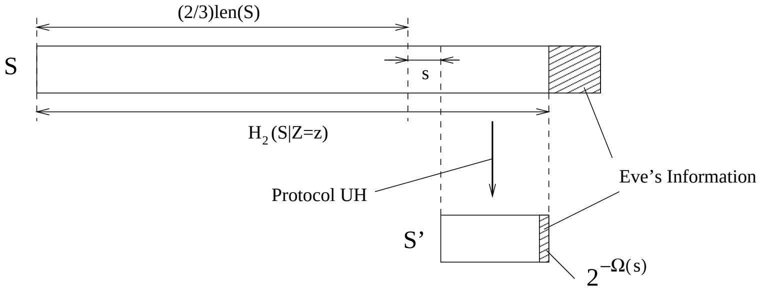


Figure 6.12: Analysis of Protocol UH

Theorem 6.19. *Let n, t, ℓ, and s be positive integers such that 3 divides n, ℓ divides $2n/3$, and $n > tn > 2n/3 + s$. Then Protocol UH is an*

$$(n\ ,\ \mathcal{D}_{n,2,tn}\ ,\ (t-2/3)n - s\ ,\ \varepsilon\ ,\ \delta)\ -\ \textit{protocol}$$

for privacy amplification by communication over an insecure and non-authentic channel for

$$\begin{aligned} \varepsilon &= r \cdot 2^{-(s/3-1)} + 2^{-s/3}/\ln 2 \\ \delta &= 2^{-(t-2/3)n/2} + 3 \cdot 2^{-(t-2/3)n/4} \\ &\quad +3 \cdot 2^{-(3\ell/4n)(1-3\ell/2n)(t-2/3)n} + \frac{2n}{3\ell 2^{\ell}}\ . \end{aligned}$$

Proof. Let $z \in \mathcal{Z}$ be the particular value known to Eve. We first assume that Eve is a *passive* wire-tapper. Let $(h,a) = (h, h \cdot S_I + S_{II})$ be the message sent from Alice to Bob, and let $\mathcal{E}$ be the event that

$$H_2(S_{III} \mid S_I = s_I, S_{II} = s_{II}, Z = z) \geq (t - 2/3)n - 2s/3 \ . \qquad (6.11)$$

Then $\mathcal{E}$ has, according to Lemma 6.16, probability at least $1-2^{-(s/3-1)}$. Let $r := (t-2/3)n - s$, and let $S' := \mathrm{LSB}_r(h \cdot S_{III})$. Because of (6.11), Theorem 4.2 implies that

$$\begin{aligned} H(S' \mid HA, \mathcal{E}, Z = z) &\geq H(S' \mid HAS_IS_{II}, \mathcal{E}, Z = z) \\ &= H(S' \mid HS_IS_{II}, \mathcal{E}, Z = z) \\ &\geq r - \frac{2^{-s/3}}{\ln 2} \ . \end{aligned}$$

We have used that $I(S_{III}; HA|S_IS_{II}, Z = z) = 0$. We conclude

$$\begin{aligned} H(S'|HA, Z = z) &\geq \mathrm{P}[\mathcal{E}] \cdot \left(r - \frac{2^{-s/3}}{\ln 2} \right) \\ &\geq r - r \cdot 2^{-(s/3-1)} - \frac{2^{-s/3}}{\ln 2} \\ &=: r - \varepsilon \ . \end{aligned}$$

Let us now consider the case where Eve is an *active* attacker. First, Lemma 6.14 implies that

$$H_2(S_I||S_{II} \mid Z = z) \geq (t - 1/3)n = n/3 + (t - 2/3)n \ .$$

Therefore, by Lemma 6.17, the probability of a successful active attack of the message authentication with strongly-universal hashing is upper bounded by

$$2^{-(t-2/3)n/2} + 3 \cdot 2^{-(t-2/3)n/4} \ .$$

On the other hand, we have to give an upper bound on the probability that Eve correctly guesses $v = f_u(s_I||s_{II})$. As above, we conclude first that

$$H_2((S_I||S_{II}) \mid H = h, A = a, Z = z) \geq (1 - 3\ell/2n)(t - 2/3)n \qquad (6.12)$$

holds with probability at least $2^{-(3\ell/4n)(t-2/3)n+1}$. If (6.12) holds, then by Lemma 6.18, the probability of correctly guessing v is at most

$$2^{-(3\ell/4n)(t-2/3)n(1-3\ell/2n)} + 2n/(3\ell 2^{\ell}) \ .$$

Hence, by the union bound, the success probability of an active attack is upper bounded by

$$2^{-(t-2/3)n/2} + 3 \cdot 2^{-(t-2/3)n/4} + 3 \cdot 2^{-(3\ell/4n)(1-3\ell/2n)(t-2/3)n} + \frac{2n}{3\ell 2^{\ell}} .$$

□

Corollary 6.20, which is an asymptotic version of Theorem 6.19, follows immediately from Theorem 6.19. Note that the required divisibility conditions can be satisfied by publicly adjoining a number of 0-bits at the end of the string. Then Theorem 6.19 can be applied for an appropriate choice of the parameters s and ℓ, both of order $\Theta(n)$, where $s \leq \Delta \cdot n$.

Corollary 6.20. *Let $2/3 < t < 1$ and $\Delta > 0$ be constants. Then Protocol UH is, for sufficiently large n and for an appropriate choice of the parameters, a*

$$\textit{robust}\ \left(n\,,\ \mathcal{D}_{n,2,tn}\,,\ \ (t-2/3-\Delta)n\,,\ 2^{-\Omega(n)}\,,\ 2^{-\Omega(n)}\right)\ -\ \textit{PA-protocol}\ .$$

6.3.6 Message Authentication with a Partially-Secret Key II: Short Messages and The Power of Feedback

One limitation of Protocol UH is due to the fact that the message to be transmitted and authenticated, i.e., the description of the function from the universal class, is as long as the string that finally forms the input to the hashing. As described in Section 4.3.2, there exist however methods for privacy amplification or, more generally, for "distribution uniformizing," that are more efficient than universal hashing with respect to the required amount of random (message) bits, namely so-called *extractors*.

The use of extractors for privacy amplification allows to reduce the size of the message to be transmitted and authenticated down to a small constant fraction of the length of the authentication key. In this case, a challenge-response authentication method, similar to the method described in Section 6.3.4, can be used: The message is not authenticated by the sender, but reconfirmed by the receiver. Intuitively, the use of such an authentication method puts the adversary, who must answer a given challenge instead of authenticate an incorrect message m' of his own choice, into a much less comfortable position.

Lemma 6.21 states that the interactive authentication method is secure against an active attacker whose Rényi entropy exceeds half the length of the authentication key. Note that the new authentication method has an important advantage compared to strongly-universal hashing in the context of authentication with a partially-secret key. When using the latter method, a correct message-authenticator pair reveals, roughly speaking, half the information about the key (namely a linear equation, two of which are sufficient to determine the key). Hence the key is "used up" after one application. With the interactive method however, only an arbitrarily small constant fraction of information about the key is gained by an opponent observing a message (i.e., challenge) together with its authenticator (i.e., response). This implies that the same key can be used for secure authentication with a partially-secret key many times. (Note that the straight-forward generalizations of SU hashing, using higher-degree polynomials to compute the authenticators, do not have the required properties with respect to an only partially-secret key.)

Lemma 6.21. *Let N and ℓ be integers such that 2ℓ divides N and $2^\ell \geq N/\ell$ holds, and let K be a random variable with range $\mathcal{K} \subseteq GF(2^N)$. Let further for any $d \in GF(2^\ell)$ the function $f_d : \{0,1\}^N \to \{0,1\}^\ell$ be defined as in Lemma 6.18. Assume that there exists a (possibly probabilistic) function s, mapping $GF(2^\ell)$ to $GF(2^\ell)$,*

$$s : d \mapsto s(d) =: d' ,$$

such that $d' \neq d$ holds for all d, and such that given $f_{d'}(K)$, the value $f_d(K)$ can, for $d \in_R GF(2^\ell)$, be guessed correctly (with some strategy) with probability α, taken over the distribution of K, the choice of d, and the coin tosses of the guessing strategy. Then

$$H_2(K) \leq \frac{N}{2} - \frac{2N}{\ell} \cdot \log\left(\alpha - \frac{N/\ell}{2^\ell}\right)$$

or, equivalently,

$$\alpha \leq 2^{-(\ell/2N)\cdot(H_2(K)-N/2)} + \frac{N/\ell}{2^\ell} .$$

Proof. Note first that, by the same arguments as used in the proof of Lemma 6.18, we can assume without loss of generality that the function $d'(d)$ and the strategy of guessing $f_d(k)$ from $f_{d'}(k)$ are deterministic, and conclude that there exist distinct elements $d_1, \ldots, d_{N/\ell}$ of $GF(2^\ell)$

such that $f_{d_i}(k)$ is guessed correctly from $f_{d'_i}(k)$, where $d'_i := d'(d_i)$, for all $i = 1, \ldots, N/\ell$ simultaneously with probability at least

$$\left(\alpha - \frac{N/\ell}{2^\ell}\right)^{N/\ell} .$$

Let $\mathcal{E}\ (\subseteq \mathcal{K})$ be this event. We prove that $|\mathcal{E}| \leq \sqrt{|\mathcal{K}|}$.

By cancelling $N/2\ell$ of the pairs (d_i, d'_i) and renumbering the remaining pairs, we can obtain $N/2\ell$ pairs (d_i, d'_i) with the property that $d_i \notin \{d'_1, \ldots, d'_{i-1}\}$ holds for all $i = 1, \ldots, N/2\ell$. (In the worst case, all the pairs (d_i, d'_i) occur twice in different orderings. Then, every second pair (d_i, d'_i) must be cancelled.)

The event $\mathcal{E}$ has the property that

$$f_{d'_1}(k) = f_{d'_1}(k^*) \implies f_{d_1}(k) = f_{d_1}(k^*)$$

for all $k, k^* \in \mathcal{E}$. Otherwise $f_{d_1}(k)$ could not be guessed correctly from $f_{d'_1}(k)$ for all $k \in \mathcal{E}$. Hence $\mathcal{E}$ must be contained in a set $\mathcal{E}_1$ of the form

$$\mathcal{E}_1 = \bigcup_{a \in GF(2^\ell)} \left\{k \,:\, f_{d_1}(k) = b(a) \text{ and } f_{d'_1}(k) = a\right\}$$

for some function $b(a)$. Analogously, $\mathcal{E}$ must also be contained in sets $\mathcal{E}_i$, $i = 2, \ldots, N/2\ell$, of the same form (with d_1 and d'_1 replaced by d_i and d'_i, respectively), hence

$$\mathcal{E} \subseteq \bigcap_{i=1}^{N/2\ell} \mathcal{E}_i \ . \tag{6.13}$$

We show that the cardinality of the set on the right-hand side of (6.13) is $\sqrt{|\mathcal{K}|}$. First, observe that every set of at most $N/\ell\,(\leq 2^\ell)$ functions f_{d_i} is, for pairwise distinct $d_i \in GF(2^\ell)$, linearly independent over $GF(2^\ell)$ (the Vandermonde determinant is nonzero in this case, as shown in the proof of Lemma 6.18). We define

$$r_l := \left|\bigcap_{i=1}^{l} \mathcal{E}_i\right| \ .$$

From the linear independence of $\{f_{d_1}, f_{d'_1}\}$, we first conclude that $r_1 = 2^{N-\ell}$. Furthermore, the linear independence of $f_{d_{l+1}}$ from the set

$$\{f_{d_1}, \ldots, f_{d_l}, f_{d'_1}, \ldots, f_{d'_{l+1}}\}$$

(because $d_{l+1} \notin \{d_1, \ldots, d_l, d'_1, \ldots, d'_{l+1}\}$ according to the choice of the pairs (d_i, d'_i)) implies that

$$r_{l+1} = r_l / 2^\ell$$

for $l = 1, \ldots, N/2\ell - 1$. Note that this also holds if $d'_{l+1} = d_i$ or $d'_{l+1} = d'_i$ for some $i < l + 1$. We conclude that

$$|\mathcal{E}| \leq r_{N/2\ell} = 2^{N-(N/2\ell)\cdot\ell} = 2^{N/2} = \sqrt{|\mathcal{K}|} \ .$$

On the other hand, $\mathrm{P}[\mathcal{E}] = \sum_{k \in \mathcal{E}} P_K(k) \geq (\alpha - (N/\ell)/2^\ell)^{N/\ell}$. In the case where P_K restricted to $\mathcal{E}$ is the uniform distribution (this case maximizes the Rényi entropy) with probability $(\alpha - (N/\ell)/2^\ell)^{N/\ell}/|\mathcal{E}|$ or greater, we have

$$\begin{aligned} \sum_{k \in \mathcal{K}} P_K(k)^2 &\geq \sum_{k \in \mathcal{E}} P_K(k)^2 \\ &\geq |\mathcal{E}| \cdot \frac{(\alpha - (N/\ell)/2^\ell)^{2N/\ell}}{|\mathcal{E}|^2} \\ &\geq \frac{(\alpha - (N/\ell)/2^\ell)^{2N/\ell}}{2^{N/2}} \ , \end{aligned}$$

and the claim follows when the negative logarithm is computed on both sides. □

6.3.7 A Protocol Using Extractors

In this section we present Protocol EX for privacy amplification secure against active adversaries. This protocol uses an extractor function for privacy amplification and the interactive challenge-response authentication methods. One important difference to Protocol UH is that a shorter string is used as the authentication key. This allows for extracting a significantly longer string in many cases.

Let the binary string S (of length n) be composed by strings S_I and S_{II} of lengths n_1 and n_2, respectively. Assume that 2ℓ divides n_1 and let $d := n_1/\ell$. The substrings S_i, $i = 0, \ldots, d-1$, of S_I are all of length ℓ. The function E is an extractor to be specified below.

Protocol EX (Extractor)

Alice		**Bob**
$S = S_I \vert\vert S_{II}$		$S = S_I \vert\vert S_{II}$
$S_I = S_0 \vert\vert \cdots \vert\vert S_{d-1}$		$S_I = S_0 \vert\vert \cdots \vert\vert S_{d-1}$
$h \in_R GF(2^\ell)$		
	$\xrightarrow{h}$	
		$a := f_h(S_I)$
		$b \in_R GF(2^\ell)$
	$\xleftarrow{(a,b)}$	
if $a \neq f_h(S_I)$: `stop`		$S'_B := E(S_{II}, h)$
if $a = f_h(S_I)$:		
$c := f_b(S_I)$		
	$\xrightarrow{c}$	
`accept`		if $c = f_b(S_I)$: `accept`
$S'_A := E(S_{II}, h)$		if $c \neq f_b(S_I)$: `reject`

The following theorem states that Protocol EX can be much more effective that Protocol UH, in particular for strings with a rather high level of initial security. Note first that Protocol EX works under the same condition as Protocol UH: The Rényi entropy of the string, given Eve's knowledge, must be larger than two thirds of the length of the string. The length of the extractable key however can be equal to roughly

$$H_\infty(S) - 2(\mathrm{len}(S) - H_2(S))$$

instead of only the excess $H_2(S) - (2/3)\mathrm{len}(S)$ as for Protocol UH. Notice that this expression can be substantially greater, in particular if $H_2(S)$ is close to $\mathrm{len}(S)$. On the other hand, note that because $H_\infty(S)$ can be smaller (by a factor up to 2) than $H_2(S)$, Protocol EX can also be less effective than Protocol UH in certain cases. An illustration of the statement of Theorem 6.22 is given in Figure 6.13.

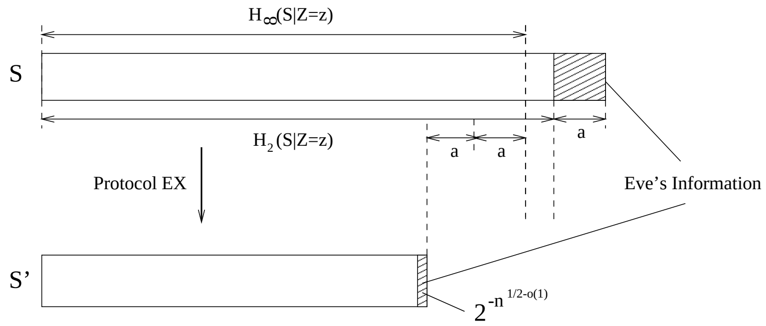


Figure 6.13: Analysis of Protocol EX

Theorem 6.22. *Let $0 < t' \leq t < 2/3$ and $\Delta > 0$ be constants. Then Protocol EX is, for sufficiently large n and for an appropriate choice of the parameters, a robust*

$$\left(n\,,\,\mathcal{D}_{n,2,tn}\cap\mathcal{D}_{n,\infty,t'n}\,,\;\;(t'-2(1-t)-\Delta)n\,,\,2^{-n^{1/2-o(1)}}\,,\,2^{-\Omega(n)}\right)\;-$$

PA-protocol.

Proof. Let $n_1 := (2(1-t)+\Delta/3)n$, $\ell := (\Delta/6)n$, and let $z \in \mathcal{Z}$ be the particular value known to Eve. We can assume without loss of generality that n_1 and ℓ are integers and that 2ℓ divides n_1. (Otherwise, n_1 and ℓ can both be chosen smaller, subject to $n_1 = (2(1-t)+\Theta(1))n$ and $l = \Theta(n)$, respectively, such that the conditions are satisfied.) Let now S_I be the first n_1 and S_{II} be the remaining $n_2 := n - n_1$ bits of S.

Assume first that Eve is *passive*. We give a lower bound on the min-entropy of the string S_{II} from Eve's point of view and given the entire communication C held over the public channel. Since this communication is, given S_I and $Z = z$, independent of S_{II}, we have

$$\begin{aligned} & H_\infty(S_{II}\,|\,C=c, S_I=s_I, Z=z) \\ & \quad = H_\infty(S_{II}\,|\,S_I=s_I, Z=z) \\ & \quad \geq (t'-2(1-t)-2\Delta/3)n \end{aligned} \qquad (6.14)$$

with probability $1-2^{-\Omega(n)}$. (Note that Alice and Bob could publish S_I at the end of the protocol, only helping a possible adversary.)

Let now E be the extractor function according to Corollary 4.10 with $d \leq \ell = \Theta(n)$,

$$\delta' = (t' - 2(1-t) - 2\Delta/3)n/n_2 \ ,$$

and

$$r \geq (\delta' - \Delta \cdot n/(3n_2))n_2 = (t' - 2(1-t) - \Delta)n \ .$$

For the choice $P_T = P_{S_{II}|C=c,Z=z}$ and $S' = E(S_{II}, V)$ (where V is composed by the first d bits of H in a fixed representation) we obtain, using (6.14) and $I(H;SZ) = 0$,

$$H(S' \,|\, C, Z = z) \geq r - 2^{-N^{1/2-o(1)}} \ .$$

We consider the case where Eve is an *active* adversary and give an upper bound on the probability of the event that Alice and Bob do not both reject and secret-key agreement has nevertheless not been successful. It is obvious that this can only occur if Eve can either guess $f_h(S)$ from some $f_{h'}(S)$ (where $h' \neq h$) or guess $f_b(S)$ correctly, where h and b are randomly chosen. The success probability δ of such an active attack is of order

$$\delta = 2^{-\Omega(n)} \ . \tag{6.15}$$

To see this, note first that

$$H_2(S_I \,|\, Z = z) \geq n_2/2 + \ell + \Delta/6$$

because of Lemma 6.14 and by the definitions of n_2 and ℓ. Then, the probabilities of the events that $f_h(S_I)$ is guessed correctly from $f_{h'}(S_I)$ (note that $\ell = \Theta(n_2)$), that

$$H_2(S_I \,|\, f_h(S_I) = f_h(s_I), Z = z) < n_2/2$$

(we call this event $\mathcal{E}$), and that $f_b(S_I)$ is guessed correctly, given that $\mathcal{E}$ does not occur, are all of order $2^{-\Omega(n)}$, because of Lemmas 6.21, 6.15, and 6.18, respectively. The bound (6.15) then follows from the union bound. □

Remark. Note that for the proof of Theorem 6.22, the following combination of the Protocols UH and EX is also sufficient: The key is partitioned as in Protocol EX, but the authentication techniques of Protocol UH are used for the shorter key, i.e., strongly-universal hashing and challenge-response confirmation (Section 6.3.4). However, Protocol EX has an important advantage compared to this protocol and to Protocol UH which is not stated explicitly in Theorem 6.22: In case of failure because of a detected active substitution attack, Protocol EX can (with roughly the same parameters) be restarted again and again ($O(1)$ times) with the same key until secret-key agreement eventually succeeds. The reason for the possibility of such multiple trials is that the observation of a correctly authenticated message reveals, in contrast to authentication with strongly-universal hashing, only a small fraction of the total information about the authentication key (see Section 6.3.6).

6.4 Independent Repetitions of a Random Experiment

The scenario where the parties have access to many independent repetitions of the same random experiment was studied in the context of active adversaries in [59]. A weaker version of the following definition, generalizing the notion of secret-key rate to this scenario, was given in [59].

Definition 6.8. The *secret-key rate* $S^*(X;Y||Z)$ *against active adversaries* (*robust secret-key rate* for short) is the least upper bound of the set of numbers $R \geq 0$ with the property that for all $\varepsilon, \delta > 0$, and for sufficiently large n, there exists a $(P^n_{XYZ}, \lfloor (R-\varepsilon)n \rfloor, \varepsilon, \delta)$-protocol for secret-key agreement by communication over an insecure and non-authentic channel. (Here, P^n_{XYZ} again stands for the distribution over $\mathcal{X}^n \times \mathcal{Y}^n \times \mathcal{Z}^n$ that corresponds to n independent realizations of the same random experiment.) ○

Clearly,

$$S^*(X;Y||Z) \leq S(X;Y||Z)$$

holds for all distributions P_{XYZ}. The following theorem expresses the robust secret-key rate $S^*(X;Y||Z)$ in terms of $S(X;Y||Z)$ and P_{XYZ}, and corrects a result of [59], where the same criterion was incorrectly given in connection with the weaker protocol definition. Quite surprisingly, $S^*(X;Y||Z)$ is equal to either $S(X;Y||Z)$ or 0, depending on whether X or Y is simulatable by Z.

Theorem 6.23. *Let P_{XYZ} be a distribution with $S(X;Y||Z) > 0$. Then $S^*(X;Y||Z) = 0$ holds if either $\mathrm{sim}_Y(Z \to X)$ or $\mathrm{sim}_X(Z \to Y)$ holds. Otherwise, we have $S^*(X;Y||Z) = S(X;Y||Z)$.*

Proof Sketch. Observe first that if $\mathrm{sim}_Y(Z \to X)$ or $\mathrm{sim}_X(Z \to Y)$ holds, secret-key agreement requires communication because every key Alice and Bob generate without communication is known at least as well to Eve, and that Eve can execute any protocol with one of the parties, making it accept although no secret key between Alice and Bob has been generated. Hence we have $S^*(X;Y||Z) = 0$ in this case.

Assume now that neither $\mathrm{sim}_Y(Z \to X)$ nor $\mathrm{sim}_X(Z \to Y)$ holds. In [59], it has been shown that in this case, Alice and Bob can send authenticated messages to each other over the insecure channel in such a way that they can detect any modification by Eve with high probability (and that also with high probability, the authentication is accepted in case of a passive adversary). The idea is that Alice [Bob] authenticates a message by sending certain realizations of X [Y], the choice of which depends on the message. Since Eve cannot simulate X [Y] with respect to Y [X], she is not able to generate a block of different realizations, i.e., to authenticate modified messages, without being detected (with high probability). It has also been shown in [59] that the number of realizations of X and Y required for this authentication can be made asymptotically negligibly small as compared to the number of realizations needed for the secret-key agreement in the passive-adversary case.

This technique allows to transform any protocol for secret-key agreement secure against passive adversaries into a protocol secure against active adversaries. First, all messages sent over the channel during the execution of the protocol by Alice or Bob are authenticated. However, this is not sufficient in general. In the resulting protocol Eve could possibly, by blocking the final message, achieve the situation where the sender of this message, say Alice, accepts, whereas Bob does not receive the last piece of information required to compute the common secret key.

This can be remedied by an additional message, which is sent from Bob to Alice (in this situation) after the last message of the original protocol from Alice to Bob has arrived properly. This final message contains a fixed block of realizations of Y, long enough such that Eve is not able to produce such a block which will not be rejected by Alice with high probability. In this new protocol, Alice computes the string S'_A after finishing the original protocol, but does not accept before receiving this final message from Bob. This protocol obviously satisfies the robustness property of Definition 6.1 because no party accepts before the key agreement is finished. We conclude that $S^*(X;Y||Z) = S(X;Y||Z)$ holds in this case. □

Corollary 6.24 completely characterizes the possibility of secret-key agreement secure against passive adversaries (in the independent-repetitions scenario) in terms of P_{XYZ} under the assumption that Conjecture 5.1 is true.

Corollary 6.24. *If Conjecture 5.1 is true, then* $S^*(X;Y||Z) > 0$ *holds if and only if both* $\mathrm{sim}_Y(Z \to X)$ *and* $\mathrm{sim}_X(Z \to Y)$ *do not hold.*

Proof. The forward implication is a consequence of Theorem 6.23 and the inverse direction follows from Theorems 6.6 and 6.23. □

6.5 Discussion

We have shown in this chapter that secret-key agreement can be possible even in the presence of active adversaries. However, the advantage of the authenticity of the public-discussion channel must in this case be replaced by some condition in terms of the distribution P_{XYZ}, called non-simulatability. It was shown that this a priori not easily verifiable condition can often be checked efficiently. The special case of privacy amplification clearly satisfies this non-simulatability restriction. It was shown that under certain conditions, which are stronger than those for the passive-adversary case, a (shorter) highly-secure string can be generated. The apparently necessary strengthening of the requirements, together with the loss of length of the final string, can be interpreted as the price the legitimate partners have to pay for the fact that the public channel is not authentic (see Figure 7.4).

Chapter 7

Conclusion

In the preceding chapters, we have investigated unconditionally-secure key agreement in a number of different scenarios. The very general question was considered what type of correlation between the knowledge of the legitimate partners and of the adversary is necessary for the possibility of secret-key agreement in the authentic- and non-authentic-channel settings. The main results are general conditions for both the possibility and impossibility of secure key agreement. An important conclusion is, due to the fact that many of these settings can be realistic in various contexts, that impracticality is far from being an inherent property of unconditionally-provable security. This should provide motivation for further research in the same direction.

7.1 Strengthening the Definitions of Secret-Key Rates

We have shown that in Wyner's original wire-tap-channel model and in Csiszár and Körner's as well as Maurer's settings, the requirements to the generated secret key can be substantially strengthened as compared to the original definitions, and that this has no effect on the amount of secret key which can be generated, nor on the efficiency of the protocols. (For Maurer's scenario, this modification of the secret-key-rate definition was already suggested in [63].)

More precisely, the *security* of the key can be much higher than previously demanded. The ability of establishing a secret key has, in accordance with many other definitions in information theory, been defined initially in terms of information *rates*. In particular, the adversary was tolerated to obtain information about the key of an amount proportional to the length of the key, hence not necessarily small in an absolute sense. Quite surprisingly, this definition can be improved without reducing the maximal rate at which a secret key can be generated.

This can be shown by repeating the key-generation procedure according to the weaker definition many times independently, and to use typical-sequences arguments. The main technique used to guarantee the virtually-perfect security of the resulting key is privacy amplification, either by universal hashing or by applying extractors.

Additionally, the resulting key can be required to be perfectly-uniformly distributed.

These results suggest that the attention can be completely restricted to the stronger and far more satisfactory definitions.

7.2 Linking One-Way and Interactive Key Agreement Secure Against Passive and Active Adversaries

In the general model, where the parties Alice and Bob, and an opponent Eve, receive independent realizations of random variables X, Y, and Z, respectively, there exist many results characterizing the possibility of secret-key agreement, both by one-way and interactive communication, secure either only against passive or even against active adversaries. In other words, connections have been derived between certain natural, intrinsic, and fundamental properties of joint distributions of three random variables in connection with information-theoretic security. These are the properties that the quantities $C_S(P_{YZ|X})$, $C_S(P_{XZ|Y})$, $S(X;Y||Z)$, $S^*(X;Y||Z)$, or $I(X;Y{\downarrow}Z)$ are nonzero, or that X (or Y) is simulatable by Z with respect to Y (X).

Note that the inequality

$$C_S(P_{YZ|X}) > 0 \tag{7.1}$$

is not precisely a property of a joint distribution P_{XYZ}, but of a conditional distribution $P_{YZ|X}$. However, inequality (7.1) is directly linked to the possibility of key agreement by one-way communication in the

"P_{XYZ} model." We introduce the *one-way secret-key rate*, denoted by

$$S(X \to Y \,||\, Z) \ ,$$

which is defined analogously to the secret-key rate $S(X;Y||Z)$, but where the communication over the public channel is restricted to messages sent from Alice to Bob.

Theorem 7.1 states simple connections between the one-way secret-key rate, the secrecy capacity, and certain information differences. (We write C_S for the secrecy capacity in the following, but because of Theorem 4.12, all the following results alternatively hold for the *strong* secrecy capacity $\overline{C_S}$.)

Theorem 7.1. *Let $P_{YZ|X}$ be a conditional probability distribution. The following statements are equivalent:*

1. $C_S(P_{YZ|X}) > 0$.
2. *There exists a distribution P_X such that $S(X \to Y \,||\, Z) > 0$ holds.*
3. *There exists a joint distribution P_{UX} of two random variables U and X such that $I(U;Y) - I(U;Z) > 0$ holds.*

Proof. The statements 1 and 3 are equivalent because of equation (4.19). Furthermore, statement 3 implies statement 2 by Theorem 4.7. Note that the proof of Theorem 4.7 actually implies that

$$S(X \to Y \,||\, Z) \geq \max_{YZ \to X \to U} (I(U;Y) - I(U;Z)) \ ,$$

where the maximum is taken over all the Markov chains $YZ \to X \to U$. First, communication is required only from Alice to Bob. Furthermore, it is clear that X can always be replaced by a random variable U if $YZ \to X \to U$ is a Markov chain.

Finally, one can show that statement 2 implies statement 3 by an argument similar to the proofs of the general lower bounds as given in [64]. □

Let us now represent all the derived connections between the mentioned properties in two diagrams. The first picture (see Figure 7.1) illustrates the relationship between the sets of distributions P_{XYZ} satisfying or not satisfying the properties of interest. We have made use of Theorems 5.1, 4.7, 6.23, and 7.1, Corollary 6.24, as well as of Lemmas 6.4 and 6.5. (Note that we need the slightly strengthened version

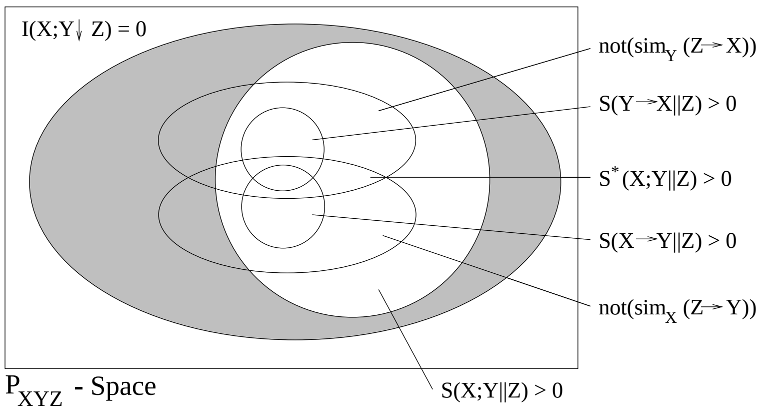


Figure 7.1: The Relationship Between Properties Related to Different Variants of Key Agreement

of Lemma 6.4, stating that if $\mathrm{sim}_Y(Z \to X)$ holds, then not even a random variable V exists, where $XZ \to Y \to V$ is a Markov chain, such that $I(V;X) - I(V;Z) > 0$ holds. However, the proof of this variant is completely analogous.)

It is not known whether all the marked sets in Figure 7.1 are non-empty. Deciding this would require to prove, or disprove, Conjecture 5.1. The second diagram (see Figure 7.2) illustrates the, simpler, situation as it arises under the assumption that Conjecture 5.1 is true.

In Figure 7.2, all the occurring subsets of the P_{XYZ}-space are non-empty. In order to show this, we give examples of distributions lying in the subsets marked by 1, 2, 3, and 4. Examples of distributions lying in the remaining subsets can either be found easily (or by symmetry).

Although, as mentioned above, $\mathrm{sim}_Y(Z \to X)$ implies that no V exists that can be generated by sending Y over some channel and such that $I(V;X) - I(V;Z) > 0$ holds, it does, somewhat surprisingly, not imply that $I(X;Y) - I(X;Z) \leq 0$ holds, as the example illustrated in Figure 7.3 shows (where all the binary random variables are symmetric). This distribution is an example lying in subset 1.

The example distribution given in Section 6.2.3, corresponding to Scenario EC1 with

$$1 - h(\alpha) \leq r < 1 - 2\alpha \tag{7.2}$$

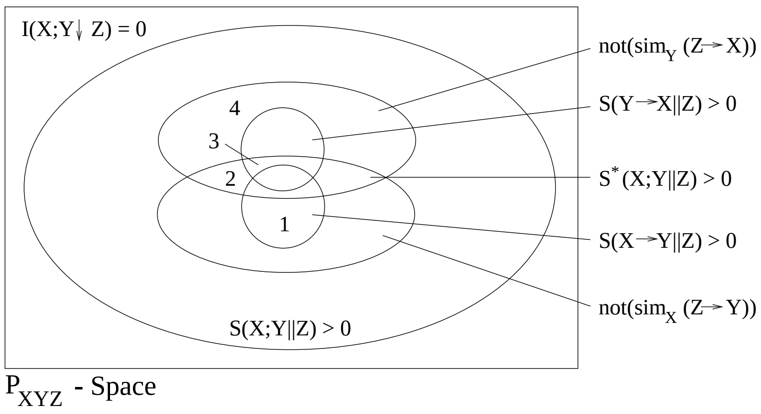


Figure 7.2: The Relationship Between the Properties of Interest Under the Assumption that Conjecture 5.1 is True

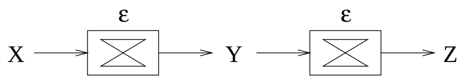


Figure 7.3: An Example Distribution of Subset 1

(and for which $S^*(X;Y||Z) > 0$ holds although

$$I(X;Y) < \min\{I(X;Z), I(Y;Z)\}$$

does) lies in subset 2. The two following variations of it are distributions in the sets 3 and 4. Let first Z be an erasure version of only X (instead of (X,Y)) with an erasure probability satisfying (7.2). This is a distribution lying in set 3. Let on the other hand Scenario EC1 be modified by an additional channel providing Eve with the correct value of X with probability

$$\frac{r' - r}{1 - r} ,$$

given that the original random variable Z equals the erasure symbol Δ. For the choice $r' = 1 - 2\alpha$, this new distribution is an example of subset 4.

Note that for these three example distributions, we must actually prove, for showing non-membership to the sets $S(X \rightarrow Y\,||\,Z) > 0$

or $S(Y \to X \,||\, Z) > 0$, not only that $I(X;Y) - I(X;Z) \leq 0$ (or the symmetric expression) holds, but that there exists no additional random variable U such that $YZ \to X \to U$ is a Markov chain and $I(U;Y) - I(U;Z) > 0$ holds. However, these stronger statements immediately follow from the weaker ones because both X and Y are *binary* random variables.

A surprising conclusion is the extent to which the required advantages in terms of P_{XYZ} for the possibility of secret-key agreement by one-way communication on one hand, and of key agreement in the presence of an active adversary on the other hand, are independent in general.

7.3 Privacy Amplification in the Presence of Active Adversaries

An additional problem in the context of secret-key agreement secure against active adversaries is privacy amplification by public discussion over a non-authentic channel. (This problem is not a special case of the independent-realizations scenario since only one string of a fixed length is known to Alice and Bob.)

