Doctoral Thesis

Computational Indistinguishability Amplification

Author(s):
Tessaro, Stefano M.

Publication Date:
2010

Permanent Link:
https://doi.org/10.3929/ethz-a-006183893

Rights / License:
In Copyright - Non-Commercial Use Permitted
Computational Indistinguishability Amplification

A dissertation submitted to

ETH ZURICH

for the degree of
Doctor of Sciences

presented by

Stefano Tessaro
MSc ETH CS, ETH Zurich

born July 4, 1981
citizen of Coldrerio, TI, Switzerland

accepted on the recommendation of

Prof. Dr. Ueli Maurer, examiner
Prof. Dr. Thomas Holenstein, co-examiner
Prof. Dr. Salil Vadhan, co-examiner

2010
Acknowledgments

First of all, I would like to thank my advisor, Ueli Maurer. His door has always been open, not only for research discussions, but also every single time I have been in need of any kind of advice. I have learned a lot from him: Most of the questions addressed within this thesis are motivated by the abstract viewpoint on cryptography Ueli has been advocating over the years, and which has had a substantial impact on my approach to research.

Moreover, I am deeply grateful to both Thomas Holenstein and Salil Vadhan for serving on my committee, for reviewing my thesis, and for their valuable feedback. In particular, Thomas is responsible to a non-negligible extent for my interest in complexity-theoretic cryptography that I developed in an earlier stage of my graduate studies.

Of course, my time at ETH would have been far less pleasant without all the great colleagues with whom I have shared so many discussions about research, life, the universe, and everything (as well as countless foosball matches...): This includes Divesh Aggarwal, Joël Alwen, Georges Baatz, Kfir Barhum, Zuzana Beerliová-Trubíniová, David Bernhard, Sandro Coretti, Chandan Dubey, Matthias Fitzi, Ghislain Fourny, Viktor Gaillard, Peter Gaží, Esther Hänggi, Martin Hirt, Robert König, Robin Künzler, Christoph Lucas, Jun Muramatsu, Krzysztof Pietrzak, Bartosz Przydatek, Dominik Raub, Renato Renner, Junji Shikata, Johan Sjödin, Björn Tackmann, Douglas Wikström, Stefan Wolf, Jürg Wullschleger, Vassilis Zikas, and many others I probably forgot to mention.

I am particularly grateful to Peter for inspiring research discussions that have led to some of the results of this thesis. Special thanks also go to Vassilis for having been a great friend within the group, and, together with Dominik, Johan, and Sandro, for having been my office mate. In addition, I thank our secretary, Beate Bernhard, for her tireless administrative support.

Finally, I am indebted to my parents, Claudio and Marilena, for their continuous support, and to Roska, for her love, and for always being on my side. Without their help and encouragement, I would probably have been far from achieving all of this... Grazie di cuore!

The research in this thesis was carried out under partial support from the Swiss National Science Foundation (SNF).
Abstract

Computational security of cryptographic schemes is always shown under the (unproven) assumption that some underlying primitive is secure. In several important examples, the security of these primitives is defined in terms of computational indistinguishability, the property that two systems, despite possibly being very different, exhibit essentially the same behavior in the eyes of a computationally bounded observer. The most prominent example is a pseudorandom generator (PRG), an efficiently computable function stretching a short random secret seed into a longer string which is computationally indistinguishable from an equally long random string. Further examples are pseudorandom functions (PRFs) and permutations (PRPs), keyed functions and permutations which, under a random secret key, behave as a uniformly chosen function and permutation, respectively. Pseudorandom systems underlie essentially all efficient secret-key cryptographic schemes, and are frequent important components in public-key schemes and cryptographic protocols. Also, it is common to assume that block ciphers, such as the Advanced Encryption Standard (AES), are secure PRPs.

Yet computational indistinguishability is a strong requirement, and continuous progress in the development of cryptanalytic techniques casts some doubt as to whether such assumptions are any longer justified for existing designs. To this end, this dissertation addresses the fundamental question of basing efficient cryptography on primitives satisfying substantially weaker forms of computational indistinguishability. In particular, we refer to computational indistinguishability amplification as the problem of strengthening such primitives, and consider, throughout this work, two main axes along which amplification is achieved.

The first and main part of this thesis addresses the case where the observer is allowed to achieve some non-negligible (albeit quantitatively bounded) advantage over random guessing in distinguishing the two systems. By means of a series of general theorems, we undertake an in-depth investigation of the behavior of the computational distinguishing advantage under different forms of system composition, with the aim of finding efficient combination operations reducing the computational distinguishing advantage. All these results apply to the general class of systems whose state does not depend on the interaction, which comprises most cryptographic systems of interest. Our most important application is an exact characterization of the security amplification achieved by the cascade (i.e., sequential composition) of PRPs with respect to the distin-
guishing advantage, a long-standing open problem. (Even stronger amplification is shown under a minimal modification of the cascade.) Also, we provide a construction for security amplification of weak PRGs with optimal output length, as well as tighter and/or simpler proofs for all existing results in the literature in the context of advantage amplification.

A key technique is the generalization of complexity-theoretic results, such as Yao’s XOR Lemma and Impagliazzo’s Hardcore Lemma, to the setting of interactive systems, which is of independent interest. Also, most of our results can be interpreted as computational analogues of information-theoretic results, and help providing a better understanding of the intrinsic relationship between information-theoretic and computational security.

In contrast, the final part of this thesis is devoted to a weaker form of computational indistinguishability where the observer is only granted restricted access to the given system. In particular, we consider PRFs where computational indistinguishability only holds for observers which are allowed a bounded number (e.g., a constant as low as two) of random (but known) queries. We provide constructions of fully secure PRFs from such weaker PRFs that even improve on the efficiency of existing constructions in the literature based on the stronger assumption where observers are allowed any number of random queries. Our results yield efficient encryption schemes from block ciphers, and efficient message authentication codes from hash functions, both under such very weak pseudorandomness assumptions on the underlying primitives.
Riassunto

Ogni dimostrazione della sicurezza computazionale di uno schema crittografico si basa sull’assunzione, non dimostrata, che una primitiva, impiegata come componente, soddisfi a sua volta dei requisiti di sicurezza computazionale. La sicurezza di tali primitive viene spesso definita tramite il concetto di indistinguibilità computazionale, ossia la proprietà secondo la quale due sistemi crittografici, sebbene sostanzialmente diversi, presentino un comportamento pressoché identico nei confronti di osservatori la cui potenza di calcolo è limitata. L’esempio più comune è rappresentato dai cosiddetti generatori pseudocasuali (PRG), funzioni efficienti il cui output, dato un input segreto distribuito uniformemente, è computazionalmente indistinguibile da una stringa distribuita uniformemente e di lunghezza maggiore dell’input. Ulteriori esempi sono le funzioni pseudocasuali (PRF) e le permutazioni pseudocasuali (PRP), funzioni (e rispettivamente permutazioni), indicizzate da una chiave segreta, che sono indistinguibili da una funzione (o permutazione) scelta uniformemente dall’insieme di funzioni (o permutazioni) con il medesimo dominio. I sistemi pseudocasuali sono alla base di quasi tutti gli schemi efficienti in crittografia simmetrica, come pure di molti schemi nella crittografia a chiave pubblica e in protocolli crittografici. È inoltre comune assumere che i cosiddetti block ciphers, come l’Advanced Encryption Standard (AES), sono delle PRP.

L’indistinguibilità computazionale rimane tuttavia una proprietà di sicurezza molto forte, e i continui progressi nello sviluppo di tecniche crittanalitiche mettono sempre più in dubbio la validità dell’assunzione che l’AES è una PRP. Pertanto, l’obbiettivo primario di questa dissertazione è uno studio dettagliato di soluzioni crittografiche efficienti basate su primitive che soddisfano unicamente forme ben più deboli di indistinguibilità computazionale. In particolar modo, studieremo il problema dell’amplificazione dell’indistinguibilità computazionale (computational indistinguishability amplification), ossia il problema di incrementare la sicurezza di queste primitive.

La prima (e maggiore) parte di questa tesi si propone di studiare il caso dove è permesso all’osservatore, confrontato con il compito di distinguere due sistemi, raggiungere un vantaggio sostanziale (sebbene limitato quantitativamente) rispetto a una semplice scelta casuale. Attraverso una serie di teoremi generali, affronteremo un’analisi quantitativa del comportamento del vantaggio nel caso di diversi tipi di operazioni di composizione di sistemi: l’obbiettivo è di individuare esempi efficienti
di operazioni in grado di ridurre sostanzialmente tale vantaggio. Questi risultati sono applicabili a una classe generale di sistemi con uno stato iniziale arbitrario, che non viene però aggiornato nel corso dell’interazione, la quale comprende la maggior parte dei sistemi crittografici d’interesse. L’applicazione principale di questi risultati è una caratterizzazione esatta dell’amplificazione della sicurezza raggiunta dalla cascata (o composizione sequenziale) di PRP, un problema finora irrisolto. Inoltre, una minima modifica della cascata raggiunge un’amplificazione ottimale. Ulteriori applicazioni consistono in un nuovo metodo per incrementare la sicurezza di PRG deboli con un output di lunghezza ottimale, come pure analisi semplificate di tutti i risultati precedenti nel contesto dell’amplificazione del vantaggio.

Una tecnica chiave nello sviluppo di questi risultati, di interesse indipendente, consiste nella generalizzazione al contesto dei sistemi interattivi di risultati fondamentali come l’XOR Lemma di Yao e l’Hardcore Lemma di Impagliazzo. Inoltre, molti risultati di questa tesi sono interpretabili come analoghi computazionali di teoremi precedentemente dimostrati nel contesto della sicurezza incondizionata (basata sulla teoria dell’informazione), e contribuiscono pertanto a una migliore comprensione della relazione fra i due tipi di sicurezza.