Here, novel message authentication methods, using interaction and a possibly highly-insecure key, and that can be used repeatedly with the same key, combined with previous and new techniques for privacy amplification, were investigated. Based on these results, it was shown that privacy amplification is also possible in the presence of active adversaries. However, this is only true if the initial key satisfies certain conditions which are stronger than the condition that the string has a substantial Rényi entropy from the adversary's point of view, as for the passive-adversary case.

More precisely, we have described two protocols, Protocol UH and Protocol EX. Protocol UH is based on universal hashing and is successful as soon as the Rényi entropy of the partially-secret key, from the adversary's viewpoint, exceeds two thirds of the length of the string. Then the protocol distills a string whose length is roughly equal to this excess. Protocol UH is computationally extremely simple and works for strings of any length. For sufficiently long strings, Protocol EX, based on extractor functions, can be used. The condition to the initial key is the same as for Protocol UH, but the extracted highly-secret key is generally longer. An additional advantage of Protocol EX is that failed privacy-amplification attempts do (almost) not use up the key: The procedure can be repeated many times with the same key.

In Figure 7.4, the required conditions to the partially-secret key as well as the possible length of the resulting secret key when using Protocol UH or EX are illustrated and compared to the same quantities in the case of privacy amplification according to [5] in the passive-adversary case. This representation shows the price that has to be paid for authentication in the context of privacy amplification.

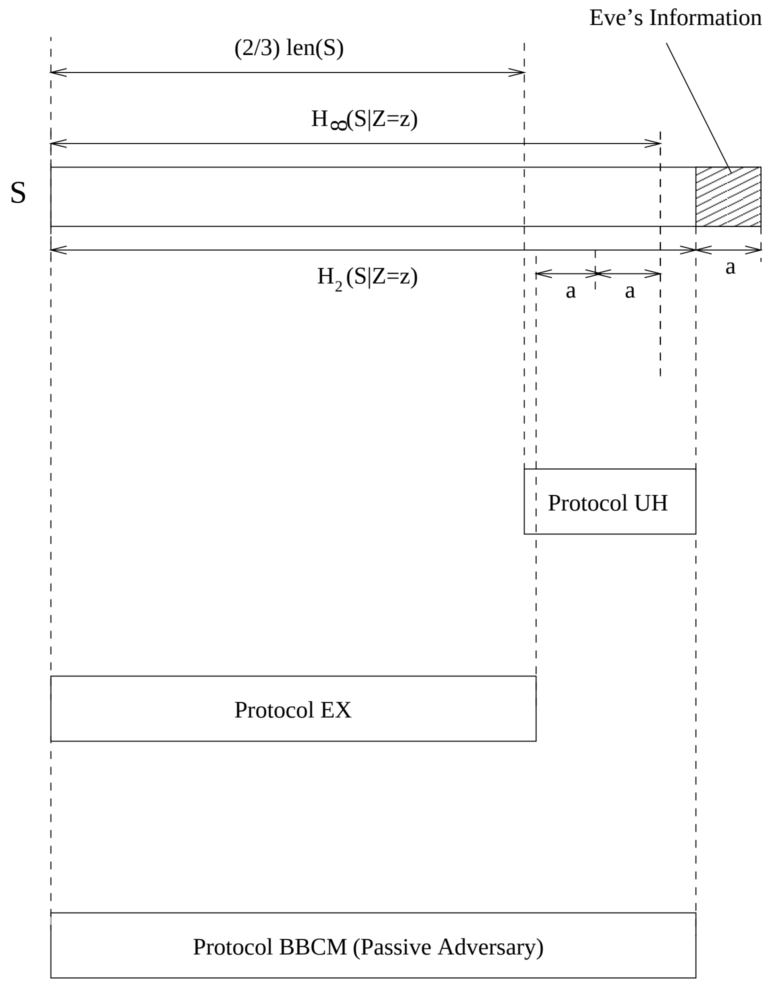


Figure 7.4: Privacy Amplification Secure Against Active Adversaries

Part III

Computationally-Secure Key Agreement

Chapter 8

Algebraic Preliminaries and Introduction

In Section 8.1 of this chapter we sketch some algebraic and number-theoretic definitions and facts required in the subsequent chapters. Good introductions are the books by Lang [50] (to algebra), Lidl and Niederreiter [55], Menezes [84] (finite fields), Geddes, Czapor, and Labhan [41] (computer algebra), Menezes [85] (elliptic curves), and Cohen [28] (computational number theory).

In Section 8.2 we describe the discrete-logarithm problem, which forms the basis of many public-key cryptographic protocols. One example is the Diffie-Hellman protocol introduced in Section 8.3. Finally, we review some previous and related results concerned with the security of the Diffie-Hellman protocol (Section 8.4).

Some parts of this introductory chapter are based on the overview article [71], covering many aspects of the Diffie-Hellman protocol and its security.

8.1 Groups, Fields, Polynomials, and Elliptic Curves

8.1.1 Groups

A *group* is a pair $(G, \cdot)$, where G is a set and $\cdot$ is mapping

$$\cdot : G \times G \longrightarrow G \ ,$$

called *group operation*, with the following properties. (Slightly abusing the notation, we sometimes only write G for the group, i.e., we identify the group with the set of elements.)

First, we have for all $a, b, c \in G$ that

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

(*associativity*). Hence we can write $a \cdot b \cdot c$ for this expression. Furthermore, there exists a *neutral element* $e \in G$ such that

$$e \cdot g = g \cdot e = g$$

holds for all $g \in G$. Finally, for every $g \in G$ there exists an *inverse* h of g such that

$$g \cdot h = h \cdot g$$

holds, and such that this group element is a neutral element of G.

The neutral element as well as the inverse of every group element are uniquely determined in every group. The inverse of g is denoted by g^{-1}. We write

$$g^n := \underbrace{g \cdot g \cdots g}_{n}$$

for repeated products.

The cardinality $|G|$ of the set G is called the *order* of the group. The *order* $\mathrm{ord}(a)$ *of a group element* $a \in G$ is the smallest positive integer i such that

$$a^i = e$$

holds (if such an integer exists). For every *finite group* (i.e., the set G is finite), and for every $a \in G$, we have

$$a^{|G|} = e \ .$$

Equivalently, the order of every element divides the order of the group.

A group is called *Abelian* or *commutative* if

$$a \cdot b = b \cdot a$$

holds for all $a, b \in G$. The group is *cyclic* if there exists an element $g \in G$, called a *generator*, such that for every $a \in G$, there exists an integer i such that $a = g^i$. We write

$$G = \langle g \rangle \ .$$

If g is a generator of the finite cyclic group G, then g^k is a generator as well if and only if

$$\gcd(k, |G|) = 1 \ .$$

Every cyclic group is Abelian.

Two groups G_1 and G_2 are called *isomorphic*, $G_1 \cong G_2$, if there exists a bijection Φ mapping G_1 to G_2 such that

$$\Phi(a \cdot b) = \Phi(a) \cdot \Phi(b)$$

holds for all $a, b \in G_1$. Then we have $\Phi(e) = e$ and $\Phi(g^{-1}) = \Phi(g)^{-1}$.

A subset H of G is a *subgroup* of G if it is closed under the group operation of G. For a set $\{g_1, g_2, \ldots, g_k\}$ of group elements,

$$\langle g_1, g_2, \ldots, g_k \rangle$$

denotes the minimal subgroup of G which contains all the elements $g_1, \ldots, g_k$.

For an Abelian group G and subgroup H, the *quotient group* G/H is defined as follows. First, the elements of G/H are the equivalence classes with respect to the relation

$$g_1 \sim g_2 :\Longleftrightarrow g_1 g_2^{-1} \in H \ .$$

The group operation on these classes can be executed on arbitrary group elements of the classes. The function

$$\pi : G \longrightarrow G/H \ ,$$

mapping each element to its equivalence class, is called the *canonical projection*.

For two groups G_1 and G_2, the *external product* $G_1 \times G_2$ is the set of pairs (g_1, g_2), $g_i \in G_i$, with component-wise group operation. An Abelian group G is the *internal product* of two subgroups H_1 and H_2 of G, $G = H_1 \times H_2$, if G is isomorphic to the external product $H_1 \times H_2$, and if $H_1 \cap H_2 = \{e\}$. The external and internal products of more than two groups are defined analogously.

A simple example of a finite group is $\mathbf{Z}_n$, which is the set

$$\{0, 1, \dots, n-1\}$$

together with addition modulo n. Note that $\mathbf{Z}_n$ is cyclic, and that every cyclic group of order n is isomorphic to $\mathbf{Z}_n$. On the other hand, every finite Abelian group G is isomorphic to a finite product of cyclic groups, i.e.,

$$G \cong \mathbf{Z}_{n_1} \times \mathbf{Z}_{n_2} \times \cdots \times \mathbf{Z}_{n_r}$$

for some $n_1, \dots, n_r$ satisfying $\prod n_i = |G|$, and such that n_{j+1} divides n_j for all j. This representation is unique, and r is called the *rank* of G. Moreover, G can be written as the internal product of cyclic subgroups:

$$G = \langle g_1 \rangle \times \langle g_2 \rangle \times \cdots \times \langle g_r \rangle \ .$$

We call $\{g_1, g_2, \dots, g_r\}$ a *generator set* of G. In Section 10.2.3 it is shown how to compute such a set efficiently.

Other important examples of finite groups are $\mathbf{Z}_n^*$ for integers n. The group $\mathbf{Z}_n^*$ is the set of numbers $a \in \{1, 2 \dots, n-1\}$ with

$$\gcd(a, n) = 1 \ ,$$

together with multiplication modulo n. Its order is

$$|\mathbf{Z}_n^*| = \varphi(n) := \prod (p_i - 1) p_i^{e_i - 1}$$

if $n = \prod p_i^{e_i}$ is the prime factorization of n. The function $\varphi(n)$ is called *Euler's totient function.* Further examples of finite groups are the multiplicative groups of finite fields (see Section 8.1.2), elliptic curves over finite fields (see Section 8.1.4), or the Jacobians of hyperelliptic curves over finite fields.

Let $m = \prod_{i=1}^{k} m_i$, where $\gcd(m_i, m_j) = 1$ if $i \neq j$. Then we have

$$\mathbf{Z}_m \cong \mathbf{Z}_{m_1} \times \mathbf{Z}_{m_2} \times \cdots \times \mathbf{Z}_{m_k} \tag{8.1}$$

and

$$\mathbf{Z}_m^* \cong \mathbf{Z}_{m_1}^* \times \mathbf{Z}_{m_2}^* \times \cdots \times \mathbf{Z}_{m_k}^* \ . \tag{8.2}$$

One (canonical) isomorphism Φ for (8.1) is given by the mapping

$$\Phi \ : \ a \mapsto (R_{m_1}(a), R_{m_2}(a), \ldots , R_{m_k}(a)) \ ,$$

where $R_l(x)$ denotes the remainder when dividing x by l. The isomorphism for (8.2) is the restriction of Φ to $\mathbf{Z}_m^* \subset \mathbf{Z}_m$. The inverse of Φ can be computed by

$$a = \Phi^{-1}((a_1, a_2, \ldots , a_k)) = R_m\left(\sum_{i=1}^{k} a_i M_i N_i\right) \ , \tag{8.3}$$

where $M_i := m/m_i$ and $N_i := M_i^{-1} \pmod{m_i}$ for $1 \leq i \leq k$. Formula (8.3) is called *Chinese remainder theorem.*

8.1.2 Fields

A *field* is a triple $(F, +, \cdot)$ such that $(F, +)$ is an Abelian group (where the neutral element is denoted by 0), and $(F^*, \cdot) := (F \setminus \{0\}, \cdot)$ is an Abelian group (with neutral element 1) such that additionally for all $a, b, c \in F$ the *distributive law*

$$a \cdot (b + c) = a \cdot b + a \cdot c$$

holds. The groups $(F, +)$ and $(F^*, \cdot)$ are called the *additive* and the *multiplicative* groups of F, respectively. The *characteristic* $\chi(F)$ of F is the smallest positive integer i such that

$$\underbrace{1 + 1 + \cdots + 1}_{i} = 0$$

holds (provided such an integer exists, and 0 otherwise). If not 0, $\chi(F)$ is a prime number.

The field $(F, +, \cdot)$ is called *finite* if the set F is. Every finite subgroup of the multiplicative group of a field is cyclic. In particular, the multiplicative group of a finite field is cyclic. A field F' with $F' \supseteq F$ is called an *extension field* of F, and is a vector space over F. In particular, if F' is finite, then its order $|F'|$ is a power of $|F|$.

Examples of finite fields are $\mathbf{F}_p$ for primes p defined by

$$(\{0, 1, \ldots, p-1\}, + \,(\mathrm{mod}\ p), \cdot\,(\mathrm{mod}\ p)) \ .$$

Every finite field F has a subfield which is isomorphic to $\mathbf{F}_{\chi(F)}$. Hence the order of a finite field is always a prime power. More precisely, for every prime power q, there exists (modulo isomorphisms) exactly one field of order q, denoted by $GF(q)$ ("Galois field") or $\mathbf{F}_q$.

The elements of a field F' can be represented with respect to some subfield F as linear combinations (with coefficients in F) of some fixed basis vectors. For a prime power q, a representation of $\mathbf{F}_{q^n}$ over $\mathbf{F}_q$ is for instance given by the polynomials in $\mathbf{F}_q[x]$ of degree at most $n-1$, where the multiplication is carried out modulo an irreducible polynomial over $\mathbf{F}_q$ of degree n. In other words, the monomials $1, x, x^2, \ldots, x^{n-1}$ correspond to the basis of $\mathbf{F}_{q^n}$ over $\mathbf{F}_q$.

Another representation is given by a *normal basis.* Let α be an element of $\mathbf{F}_{q^n}$, and let

$$\alpha_i := \alpha^{q^i}$$

for $i = 0, \ldots, n-1$. Then $\{\alpha_0, \ldots, \alpha_{n-1}\}$ is called a *normal basis* if it is linearly independent over $\mathbf{F}_q$, in which case α is called a *normal element.* Let $\alpha := (\alpha_0, \ldots, \alpha_{n-1})$. The matrix T in $(\mathbf{F}_q)^{n\times n}$ satisfying $\alpha_0\alpha = \alpha T$ is called the *multiplication table* of the basis.

8.1.3 Solving Polynomial Equations over Finite Fields

In this section we describe techniques for solving polynomial equations, i.e., finding roots of polynomials, over finite fields. We distinguish between the cases of *univariate* and *multivariate* polynomials.

Univariate Equations: The Cantor-Zassenhaus Algorithm

The roots of a polynomial $f(\gamma)$ over a finite field $\mathbf{F}_{p^k}$ can be computed in probabilistic polynomial time by the Cantor-Zassenhaus algorithm (see [84], [55]). The key idea of this algorithm is to factor the polynomial $f(\gamma)$ into

$$\gcd(f(\gamma), (\gamma+\delta)^{(p^k-1)/2} - 1) \ \text{ and } \ \gcd(f(\gamma), (\gamma+\delta)^{(p^k-1)/2} + 1)$$

for random $\delta \in \mathbf{F}_{p^k}$. This is repeated with different δ and leads to the linear factors of $f(\gamma)$.

Multivariate Equations: Gröbner Bases and Buchberger's Algorithm

Let F be a field and R be the ring $F[x_1, \ldots, x_n]$ of polynomials in $x_1, \ldots, x_n$ over F. Let further $p_i = 0$ (where $i = 1, \ldots, l$ and $p_i \in R$ for all i) be a system of polynomial equations. We also write $P = 0$, where $P := \{p_i\}$. Every basis of the generated ideal $\langle P \rangle$ in the ring R leads to an equivalent system of equations. Gröbner bases with respect to the lexicographic term ordering have the property that the system can be solved if univariate equations can be solved. The *lexicographic term ordering* is defined as follows:

$$\prod x_j^{i_j} <_L \prod x_j^{i'_j}$$

if and only if for some l, $i_j = i'_j$ holds for all $j = 1, \ldots, l-1$, and $i_l < i'_l$ holds.

We motivate the definition of Gröbner bases. Let f and g be polynomials, and let t be the leading term of g. One can reduce f modulo g if a monomial of f is a multiple of t, $f = \alpha t + r$. The *reduction* of f modulo g is then

$$f - \frac{\alpha t}{\mathrm{M}(g)} \cdot g \,, \tag{8.4}$$

where $\mathrm{M}(g)$ denotes the leading monomial of g. Let Q be a set of polynomials. The *reducer set* of the polynomial f with respect to Q are the polynomials g in Q with the property that the leading monomial of f can be reduced modulo g. There exists a simple algorithm for a maximal reduction of a polynomial f modulo a set Q of polynomials based on (8.4). Since R is not a principal ideal domain (if $n > 1$), the maximal reduction is not unique, and an element q of $\langle Q \rangle$ can be irreducible modulo Q. A *Gröbner basis* G (with respect to a term ordering) is defined and characterized by the following equivalent conditions:

1. Maximal reductions modulo G are unique.
2. If $f \in \langle G \rangle$, then f reduces to 0 modulo G.
3. For all f and g in G,

$$\operatorname{lcm}\left(\mathrm{M}(f), \mathrm{M}(g)\right) \cdot \left(\frac{f}{\mathrm{M}(f)} - \frac{g}{\mathrm{M}(g)}\right) =: \text{s-poly}(f, g)$$

reduces to 0 modulo G.

For given P, the third criterion leads to a simple algorithm for the computation of a Gröbner basis G of $\langle P \rangle$ by extending P.

Algorithm (Buchberger). *Choose any pair* (f_1, f_2) *in* $P \times P$ *and compute a maximal* P*-reduction of* s-poly(f_1, f_2). *If it is different from zero, extend* P *by this polynomial. Repeat the process for all pairs, including the pairs with components added to* P *during the execution of the algorithm.*

This algorithm can be improved by criteria indicating when s-poly(f, g) reduces to 0, which allow to decrease the number of s-polynomial reductions. The complexity of Gröbner basis computations is a subject of ongoing research. If the system $P = 0$ has only finitely many solutions over $\mathbf{C}$, the computation of a lexicographic Gröbner basis for $\langle P \rangle$ has complexity $O(D^{n^2})$, where n is a bound for the number of variables and polynomials and D is the maximal degree. The degrees of the polynomials in the Gröbner basis are of order $O(D')$, where

$$D' := (nD)^{(n+1)^{2^{s+1}}}, \tag{8.5}$$

and s is the dimension of the ideal, $s \leq n$.

The following are key properties of Gröbner bases. Let P be a set of polynomials and G a monic Gröbner basis for $\langle P \rangle$ (where *monic* means that the coefficients of the leading monomials of all the polynomials are 1).

Property 1. $P = 0$ has a solution if and only if $1 \notin G$.

Property 2. Let H be the set of all leading terms occurring in G. Then the following statements are equivalent:

1. P has finitely many solutions over $\mathbf{C}$,
2. For all i, there exists m_i such that $(x_i)^{m_i} \in H$.

The first property is a criterion for solvability, and the second property implies, when using the lexicographic ordering, that a subset of the equations coming from the polynomials of the Gröbner basis is a system of triangular form and can be solved if univariate polynomial equations can be solved. The fact that $(x_i)^{m_i}$ is the leading term of a polynomial implies that the variables $x_1, \dots, x_{i-1}$ do *not* occur in the polynomial. For example the polynomial with $(x_n)^{m_n}$ as leading term is *univariate* (with the only variable x_n). Analogously, there is a polynomial containing x_{n-1} and x_n only, etc.

8.1.4 Elliptic Curves

Let F be a field (whose characteristic is not 2 or 3) and let $A, B \in F$ with $4A^3 + 27B^2 \neq 0$ (in F). The elliptic curve $E_{A,B}(F)$ (with parameters A and B in F) is the set

$$\{(x, y) \in F^2 : y^2 = x^3 + Ax + B\} \cup \{\mathcal{O}\}$$

(the additional point $\mathcal{O}$ is the *point at infinity*). Together with the following operation, called (and written as an) addition, on the set of points, $E_{A,B}(F)$ forms an Abelian group of rank at most 2.

First, $\mathcal{O}$ is defined to be the neutral element,

$$P + \mathcal{O} = \mathcal{O} + P = P \; .$$

Then, if $P = (x, y)$,

$$-P := (x, -y) \; .$$

Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be such that $Q \notin \{P, -P\}$. Then we have for $(x_3, y_3) = P + Q$

$$\begin{aligned} x_3 &= \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \\ y_3 &= -y_1 + \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) \; . \end{aligned}$$

Finally, if $P = Q$, the addition is defined by $2P = (x_3, y_3)$, where

$$\begin{aligned} x_3 &= \left(\frac{3x_1^2 + A}{2y_1}\right)^2 - 2x_1 \\ y_3 &= -y_1 + \left(\frac{3x_1^2 + A}{2y_1}\right)(x_1 - x_3) \; . \end{aligned}$$

For the case $F = \mathbf{R}$, this group operation and the point at infinity have intuitive geometric interpretations, as shown in Figure 8.1.

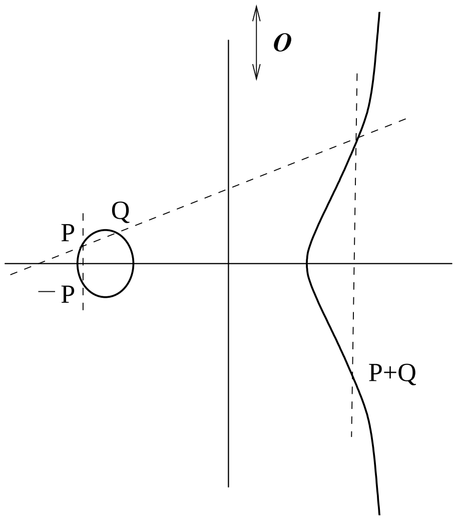


Figure 8.1: Addition and Inversion of Points of an Elliptic Curve

Alternatively to *affine coordinates* (x, y), the points of an elliptic curve can be represented in *projective coordinates* $(\lambda x, \lambda y, \lambda)$ (for arbitrary $\lambda \in F^*$, where $\mathcal{O} = (0, 1, 0)$). An advantage of this representation is that the addition of points does not require divisions in F.

Let now F be a *finite* field. Then $E_{A,B}(F)$ is a finite group, and it is well-known that for any $A, B \in \mathbf{F}_p$

$$p - 2\sqrt{p} + 1 \leq |E_{A,B}(\mathbf{F}_p)| \leq p + 2\sqrt{p} + 1$$

holds. Conversely, for every

$$d \in [p - 2\sqrt{p} + 1, p + 2\sqrt{p} + 1] \ ,$$

there exists a cyclic elliptic curve over $\mathbf{F}_p$ with order d [101].

8.2 The Discrete-Logarithm Problem

8.2.1 Definition

The security of most public-key cryptosystems is based on the (assumed) hardness of *factoring integers*, on the difficulty of computing *discrete logarithms* in certain groups, or on both. We describe the second problem and mention some algorithms for solving it. For a good introduction and overview, see [82].

Definition 8.1. Let $G = \langle g \rangle$ be a finite cyclic group. The *discrete-logarithm problem* (DLP for short) in G is, for given $a = g^x$, to compute x. ○

It is widely believed that there exist groups G for which the DLP is a hard problem. In contrast to the integer-factoring problem, whose hardness is solely supported by the fact that no one has published an efficient solution so far, some evidence suggests that the DLP is intrinsically a hard problem. This is one of the reasons why DLP-based systems are generally considered more secure than integer-factoring-based schemes such as RSA [100]. In Chapter 9, the (generic) DLP is described as a special case of a generalized index-search problem.

8.2.2 Solving the DLP in Specific Groups

For the group $\mathbf{Z}_n$, the DLP can be efficiently solved by the extended Euclidean algorithm. Although every cyclic group $G = \langle g \rangle$ of order n is isomorphic to $\mathbf{Z}_n$, this does *not* imply that the DLP is easy for all groups G. The reason is that the *existence* of an isomorphism does not imply that this bijection is efficiently computable in both directions. Indeed, the DLP is exactly the problem of evaluating the inverse of the isomorphism Φ given by

$$\Phi : \mathbf{Z}_n \longrightarrow G\,,\ x \mapsto g^x\ .$$

This shows that the hardness of the DLP is not a matter of the *group structure*, but of the *representation* of the group elements. "Different groups," such as additive or multiplicative groups of finite fields or elliptic curves, are different ways of representing the group elements.

For the groups $\mathbf{Z}_n^*$ and $\mathbf{F}_q^*$, there exist so-called *index-calculus methods* for solving the DLP in subexponential time. The index-calculus method for the multiplicative group of a prime field for instance is based on the fact that the group elements of $\mathbf{Z}_p^*$ can be interpreted as integers, which can be easily factored when they consist only of small prime factors. Examples are the *number-field sieve* and *Coppersmith's algorithm*, which allow to compute discrete logarithms in $G = \mathbf{Z}_p^*$ and $G = \mathbf{F}_{2^m}^*$, respectively, in time

$$e^{O((\log|G|)^{1/3}(\log\log|G|)^{2/3})}\ .$$

The number-field sieve is also the fastest known integer-factoring algorithm. It should be pointed out that none of these running times has been rigorously proven.

8.2.3 Generic Algorithms for the DLP

In contrast to algorithms especially designed for certain (classes of) groups, there exist *universal* or *generic algorithms*, working for all groups (of certain orders). The generic DLP (more precisely, generic algorithms solving the DLP) and related problems will be extensively studied in Chapter 9.

For certain (particular) groups the fastest known methods for solving the DLP are the generic algorithms. Examples of such groups are non-supersingular elliptic curves and Jacobians of hyperelliptic curves, which were proposed by Miller [87] and Koblitz [49], [48] to be used in discrete-logarithm-based cryptosystems. They appear to have the advantage that shorter secret keys are sufficient for the same security level. Menezes *et. al.* [83] have shown that the DLP in a *supersingular* elliptic curve over a finite field can be efficiently reduced to the same problem in the multiplicative group of an extension field of small degree.

For groups of order n, the *Pohlig-Hellman-Shanks algorithm* is generic and matches Shoup's [109] lower bound $\Omega(\sqrt{p})$ (where p is the largest prime factor of n) on the running time of every generic algorithm solving this problem (see Chapter 9 for a discussion of Shoup's result and for a description of the algorithm).

Other examples of generic discrete-logarithm algorithms are Pollard's ρ- and λ-methods [97]. Heuristic arguments suggest that *Pollard* ρ has approximately the same running time as the Pohlig-Hellman-Shanks method, but this has not been rigorously proved. Its advantage is that virtually no memory space is required. *Pollard* λ allows to find a discrete logarithm that is known to lie in a fixed interval $[A, B]$ of length $L = B - A$ in heuristic expected time $O(\sqrt{L})$ (instead of $O(\sqrt{L}\log L)$ by a straight-forward generalization of the baby-step giant-step method). For a description of this algorithm, see Section 9.4.1.

8.3 The Diffie-Hellman Protocol

8.3.1 Key Exchange over an Insecure Channel

In 1976, Whitfield Diffie and Martin Hellman published their celebrated paper [35] which initiated a revolution in cryptography. Diffie and Hellman can be seen as the founders of modern cryptography. The theoretical concepts of a public-key cryptosystem and of a digital signature were described by Diffie and Hellman, and have been realized only two years after their paper by Rivest, Shamir, and Adleman in the RSA system [100]. However, Diffie and Hellman presented the first protocol with public-key properties, the so-called *Diffie-Hellman (DH) protocol* for key distribution. An earlier protocol due to Merkle [86], called *Merkle's puzzles*, achieved the same goals, but the DH protocol enjoys the better ratio between security and efficiency. Both protocols allow two parties Alice and Bob, who are connected by an authenticated but otherwise insecure channel, to generate a secret key which is (believed to be) difficult to compute for an adversary Eve overhearing the communication between Alice and Bob.

The DH protocol works as follows. Let $G = \langle g \rangle$ be a finite cyclic group of order $|G|$. In order to establish a mutual secret key, Alice and Bob secretly choose integers s_A and s_B, respectively, at random from the interval $[0, |G|-1]$. (It is not necessary that Alice and Bob know the order of the group G. If $|G|$ is not known, Alice and Bob choose their integers from a sufficiently large interval. Moreover, if no generator of G is known, one can find such a g by trial and error if the group order $|G|$ and its factorization $|G| = \prod q_i^{f_i}$ are known (see Section 10.2.3). However, it is sufficient for the DH protocol if g is an element of G of high order.) Then they compute secretly $a_A = g^{s_A}$ and $a_B = g^{s_B}$, respectively, and exchange these group elements over the insecure public channel. Finally, Alice and Bob compute

$$a_{AB} = a_B^{s_A} = g^{s_A s_B}$$

and $a_{BA} = a_A^{s_B} = g^{s_B s_A}$, respectively. Because of

$$a_{AB} = a_{BA} \ ,$$

this quantity can be used as a secret key shared by Alice and Bob. More precisely, they can apply a function mapping elements of G to the key space of a cryptosystem. For instance, they can use an appropriate block (e.g., the most or least significant bits of a_{AB}) as the secret key of a conventional block cipher. Figure 8.2 shows a mechanical analog of the Diffie-Hellman protocol.

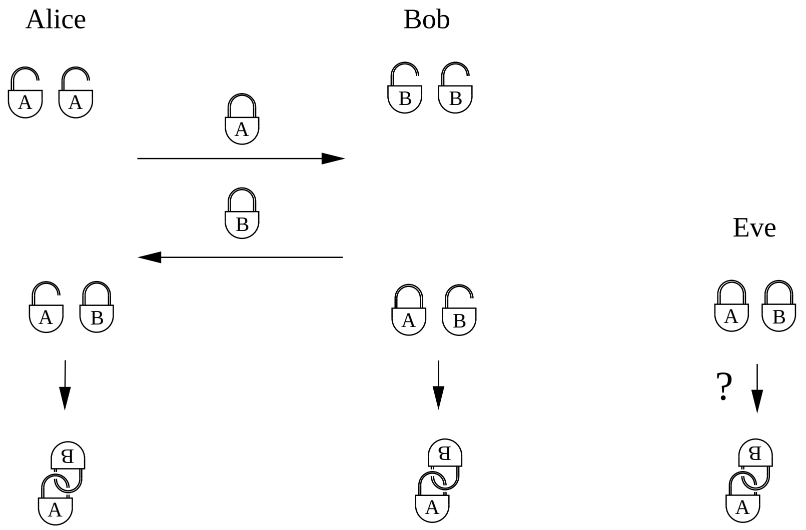


Figure 8.2: A mechanical analog of the Diffie-Hellman protocol. The padlocks have no keys. They are easy to lock, but hard to open. The two padlocks linked to each other are the secret key. The adversary Eve's task is to link two locked padlocks, which seems to require to open one of the locks first.

Specific groups that have been proposed for application in the DH protocol are the multiplicative groups of large finite fields (prime fields [35] or extension fields), the multiplicative group of residues modulo a composite number [80], [81], elliptic curves over finite fields [87], [49], the Jacobian of a hyperelliptic curve over a finite field [48], and the class groups of imaginary quadratic fields [17].

8.3.2 Security of the DH Protocol

Clearly, the security of the DH protocol relies on the hardness of the following computational problem.

Definition 8.2. Let $G = \langle g \rangle$ be a finite cyclic group. The *Diffie-Hellman problem* (DHP for short) in G with respect to the generator g is, for given g^x and g^y, to compute g^{xy}. ○

One way of solving the DHP is to compute x from g^x (i.e., to solve the DLP) first. Hence the security of the DH protocol is alternatively based on the hardness of the DLP. However, this is a priori a purely pessimistic statement: Whenever the DLP is tractable, then the DH protocol is insecure. It is not clear whether the most efficient way of solving the DHP is by solving the DLP first. It is conceivable that there exist groups for which the DHP is substantially easier than the DLP. The relationship between the DLP and the DHP is addressed in Chapter 11. It is shown that in many cases, the two problems are computationally equivalent. This is an additional advantage, in terms of security, over the RSA system [100]: Boneh and Venkatesan [12] have provided evidence that breaking RSA may be easier than factoring.

When using the generated DH key as the secret key in a cryptosystem, e.g., the one-time pad, then actually a stronger security requirement is necessary than the hardness of the DLP. For instance, the latter condition does not exclude that certain bits of the key are insecure. A stronger requirement is that the resulting DH key "looks random," i.e., is not efficiently distinguishable from a random string. This is a formalization of the fact that no partial information about g^{xy} can be extracted efficiently from g^x and g^y. If such a DH key is used as the secret key in the one-time pad, then the resulting cipher is "computationally-perfectly secret," i.e., the actual message cannot be distinguished from a random string in polynomial time with probability substantially exceeding 1/2. The following problem was first explicitly formulated in [13].

Definition 8.3. Let $G = \langle g \rangle$ be a finite cyclic group, and let g^x, g^y, g^c be chosen independently and randomly in G according to the uniform distribution. Given the triples (g^x, g^y, g^{xy}) and (g^x, g^y, g^c) in random order, the *Diffie-Hellman decision problem* (DHDP for short) is to decide, with probability of order $1/2 + \Omega(1)$, taken over the random input, which of the triples is the correct DH triple. ○

Note that the required success probability $1/2 + \Omega(1)$ in Definition 8.3 must only hold *in average*, taken over the random input, but not for every particular input. For instance, it is possible that the algorithm is successful only with probability 1/2 for a large portion of the set of all inputs. This property makes the DHDP (generically) easier than the

DHP for many groups. Example 8.1 shows for instance that if the group order is an even number, then the DHDP is always easy, whereas the DHP can nevertheless be (generically) hard. The technique used in the example generalizes in a straight-forward way to the case for which any small prime divides the group order.

Example 8.1. Consider a group $G = \langle g \rangle$ with order $|G| = 2p$ where p is a prime, and for which the DHP is hard. When given the random triple (g^x, g^y, g^c) and the DH triple (g^x, g^y, g^{xy}), then with probability $3/4$, the correct DH triple can be recognized. The reason is that from g^x, one can determine x modulo 2 by computing $(g^x)^p$, which is equal to e if $x \equiv 0 \pmod 2$ and to g^p $(\neq e)$ if $x \equiv 1 \pmod 2$. The parities of y and c can be obtained analogously, and with probability $1/2$

$$x \cdot y \not\equiv c \pmod 2$$

holds, and in this case the correct DH triple can be determined. Otherwise, the success probability is $1/2$. $\diamondsuit$

A conclusion of the technique used in Example 8.1 is that the DHDP can only be hard if $|G|$ is *free of small prime factors* (whereas the DHP can be hard in a group G if the group order $|G|$ contains *at least one* large prime factor). More precisely, the DHDP can be solved in time

$$O_s\left(\sqrt{p'}\right)$$

in any group G of order n, where p' is the *smallest* prime factor of n. The computational relationship between the DHP and the DHDP will be studied in Section 11.3.

8.4 Previous Results and Related Work

As we expose in Chapters 9 and 11, Shoup [109] has shown, thereby generalizing and extending results by Nechaev [90], that with respect to *generic* algorithms, the complexities of the DLP, DHP, and DHDP are of order $\Omega(\sqrt{p})$, $\Omega(\sqrt{p})$, and $\Omega(\sqrt{p'})$, where p and p' are the largest and smallest prime factors of the group order, respectively.

The computational relationship between the DHP and the DLP was studied by several authors. The results of den Boer [8], Maurer [62], and Boneh and Lipton [10] will be described, and compared to our results, below.

Shoup [109] presented a method for amplifying a faulty DHP solver. We give an alternative method in Section 11.1. Stadler [112] as well as Naor and Reingold [88] described such a self corrector for the DHDP.

Coppersmith and Shparlinsky [30] proved the impossibility of approximating the discrete logarithm function $g^x \mapsto x$ (modulo p) and the function $g^x \mapsto g^{x^2}$ (modulo p) by certain simple classes of functions such as low-degree polynomials.

A typical way of using the Diffie-Hellman protocol is to take a part of the generated secret key as the session key for encryption with a conventional block cipher. This key is usually shorter than the Diffie-Hellman key, and a natural method is to use the block consisting of its first k bits in a binary representation, the so-called most significant bits. However, it is conceivable that an adversary who is not able to break the Diffie-Hellman protocol can nevertheless compute these bits efficiently. Motivated by this fact, Boneh and Venkatesan [11] investigated the security of the most significant bits in the Diffie-Hellman protocol (and other schemes) in the groups $\mathbf{Z}_p^*$. They showed that the $O(\log p)$ most significant bits of the DH key are simultaneously as secure as the entire DH key.

Waldvogel and Massey [117] have studied the distribution of the DH key $g^{s_A s_B}$ under the assumption that the secret values s_A and s_B are chosen independently and uniformly in $[0, |G|-1]$. They showed that if the group order contains at least one large prime factor (the Diffie-Hellman protocol is insecure otherwise anyway), the distribution of the group element $g^{s_A s_B}$ is close to uniform over G. Moreover, when s_A and s_B are chosen uniformly in $\mathbf{Z}_{|G|}^*$ (instead of $\mathbf{Z}_{|G|}$), then the resulting key is perfectly-uniformly distributed in the set $\{g^s \, : \, s \in \mathbf{Z}_{|G|}^*\}$.

Van Oorschot and Wiener [92] have investigated the risk of choosing short exponents in the DH protocol. They presented a combination of Pollard λ and the Pohlig-Hellman decomposition. Their conclusion is that it is dangerous to use short exponents in the group $\mathbf{Z}_p^*$ with randomly chosen p, because $p-1$ is likely to have a substantial smooth part. This can be avoided by using a subgroup of prime order for the Diffie-Hellman protocol, for instance by selecting a prime p such that $(p-1)/2$ is also prime. See Section 9.4.1 for a discussion of the method.

Chapter 9

The Index-Search Problem

In this chapter we define and study so-called index-search problems. The underlying computational model is purely probability-theoretic, and similar to the model of generic algorithms in groups introduced by Shoup [109] (see Chapter 10). Due to the close relationship of the two models, both the positive and the negative results (i.e., algorithms and lower bounds, respectively) of this section immediately carry over to generic group algorithms, in particular algorithms solving the DLP and closely related problems.

In Section 9.1 we define the special and general index-search problems. In Section 9.2, a lower bound on the complexity of the general index-search problem is proved. Section 9.3 studies a variety of different index-search problems, starting from the special case and moving to the general problem step by step. It is shown that algorithms exist for solving these problems, the complexity of which match the lower bound stated in Section 9.2. In Section 9.4, certain variants of the DLP are studied with respect to generic algorithms. Examples are the short-exponent DLP or the DLP with an infinitely powerful helping oracle.

The purpose of this chapter is to introduce various generic algorithms solving index-search problems (in particular those for solving the DLP or variants of it, which will play a central role in Chapters 10 and 11). Furthermore, it is demonstrated for each algorithm what structure precisely is necessary for applying it.

The index-search problem was first defined in [70], and some results of this chapter were presented at ISIT '98 [72]. The results of Section 9.4 are new and have not been published before.

9.1 The Special and the General Index-Search Problem

Let $(a_i)_{i \in I}$ be a list of elements of some set S. The index-search problem was defined in [70] as follows: Given $b \in S$, find an index $i \in I$ such that $b = a_i$. Motivated by the model of generic algorithms, we formalize this problem as follows.

Definition 9.1. Let S be a set of cardinality n (for example a set of arbitrary binary strings), and let a bijective encoding function

$$\sigma : \{0, \ldots, n-1\} \longrightarrow S$$

be chosen from the set of all such functions at random (according to the uniform distribution). Then the *index-search problem* for sets of size n (ISP(n) for short) is, given $\sigma(x)$ (for randomly chosen x), to compute x. The algorithm is allowed to make calls to an oracle that computes the function σ, i.e., that takes as input $c \in \mathbf{Z}_n$ and outputs $\sigma(c)$. ○

Notice that the introduced computational model is probability- or information-theoretic: Given $\sigma(x)$, the algorithm must reduce the (initially maximal) uncertainty about x by calling the given oracle and by equality tests only. In other words, the assumption of the randomly-chosen encoding function ensures that the algorithm is universal and does not exploit any particular property of σ, i.e., of the representation of $\mathbf{Z}_n$.

In the case of the *general* ISP, we additionally allow the algorithm to make queries to oracles computing

$$\sigma(g(x_1, \ldots, x_r))$$

from $\sigma(x_1), \ldots, \sigma(x_r)$ for certain fixed functions g. We will restrict ourselves to the case of *polynomial* functions g, assuming the natural ring structure on the index set $\{0, \ldots, n-1\}$.

Definition 9.2. Let S be a set with $|S| = n$, and let

$$P_i(x_1, x_2, \ldots, x_{s_i}) \ ,$$

for $i = 1, \ldots, l$, be polynomials in $\mathbf{Z}_n[x_1, x_2, \ldots, x_{s_i}]$. The *general index-search problem* for sets of size n, and with respect to $(P_1, \ldots, P_l)$ ($(P_1, \ldots, P_l)$-GISP(n) for short), is defined similarly to the ISP(n), but the algorithm is additionally allowed to call (for $i = 1, \ldots, l$) oracles answering the query $(\sigma(x_1), \ldots, \sigma(x_{s_i}))$ by $\sigma(P_i(x_1, \ldots, x_{s_i}))$. ○

It is intuitively clear that there is a direct correspondence between algorithms solving the $(x_1 + x_2)$-GISP(n) and generic algorithms for the DLP in groups of order n.

9.2 The Complexity of the GISP

In this section we prove a universal upper bound on the success probability of any, possibly probabilistic, algorithm solving the GISP. This bound depends on the running time of the algorithm, on the factorization of n, and on the degrees of the polynomials P_i.

Theorem 9.1. *Consider the $(P_1, \ldots, P_l)$-GISP(n). Let d_i be the total degree of the polynomial P_i. Every probabilistic algorithm that runs in time at most T outputs the correct answer with probability (taken over the random choices of x and σ and the coin tosses of the algorithm) at most*

$$\alpha \leq \min\left\{ \frac{(T+1)T(\max\{d_i\})^T + 2}{2p} , \frac{(T+1)T + 2}{2q} \right\} ,$$

where p and q are the largest and the largest multiple *prime factor of n, respectively. In the case where no oracle is given, i.e., $l = 0$, the success probability is at most*

$$\alpha \leq (T+1)/n .$$

Remark. Note that the bound of Theorem 9.1 only depends on the maximal total degree occurring. We could also have assumed that the algorithm can make calls to an oracle computing *arbitrary* multivariate polynomials of degree at most d for some $d \geq 1$.

For the proof we need the following lemma (see [105] or [109]).

Lemma 9.2. **[105]** *When given a nonzero polynomial $P(X_1, \ldots, X_k)$ over $\mathbf{Z}_{p^e}$ of total degree $d \geq 0$, the probability that $P(x_1, \ldots, x_k) = 0$ holds for random $(x_1, \ldots, x_k) \in \mathbf{Z}_{p^e}^k$ is at most d/p.*

Proof of Theorem 9.1. Before the algorithm has performed a call to one of the oracles, we have

$$H(x \mid \sigma(x)) = \log n$$

or in other words,

$$I(x\,;\,\sigma(x)) = 0\ .$$

This holds because x and the encoding function σ are randomly chosen. (Note that x and σ stand for the random variables, and not for the particular values occurring.)

Let us first consider the case where at least one additional oracle is given ($l \geq 1$). We analyze one particular execution of the algorithm, i.e., with fixed coin flips if the algorithm is probabilistic.

Let first p be the largest prime factor of n. We give a lower bound on the probability that the algorithm finally outputs x modulo p^e correctly. For $n = p^e s$ with $\gcd(s, p) = 1$ and $e \geq 1$, we can assume without loss of generality that $s = 1$ holds, i.e.,

$$n = p^e\ .$$

Clearly, this cannot decrease the success probability of computing x modulo p^e.

By interaction with the given oracles T times, the algorithm can compute a list of expressions

$$\sigma(f_j(x))$$

for $j = 1, \ldots, T+1$ (where we can assume $f_j(X) \not\equiv f_{j'}(X)$ for $j \neq j'$) with

$$f_1(X) = X$$

and for all $j \geq 2$ either

$$f_j(X) = c$$

for a constant $c \in \mathbf{Z}_n$ or

$$f_j(X) = P_i(f_{j_1}(X), \ldots, f_{j_{s_i}}(X))$$

for some i and $j_1, \ldots, j_{s_i} < j$. Clearly, the algorithm is free to perform different computations. However, given $\sigma(x)$ and all the oracles' answers, these computations and their results are statistically independent of σ, hence of x. Therefore, these additional computations have no influence on the subsequent information-theoretic arguments.

Let $\mathcal{E}$ be the event that

$$f_j(x) = f_{j'}(x)$$

holds for some $j \neq j'$, $1 \leq j, j' \leq T+1$, and let $\overline{\mathcal{E}}$ be the complementary event. We have

$$I(x\,;\,[\sigma(x), \sigma(f_2(x)), \ldots, \sigma(f_{T+1}(x))]\,|\,\overline{\mathcal{E}}) = 0\ . \qquad (9.1)$$

Equality (9.1) states that if all the encoded elements obtained previously are distinct, then the particular encodings of the elements are statistically independent of x. This follows from the fact that the encoding function σ is chosen randomly among all possible encodings of $\{0, \ldots, n-1\}$ with respect to S.

According to Lemma 9.2, the probability, for randomly chosen $y \in \mathbf{Z}_n$, that $f_j(y) = f_{j'}(y)$ holds for fixed $j \neq j'$ is at most d/p, where d is an upper bound on the degrees of f_j and $f_{j'}$. Thus the probability of the event $\mathcal{E}$ is upper bounded by

$$\mathrm{P}[\mathcal{E}] \leq \frac{(T+1)T \cdot (\max\{d_i\})^T}{2p}\ .$$

Notice that this argument, using the union bound, is also true for adaptive strategies of the algorithm. This implies that the probability that the algorithm correctly outputs x modulo p^e is at most

$$\mathrm{P}[\mathcal{E}] + \mathrm{P}[\overline{\mathcal{E}(\delta)}] \cdot \frac{1}{p^e \cdot \mathrm{P}[\overline{\mathcal{E}(\delta)}]} \leq \frac{(T+1)T(\max\{d_i\})^T + 2}{2p}\ . \qquad (9.2)$$

This is true because if $\mathcal{E}$ does not occur, then random guessing among the values x modulo p^e which are consistent with $\overline{\mathcal{E}(\delta)}$ is the optimal strategy.

Let now q be the largest *multiple* prime factor of n, i.e., $n = q^{e'}s'$ with $\gcd(s', q) = 1$ and $e' \geq 2$. As above, we can assume $n = q^{e'}$. Let

$$x \equiv x_0 + x_1 q + \ldots + x_{e'-1} q^{e'-1} \pmod{q^{e'}}$$

be the q-adic expansion of x modulo $q^{e'}$. We can restrict the attention to computing the coefficient $x_{e'-1}$ (where $e' - 1 \geq 1$). We assume

$$x \equiv x_{e'-1} q^{e'-1} (\mathrm{mod}\ q^{e'})\ .$$

By interacting T times with the given oracles, the algorithm can compute

$$\sigma(f_j(x_{e'-1}))$$

for $j = 1, \ldots, T+1$ with

$$f_1(X) = X \cdot q^{e'-1}$$

and for $j \geq 2$ either

$$f_j(X) = c$$

for some constant $c \in \mathbf{Z}_n$ or

$$f_j(X) = P_i(f_{j_1}(X), \ldots, f_{j_{s_i}}(X))$$

for some i and $j_1, \ldots, j_{s_i} < j$. The crucial fact is that $f_j(X)$ is a *linear* polynomial in X for all j. This follows by induction and from

$$q^{e'} \equiv 0 \pmod{n} \ .$$

Hence the probability of the event $\mathcal{E}'$ of a nontrivial collision (defined similarly to $\mathcal{E}$) is upper bounded by

$$\mathrm{P}[\mathcal{E}'] \leq \frac{(T+1)T}{2q} \ . \tag{9.3}$$

Therefore, the probability that the algorithm outputs $x_{e'-1}$ correctly is at most

$$\frac{(T+1)T+2}{2q} \ . \tag{9.4}$$

From (9.2) and (9.4) we can now conclude that the probability that the algorithm outputs x modulo n correctly is upper bounded by

$$\alpha \leq \min\left\{\frac{(T+1)T(\max\{d_i\})^T+2}{2p} \ , \ \frac{(T+1)T+2}{2q}\right\} \ .$$

Let us finally consider the special case where no oracle for the computation of a function P_i is given. Then the only quantities the algorithm can obtain in time T which are *not* statistically independent of σ are

$$\sigma(c_1), \sigma(c_2), \ldots, \sigma(c_T)$$

for certain (pairwise distinct) $c_1, \ldots, c_T$. With probability at least $1 - T/n$, we have $c_j \neq x$ for all $j = 1, \ldots, T$, and in this case,

$$H\left(x \mid \sigma(x), (c_1, \sigma(c_1)), \ldots, (c_T, \sigma(c_T))\right) = \log(n - T)$$

holds because σ is completely random. More precisely, the conditional distribution of x is the uniform distribution over the set

$$\{0, 1, \ldots, n-1\} \setminus \{c_1, c_2, \ldots, c_T\} \ .$$

Thus the success probability α of the algorithm is at most

$$\alpha \leq \frac{T}{n} + \left(1 - \frac{T}{n}\right) \cdot \frac{1}{n - T} = \frac{T+1}{n} \ .$$

This concludes the proof of the theorem. □

The bound of Theorem 9.1 on the success probability of a probabilistic algorithm is in terms of the *worst-case* running time of the algorithm. The following corollary, which is an immediate consequence of Theorem 9.1, gives a lower bound on the *expected* running time of an algorithm solving the GISP.

Corollary 9.3. *The expected running time T of any probabilistic algorithm solving the $(P_1, \ldots, P_l)$-GISP(n), where the total degree of every polynomial P_i is at most $d\,(\geq 1)$, must satisfy both*

$$\frac{(2T+1)T(\max\{d_i\})^T + 1}{p} \geq \frac{1}{2}$$

and

$$\frac{(2T+1)T + 1}{q} \geq \frac{1}{2} \ ,$$

where p and q are the largest and largest multiple prime factors of n, respectively. Hence the running time T is of order

$$T = \Omega(\log p)$$

and

$$T = \Omega(\sqrt{q})$$

if $d \geq 2$*. If* $d = 1$*, we have*

$$T = \Omega(\sqrt{p}) \ .$$

For $d = 0$*, the running time is lower bounded by*

$$T \geq n/2 \ .$$

Proof. The bound for the case $d = 0$ is obvious. Let $d \geq 1$. Assume that a probabilistic algorithm exists which solves the problem in expected time T. We can construct a new algorithm by stopping the execution after time $2T$. This new algorithm has worst-case running time $2T$ and success probability at least $1/2$. The statement now follows from Theorem 9.1 applied to the new algorithm. □

9.3 Analysis of Specific GISPs

In this section we consider a number of specific GISPs and algorithms for solving them. Starting with the original ISP, we add structure to the problem step by step. With increasing power of the given computational oracles, the provable lower bound on the complexity becomes weaker, and algorithms are described that achieve the bounds. For the special case of the "linear GISP," corresponding to the DLP, these algorithms are the baby-step giant-step method and the Pohlig-Hellman decomposition, which are both of paramount importance in Chapters 10 and 11.

9.3.1 The Incremental GISP and the Baby-Step Giant-Step Algorithm

For the case of the special ISP, where no additional oracle besides the oracle computing σ is given, it is clear that the bound $\alpha \leq (T+1)/n$ of Theorem 9.1 is tight because random brute-force search (and random guessing unless $\sigma(c_i)$ is equal to $\sigma(x)$ for some $i = 1, \ldots, T$) exactly matches this bound.

Let us now consider the case where an additional oracle is given that computes a fixed cyclic permutation π of $\{0, \ldots, n-1\}$. More precisely, the oracle computes

$$\sigma(x_1) \ \rightsquigarrow \ \sigma(\pi(x_1)) \ .$$

We can assume without loss of generality that this permutation is

$$\pi(x_1) :\equiv x_1 + 1 \pmod{n} .$$

An example where the $(x_1 + 1)$-GISP(n) can occur is the computation of discrete logarithms in a cyclic group G with generator g in which it is possible to apply the group operation to an arbitrary group element and g (but not to two arbitrary group elements). In this case the bound of Corollary 9.3 on the running time of an algorithm solving the GISP is of order $\Theta(\sqrt{p})$, where p is the largest prime factor of n.

Shanks' *baby-step giant-step algorithm* is applicable in this situation and its running time is of order $O(\sqrt{n}\log n)$. The algorithm works as follows. First, the expressions

$$\sigma(x), \sigma(x+1), \ldots, \sigma(x+M)$$

are computed and stored in sorted order for $M := \lceil\sqrt{n}\rceil$. The sorting requires time $O(\sqrt{n}\log n)$. Then

$$\sigma(0), \sigma(M), \sigma(2M), \ldots$$

are computed (by calls to the oracle computing σ) until one of the expressions (say $\sigma(iM)$) is contained in the list $(\sigma(x+j))$. Then we have $x \equiv iM - j \pmod{n}$. The running time of this algorithm is $O(\sqrt{n}\log n)$ and memory space for storing $\sqrt{n}$ pairs $(\sigma(y), y) \in S \times \mathbf{Z}_n$ is required. Hence the bound given by Corollary 9.3 is asymptotically tight in this case at least if n is a prime number.

9.3.2 The Linear GISP, the DLP, and Pohlig-Hellman Decomposition

Let us now consider the

$$(x_1 + x_2)\text{-GISP}(n) ,$$

i.e., there is an oracle for

$$\sigma(x_1) ,\ \sigma(x_2) \rightsquigarrow \sigma(x_1 + x_2) .$$

Note that the oracle evaluating σ is not even required anymore in this case, because this can efficiently be done with the addition oracle, provided that $\sigma(1)$ is given.

The following algorithm, the so-called *Pohlig-Hellman decomposition* [95] combined with the baby-step giant-step method, achieves a running time of order $\Theta_s(\sqrt{p})$, which matches the lower bound of Corollary 9.3 for this case.

Theorem 9.4. *Let $n = \prod p_i^{e_i}$ be the prime factorization of n, and let p be the largest prime factor of n. Then the $(x_1 + x_2)$-GISP(n) can be solved in time*

$$O\left(\sum e_i(\log n + p_i)\right) = O_s(p) \ .$$

If memory space for storing $\lceil\sqrt{p}\rceil$ group elements is available, the running time can be reduced to

$$O\left(\sum e_i(\log n + \sqrt{p_i}\log p_i)\right) = O_s(\sqrt{p}) \ .$$

Proof. To solve $\sigma(x)$ for x, we first compute x modulo $p_i^{e_i}$ for all i. This is done by determining, modulo p_i, the coefficients x_{ij} of the p_i-adic representation of x modulo $p_i^{e_i}$,

$$x \equiv \sum_{j=0}^{f_i-1} x_{ij} p_i^j \pmod{p_i^{e_i}} \ .$$

The number x_{i0} can be obtained by comparing

$$\sigma((n/p_i)x) \ ,$$

which can be obtained by $O(\log n)$ calls to the addition oracle, with each of the elements

$$\sigma(0)\, ,\ \sigma(n/p_i)\, ,\ \sigma(2(n/p_i))\, ,\ \ldots\, ,\ \sigma((p_i - 1)(n/p_i)) \ . \tag{9.5}$$

The number of oracle calls for computing these expressions is

$$O(\log n + p_i) \ .$$

Assume now that $x_{i0}, \ldots, x_{i,k-1}$ have already been computed. The number x_{ik} can be found by testing equality of

$$\sigma((n/p_i^{k+1})(x - x_{i0} - p_i x_{i1} - \cdots - p_i^{k-1} x_{i,k-1}))$$

with the elements of the same list (9.5). Given x modulo $p_i^{e_i}$ for all i, Chinese remaindering (according to (8.3)) yields x modulo n. The complexity of the entire algorithm is

$$O\left(\sum e_i(\log n + p_i)\right) .$$

The baby-step giant-step time-memory trade-off (see Section 9.3.1) allows to speed up the search in the lists by a factor M when memory space for M pairs in $\mathbf{Z}_n \times S$ is used (these pairs can be sorted in time $O(M \log M)$). Hence the running time of the Pohlig-Hellman-Shanks algorithm is

$$O\left(\sum e_i(\log n + \sqrt{p_i}\log p_i)\right)$$

for the choice $M = \lceil\sqrt{p_i}\rceil$. □

The Pohlig-Hellman method allows to reduce the DLP in a group G to the same problem in the minimal nontrivial subgroups of G (these are exactly the subgroups of prime order). The algorithm is efficient if n has only small prime factors.

Definition 9.3. A number n is called *B-smooth* if all the prime factors p_i of n satisfy $p_i \leq B$. ○

If n is B-smooth, we have in the worst case that $p_i \approx B$ for all i, i.e., the number of factors is $\log n / \log B$, and the complexity is

$$O\left((\log n)^2 + \frac{B}{\log B}\log n\right) = O_s(B) ,$$

or

$$O\left((\log n)^2 + \sqrt{B}\log n\right) = O_s(\sqrt{B})$$

when the baby-step giant-step trade-off is used.

As mentioned, the linear GISP is equivalent to the DHP. In particular, the baby-step giant-step and Pohlig-Hellman algorithms can be used to compute discrete logarithms in groups. Their combination achieves the running time $\Theta_s(\sqrt{p})$ and matches Shoup's lower bound on the complexity of such algorithms. The Pohlig-Hellman-Shanks algorithm will play a central role in Chapters 10 and 11. Special variants of the DLP in the generic model are investigated in Section 9.4.

9.3.3 The Squaring GISP

Let us now consider the "squaring-GISP," i.e., the

$$(x_1^2)\text{-GISP}(n) .$$

Corollary 9.3 does not exclude this problem to be efficiently solvable unless n has multiple large prime factors. However, it is a priori not clear whether an efficient algorithm exists for this GISP. This problem will be studied in generality, for the "addition-and-multiplication-GISP," in Chapter 10. Solving the latter problem is equivalent to generically reducing the DLP to the DHP in the same group. Boneh and Lipton [10] (see also [9]) called this problem, for a prime $n = p$, the *black-box-field problem* (BBFP), and described subexponential *uniform* algorithms for solving it.

The following example, due to den Boer [8], shows that in the special case where n is a prime p such that $p-1$ is a power of 2, the lower bound is achieved by the running time of a simple algorithm.

Lemma 9.5. [8] *Assume that $n = p$ is a prime with $p-1 = 2^l$ for some integer l. Then there exists an algorithm that solves the (x_1^2)-GISP(n) by totally*

$$2(\log(p-1) - 1)$$

calls to one of the oracles (i.e., the "squaring oracle" and the σ-evaluation oracle).

Proof. Let $\sigma(x)$ be given. By calling the "squaring oracle" one can compute

$$b_0 = \sigma(x) \ , \quad b_1 = \sigma(x^2) \ , \ \ldots \ , \ b_{l-1} = \sigma(x^{2^{l-1}}) = \sigma(x^{(p-1)/2}) \ .$$

Let $x = c^w \pmod p$, where c is a fixed generator of $GF(p)^*$, and let further

$$w \equiv w_0 + 2w_1 + \cdots + 2^{l-1}w_{l-1} \pmod{p-1} \ .$$

First, w_0 can be determined because

$$b_{l-1} = \sigma(1) \quad \text{if} \quad w_0 = 0$$

and

$$b_{l-1} = \sigma(-1) \ (\neq \sigma(1)) \quad \text{if} \quad w_0 = 1 \ .$$

Furthermore

$$b_{l-2} = \sigma(x^{(p-1)/4}) = \sigma(c^{(w_0+2w_1)(p-1)/4}) = \sigma(c^{w_0(p-1)/4} \cdot c^{w_1(p-1)/2}) \ , \tag{9.6}$$

where

$$c^{w_0(p-1)/4} \not\equiv 0 \pmod{p} \ . \tag{9.7}$$

The last expression of (9.6) is equal to $\sigma(c^{w_0(p-1)/4})$ if $w_1 = 0$ and to $\sigma(-c^{w_0(p-1)/4})$ if $w_1 = 1$. These values are different because of (9.7), hence w_1 is determined.

Proceeding like this, w and $x = c^w$ can be obtained by

$$l - 1 = \log(p-1) - 1$$

calls to the "squaring oracle" and the same number of queries to the σ-evaluation oracle. □

9.4 Variations of the Linear GISP and the DLP

In this section we consider variants of, and special questions related to, the linear GISP or, alternatively, the DLP with respect to generic algorithms. In particular, we consider the short-exponent DLP (Section 9.4.1), the DLP under the assumption that the input group element is distributed according to some given distribution (Section 9.4.2), and the so-called oracle complexity of the DLP (Section 9.4.3). The latter is the number of yes/no-questions that must be asked to an infinitely-powerful oracle in order to obtain enough information for solving the DLP efficiently. The formalism we will use in this section is the one of the linear GISP or, equivalently, of generic algorithms in cyclic groups (see also Section 10.1.1).

9.4.1 The Short-Exponent DLP

A variation of the DLP of great interest is the short-exponent DLP, which is defined as follows.

Definition 9.4. Let G be a cyclic group of order n with generator g. Then the L-*short-exponent* DLP for groups of order n (L-SEDLP(n) for short) is, given g^x with $x \in [0, \ldots, L]$, to compute x. ○

A motivation for studying this problem is that it can be desirable to use the DH protocol with short exponents by efficiency reasons. It is then a natural and important question how much less secure this variant is. Secondly, it was shown in [93] that pseudorandom-number generators can be constructed whose security is based on the sole assumption that the SEDLP is hard.

Van Oorschot and Wiener [92] described the following method, which is a combination of the Pollard λ and Pohlig-Hellman algorithms, for solving the SEDLP. The idea of Pollard λ is as follows. Let $G = \langle g \rangle$ be a cyclic group. For the computation of the discrete logarithm x (of a group element g^x) lying in the interval $[A, B]$ of length $L := B - A$, two sequences of group elements, one starting with the upper limit B of the interval (the "trail of the tame kangaroo") and the other with the group element y of which the discrete logarithm should be computed (the "trail of the wild kangaroo"). The behavior of both sequences is given by $x_{i+1} = x_i \cdot g^{f(x_i)}$, where f is a "random-like" function taking integer values in a range R of mean m, where $m = \alpha \cdot L^{1/2}$ for some α depending on the tolerated failure probability. The starting point of the trail of the tame kangaroo (the sequence $x_0, x_1, \ldots$) is $x_0 = g^B$, and the group elements $x_0, x_1, \ldots, x_N$ are computed (for some fixed N). The trail of the wild kangaroo (the sequence $x'_0, x'_1, \ldots$) starts at $x'_0 = y$ and stops with x'_M if

$$\sum_{j=0}^{M-1} f(x'_j) > \sum_{i=0}^{N-1} f(x_i) + (B - A) ,$$

because the wild kangaroo has passed the tame kangaroo and escaped. Capture is indicated by $x'_m = x_N$ for some m. Then, the discrete logarithm of y is

$$x_N - f(x'_0) - \cdots - f(x'_{m-1}) .$$

Let now $y = g^x$, where x is known to be smaller than L for some L, and let $|G| = S \cdot T$, where S is the smooth part of the group order. By the Pohlig-Hellman decomposition, k can be computed such that $k \equiv x \pmod{S}$. Then $x = k + r \cdot S$ for some $r \le L/S$ or, equivalently, $y' = (g')^r$, where $y' := y \cdot g^{-k}$ and $g' := g^S$. The discrete logarithm r of

y' with respect to the base g' can now be computed in probabilistic time $O(\sqrt{L/S})$ using Pollard's λ method. In other words, the information obtained by the smooth part can be used to reduce the running time of the λ method by $\sqrt{S}$.

Corollary 9.7 shows that the van Oorschot-Wiener method is optimal with respect to generic algorithms in the case where all the prime factors smaller than the bound L on the discrete logarithm are of size polynomial in $\log n$. Note again that the SEDLP corresponds to the linear GISP (with restricted domain).

Theorem 9.6. *Let n be a positive integer with factorization $n = \prod p_i^{e_i}$. Assume that an algorithm is given that solves the L-SEDLP(n) in at most T steps. Then the probability α that the algorithm outputs the correct discrete logarithm is at most*

$$\alpha \leq \frac{(T+2)(T+1)+2}{2\lfloor (L+1)/\prod_{p_i<L} p_i^{e_i} \rfloor} \; .$$

Proof. Let $\sigma(1), \sigma(x)$ be the input to the algorithm, where x is uniformly distributed in $[0, L]$. Let p be the smallest prime factor of n with $p \geq L$. Assume that for all $p_i < L$, x modulo $p_i^{e_i}$ is given. (Of course, this extra information can only reduce the running time of the algorithm.) Hence the number of remaining possibilities for the value x modulo p is at least

$$L' := \left\lfloor \frac{L+1}{\prod_{p_i<L} p_i^{e_i}} \right\rfloor \; . \tag{9.8}$$

(Clearly, finding this value is equivalent to finding x.) Furthermore, x modulo p is uniformly distributed among the remaining possibilities, given that $x \in [0, L]$ is.

Assume now that $n = p^e$ holds. Given that the algorithm performs T steps and computes the representations $\sigma(P_i(x))$ of $T+2$ linear expressions in x, the number of x (modulo p) such that $P_i(x) = P_j(x)$ holds for some $i \neq j$ (and where $P_i \not\equiv P_j \pmod{p}$) is at most

$$N := \frac{(T+2)(T+1)}{2} \; .$$

Hence, and because random guessing among the remaining values is optimal given that the collision event $\mathcal{E}$ does not occur, the success probability of the algorithm is at most

$$\frac{N}{L'} + \left(1 - \frac{N}{L'}\right) \frac{1}{L'-N} = \frac{N+1}{L'} \; .$$

This concludes the proof. □

Corollary 9.7 follows from Theorem 9.6 in exactly the same way as Corollary 9.3 was derived from Theorem 9.1.

Corollary 9.7. *Every algorithm solving the* L-SEDLP *in groups of order* $n = \prod p_i^{e_i}$ *has expected running time of order*

$$\Omega\left(\sqrt{\frac{L}{\prod_{p_i<L} p_i^{e_i}}}\right) .$$

Remark. In the case where all the prime factors p_i of n smaller than L are of order $(\log n)^{O(1)}$, this is exactly the complexity achieved by the van Oorschot-Wiener algorithm. Note further that in the complementary case, in particular if there exist prime factors p of n only slightly smaller than L, then the bound of Corollary 9.7 is weak and far from being tight. The bound can be improved by applying the results concerning the GDDLP (see Section 9.4.2) to this situation.

The lower bound of Corollary 9.7 also holds for the more general situation where a discrete logarithm must be computed that lies in a fixed set of size L. The reason is that the SEDLP corresponds to the best case (in the point of view of someone willing to compute the discrete logarithm), because the moduli of the possible discrete logarithms with respect to the powers of the small prime factors of the group order have "maximal diversity." The other (worst-case) extreme would be that all the discrete logarithms in the possible subset are equivalent modulo the smooth part of n, in which case the efficiently-computable information about the discrete logarithm is clearly of no help.

Together with the above-mentioned result of [93], Corollary 9.7 implies that pseudorandom-number generators can be easily constructed with provable security against adversaries restricted to generic attacks.

9.4.2 Discrete Logarithms with General Distributions

A natural generalization of the problem studied in the previous section, the SEDLP and the "general-subset DLP," is the "general-distribution DLP."

Definition 9.5. Let G be a cyclic group with generator g and of order $|G| = n$. For a distribution P on G (or, equivalently, on $\mathbf{Z}_{|G|} = \mathbf{Z}_n$), the *$P$-general-distribution* DLP(n) (P-GDDLP(n) for short) is, given a random group element g^x distributed according to P, to compute x. ○

We study this problem for groups whose order is a prime p. More precisely, we derive a lower bound on the *expected* running time of any generic algorithm solving it. The expectation of the running time is, as in all other bounds derived in the generic model, taken not only over the coin tosses of the algorithm, but also over the input. Clearly, the worst-case (with respect to the input) expected (with respect to the coin tosses of the algorithm) running time would only depend on the size of the set $\{h \in G : P(h) > 0\}$, and is in particular the same as for the general DLP if $P(h) > 0$ holds for all $h \in G$.

The derived lower bound depends on the maximal probability occurring or, equivalently, on the min-entropy $H_\infty(P)$ of the distribution P.

Theorem 9.8. *Let p be a prime, and let P be a probability distribution on the set $\{0, 1, \ldots, p-1\}$ with min-entropy $H_\infty(P)$. Let a generic algorithm be given for groups of order p that performs at most T computation steps. Then the probability, for a given element $h \in G$ distributed according to P, that the algorithm outputs the correct discrete logarithm h is at most*

$$\alpha \leq \frac{(T+2)(T+1)+2}{2} \cdot 2^{-H_\infty(P)} \ .$$

Proof. The proof is analogous to the proof of Theorem 9.6, except that the maximal occurring probability is now $2^{-H_\infty(P)}$. □

Corollary 9.9. *An algorithm solving the P-GDDLP(p) must have expected running time (taken over the input and the coin tosses of the algorithm) of order*

$$T = \Omega\left(\sqrt{2}^{H_\infty(P)}\right) = \Omega\left((\max_y\{P(y)\})^{-1/2}\right) \ .$$

9.4.3 The Oracle Complexity of the DLP

In this section we consider the DLP when given an infinitely-powerful helping oracle answering arbitrary yes/no-questions correctly. We call such an oracle *universal*. The model was used in [67], where it was shown that a given number n can be factored in time polynomial in $\log n$ (with small failure probability) if $\varepsilon \log n$ queries can be made to the universal oracle for arbitrarily small $\varepsilon > 0$. (Clearly, a trivial solution uses $(1/2)\log n$ queries and asks for the binary expansion of the smallest prime factor of n.)

The notion of *oracle complexity* can be interesting in the context of an adversary having some partial knowledge about an RSA secret key or a discrete logarithm. Given an upper bound on the *quantity* of the information (e.g., measured in the number of physical bits), there are various ways of characterizing the *type* of information the opponent can have. In the worst case, this side information must be modeled in such a way that it can be chosen by the adversary, i.e., that the opponent can call a universal oracle a certain number of times.

In this context, the result of [67] concerning the integer-factoring problem is pessimistic: However small the ratio between the length of the side-information string and the length of the number n to be factored is, this information can be such that it allows efficient factorization of n. (Notice that the number of required oracle questions can be of size $O(\log\log n)$ only if the original problem, e.g., integer factoring, is solvable in polynomial time even without an oracle.)

We show an optimistic result for the DLP in the generic computation model. Namely, the number of possibly adaptively-chosen universal-oracle queries for computing a discrete logarithm efficiently must be roughly $\log(|G|/S)$, where S is the smooth part of the group order. This means that for an adversary limited to generic attacks, no short piece of side information about a discrete logarithm can exist that makes the discrete logarithm efficiently computable.

Theorem 9.10 gives, in the generic model, an upper bound on the success probability of a discrete-logarithm algorithm in terms of the number of queries to the universal oracle and the number of group operations. Notice that no assumption on the overall running time of the algorithm is necessary.

Theorem 9.10. *Let G be a group of order n, and let p be a prime factor of $|G|$ not of order $(\log|G|)^{O(1)}$ and with multiplicity $e \geq 1$. Assume that an algorithm performs at most T calls to the group-operation oracle and R queries to the universal oracle. For given g^x, where*

$$x \equiv x_0 + x_1 p + \cdots + x_{e-1} p^{e-1} \pmod{p^e} ,$$

and for fixed i*, the probability* α *that the algorithm successfully computes* x_i *is at most*

$$\alpha \leq 2^R \cdot \frac{(T+2)(T+1)+1}{2p} \ .$$

Proof. We can assume $n = p^e$ and $x \equiv x_i p^i \pmod{n}$. The oracle questions can be assumed without loss of generality to be deterministic, i.e., the oracle's answer to a yes/no-question leads to a partition of all the possibilities into two distinct subsets. Moreover, we can assume that all the queries to the universal oracle are performed at the beginning of the computation. The reason is that an algorithm making at most T queries to the group-operation oracle and R (adaptively-chosen) queries to the universal oracle can be transformed, by conditioning the oracle questions on later computation results and oracle answers, into an algorithm with the same number of oracle queries (both to the group-operation oracle and to the universal oracle), and for which the universal-oracle queries are all chosen and made initially. Notice that this new algorithm is generally much less efficient with respect to oracle-independent (i.e., "off-line") computations, but not with respect to group operations.