La seconda parte della tesi considera un indebolimento dell’indistinguibilità computazionale attraverso la limitazione dell’accesso. In modo particolare, studieremo funzioni pseudocasuali dove all’osservatore èunicamente permessa la valutazione presso un numero limitato (che può essere addirittura una costante maggiore o uguale a due) di input casuali. Il risultato principale di quest’ultima parte consiste in nuove costruzioni di PRF (per osservatori con valutazioni arbitrarie) da queste varianti di PRF deboli, la cui efficienza è addirittura superiore a quella di precedenti costruzioni richiedenti un’assunzione ben più forte dove il numero di valutazioni a input casuali non è limitato. Due conseguenze dirette sono un nuovo sistema di cifratura basato su block ciphers, come pure un nuovo metodo di autentificazione di messaggi a partire da funzioni di hash, entrambi basati unicamente su tali deboli assunzioni pseudocasuali.
Contents

1 Introduction ........................................ 1
   1.1 Cryptographic Security ......................... 1
   1.2 Cryptographic Assumptions ..................... 2
   1.3 Indistinguishability and Pseudorandomness .... 3
   1.4 Computational Indistinguishability Amplification ... 5
   1.5 Outline and Contributions of This Thesis ....... 6

2 Preliminaries ...................................... 13
   2.1 Basic Notation .................................. 13
   2.2 Probabilities and Information-Theoretic Quantities .... 14
   2.3 Discrete Systems ................................ 17
   2.4 Indistinguishability of Systems and Game Winning .... 24
   2.5 Algorithms and Complexities .................... 31
   2.6 Pseudorandom Systems ............................ 34

3 Weakly Unpredictable Bits .......................... 37
   3.1 System-Bit Pairs .................................. 41
   3.2 The Hardcore Lemma for System-Bit Pairs ......... 44
   3.3 The XOR Lemma for System-Bit Pairs .............. 79

4 Security Amplification for Neutralizing Constructions ... 95
   4.1 Neutralizing Constructions ....................... 97
   4.2 Product Theorem for Neutralizing Constructions .... 99
4.3 Applications of the Product Theorem . . . . . . . . . . . . . 104
4.4 A Strong Product Theorem . . . . . . . . . . . . . . . . . . . 106
4.5 Applications of the Strong Product Theorem . . . . . . . . 114

5 Security Amplification via Extracting Constructions 121
5.1 Hardcore Lemmas for Computational Indistinguishability 124
5.2 Extracting Constructions . . . . . . . . . . . . . . . . . . . . 134
5.3 Security Amplification for PRGs . . . . . . . . . . . . . . . 139
5.4 High-Min-Entropy Function Tables . . . . . . . . . . . . . . 144
5.5 Cascade of Weak Permutations . . . . . . . . . . . . . . . . 149
5.6 Randomized Cascade and Randomized XOR . . . . . . . . . 161

6 From Random-Input to Chosen-Input Indistinguishability 165
6.1 The Randomized Cascade Construction . . . . . . . . . . . 168
6.2 The Nested Randomized Cascade Construction . . . . . . . 178
6.3 Black-Box Keying of Iterated Hash Functions . . . . . . . . 185

7 Concluding Remarks and Open Problems 189

A Additional Technical Material 203
A.1 A Technical Lemma on Conditional Expectations . . . . . 203
A.2 Details of the Proof of Theorem 5.3 . . . . . . . . . . . . . . 204
Chapter 1

Introduction

1.1. Cryptographic Security

The primary goal of cryptography can be broadly summarized as the design of schemes that allow a set of honest entities to securely achieve some pre-specified goals in presence an adversary. The most traditional and ubiquitous example is the private transmission of information between two parties, Alice and Bob, where an adversary, Eve, eavesdrops their communication, this task being usually referred to as message encryption. A further example is message authentication, i.e., the authentic transmission of information in the case where Eve is additionally allowed to tamper with the communication between Alice and Bob.

Despite wide interests in the deployment of such schemes spanning over two thousand years, only the last decades (starting from the seminal work of Shannon [Sha49]) have seen cryptography develop itself into a scientific discipline centered around the principle of provable security. Security of cryptographic schemes is formalized in terms of the inability of an adversary to win a certain game associated with some desired property of the scheme. The ultimate goal is a security proof, a formal proof that no adversary is able to win the game.

A central parameter of such security proofs is the computational power given to the adversary. The strongest notion (which was considered in
Shannon’s work) is called information-theoretic (IT) security and requires security for all adversaries, regardless of their computing power. (These adversaries are usually called computationally unbounded.) While some prominent tasks, such as message encryption and authentication, enjoy solutions with IT security (granted entities share a private key), the resulting schemes are impractical. Even more importantly, a number of additional tasks, such as public-key encryption (which is essential for secure communication over the Internet, and which has been introduced in the seminal work of Diffie and Hellman [DH76]), cannot be realized with information-theoretic security.

To overcome these limitations, the weaker, yet more realistic, notion of computational security allows secure and efficient realization of a wider range of functionalities, but only requires the scheme to resist attacks from so-called computationally bounded adversaries, i.e., restricted to feasible computations: While a computationally secure scheme can, in principle, always be broken, any successful attack would take an excessive amount of time (e.g., thousands or even millions of years).

1.2. Cryptographic Assumptions

Information-theoretic security statements are usually unconditional, that is, their proofs do not rely on any assumptions. In contrast, non-trivial unconditional computational infeasibility proofs remain beyond reach of known techniques,¹ and computational security is always proven under the assumption that some simpler cryptographic component (often called a primitive) is secure or, more generally, that some underlying computational problem is hard. For example, a symmetric encryption scheme may rely on a one-way function \( f : \mathcal{X} \to \mathcal{Y} \), an efficiently computable function which is hard to invert,² or, in the public-key setting, may make use of a group (such as \( \mathbb{Z}_{pq}^* \) for large primes \( p \) and \( q \)) where the computation of \( e \)-th roots is hard (this is the well-known RSA assumption [RSA78]).

Deployment of provably secure cryptographic schemes thus follows a two-step approach: First, one identifies a scheme with provable-security guarantees under a computational assumption on the underlying component, and second, the underlying component is instantiated (e.g., by means of a block cipher, a hash function, or a certain algebraic structure) so that the computational assumption is conjectured to be true.

¹Such proofs usually imply \( \mathbb{P} \neq \mathbb{NP} \).
²More specifically, this means that given \( f(x) \) for a random secret preimage \( x \), it is computationally hard to find a preimage \( x' \) such that \( f(x) = f(x') \).
In general, such conjecture is solely justified by resistance to all currently known attack types, and sudden cryptanalytic breakthroughs may falsify it. Consequently, such assumptions need to be as weak as possible: On the one side, it is plausible that designing a primitive satisfying such weaker assumption is a much easier task. On the other side, it is a safe practice to only assume that already existing primitives only satisfy much weaker security properties than their original design goals.

However, while for example all symmetric cryptographic primitives can be built based on primitives as weak as one-way functions, the resulting constructions are way to inefficient to be any practical use, and the security losses contained in their security proofs are way too large. We thus experience a trade off between the strength of the assumption and the efficiency of the scheme relying on it, and the actual problem, in practice, is to find the best compromise between the strength of the assumption and the desired efficiency of the scheme.

### 1.3. Indistinguishability and Pseudorandomness

Several computational assumptions are formulated in terms of the hardness to compute a certain secret, such as the preimage of a one-way function or the $e$-th root of a group element. Another important class of assumptions (and cryptographic primitives), first considered by Blum and Micali [BM82], and which is the object of this thesis, is based on computational indistinguishability, i.e., the property that two systems, which are possibly completely different, express the same behavior in the eyes of a computationally bounded observer, usually called a distinguisher.

More in detail, for two interactive systems $S$ and $T$, and for a distinguisher $D$ outputting a decision bit $D(F)$ when interacting with some given system $F$, the distinguishing advantage is defined as

$$\Delta^D(S, T) := |P[D(S) = 1] - P[D(T) = 1]|,$$

and, for a set of distinguishers $D$, we define

$$\Delta^D(S, T) := \sup_{D \in D} \Delta^D(S, T).$$

Two systems $S$ and $T$ are computationally indistinguishable if $\Delta^E(S, T)$ is “negligible”, where $E$ is the set of all computationally bounded distinguishers performing feasible computations.

$^3$We provide a formal definition of such systems below, for now an intuitive understanding is sufficient.
A special case of cryptographic primitives relying on computational indistinguishability are pseudorandom systems, i.e., systems which only use few random bits as a seed (e.g., from a short secret key) and which are computationally indistinguishable from an ideal system containing much more randomness than the seed. Examples of relevant ideal systems range from a random string longer than the seed (so-called pseudorandom generators (PRG) [BM82]) to a randomly chosen function (these are referred to as pseudorandom functions (PRFs) [GGM84]) and to a randomly chosen permutation (pseudorandom permutations (PRPs) [LR86]).

Pseudorandom systems underlie essentially all efficient symmetric cryptographic schemes, and are also central components in a wider context. Also, it is a common assumption that a block cipher, such as the Advanced Encryption Standard (AES), is a PRP under a random secret key.

Building pseudorandom systems appears to be a significantly harder task than designing primitives with computational hardness guarantees. While in the latter case it suffices to make sure that, with very high probability, some secret is hard to guess, in the former case no efficient distinguisher must detect any minimal deviation from the behavior of the ideal random system. However, we do expect simpler constructions of pseudorandom primitives if we allow for non-random distribution of their outputs. Despite distinguishing from random being much easier, such primitives may still ensure the availability of sufficient pseudorandomness to be useful in a cryptographic sense.

More concretely, we identify two orthogonal axes along which (computational) indistinguishability can be relaxed:

(i) On the one hand, we can require the advantage $\Delta^\varepsilon(S, T)$ to be only upper bounded by some $\varepsilon \in [0, 1)$ which is not necessarily negligible, and might even be very close to one. In particular, we can consider a weakening of pseudorandom systems where the advantage is upper bounded by $\varepsilon$, and refer to the corresponding notions as $\varepsilon$-PRGs, $\varepsilon$-PRFs, $\varepsilon$-PRPs, etc.

(ii) On the other hand, we can restrict the class of distinguishers, i.e., we only consider the distinguishing advantage $\Delta^{D'}(S, T)$ for $D' \subset \mathcal{E}$. For instance, $D'$ may consist of distinguishers choosing their inputs non-adaptively or even randomly, or may contain distinguishers which are only allowed a very small number of queries, and so on.