The value x_i is, after the R initial oracle queries, known to lie in one of 2^R sets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{2^R}$ with

$$\bigcup_{i=1}^{2^R} \mathcal{D}_i = \mathbf{Z}_p$$

and $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$ for $i \neq j$. The numbers $N_i := |\mathcal{D}_i|$ hence fulfill

$$\sum_{i=1}^{2^R} N_i = p \ .$$

Given that x_i is contained in $\mathcal{D}_i$, the probability that the algorithm answers correctly after T computation steps is at most

$$\left(\frac{(T+2)(T+1)}{2} + 1 \right) \cdot \frac{1}{N_i}$$

(by an argument similar to the one used in previous proofs, e.g., of Theorem 9.1). Hence the success probability, taken over the input, is at most

$$\sum_{i=1}^{2^R} \frac{N_i}{p} \left(\frac{(T+2)(T+1)}{2} + 1 \right) \cdot \frac{1}{N_i} \leq \frac{2^R((T+2)(T+1)+1)}{2p} \ ,$$

and this concludes the proof. □

Corollary 9.11 follows from Theorem 9.10 by precisely the same arguments as Corollary 10.7 follows from Theorem 10.6 in Section 10.3.2.

Corollary 9.11. *Let G be a group of order n, and let p be a prime factor of n. Assume that a generic algorithm computes, given a group element g^x, the i-th coefficient of the p-adic expansion of x modulo p^e in expected time T with the help of a universal oracle. Then the number R of oracle queries satisfies*

$$\mathrm{E}[R] \geq \left(1 - \frac{1}{\log p}\right) \log\left(\frac{p}{T^2}\right) - O(\log\log p) \ .$$

Corollary 9.12. *Let G be a cyclic group of order n, and let $\prod p_i^{e_i}$ be the prime factorization of n. Let furthermore*

$$S := \prod_{p_i = (\log n)^{O(1)}} p_i^{e_i}$$

be the smooth part of n. Assume that a generic algorithm is given that solves the DLP *in expected polynomial time in G, and which is allowed to make queries to a universal oracle. Then the expected number of questions to the oracle is lower bounded by*

$$\mathrm{E}[R] = \log\left(\frac{n}{S}\right) - o(\log n) \ .$$

Proof. Note first that for the different prime factors p of n, the distinct pieces of information to be computed (i.e., the coefficients of the p-adic representation of x) are statistically independent. In this case, the oracle complexity is additive. Let k be the number (with multiplicity) of prime factors of n which are not of order $(\log n)^{O(1)}$. Then we conclude from Corollary 9.11, for $T = (\log n)^{O(1)}$,

$$\begin{aligned}
\mathrm{E}[R] &\geq \sum_{p_i > (\log n)^{O(1)}} \left(e_i \left(1 - \frac{1}{\log p_i}\right) \cdot \log p_i - O(\log\log n)\right) \\
&= \sum_{p_i > (\log n)^{O(1)}} e_i \log p_i - k \cdot O(\log\log n) \\
&= \log(n/S) - k \cdot O(\log\log n) \ ,
\end{aligned}$$

where $p_i > (\log n)^{O(1)}$ stands for the fact that the prime factor p_i is not of order $(\log n)^{O(1)}$. Now, because

$$\left((\log n)^{1/o(1)}\right)^k = n \ ,$$

(the prime factors p_i are of superpolynomial size), we have

$$k = o(1) \cdot \log n / \log\log n \ ,$$

hence

$$\mathrm{E}[R] = \log\left(\frac{n}{S}\right) - o(\log n) \ .$$

□

We state as an open problem to determine the oracle complexity of the DLP in specific classes of groups, such as multiplicative groups of finite fields or elliptic curves.

9.5 Discussion

As pointed out, the model considered in this chapter is a generalization of the setting of generic algorithms for computation in groups. The special case of the linear GISP corresponds to the generic DLP, whereas solving the "addition-and-multiplication GISP" means reducing the DLP to the DHP, i.e., proving that breaking the DH protocol cannot be easier than computing discrete logarithms *in any group*. Corollary 9.3, applied to these special problems, implies on one hand Shoup's result that the Shanks-Pohlig-Hellman algorithm is an optimal generic algorithm for solving the DLP, and on the other hand that the DHP and the DLP *may* be computationally equivalent for groups whose order is *not* divisible by a large multiple prime factor, but that the two problems are certainly *not* equivalent in a generic sense for the remaining groups. These and other implications of Corollary 9.3, as well as further considerations in the model of generic algorithms in groups, are the subject of Chapters 10 and 11.

Special variants of the DLP have been studied with respect to the generic model. It was shown that the DLP with short exponents is hard (unless the range of the exponents is too small), and that the oracle complexity of the DLP in G is roughly $\log(|G|/S)$ in the generic model, where S is the smooth part of $|G|$. The latter implies that side information at an amount substantially smaller than $\log(|G|/S)$, and of *any type*, cannot help computing a discrete logarithm efficiently by generic algorithms.

Chapter 10

The Generic Relationship Between the DHP and the DLP

This chapter, and also Chapter 11, is concerned with different aspects of the security of the DH protocol. In this chapter, the relative hardness of the DHP with respect to the DLP is analyzed. Clearly, the ultimate goal of such an approach is to show that the DHP is computationally equivalent to the DLP in an "intrinsic sense," and for any specific group.

The model of generic algorithms, which is closely related to the information-theoretic model considered in Chapter 9, is a simple and natural formalization of an algorithm being *universal*. More precisely, a generic algorithm works for all groups in which group operations and tests of equality of group elements can be performed, and does not depend on any property of the particular representation of the group elements. The model of generic algorithms is introduced in Section 10.1.

A distinction is made between the cases of "uniform" and "nonuniform" generic algorithms. A uniform algorithm takes the group order as one input and then solves a particular problem in groups of this order, whereas a nonuniform algorithm for solving a certain problem is only required to solve the problem efficiently for every fixed group order, but may not be efficiently constructible when a group order is given as an input.

In Section 10.3, the entire picture is shown to be quite complete with respect to nonuniform reductions of the DLP to the DHP. If a plausible number-theoretic conjecture is true, then the DHP and the DLP are computationally equivalent exactly for groups whose order is not divisible by the square of a large prime. This characterization is based on two complementary results: An efficient generic reduction algorithm (Section 10.2) and a complexity lower bound in the generic model by "birthday-attack" arguments (Section 10.3.3).

Section 10.4 is concerned with, stronger, uniform reductions. It appears unlikely that a polynomial-time uniform reduction of the DLP to the DHP (for the class of groups for which a nonuniform reduction is not excluded by the arguments of Section 10.3.3) exists unless integer-factoring is easy. However, such a reduction can exist for groups whose orders are contained in a certain infinite subset of **N** (Section 10.4.1). A uniform *subexponential* reduction even exists for all groups, as was conjectured by Boneh and Lipton [10], based on a technique by Maurer [62], and is rigorously proved, using a result by Lenstra, Pila, and Pomerance [54], in Section 10.4.3. Finally, the case of *algebraic* algorithms (a class of algorithms lying between arbitrary and generic algorithms) solving the DHP is studied in Section 10.4.4.

Some results of this chapter have already been published in [78], [70], [71], [73], [72], [121]. The model of generic algorithms was introduced by Shoup [109]. The general technique for reducing the DLP to the DHP, in particular the use of elliptic curves for this reduction, is based on work by Maurer [62], extending an earlier result due to den Boer [8]. Boneh and Lipton [10] used Maurer's elliptic-curve technique for a heuristically-subexponential-time uniform reduction. The same authors independently also proposed to iterate algebraic DH algorithms.

10.1 Generic Algorithms in Groups

10.1.1 The Model

Let G be a group of order n. Clearly, an algorithm carrying out certain computations in this group should be able to perform the group operation, to compute inverses in G, and to decide whether two group elements are equal or not. The idea of a generic algorithm is that such an algorithm can *only* use these operations, but exploit no particular additional property of the group or, more precisely, of the representation of the group elements.

The generic model formalizes this idea by assuming that given a certain group structure, the representation of the group elements is chosen randomly among all possible representations, i.e., encodings with elements of some set S of size equal to the group order. Additionally, oracles are given for carrying out group operations and inversions. The model is closely related to the setting of the linear GISP of Section 9.3.2. We restrict our attention to *finite cyclic groups*, which are isomorphic to $\mathbf{Z}_n$ for some n, in the following.

Definition 10.1. [**109**] Let n be an integer, let S be a set of cardinality $|S| = n$, and let a bijective *representation function*

$$\sigma : \mathbf{Z}_n \longrightarrow S$$

be chosen at random from the set of all such functions according to the uniform distribution. A *generic algorithm* for groups of order n takes as input a list $(\sigma(x_1), \ldots, \sigma(x_l))$, where the x_i are elements of $\mathbf{Z}_n$. The algorithm is then allowed to make calls to oracles which compute the functions add, mapping $S \times S$ to S, and inv, mapping S to S, with

$$\mathrm{add}(\sigma(x), \sigma(y)) = \sigma(x+y) \quad \text{and} \quad \mathrm{inv}(\sigma(x)) = \sigma(-x).$$

○

As the model of the GISP, this is a purely probability-theoretic setting. Generic algorithms in groups must be based on testing equality of certain group elements computed with the help of the given oracles.

Notice that Definition 10.1 requires the representation of every group element to be unique (hence no equality-testing oracle is needed). In a more general model for which this representation is not supposed to be unique (but an equality-testing oracle is given instead), the complexity of certain problems is greater. For instance, the baby-step giant-step algorithm can only be applied if every group element has a unique or at least a canonical representation. The complexity of the DLP would in the general model be of order $\Theta_s(p)$ instead of $\Theta_s(\sqrt{p})$.

10.1.2 Generic Complexity, Reducibility, and Equivalence

We now define the notions of generic complexity, reducibility, and equivalence, which are natural measures of the *intrinsic* hardness of and relationship between computational problems in cyclic groups of certain orders. All these notions are defined both in a *uniform* and a *nonuniform* sense. In the nonuniform case, the mere *existence* of algorithms is crucial for fixed integers n, whereas in the uniform case such an algorithm must additionally be constructible efficiently for given n (which is in this case one of the inputs to the algorithm).

Definition 10.2. Let n be an integer. For a computational problem $\mathrm{A}(n)$ defined in groups of order n, we call the *(nonuniform) generic complexity* of $\mathrm{A}(n)$, denoted by

$$\mathrm{GC}(\mathrm{A}(n)) ,$$

the (expected) running time of the fastest (probabilistic) generic algorithm solving problem $\mathrm{A}(n)$. Let $\mathrm{B}(n)$ be a second problem. Then the *(nonuniform) generic reducibility* of $\mathrm{A}(n)$ to $\mathrm{B}(n)$,

$$\mathrm{GC}(\mathrm{A}(n) \downarrow \mathrm{B}(n)) ,$$

is the running time of the fastest generic algorithm solving $\mathrm{A}(n)$ and making queries to an oracle that solves problem $\mathrm{B}(n)$.

Let $\mathcal{M} \subseteq \mathbf{N}$ be an infinite set. We say that A is *efficiently (nonuniformly) generically reducible* to B with respect to $\mathcal{M}$,

$$\mathrm{A}(\mathcal{M}) \prec_g \mathrm{B}(\mathcal{M}) ,$$

if we have

$$\mathrm{GC}(\mathrm{A}(n) \downarrow \mathrm{B}(n)) = (\log n)^{O(1)}$$

for $n \in \mathcal{M}$. If $\mathrm{A}(\mathcal{M}) \prec_g \mathrm{B}(\mathcal{M})$ and $\mathrm{A}(\mathcal{M}) \succ_g \mathrm{B}(\mathcal{M})$ both hold, we say that A and B are *generically equivalent* (with respect to $\mathcal{M}$):

$$\mathrm{A}(\mathcal{M}) \sim_g \mathrm{B}(\mathcal{M}) . \tag{10.1}$$

Finally, we write

$$\mathrm{A}(\mathcal{M}) \not\sim_g \mathrm{B}(\mathcal{M})$$

not for the fact that (10.1) does not hold, but for the stronger condition that for every infinite set $\mathcal{M}' \subseteq \mathcal{M}$, $\mathrm{A}(\mathcal{M}') \sim_g \mathrm{B}(\mathcal{M}')$ is not true.

The *uniform* notions are defined as follows. For the problem $\mathrm{A}(n)$, we denote by

$$\mathrm{GC}_u(\mathrm{A}(\mathcal{M}))$$

the *uniform generic complexity* of A (with respect to $\mathcal{M}$), i.e., the running time of the fastest generic algorithm that takes $n \in \mathcal{M}$ as an input and solves problem A(n) (in groups of order n). The *uniform generic reducibility* of A to B with respect to $\mathcal{M}$,

$$\mathrm{GC}_u(\mathrm{A}(\mathcal{M}) \downarrow \mathrm{B}(\mathcal{M})) ,$$

is defined in analogy to the nonuniform notion. However, the natural restriction is made that the reduction algorithm only makes calls to the oracle that solves problem B(n) for the particular n which was the input to the reduction. The notions of *efficient uniform generic reducibility*,

$$\mathrm{A}(\mathcal{M}) \prec_{u,g} \mathrm{B}(\mathcal{M}) ,$$

and *uniform generic equivalence*,

$$\mathrm{A}(\mathcal{M}) \approx_g \mathrm{B}(\mathcal{M}) , \tag{10.2}$$

are defined in analogy to the nonuniform case. Here,

$$\mathrm{A}(\mathcal{M}) \not\approx_g \mathrm{B}(\mathcal{M})$$

means the complementary event of (10.2). ○

Remark. Somewhat abusing the notation, we will sometimes write that for instance "A$(n) \prec_g$ B(n) holds for $n \in \mathcal{M}$." Clearly, such a relation makes no sense for *fixed* n, and hence stands for the asymptotic statement A$(\mathcal{M}) \prec_g$ B$(\mathcal{M})$.

Shoup [109] has proved the following complexity results:

$$\begin{aligned} \mathrm{GC}(\mathrm{DLP}(n)) &= \mathrm{GC}(\mathrm{DHP}(n)) = \Omega\left(\sqrt{p}\right) \\ \mathrm{GC}(\mathrm{DHDP}(n)) &= \Omega\left(\sqrt{p'}\right) \end{aligned}$$

if p and p' are the largest and smallest prime factors of n, respectively. However, these lower bounds have no direct security implications for any *particular* group G.

On the other hand, an efficient generic *reduction* from the DLP to the DHP for instance proves the equivalence of the two problems, i.e., the *relative* hardness of the DHP with respect to the DLP, for *every particular group*. It is therefore an interesting problem to study the generic relationship even between problems with equal generic complexities. Motivated by this, we analyze the generic (complexity of and) relationship between a number of problems that arise in the context of the DH protocol or other cryptographic protocols.

10.2 A Generic Technique for Reducing the DLP to the DHP

In the rest of this chapter we study the generic relationship between the DHP and the DLP. It is clear that

$$\text{DHP} \prec_{u,g} \text{DLP}$$

holds: The DLP cannot be easier than the DHP. The interesting question however in the context of the DH protocol is whether this is true also in the inverse direction. Is it possible that breaking the DH protocol is substantially easier than solving the DLP ?

We describe a general technique for reducing the DLP to the DHP in a group G. More precisely, a class of generic algorithms is given that compute discrete logarithms in a cyclic group $G = \langle g \rangle$, and which are allowed to make queries to a DH *oracle* for this group.

Definition 10.3. A *Diffie-Hellman oracle* (DH oracle) for a group G with respect to a given generator g takes as inputs two elements $a, b \in G$ (where $a = g^u$ and $b = g^v$) and returns the element g^{uv}. ○

The DLP when given this oracle corresponds to the

$$(x_1 + x_2, x_1 \cdot x_2)\text{-GISP}$$

as described in Chapter 9. The presented technique uses certain additional structures, called *auxiliary groups*. In this section we use the notion of a "general" group G with generator g and of order $|G|$. Every algorithm that works in G is clearly a generic algorithm (as defined in Section 10.1.1) since nothing is assumed about the representation of the elements of G.

Assume that the prime factorization of $|G|$,

$$|G| = \prod_i p_i^{e_i} \ ,$$

is known. Let $a = g^s$ be a given group element for which we want to compute the discrete logarithm s using a DH oracle for G. It is sufficient to compute s modulo each prime factor of $|G|$ (or modulo the prime powers if $|G|$ contains multiple prime factors) and to combine these values by Chinese remaindering. Only large prime factors are relevant because the Pohlig-Hellman algorithm allows to compute s modulo powers of small prime factors of $|G|$. Hence we can restrict our attention to the problem of computing s modulo p for a large prime factor p of $|G|$. We assume that p is a *single* prime factor of $|G|$ (the case of $|G|$ having multiple large prime factors is discussed in Section 10.3.3). Let x be the element of $\mathbf{F}_p$ defined by

$$x \equiv s \pmod{p} .$$

In the following sections, the problem of computing x from the group element g^s is investigated.

10.2.1 Computing with Implicit Representations Using a DH Oracle

In this section we introduce the notion of implicit representations of elements of $\mathbf{F}_p$. Instead of a random string (as in the general generic computation model), the implicit representation of a field element is given by a certain group element of G. The problem of determining a field element *explicitly* when given an implicit representation (when given the ability to carry out the field operations "implicitly") is similar to Boneh and Lipton's [10] black-box-field problem.

Every element y of the field $\mathbf{F}_p$ can be interpreted as corresponding to a set of elements of G, namely those whose discrete logarithm is congruent to y modulo p. Every element of this set is then a representation of the field element y.

Definition 10.4. Let G be a cyclic group with a fixed generator g, and let p be a prime divisor of the group order $|G|$. Then, a group element $a = g^{y'}$ is called an *implicit representation* (with respect to G and g) of the element $y \in \mathbf{F}_p$ if $y \equiv y' \pmod{p}$. We write $y \leadsto a$. ○

Note that the implicit representation of a field element is not unique if $|G| \neq p$.

The following operations on elements of $\mathbf{F}_p$ can be performed efficiently on implicit representations of these elements (i.e., by operating in the group G), where the result is also in implicit form. Let y and z be elements of $\mathbf{F}_p$, with

$$y \leadsto a\ , \quad z \leadsto b\ .$$

Because of

$$y = z \iff a^{|G|/p} = b^{|G|/p}\ ,$$

equality of two implicitly-represented elements of $\mathbf{F}_p$ can be tested by $O(\log |G|)$ group operations. Note that the elements $a^{|G|/p}$ and $b^{|G|/p}$ are also implicit representations of y and z. More precisely, these are the only representations which are p-th roots of the neutral element in G. We call these representations *canonical*.

Furthermore we have

$$\begin{aligned} y+z &\rightsquigarrow a \cdot b \\ yz &\rightsquigarrow \mathrm{DH}_g(a,b) \\ -y &\rightsquigarrow a^{-1} \;=\; a^{|G|-1} \end{aligned}$$

(where DH_g stands for the DH function with respect to the generator g), and these implicitly-executed operations on $\mathbf{F}_p$-elements require a group operation in G, a call to the DH oracle, and $O(\log |G|)$ group operations, respectively.

In order to simplify the notation, we also introduce the notion of an e-th-*power*-DH oracle ($\mathrm{PDH}_{g,e}$ oracle) that computes an implicit representation of the e-th power of an implicitly-represented element. A possible implementation of a $\mathrm{PDH}_{g,e}$ oracle is to use a "square and multiply" algorithm for obtaining an implicit representation of y^e, denoted by $\mathrm{PDH}_{g,e}(a)$, by $O(\log e)$ calls to a normal DH oracle. In particular we can compute inverses of implicitly-represented elements because

$$y^{-1} \rightsquigarrow \mathrm{PDH}_{g,p-2}(a) \ .$$

We call addition, subtraction, multiplication, division, and equality testing in $\mathbf{F}_p$ *algebraic operations*. Any efficient computation in $\mathbf{F}_p$ can be performed equally efficiently on implicit representations whenever it makes use only of algebraic operations. Examples are the evaluation of a rational function, testing quadratic residuosity of y by testing equality of

$$(\mathrm{PDH}_{g,(p-1)/2}(a))^{|G|/p} \quad \text{and} \quad g^{|G|/p} \ ,$$

or the computation of square roots using an algorithm of Massey [56].

Finally, we extend the definition of implicit representations from elements of $\mathbf{F}_p$ to m-tuples over $\mathbf{F}_p$.

Definition 10.5. Let p and G be as above and let $a_i \in G$ and $y_i \in \mathbf{F}_p$ (for $i = 1, \ldots, m$). We say that $(a_1, \ldots, a_m)$ is an implicit representation of $(y_1, \ldots, y_m)$,

$$(y_1, \ldots, y_m) \rightsquigarrow (a_1, \ldots, a_m) \ ,$$

if $y_i \rightsquigarrow a_i$ for $1 \leq i \leq m$. ○

10.2.2 Auxiliary Groups

When given a DH oracle for G, the computation of x is shown to work efficiently if an auxiliary group H over $\mathbf{F}_p$ with certain properties exists. The basic idea is to embed the unknown x into an implicitly-represented element c of H and to compute the discrete logarithm of this element explicitly. We now define a first required property of the auxiliary group H.

Definition 10.6. A finite group H is *defined* (m, α)*-algebraically over* $\mathbf{F}_p$ if the elements of H can be represented as m'-tuples (for some $m' \leq m$) of elements of $\mathbf{F}_p$ such that the group operation in this representation can be carried out by at most α algebraic operations in $\mathbf{F}_p$. ○

We will need the following stronger property for auxiliary groups.

Definition 10.7. A group H is said to be *defined strongly-*(m, α)*-algebraically over* $\mathbf{F}_p$ if H is defined (m, α)-algebraically over $\mathbf{F}_p$ and if there exist two algorithms, EMBED and EXTRACT, with the following properties.

1. For all $(x, e) \in \mathbf{F}_p^2$ the EMBED algorithm with input (x, e) either outputs a group element c of H, or reports failure.
2. If the EMBED algorithm is repeated with the input (x, e) for fixed x and randomly chosen e until the algorithm does not fail, then the expected running time until an element $c \in H$ is computed is at most α algebraic operations in $\mathbf{F}_p$.
3. If the EMBED algorithm does not fail for the input (x, e), then
$$\text{EXTRACT}(\text{EMBED}(x, e), e) = x \ .$$
4. The EXTRACT algorithm runs in time at most α.

○

In the examples considered below, the EMBED algorithm computes a group element c that contains $x + e$ as a coordinate, and the EXTRACT procedure outputs this particular coordinate minus e.

In Section 10.2.4 we show that an Abelian group H which is defined strongly-algebraically over $\mathbf{F}_p$ can be used as a powerful tool for the reduction of the computation of discrete logarithms modulo p in G to breaking the DH protocol for G. As a preparation we show in Section 10.2.3 how a generator set of an the Abelian group H can be computed. This algorithm is necessary for the *uniform* results described in Section 10.4.

10.2.3 Finding the Structure and a Generator Set of a Finite Abelian Group

Let H be an (additively-written) finite Abelian group H of rank r and with B-smooth order $|H| = \prod_{i=1}^{l} q_i^{f_i}$. Then

$$H \cong \mathbf{Z}_{n_1} \times \cdots \times \mathbf{Z}_{n_r}$$

(such that n_{j+1} divides n_j for $j = 1, \ldots, r-1$), where the numbers r and $n_1, \ldots, n_r$ are not known a priori. For $j \geq 0$, suppose that we have already found n_i and points h_i with order n_i in the quotient group

$$H/\langle h_1, \ldots, h_{i-1} \rangle$$

for $i = 1, \ldots, j-1$. Let $n_1 = \prod q_i^{g_i}$, and let π_{j-1} denote the canonical projection to the quotient group $H/\langle h_1, \ldots, h_{i-1} \rangle$, i.e., $\pi_{j-1}(u)$ is the element of $H/\langle h_1, \ldots, h_{j-1} \rangle$ containing u. For the construction of h_j such that $\pi_{j-1}(h_j)$ has maximal order in $H/\langle h_1, \ldots, h_{j-1} \rangle$, we choose $\Theta(\log\log|H|)$ points h in H at random and compute

$$\mathrm{ord}_{H/\langle h_1, \ldots, h_{j-1} \rangle} \pi_{j-1}(h)$$

by testing equality of

$$\frac{n_1}{q_i^k} \pi_{j-1}(h) = \pi_{j-1}\left(\frac{n_1}{q_i^k} h\right)$$

(for $i = 1, \ldots, l$ and $k = g_i, g_i - 1, \ldots, 0$) with the neutral element

$$e_{H/\langle h_1, \ldots, h_{j-1} \rangle} \tag{10.3}$$

of the quotient group. Testing equality of $\pi_{j-1}(h')$ and the element (10.3) is equivalent to deciding if $h' \in \langle h_1, \ldots, h_{j-1} \rangle$, which is done by the generalized Pohlig-Hellman discrete-logarithm algorithm (see Section 10.2.4). This leads to an element h_j with maximal order in

$$H/\langle h_1, \ldots, h_{j-1} \rangle \ .$$

It is possible that the algorithm makes a mistake here, i.e., that the generated element does not have maximal order. Such an error occurs only with probability exponentially small in the number of trials and can be detected as follows. In the case where

$$\mathrm{ord}_{H/\langle h_1,\dots,h_{j-1}\rangle}\pi_{j-1}(h_j)$$

does not divide

$$\mathrm{ord}_{H/\langle h_1,\dots,h_{j-2}\rangle}\pi_{j-2}(h_{j-1}) ,$$

the process must be restarted because one of the preceding points has not had maximal order. The same holds if

$$j > \max\{f_i\} .$$

The latter is a bound for the rank r of H.

The algorithm stops if $\langle h_1,\dots,h_r\rangle = H$, that is

$$H/\langle h_1,\dots,h_r\rangle = \{e\} ,$$

and $\{h_1,\dots,h_r\}$ is then a generator set of H. Every element c of H has a unique representation

$$c = \sum_{j=1}^{r} k_j h_j$$

with $k_j \in \{0,\dots,n_j-1\}$ with respect to this set. The expected number of operations in H to determine the generator set is

$$O\left(r^2 \log\log|H| \cdot \frac{\sqrt{B^r}}{\log B}(\log|H|)^3\right)$$

(using the time-memory trade-off in the Pohlig-Hellman algorithm).

10.2.4 The Reduction Algorithm

Theorem 10.1. *Let $G = \langle g\rangle$ be a cyclic group and let p be a prime factor of $|G|$. Assume that a finite Abelian group exists with rank r which is defined strongly-(m,α)-algebraically over $\mathbf{F}_p$ and whose order is B-smooth. Given a* DH *oracle for $G = \langle g\rangle$ and a group element g^s of G, the unique field element $x \in \mathbf{F}_p$ with*

$$x \equiv s \pmod{p}$$

can be computed by

$$O\left(m^2 \frac{B^r}{r\log B}\log p \log|G| + m^2(\log p)^2 \alpha \log|G|\right)$$

group operations in G,

$$O(m^2(\log p)^2\alpha\log|G|)$$

calls to the DH *oracle for* G, *and*

$$O\left(\left(m^2(\log p)^2 + m\log p\,\frac{B^r}{r\log B}\right)\cdot\alpha\log p\right)$$

field operations in $\mathbf{F}_p$.

Proof. The group H has the property that when given an implicitly-represented field element $x \in \mathbf{F}_p$, then an implicitly-represented *group* element c of H (and an explicit element e of $\mathbf{F}_p$) can be found efficiently with the property that from the *explicit* representation of c (and from e), the EXTRACT algorithm leads to x. The reason is that because the EMBED procedure uses only algebraic operations, it works also on implicitly-represented inputs (where the output group element is also implicitly represented). This fact allows to reduce the computation of discrete logarithms in G (modulo p) to the same problem in the group H. More precisely, the field element x is computed from an implicit representation of x in four steps.

Step 1. Use the EMBED algorithm to obtain, when given an implicit representation of x and a random $e \in \mathbf{F}_p$, an implicit representation of a group element c of H.

Step 2. Compute the discrete logarithm of c in H (with respect to some generator set).

Step 3. Compute c explicitly.

Step 4. Use the EXTRACT algorithm to obtain x explicitly:

$$x = \text{EXTRACT}(c, e)\ .$$

We have to prove the stated complexity bounds for Step 2. The group H is Abelian of rank r, i.e., H is isomorphic to $\mathbf{Z}_{n_1} \times \cdots \times \mathbf{Z}_{n_r}$ for some $n_1, \ldots, n_r$ satisfying $\prod_{j=1}^{r} n_j = |H|$ and such that n_{j+1} divides n_j for $j = 1, \ldots, r-1$. Let $\{h_1, \ldots, h_r\}$ be a generator set of H, i.e., $|\langle h_j \rangle| = n_j$ holds and H is the internal product of the cyclic subgroups $\langle h_1 \rangle, \ldots, \langle h_r \rangle$:

$$H = \langle h_1 \rangle \times \cdots \times \langle h_r \rangle \ .$$

It was shown in Section 10.2.3 how such a set can be computed efficiently.

The element $c \in H$ has a unique representation

$$c = \sum_{j=1}^{r} k_j h_j \qquad 0 \leq k_j < n_j$$

(the group H is written additively). We address the problem of computing the coefficients k_j. This can be done by a generalization of the Pohlig-Hellman algorithm (see Section 9.3.2), applied to implicitly-represented group elements.

The following is repeated for every prime q dividing $|H|$. We describe the first and second iteration step of an algorithm that computes k_j modulo the highest power of q dividing n_j for all $j = 1, \ldots, r$. The algorithm uses v_j ($j = 1, \ldots, r$) as local variables (initialized by $v_j \leftarrow 0$).

For the first step, let α_1 be the number of generators h_j whose order contains the same number of factors q as n_1. In other words, $(n_1/q)h_j$ is different from the neutral element e of H exactly for $j = 1, \ldots, \alpha_1$. Because H is defined algebraically over $\mathbf{F}_p$, an implicit representation of

$$\frac{n_1}{q} c$$

can be efficiently computed from an implicit representation of c. For all $(t_1, \ldots, t_{\alpha_1}) \in \{0, \ldots, q-1\}^{\alpha_1}$, we compute (explicitly)

$$\frac{n_1}{q} t_1 h_1 + \cdots + \frac{n_1}{q} t_{\alpha_1} h_{\alpha_1} \ ,$$

transform the coordinates to an implicit representation, and test equality with $(n_1/q)c$. Equality indicates that the t_j are congruent to the coefficients k_j modulo q. We set $v_j \longleftarrow t_j$ for these t_j, and for $1 \leq j \leq \alpha_1$.

For the second step, let α_2 be the number of elements h_j whose order contains at most one factor q less than n_1, i.e.,

$$\frac{n_1}{q^2} h_j \neq e$$

holds exactly for $j = 1, \ldots, \alpha_2$. Implicit representations of the group elements

$$\frac{n_1}{q^2}(t_1 q + v_1)h_1 + \cdots + \frac{n_1}{q^2}(t_{\alpha_1} q + v_{\alpha_1})h_{\alpha_1} + \frac{n_1}{q^2} t_{\alpha_1+1} h_{\alpha_1+1} + \cdots + \frac{n_1}{q^2} t_{\alpha_2} h_{\alpha_2}$$

are computed for all $(t_1, \ldots, t_{\alpha_2}) \in \{0, \ldots, q-1\}^{\alpha_2}$ until equality with the implicitly-represented element

$$\frac{n_1}{q^2}\, c$$

holds. Then assign

$$v_j \longleftarrow t_j q + v_j \quad (j = 1, \ldots, \alpha_1) ,$$

$$v_j \longleftarrow t_j \quad (j = \alpha_1 + 1, \ldots, \alpha_2) .$$

After repetition of this process up to the maximal q-power q^g dividing n_1, the resulting v_j satisfy

$$\frac{n_1}{q^g}\, c = \sum_{j=1}^{r} \frac{n_1}{q^g}\, v_j h_j ,$$

or, equivalently, k_j is congruent to v_j modulo the highest power of q dividing $n_i = \operatorname{ord}(h_j)$ for $j = 1, \ldots, r$.

After running the algorithm for all primes q dividing $|H|$, one can compute the coefficients k_j modulo $\operatorname{ord}(h_j)$ by Chinese remaindering. The complexity of the algorithm is

$O((\log |H|)^2)$ operations in H with implicitly-represented elements,

$O\left(m \frac{B^r}{r \log B} \log |H| \log |G| + \alpha \log |G|\right)$ operations in G,

$O(\alpha \log |G|)$ calls to the DH oracle for G, and

$O\left(r(\log |H|)^2 + \log |H| \frac{B^r}{\log B}\right)$ explicit operations in H.

The first part of the number of group operations comes from the equality tests of implicitly-represented elements of H or, more precisely, transformations of implicit representations into the canonical form. Note that

$$|H| \leq p^m$$

holds because H is defined (m, α)-algebraically over $\mathbf{F}_p$. The implicit and explicit operations in H can be further reduced to operations and DH-oracle calls in G as well as operations in $\mathbf{F}_p$. This leads to the complexities stated in the theorem, and concludes the proof. □

The efficiency can be improved by Shank's time-memory trade-off. The (typically dominant) term B^r for instance can be reduced to $O_s(\sqrt{B^r})$ if memory space of size $\Omega_s(\sqrt{B^r})$ is available.

Corollary 10.2 is an immediate consequence of Theorem 10.1. Notice that the discrete logarithm modulo $|G|$ can be efficiently computed from the same quantity modulo $p_i^{e_i}$, for all i, by Chinese remaindering.

Corollary 10.2. *Let $G = \langle g \rangle$ be a cyclic group with order $|G| = \prod p_i^{e_i}$. Assume that every* multiple *prime factor of $|G|$ is of order $(\log |G|)^{O(1)}$, and that for every greater (hence single) prime factor p_i there exists a finite Abelian group H_i with rank $r = O(1)$, defined*

$$\textit{strongly-}((\log |G|)^{O(1)}, (\log |G|)^{O(1)})\textit{-algebraically over } \mathbf{F}_{p_i}$$

and whose order $|H_i|$ is $(\log |G|)^{O(1)}$-smooth. Then there exists an algorithm, making calls to a DH *oracle for G, that solves the* DLP *in G in time polynomial in* $\log |G|$.

The case where the group order $|G|$ contains *multiple large* prime factors will be considered in Section 10.3. In fact, no efficient (generic) reduction algorithm can exist in this case.

10.3 Nonuniform Reductions and a Completeness Result

This section deals with nonuniform reductions of the DLP to the DHP. We show that for "most" n, an efficient reduction algorithm exists for groups of order n. However, this does a priori not imply that this algorithm can be found efficiently when given n as an input. An important consequence of the nonuniform generic equivalence is that for every fixed n, the two problems are roughly equally hard in all groups of order n: Whenever the DLP is hard, the DH protocol is secure. (The conclusion however cannot be made that if no efficient *uniform* algorithm solves the DLP, then no efficient uniform solution of the DHP can exist. This implication must be proved by a uniform reduction, as considered in Section 10.4.)

The completeness result shown in this section roughly states that the DHP and DLP are computationally equivalent exactly for groups whose order does *not* contain large prime factors with multiplicity greater than one.

10.3.1 Elliptic and Hyperelliptic Curves as Auxiliary Groups

Maurer [62] (see also [70] and [121]) proposed a generalization of Den Boer's [8] technique (see Lemma 9.5 and the proof thereof), following the lines of Lenstra's [53] improvement of Pollard's $(p-1)$-factoring algorithm [97], by using elliptic curves. The use of Jacobians of hyperelliptic curves was investigated in [121].

First of all, it is clear that an elliptic curve E over the field $\mathbf{F}_p$ is defined strongly-$(2, O((\log p)^2))$-algebraically over $\mathbf{F}_p$. The points of E can be represented as pairs of $\mathbf{F}_p$-elements, and the group operation can be executed in this representation by a constant number of additions, multiplications, divisions, and equality tests in $\mathbf{F}_p$ (see Section 8.1.4). (Note that no such representation a priori exists for the point at infinity. However, because this point can only result from a division by zero in the underlying field, which is easily detectable, this fact can be ignored in the following.)