---

4For example, PRFs are good message authentication codes and are also essential for all practical encryption schemes.
1.4 Computational Indistinguishability Amplification

In view of continuous progress in block-cipher cryptanalysis, it is for instance completely unclear for how long AES will remain a good PRP. In contrast, it is a well plausible and significantly safer assumption that AES is computational indistinguishable from a random permutation if we restrict ourselves to 10-query distinguishers, or only require the best advantage to be upper bounded by 0.99.

1.4. Computational Indistinguishability Amplification

The primary objective of this thesis is to study the general problem of computational indistinguishability amplification, i.e., the problem of amplifying the security of primitives defined in terms of weaker indistinguishability notions.

This can be rephrased as follows: Assume that, for a system $S$ and a corresponding ideal system $I$, and for some class of distinguishers $D$, $\Delta^D(S, I) \leq \epsilon$.

We seek for efficient constructions $C(\cdot)$ using $S$ (or, alternatively, sufficiently many instances $S_1, \ldots, S_m$ of $S$) such that $\Delta^{D'}(C(S), J) \leq \epsilon'$, for some class of distinguishers $D'$, some $\epsilon'$, and some target ideal systems $J$: Generally, we have $I = J$, or $J$ is of the same type of $I$, but possibly with different input-output sizes. We want at least one of the following two goals to be achieved:

**Advantage amplification.** We want to achieve $\epsilon' < \epsilon$: This is along the same lines as the huge body of previous work on security amplification started by Yao [Yao82] in the context of one-way functions, and subsequently extended to other cryptographic primitives, including regular OWFs and OWPs [GIL+90, HHR06b], two-party protocols (cf. e.g. [BIN97, PV07, PW07, Wul07, HR08, Hai09, HPWP10, CL10]), key-agreement and public-key encryption [DNR04, Hol05, HR05], and collision-resistant hash functions [CRS+07].

**Distinguisher class amplification.** In this case, we want $D' \supseteq D$. For example, we may have $D' = \mathcal{E}$ and let $D$ be the class of distinguishers issuing random (but known) queries. PRF secure only against such

---

5i.e., for how long no attacks breaking this property will not be found
distinguishers will be called random-input PRFs (RI-PRFs) in the following.\(^6\) Alternatively, one may consider the slightly larger class of distinguishers choosing their queries in a non-adaptive fashion.

The next section will provide an overview of the contents and contributions of this thesis.

### 1.5. Outline and Contributions of This Thesis

We outline in detail the contributions of this thesis, which covers the contents of [MT08], [MT09], [MT10], and [Tes10]. An in-depth comparison with previous work is found at the beginning of each chapter, whereas notation and fundamental tools are introduced in Chapter 2.

#### 1.5.1. Advantage Amplification

A major part of this thesis is devoted to the question of advantage amplification. Our presentation adopts an abstract viewpoint to the greatest extent: We provide general system composition theorems which are exercised to obtain new security amplification results for weak primitives such as \(\varepsilon\)-PRGs, \(\varepsilon\)-PRFs, and \(\varepsilon\)-PRPs for arbitrary \(\varepsilon < 1\).

Our composition theorems apply to the general class of systems \(S\) whose input-output behavior can equivalently be realized by a (possibly inefficient) construction \(S(S)\) with an initial state \(S\) such that the answer of every query to \(S(S)\) only depends on the input of the query and on \(S\), but not on previous queries. We refer to such systems as cc-stateless. In particular, all (weak) pseudorandom systems are cc-stateless systems.

**System-Bit Pairs.** As the first special case of computational indistinguishability amplification, Chapter 3 considers the case of pseudorandom *bits*: We introduce the concept of system-bit pairs, that is, pairs consisting of a bit \(B\) and a system \(S\) whose behavior is correlated with \(B\). In particular, we study the guessing advantage \(\text{Guess}^A(B \mid S)\), a normalized measure of the probability of an adversary \(A\) guessing the bit \(B\) when interacting with the system \(S\).\(^7\) Note that the fact that \(\text{Guess}^A(B \mid S)\) is small is equivalent to the fact that \(B\) looks random to \(A\) when given access to \(S\).

\(^6\)This notion was introduced by Naor and Reingold [NR99] under the name “weak PRF”, which we will not use, as the term “weak” will have a broader meaning within this thesis.

\(^7\)Advantage \(\varepsilon\) means that the corresponding guessing probability is \(\frac{1+\varepsilon}{2}\).
1.5 Outline and Contributions of This Thesis

Our first contribution is a characterization of the guessing advantage for the class of (efficiently-implementable) cc-stateless system-bit pairs \((S, B) \equiv (S(S), B(S))\) with initial state \(S\): Informally, we prove that if

\[
\text{Guess}^A(B \mid S) \leq \varepsilon,
\]

for all efficient adversaries \(A\), then there exists an event \(A\), defined on \(S\) and occurring with probability at least \(1 - \varepsilon\), such that for all efficient adversaries \(A'\),

\[
\text{Guess}^{A'}(B(S') \mid S(S')) \leq \nu
\]

for \(S'\) sampled as \(S\) conditioned on \(A\), for all efficient adversaries \(A'\), and for a negligible quantity \(\nu\). This result generalizes Impagliazzo’s Hardcore Lemma (HCL) [Imp95] (and its tight version by Holenstein [Hol05]), which considered the special case where \(S\) is a random variable. The major challenge in the interactive setting stems from the fact that the only efficient implementation of a cc-stateless system-bit pair may be completely stateful, and, in order to use the lemma in many settings, we need to prove that the event \(A\) has the property that \((S(S'), B(S'))\) can be efficiently simulated using any such efficient implementation.

One main implication of this result is a generalization of Yao’s XOR Lemma [Yao82, GNW95] to interactive systems: For \(m\) independent copies \((S_i, B_i) (i = 1, \ldots, m)\) of \((S, B)\) as above, we prove that the XOR \(B_1 \oplus \cdots \oplus B_m\) can be guessed with advantage at most \(\varepsilon^m + \nu\) for all efficient adversaries accessing \(S_1, \ldots, S_m\) in parallel.\(^8\) In addition to a proof based on the HCL, we also provide a direct proof of the XOR Lemma which yields a slightly stronger statement tolerating one stateful system-bit pair.

Neutralizing Constructions. In Chapter 4, we study computational indistinguishability amplification in the full-fledged setting of interactive systems. In particular, we consider constructions abstracting the XOR of random bits: A construction \(C(\cdot, \cdot)\) is called neutralizing for real systems \(S\) and \(T\), and corresponding ideal systems \(I\) and \(J\), if we have

\[
C(I, T) \equiv C(S, J) \equiv C(I, J),
\]

where the equivalence \(\equiv\) indicates that both systems express the same input-output behavior. Such constructions have been introduced by Maurer, Pietrzak, and Renner [MPR07], and the notion extends naturally to

\(^8\)The additive negligible term \(\nu\) is unavoidable and typical of security amplification results.
any number of subsystems. In other words, a neutralizing construction is a combiner with the property that whenever at least one of the subsystems is ideal, the resulting system is ideal, too.

**FIRST PRODUCT THEOREM.** Our first result relies on the XOR Lemma and implies security amplification via any efficiently implementable neutralizing construction $C(\cdot)$ for cc-stateless real systems $S_1, \ldots, S_m$ and cc-stateless ideal systems $I_1, \ldots, I_m$, as long as $\Delta^E(S_i, I_i) < \frac{1}{2}$ holds for all $i = 1, \ldots, m$. More concretely, we prove that

$$\Delta^E(C(S_1, \ldots, S_m), C(I_1, \ldots, I_m)) \leq 2^{m-1} \prod_{i=1}^{m} \Delta^E(S_i, I_i) + \nu$$

for a negligible function $\nu$. We apply this result to two special cases:

- First, we prove security amplification for $\varepsilon$-PRFs and $\varepsilon$-PRGs with $\varepsilon < \frac{1}{2}$ by combining their outputs via a quasi-group operation. The obtained bounds are tight, and improve on previous work by Dodis et al. [DIJK09].

- Second, this result also yields security amplification for the cascade (i.e., sequential composition) of $\varepsilon$-PRPs with $\varepsilon < \frac{1}{2}$. This also holds for two-sided PRPs, where computational indistinguishability is required even against distinguishers issuing inversion queries. This significantly improves on previous work by Luby and Rackoff [LR86] and by Myers [Mye99], which only considered constant-length cascades, and was therefore not sufficient to infer that a sufficiently long cascade yields a fully secure PRP for a non-negligible $\varepsilon$.

**SECOND PRODUCT THEOREM.** In the second part of Chapter 4, we prove that for a subfamily of randomized neutralizing constructions, which satisfy a special property called self-independence, essentially optimal amplification is achieved, i.e., the above bound is improved to

$$\Delta^E(C(S_1, \ldots, S_m), C(I_1, \ldots, I_m)) \leq \prod_{i=1}^{m} \Delta^E(S_i, I_i) + \nu.$$ 

We present some examples where the theorem can be used:

---

9Recall that $\ast$ is a quasi-group operation on $X$ if for all $a, c \in X$ there exists a unique $b \in X$ such that $a \ast b = c$ and a unique $b' \in X$ such that $b' \ast a = c$. 
1.5 Outline and Contributions of This Thesis

- We prove that cascading amplifies the security of (possibly two-sided) \( \varepsilon \)-PRPs for \( \varepsilon < 1 \) in an essentially optimal way if two additional secret random offsets are added at both ends of the cascade.

- In addition, we show that in contrast to the case of \( \varepsilon \)-PRFs, composing \( \varepsilon \)-RI-PRFs by quasi-group operations amplifies security for essentially all \( \varepsilon < 1 \).

- Finally, we prove that adding a secret random offset to the input yields nearly optimal security amplification for \( \varepsilon \)-PRPs via composition with a quasi-group operation. This slightly simplifies a previous result by Myers [Mye99, Mye03].

Extracting Constructions. Even though the above product theorems for neutralizing constructions are very general, some questions still remain open: For example, what are the exact bounds for computational indistinguishability amplification achieved by the (plain) cascade of \( \varepsilon \)-PRPs? And how can we obtain security amplification for \( \varepsilon \)-PRGs in the case where \( \varepsilon \geq \frac{1}{2} \)? Chapter 5 answers these questions.

Our key step uses the HCL for system-bit pairs to prove that for any two cc-stateless systems \( S \equiv S(S) \) and \( T \equiv T(T) \) with

\[
\Delta^\varepsilon (S, T) \leq \varepsilon,
\]

there always exist events \( A \) and \( B \), defined on \( S \) and \( T \), respectively, occurring each with probability at least \( 1 - \varepsilon \), such that for \( S' \) and \( T' \) sampled conditioned on these events,

\[
\Delta^\varepsilon (S(S'), T(T')) \leq \nu
\]

for a negligible function \( \nu \). We refer to this result as the **Hardcore Lemma for computational indistinguishability**.

We also introduce a further class of constructions, called **extracting constructions**, which implement some target ideal system given that sufficiently many subsystems implement a function table with high entropy. We prove a general product theorem for such constructions based on the HCL for computational indistinguishability, and exercise this result with a number of examples:

- We prove a strong product theorem for \( \varepsilon \)-PRG for arbitrary \( \varepsilon < 1 \), which significantly improves on previous work and on the result given in Chapter 4: We show that applying a good randomness
extractor to the concatenation of sufficiently many outputs of an ε-PRG under independent seeds yields a fully secure PRG with optimal output length. This result can be seen as an extension of the well-known extraction technique [HILL99, STV01, Hol06] (which is used in PRG constructions from one-way functions) from the case of random bits to the general setting of random variables.

- The most important application is a proof of tight bounds for security amplification achieved by the cascade of (possibly two-sided) ε-PRPs. This result shows that security amplification via plain cascading is, for all useful parameters, only marginally worse than by using randomized offsets as proposed in Chapter 4. This settles the question of understanding the security amplifying properties of cascades, a problem open since Luby and Rackoff’s original work [LR86] introducing PRPs.

- Finally, we provide alternative security amplification statements, with some quantitative improvements, for the random-offset variants of the cascade and of the composition of functions by quasigroup operations considered in Chapter 4.

1.5.2. Distinguisher Class Amplification

The last part of this dissertation (Chapter 6) considers the problem of achieving amplification with respect to the distinguisher class: In particular, we study a weak form of PRFs, which we refer to as s-RI-PRFs, where distinguishers are only allowed to issue s random (but known) queries to the given system, where s is a constant or some fixed function growing slowly. This notion weakens significantly the concept of a random-input PRF (RI-PRF), where indistinguishability holds against any efficient distinguisher asking an arbitrary number of random queries. We consider constructions of full-fledged PRFs from s-RI-PRFs, for s as low as 2, which outperform in efficiency even existing constructions [NR99, MS07] which require the stronger notion of a RI-PRF.

The results of this chapter show the feasibility of efficient symmetric cryptography from such very weak assumptions. In particular, a first application of our constructions is a new mode of operation for block ciphers allowing secure encryption nearly as efficient as in existing modes of operation (such as CBC and counter-mode encryption), but only requiring s-RI-PRF security for the underlying block cipher, for s as low as two. Also, our constructions are amenable to being instantiated from
iterated hash functions, such as SHA-1, under the assumption that the compression function is an $s$-RI-PRF. Overall, we remark that existing attacks on concrete cryptographic functions are far from disproving these security properties, consequently making the results of this chapter very attractive in practice.

1.5.3. Connection with Information-Theoretic Statements

In contrast to the limited amount of previous work on computational indistinguishability amplification, a longer line of research on indistinguishability amplification has addressed the information-theoretic setting, where indistinguishability holds also for computationally unbounded distinguishers, both with respect to the advantage and to the distinguisher class. For example, fundamental work in this direction has been done by Vaudenay [Vau98, Vau99]. A generalized treatment of these results has been subsequently presented by Maurer and Pietrzak [MP04], by Maurer, Pietrzak, and Renner [MPR07], and by Gaži and Maurer [GM09b].

While information-theoretic statements highlight fundamental properties of systems, and enjoy much tighter reductions than their computational counterparts, they are not realistic in the sense that concrete cryptographic primitives, such as block ciphers, are (at best) computationally secure. Yet, such results should nonetheless be taken as an indication of which properties we may hope to lift\(^{10}\) to the computational setting.

In particular, many results of this thesis are computational analogues of information-theoretic statements. Whenever possible, we will introduce them by first presenting the information-theoretic statement, which is generally simpler and often serves as a useful intuition, and then move on to presenting our computational counterpart.

---

\(^{10}\)The computational statement is usually stronger, i.e., it implies the information-theoretic statement, whereas the converse is obviously not true.
Preliminaries

Contributions and Outline of this Chapter. The main purpose of this chapter is to introduce the basic notation and tools employed throughout this thesis. In particular, Section 2.1 presents some basic notational conventions, whereas Section 2.2 provides an overview of probabilistic and information-theoretic tools needed in the following chapters.

The remainder (and bulk) of this chapter (Sections 2.3 and 2.4) is devoted to the framework of discrete interactive systems and to introducing the basic tools for information-theoretic indistinguishability proofs (one of which is novel to this thesis). Our treatment is based on Maurer’s original random framework [Mau02], with notational changes borrowed from [MPR07].

Finally, in order to formulate computational statements, we introduce a formal model of implementations of discrete systems in Section 2.5, as well as the associated complexity measures, and use this to define pseudorandom systems in Section 2.6.

2.1. Basic Notation

Throughout this thesis, we denote sets by calligraphic letters $\mathcal{X}, \mathcal{Y}, \ldots$, and use lower-case letters $x, y, \ldots$ to denote their elements. As usual, $\mathbb{N}$ and $\mathbb{R}$ denote the sets of natural and real numbers, and $\mathbb{R}_{\geq 0}$ and $\mathbb{R}_{>0}$
stand for the sets of non-negative and positive real numbers, respectively. (Analogously, \( \mathbb{N}_{>0} \) indicates the set of strictly positive integers.) Also, for a set \( \mathcal{X} \) and \( \ell \in \mathbb{N} \), we denote as \( \mathcal{X}^\ell \) the set of \( \ell \)-tuples \( x^\ell = (x_1, \ldots, x_\ell) \) of elements of \( \mathcal{X} \) with length \( |x^\ell| = \ell \). (Also, for a given \( x^\ell \), we adopt the convention that \( x_i \) is its \( i \)-th component, and \( x^i \) is its prefix of \( i \) elements.) Moreover, the \textit{empty tuple} \( \bot \) is the only tuple with length \( 0 \). We use the shorthands \( \mathcal{X}^\leq \ell, \mathcal{X}^\ast \) and \( \mathcal{X}^+ \) to denote the sets of tuples of elements of \( \mathcal{X} \) of length at most \( \ell \), of arbitrary length, and of arbitrary positive length, respectively. Additionally, the operator \( \| \) is the concatenation of tuples. Whenever \( \mathcal{X} \in \{0, 1\} \), we generally refer to a tuple in \( \mathcal{X}^\ell \) as an \( \ell \)-bit string, and write \( x = x_1 x_2 \ldots x_\ell \) instead of \((x_1, \ldots, x_\ell)\).

A function \( f : \mathbb{N} \to \mathbb{R}_{\geq 0} \) is \textit{negligible} if it vanishes faster than the inverse of any polynomial, i.e., for all \( c \geq 0 \) there exists \( k_0(c) \in \mathbb{N} \) such that \( f(k) \leq \frac{1}{k^c} \) for all \( k \geq k_0(c) \). Furthermore, a function is \textit{noticeable} if there exists some \( c \in \mathbb{N} \) such that \( f(k) \geq \frac{1}{k^c} \) for all sufficiently large \( k \)'s. We also take the convention that for a two-argument function \( f \) the notation \( f(x, \cdot) \) indicates the function \( y \mapsto f(x, y) \). This extends naturally to a higher number of arguments and/or open parameters. Also, for two real-valued functions \( f \) and \( g \), we write \( f \leq g \) if \( f(x) \leq g(x) \) holds for all arguments \( x \).

Finally, we follow the convention that \( \log(x) \) indicates the logarithm with base two, whereas \( \ln(x) \) is the natural logarithm.

### 2.2. Probabilities and Information-Theoretic Quantities

#### 2.2.1. Probabilities and Random Variables

Recall that a (discrete) \textit{random experiment} consists of a pair \((\mathcal{E}, \mathbb{P})\), where \( \mathcal{E} \) is the (discrete) set of \textit{elementary} events, and \( \mathbb{P} : \mathcal{E} \to [0, 1] \) is a probability measure associating with each event \( e \in \mathcal{E} \) its probability \( \mathbb{P}[e] \) and such that \( \sum_{e \in \mathcal{E}} \mathbb{P}[e] = 1 \). The probability of an event \( A \subseteq \mathcal{E} \) is denoted as \( \mathbb{P}[A] := \sum_{e \in A} \mathbb{P}[e] \). With a slight abuse of notation, we denote as \( A \lor B \) and \( A \land B \) the disjunction and the conjunction of the events \( A \) and \( B \), respectively. The conditional probability of \( A \) given \( B \) is \( \mathbb{P}[A \mid B] := \frac{\mathbb{P}[A \land B]}{\mathbb{P}[B]} \).

A \textit{random variable} \( X \) with \textit{range} \( \mathcal{X} \) is a mapping \( X : \mathcal{E} \to \mathcal{X} \), and defines a corresponding probability distribution \( \mathbb{P}_X : \mathcal{X} \to [0, 1] \) such that \( \mathbb{P}_X(x) := \mathbb{P}[X = x] = \sum_{e : X(e) = x} \mathbb{P}[e] \) for all \( x \in \mathcal{X} \). Joint probability distributions \( \mathbb{P}_{XY}, \mathbb{P}_{XYZ}, \ldots \) are defined accordingly by seeing multiple random variables as one single vector-valued variable. Frequently,
we specify the distribution of a random variable first, and a corresponding random experiment is defined implicitly. In particular, we generically refer to a function \( P : S \to [0, 1] \) with \( \sum_{s \in S} P(s) = 1 \) as a probability distribution on a set \( S \). We use the shorthand \( X \overset{\text{d}}{\leftarrow} P \) to indicate a random value \( X \) drawn according to the distribution \( P \). (We use \( X \overset{\text{d}}{\leftarrow} S \) to indicate sampling from the uniform distribution on \( S \).) The expected value of a distribution \( P \) on \( S \subseteq \mathbb{R} \) is
\[
E[P] := \sum_{s \in S} P(s) \cdot s,
\]
and \( E[X] := E[P_X] \) for a random variable. Recall that the expected value is linear, i.e.,
\[
E[X + Y] = E[X] + E[Y],
\]
for arbitrary \( X \) and \( Y \).