We describe the EMBED algorithm. Let $x, e \in \mathbf{F}_p$ be given. First the expression $D = (x+e)^3 + A(x+e) + B$ is computed and its quadratic residuosity is tested. If D is not a quadratic residue, the algorithm reports failure (and a new value for e is chosen). If D is a quadratic residue, then a square root y of D is computed by an algorithm due to Massey [56] (see Theorem 11.5). The EMBED algorithm outputs $c = (x+e, y)$. The necessary executions of the EMBED algorithm require $O((\log p)^2)$ algebraic operations in $\mathbf{F}_p$.

Together with the fact that for every number

$$d \in [p - 2\sqrt{p} + 1, p + 2\sqrt{p} + 1] \ ,$$

there exists a cyclic elliptic curve over $\mathbf{F}_p$ of order d (see also Section 8.1.4), we get the following result.

Definition 10.8. For a number n, let $\nu(n)$ be the minimum, taken over all d in the interval $[n-2\sqrt{n}+1, n+2\sqrt{n}+1]$, of the largest prime factor of d. ○

Theorem 10.3. *For all numbers $n = \prod p_i^{e_i}$ we have*

$$\mathrm{GC}(\mathrm{DLP}(n) \downarrow \mathrm{DHP}(n)) = O_s\left(\max_{i\,:\,e_i=1}\left\{\sqrt{\nu(p_i)}\right\} + \max_{i\,:\,e_i>1}\left\{\sqrt{p_i}\right\}\right) .$$

Proof. Let p_i be a *single* prime factor of n. According to the definition of $\nu(n)$ there exists a cyclic elliptic curve over $\mathbf{F}_{p_i}$ with $\nu(p_i)$-smooth order. Theorem 10.1, together with the time-memory trade-off, states that discrete logarithms (modulo p_i) can be computed in time

$$\sqrt{\nu(p_i)} \cdot (\log n)^{O(1)}$$

in groups of order n.

If p_i is a multiple prime factor of n, then discrete logarithms modulo $p_i^{e_i}$ can clearly be obtained in time

$$\sqrt{p_i} \cdot (\log n)^{O(1)}$$

by the Pohlig-Hellman algorithm. The statement now follows from the Chinese remainder theorem (8.3). □

Unfortunately, it is not clear whether the running time of Theorem 10.3 is always polynomial in $(\log n)^{O(1)}$ if all the multiple prime factors of n are of this order. The reason is that very little is known about $\nu(p)$ for a given prime p, i.e., the existence of smooth numbers in the interval of interest. However, it is known [24] that for every fixed u,

$$\psi(n, n^{1/u})/n = u^{-(1+o(u))u} , \qquad (10.4)$$

where $\psi(n, y)$ denotes the number of y-smooth integers in the interval $[0, \ldots, n]$. This fact suggests that $\nu(n)$ is polynomial in $\log n$.

Smoothness Assumption 1. $\nu(n)$ is of order $(\log n)^{O(1)}$.

Corollary 10.4. *If the smoothness assumption 1 is true, then*

$$\mathrm{GC}(\mathrm{DLP}(n) \downarrow \mathrm{DHP}(n)) = (\log n)^{O(1)}$$

holds for all n with the property that every multiple *prime factor p_i of n is of order $p_i = (\log n)^{O(1)}$.*

Corollary 10.5. *Let n be such that all its multiple prime factors p_i are of order $p_i = (\log n)^{O(1)}$. If for some group G of order n the* DLP *is hard, i.e., not solvable in polynomial time, then the same holds for the* DHP *in G, i.e., the* DH *protocol cannot be broken in polynomial time.*

In [121] it was shown that Jacobians of hyperelliptic curves (see for example [48]) can also be used as auxiliary groups in the reduction technique of Section 10.2. Jacobians of genus 2 for instance have the advantage over elliptic curves that their order varies in a larger interval, namely of type $[n - \Theta(n^{3/4}), n + \Theta(n^{3/4})]$. In [1], the first polynomial-time primality prover, based on Jacobians of genus 2 instead of elliptic curves, was presented. However, the results about the distribution of the orders of Jacobians that are proved in [1] are not sufficient to obtain a rigorously-provable polynomial-time nonuniform generic reduction of the DLP to the DHP. The reason is that the existence of Jacobians with *prime* order is analyzed, whereas Jacobians with *smooth* order are required for our purpose.

10.3.2 Lower Bounds I: The Number of Oracle Queries

A (generic) reduction of a computational problem $\mathrm{A}(n)$ to another problem $\mathrm{B}(n)$ is stronger (leads to better bounds on the running times) if the reduction algorithm needs less queries to the oracle solving $\mathrm{B}(n)$. Sometimes, one such oracle call can be sufficient, as the example of reducing the DHP to the DLP shows. For the inverse direction however, Theorem 10.6 gives a lower bound on the number of oracle calls, and implies in particular that no reduction using only one oracle query (or a constant number of queries) can exist. The proof of Theorem 10.6 is closely related to the proof of Theorem 9.1.

Theorem 10.6. *Let n be a positive integer and let p be a prime factor of n. For any (possibly probabilistic) generic reduction algorithm of the* DLP *to the* DHP *that works for groups of order n, runs in time at most T, and computes the correct discrete logarithm with probability α, the expected number of calls to the* DH *oracle is at least*

$$\alpha \log\left(\frac{p}{T^2}\right) .$$

Proof. Let $n = p^e s$ with $\gcd(s, p) = 1$. The generic algorithm must compute discrete logarithms, i.e., it takes as inputs $\sigma(1)$ and $\sigma(x)$ and should output x. Since we are proving a lower bound, and because additional knowledge can only reduce the running time, we can assume that discrete logarithms modulo s are easy to compute. Hence in what follows, let without loss generality $n = p^e$. The DH oracle, which can be used in addition to the oracles for addition and inversion, is an oracle for multiplication, i.e.,

$$\mathrm{DH}(\sigma(y), \sigma(z)) = \sigma(y \cdot z) \ .$$

Let the generic algorithm make A calls to the addition and inversion oracles and M calls to the multiplication oracle during a particular execution. Hence we have $A + M \leq T$ if T is an upper bound on the number of steps the algorithm performs. The algorithm can interact with the oracles in the following way. By calls to the oracles, $\sigma(P_i(x))$ can be computed for (without loss of generality pairwise distinct) polynomial expressions $P_i(X)$, $i = 1, \ldots, A + M + 2$, with

$$P_1(X) = 1 \ , \ \ P_2(X) = X \ ,$$

and where, for all $i \geq 3$, $P_i(X)$ is equal to either

$$P_k(X) + P_l(X) \ , \ \ -P_k(X) \ , \ \text{ or } \ P_k(X) \cdot P_l(X)$$

for some $k, l < i$. All the additional computations the algorithm performs are statistically independent of x and σ (given the oracle queries and answers), and do not influence the information-theoretic arguments.

The crucial observation is that, unless $P_i(x) = P_j(x)$ for some $i \neq j$, the algorithm sees completely random elements of S and cannot learn anything about x other than the fact that $P_i(x) \neq P_j(x)$ holds for all $i \neq j$. In other words, if $\mathcal{E} \subseteq \mathbf{Z}_n$ is the event that $P_i(x) = P_j(x)$ for some $i \neq j$, and if $\overline{\mathcal{E}(\delta)}$ is the complementary event, then, we have

$$I(x \, ; \, [\sigma(P_i(x))]_{i=1,\ldots,A+M+2} \,|\, \overline{\mathcal{E}(\delta)}) = 0 \ . \tag{10.5}$$

The random variable x is statistically independent of

$$[\sigma(P_1(x)), \ldots, \sigma(P_{A+M+2}(x))] \ ,$$

given the event $\overline{\mathcal{E}(\delta)}$, because σ is chosen randomly among all possible encodings.

Lemma 9.2 implies that

$$\mathrm{P}[\mathcal{E}] \leq \frac{2^M (A + M + 2)(A + M + 1)}{2p} \ , \tag{10.6}$$

i.e., that the probability of the event $\mathcal{E}$ is upper bounded by D/p, where

$$D := 2^M (A + M + 2)(A + M + 1)/2 \ .$$

The number D is equal to the maximal possible degree of the polynomials P_i, namely 2^M, multiplied with the number $(A + M + 2)(A + M + 1)/2$ of two-element sets $\{i, j\} \subseteq \{1, \ldots, A + M + 2\}$.

We conclude from (10.5) and (10.6) that the probability that the algorithm answers correctly is upper bounded by

$$\mathrm{P}[\mathcal{E}] + \mathrm{P}[\overline{\mathcal{E}(\delta)}] \cdot \frac{1}{n \cdot \mathrm{P}[\overline{\mathcal{E}(\delta)}]} \leq \mathrm{P}[\mathcal{E}] + \frac{1}{p^e} \leq \frac{D+1}{p} \tag{10.7}$$

because x is a random element of $\mathbf{Z}_n$. Note that

$$n \cdot \mathrm{P}[\overline{\mathcal{E}(\delta)}]$$

is the number of elements x of $\mathbf{Z}_{p^e}$ compatible with the event $\overline{\mathcal{E}(\delta)}$. The best strategy of the algorithm, given $\overline{\mathcal{E}(\delta)}$, is to output one of these elements at random. From the assumption in the theorem, and from (10.7), we conclude

$$\mathrm{E}\left[\min\left\{\frac{D+1}{p}, 1\right\}\right] \geq \alpha \ ,$$

where the expectation is taken over the executions of the probabilistic algorithm and over the random choices of σ and x. In the worst case, we have $(D + 1)/p = 1$ with probability α, hence

$$M \geq \log\left(\frac{2p-2}{(T+2)(T+1)}\right)$$

with probability α, which implies that

$$\mathrm{E}[M] \geq \alpha \cdot \log\left(\frac{p}{T^2}\right) \ .$$

□

By roughly the same trick as for Corollary 9.3, we obtain the following statement connecting the number of required oracle calls with the *expected* (instead of worst-case) running time of the algorithm: From the probabilistic algorithm with expected running time T, a new algorithm is constructed by stopping the execution after kT steps for some constant k. This algorithm has success probability at least $1 - 1/k$ and worst-case running time kT. Then, Theorem 10.6 is applied to the new algorithm.

Corollary 10.7. *Let n be a positive integer, and let p be the largest prime factor of n. Assume that a generic reduction of the* DLP(n) *to the* DHP(n) *exists that works in expected time T. Then the number M of* DH *oracle queries satisfies, for all $k > 1$,*

$$\mathrm{E}[M] \geq \left(1 - \frac{1}{k}\right) \cdot \log\left(\frac{p}{k^2 T^2}\right) .$$

The choice $k = \log p$ leads to

$$\mathrm{E}[M] = (1 - o(1)) \log\left(\frac{p}{T^2}\right) - O(\log\log p) .$$

Remark. The algorithm of Lemma 9.5, solving the squaring GISP, shows that the bound of Theorem 10.6 is tight: This algorithm computes a discrete logarithm in a group of order $|G| = p$ by using only $\log(p-1) - 1$ calls to the DH oracle if $p - 1$ is a power of 2.

The result of Corollary 10.7 is pessimistic in the following sense. Assume that one can prove for a particular class of groups that discrete logarithms cannot be computed faster than in time T_{DL}. Then the best result one can hope for by using a general-purpose reduction is that the DHP cannot be solved faster than in time roughly

$$T_{DH} = T_{DL} / \log p$$

in the same class.

10.3.3 Lower Bounds II: Multiple Prime Factors

The problem of a generic reduction of the DLP to the DHP for groups G whose order contains a multiple large prime factor q already appeared implicitly in [8] and explicitly in [62]. It is not difficult to see that the reduction is also possible in this case if the DH oracle can be used to either efficiently construct a DH oracle for certain subgroups of $G = \langle g \rangle$ such as $\langle g^q \rangle$, or to compute q-th roots in G. Shoup proved in [109] that the first problem cannot be solved with generic algorithms, and Theorem 11.6 below states that the same is true for the second problem.

However, at first sight it appears plausible that there exist different ways of constructing generic reductions for such groups, for example by using the general reduction technique of Section 10.2 using auxiliary groups defined directly over the ring $\mathbf{Z}_{q^e}$ and not over the field $\mathbf{F}_q$. But, perhaps somewhat surprisingly, it can be shown that no efficient generic reduction exists for such groups. In other words, it is impossible to prove the computational equivalence of the DHP and DLP for all groups when no assumption on the representation of the group elements is made. Theorem 10.8 is an immediate consequence of Theorem 9.1.

Theorem 10.8. *Let n be a positive integer and let q be a multiple prime factor of n. For any (possibly probabilistic) generic reduction algorithm of the* DLP *to the* DHP *that works for groups of order n, runs in time at most T, and outputs the correct discrete logarithm with probability α, we have*

$$T \geq \sqrt{2\alpha q} - 2 \; .$$

Proof. From Theorem 9.1 (replacing T by $T+1$, because $\sigma(1)$ is given as an input here), we can directly conclude that

$$\alpha \leq \frac{(T+2)(T+1)+2}{2q} \; . \tag{10.8}$$

(Clearly, the fact that no σ-evaluation oracle is given here cannot decrease the running time.) Inequality (10.8) implies the statement of the theorem. □

By using the same argument as needed for the proof of Corollary 9.3 we get the following lower bound on the expected running time.

Corollary 10.9. *Let n be a positive integer and let q be the largest multiple prime factor of n. For any generic reduction algorithm of the* DLP *to the* DHP *for cyclic groups of order n, the expected running time T of the algorithm is lower bounded by*

$$T \geq \frac{\sqrt{q}}{2} - 1 \; .$$

10.3.4 The Completeness Result

From Corollaries 10.4 and 10.9 we obtain the following complete characterization of the existence of efficient nonuniform reductions of the DLP to the DHP.

Theorem 10.10. *For numbers n whose largest multiple prime factor is of order* $(\log n)^{O(1)}$*, we have under the smoothness assumption 1 that*

$$\mathrm{DHP}(n) \sim_g \mathrm{DLP}(n) \ .$$

For the remaining n, however,

$$\mathrm{DHP}(n) \not\sim_g \mathrm{DLP}(n)$$

holds.

Despite the fact that many questions in the context of the relationship between the DHP and the DLP remain open, Theorem 10.10 shows that with respect to *nonuniform generic* reduction algorithms, the picture is (provided that a plausible yet unproven number-theoretic conjecture is true) quite complete: The problems are equivalent if and only if the order of the underlying group contains no large prime factors with multiplicity greater than one.

10.4 Uniform Reductions

In contrast to the previous section, in which the existence of nonuniform reduction algorithms from the DLP to the DHP was (almost) completely characterized, this section deals with *uniform* generic reductions for all groups or for certain classes of groups or, more precisely, groups whose orders belong to a certain unbounded subset of $\mathbf{N}$.

Again, *uniform* in this context means that the group order n is also an input to the reduction algorithm. Informally but intuitively spoken, a uniform reduction is a nonuniform reduction with the additional property that for given n, the corresponding reduction algorithm (the *existence* of which is guaranteed) can be *found* efficiently. More specifically, in connection with the reduction technique of Section 10.2, this means that for a given prime p, a suitable auxiliary group H_p over $\mathbf{F}_p$ with smooth order can be constructed efficiently.

Clearly, a uniform reduction of the DLP to the DHP for groups with orders in $\mathcal{M}$ leads to a stronger result. For instance, from the fact that no uniform discrete-logarithm algorithm exists for groups with orders in $\mathcal{M}$, one can conclude that the same holds for the DHP, namely, that the DH protocol cannot be broken by a uniform algorithm. If only the nonuniform equivalence is proven between two problems, nothing can be concluded from the same assumption.

Section 10.4.1 deals with polynomial-time uniform generic reductions working for a certain infinite set $\mathcal{M} \subset \mathbf{N}$, whereas in Section 10.4.3 a subexponential-time reduction is given for all orders (not divisible by the square of a large prime factor). Here, previous heuristic results by Boneh and Lipton [10] are described as well as a new, less efficient yet rigorously-analyzed reduction based on novel results on integer-factoring using hyperelliptic curves. In Section 10.4.2, a method is described that allows to efficiently construct DH groups whose orders lie in a set $\mathcal{M}$ with

$$\mathrm{DHP}(\mathcal{M}) \approx_g \mathrm{DLP}(\mathcal{M}) .$$

The assumption that so-called *algebraic* algorithms exist for solving the DHP in certain classes of groups, such as the multiplicative groups of finite fields or elliptic curves, leads to even stronger generic uniform equivalence results for general groups (Section 10.4.4).

10.4.1 Polynomial-Time Equivalence

For a class $\mathcal{M} \subseteq \mathbf{N}$ of group orders (namely, those for which the large prime factors are single and have some additional properties to be specified) we show that the DHP and DLP are equivalent in a generic and uniform sense. This is achieved by using subgroups of the multiplicative groups of extension fields of $\mathbf{F}_p$ as the auxiliary groups in the reduction technique of Section 10.2. Clearly, and in contrast to an elliptic curve with smooth order, these groups can be found efficiently.

Let H be a subgroup of $\mathbf{F}_{p^n}^*$. We derive conditions under which such subgroups are defined strongly-algebraically over $\mathbf{F}_p$. The group operation of H is a multiplication in $\mathbf{F}_{p^n}^*$ and requires, in a normal basis representation, $O(n^3)$ multiplications in $\mathbf{F}_p$. We conclude that every subgroup of $\mathbf{F}_{p^n}^*$ (for n polynomial in $\log p$) is defined $(n, (\log p)^{O(1)})$-algebraically over $\mathbf{F}_p$. For all n, $\mathbf{F}_{p^n}^*$ is a cyclic group. This implies that a subgroup of $\mathbf{F}_{p^n}^*$ is uniquely determined by its order $|H|$ or, more precisely, for every divisor d of $|H|$ there exists exactly *one* subgroup of $\mathbf{F}_{p^n}^*$ with $|H| = d$. Furthermore, all these subgroups are cyclic.

The next theorem states conditions on n and $|H|$ under which H is defined *strongly*-algebraically over $\mathbf{F}_p$.

Theorem 10.11. *Let H be the subgroup of $\mathbf{F}_{p^n}^*$ of order $|H|$. Then H is defined*

$$\textit{strongly-}((\log p)^{O(1)}, (\log p)^{O(1)})\textit{-algebraically over } \mathbf{F}_p$$

if one of the following two conditions is satisfied.

Condition 1. The number n is of order $(\log p)^{O(1)}$, and there exists a divisor $k < n$ of n such that

$$|H| = \frac{p^n - 1}{p^k - 1} = p^{n-k} + p^{n-2k} + \cdots + p^k + 1$$

holds.

Condition 2. The number n is of order $n = O(1)$, and there exists a non-constant polynomial $f(x)$ (with integer coefficients) dividing $x^n - 1$ such that

$$|H| = f(p)$$

holds.

Remark. An alternative formulation of Condition 2 is that $|H|$ is a multiple of $\Phi_n(p)$ for some $n = O(1)$, where Φ_n stands for the n-th cyclotomic polynomial. The cyclotomic polynomials are the irreducible factors of the polynomials $x^N - 1$ over the ring $\mathbf{Z}$ of integers (see for example [55]). More precisely, we have

$$x^N - 1 = \prod_{d|N} \Phi_d(x) \ ,$$

and the polynomials Φ_d are irreducible over $\mathbf{Z}$. The degree of $\Phi_n(x)$ is $\varphi(N)$, where φ is Euler's totient function. Because the cyclotomic polynomials are irreducible over $\mathbf{Z}$, any (non-constant) polynomial $f(x)$ (that divides $x^N - 1$) must be a multiple of at least one cyclotomic polynomial $\Phi_n(x)$. Examples of cyclotomic polynomials are

$$\begin{aligned} \Phi_6(p) &= p^2 - p + 1 \ , \\ \Phi_8(p) &= p^4 + 1 \ , \\ \Phi_9(p) &= p^6 + p^3 + 1 \ . \end{aligned}$$

The alternating sums

$$p^{2l} - p^{2l-1} + - \cdots - p + 1$$

also satisfy Condition 2 for $l = O(1)$.

Proof of Theorem 10.11. We show that if one of the conditions is satisfied there exists an EMBED algorithm that takes as input two elements x and e of $\mathbf{F}_p$ and computes coordinates $\beta_1, \ldots, \beta_{n-1}$ (in the normal basis representation) in $\mathbf{F}_p$ by a polynomial number of algebraic operations such that

$$\beta = (x + e, \beta_1, \ldots, \beta_{n-1}) \in H \ .$$

One possibility of designing the EMBED algorithm is to express membership of an element $\beta = (\beta_0, \ldots, \beta_{n-1})$ to the subgroup H by an equation (or a system of equations) in the coordinates. Then, the element $x + e$ can be assigned to one of the coordinates, say β_0, and the equation is solved for the remaining coordinates (by using only algebraic operations in the field $\mathbf{F}_p$).

For an element β of $\mathbf{F}_{p^n}^*$, we have

$$\beta \in H \iff \beta^{|H|} = 1 \ . \tag{10.9}$$

Clearly, the equation on the right-hand side of (10.9) corresponds to a set of polynomial equations (with coefficients in $\mathbf{F}_p$) in the coordinates β_i.

We will show that if the first condition is satisfied, then it is sufficient to solve one univariate polynomial equation over a subfield $\mathbf{F}_{p^k}$ of $\mathbf{F}_{p^n}$ for finding such a β, and that this can be reduced to a polynomial number of algebraic operations in the field $\mathbf{F}_p$. The situation when the second condition is satisfied is more difficult. Here, a *system of multivariate* polynomial equations over $\mathbf{F}_p$ has to be solved by algebraic operations.

Proof that Condition 1 is sufficient. The EMBED algorithm works as follows in this situation. Let $x, e \in \mathbf{F}_p$ be given. For $l := n/k$ let $\{\alpha_0, \ldots, \alpha_{k-1}\}$ and $\{\alpha'_0, \ldots, \alpha'_{l-1}\}$ be normal bases of $\mathbf{F}_{p^k}$ over $\mathbf{F}_p$ and of $\mathbf{F}_{p^n}$ over $\mathbf{F}_{p^k}$, respectively. For an element $\beta = (\beta_0, \ldots, \beta_{l-1}) \in \mathbf{F}_{p^n}$ (with $\beta_i \in \mathbf{F}_{p^k}$), we have that $\beta \in H$ is equivalent to

$$\beta^{(p^n-1)/(p^k-1)} = 1 \ . \tag{10.10}$$

Equation (10.10) is equivalent to

$$\left(\sum_{i=0}^{l-1} \beta_i \alpha'_i \right)^{p^{(l-1)k} + p^{(l-2)k} + \cdots + p^k + 1} = 1 \ . \tag{10.11}$$

Now, we have $(\beta_i)^{p^{jk}} = 1$ (because $\beta_i \in \mathbf{F}_{p^k}$) and $(\alpha'_i)^{p^{jk}} = \alpha'_{i+j}$ (where the index is reduced modulo l) by the definition of the normal basis. Hence (10.11) is equivalent to

$$\left(\sum_{i=0}^{l-1} \beta_i \alpha'_{i+l-1}\right) \cdot \left(\sum_{i=0}^{l-1} \beta_i \alpha'_{i+l-2}\right) \cdots \left(\sum_{i=0}^{l-1} \beta_i \alpha'_i\right) = 1 \tag{10.12}$$

(where the indices are reduced modulo l).

Because of

$$(\beta^{(p^n-1)/(p^k-1)})^{p^k-1} = \beta^{p^n-1} = 1 \ ,$$

$\beta^{(p^n-1)/(p^k-1)}$ is an element of the subfield $\mathbf{F}_{p^k}$ of $\mathbf{F}_{p^n}$. Such elements are easy to characterize in terms of their coordinates. An element $(\gamma_0, \ldots, \gamma_{l-1})$ is an element of $\mathbf{F}_{p^k} \subset \mathbf{F}_{p^n}$ if and only if $\gamma_0 = \gamma_1 = \cdots = \gamma_{l-1}$. The reason for this fact is that $\alpha'_0 + \alpha'_1 + \cdots + \alpha'_{l-1}$ (the *trace* $\mathrm{Tr}(\alpha'_0)$ of α'_0) is also an element of $\mathbf{F}_{p^k}$. Because both $\beta^{(p^n-1)/(p^k-1)}$ and 1 are elements of $\mathbf{F}_{p^k}$, they are equal if and only if their first coordinates are equal. The equation coming from (10.12), restricted to the first coordinate, is equivalent to

$$g(\beta_0, \ldots, \beta_{l-1}) = 1/\mathrm{Tr}(\alpha'_0) \tag{10.13}$$

for some l-degree polynomial g with $\mathbf{F}_{p^k}$-coefficients.

The construction of a group element β of H with the desired property now works as follows. Let the first coordinate $\beta_{0,0}$ of β_0 (note that β_i corresponds to a k-tuple $(\beta_{i,0}, \ldots, \beta_{i,k-1})$ of $\mathbf{F}_p$-elements with respect to the normal basis $\{\alpha_0, \ldots, \alpha_{k-1}\}$) be equal to $x + e$. Choose the coefficients $\beta_{0,1}, \ldots, \beta_{0,k-1}$ and the coefficients $\beta_1, \ldots, \beta_{l-2}$ randomly in $\mathbf{F}_p$ and in $\mathbf{F}_{p^k}$, respectively. Then (10.13) is equivalent to a polynomial equation for β_{l-1} with coefficients in $\mathbf{F}_{p^k}$ which can be solved by the Cantor-Zassenhaus algorithm (see Section 8.1.3).

The computation of polynomial gcd's, and thus the execution of the entire root-finding algorithm, require only algebraic operations in $\mathbf{F}_{p^k}$, and the latter can be reduced to algebraic operations (and equality tests) in $\mathbf{F}_p$ (with respect to the normal-basis representation). The expected number of solutions for β_{l-1} is roughly 1 because

$$\frac{|H|}{p^n} = \frac{1}{p^k} \pm \Theta\left(\frac{1}{p^{k+1}}\right) \ .$$

If no solution is found, then failure is reported.

Because the Cantor-Zassenhaus algorithm has probabilistic running time polynomial in n and $\log p$ and uses only algebraic operations in $\mathbf{F}_p$, the required executions of the EMBED procedure run in a probabilistic polynomial (in $\log |G|$) number of algebraic operations if n is polynomial in $\log p$.

Proof that Condition 2 is sufficient. Let $|H| = f(p)$ for some non-constant polynomial (with integer coefficients) $f(x)$ dividing $x^N - 1$, where $N = O(1)$. We can assume without loss of generality that $f(x)$ equals a cyclotomic polynomial $\Phi_n(x)$ for some $n = O(1)$.

We show that the subgroup H of $\mathbf{F}_{p^n}^*$ with

$$|H| = \Phi_n(p)$$

(for $n = O(1)$) is defined strongly-(n, α)-algebraically over $\mathbf{F}_p$ for some $\alpha = (\log p)^{O(1)}$. This proves that Condition 2 is sufficient because a group which has a subgroup with this property has the property itself (the same EMBED algorithm can be used). Let

$$\Phi_n(x) = \sum_{j=0}^{\varphi(n)} c_j x^j$$

(with $c_j \in \mathbf{Z}$). Let further $\{\alpha_0, \ldots, \alpha_{n-1}\}$ be a normal basis of $\mathbf{F}_{p^n}$ over $\mathbf{F}_p$.

We describe the EMBED algorithm for H. Let $x, e \in \mathbf{F}_p$ be given. We compute, by a polynomial number of algebraic operations in $\mathbf{F}_p$, an element $\beta = (\beta_0, \ldots, \beta_{n-1})$ such that $x + e$ is one of the coordinates of β, for instance $x + e = \beta_0$. Again, we need an alternative characterization, in terms of the $\mathbf{F}_p$-coordinates of β, of the fact that $\beta \in H$ holds. The following conditions are equivalent for $\beta = \sum \beta_i \alpha_i$.

$$\begin{aligned}
\beta \in H \quad &\Longleftrightarrow \quad \beta^{|H|} = 1 \\
&\Longleftrightarrow \quad \left(\sum_{i=0}^{n-1} \beta_i \alpha_i \right)^{\sum_{j=0}^{\varphi(n)} c_j p^j} = 1 \\
&\Longleftrightarrow \quad \prod_{j=0}^{\varphi(n)} \left(\left(\sum_{i=0}^{n-1} \beta_i \alpha_i \right)^{p^j} \right)^{c_j} = 1
\end{aligned}$$

$$\iff \prod_{j=0}^{\varphi(n)} \left(\sum_{i=0}^{n-1} \beta_i \alpha_{i+j} \right)^{c_j} = 1$$

$$\iff \prod_{c_j \geq 0} \left(\sum_{i=0}^{n-1} \beta_i \alpha_{i+j} \right)^{c_j} - \prod_{c_j < 0} \left(\sum_{i=0}^{n-1} \beta_i \alpha_{i+j} \right)^{-c_j} = 0 \ .$$

In the fourth step, we have made use of $\beta_i^{p^j} = \beta_i$ (because $\beta_i \in \mathbf{F}_p$) and $\alpha_i^{p^j} = \alpha_{i+j}$ (by the definition of the normal basis). The last condition corresponds to a system of n polynomial equations (with $\mathbf{F}_p$-coefficients) in the β_i, where the maximal degree D of the polynomials is bounded by

$$D \leq \max \left\{ \sum_{c_j > 0} c_j \, , \, \sum_{c_j < 0} |c_j| \right\} \leq \varphi(n) \cdot \max_j |c_j| \ .$$

As in the first part of the proof, the EMBED algorithm assigns $x + e$ to one of the β_i's, random values to some of the other β_i's, and solves the arising equations over $\mathbf{F}_p$ for the remaining β_i's. Because

$$\frac{|H|}{p^n} = \frac{1}{p^{n-\varphi(n)}} \pm \Theta\left(\frac{1}{p^{n-\varphi(n)+1}} \right) \ ,$$

i.e., approximately every $p^{n-\varphi(n)}$-th element β of $\mathbf{F}_{p^n}$ is also an element of H, we have to solve the equations for $n - \varphi(n)$ different coordinates β_i simultaneously in order to have an expectation of one solution. (If no solution is found, the algorithm reports failure.)

Using Gröbner bases, this system of polynomial equations can now be transformed into an equivalent system of triangular form (see Section 8.1.3). The computation of the Gröbner basis uses only algebraic operations in $\mathbf{F}_p$, and its complexity is of order $O(D^{n^2})$ (see [41]). The triangular system of equations can be solved by the Cantor-Zassenhaus algorithm for solving *univariate* polynomial equations.

According to the result of Gianni and Kalkbrener (see [41]), it suffices to solve a subset of $n - \varphi(n)$ equations. The first polynomial has to be solved once, the second one D' times (where D' is defined as in (8.5); the reason is that in the worst case, the first polynomial has D' different solutions), the third one $(D')^2$ times, etc. This yields $O((D')^n)$ executions of the Cantor-Zassenhaus algorithm. (The *effective* number of executions will be much smaller in a typical case, since only about one solution is expected.)

The expected complexity of the required executions of the EMBED algorithm is polynomial in $\log p$ (and the algorithm uses only algebraic operations in $\mathbf{F}_p$) if $n = O(1)$. □

Corollary 10.12. *Let $\mathcal{M} \subset \mathbf{N}$ be the set of numbers n with the following properties. First, all the multiple prime factors of n are of order $(\log n)^{O(1)}$. Then, for every prime factor p not of this order, one of the following numbers must be $(\log n)^{O(1)}$-smooth: either*

$$\frac{(p^k)^l - 1}{p^k - 1} = (p^k)^{l-1} + \cdots + p^k + 1 \ ,$$

where $k, l = (\log p)^{O(1)}$, or

$$f(p) \ ,$$

where $f(x) \in \mathbf{Z}[x]$ is a non-constant polynomial dividing $x^N - 1$ for some $N = O(1)$. Then the DHP *and* DLP *are polynomial-time computationally equivalent in a generic and uniform sense for groups G with order $|G|$ in $\mathcal{M}$, i.e.,*

$$\mathrm{DHP}(\mathcal{M}) \approx_g \mathrm{DLP}(\mathcal{M}) \ .$$

Proof. Let $n \in \mathcal{M}$. According to a result by Bach and Shallit [3], the number n can efficiently be entirely factored because for every large prime p dividing n, $\Phi_i(p)$ is $(\log n)^{O(1)}$-smooth for some $i = (\log n)^{O(1)}$, where Φ_i is the i-th cyclotomic polynomial. Given the factorization of n, the statement follows immediately from Theorems 10.1 and 10.11. □

Corollary 10.13. *Let $\mathcal{M}$ be defined as in Corollary 10.12. If there exists no generic uniform polynomial-time algorithm computing discrete logarithms in groups G with order $|G| \in \mathcal{M}$, then the* DH *protocol cannot be broken by a generic uniform polynomial-time algorithm in these groups.*

10.4.2 Explicit Construction of Groups with Uniform Polynomial-Time Equivalence

We describe a set $\mathcal{M} \subset \mathbf{N}$ with the property that

$$\mathrm{DHP}(\mathcal{M}) \approx_g \mathrm{DLP}(\mathcal{M})$$

holds, and such that groups whose orders belong to this set can be efficiently constructed. These group orders n are such that for every large prime factor p of n, an elliptic curve over $\mathbf{F}_p$ with $(\log n)^{O(1)}$-smooth order exists and can even be found efficiently.

However, such groups should be used with care since it appears conceivable, though not likely, that knowledge of suitable auxiliary groups can help an adversary computing discrete logarithms even without the help of a DH oracle.

We generalize an algorithm, presented in [115] by Vanstone and Zuccherato, for constructing a large prime p such that certain specific elliptic curves over $\mathbf{F}_p$ have smooth order.

The curve over $\mathbf{F}_p$ defined by the equation

$$y^2 = x^3 - Dx \tag{10.14}$$

has order $p+1$ if $p \equiv 3 \pmod 4$, and the curve

$$y^2 = x^3 + D \tag{10.15}$$

has also order $p+1$ if $p \equiv 2 \pmod 3$. The following statements about the orders of curves of the form (10.14) or (10.15) in the case they are *not* $p+1$ are proved in [47].

If $p \equiv 1 \pmod 4$, then p can uniquely be represented as the sum of two squares, i.e.,

$$p = a^2 + b^2 \ .$$

Then the curves $y^2 = x^3 - Dx$ have the orders

$$p+1 \pm 2a \ , \quad p+1 \pm 2b \ , \tag{10.16}$$

and the four orders occur equally often over the choices of D.

If $p \equiv 1 \pmod 3$, then p can uniquely be represented as

$$p = a^2 - ab + b^2$$

with $a \equiv 2 \pmod 3$ and $b \equiv 0 \pmod 3$. Then the curves $y^2 = x^3 + D$ have the orders

$$p+1 \pm 2a \ , \quad p+1 \pm a \mp 2b \ , \quad p+1 \pm (a+b) \ , \tag{10.17}$$

and the six orders occur equally often over the choices of D.

The construction now works as follows. First, we show how to find primes

$$p = a^2 + (k \pm 1)^2$$

(for a fixed k with l digits) such that $a^2 + k^2$, which is then one of the possible orders of the curves $y^2 = x^3 - Dx$ is smooth.