The conditional probability distribution of \( X \) given \( Y \) is a two-parameter function
\[
P_{X|Y} : X \times Y \to [0, 1]
\]
such that
\[
P_{X|Y}(x, y) := \frac{P_{XY}(x, y)}{P_Y(y)}.
\]
We also extend this notation to consider events. For example, \( P_{A|BY} : X \times Y \to [0, 1] \) is such that
\[
P_{A|BY}(x, y) := P[A \wedge X = x \wedge B \wedge Y = y].
\]
Given distributions involving multiple variables (e.g., \( P_{XY|UV} \)), projections obtained by setting variables to fixed values (e.g., \( P_{XY=y|UV=v,A} \)) are defined in the canonical way.

Finally, random variables \( X_1, \ldots, X_n \) are independent if and only if \( P_{X_1 \ldots X_n}(x_1, \ldots, x_n) = \prod_{i=1}^n P_{X_i}(x_i) \) for all \( x_1, \ldots, x_n \) (which we abbreviate as \( P_{X_1 \ldots X_n} = P_{X_1} \cdots P_{X_n} \)). Also, they are \( k \)-wise independent if all subsets of \( k \leq n \) variables are independent.

### 2.2.2. Tail Estimates

The following inequalities [Hoe63] are used repeatedly throughout this thesis: They state that the average of sufficiently many independent random variables \( X_1, \ldots, X_r \in [0, 1] \) is concentrated around its expected value.

**Lemma 2.1** (Hoeffding’s Inequalities). Let \( X_1, \ldots, X_r \) be independent random variables with range \([0, 1]\), and let \( \bar{X} := \frac{1}{r} \sum_{i=1}^r X_i \). Then, for all \( \varepsilon > 0 \),
\[
P[\bar{X} \geq E[\bar{X}] + \varepsilon] \leq e^{-r\varepsilon^2}
\]
and
\[
P[\bar{X} \leq E[\bar{X}] - \varepsilon] \leq e^{-r\varepsilon^2}.
\]
In particular,
\[
P[|\bar{X} - E[\bar{X}]| \geq \varepsilon] \leq 2 \cdot e^{-r\varepsilon^2}.
\]

### 2.2.3. Entropies

The entropy of a distribution \( P \) on \( S \) is a measure of the uncertainty about \( X \overset{\text{d}}{\leftarrow} P \) before it is sampled, or, equivalently, of the information provided
by \( X \) after it is sampled. The most common entropy measure is the Shannon entropy \( H(P) \)

\[
H(P) := \sum_{s \in S} P(s) \cdot \log \left( \frac{1}{P(s)} \right),
\]

with the convention that \( 0 \cdot \log \left( \frac{1}{0} \right) = 0 \), and for a random variable \( X \) we define \( H(X) := H(P_X) \).\(^1\) Recall that \( 0 \leq H(X) \leq \log |X| \), where \( 0 \) is achieved if there exists \( x \) such that \( P_X(x) = 1 \), whereas \( H(X) \) is maximal if \( X \) is uniform on \( X \). The joint entropy \( H(XY) \) of \( X \) and \( Y \) is defined as \( H(P_{XY}) \), and for an event \( A \) we define \( H(X|A) \) as the entropy of \( P_{X|A} \). Also, the conditional entropy \( H(X|Y) \) is

\[
H(X|Y) = \mathbb{E}_{y \sim P_Y} [H(X|Y = y)] = H(XY) - H(Y).
\]

Note that in general \( H(X|Y) \leq H(X) \leq H(XY) \leq H(X) + H(Y) \) for all random variables \( X \) and \( Y \).

An alternative measure is the min-entropy

\[
H_\infty(P) := \min_{s \in S} \left[ \log \left( \frac{1}{P(s)} \right) \right],
\]

with, once again, \( H_\infty(X) := H_\infty(P_X) \). Also, it holds that \( 0 \leq H_\infty(X) \leq \log |X| \), as well as \( H_\infty(X) \leq H(X) \).

2.2.4. Distance Measures

The most natural distance measure between two probability distributions \( P \) and \( Q \) is the statistical distance (or variational distance)

\[
d(P, Q) := \frac{1}{2} \sum_{s \in S} |P(s) - Q(s)| = \frac{1}{2} \| P - Q \|_1,
\]

where \( \| \cdot \|_1 \) is the \( L_1 \)-norm. We also write \( d(X, Y) := d(P_X, P_Y) \) for any two random variables \( X \) and \( Y \) with range \( X = Y = S \). Clearly, the statistical distance is a metric (on the set of probability distributions on a finite set \( S \)), and satisfies in particular the triangle inequality, i.e., \( d(P, P') \leq d(P, P'') + d(P'', P') \) for any distributions \( P, P', \) and \( P'' \).

\(^1\)This last notation is more common, and refers to an understood random experiment.
It can be shown that
\[ d(P, Q) = \max_{A \subseteq S} [P(A) - Q(A)] = \max_{A \subseteq S} [Q(A) - P(A)], \] (2.1)
with \( P(A) := \sum_{s \in A} P(s) \) and \( Q(A) := \sum_{s \in A} Q(s) \). The sets achieving the maxima are \( A^+ := \{ s : P(s) \geq Q(s) \} \) in the former case and \( A^- := \{ s : Q(s) \geq P(s) \} \) in the latter case.

A further important property is that two random variables \( X \) and \( Y \) can always be seen as being equal with probability \( 1 - d(X, Y) \), in the following sense.

**Lemma 2.2.** Let \( X \) and \( Y \) be random variables with range \( S \). Then, there exist events \( A \) and \( B \) defined on \( X \) and \( Y \), respectively, such that \( P_X|A = P_Y|B, \) and \( P[A] = P[B] = 1 - d(X, Y) \).

A further important distance measure is the relative entropy (or Kullback-Leibler divergence) \( D(P\|Q) \) defined as
\[ D(P\|Q) := \sum_{s \in S} P(s) \cdot \log \left( \frac{P(s)}{Q(s)} \right). \]

As usual, for random variables \( X \) and \( Y \), we let \( D(X \| Y) := D(P_X \| P_Y) \). We stress that \( D(P\|Q) \) is not a metric, as it is neither symmetric, nor it satisfies the triangle inequality. It can be shown that (cf. e.g. [CT91], Lemma 12.6.1)
\[ d(P, Q)^2 \leq D(P\|Q), \] (2.2)
which is referred to as Pinsker’s inequality. A final distance measure is the difference \( |H(P) - H(Q)| \) between the entropies of two probability distributions: If \( d(P, Q) \leq \frac{1}{4} \), then (cf. [CT91], Lemma 16.3.2)
\[ |H(P) - H(Q)| \leq 2 \cdot d(P, Q) \cdot \log \left( \frac{|S|}{2d(P, Q)} \right). \] (2.3)

## 2.3. Discrete Systems

### 2.3.1. Definition and Basic Properties

The basic objects considered in this thesis are discrete interactive systems which take inputs \( X_1, X_2, \ldots \in \mathcal{X} \) and return the corresponding output values \( Y_1, Y_2, \ldots \in \mathcal{Y} \) (where \( \mathcal{X} \) and \( \mathcal{Y} \) are discrete sets), and which have
the property that the $i$-th output $Y_i$ only depends on the first $i$ inputs $X^i = [X_1, \ldots, X_i]$, the first $i - 1$ outputs $Y^i = [Y_1, \ldots, Y_{i-1}]$, and some internal state. (Also cf. Figure 2.1.) Such systems may, for example, be implemented by some physical device or realized by an interactive algorithm, as we will see below in Section 2.5. We present a language for expressing implementation-independent statements involving such discrete interactive systems. To this end, we follow the approach of Maurer’s random systems framework [Mau02], with minor modifications in terminology from [MPR07].

An $(X, Y)$-system $S$ is the abstraction of an instance of such a discrete system, i.e., an abstract object with a well-defined input-output behavior, but for which we do not specify any further implementation details. In particular, the input-output behavior (and hence $S$) is fully characterized by the conditional probability distributions of the $i$-th output $Y_i$ given $X^i$ and $Y^{i-1}$.

**Definition 2.1.** The input-output behavior of an $(X, Y)$-system $S$ is the infinite family of conditional probability distributions $p_{Y_i|X^i,Y^{i-1}}^S$ for all $i \geq 1$.

The lower-case $p$ explicits the fact that we only consider conditional probability distributions which are independent of the random experiment the system is used in. Also, we often omit the parameters $(X, Y)$ whenever they are clear from the context, or irrelevant.

There are different implicit ways of specifying the input-output behavior of $S$, such as by describing its implementation in some understood model. However, it is crucial to remark that every statement about systems holds for any object expressing the same behavior, regardless of its actual implementation.

In particular, two systems with completely different implementations can express the same input-output behavior, justifying the following notion.

**Definition 2.2.** Two $(X, Y)$-systems $S$ and $T$ are equivalent, denoted $S \equiv T$, if
\[ p_{Y_i|X^i,Y^{i-1}}^S = p_{Y_i|X^i,Y^{i-1}}^T \]
holds for all $i \geq 1$.