Let l'-digit numbers x_1, x_2, y_1, and y_2 be chosen at random. Define

$$u + vi := (x_1 + y_1 i)(x_2 + y_2 i) \ ,$$

that is

$$u = x_1 x_2 - y_1 y_2, \quad v = x_1 y_2 + x_2 y_1 \ .$$

Then u and v have at most $2l'$ digits. If $\gcd(u, v)$ divides k (otherwise choose again), one can compute numbers c and d (of at most $2l' + l$ digits) such that

$$cv + du = k \ .$$

Define

$$a := cu - dv \ ,$$

and restart the process if a is even. Then

$$a + ki = (c + di)(u + vi) = (c + di)(x_1 + y_1 i)(x_2 + y_2 i)$$

and

$$a^2 + k^2 = (c^2 + d^2)(x_1^2 + y_1^2)(x_2^2 + y_2^2) \ .$$

The process is repeated until $a^2 + k^2$ is s-digit-smooth, which happens with probability approximately

$$\left(\frac{4l' + 2l}{s}\right)^{-\frac{4l'+2l}{s}} \cdot \left(\frac{2l'}{s}\right)^{-\frac{2l'}{s}} \cdot \left(\frac{2l'}{s}\right)^{-\frac{2l'}{s}}$$

(according to (10.4)), and smoothness can be tested with the elliptic-curve integer-factoring algorithm [53]. Because a and k are odd, exactly one of the expressions $a + (k \pm 1)i$ is congruent to 1 modulo $2 + 2i$. Let $\alpha := a + (k \pm 1)i$, respectively. Repeat the computations until

$$p := \alpha\overline{\alpha} = a^2 + (k \pm 1)^2$$

is prime. According to (10.16), a quarter of the curves $y^2 = x^3 - Dx$ over $\mathbf{F}_p$ have (smooth) order $a^2 + k^2$. Hence p is an $(8l' + 2l)$-digit prime such that an elliptic curve with s-digit-smooth order is constructible over $\mathbf{F}_p$. The expected number of necessary trials is

$$O\left(\left(\frac{4l' + 2l}{s}\right)^{\frac{4l'+2l}{s}} \cdot \left(\frac{2l'}{s}\right)^{\frac{4l'}{s}} \cdot (8l' + 2l)\right) . \tag{10.18}$$

In a similar way, primes can be constructed such that curves of type $y^2 = x^3 + D$ have smooth order. More precisely, one can generate primes

$$p = a^2 - a(k \pm 1) + (k \pm 1)^2$$

such that $a^2 - ak + k^2$ is one of the possible orders of the curves $y^2 = x^3 + D$ over $\mathbf{F}_p$, and s-digit-smooth.

In case of a small k, an L-digit prime p such that an s-digit-smooth curve is constructible over $\mathbf{F}_p$ can be found by

$$O\left(\left(\frac{L}{\sqrt{8} \cdot s}\right)^{\frac{L}{s}} \cdot L\right)$$

trials instead of

$$O\left(\left(\frac{L}{s}\right)^{\frac{L}{s}} \cdot L\right)$$

trials when varying p among L-digit numbers until p is prime and one of the considered curves is s-digit-smooth. For example, a 100-digit prime p such that a 10-digit-smooth curve over $\mathbf{F}_p$ is constructible can be found by approximately $3 \cdot 10^6$ trials (instead of about 10^{11} trials when using the straight-forward strategy).

10.4.3 Subexponential-Time Equivalence: Trading Rigor for Speed

In Section 10.4.1, polynomial-time uniform reduction algorithms of the DLP to the DHP were described that work for certain classes of groups. In this section we show that a *subexponential-time* reduction exists for all groups (of course with the exception of groups whose orders contain multiple prime factors of exponential size, for which not even a nonuniform subexponential-time reduction exists). Such a result is of interest for classes of groups for which no subexponential-time algorithm for the computation of discrete logarithms is known, such as non-supersingular elliptic curves or Jacobians of hyperelliptic curves.

Boneh and Lipton [10] gave a uniform reduction algorithm with heuristic subexponential running time using elliptic curves. In the same way as the Goldwasser-Kilian [44] primality prover can be improved by the use of Jacobians of hyperelliptic curves instead of elliptic curves [1], a rigorously-provable (yet apparently less efficient) uniform reduction of the DLP to the DHP can be achieved by using Jacobians as the auxiliary groups in the technique of Section 10.2 (see [121]). The running time of the reduction algorithm follows from the complexity analysis of a hyperelliptic factoring algorithm [54].

The basic idea of Boneh and Lipton's [10] reduction is as follows. Like in Lenstra's elliptic-curve integer-factoring method [53], elliptic-curve parameters are chosen at random until a curve with subexponentially-smooth order is generated. The running-time analysis of such an algorithm is based on the following conditions. First, the orders of elliptic curves with randomly-chosen parameters should be (almost) uniformly distributed in the interval of interest (or a substantial sub-interval). Secondly, the density of smooth numbers in this interval must be high enough. The first problem was solved by Lenstra [53]. The orders of a non-negligible fraction of the elliptic curves with randomly chosen a, x, and y lie in the interval $(p+1-\sqrt{p}, p+1+\sqrt{p})$ and are distributed close to uniformly in this interval. The second problem, concerning smooth numbers in small intervals of type $[x, x+\Theta(\sqrt{x})]$ has not been solved, hence the running time of the algorithm is heuristic. As mentioned earlier, the density of smooth numbers in sufficiently large intervals satisfies (10.4). Motivated by this, the plausible assumption is often made that the statement concerning the density also holds for smaller intervals (see [53] or [10]). We define

$$L_\alpha(n) := e^{(\log n)^\alpha (\log\log n)^{1-\alpha}} \ .$$

Running times of order $L_\alpha(n)^{O(1)}$, where $\alpha < 1$, are called *subexponential.*

Smoothness Assumption 2. Integers that are randomly chosen from the interval $(p+1-\sqrt{p}, p+1+\sqrt{p})$ are $L_\alpha(p)$-smooth with probability at least

$$\frac{1}{L_{1-\alpha}(p)^{1-\alpha+o(1)}}$$

for any α.

This statement for $\alpha = 1/2$ allows a running-time analysis of both the elliptic-curve integer-factoring algorithm and of the reduction of the DLP to the DHP. (The assumption for $\alpha = 2/3$ and $\alpha = 1/2$ is necessary for a reduction of algebraic algorithms which is described in Section 10.4.4.)

By using the described method of choosing elliptic curves over $\mathbf{F}_p$ at random, computing their order in polynomial time by a method due to Schoof [104], checking the smoothness of this order with the elliptic-curve factoring algorithm (choosing a new curve unless the order is $L_{1/2}(p)$-smooth), and applying the technique described in the previous sections, Boneh and Lipton [10] showed the following result.

Theorem 10.14. [**10**] *Assume that the smoothness assumption 2 is true for $\alpha = 1/2$. Let $\mathcal{M} \subset \mathbf{N}$ be the set of numbers n with the property that all the multiple prime factors of n are of order $L_{1/2}(n)$. Then*

$$\mathrm{GC}_u(\mathrm{DLP}(\mathcal{M}) \downarrow \mathrm{DHP}(\mathcal{M})) = L_{1/2}(n)^{2+o(1)} \ .$$

Of course this result is useless for groups in which the DLP can be solved in time $L_{1/2}(|G|)^{2+o(1)}$ such as the multiplicative group of a finite field. Theorem 10.14 implies that *if* elliptic curves have the advantage that no subexponential solution to the DLP exists, *then* they have the *additional* advantage that this also leads to a lower bound on the complexity of breaking the DH protocol.

In Section 10.4.4 an even faster subexponential reduction will be described that works under the additional condition that a DH oracle for elliptic curves is given by a subexponential-time *algebraic* algorithm.

In order to obtain an integer-factoring algorithm with rigorously-proven running time, Lenstra, Pila, and Pomerance [54] have studied the use of hyperelliptic curves of genus 2 instead of elliptic curves. Jacobians of these curves have the advantage that the group order varies in an interval of size

$$[x, x + \Theta(x^{3/4})]$$

instead of $[x, x + \Theta(\sqrt{x})]$. In a series of papers, the first of which has already been published [54], they prove that the factoring algorithm using Jacobians of hyperelliptic curves of genus 2 has running time at most

$$L_{2/3}(p)^{O(1)} \cdot (\log n)^2 ,$$

where p is the largest prime factor of n. In the mentioned first paper of the series, a result on smooth numbers is shown which roughly corresponds to the statement of the smoothness assumption 2 for $\alpha = 2/3$, and for larger intervals of the form $[x, x+y]$, where $x^{3/5} \leq y \leq x$.

Because of the obvious similarities of the two methods, the analysis of the integer-factoring algorithm also yields a *provable* uniform and generic subexponential-time reduction of the DLP to the DHP. (It was shown in [121] that Jacobians can be used as auxiliary groups in the technique of Section 10.2.) As in the case of the factoring algorithms, this reduction is less efficient and less practical than the reduction using elliptic curves, but it is an improvement in the sense that it is so far the best *rigorously-proven* complexity result.

Theorem 10.15. *Let $\mathcal{M} \subset \mathbf{N}$ be the set of numbers n with the property that all the multiple prime factors of n are of order $L_{2/3}(n)^{O(1)}$. Then*

$$\mathrm{GC}_u(\mathrm{DLP}(\mathcal{M}) \downarrow \mathrm{DHP}(\mathcal{M})) = L_{2/3}(n)^{O(1)} .$$

Corollary 10.16. *Let $2/3 \leq \alpha < 1$, and let $\mathcal{M} \subset \mathbf{N}$ be the set of numbers n with the property that all the multiple prime factors of n are of order $L_\alpha(n)^{O(1)}$. Then*

$$\mathrm{GC}_u(\mathrm{DLP}(\mathcal{M}) \downarrow \mathrm{DHP}(\mathcal{M})) = L_\alpha(n)^{O(1)} .$$

Corollary 10.17. *Let α and $\mathcal{M}$ be as in Corollary 10.16. Assume that for a certain class $\mathcal{G} = (G_i)_{i \in I}$ of groups with $|G_i| \in \mathcal{M}$ for all i, there exists no uniform subexponential-time algorithm solving the* DLP. *Then there exists no uniform subexponential-time algorithm solving the* DHP *in the groups of $\mathcal{G}$.*

The statement of Corollary 10.17 is stronger if α is greater (but would be false for $\alpha = 1$). It is widely believed that for the classes of elliptic curves or Jacobians of hyperelliptic curves over finite fields, there exist no subexponential discrete-logarithm algorithms. It should be pointed out however that Corollary 10.17 has no significance for subgroups of finite fields for instance.

10.4.4 Algebraic Algorithms Solving the DHP

The results described in this section are based on the following observation. Assume that not only a DH oracle for a group G, but also an efficient *algorithm* which solves the DHP in an entire class of groups, such as elliptic curves over a finite field or the groups $\mathbf{F}_p^*$, is given. If this algorithm additionally has the property that it uses only algebraic operations in the underlying field, then it can be executed on inputs that are not explicitly known, but only implicitly represented (in the sense of Section 10.2.1). Notice that the condition that an algorithm, working for a certain class of groups defined over a finite field, is algebraic over this field is much weaker than the condition that an algorithm is generic. This property allows to iterate the reduction algorithm of Section 10.2, i.e., computing discrete logarithms in G is reduced to the same problem in a group $\mathbf{F}_p^*$, which is further reduced to the DLP of another group $\mathbf{F}_q^*$, and so on.

Example 10.1. Assume that a uniform polynomial-time algorithm exists for solving the DHP in the groups $\mathbf{F}_p^*$, and that this algorithm only uses algebraic operations in the corresponding field $\mathbf{F}_p$. Let $B = (\log|G|)^{O(1)}$ be a smoothness bound, p_0 a prime factor of $|G|$ greater than B, and let p_1 be the only prime factor of $p_0 - 1$ greater than B. Assume further that p_i is the only prime factor of $p_{i-1} - 1$ greater than B for all $i = 2, \ldots, k$, that $p_k - 1 =: T = \prod_i r_i^{n_i}$ is B-smooth, and $k = O(1)$.

Given $a = g^s$, it is possible to compute $x_0 \equiv s \pmod{p_0}$, $x_0 \in \mathbf{F}_{p_0}$, in polynomial time as follows when given a DH oracle for G. Let

$$h_0 := \frac{|G|}{p_0}, \quad h_i := \frac{p_{i-1} - 1}{p_i} \quad (\text{for } i = 1, \ldots, k) ,$$

and let

$$\mathbf{F}_{p_i}^* = \langle c_i \rangle \quad (\text{for } i = 0, \ldots, k-1) .$$

If $x_0 \neq 0$, then $x_0 = c_0^{w_0}$ (in $\mathbf{F}_{p_0}$) holds for some w_0, and g^s is an implicit representation of x_0. Since $p_0 - 1$ has a large prime factor p_1, w_0 modulo p_1 cannot be obtained directly. But $x_1 \equiv w_0 \pmod{p_1}$ (with $x_1 \in \mathbf{F}_{p_1}$) can be written as $x_1 = c_1^{w_1}$ (in $\mathbf{F}_{p_1}$, if $x_1 \neq 0$), and g^s is a "double-implicit" representation of x_1. Our assumptions allow efficient computation with these elements of $\mathbf{F}_{p_1}$ which are "double-implicitly" represented. For example, an addition of two $\mathbf{F}_{p_1}$-elements requires multiplication of the corresponding implicitly-represented $\mathbf{F}_{p_0}^*$-elements and can be obtained by a call to the DH oracle for G. A multiplication in $\mathbf{F}_{p_1}^*$ is done by an oracle call for $\mathbf{F}_{p_0}^*$ with implicitly-represented arguments and an implicitly-represented answer. This works (in polynomial time) because of the stated properties of the DH algorithm for $\mathbf{F}_{p_0}^*$.

Analogously, computation with $(k+1)$-fold implicitly-represented arguments is possible in the smooth group $\mathbf{F}_{p_k}^*$. The index-search problem for the list of elements

$$g^{h_0 c_0^{h_1 c_1^{\cdots^{h_k c_k^{\frac{T}{r_i} t}}}}}$$

where $t = 0, \ldots, r_i - 1$, and the element

$$g^{h_0 c_0^{h_1 c_1^{\cdots^{h_k c_k^{\frac{T}{r_i} w_k}}}}}$$

which can be obtained in polynomial time by computation with multiply-implicitly-represented arguments, is solved and leads to w_k modulo r_i. When this is done for all prime powers $r_i^{n_i}$, w_k is computable modulo T. Then $x_k = c_k^{w_k}$ (in $\mathbf{F}_{p_k}$), and one can get w_{k-1} modulo $p_{k-1} - 1$ in polynomial time because the other prime factors of $p_{k-1} - 1$ are smaller than B. Finally, we obtain w_0 modulo $p_0 - 1$ and x_0. ♢

Remark. The reason for assuming that $p_{i-1} - 1$ has *only one* large prime factor p_i is that otherwise it would not be possible to find the factors of $p_{i-1} - 1$ in polynomial time. When these factors are given, then the condition is unnecessary.

Theorem 10.18 is an immediate consequence of the technique described in Example 10.1.

Theorem 10.18. *Assume that a uniform polynomial-time algebraic algorithm solving the* DHP *in groups* $\mathbf{F}_p^*$ *exists. Let* $\mathcal{M} \subset \mathbf{N}$ *be the set of numbers* n *with the property that there exists* $k = O(1)$ *and primes* p_i $(i = 1, \ldots, k)$ *such that* p_i *is the only prime factor of* $p_{i-1} - 1$ *not of order* $(\log n)^{O(1)}$ *for* $i = 1, \ldots, k$*, and* $p_k - 1$ *is* $(\log n)^{O(1)}$*-smooth. Then*

$$\mathrm{DHP}(\mathcal{M}) \approx_g \mathrm{DLP}(\mathcal{M}) \ .$$

The process works in an analogous way if some of the used groups are cyclic elliptic curves or Jacobians, provided an efficient algebraic (with respect to the underlying field $\mathbf{F}_p$) DH algorithm is given for these groups.

Corollary 10.19. *Let $\mathcal{M}$ be defined as in Theorem 10.18. Assume that for a class $\mathcal{G} = (G_i)_{i \in I}$ of groups, with the property that $|G_i| \in \mathcal{M}$ for all $i \in I$, there exists no uniform polynomial-time discrete-logarithm algorithm. Then either there exists no uniform polynomial-time algorithm solving the* DHP *for the groups in $\mathcal{G}$, or the* DH *protocol cannot be broken in polynomial time by a uniform* algebraic *algorithm in the groups $\mathbf{F}_p^*$.*

In analogy to the results on polynomial- and subexponential-time uniform generic equivalence of Sections 10.4.1 and 10.4.3, the result on polynomial-time equivalence for a certain subclass of groups is complemented by a subexponential-time result concerning (almost) all groups.

Boneh and Lipton [10] independently used the same technique (with an iteration depth 2, and with randomly-chosen elliptic curves as auxiliary groups) in the context with algebraic algorithms. They showed that under the smoothness assumption 2 and under the additional assumption that the DHP can be solved in time $L_{1/3}(q)$ in an elliptic curve over $\mathbf{F}_q$ by an algebraic algorithm, a uniform reduction of the DLP to the DHP exists of complexity $L_{1/3}(p)^{2+o(1)}$ (which is substantially faster than the $L_{1/2}(p)^{2+o(1)}$-reduction of Theorem 10.14).

Theorem 10.20. **[10]** *Assume that the smoothness assumption 2 is true for $\alpha = 1/2$ and $\alpha = 2/3$, and that there exists a uniform algebraic algorithm solving the* DHP *in elliptic curves over $\mathbf{F}_q$ in time $L_{1/3}(q)$. Let $\mathcal{M} \subset \mathbf{N}$ be the set of numbers n such that every prime factor of n greater than $L_{1/3}(n)$ is single. Then*

$$\mathrm{GC}_u(\mathrm{DLP}(\mathcal{M}) \downarrow \mathrm{DHP}(\mathcal{M})) = L_{1/3}(p)^{2+o(1)} .$$

Here, an increase of the iteration depth does not lead to a more efficient reduction. The reason is that according to the smoothness assumption 2 for $\alpha = 2/3$, a $L_{2/3}(p)$-smooth curve can be found in time $L_{1/3}(p)$, and that it takes roughly time $L_{1/2}(L_{2/3}(p)) \approx L_{1/3}(p)$ (under the smoothness assumption 2 for $\alpha = 1/2$) to factor the order of this curve. Hence the running time of the first iteration step (and hence of the entire reduction) cannot be smaller than $L_{1/3}(p)$, regardless of the iteration depth.

10.5 Discussion

The main conclusion of this chapter is that in many cases, the security of the DH protocol follows from the hardness of the DLP for the same group. More precisely, this holds for groups whose order is free of squares of large primes. Furthermore, if no uniform subexponential-time discrete-logarithm algorithm exists for a specific class of groups (e.g., elliptic curves) of certain orders, then the same is true for the DHP with respect to the same class. Additionally, polynomial-time uniform reductions have been given for groups whose orders are contained in certain subsets of $\mathbf{N}$.

The problem of *uniform polynomial-time* generic equivalence of the DHP and the DLP however remains open. With the technique of Section 10.2, such an equivalence could be shown only if integer factoring is easy (because the group order must be factored first). The relationship between uniformly reducing the DLP to the DHP and integer factoring (by algebraic algorithms) is even closer: The required structures are exactly the same, namely, (auxiliary) groups of smooth order defined algebraically over prime fields $\mathbf{F}_p$ and rings $\mathbf{Z}_n$. Groups of the same type can also be used for primality provers. The following table illustrates this correspondence.

Groups / Problems	$\mathbf{F}_p^*$	Subgroups of $\mathbf{F}_{p^N}^*$	Elliptic curves over $\mathbf{F}_p$	Jacobians over $\mathbf{F}_p$
Integer factoring	$(p-1)$-Pollard	Bach-Shallit	Lenstra	Lenstra-Pila-Pomerance
Primality proving	Pockling-ton		Goldwasser-Kilian	Adleman-Huang
Reducing DLP to DHP	den Boer	Maurer-Wolf	Maurer	Maurer-Wolf

Motivated by these close connections, we conjecture that the DHP and the DLP are not polynomial-time equivalent in a uniform generic sense unless factoring is easy.

Conjecture 10.1. *Let $\mathcal{M} \subset \mathbf{N}$ be the set of numbers n with the property that every multiple prime factor of n is of order $(\log n)^{O(1)}$. If there* *exists no polynomial-time integer-factoring algorithm, then*

$$\mathrm{DHP}(\mathcal{M}) \not\approx_g \mathrm{DLP}(\mathcal{M}) \ .$$

It should be pointed out in this context however that no evidence, except the apparent failure of many brilliant minds' efforts solving the problem, supports the fact that factoring is hard. In a remarkable contrast to this stands the wide-spread trust in RSA and related schemes.

Chapter 11

Generic Analysis and Comparison of Variants of the DHP and Related Problems

In this chapter we study the generic complexity and relationship of a number of problems related to the DHP. The objective is to get a complete classification, in the generic model, of the problems of interest (see Chapter 12).

Two problems are compared with the DHP which are directly connected with the security of the DH key. These are the problems of solving the DHP with a certain small but non-negligible probability ε (Section 11.1), and the apparently even easier problem of distinguishing the correct DH key from a random string (Section 11.3).

Furthermore, the hardness of the DHP in subgroups of a group G is compared to the same problem in G (Section 11.2). Finally, an interesting variation of the DHP, which is also of practical importance, is the so-called squaring-DHP: Given g^x, compute g^{x^2} (Section 11.4). The relationship between both the computational (Section 11.4.1) and decisional (Section 11.4.2) variants of the problem and the corresponding variants of the DHP is investigated.

Some of the results of this chapter have already been published in [70], [71], [72], [78], [73]. Most of the results of the Sections 11.3 and 11.4 are new.

11.1 Amplifying a Faulty DH Oracle

11.1.1 Motivation

The security of the DH protocol relies on the hardness of the DHP. However, according to any reasonable security definition, the underlying group must be such that the DHP cannot be solved efficiently even with small probability of, say, 0.01%. Motivated by this, we define a weak variant of the DHP and show that it is roughly as hard as the DHP. More precisely, we prove that a (probabilistic) DH oracle answering correctly with small probability is virtually as strong as an (almost) perfect DH oracle. For example, an oracle answering correctly with probability 0.01% can efficiently be transformed into an oracle that answers correctly with probability 99.99%.

11.1.2 The ε-DHP

We first define the ε-DHP and the notion of an ε-DH oracle for a cyclic group G with respect to a generator g. Note that such an "oracle" is probabilistic in general rather than deterministic.

Definition 11.1. Let G be a cyclic group with generator g, and let $\varepsilon > 0$ be a constant. The ε-DHP in G is, given two group elements g^u and g^v chosen randomly according to the uniform distribution in $G \times G$, to compute a group element g^w such that $g^w = g^{uv}$ holds with probability at least ε.

An ε-DH *oracle* is a probabilistic oracle which returns for an input (g^u, g^v) the correct answer g^{uv} with probability at least ε if the input is uniformly distributed over $G \times G$. The *offset* of the oracle's answer g^t to the input (g^u, g^v) is defined as $t - uv \pmod{|G|}$. A *translation-invariant* ε-DH oracle is an ε-DH oracle whose offset distribution is the same for every input (g^u, g^v). ○

A special case of (non-translation-invariant) ε-DH oracles are *deterministic* oracles answering correctly for a fraction ε of all inputs. We describe a technique for improving an ε-DH oracle in two steps. First, the oracle is made translation-invariant by randomization of the input, and then, the translation-invariant oracle is "amplified" to an (almost) perfect oracle.

Lemma 11.1. [**62**] *An ε-DH oracle for a cyclic group G with order $|G|$ can efficiently be transformed into a translation-invariant ε-DH oracle. More precisely, implementing one call to the latter requires one call to the former and $O(\log|G|)$ group operations.*

Proof. Given the group elements $a = g^u$ and $b = g^v$ we can randomize the input by choosing r and s at random from $[0, |G|-1]$, providing the oracle with $a' = ag^r$ and $b' = bg^s$ and multiplying the oracle's answer

$$g^{(u+r)(v+s)+t} = g^{uv+rv+su+rs+t}$$

with

$$(a^{-1})^s \cdot (b^{-1})^r \cdot g^{-rs} = g^{-(rv+su+rs)}$$

to obtain g^{uv+t}. Note that a' and b' are random group elements and statistically independent of a and b. The ε-DH oracle with randomized input is thus a translation-invariant ε-DH oracle. □

Remark. If $|G|$ is unknown, the input can be randomized nevertheless, where the random numbers are chosen from a larger interval. The resulting ε-DH oracle is then "almost translation-invariant" and applicable in the proof of Theorem 11.2 if the interval is of size at least $2 \cdot |G|/(\varepsilon^2 \cdot \min\{s, 0.1\})$ (where s is as in Theorem 11.2). This is the reason for the greater number of group operations for this case in Theorem 11.2.

In the proof of Theorem 11.2 it is shown that a translation-invariant ε-DH oracle can be transformed into an almost-perfect DH oracle. The straight-forward approach to using a translation-invariant ε-DH oracle may at first sight appear to be to run it $O(1/\varepsilon)$ times until it produces the correct answer. However, because of the hardness of the DHDP, a more complicated approach must be used.

Theorem 11.2. *Let $\varepsilon > 0$ and $\beta > 0$ be arbitrary constants. For every cyclic group G with generator g and known order $|G|$ there exists an algorithm for solving the $(1-\beta)$-DHP in G which makes calls to an ε-DH oracle. The number of required oracle calls is*

$$O\left(\frac{\log(1/\beta\varepsilon)}{\varepsilon^4}\right) .$$

If the order of G is unknown, then the reduction is also possible if all the prime factors of $|G|$ are greater than $(1+s)/\varepsilon$ for some $s > 0$. The number of required calls to the ε-DH oracle is then

$$O\left(\frac{1}{(\varepsilon^2 \cdot \min\{s,1\})^2} \cdot \log \frac{1}{\beta\varepsilon}\right) .$$

The number of required group operations is $O(\log|G|)$ times the number of oracle calls if $|G|$ is known and

$$O\left(\log\left(\frac{|G|}{\varepsilon^2 \cdot \min\{s,1\}}\right)\right)$$

times this number if $|G|$ is not known, respectively.

Proof. The basic idea of the amplification of the ε-DH oracle is as follows. In a precomputation phase, which is independent of the actual input, the oracle's offset distribution is determined. Then, the oracle is called with the given input to compute the correct solution with overwhelming probability.

More precisely, the reduction from an ε-DH oracle to an oracle answering correctly with high probability consists of the following steps which we first describe intuitively.

Step 1. The ε-DH oracle is transformed into a *translation-invariant* ε-DH oracle.

Step 2. We compute an estimate ε' for the probability that the (translation-invariant) oracle answers correctly.

Step 3. A list L_1 of group elements g^e is computed with the property that g^e is contained in L_1 if and only if the probability of the offset e is close to ε'.

Step 4. A second list L_2 of group elements is generated which contains exactly those group elements that occur with frequency close to ε' when the oracle is called with the input (g^u, g^v).

Step 5. The lists L_1 and L_2 have (with high probability) the property that the elements of L_2 are exactly the elements of L_1 multiplied by the group element g^{uv} (which is itself contained in L_2). In order to determine this switch element, the lists aL_1 are generated for all elements a in L_2 (the list aL_1 contains exactly the elements al_1, where l_1 is contained in L_1). The list L_2 is compared to all the lists aL_1, and equality yields a candidate a for g^{uv}.

Step 6. In case of one single candidate a for g^{uv}, this is the output of the algorithm. In the case of *several* candidates and if the group order $|G|$ is known, the discrete logarithms of all the candidates and of g^u and g^v are determined modulo the smooth part of $|G|$. This yields the correct candidate for g^{uv}, which is then the output of the algorithm.

Note that the first three steps are a precomputation which is independent of the particular input (g^u, g^v). The list L_1 which is generated in these steps is a reference list describing the offset behavior of the faulty oracle. We describe the steps in detail and analyze their correctness and efficiency.

Step 1. According to Lemma 11.1, one can construct a translation-invariant ε-DH oracle which uses $O(\log |G|)$ group operations and one call to an ε-DH oracle per call if $|G|$ is known. If $|G|$ is unknown, the number of group operations is $O(\log(|G|/(\varepsilon^2 \cdot \min\{s, 1\})))$.

Step 2. Let $\alpha := \beta\varepsilon/8$. An event with probability at least $1 - \alpha$ will be called *almost certain*. Let $\delta := \varepsilon/10$ and $\delta' := \delta\varepsilon/100 = \varepsilon^2/1000$. (The proof does not depend on the choice of the constants, which is somewhat arbitrary. Intuitively, we need that $\delta \ll \varepsilon$ and $\delta' \ll \varepsilon\delta$.) If ε is not known we take a lower bound instead. In order to determine the probability of a correct answer, the translation-invariant oracle is called repeatedly with the input (g^0, g^0), and ε' is the fraction of correct answers g^0. The number t of oracle calls is such that the true probability of a correct answer lies almost certainly in the interval $[\varepsilon' - \delta', \varepsilon' + \delta']$. It follows from Lemma 2.2 that t can be of order

$$O\left(\frac{\log(1/\alpha)}{(\delta')^2}\right) .$$

Step 3. In this step the reference list L_1 is generated as follows. The faulty oracle is called t times (for the same value of t as in the previous step), and all the occurring group elements are stored. Let the list L_1 consist of those group elements whose fraction in the set of all answers lies in the interval $[\varepsilon' - (\delta + \delta'), \varepsilon' + (\delta + \delta')]$. According to Lemma 2.2, and because $2/\varepsilon$ is an upper bound on the number of offsets occurring with probability at least $\varepsilon/2$, with probability $(1-\alpha)^{4/\varepsilon}$ the following two statements are both true.

1. If e is an offset with probability in $[\varepsilon'-\delta, \varepsilon'+\delta]$, then g^e is contained in L_1.
2. If g^e is in L_1, then the offset e has probability in $[\varepsilon'-(\delta+2\delta'), \varepsilon'+(\delta+2\delta')]$.

Step 4. The translation-invariant faulty oracle is called repeatedly with the input (g^u, g^v), where (g^u, g^v) is the input to the (reliable) algorithm for G. Let the list L_2 then consist of those group elements which occur as answers of the oracle with a frequency in

$$[\varepsilon'-(\delta+3\delta'), \varepsilon'+(\delta+3\delta')] \ .$$

Then, for the same number of trials t as in the previous step, with probability at least $(1-\alpha)^{4/\varepsilon}$ the following statements are true.

1. If e is an offset with probability in $[\varepsilon'-(\delta+2\delta'), \varepsilon'+(\delta+2\delta')]$, then g^{uv+e} is contained in L_2.
2. If g^{uv+e} is in L_2, then the offset e has probability in $[\varepsilon'-(\delta+4\delta'), \varepsilon'+(\delta+4\delta')]$.

Step 5. With high probability, the list L_2 is equal to L_1, switched by g^{uv} (which is itself in L_2). More precisely, it follows from the above that the probability that L_1 contains *all* the offsets which have their probability in the interval $[\varepsilon'-\delta, \varepsilon'+\delta]$, that all the offsets of L_1 also occur in L_2, and that all the offsets of L_2 have probability in $[\varepsilon'-(\delta+4\delta'), \varepsilon'+(\delta+4\delta')]$ is at least

$$(1-\alpha)^{8/\varepsilon} \geq 1-\frac{8\alpha}{\varepsilon} = 1-\beta \ . \tag{11.1}$$

If this is fulfilled, then L_2 contains more elements than L_1 only if there exists an offset whose probability lies in the set

$$[\varepsilon'-(\delta+4\delta'), \varepsilon'-\delta] \cup [\varepsilon'+\delta, \varepsilon'+(\delta+4\delta')] \ . \tag{11.2}$$

In this case we replace δ by $\delta+i\cdot 5\delta'$ (for an integer i randomly chosen in $[-2/\varepsilon, 2/\varepsilon]$), leave δ' unchanged, and run the entire algorithm (except

Steps 1 and 2) again. Because the sets (11.2) are disjoint for different i, and because there can be at most $2/\varepsilon$ offsets with probability at least $\varepsilon/2$, L_1 and L_2 contain the same number of elements for at least half of the possible choices for i.

If the lists L_1 and L_2 have equal length, then with probability at least $1-\beta$ we have that g^{uv} is contained in L_2 and $L_2 = g^{uv}L_1$, i.e., L_2 contains exactly the elements $g^{uv}l_1$ for l_1 in L_1. The lists aL_1 are computed for all elements a of L_2 and compared to L_2. If equality holds, then a is a candidate for g^{uv}.

Step 6. Let c be the number of elements of L_1 and L_2. If there exists only *one* candidate for g^{uv}, then this group element is the output of the algorithm. If there exist *several* such elements, this means that the lists have a nontrivial translation symmetry or, more precisely, that they are invariant under a multiplication with $g^{|G|/c'}$ for a divisor c' of c and $|G|$. Let c' be the maximal number with this property. Note that $|G|$ has a factor $c' \leq c \leq 1/(\varepsilon' - (\delta + 2\delta'))$ in this case. There are c' candidates for g^{uv}, namely,

$$g^{uv},\ g^{uv+\frac{|G|}{c'}}, \dots ,\ g^{uv+(c'-1)\frac{|G|}{c'}} .$$

We show that if $|G|$ is known, the correct one of them can be determined. Let $p_1, \dots , p_l$ be the distinct prime factors of c' (they can be found in time

$$O\left(\frac{(\log(1/\varepsilon))^2}{\varepsilon}\right)$$

because $c' = O(1/\varepsilon)$), i.e., $c' = \prod_{i=1}^{l} p_i^{f_i}$. Let further $d = \prod_{i=1}^{l} p_i^{e_i}$ be the product of the maximal powers of the p_i dividing $|G|$. The number d can be computed in time $O((\log|G|)^3)$ and is $(2/\varepsilon)$-smooth because $c' \leq c \leq 2/\varepsilon$. Hence u and v (and consequently uv) are computable modulo d from g^u and g^v by the Pohlig-Hellman algorithm by $O((\log|G|)^2 + \log|G|/\varepsilon)$ group operations. Analogously, we can also compute the discrete logarithms of all the candidates modulo d.

The discrete logarithm of exactly one of the candidates has the correct remainder with respect to d. This is true because for every i, exactly every $p_i^{f_i}$-th candidate has the correct remainder with respect to $p_i^{e_i}$. The primes are distinct, thus the $p_i^{f_i}$ are relatively prime, and hence every $\prod_{i=1}^{l} p_i^{f_i}$-th candidate, that is *exactly one* of them, has the correct remainder. This group element is the output of the algorithm.

In the case where $|G|$ is *not* known, this last step of finding the correct candidate does not work. The only possibility is to choose a smaller value for δ. This is always successful if all the prime factors of $|G|$ are greater than $(1+s)/\varepsilon$ for some positive s. Then δ must be chosen smaller than $s\varepsilon/2$, such that

$$\frac{1}{\varepsilon' - (\delta + 2\delta')} < \frac{1+s}{\varepsilon} \tag{11.3}$$

holds. Inequality (11.3) implies that such a symmetry of the lists L_1 and L_2 (this symmetry is a necessary condition for the case of more than one candidate for g^{uv}) is not possible. □

Remark. An alternative technique for amplifying an ε-DH oracle was given by Shoup [109]. His method is more efficient than ours, but the advantage of our algorithm is that in the case of unknown group order, the condition concerning small prime factors is weaker. Our method works if all the prime factors of $|G|$ are greater than $1/\varepsilon$, whereas for Shoup's algorithm, this bound is of order $1/\varepsilon^2$. Examples of "malicious" oracles that *cannot* be transformed into perfect oracles with either of the methods when $|G|$ is unknown are those which answer the input (g^u, g^v) by one of the values $g^{uv+i|G|/z}$, where $z \leq 1/\varepsilon$ is a factor of $|G|$, and where all the values of i between 0 and $z-1$ are equally likely.

Note that a DH oracle as obtained in Theorem 11.2, for β exponentially small in $\log|G|$, is virtually equivalent to a perfect DH oracle in a polynomial-time (or subexponential-time) reduction of the DLP to the DHP because the correctness of the output of a probabilistic algorithm computing discrete logarithms can be tested, and because only a polynomial (or subexponential) number of oracle calls is required for the computation of a discrete logarithm.

Corollary 11.3. *Let* $0 < \alpha_1 < 1$ *and* $0 < \alpha_2 < 1$ *be arbitrary constants. Then*

$$\alpha_1\text{-DHP}(\mathbf{N}) \approx_g \alpha_2\text{-DHP}(\mathbf{N}) \ .$$

Clearly, another consequence of Theorem 11.2 is that the ε-DHP is as hard as the DHP with respect to generic algorithms. In Section 11.4, we analyze DH oracles that are even weaker than an ε-DH oracle with respect to the fraction of correctly answered inputs, namely so-called *squaring*-DH oracles.

11.2 Computing Roots and the Security of Subgroups

In this section we consider the problems of computing roots in cyclic groups and the related question whether the DH protocol in a subgroup of a group G is more or less secure than the same problem in G. Another natural question is whether the DHP in G is equally hard with respect to every generator of G.

The first two problems, which are both of independent interest, arose in the context of reducing the DLP to the DHP in groups whose orders contain multiple large prime factors (see Section 10.3). Note that according to Theorem 10.8, the equivalence of the DHP and DLP can in this case not be shown in the generic model.

11.2.1 Computing Roots in Cyclic Groups

Let G be a cyclic group. For two group elements a and b with $a^\ell = b$, we say that a is an *ℓ-th root* of b. The element b is then called an *ℓ-th power* in G. Exactly $|G|/\gcd(|G|, \ell)$ elements of G are ℓ-th powers, and each of them has exactly $\gcd(|G|, \ell)$ different ℓ-th roots.

We give a complete characterization of which roots can be computed in cyclic groups by efficient generic algorithms. Let us start with the possibility result.

Note first that it is sufficient to characterize when p^r-th roots can be computed in a group G for a prime power p^r. The reason is that the s_1-th root of a s_2-th root is always a $(s_1 s_2)$-th root, and that if $\gcd(s_1, s_2) = 1$ holds, then s_1-th and s_2-th roots are "independent," i.e., the s_1-th root of a group element is an s_2-th root if and only if the original group element is. Lemma 11.4 gives conditions under which p^r-th roots can be easily computed in G.

Lemma 11.4. *Let $G = \langle g \rangle$, $|G| = p^e s$, $\gcd(p, s) = 1$, and $0 \le e \le r$. Then p^r-th roots can be computed in G as follows. Let $d = g^{p^r k}$ be a p^r-th power in G, and let*

$$z :\equiv (p^r)^{-1} \pmod{s} .$$

Then,

$$d^z = g^{p^r k z}$$

is a p^r-th root of d.

Proof. From

$$p^r kz \cdot p^r \equiv 0 \equiv p^r k \pmod{p^e}$$

and

$$p^r kz \cdot p^r \equiv p^r k \pmod{s}$$

we can conclude

$$(d^z)^{p^r} = d$$

because of the Chinese remainder theorem (8.3). □

Hence we can assume in the following that $r < e$ holds. In the proof of Theorem 11.5, an algorithm is described which generalizes a method due to Massey [56] for the computation of square roots modulo a prime number.

Theorem 11.5. *Let $G = \langle g \rangle$ be a cyclic group of order $|G| = p^e s$ with* $\gcd(s, p) = 1$. *For $1 \le r < e$, p^r-th roots can be computed in G in time $O_s(\sqrt{p})$ by a generic algorithm.*

Proof. Let h be a p^r-th power in G. With the Pohlig-Hellman method, together with the baby-step giant-step time-memory trade-off, the discrete logarithm k (modulo p^e) of h can be computed in time $O_s(\sqrt{p})$, where memory space for storing $\sqrt{p}$ group elements is needed. Since h is a p^r-th power in G, k is a multiple of p^r. Let

$$d :\equiv -s^{-1} \pmod{p^r}$$

(note that this inverse exists and can be computed efficiently). Then

$$\left(g^{s \cdot (k/p^r) \cdot d}\right)^{-1} h^{(sd+1)/p^r}$$

is a p^r-th root of h. This algorithm is obviously generic and runs in time $O_s(\sqrt{p})$. □

The algorithm of Theorem 11.5 is, quite surprisingly, optimal among all generic algorithms for computing such roots in cyclic groups. (Interestingly enough, this even holds when we allow the algorithm to make queries to a DH oracle for the group.)

Theorem 11.6. *Let a generic algorithm be given that computes p^r-th ($r \geq 1$) roots in a cyclic group $G = \langle g \rangle$ of order $n = p^e s$, where* $\gcd(s, p) = 1$ *and $e > r$ hold. Assume further that the algorithm is (besides the usual operations) allowed to make calls to a* DH *oracle for G (with respect to some generator g). Then the probability that the algorithm outputs a correct root is at most*

$$\frac{((T+2)(T+1)+2)(\lceil e/r \rceil - 1)}{2p}$$

if T is an upper bound on the number of steps the algorithm performs.

Proof. The proof is related to the proof of Theorem 9.1 in Chapter 9 or of Theorem 5 in [109]. Assume without loss of generality that $n = p^e$, and that an encoding σ of the group elements is chosen randomly. The input to the algorithm is $\sigma(1)$ and $\sigma(p^r x)$. The algorithm can interact with the oracles for addition, inversion, and multiplication (the DH oracle) by computing a list of expressions $\sigma(P_i(x))$ for $i = 1, \ldots, T+2$ where $P_1(X) = 1$, $P_2(X) = p^r X$, and, for $i \geq 3$, $P_i(X) = P_k(X) + P_l(X)$, $P_i(X) = -P_k(X)$, or $P_i(X) = P_k(X) \cdot P_l(X)$, where $k, l < i$. It is not difficult to see by induction that the resulting polynomials have the form

$$P_i(X) = \sum_{m=0}^{d} a_{i,m} (p^r X)^m \ ,$$

where $d := \lceil e/r \rceil - 1$. Note that, although the polynomials can also be multiplied, the degrees are bounded by $d \leq e/r$.

Let the "collision event" $\mathcal{E}$ be defined as in the proof of Theorem 9.1, and let $\overline{\mathcal{E}(\delta)}$ be the complementary event. From Lemma 9.2 we conclude

$$\mathrm{P}[\mathcal{E}] \leq \frac{(T+2)(T+1)d}{2p} \ .$$

If $\overline{\mathcal{E}(\delta)}$ occurs, the best the algorithm can do at the end is to output $\sigma(P_i(x))$ for some $1 \leq i \leq T+2$. The probability that this answer is correct, i.e., that $p^r P_i(x) - p^r x = 0$, given $\overline{\mathcal{E}(\delta)}$, is at most $d/(p \cdot \mathrm{P}[\overline{\mathcal{E}(\delta)}])$, according to Lemma 9.2. Hence the probability that the algorithm answers correctly is at most

$$\mathrm{P}[\mathcal{E}] + \mathrm{P}[\overline{\mathcal{E}(\delta)}] \cdot \frac{d}{p \cdot \mathrm{P}[\overline{\mathcal{E}(\delta)}]} \leq \frac{(T+2)(T+1)d + 2d}{2p} \ ,$$

and this concludes the proof. □

Definition 11.2. We denote by

$$p^r\text{-ROOTP}(n)$$

the problem of computing p^r-th roots in groups of order n. ○

Corollary 11.7. *We have*

$$\mathrm{GC}(p^r\text{-ROOTP}(n)) = \Theta_s(\sqrt{p})$$

if p^{r+1} divides n, and

$$\mathrm{GC}(p^r\text{-ROOTP}(n)) = (\log n)^{O(1)}$$

otherwise.

Proof. The statements are consequences of Theorems 11.5 and 11.6, and of Theorem 11.4, respectively.

11.2.2 The DHP in Subgroups and Changing Generators

Theorem 11.8 implies a characterization of when the DHP in a subgroup of G can be reduced to the same problem in G efficiently. An additional consequence is that the DHP is equally hard with respect to any generator of G.

Theorem 11.8. *Let G be a cyclic group with generator g. If the number r is such that every prime factor of r is either of order $(\log |G|)^{O(1)}$ or has at least the same multiplicity in r as in $|G|$, then there exists a generic algorithm solving the* DHP *in the group $\langle g^r \rangle$, making one call to the* DH *oracle for $\langle g \rangle$ and using a polynomial number of group operations per call.*

Proof. Let $|G| = \prod p_i^{e_i}$ and $r = \prod p_i^{f_i}$ (where $f_i > e_i$, $e_i = 0$, or $f_i = 0$ is possible). The DH algorithm for the group $\langle g^r \rangle$ takes as inputs two elements $(g^r)^a$ and $(g^r)^b$ and must output $(g^r)^{ab}$. Using the DH oracle for the group $G = \langle g \rangle$ with the same input, one obtains g^{r^2ab}, i.e., the r-th power of g^{rab}. Now, g^{rab} is computed from g^{r^2ab} by computing the r-th root. More precisely, the $p_i^{f_i}$-th root of g^{r^2ab} has to be computed for all i with $f_i > 0$, and the correct root, i.e., the particular root that is a power of g^{rab}, must be determined. Assume that we have already computed

$$g^{p_1^{f_1} \cdots p_{i-1}^{f_{i-1}} p_i^{2f_i} \cdots p_s^{2f_s} ab} = g^{cp_i^{2f_i} ab} =: d_i \ ,$$

where c is explicitly known. We describe the computation of the correct $p_i^{f_i}$-th root of this group element separately for the cases $f_i \geq e_i$ and $p_i \leq B$.

Case 1: $f_i \geq e_i$. We compute z with

$$z :\equiv (p_i^{f_i})^{-1} \pmod{|G|/p_i^{e_i}} \ ,$$

and d_i^z, which is the desired group element. First, we have seen in the proof of Lemma 11.4 that it is a $p_i^{f_i}$-th root of d_i. Additionally, it is the only $p_i^{f_i}$-th root of this element which is a power of $g^{p_i^{f_i}}$ (the $p_i^{e_i} - 1$ additional roots are

$$g^{cp_i^{2f_i} abz + j|G|/p_i^{e_i}}$$

for $j = 1, \ldots, p_i^{e_i} - 1$, and they are not even powers of $g^{p_i^{e_i}}$).

Case 2: $p_i \leq B$ *and* $f_i < e_i$. Here we repeat the following two steps f_i times.

Step 1. Compute the p_i-th roots of the group element.

Step 2. Decide which of the roots is a power of g^{rab} and continue with this element.

Assume for some $k = 2f_i - 1, 2f_i - 2, \ldots, f_i$ that we have already computed

$$g^{p_1^{f_1} \cdots p_{i-1}^{f_{i-1}} \cdot p_i^{k+1} \cdot p_{i+1}^{2f_{i+1}} \cdots p_s^{2f_s} ab} = g^{c' p_i^{k+1} ab} \ ,$$

where c' is explicitly known. Then the two steps work as follows.

Step 1. According to Theorem 11.5 we can compute a p_i-th root of the group element in time $O((\log |G|)^2 + p_i \log |G|)$.

Step 2. Because a and b can be obtained modulo $p_i^{e_i - f_i}$ directly from g^{ra}

and g^{rb} by the Pohlig-Hellman algorithm and c' is explicitly known, and because $k \geq f_i$, we can compute $c'p_i^k ab$ modulo $p_i^{e_i}$. The root obtained in Step 1 equals

$$g^{c' p_i^k ab + j \cdot |G|/p_i}$$

for some $0 \leq j < p_i$ which can be computed by the Pohlig-Hellman algorithm because for $j' \neq j$, we have

$$j \cdot |G|/p_i \not\equiv j' \cdot |G|/p_i \pmod{p_i^{e_i}} .$$

This leads to the correct group element $g^{c' p_i^k ab}$.

The entire procedure, executed for all prime factors p_i of r, ends up with g^{rab}, and the running time of the algorithm is polynomial in $\log |G|$. □

Remark. It has been pointed out in a preliminary version of [10] that in case of a generator change, i.e., if $(r, |G|) = 1$, it is not even necessary to know r. Let $h = g^r$, and let DH_g and DH_h be the DH functions in G with respect to the generator g and h, respectively. Then

$$\begin{aligned} \mathrm{DH}_h(h^a, h^b) &= h^{ab} = g^{rab} = \mathrm{DH}_g(g^{r^2 ab}, g^{r^{-1}}) \\ &= \mathrm{DH}_g(\mathrm{DH}_g(h^a, h^b), \mathrm{PDH}_{g,\varphi(|G|)-1}(h)) , \end{aligned}$$

and the last expression can be computed by $O(\log |G|)$ applications of the oracle with respect to the basis g.

It follows from a result by Shoup [109] that also the bound of Theorem 11.8 is tight. This leads to the following completeness result.

Definition 11.3. For groups of order n, let

$$\langle g^\ell \rangle\text{-DHP}(n)$$

be the DHP with respect to the subgroup $\langle g^\ell \rangle$ of G. ○

Corollary 11.9. *We have for integers n and ℓ that*

$$\mathrm{GC}(\langle g^\ell \rangle\text{-DHP}(n) \downarrow \mathrm{DHP}(n)) = \Theta_s(\sqrt{p}) ,$$

where p is the largest common prime factor of n and ℓ such that the multiplicity of p in n is greater than in ℓ (and 1 if no such prime factor exists).

Let us finally study the inverse problem: Does a DH oracle for a subgroup of G, or a set of such oracles, help solving the DHP in G ? Theorem 11.10 provides a criterion.

Theorem 11.10. *Let G be a cyclic group with generator g and order $|G| = \prod_{i=1}^{r} p_i^{e_i}$. If for all p_i not of order $(\log|G|)^{O(1)}$ a number s_i, not divisible by p_i, and a* DH *oracle for the group $\langle g^{s_i} \rangle$ are given, then there exists a polynomial-time uniform generic algorithm solving the* DHP *in G with respect to g which calls each oracle for such a subgroup once.*

Proof. Let g^u and g^v be given. We compute g^{uv} using the available oracles. Let $m_i := p_i^{e_i}$, $M_i := |G|/m_i$, and $N_i :\equiv M_i^{-1} \pmod{m_i}$. For prime factors $p_i = (\log|G|)^{O(1)}$, u and v, and hence also uv, can be computed in polynomial time modulo m_i by the Pohlig-Hellman algorithm. For a prime factor p_i not of order $(\log|G|)^{O(1)}$, assume that a DH oracle for the subgroup $\langle g^{s_i} \rangle$ is given, where p_i does not divide s_i. We apply the oracle for $\langle g^{s_i} \rangle$ to $(g^{s_i})^u = (g^u)^{s_i}$ and $(g^{s_i})^v$ to obtain $(g^{s_i})^{u \cdot v}$, where u, v and $u \cdot v$ are modulo $|G|/s_i$. Let $z_i :\equiv s_i^{-1} \pmod{m_i}$ and

$$U_i := \left(g^{s_i(u \cdot v)}\right)^{M_i \cdot z_i} = g^{M_i \cdot (u \cdot v)} ,$$

where $u \cdot v$ is modulo m_i. Finally, g^{uv} is computable by Chinese remaindering with implicitly-represented arguments by applying only group operations in G:

$$g^{uv} = g^{\sum_i M_i N_i (u \cdot v)} = \prod_i U_i^{N_i} .$$

□

Note that the condition given in Theorem 11.10 is sufficient but *not* necessary. For instance, the ability of solving the DHP in the *minimal* nontrivial subgroups $\langle g^{|G|/p_i} \rangle$ of G can be sufficient even for computing discrete logarithms in G efficiently, as shown in Section 10.2. However, the reduction in the proof of Theorem 11.10 is uniform, simpler, and more efficient if applicable. Furthermore, the reduction is not based on any unproven assumption. On the other hand, a condition that is clearly *necessary* for efficiently constructing a DH oracle for G from such oracles for subgroups of G is that for every large prime factor p of $|G|$, at least one DH oracle must be given for a group $\langle g^s \rangle$ such that p divides $|G|/\gcd(s, |G|)$. Otherwise, the complexity of the reduction is of order $\Omega(\sqrt{p})$ in the generic model.

Corollary 11.11 is a consequence of Theorems 11.8 and 11.10. The statement of Corollary 11.11 is very intuitive and not surprising. An analog statement clearly holds for the DLP.

Corollary 11.11. *Let* ℓ *be a* $(\log n)^{O(1)}$*-smooth divisor of* n*. Then*

$$\langle g^{\ell} \rangle\text{-DHP}(n) \approx_g \text{DHP}(n) \ .$$

11.3 The Decisional DHP

As pointed out in Section 8.3.2, a satisfactory security definition for the DH protocol would not only require that the (computational) DHP is hard, but even its *decisional variant*: Given (g^x, g^y, g^z), decide whether $g^z = g^{xy}$ holds, i.e., whether this is a correct DH triple.

Definition 11.4. Let G be a cyclic group with generator g. A DH *decision oracle* (DHD oracle for short) takes as input a triple (g^u, g^v, g^w) of group elements and outputs `yes` if $w \equiv uv \pmod{|G|}$ and `no` otherwise. ○

The given definition of the DHDP (Definition 8.3) requires that for random group elements g^x and g^y and a pair (g^{xy}, g^c), where g^c is a random group element and the pair is in random order, one must identify the element g^{xy} with probability $1/2 + \Omega(1)$, taken over the choice of the input and the algorithm's coin tosses. As pointed out, it is in particular not required that the algorithm is successful with probability greater than $1/2$ for every specific input.

Let $|G| = \prod p_i^{e_i}$ be the factorization of the order of a group G. For some input (g^x, g^y, g^{xy}) and (g^x, g^y, g^c), the group element g^c is *not* the correct DH key if and only if

$$c \not\equiv x \cdot y \pmod{p_i^{e_i}} \tag{11.4}$$

holds for at least one i. This is a consequence of the Chinese remainder theorem. The validity of (11.4) can be verified in time $\Theta_s(\sqrt{p_i})$ by the Pohlig-Hellman algorithm. Hence the complexity of the DHDP for a *particular* input is $\Theta_s(\sqrt{p_{i_0}})$, where p_{i_0} is the smallest prime factor of $|G|$ for which inequality (11.4) holds with respect to this input.

Stadler [112] and Naor and Reingold [88] have described how an instance of the DHDP can be randomized. However, this randomization is such that the inequality (11.4), for every prime factor p_i of $|G|$, is simultaneously satisfied or violated for the original and the randomized input. Hence one cannot conclude from this randomization that the DHDP is equally hard in the worst and average cases. This conclusion is only true if the group order is a prime power: In this case, solving the DHDP is equivalent to detecting correct DH triples with arbitrarily high probability.

According to Section 8.3.2, in particular the generalization of the technique used in Example 8.1 to arbitrary prime factors of the group order, the complexity of the DHDP is of order $O_s(\sqrt{p'})$, where p' is the smallest prime factor of the group order. This is complemented by Shoup's [109] result that a generic algorithm with running time at most T can solve the DHDP correctly only with probability

$$\frac{1}{2} + O\left(\frac{T^2}{p'}\right) \ .$$

Lemma 11.12. [**109**] *We have*

$$\mathrm{GC}(\mathrm{DHDP}(n)) = \Theta_s\left(\sqrt{p'}\right) \ ,$$

where p' is the smallest prime factor of n.

The complexity given in Lemma 11.12 is different from the corresponding complexity for the DHP (which depends on the *largest* prime factor of n), thus the DHDP and the DHP cannot be generically computationally equivalent for a group whose order contains prime factors of substantially different size.

It should be pointed out that the generic complexity of the *stronger version* of the DHDP(n), requiring that DH triples must be correctly recognized with probability 1, is the same as for the DHP(n), namely of order $\Theta_s(\sqrt{p})$, where p is the *largest* prime factor of n. This is intuitively clear from the discussion above, and can be shown by a combination of the arguments in the proofs of Theorems 9.1 and 11.20. Note that the DHD oracle solves this stronger problem.

We prove that *none* of the versions of the DHDP is computationally equivalent to the DHP in any group (except in groups with smooth orders, for which both problems are generically easy). A similar observation has been independently made by Shoup.

Theorem 11.13 states that a (perfect) DHD oracle is of virtually no help for solving the DHP. This is based on two facts. First, it is hard to generate a query to the DHD oracle for which the oracle's answer is not either trivially `yes`, or `no` with overwhelming probability. Second, given that the oracle answers every nontrivial query by `no`, the success probability of the algorithm is roughly the same as without knowledge of the oracle's answers. Informally speaking, the oracle is of no help because its answers are known in advance.

Note first that

$$\mathrm{DHDP}(\mathbf{N}) \prec_{u,g} \varepsilon\text{-}\mathrm{DHP}(\mathbf{N})$$

holds for every constant $\varepsilon > 0$.

Theorem 11.13. *Let n be a positive integer and let p be the largest prime factor of n. Assume that a generic algorithm is given that works for groups of order n, makes calls to a* DHD *oracle for G and runs in time at most T. Then the probability, taken over the input and the coin tosses of the algorithm, that the algorithm correctly solves the* DHP *is at most*

$$\alpha \leq \frac{(T+3)(T+2)+4}{2p} .$$

Proof. Let $n = p^e s$ with $e \geq 1$ and $\gcd(s,p) = 1$. As in the proof of Theorem 9.1, we can assume $n = p^e$. The generic algorithm takes as inputs $\sigma(1)$, $\sigma(x)$, and $\sigma(y)$ and should compute $\sigma(xy)$. The algorithm is allowed to call, in addition to the usual oracles for addition and inversion, an oracle that solves the DHDP, i.e., that computes the function DHD with

$$\mathrm{DHD}(\sigma(u), \sigma(v), \sigma(w)) = \texttt{yes}$$

if $w \equiv uv \pmod{|G|}$ and $\mathrm{DHD}(\sigma(u), \sigma(v), \sigma(w)) = \texttt{no}$ otherwise. Assume that the algorithm makes A calls to the addition or inversion oracle and B calls to the DHD oracle in a particular execution. Hence we have $A + B \leq T$. By calling the oracles, the algorithm can compute $P_i(x,y)$, $i = 1, \ldots, A+3$, for bivariate polynomials $P_i(X,Y)$ with $P_1(X,Y) = 1$, $P_2(X,Y) = X$, $P_3(X,Y) = Y$, and for $i > 3$ either $P_i(X,Y) = P_k(X,Y) + P_l(X,Y)$ or $P_i(X,Y) = -P_k(X,Y)$ for some $k, l < i$. Clearly, $P_i(X,Y)$ is a linear polynomial for all i. We can assume that all the polynomials are distinct. Furthermore, the algorithm calls the DHD oracle for B input triples $(P_i(x,y), P_j(x,y), P_k(x,y))$. Here, we can assume that none of these polynomials is constant, in particular, that the answer of the DHD oracle is not trivially `yes`.

Let $\mathcal{E}$ be the event that either $P_i(x,y) = P_j(x,y)$ holds for some $i \neq j$, or that the DHD oracle answers `yes` at least once. Observe first that, given $\overline{\mathcal{E}(\delta)}$, everything the algorithm sees is statistically independent from x. Second,

$$\mathrm{P}[\mathcal{E}] \leq \frac{(A+3)(A+2)}{2p} + \frac{2B}{p} \leq \frac{(T+3)(T+2)}{2p} \ . \tag{11.5}$$

The first expression in (11.5) is the number of two-element sets

$$\{i,j\} \subseteq \{1,\dots,A+3\}$$

times the probability that a linear polynomial takes the value 0 for random values of the variables. The second expression on the other hand is B times the probability $2p$ that a relation of the form

$$P_i(x,y) \cdot P_j(x,y) = P_k(x,y)$$

is satisfied for random x and y (i.e., that a certain *quadratic* polynomial takes on the value 0).

Finally, the success probability α of the algorithm satisfies

$$\alpha \leq \mathrm{P}[\mathcal{E}] + \mathrm{P}[\overline{\mathcal{E}(\delta)}] \cdot \frac{2}{p \cdot \mathrm{P}[\overline{\mathcal{E}(\delta)}]} \leq \frac{(T+3)(T+2)+4}{2p} \ .$$

The reason is that, given $\overline{\mathcal{E}(\delta)}$, the best the algorithm can do is output one of the values $P_i(x,y)$. However, $P_i(x,y) = xy$ holds with probability at most $2/(p \cdot \mathrm{P}[\overline{\mathcal{E}(\delta)}])$ over the random choices of x and y. □

Corollary 11.14. *Let n be an integer and p be the largest prime factor of n. Let a generic reduction of the* DHP *to the* DHDP *for groups of order n be given with expected running time T. Then*

$$T \geq \sqrt{p}/2 - 3/2 \ .$$

Proof. Assume as in previous proofs that the execution of the probabilistic algorithm is aborted after $2T$ steps. This new algorithm has running time at most $2T$ and answers correctly with probability at least $1/2$. Hence the result follows from Theorem 11.13. □

Corollary 11.15. *We have for all n that*

$$\mathrm{GC}(\mathrm{DHP}(n) \downarrow \mathrm{DHDP}(n)) = \Theta_s(\sqrt{p}) \ ,$$

where p is the largest prime factor of n.

Corollary 11.15 follows from Theorem 11.13 and from the fact that the DHP cannot be harder with the help of a DHD oracle than without. The conclusion however is that it is not easier neither.

Corollary 11.16. *For numbers n which are not $(\log n)^{O(1)}$-smooth, we have*

$$\mathrm{DHDP}(n) \not\to_g \mathrm{DHP}(n) \ .$$

11.4 The Squaring-DH Problems

In this section we study both the computational and decisional versions of the so-called squaring-DH problem, which is to compute g^{x^2} from g^x. A motivation for the analysis of this variant of the DHP (and DHDP, respectively) is that certain cryptographic systems exist whose security relies on this problem. An example is a scheme for key escrow with limited time span [18], which has been proven secure under the assumption that the decisional squaring-DH problem is hard.

In the context of such schemes, the best case would be to show that the hardness of the computational and decisional variants of the DHP is concentrated in the two (clearly at most as hard) squaring versions. This is true however only for the computational problem. The decisional problem on the other hand is *not generically equivalent* to the DHDP, but is at least shown to be *equally hard* with respect to generic algorithms.

11.4.1 The Computational Problem

Definition 11.5. Let G be a cyclic group with generator g. Then the *squaring*-DHP (SDHP for short) in G is, given g^x, to compute g^{x^2}. The ε-SDHP is, for a group element g^x chosen at random according to the uniform distribution in G, to compute a group element g^z such that $g^z = g^{x^2}$ holds with probability at least ε (taken over the choice of g^x and over the coin tosses of the algorithm). The SDH and ε-SDH *oracles* solve the SDHP and ε-SDHP, respectively. ○

The SDHP is computationally equivalent to the DHP in a uniform generic sense.

Theorem 11.17. *We have*

$$\mathrm{SDHP}(\mathbf{N}) \approx_g \mathrm{DHP}(\mathbf{N}) \ .$$

Proof. First,

$$\mathrm{SDHP}(\mathbf{N}) \prec_{u,g} \mathrm{DHP}(\mathbf{N})$$

clearly holds because the same problem must be solved for an instance out of a subset of the instances of the DHP.

We show the inverse direction. Given g^x and g^y, we compute $g^{x+y} = g^x \cdot g^y$, and with the SDH oracle

$$g^{(x+y)^2} \cdot \left(g^{(x^2)}\right)^{-1} \cdot \left(g^{(y^2)}\right)^{-1} = g^{(x+y)^2 - x^2 - y^2} = g^{2xy} = (g^{xy})^2 \ . \tag{11.6}$$

For given $|G|$, the square root g^{xy} of $(g^{xy})^2$ can efficiently be computed (see Section 11.2.1). If $|G|$ is odd, the square root is unique, but if $|G|$ is even, there exist two square roots,

$$g^{xy} \quad \text{and} \quad g^{xy+|G|/2}$$

which can be computed efficiently. Let $|G|$ be even, and let 2^e be the maximal power of 2 dividing $|G|$. From g^x and g^y, one can compute x and y, and hence xy, modulo 2^e with $O((\log|G|)^2)$ group operations by the Pohlig-Hellman algorithm. Because $|G|/2$ is not a multiple of 2^e, we have

$$xy \not\equiv xy + |G|/2 \pmod{2^e} \ ,$$

and one can determine the correct root g^{xy} by computing the discrete logarithms of one of the roots modulo 2^e. □

Furthermore, an ε-SDH oracle for a group with known order can be transformed into a translation-invariant ε^3-DH oracle by randomizing the inputs in (11.6), using that

$$g^{(u+r)^2} = g^{u^2} \cdot (g^u)^{2r} \cdot g^{r^2} \ .$$

The complexity is $O((\log|G|)^2)$ group operations per call. Hence Theorem 11.18 is a consequence of Theorem 11.2.

Theorem 11.18. *Let $\varepsilon > 0$ and $\beta > 0$ be arbitrary constants. For every cyclic group G with generator g and known order $|G|$ there exists an algorithm solving the $(1-\beta)$-DHP in G, making calls to an ε-SDH oracle for G. The number of oracle calls is*

$$O\left(\frac{\log(1/\beta\varepsilon^3)}{\varepsilon^{12}}\right) .$$

The number of required group operations is $O((\log|G|)^2)$ times the number of oracle calls.

Corollary 11.19 is a consequence of Corollary 11.3, Theorem 11.18, and of the fact that for every $\gamma > 0$,

$$\gamma\text{-SDHP}(\mathbf{N}) \prec_{u,g} \gamma\text{-DHP}(\mathbf{N}) ,$$

which is true because a γ-DH oracle can be made translation-invariant (see Lemma 11.1).

Corollary 11.19. *Let $0 < \alpha_1 < 1$ and $0 < \alpha_2 < 1$ be arbitrary constants. Then*

$$\alpha_1\text{-SDHP}(\mathbf{N}) \approx_g \alpha_2\text{-DHP}(\mathbf{N}) .$$

11.4.2 The Decisional Problem

Definition 11.6. Let G be a cyclic group with generator g. Then the *squaring-DH decision problem* (SDHDP for short) is, given a random group element g^x (chosen according to the uniform distribution in G) and, in random order, g^{x^2} and another random group element g^c, to identify g^{x^2} with probability $1/2 + \Omega(1)$ (taken over the input and the coin tosses of the algorithm). ○

In analogy to the DHDP, a stronger version of this definition could require that pairs of the form (g^x, g^{x^2}) can be recognized with probability 1.

We show that with respect to generic algorithms, the SDHDP is hard if the group order is not divisible by small primes. This implies the security of the scheme of [18] with respect to adversaries restricted to generic attacks.

Theorem 11.20. *Let n be a positive integer and let p' be the smallest prime factor of n. Assume that a generic algorithm exists that works for groups G of order n, runs in time at most T, and outputs a single bit. Let g^x and g^c be random group elements, and let b be a symmetric random bit. The probability that given the input (g^x, g^{x^2}, g^c) if $b = 0$ or (g^x, g^c, g^{x^2}) if $b = 1$, respectively, the algorithm outputs the bit b is at most*

$$\alpha \leq \frac{1}{2} + \frac{(T+4)(T+3)}{2p'} \ .$$

Proof. For given $\sigma(1), \sigma(x)$ and $\{\sigma(x^2), \sigma(c)\}$, all the $T+4$ expressions the algorithm can compute in T steps are the representations of pairwise distinct linear combinations $P_i(1, x, x^2, c)$ of $1, x, x^2$, and c. Note that P_i is a quadratic (or linear) polynomial in x and c. The probability of the collision event

$$\mathcal{E} := \{P_i(x,c) = P_j(x,c) \quad \text{for some} \quad i \neq j\}$$

is at most

$$\mathrm{P}[\mathcal{E}] \leq \frac{(T+4)(T+3)}{p'} \ . \tag{11.7}$$

The reason for (11.7) is that the two (distinct) polynomials must be non-identical modulo p^e for some prime power dividing n. Furthermore, because the projections modulo p^e of uniformly-distributed random elements modulo n are also uniformly distributed modulo p^e, the probability of a collision modulo p^e is $\underline{2/p} \leq 2/p'$.

If the complementary event $\overline{\mathcal{E}(\delta)}$ occurs, then the algorithm clearly obtains no information about the bit b:

$$I(b\,;\,[\sigma(P_i(x,c))]_{i=1,\dots,T+5} \,|\, \overline{\mathcal{E}(\delta)}) = 0 \ .$$

Hence the probability of correctly guessing b is at most

$$\mathrm{P}[\mathcal{E}] + \frac{\mathrm{P}[\overline{\mathcal{E}(\delta)}]}{2} \leq \frac{1}{2} + \frac{(T+4)(T+3)}{2p'} \ .$$

This concludes the proof. □

Corollary 11.21. *Let n be a positive integer and let p' be the smallest prime factor of n. Every generic algorithm solving the* SDHDP *in groups of order n has expected running time T at least*

$$T \geq \sqrt{p'/3} - 4/3 \ .$$

Proof. Construct from the generic algorithm a new algorithm by aborting after $3T$ steps. Then the new algorithm has success probability at least $2/3$. The statement follows by applying Theorem 11.20 to the new algorithm. □

Corollary 11.22 is a consequence of Corollary 11.21, of the upper bound on the DHP of Section 8.3.2, and of Lemma 11.23 below.

Corollary 11.22. *We have for all n that*

$$\mathrm{GC}(\mathrm{SDHDP}(n)) = \Theta_s\left(\sqrt{p'}\right) \ ,$$

where p' is the smallest prime factor of n.

Note that, in analogy to the DHDP, the *stronger* variant of the SDHDP, excluding an error, has complexity of order $\Theta_s(\sqrt{p})$, where p is the *largest* prime factor of n.

Let us now consider the computational relationship between the SDHDP and the DHDP. We first show the apparently trivial fact that the SDHDP is at most as hard as the DHDP. However, due to the fact that the definitions of these decisional problems are such that the success probability is only lower bounded in average, taken over all inputs, this fact is not immediately clear although the set of instances for the SDHDP is a subset of the corresponding set for the DHDP. We show that the random input to the SDHDP can be transformed in such a way that a random instance for the DHDP, with identical correct answer, results.

Lemma 11.23. *We have*

$$\mathrm{SDHDP}(\mathbf{N}) \prec_{u,g} \mathrm{DHDP}(\mathbf{N}) \ .$$

Proof. Let G be a cyclic group with generator g and of order n. We randomize the input (g^x, g^{x^2}, g^c) (where the last two group elements are in random order) to the SDHDP as follows. Choose $r \in_R \mathbf{Z}_{|G|}$, and solve the DHDP for the input

$$\left(g^x\ ,\ g^{x+r}\ ,\ g^{x^2}\cdot(g^x)^r\ ,\ g^c\cdot(g^x)^r\right)\ .$$

Clearly, $g^y := g^{x+r}$ is a random group element and statistically independent of g^x, and one of the last two elements is equal to g^{xy}, whereas the other one is a random group element, statistically independent of (x, y). Hence the SDHDP can be solved with the same success probability as the DHDP in G. □

Let us look at the inverse reduction. Despite the fact that the SDHDP and the DHDP have equal generic complexities, they are *not* computationally equivalent.

Definition 11.7. Let G be a cyclic group with generator g. A *squaring-DH decision oracle* (SDHD oracle for short) takes as input a pair (g^u, g^w) of group elements and outputs `yes` if $w \equiv u^2 \pmod{|G|}$ and `no` otherwise. ○

Theorem 11.24 states that a SDHD oracle is of no help for solving the DHDP. The intuitive reason is the same as when trying to reduce the DHP to the DHDP: The decision oracle is of no help because its answer is known in advance with overwhelming probability (and because the question to be answered cannot be directly solved by the given oracle).

Theorem 11.24. *Let n be a positive integer and let p' be the smallest prime factor of n. Assume that a generic algorithm exists that works for groups G of order n, runs in time at most T, makes calls to a* SDHD *oracle, and outputs a single bit. Let g^x, g^y, and g^c be random group elements, and let b be a symmetric random bit. The probability, taken over the random choice of the input and over the coin tosses of the algorithm, that given the input (g^x, g^y, g^{xy}, g^c) if $b = 0$ or (g^x, g^y, g^c, g^{xy}) if $b = 1$, respectively, the algorithm outputs the bit b is at most*

$$\alpha \leq \frac{1}{2} + \frac{(T+5)(T+4)}{2p'}\ .$$

Proof. Let $\sigma(1), \sigma(x), \sigma(y), \sigma(xy)$, and $\sigma(c)$ be given. By A computation steps, the algorithm can compute the representations of $A + 5$ distinct linear combinations $P_i(1, x, y, xy, c)$ of $1, x, y, xy$, and c, i.e., expressions of the form

$$\sigma(P_i(1, x, y, xy, c)) = \sigma(\ell + \ell_x x + \ell_y y + \ell_{xy} xy + \ell_c c)$$

for constants ℓ and ℓ_J $(J \in \{x, y, xy, c\})$. (Here, it is not "known to the algorithm" which element of the set $\{\ell_{xy}, \ell_c\}$ is ℓ_{xy} and which is ℓ_c.)