An $(X, Y)$-system $S$ is deterministic if $p_{Y_i|X^i,Y^{i-1}}^S(y_i, x^i, y^{i-1}) = 0, 1$ for all $i \geq 1$, $x^i \in X^i$, and $y^i \in Y^i$. Moreover, it is stateless if there exists a conditional probability distribution $p_{Y_i|X}^S$ such that for all $i \geq 1$, $x^i \in X^i$
and \( y^i \in Y^i \) we have
\[
p_{Y_i|X^i \cup Y^i}^S(y_i, x^i, y^{i-1}) = p_{Y_i|X}^S(y_i, x_i).
\]
Note that a stateless system corresponds to the traditional notion of a communication channel used in information theory.

The input-output behavior of a system \( S \) can equivalently (and sometimes more conveniently) be described by the probability distributions \( p_{Y_i|X^i}^S \) for all \( i \geq 1 \), of the first \( i \) outputs \( Y^i \) in a random experiment where the system \( S \) is fed the first \( i \) inputs \( X^i \). It is important to avoid any confusion due to the fact that, in a random experiment where \( S \) is used, we usually have \( P_{Y_i|X^i} \neq p_{Y_i|X^i}^S \), whereas \( P_{Y_i|X^i \cup Y^i} = p_{Y_i|X^i \cup Y^i}^S \) always holds.

Note that both views are related by the equalities
\[
p_{Y_i|X^i \cup Y^i}^S(y_i, x^i, y^{i-1}) = \frac{p_{Y_i|X^i}^S(y_i, x^i)}{p_{Y_i|X^i \cup Y^i}^S(y_i, x^i, y^{i-1})} \tag{2.4}
\]
and
\[
p_{Y_i|X^i}^S(y_i, x^i) = \prod_{j=1}^{i} p_{Y_j|X^j \cup Y^{j-1}}^S(y_j, x^j, y^{j-1}) \tag{2.5}
\]
for all \( i \geq 1 \), \( x^i \in X^i \), and \( y^i \in Y^i \). Moreover, it is not hard to verify that \( S \equiv T \) holds if and only if \( p_{Y_i|X^i}^S = p_{Y_i|X^i}^T \), for all \( i \geq 1 \).

### 2.3.2. Examples

The following are important examples of systems considered in this thesis.
Example 2.1 (Random Variables). A random variable $X$ is as a special case of a system with no interaction. There are two natural approaches to model random variables as systems. The first one is as a system which returns $X$ upon its first invocation (on input the dummy symbol $\bot$), and no value is returned upon any subsequent invocation.

An alternative is to think of $X$ as a system initially sampling a single value $x \leftarrow P_X$, and subsequently returning the same $x$ upon each invocation. We adopt this second view, as it helps providing homogeneous statements in our results.

Example 2.2 (Functions). A function $f : \mathcal{X} \rightarrow \mathcal{Y}$ is also an $(\mathcal{X}, \mathcal{Y})$-system for which $Y_i = f(X_i)$ for all queries $X_i$. In particular, functions are both deterministic and stateless systems, and any deterministic stateless system can be seen as a function.

Example 2.3 (Random Beacon). A random beacon $B$ with range $\mathcal{Y}$ is a stateless system that upon each invocation ignores its input and returns an independent and uniformly distributed element from $\mathcal{Y}$, i.e., for all $i \geq 1$,

$$p_{Y_i|X_{i-1}}(y_i, x^i, y^{i-1}) = \frac{1}{|\mathcal{Y}|}$$

for all $x^i \in \mathcal{X}^i$ and $y^i \in \mathcal{Y}^i$.

Example 2.4 (Random Functions). A random function $F : \mathcal{X} \rightarrow \mathcal{Y}$ is an $(\mathcal{X}, \mathcal{Y})$-system which answers consistently, i.e., with the property that if $X_i = X_j$ for the $i$-th and $j$-th query, then $Y_i = Y_j$ holds for the corresponding outputs, too. In particular, such system answers consistently with a function table which is defined (possibly on the fly) during the interaction with it, and, if $x$ has been queried, we denote as $F(x)$ the value associated with $x$.

An important special case is a uniform random function (URF) $R : \mathcal{X} \rightarrow \mathcal{Y}$ where, for all $x \in \mathcal{X}$, the value $R(x)$ is uniform and independent of $R(x')$ for all $x' \neq x$. Note that $\mathcal{X}$ is usually finite, but we also might have $\mathcal{X} = \{0, 1\}^*$, the set of all binary strings.

Example 2.5 (Random Permutations). If a random function $Q : \mathcal{X} \rightarrow \mathcal{X}$ additionally satisfies that $Y_i = Y_j$ implies $X_i = X_j$, then it is called a random permutation. A uniform random permutation (URP) $P : \mathcal{X} \rightarrow \mathcal{X}$ sets $P(x)$ to a uniformly distributed value different from $P(x')$ for all previously queried $x' \neq x$.

2.3.3. Constructions and System Composition

A $(\mathcal{U}, \mathcal{V}, \mathcal{X}, \mathcal{Y})$-construction $C$ is a discrete system with two interfaces, called the inner and the outer interface, respectively. Upon each query $u \in$
2.3 Discrete Systems

At the outer interface, the construction interacts with a given $((X', Y'))$-system at the inner interface, and after a certain number of queries, returns an answer $v \in V$ at the outer interface. In particular, for an $((X, Y))$-system $S$ and a $((U, V, X', Y'))$-construction $C$, we denote as $C(S)$ the resulting $((U, V))$-system. (See Figure 2.1.) In this case, we usually refer to $S$ as the subsystem. We typically omit the sets $U, V, X$, and $Y$ whenever they are either clear from the context or not relevant. Also, the input-output behavior of a construction can be defined by a family of conditional probability distributions, even though this will not be necessary, except for the case of system users, introduced below.

Constructions can be composed sequentially.

**Definition 2.3.** Given a $((U, V, U', V'))$-construction $C$ and a $((U', V', U'', V''))$-construction $C'$, we denote as $CC'$ the sequential composition of both constructions, i.e., the $((U, V, U'', V''))$-construction such that

$$
(CC')(S) = C(C'(S))
$$

for all $((U'', V''))$-systems $S$.

We also allow constructions with multiple subsystems, and denote as $C(S_1, \ldots, S_m)$ the resulting system. Also, for a two-subsystem construction $C(\cdot, \cdot)$, the shorthand $C(S, \cdot)$ stands for the construction such that

$$
C(S, \cdot)(T) = C(S, T)
$$

for all compatible systems $T$, as illustrated in Figure 2.2. The notion naturally extends to higher numbers of subsystems.
We conclude this section by listing recurring examples of constructions combining an \((\mathcal{X},\mathcal{Y})\)-system \(S\) and an \((\mathcal{X}',\mathcal{Y}')\)-system \(T\). It is convenient to write these constructions in terms of operators, using infix notation, rather than using the standard notation for constructions.

**Example 2.6 (Parallel composition).** The parallel composition \(S \parallel T\) is the \((\{1,2\} \times (\mathcal{X} \cup \mathcal{X}'),(\mathcal{Y} \cup \mathcal{Y}'))\)-system which, on query \((i,x)\), first issues the query \(x\) to \(S\) if \(i = 1\), and to \(T\) if \(i = 2\), and subsequently returns, at the outer interface, the corresponding output.

**Example 2.7 (System mixture).** If \(\mathcal{X} = \mathcal{X}'\) and \(\mathcal{Y} = \mathcal{Y}'\), and for a random bit \(B\), the system \(\langle S, T \rangle_B\) first samples the bit \(B\) and then acts as the system \(S\) if \(B = 0\), and as \(T\) otherwise.

**Example 2.8 (Output composition).** If \(\mathcal{X} = \mathcal{X}'\) and \(\mathcal{Y} = \mathcal{Y}'\) and for any operation \(*\) on \(\mathcal{Y}\), the \((\mathcal{X},\mathcal{Y})\)-system \(S \ast T\), on input \(x\), invokes both \(S\) and \(T\) with input \(x\), obtaining \(y\) and \(y'\), and returns \(y \ast y'\).

**Example 2.9 (Cascades).** If \(\mathcal{Y} = \mathcal{X}'\), we denote with \(S \triangleright T\) the cascade of \(S\) and \(T\), i.e., the system which on input \(x\) first invokes \(S\) on this input, and the resulting output is fed into \(T\) to obtain the final output.

Also, all these notions extend naturally to an arbitrary number of subsystems.\(^2\) Figure 2.3 provides a pictorial representation of the abovely defined composition operators.

### 2.3.4. Convex-Combination Stateless Systems and Functions

We often consider systems \(S \equiv C(S)\) where \(C(\cdot)\) is a construction accessing a random variable \(S\) called the *initial state*. Usually, with a slight abuse of notation, we will use \(S(S)\) instead of \(C(S)\) to save on notation. The following class of systems will be central in the following.

**Definition 2.4.** A system \(S\) is *convex-combination stateless* (or *cc-stateless*, for short) if there exists a construction \(S(\cdot)\) and a random variable \(S\) with \(S(S) \equiv S\) and \(S(s)\) is stateless for all possible initial states \(s\).

Most cryptographic systems of interest turn out to be cc-stateless. We provide some examples.

**Example 2.10 (CC-stateless random functions).** It is easy to verify that a cc-stateless random function \(F \equiv F(F) : \mathcal{X} \rightarrow \mathcal{Y}\) is such that \(F(f)\) is a

\(^2\)In the output-composition case, since \(*\) may not necessarily be associative, we denote as \(S_1 \ast \cdots \ast S_m\) the system \((\cdots ((S_1 \ast S_2) \ast S_3) \cdots ) \ast S_m\).
2.3 Discrete Systems

Figure 2.3: Different system-composition examples. Top, left: $S \parallel T$. Top, right: $\langle S, T \rangle_B$. Bottom, left: The output composition $S \star T$. Bottom, right: The cascade $S \triangleright T$.

function $X \to Y$ for all $f$. Without loss of generality, we see the initial state of $F$ as the function table $F : X \to Y$ according to which $F$ answers its queries. Examples of cc-stateless random functions are a URF $R : X \to Y$ (the function table is chosen uniformly at random from the set of all functions $X \to X$), a URP $P : X \to X$ (a permutation on $X$ is chosen uniformly at random), and the system $E(K, \cdot)$, where $E : \{0, 1\}^k \times \{0, 1\}^n \to \{0, 1\}^n$ is a function (e.g. implemented by a block cipher), and $K$ is a uniform $k$-bit string.

Example 2.11 (CC-stateless random permutations and inverses.) A cc-stateless random permutation $Q : X \to X$ with function table $Q$ has a well defined inverse $Q^{-1}$ which answers according to $Q^{-1}$. In particular, we define the system $\langle Q \rangle$ as the two-sided random permutation which answers forward queries $(x, +)$ as $Q(x)$ and backward queries $(y, -)$ as $Q^{-1}(y)$.

---

3Namely, if the system where randomized and stateless, there must be a probability that two queries $X_i = X_j$ satisfy $Y_i \neq Y_j$. 

---
Example 2.12 (Randomized Encryption Schemes). A natural example of a cc-stateless system which is not a random function is given by a randomized encryption scheme, i.e., for a function $\text{encrypt} : \{0,1\}^k \times \{0,1\}^r \times \{0,1\}^* \to \{0,1\}^*$ we consider the system initially choosing a uniform key $K \in \{0,1\}^k$, and upon each query $X_i \in \{0,1\}^*$ it generates a uniform $r$-bit string $R_i$ and returns $Y_i := \text{encrypt}(K, R_i, X_i)$.

2.4. Indistinguishability of Systems and Game Winning

2.4.1. Distinguishers and Distinguishing Advantage

We consider random experiments where an entity interacts with a given system. A $q$-query $(X, Y)$-user $U$ is a construction which takes no input (or alternatively, the first input is ignored), interacts with some given system $S$ by means of $q$ queries, and returns a final output value $V$, which we denote by $U(S)$. It does not accept any further inputs. It can formally be described by the conditional probability distributions $p^U_{X_i|X_{i-1}Y_{i-1}}$ for all $i = 1, \ldots, q$, as well as $p^S_{Y_i|X_iY_{i-1}}$. We let $U_q$ be the set of all $q$-query $(X, Y)$-users (for understood $X$ and $Y$). We denote as $US$ the experiment where $U$ interacts with $S$, generating a $q$-query transcript $(X_q, Y_q) \in \mathcal{X}^q \times \mathcal{Y}^q$ such that for all possible $(x^q, y^q) \in \mathcal{X}^q \times \mathcal{Y}^q$,

$$p^US_{X^qY^q}(x^q, y^q) = \prod_{i=1}^q p^U_{X_i|X_{i-1}Y_{i-1}}(x_i, x^{i-1}, y^{i-1}) \cdot p^S_{Y_i|X_iY_{i-1}}(y_i, x^i, y^{i-1})$$

$$= p^U_{X_qY_{q-1}}(x^q, y^{q-1}) \cdot p^S_{Y_q|X_q}(y^q, x^q),$$

where $p^U_{X_qY_{q-1}}$ is the probability distributions of $U$’s $q$ queries given that the first $q - 1$ answers $Y^{q-1}$ are fixed to a given value. Furthermore, we have, for all values $v$ which can be taken by $U(S)$,

$$P[U(S) = v] = \sum_{x^q, y^q} p^U_{V|X^qY^q}(v, x^q, y^q) \cdot p^US_{X^qY^q}(x^q, y^q).$$

An important special case is a $q$-query $(X, Y)$-distinguisher $D$, which is defined as a $q$-query $(X, Y)$-user with binary output. We associate with each distinguisher $D$ the corresponding distinguishing advantage measuring the ability of $D$ to tell apart two systems $S$ and $T$.

---

4This notion does not appear in previous work on systems, but it will be convenient to unify statements in the following.
Definition 2.5. The distinguishing advantage of \( D \) in distinguishing \( S \) and \( T \) is defined as

\[
\Delta^D(S, T) := |P[D(S) = 1] - P[D(T) = 1]|.
\]

Furthermore, with \( D_q \subseteq U_q \) being the set of all distinguishers making \( q \) queries, we define

\[
\Delta_q(S, T) := \max_{D \in D_q} \Delta^D(S, T).
\]

We remark that the maximum in the definition of \( \Delta_q(S, T) \) is well defined, since, by a standard argument, for every distinguisher \( D \) there exists a deterministic distinguisher \( D^* \) such that \( \Delta^D(S, T) \leq \Delta^{D^*}(S, T) \), and thus we can restrict the maximum to the set of \( q \)-query deterministic distinguishers, which is finite. (In the following, we silently assume that the output sets of systems are always finite.)

The distinguishing advantage is a pseudo-metric: On top of being non-negative, and symmetric (i.e., \( \Delta^D(S, T) = \Delta^D(T, S) \) for all \( D, S, \) and \( T \)), it satisfies the triangle inequality, i.e.,

\[
\Delta^D(S, S'') \leq \Delta^D(S, S') + \Delta^D(S', S'')
\]

for all \( D, S, S', \) and \( S'' \), and the inequality also holds when maximizing over some class of distinguishers, such as \( D_q \).

Moreover, for the special case of two random variables \( X \) and \( Y \), it follows directly from Equation (2.1) in Section 2.2 that \( \Delta_1(X, Y) = d(X, Y) \). This is generalized to any two systems by the following lemma.

Lemma 2.3. For all \( q \geq 1 \),

\[
\Delta_q(S, T) = \max_{U \in U_q} d(P^{US}_{X^{q}Y^{q}}, P^{UT}_{X^{q}Y^{q}})
\]

Proof. For a user \( U \), a distinguisher \( D \) with \( \Delta^D(S, T) \geq d\left(P^{US}_{X^{q}Y^{q}}, P^{UT}_{X^{q}Y^{q}}\right) \) runs \( U \) and outputs a bit according to the optimal distinguisher for the statistical distance \( d\left(P^{US}_{X^{q}Y^{q}}, P^{UT}_{X^{q}Y^{q}}\right) \). Conversely, given a distinguisher \( D \), we have \( \Delta^D(S, T) \leq d\left(P^{DS}_{X^{q}Y^{q}}, P^{DT}_{X^{q}Y^{q}}\right) \), as one particular strategy for a distinguisher for the statistical distance is to output a bit according to \( D \)'s behavior, and distinguisher is a user. \( \square \)
2.4.2. Monotone Binary Outputs (MBOs) and Game Winning

We consider systems with a designated binary output, and we are interested in the special case where this binary output is monotone, i.e., once it becomes one, the system will always output one. The following definition was first given in \[MPR07\].

**Definition 2.6.** An \((X, Y)\)-system \(\hat{S}\) with an MBO is an \((X, Y \times \{0, 1\})\)-system \(\hat{S}\), where the binary component \(A_1, A_2, \ldots\) of the output is called a monotone binary output (MBO), and has the property that \(A_i = 1\) implies \(A_j = 1\) for all \(j \geq i\). Furthermore, we define the following two derived systems:

(i) \(\hat{S}^-\) is the \((X, Y)\)-system obtained by ignoring the MBO.

(ii) \(\hat{S}^\dagger\) is the \((X, Y \times \{0, 1\})\)-system obtained by setting, for all \(i \geq 1\), the \(Y\)-output \(Y_i := \perp\) whenever \(A_i = 1\), whereas \(Y_i\) is unchanged if \(A_i = 0\).

Every \((X, Y)\)-system \(\hat{S}\) with an MBO implicitly defines a game where an \((X, Y)\)-user \(U\) interacts with \(\hat{S}\) and wins if the MBO becomes one. For an \((X, Y)\)-system \(\hat{S}\) with MBO \(A_1, A_2, \ldots\), and a \(q\)-query \((X, Y)\)-user \(U \in U_q\), we define

\[
\nu^U(\hat{S}) := \mathbb{P}_{A_q}^{\hat{S}}(1) = \sum_{x^q, y^q} \mathbb{P}_{X^q, Y^q | A_q}^{\hat{S}}(x^q, y^q, 1) = \sum_{x^q, y^q} \mathbb{P}_X^{U} x^q y^q \mathbb{P}_A^{\hat{S}}(1, y^q, x^q),
\]

that is, this is the probability that the MBO becomes 1 during \(U\)'s interaction with \(\hat{S}\). Also, we let

\[
\nu_q(\hat{U}) := \max_{U \in U_q} \nu^U(\hat{S}).
\]

It is also convenient to define the parallel composition \(\hat{S}_1 \parallel \ldots \parallel \hat{S}_m\) of systems with MBOs \(\hat{S}_1, \ldots, \hat{S}_m\) as the parallel composition of \(\hat{S}_1^- \parallel \ldots \parallel \hat{S}_m^-\) with an MBO which is one if and only if all \(m\) MBOs associated with the

---

5Earlier work on random systems [Mau02] has been using the equivalent notion of a monotone event sequence (MES).

6In particular, we assume that it does not obtain the MBO value in the replies to its queries, as it will be syntactically convenient, yet this assumption can be dropped.
2.4 Indistinguishability of Systems and Game Winning

$m$ subsystems are 1. A fundamental property is that the probability of winning multiple games in parallel decreases exponentially in the number of games. (A proof is given in [MPR07].)

**Theorem 2.4.** Let $\hat{S}_1, \ldots, \hat{S}_m$ be systems with MBOs. Then, for all $q_1, \ldots, q_m$,

$$\nu_{q_1, \ldots, q_m}(\hat{S}_1 \parallel \cdots \parallel \hat{S}_m) \leq \prod_{i=1}^{m} \nu_{q_i}(\hat{S}_i).$$

2.4.3. From Indistinguishability to Game Winning

We consider the general problem of upper bounding $\Delta_D(S, T)$ (for a distinguisher $D$) or $\Delta_q(S, T)$. The following lemma (which was initially shown in [Mau02], and which we formulate in the version of [MPR07], to which we refer for a proof) states that if both systems can be extended to systems with MBOs $\hat{S}$ and $\hat{T}$ that are equivalent as long as the MBO is $0$, then the distinguishing advantage $\Delta_D(S, T) = \Delta_D(\hat{S}^-, \hat{T}^-)$ is upper bounded by the probability that $D$, when interacting with either $\hat{S}$ or $\hat{T}$, provokes the MBO to become $1$.