Let furthermore the algorithm perform B calls to the SDHD oracle, and let $\mathcal{E}$ be the event that either

$$P_i(1, x, y, xy, c) = P_j(1, x, y, xy, c)$$

holds for some $i \neq j$, or that the oracle answers `yes` at least once (for a nontrivial input). Clearly, we have for P_i and P_j (where $i \neq j$) that

$$P_i \not\equiv P_j^2$$

(where P_i and P_j are now interpreted as polynomials over x, y, and c) unless both polynomials are constant and the input

$$(\sigma(P_i(x, y, c), \sigma(P_j(x, y, c))$$

is trivial. The total degree of the polynomial P_j^2 is at most 4. Hence the probability of $\mathcal{E}$ is upper bounded by

$$\mathrm{P}[\mathcal{E}] \leq \frac{(A+5)(A+4)}{2} \cdot \frac{2}{p'} + B \cdot \frac{4}{p'} \leq \frac{(T+5)(T+4)}{p'} ,$$

where $T \geq A + B$ is the total number of steps of the algorithm. By the same argument as at the end of the proof of Theorem 11.20, we obtain that the success probability of the algorithm is at most

$$\frac{1}{2} + \frac{\mathrm{P}[\mathcal{E}]}{2} \leq \frac{1}{2} + \frac{(T+5)(T+4)}{2p'} .$$

□

Corollary 11.25. *For every number n we have*

$$\mathrm{GC}(\mathrm{DHDP}(n) \downarrow \mathrm{SDHDP}(n)) = \Theta\left(\sqrt{p'}\right)$$

if p' is the smallest prime factor of n.

Finally, it is not surprising that, given that the computational and decisional variants of the DHP are not computationally equivalent, that the same also holds for the SDHP. Indeed, this even follows directly from (the proof of) Theorem 11.13 because a DHD oracle is strictly stronger than a SDHD oracle, and because of

$$\mathrm{SDHP}(\mathbf{N}) \approx_g \mathrm{DHP}(\mathbf{N}) \ .$$

Corollary 11.26. *Let $\mathcal{M} \subset \mathbf{N}$ be the set of numbers which are not $(\log n)^{O(1)}$-smooth. Then*

$$\mathrm{SDHDP}(\mathcal{M}) \not\succeq_g \mathrm{SDHP}(\mathcal{M}) \ .$$

More precisely, we have for all numbers n that

$$\mathrm{GC}(\mathrm{SDHP}(n) \downarrow \mathrm{SDHDP}(n)) = \Theta_s(\sqrt{p}) \ ,$$

where p is the largest prime factor of n.

Chapter 12

Conclusion

In Chapters 9, 10, and 11 we have investigated a number of computational problems related to the security of the DH protocol. Examples are the DLP, DHP, or the DHDP. Not much is known about the hardness of these problems in specific groups. However, with respect to general-purpose algorithms working in any group of a certain order, not exploiting special properties of the representation of the group elements, a complete classification of these problems can be given with respect to their computational hardness and relationship. The underlying computational model is purely probability-theoretic: A group element is represented by a unique random string, and in this representation, the group operation can be carried out easily (e.g., by calling an oracle). The model is very natural and canonical. There exist groups, e.g., non-supersingular elliptic curves over finite fields, for which no faster algorithms than the generic methods are known for solving the DLP or the DHP.

The DLP, as well as the problem of reducing the DLP to the DHP, can be seen as special cases of a general index-search problem, for which a lower bound on the complexity has been shown. The model of the ISP and GISP is similar to the generic model for group algorithms.

12.1 Generic Complexity

The following table summarizes the (uniform) generic complexities of the considered problems. Let $n = \prod p_i^{e_i}$ be the group order and let p and

p' be the largest and smallest prime factors of n, respectively, whereas q denotes the largest *multiple* prime factor of n. The complexities of the DLP, DHP, and DHDP are due to Shoup [109].

Problem	Complexity $\mathrm{GC}_u(\cdot)$
$\mathrm{ISP}(n)$	$n/2$
$(P_1,\ldots,P_l)$-$\mathrm{GISP}(n)$	$\Omega(\sqrt{q})$
$\mathrm{DLP}(n)$	$\Theta_s(\sqrt{p})$
L-$\mathrm{SEDLP}(n)$	$\Omega\left(\sqrt{L/\prod_{p_i<L} p_i^{e_i}}\right)$
P-$\mathrm{GDDLP}(p)$	$\Omega\left(\sqrt{2}^{H_\infty(P)}\right)$
$\tilde{p}^r$-$\mathrm{ROOTP}(n)$ (if $r>0$ and $\tilde{p}^{r+1} \mid n$)	$\Theta_s\left(\sqrt{\tilde{p}}\right)$
$\tilde{p}^r$-$\mathrm{ROOTP}(n)$ (otherwise)	$(\log n)^{O(1)}$
$\mathrm{DHP}(n)$	$\Theta_s(\sqrt{p})$
$\mathrm{DHDP}(n)$	$\Theta_s\left(\sqrt{p'}\right)$
$\mathrm{SDHP}(n)$	$\Theta_s(\sqrt{p})$
$\mathrm{SDHDP}(n)$	$\Theta_s\left(\sqrt{p'}\right)$

In addition, faulty variants of the DHP and the SDHP were investigated. It was shown that solving the DHP or the SDHP with small but non-negligible probability is computationally equivalent to, and hence equally hard as, solving the problems with arbitrarily high probability.

Furthermore, the oracle complexity of the generic DLP was proved to be roughly equal to

$$\log(|G|/S)\ ,$$

where S is the smooth part of the group order $|G|$. This implies that no side information about a discrete logarithm of any type allows to compute it efficiently unless the quantity of this information (in bits) is almost equal to the length of the discrete logarithm (minus the length of the smooth part of the group order).

12.2 Generic Reducibility

The computational relationship between the above-mentioned problems was extensively studied in the generic model. In contrast to the fact that the generic complexity of a problem does not lead to a lower bound on the complexity of the same problem in a specific group, the existence of an efficient generic reduction between two problems guarantees a reduction that is at least as efficient in every particular group.

The following table contains the generic reducibilities between the DLP and the computational and decisional variants of the DHP and of the SDHP. The running times are listed in $\Theta_s(\cdot)$-notation. All the complexities are meant in a *uniform* sense except the reduction of the DLP to the DHPand to the SDHP. Furthermore, the stated running time of these two particular reductions depends on a plausible yet unproven number-theoretic conjecture. As above, n is the group order, of which p, q, and p' are the largest, largest multiple, and smallest prime factors.

Reducing the / to the	DLP	DHP	SDHP	DHDP	SDHDP
DLP	1	1	1	1	1
DHP	$\sqrt{q}$	1	1	1	1
SDHP	$\sqrt{q}$	1	1	1	1
DHDP	$\sqrt{p}$	$\sqrt{p}$	$\sqrt{p}$	1	1
SDHDP	$\sqrt{p}$	$\sqrt{p}$	$\sqrt{p}$	$\sqrt{p'}$	1

In addition, the relationship between the DHP in $G = \langle g \rangle$ and in a subgroup $\langle g^\ell \rangle$ of G was studied. We have that

$$\mathrm{GC}(\langle g^\ell \rangle\text{-DHP}(n) \downarrow \mathrm{DHP}(n)) = \mathrm{GC}_u(\langle g^\ell \rangle\text{-DHP}(n) \downarrow \mathrm{DHP}(n)) = \Theta_s\left(\sqrt{\tilde{p}}\right) ,$$

and

$$\mathrm{GC}(\mathrm{DHP}(n) \downarrow \langle g^\ell \rangle\text{-DHP}(n)) = \Omega\left(\sqrt{\tilde{p}'}\right) ,$$

where $\tilde{p}$ and $\tilde{p}'$ are the largest common prime factors of n and ℓ whose multiplicity in n is greater ($\tilde{p}$) and not greater ($\tilde{p}'$) than in ℓ, respectively. Given that the smoothness assumption 1 is true, we even have

$$\mathrm{GC}(\mathrm{DHP}(n) \downarrow \langle g^\ell \rangle\text{-DHP}(n)) = \Theta_s\left(\sqrt{\tilde{p}'}\right) ,$$

as can be shown by a straight-forward generalization of the proof method of Theorem 11.10.

12.3 Generic Equivalence

An important conclusion is the following complete characterization of the computational equivalence of the main problems considered. Note first that

$$\mathrm{DLP}(\mathbf{N}) \succ_{u,g} \mathrm{DHP}(\mathbf{N}) \succ_{u,g} \mathrm{SDHP}(\mathbf{N}) \succ_{u,g} \mathrm{DHDP}(\mathbf{N}) \succ_{u,g} \mathrm{SDHDP}(\mathbf{N})$$

holds. Consider now the partition of $\mathbf{N}$ into five sets,

$$\mathbf{N} = \mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3 \cup \mathcal{M}_4 \cup \mathcal{M}_5 \ ,$$

which are defined as follows.

$\mathcal{M}_1$ is the set of $(\log n)^{O(1)}$-smooth numbers n.

$\mathcal{M}_2$ is the set of numbers n that are not $(\log n)^{O(1)}$-smooth, but for which both the smallest prime factor and the largest multiple prime factor are of order $(\log n)^{O(1)}$.

$\mathcal{M}_3$ consists of the numbers n which contain no prime factor of order $(\log n)^{O(1)}$, and no multiple prime factor.

$\mathcal{M}_4$ is the set of numbers n which contain a prime factor of order $(\log n)^{O(1)}$ and a multiple prime factor not of this order.

$\mathcal{M}_5$ finally contains the numbers n for which all the prime factors are not of order $(\log n)^{O(1)}$, and at least one of them is multiple.

The generic computational relationship between the considered problems with respect to these sets of orders is as follows.

	DLP	$\diamond_g$	DHP	$\diamond_g$	SDHP	$\diamond_g$	DHDP	$\diamond_g$	SDHDP
$\mathcal{M}_1$		$\approx_g$		$\approx_g$		$\approx_g$		$\approx_g$	
$\mathcal{M}_2$		$\sim_g$		$\approx_g$		$\not\sim_g$		$\approx_g$	
$\mathcal{M}_3$		$\sim_g$		$\approx_g$		$\not\sim_g$		$\not\sim_g$	
$\mathcal{M}_4$		$\not\sim_g$		$\approx_g$		$\not\sim_g$		$\approx_g$	
$\mathcal{M}_5$		$\not\sim_g$		$\approx_g$		$\not\sim_g$		$\not\sim_g$	

Part IV

Concluding Remarks

Chapter 13

Conclusions, Open Problems, and Discussion

13.1 Conclusions

We summarize the most important technical conclusions of this thesis.

- The security definitions in all the considered information-theoretic secret-key agreement scenarios can be significantly strengthened without influence on the achievable rates.
- The intrinsic conditional information measure characterizes the possibility of secret-key agreement against *passive* adversaries in many cases.
- Simulatability, which characterizes the possibility of secret-key agreement against *active* adversaries, is easily characterizable in terms of the probability distribution, and can be checked efficiently.
- Simulatability of a random variable implies the impossibility of one-way secret-key agreement into one particular direction. On top of that, however, these properties of distributions are independent.
- Interaction is more powerful than one-way transmission in the context of message authentication with a partially-secret key.
- Privacy amplification is often possible even if the adversary is active.

- Unconditional security is not inherently impractical.
- Breaking the Diffie-Hellman protocol is computationally equivalent in a generic sense to computing discrete logarithms only for groups whose orders are free of squares of large primes.
- The Diffie-Hellman decision problem is never equivalent to the Diffie-Hellman problem in a generic sense.
- In the generic computational model, the lower and upper bounds on the hardness and reducibility of many computational problems in cyclic groups nearly match.
- In cyclic groups whose orders are divisible by the square of a large prime factor, there exists a hierarchy of non-equivalent problems related to the Diffie-Hellman protocol.
- Although practical computational security cannot be unconditional up to date, substantial statements concerning the hardness and relationship of many problems in groups can be proven which have implications on the security of the Diffie-Hellman protocol.

13.2 Open Problems

These are some examples of open questions left. They provide motivation for further research in the field.

- To prove or disprove Conjecture 5.1.
- If Conjecture 5.1 is true, is it even the case that

  $$S(X;Y||Z) = I(X;Y{\downarrow}Z)$$

  holds for all distributions P_{XYZ}?
- How can simulatability be easily characterized if the reference random variable is ternary (or n-ary for $n \geq 3$)?
- Can privacy amplification be secure against active adversaries under the same condition as against passive adversaries, and can the resulting key be equally long?
- To determine the oracle complexity of the discrete-logarithm problem in specific classes of groups, such as the groups $\mathbf{F}_p^*$.

- To prove or disprove smoothness assumption 1.
- To show that the Diffie-Hellman problem is uniformly generically equivalent to the discrete-logarithm problem only if factoring is easy.
- Are there groups in which the Diffie-Hellman problem is hard?

13.3 Discussion

We have analyzed methods for secret-key agreement by insecure communication. The security analysis of such schemes can be based on information theory (probability theory) or on complexity theory, which are both of paramount importance in modern cryptography. An alternative paradigm, closely related to the methods of information theory, is to exploit the laws of quantum physics. Computationally-secure cryptographic protocols, e.g., public-key schemes, appear to be more practical, but suffer from the lack of unconditionally-valid proofs of security. The reason for this is the absence of *general* lower bounds in complexity theory. Hence "provable" security in this context is either restricted to special computational models, such as the generic model, or is based on a set of unproven standard assumptions.

A rigorous security analysis becomes more and more important in cryptography. This is on one hand motivated by many success stories in breaking (widely-used) cryptographic schemes, but also by the general scientific demand to make statements which are based on firm grounds. We strongly believe that the validation of cryptographic algorithms will be an increasingly important task in the future.

The destiny of public-key cryptography depends closely on the future progress in the development of efficient algorithms, e.g., for factoring integers, or in the construction of computing devices exploiting quantum-mechanic effects. Such computers were shown to be able to efficiently solve both the discrete-logarithm problem and the integer-factoring problem, on which the security of all relevant public-key-cryptographic schemes is based. This is a strong justification of the study of alternative paradigms, e.g., information-theoretic security.

Cryptography is concerned with problems of dramatically-increasing importance considering the growing possibilities of information and communication technology, which has a great influence on our technical and economic world. Since its future depends so directly on the achievements in various fields, such as complexity theory, information theory, quantum physics, or computer engineering, it is definitely wise to be prepared for all possible developments in these areas.

Bibliography

[1] L. M. Adleman and M. A. Huang, "Primality testing and abelian varieties over finite fields," *Lecture Notes in Mathematics*, vol. 1512, Springer-Verlag, 1992.

[2] R. Ahlswede and I. Csiszár, "Common randomness in information theory and cryptography - Part I: secret sharing," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1121–1132, 1993.

[3] E. Bach and J. Shallit, "Factoring with cyclotomic polynomials," *Math. Comp.*, vol. 52, pp. 201–219, 1989.

[4] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental quantum cryptography," *Journal of Cryptology*, vol. 5, no. 1, pp. 3–28, Springer-Verlag, 1992.

[5] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer, "Generalized privacy amplification," *IEEE Transactions on Information Theory*, vol. 41, no. 6, pp. 1915–1923, 1995.

[6] C. H. Bennett, G. Brassard, and J.-M. Robert, "Privacy amplification by public discussion," *SIAM Journal on Computing*, vol. 17, pp. 210–229, 1988.

[7] R. E. Blahut, *Principles and practice of information theory*, Addison-Wesley Publishing Company, 1988.

[8] B. den Boer, "Diffie-Hellman is as strong as discrete log for certain primes," *Advances in Cryptology - CRYPTO '88*, Lecture Notes in Computer Science, vol. 403, pp. 530–539, Springer-Verlag, 1989.

[9] D. Boneh, *Studies in computational number theory with applications to cryptography*, Ph. D. Thesis, Princeton University, Nov. 1996.

[10] D. Boneh and R. J. Lipton, "Algorithms for black-box fields and their application to cryptography," *Advances in Cryptology - CRYPTO '96*, Lecture Notes in Computer Science, vol. 1109, pp. 283–297, Springer-Verlag, 1996.

[11] D. Boneh and R. Venkatesan, "Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes," *Advances in Cryptology - CRYPTO '96*, Lecture Notes in Computer Science, vol. 1109, pp. 129–142, Springer-Verlag, 1996.

[12] D. Boneh and R. Venkatesan, "Breaking RSA may not be equivalent to factoring," *Advances in Cryptology - EUROCRYPT '98*, Lecture Notes in Computer Science, vol. 1403, pp. 59–71, Springer-Verlag, 1998.

[13] S. Brands, "An efficient off-line electronic cash system based on the representation problem," Tech. Rep. CS-R9323, CWI, Amsterdam, 1993.

[14] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," *Advances in Cryptology - EUROCRYPT '93*, Lecture Notes in Computer Science, vol. 765, pp. 410–423, Springer-Verlag, 1994.

[15] W. Brunner, C. Cachin, U. M. Maurer, and C. Vonäsch, *Demonstration system of secret-key agreement by public discussion*, ETH Zürich, 1996. Accessible at http://www.inf.ethz.ch/department/TI/um/keydemo/ .

[16] J. Buchmann and V. Müller, "Computing the number of points of elliptic curves over finite fields," *Proc. ISSAC '91*, pp. 179–182, ACM press, 1991.

[17] J. Buchmann and H. C. Williams, "A key-exchange system based on imaginary quadratic fields," *Journal of Cryptology*, vol. 1, no. 2, pp. 107–118, 1988.

[18] M. Burmester, Y. Desmedt, and J. Seberry, "Equitable key escrow with limited time span (or, how to enforce time expiration cryptographically)," *Advances in Cryptology - ASIACRYPT '98*, Lecture Notes in Computer Science, vol. 1514, pp. 380–391, Springer-Verlag, 1998.

[19] C. Cachin, *Entropy measures and unconditional security in cryptography*, Ph. D. Thesis, ETH Zürich, Hartung-Gorre Verlag, Konstanz, 1997.

[20] C. Cachin, "Smooth entropy and Rényi entropy," *Advances in Cryptology - EUROCRYPT '97*, Lecture Notes in Computer Science, vol. 1233, pp. 193–208, Springer-Verlag, 1997.

[21] C. Cachin and U. M. Maurer, "Unconditional security against memory-bounded adversaries," *Advances in Cryptology - CRYPTO '97*, Lecture Notes in Computer Science, vol. 1294, pp. 292–306, Springer-Verlag, 1997.

[22] C. Cachin and U. M. Maurer, "Linking information reconciliation and privacy amplification," *Journal of Cryptology*, vol. 10, no. 2, pp. 97–110, Springer-Verlag, 1997.

[23] R. Canetti, "Towards realizing random oracles: hash functions that hide all partial information," *Advances in Cryptology - CRYPTO '97*, Lecture Notes in Computer Science, vol. 1294, pp. 455–469, Springer-Verlag, 1997.

[24] E. R. Canfield, P. Erdös, and C. Pomerance, "On a problem of Oppenheim concerning *Factorisatio Numerorum*," *Journal of Number Theory*, vol. 17, pp. 1–28, 1983.

[25] D. G. Cantor, "Computing in the Jacobian of a hyperelliptic curve," *Math. Comp.*, vol. 48, no. 177, pp. 95–101, 1987.

[26] J. L. Carter and M. N. Wegman, "Universal classes of hash functions," *Journal of Computer and System Sciences*, vol. 18, pp. 143–154, 1979.

[27] M. A. Cherepnev, "On the connection between discrete logarithms and the Diffie-Hellman problem," *Discrete Math. Appl.*, 1996.

[28] H. Cohen, *A course in computational algebraic number theory*, Graduate Texts in Mathematics, Springer-Verlag, 1995.

[29] T. M. Cover and J. A. Thomas, *Elements of information theory*, Wiley Series in Telecommunications, 1992.

[30] D. Coppersmith and I. Shparlinsky, "On polynomial approximation and the parallel complexity of the discrete logarithm problem and breaking the Diffie-Hellman cryptosystem," Research Report RC 20724, IBM T. J. Watson Research Centre, 1997.

[31] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," *Advances in Cryptology - CRYPTO '98*, Lecture Notes in Computer Science, vol. 1462, pp. 13–25, Springer-Verlag, 1998.

[32] C. Crépeau, "Efficient cryptographic protocols based on noisy channels," *Advances in Cryptology - EUROCRYPT '97*, Lecture Notes in Computer Science, vol. 1233, pp. 306–317, Springer-Verlag, 1997.

[33] C. Crépeau and J. Kilian, "Achieving oblivious transfer using weakened security assumptions," *29th Symposium on Foundations of Computer Science*, pp. 42–52, IEEE, 1988.

[34] I. Csiszár and J. Körner, "Broadcast channels with confidential messages," *IEEE Transactions on Information Theory*, vol. IT-24, pp. 339–348, 1978.

[35] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[36] M. van Dijk, *Secret key sharing and secret key generation*, Ph. D. Thesis, Technische Universiteit Eindhoven, 1997.

[37] T. El-Gamal, "A public key cryptosystem and a signature scheme based on the discrete logarithm," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[38] W. Feller, *An introduction to probability theory and its applications*, 3rd edition, vol. 1, Wiley International, 1968.

[39] M. J. Fischer and R. N. Wright, "Bounds on secret key exchange using a random deal of cards," *Journal of Cryptology*, vol. 9, no. 2, pp. 71–99, Springer-Verlag, 1996.

[40] M. J. Gander and U. M. Maurer, "On the secret-key rate of binary random variables," *Proc. of the 1994 IEEE Symp. on Information Theory*, p. 351, 1994.

[41] K. O. Geddes, S. R. Czapor, and G. Labhan, *Algorithms for computer algebra*, Kluwer Academic Publisher, 1992.

[42] P. Gemmell and M. Naor, "Codes for interactive authentication," *Advances in Cryptology - CRYPTO '93*, Lecture Notes in Computer Science, vol. 773, pp. 355–367, Springer-Verlag, 1993.

[43] N. Gisin and S. Wolf, "Quantum cryptography on noisy channels: quantum versus classical key agreement protocols," submitted to *Phys. Rev. Lett.*, 1999.

[44] S. Goldwasser and J. Kilian, "Almost all primes can be quickly certified," *Proc. of the 18th Annual ACM Symposium on the Theory of Computing*, pp. 316–329, 1986.

[45] G. H. Hardy and E. M. Wright, *An introduction to the theory of numbers*, University Press, Oxford, 1979.

[46] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby, "Construction of a pseudo-random generator from any one-way function," ICSI Tech. Rep. 91–068, 1991. To appear in *SIAM Journal on Computing.*

[47] K. Ireland and M. Rosen, *A classical introduction to modern number theory*, Springer-Verlag, 1982.

[48] N. Koblitz, "Hyperelliptic cryptosystems," *Journal of Cryptology*, vol. 1, pp. 139–150, 1989.

[49] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comp.*, vol. 48, pp. 203–209, 1987.

[50] S. Lang, *Algebra*, Addison-Wesley Publ. Comp., 1984.

[51] G.-J. Lay and H. G. Zimmer, "Constructing elliptic curves with given group order over large finite fields," *Proc. of ANTS-I*, Lecture Notes in Computer Science, vol. 877, pp. 250–263, Springer-Verlag, 1994.

[52] A. Lenstra, H. W. Lenstra, Jr., and L. Lovasz, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, pp. 515–534, 1982.

[53] H. W. Lenstra, Jr., "Factoring integers with elliptic curves," *Annals of Mathematics*, vol. 126, pp. 649–673, 1987.

[54] H. W. Lenstra, Jr., J. Pila, and C. Pomerance, "A hyperelliptic smoothness test. I," *Philosophical Transactions of the Royal Society*, Series A, vol. 345, no. 1676, pp. 397–408, London, 1993.

[55] R. Lidl and H. Niederreiter, *Introduction to finite fields and their application*, Cambridge University Press, 1986.

[56] J. L. Massey, Advanced Technology Seminars Short Course Notes, pp. 6.66–6.68, Zürich, 1993.

[57] J. L. Massey, "Contemporary cryptology – an Introduction," *Contemporary cryptology – the science of information integrity*, G. J. Simmons (Ed.), IEEE Press, 1992.

[58] J. L. Massey, "A simplified treatment of Wyner's wire-tap channel," *Proceedings of the 21st Annual Allerton Conference of Communication, Control, and Computing*, Monticello, IL, pp. 268–276, 1983.

[59] U. M. Maurer, "Information-theoretically secure secret-key agreement by NOT authenticated public discussion," *Advances in Cryptology - EUROCRYPT '97*, Lecture Notes in Computer Science, vol. 1233, pp. 209–225, Springer-Verlag, 1997.

[60] U. M. Maurer, "A unified and generalized treatment of authentication theory," *Proceedings 13th Symp. on Theoretical Aspects of Computer Science - STACS '96*, Lecture Notes in Computer Science, vol. 1046, pp. 387–398, Springer-Verlag, 1996.

[61] U. M. Maurer, "The role of information theory in cryptography," *Codes and Ciphers*: 4th IMA Conference on Cryptography and Coding, Cirencester, UK, Dec. 1993, pp. 49–71, Southend-on-Sea, 1995.

[62] U. M. Maurer, "Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms," *Advances*

in Cryptology - CRYPTO '94, Lecture Notes in Computer Science, vol. 839, pp. 271–281, Springer-Verlag, 1994.

[63] U. M. Maurer, "The strong secret key rate of discrete random triples," *Communication and Cryptography – Two Sides of One Tapestry*, Kluwer Academic Publishers, pp. 271–285, 1994.

[64] U. M. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, 1993.

[65] U. M. Maurer, "Protocols for secret key agreement based on common information," *Advances in Cryptology - CRYPTO '92*, Lecture Notes in Computer Science, vol. 740, pp. 461–470, Springer-Verlag, 1993.

[66] U. M. Maurer, "Conditionally-perfect secrecy and a provably-secure randomized cipher," *Journal of Cryptology*, vol. 5, pp. 53–66, Springer-Verlag, 1992.

[67] U. M. Maurer, "On the oracle complexity of factoring integers," *Computational Complexity*, vol. 4, pp. 237–247, Birkhäuser Verlag, 1995.

[68] U. M. Maurer, *Provable security in cryptography*, Ph. D. thesis, ETH Zürich, Zürich, 1990.

[69] U. M. Maurer and S. Wolf, "Unconditionally secure key agreement and the intrinsic conditional information," to appear in *IEEE Transactions on Information Theory*, 1999.

[70] U. M. Maurer and S. Wolf, "The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms," to appear in *SIAM Journal on Computing*, 1999.

[71] U. M. Maurer and S. Wolf, "The Diffie-Hellman protocol," to appear in *Designs, Codes, and Cryptography*, Special Issue Public-Key Cryptography, 1999.

[72] U. M. Maurer and S. Wolf, "Diffie-Hellman, decision Diffie-Hellman, and discrete logarithms," *Proc. of the 1998 IEEE Symp. on Information Theory*, Cambridge, U.S.A., p. 327, 1998.

[73] U. M. Maurer and S. Wolf, "Lower bounds on generic algorithms in groups," *Proceedings of EUROCRYPT '98*, Lecture Notes in Computer Science, vol. 1403, pp. 72–84, Springer-Verlag, 1998.

[74] U. M. Maurer and S. Wolf, "Privacy amplification secure against active adversaries," *Advances in Cryptology - CRYPTO '97*, Lecture Notes in Computer Science, vol. 1294, pp. 307–321, Springer-Verlag, 1997.

[75] U. M. Maurer and S. Wolf, "The intrinsic conditional mutual information and perfect secrecy," *Proc. of the 1997 IEEE Symp. on Information Theory*, Ulm, Germany, p. 88, 1997.

[76] U. M. Maurer and S. Wolf, "Unconditionally secure secret-key agreement and the intrinsic conditional mutual information," Tech. Rep. 268, Department of Computer Science, ETH Zürich, May 1997.

[77] U. M. Maurer and S. Wolf, "Towards characterizing when information-theoretic secret key agreement is possible," *Advances in Cryptology - ASIACRYPT '96*, Lecture Notes in Computer Science, vol. 1163, pp. 196–209, Springer-Verlag, 1996.

[78] U. M. Maurer and S. Wolf, "Diffie-Hellman oracles," *Proceedings of CRYPTO '96*, Lecture Notes in Computer Science, vol. 1109, pp. 268-282, Springer-Verlag, 1996.

[79] U. M. Maurer and S. Wolf, "On the complexity of breaking the Diffie-Hellman protocol," Tech. Rep. 244, Department of Computer Science, ETH Zürich, April 1996.

[80] U. M. Maurer and Y. Yacobi, "Non-interactive public-key cryptography," *Designs, Codes, and Cryptography*, vol. 9, pp. 305–316, 1996.

[81] K. S. McCurley, "A key distribution system equivalent to factoring," *Journal of Cryptology*, vol. 1, no. 2, pp. 95–105, 1988.

[82] K. S. McCurley, "The discrete logarithm problem," in *Cryptology and computational number theory*, C. Pomerance (Ed.), Proc. of Symp. in Applied Math., vol. 42, pp. 49–74, American Mathematical Society, 1990.

[83] A. J. Menezes, T. Okamoto, and S. A. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field," *IEEE Transactions on Information Theory*, vol. 39, pp. 1639–1646, 1993.

[84] A. J. Menezes (Ed.), *Applications of finite fields*, Kluwer Academic Publishers, 1992.

[85] A. J. Menezes, *Elliptic curve public key cryptosystems*, Kluwer Academic Publishers, 1993.

[86] R. C. Merkle, "Secure communications over insecure channels," *Communications of the ACM*, vol. 21, pp. 294–299, 1978.

[87] V. Miller, "Uses of elliptic curves in cryptography," *Advances in Cryptology - CRYPTO '85*, Lecture Notes in Computer Science, vol. 218, pp. 417–426, Springer-Verlag, 1986.

[88] M. Naor and O. Reingold, "Number-theoretic constructions of efficient pseudo-random functions," *Proceedings 30th Annual ACM Symp. on the Theory of Computing*, pp. 131–140, 1998.

[89] M. Naor and A. Shamir, "Visual cryptography," *Advances in Cryptology - CRYPTO '94*, Lecture Notes in Computer Science, vol. 950, pp. 1–12, Springer-Verlag, 1995.

[90] V. I. Nechaev, "Complexity of a determinate algorithm for the discrete logarithm," *Mathematical Notes*, vol. 55, no. 2, pp. 91–101, 1994.

[91] N. Nisan and D. Zuckerman, "Randomness is linear in space," *Journal of Computer and System Sciences*, vol. 52, no. 1, pp. 43–52, 1996.

[92] P. C. van Oorschot and M. Wiener, "On Diffie-Hellman key agreement with short exponents," *Advances in Cryptology - EUROCRYPT '96*, Lecture Notes in Computer Science, vol. 1070, pp. 332–343, Springer-Verlag, 1996.

[93] S. Patel and G. S. Sundaram, "An efficient discrete log pseudo random generator," *Advances in Cryptology - CRYPTO '98*, Lecture Notes in Computer Science, vol. 1462, pp. 304–317, Springer-Verlag, 1998.

[94] R. Peralta, "A simple and fast probabilistic algorithm for computing square roots modulo a prime number," *IEEE Transactions on Information Theory*, vol. 32, no. 6, pp. 846–847, 1986.

[95] S. C. Pohlig and M. E. Hellman, "An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance," *IEEE Transactions on Information Theory*, vol. 24, no. 1, pp. 106–110, 1978.

[96] J. M. Pollard, "Monte-Carlo methods for index computation mod p," *Math. Comp.*, vol. 32, pp. 918–924, 1978.

[97] J. M. Pollard, "Theorems on factorization and primality testing," *Proceedings of the Cambridge Philosophical Society*, vol. 76, pp. 521–528, 1974.

[98] A. Rényi, *A diary on information theory*, Akadémiai Kiadó, Budapest, 1978.

[99] A. Rényi, "On measures of entropy and information," *4th Berkeley Symp. on Math. Statistics and Probability, Berkeley*, vol. 1, pp. 547–561, Univ. of Calif. Press, 1961.

[100] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[101] H. Rück, "A note on elliptic curves over finite fields," *Math. Comp.*, vol. 49, pp. 301–304, 1987.

[102] K. Sakurai and H. Shizuya, "Relationships among the computational powers of breaking discrete log cryptosystems," *Advances in Cryptology - EUROCRYPT '95*, Lecture Notes in Computer Science, vol. 921, pp. 341–355, Springer-Verlag, 1995.

[103] C. P. Schnorr, "Efficient identification and signatures for smart cards," *Advances in Cryptology - CRYPTO '89*, Lecture Notes in Computer Science, vol. 435, pp. 239–252, Springer-Verlag, 1990.

[104] R. Schoof, "Elliptic curves over finite fields and the computation of square roots mod p," *Math. Comp.*, vol. 44, no. 170, pp. 483–494, 1985.

[105] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *Journal of the ACM*, vol. 27, no. 4, pp. 701–717, 1980.

[106] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, pp. 656–715, 1949.

[107] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[108] P. W. Shor, "Algorithms for quantum computation: discrete log and factoring," *Proceedings of the 35th IEEE Symposium on the Foundations of Computer Science (FOCS '94)*, pp. 124–134, 1994.

[109] V. Shoup, "Lower bounds for discrete logarithms and related problems," *Advances in Cryptology - EUROCRYPT '97*, Lecture Notes in Computer Science, vol. 1233, pp. 256–266, Springer-Verlag, 1997.

[110] I. E. Shparlinsky, *Computational problems in finite fields*, Kluwer Academic Publishers, 1992.

[111] G. J. Simmons, "A survey of information authentication," *Proceedings of the IEEE*, vol. 76, pp. 603–620, 1988.

[112] M. Stadler, "Publicly verifiable secret sharing," *Advances in Cryptology - EUROCRYPT '96*, Lecture Notes in Computer Science, vol. 1070, pp. 190–199, Springer-Verlag, 1996.

[113] D. R. Stinson, "Universal hashing and authentication codes," *Advances in Cryptology - CRYPTO '91*, Lecture Notes in Computer Science, vol. 576, pp. 74–85, Springer-Verlag, 1992.

[114] S. P. Vadhan, "Extracting all the randomness from a weakly random source," manuscript, MIT, 1998.

[115] S. A. Vanstone and R. J. Zuccherato, "Elliptic curve cryptosystems using curves of smooth order over the ring $\mathbf{Z}_n$," *IEEE Transactions on Information Theory*, vol. IT-43, pp. 1231–1237, 1997.

[116] G. S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *Journal of the American Institute for Electrical Engineers*, vol. 55, pp. 109–115, 1926.

[117] C. P. Waldvogel and J. L. Massey, "The probability distribution of the Diffie-Hellman key," *Advances in Cryptology - AUSCRYPT '92*, Lecture Notes in Computer Science, vol. 718, pp. 492–504, Springer-Verlag, 1993.

[118] M. N. Wegman and J. L. Carter, "New hash functions and their use in authentication and set equality," *Journal of Computer and System Sciences*, vol. 22, pp. 265-279, 1981.

[119] S. Wolf, "Strong security against active attacks in information-theoretic secret-key agreement," *Advances in Cryptology - ASIACRYPT '98*, Lecture Notes in Computer Science, vol. 1514, pp. 405–419, Springer-Verlag, 1998.

[120] S. Wolf, "Unconditional security in cryptography," to appear in *Proceedings of Summer School in Cryptology and Data Security*, July 1998, Aarhus, Denmark, Lecture Notes in Computer Science, vol. 1561, pp. 217–250, Springer-Verlag, 1999.

[121] S. Wolf, *Diffie-Hellman and discrete logarithms*, Diploma Thesis, ETH Zürich, May 1995.

[122] A. D. Wyner, "The wire-tap channel," *Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, 1975.

[123] R. W. Yeung, "A new outlook on Shannon's information measures," *IEEE Transactions on Information Theory*, vol. 37, no. 3, pp. 466–474, 1991.

Index

Curriculum Vitae

14. Sep. 1969: Born in Schaffhausen, Switzerland.

1976–1983: Primarschule and Sekundarschule in Schaffhausen.

1983–1988: Kantonsschule Schaffhausen: Matura Typus C.

1988–1995: ETH Zürich: Dipl. Math. ETH (with honors).

1995–1999: Teaching and research assistant at ETH Zürich, Institute for Theoretical Computer Science, Information Security and Cryptography Research Group, with Prof. U. Maurer.