**Lemma 2.5.** Let $\hat{S}$ and $\hat{T}$ be two systems with MBOs such that $\hat{S}^\parallel = \hat{T}^\parallel$. Then, for all distinguishers $D$,

$$\Delta_D(\hat{S}^-, \hat{T}^-) \leq \nu^D(\hat{S}) = \nu^D(\hat{T}).$$

Note that the condition $\hat{S}^\parallel = \hat{T}^\parallel$ means that

$$p_{Y_i, A_i = 0 | X, Y_{i-1}} = p_{Y_i, B_i = 0 | X, Y_{i-1}, B_{i-1}}$$

for all $i \geq 1$, where $A_1, A_2, \ldots$ and $B_1, B_2, \ldots$ are the MBOs of $\hat{S}$ and $\hat{T}$, respectively. This is equivalent to $p_{Y_i, A_i = 0 | X_i} = p_{Y_i, B_i = 0 | X_i}$ for all $i \geq 1$.

In the remainder of this section, we discuss different conditions which, thanks to Lemma 2.5, can be used to obtain an upper bound on the distinguishing advantage in terms of the winning probability.

---

7We note that this result heavily relies on allowing all possible entities playing the games. Restricting this class, even in a truly information-theoretic setting, leads to less well-understood phenomena, cf. e.g. [Raz98].
A weaker condition. We start by giving a weaker condition than $\hat{S}^{-} \equiv \hat{T}^{-}$ which suffices for an upper bound.

Lemma 2.6. Let $\hat{S}$ and $\hat{T}$ be two systems with MBOs $A_1, A_2, \ldots$ and $B_1, B_2, \ldots,$ respectively, such that for all $i \geq 1,$

\[ p_{Y \mid X^{i-1} A_i = 0 \mid X^{i-1} B_i = 0} \quad (2.6) \]

and

\[ p_{A_i = 0 \mid X^{i-1} A_{i-1} = 0} \geq p_{B_i = 0 \mid X^{i-1} B_{i-1} = 0}. \]

Then,

\[ \Delta^D(\hat{S}^-, \hat{T}^-) \leq \nu^D(\hat{T}). \]

Proof. Define $\hat{S}'$ with an MBO $A'_1, A'_2, \ldots$ where $A'_i := A_i \lor C_i,$ with, for all $i \in \mathbb{N},$ and all monotone sequences $c^{i-1} \in \{0, 1\}^{i-1},$

\[ p_{Y_i ; A'_i \mid X^{i-1} A_i = c^{i-1}} = \hat{p}_{Y_i ; A_i \mid X^{i-1} A_i = c^{i-1}} \quad (2.7) \]

and $C_i$ is such that for all $x_i, y_i$

\[ p_{\hat{S}'_{C_i} \mid X^{i-1} A_i = c_i - 0} = \frac{p_{\hat{B} ; X^{i-1} B_i = 0} (0, x_i, y_i, 0, 0)}{p_{\hat{S} ; X^{i-1} A_i = c_i - 0} (0, x_i, y_i, 0)}. \]

Clearly, $(\hat{S}')^- \equiv \hat{S}^-.$ Further,

\[ p_{Y_i ; A'_i = 0 \mid X^{i-1} A_i = 0} = p_{\hat{S}'_{C_i} = 0 \mid X^{i-1} A_i = 0 \mid X^{i-1} A_{i-1} = 0} = p_{\hat{S}'_{C_i} = 0 \mid X^{i-1} A_i = 0} \cdot p_{\hat{S}'_{C_i} = 0 \mid X^{i-1} A_{i-1} = 0} \]

\[ = p_{\hat{S}'_{C_i} = 0 \mid X^{i-1} A_i = 0} \cdot p_{\hat{S}'_{C_i} = 0 \mid X^{i-1} A_{i-1} = 0} \]

Moreover, $(2.6)$ and $(2.8)$ yield

\[ p_{Y_i ; A'_i = 0 \mid X^{i-1} A_i = 0} = \hat{p}_{Y_i ; B_i = 0 \mid X^{i-1} B_i = 0} \]

which implies $(\hat{S}')^{-1} \equiv \hat{T}^{+1},$ and Lemma 2.5 hence concludes the proof. □

The following direct corollary follows by defining the all-zero MBO on $S.$
Corollary 2.7. Let $S$ be a system and let $\hat{T}$ be a system with MBO $B_1, B_2, \ldots$, such that for all $i \geq 1$,

$$
\mathbb{P}_{Y_i|X^i Y^{i-1}} = \mathbb{P}_{Y_i|X^i Y^{i-1} B_i = 0}.
$$

Then,

$$
\Delta^D(S, \hat{T}) \leq \nu^D(\hat{T}).
$$

Non-adaptive strategies. We give a sufficient condition such that the distinguisher maximizing $\nu^D(S)$ is non-adaptive, thus considerably simplifying the task of bounding the advantage.

Lemma 2.8. Let $\hat{S}$ be a system with an MBO $A_1, A_2, \ldots$ such that, for another system $T$,

$$
\mathbb{P}_{Y_i|X^i A_i = 0} = \mathbb{P}_{Y_i|X^i}
$$

for all $i \geq 1$. Then, for all users $U$,

$$
\nu^U(\hat{S}) \leq \max_{x^q \in \mathcal{X}(U)} \mathbb{P}_{A_q|X^q}(1, x^q).
$$

where $\mathcal{X}(U)$ is the set of input sequences which occur in some interaction of $U$ with positive probability.

Proof. First, note that if there exists an $x^q \in \mathcal{X}^q$ such that $\mathbb{P}_{A_q|X^q}(0, x^q) = 0$, then the claim of the lemma is vacuously true. Otherwise, fix some user $U$, and let $Z$ be the set of pairs $(x^q, y^q)$ with $\mathbb{P}_{X^q Y^q}(x^q, y^q) > 0$. Then,

$$
\begin{align*}
\mathbb{P}_{A_q}(0) &= \sum_{(x^q, y^q) \in Z} \mathbb{P}_{A_q X^q Y^q}(0, x^q, y^q) \\
&= \sum_{(x^q, y^q) \in Z} \mathbb{P}_{X^q Y^q-1}(x^q, y^{q-1}) \cdot \mathbb{P}_{Y_q A_q|X^q}(y^q, 0, x^q) \\
&= \sum_{(x^q, y^q) \in Z} \mathbb{P}_{X^q Y^q-1}(x^q, y^{q-1}) \cdot \mathbb{P}_{Y_q|X^q A_q}(y^q, x^q, 0) \cdot \mathbb{P}_{A_q|X^q}(0, x^q) \\
&= \sum_{(x^q, y^q) \in Z} \mathbb{P}_{X^q Y^q-1}(x^q, y^{q-1}) \cdot \mathbb{P}_{Y_q|X^q}(y^q, x^q) \cdot \mathbb{P}_{A_q|X^q}(0, x^q).
\end{align*}
$$

In particular, we conclude that

$$
\nu^U(\hat{S}) = \mathbb{P}_{A_q}(1) = 1 - \mathbb{P}_{A_q}(0) \\
\leq \max_{x^q \in \mathcal{X}(U)} \mathbb{P}_{A_q|X^q}(1, x^q).
$$
by using the fact that $\sum_{x^q, y^q} p_{X^q|Y^{q-1}}^U(x^q, y^{q-1}) \cdot p_{Y^q|X^q}^T(y^q, x^q) = 1$.  

There are situations where Lemma 2.8 cannot be used. Still, we would like to avoid computing the best adaptive probability of provoking $A_q$ to be 1. We consider the setting where an MBO is defined on a system $\hat{S} = \hat{S}(S)$ (i.e., where $S$ is some initial state) such that the behavior of $\hat{S}$ is independent of $S$ as long as the MBO is 0. We show that we can equivalently consider the maximal probability over all compatible transcripts $(x^q, y^q)$ that an independently chosen state $S$ provokes the MBO to be one. (To the best of our knowledge, no such result appears in the literature.)

**Lemma 2.9.** Let $\hat{S} \equiv \hat{S}(S)$ be an $(\mathcal{X}, \mathcal{Y})$-system with an MBO $A_1, A_2, \ldots$ and an internal variable $S$. If there exists an $(\mathcal{X}, \mathcal{Y})$-system $T$ such that, for all $i \geq 1$, $s$, $x^i$, and $y^i$,

$$p_{A_i|X^i, Y^i}^S(0, y^i, s, x^i) = p_S(s) \cdot p_{T|X^i}^T(y^i, x^i) \cdot p_{A_i|X^i, Y^i}^S(0, x^i, y^i, s),$$

then, for all users $U$,

$$\nu^U(\hat{S}) \leq \max_{(x^q, y^q) \in \mathcal{Z}(U)} \sum_{s \in S} p_S(s) \cdot p_{A_q|X^q, Y^q}^S(1, x^q, y^q, s),$$

where $\mathcal{Z}(U)$ is the set of input-output sequences which occur with positive probability in some interaction of $U$ with some system.

**Proof.** Let $U$ be a $q$-query user interacting with $\hat{S}$,

$$P_{A_q}^{U\hat{S}}(0) = \sum_{x^q, y^q, s} p_{X^q Y^q S A_q}^U(x^q, y^q, s, 0)$$

$$= \sum_{x^q, y^q, s} p_{X^q|Y^{q-1}}^U(x^q, y^{q-1}) \cdot p_{A_q Y^q S|X^q}^S(0, y^q, s, x^q)$$

$$= \sum_{x^q, y^q} p_{X^q Y^q}^U(x^q, y^q) \cdot \sum_{s \in S} p_S(s) \cdot p_{A_q|X^q, Y^q}^S(0, x^q, y^q, s),$$

and thus, since $P_{A_q}^{U\hat{S}}(1) = 1 - P_{A_q}^{U\hat{S}}(0)$,

$$P_{A_q}^{U\hat{S}}(1) = \sum_{x^q, y^q} p_{X^q Y^q}^U(x^q, y^q) \cdot \sum_{s \in S} p_S(s) \cdot p_{A_q|X^q, Y^q}^S(1, x^q, y^q, s)$$

$$\leq \max_{(x^q, y^q) \in \mathcal{Z}(U)} \sum_{s \in S} p_S(s) \cdot p_{A_q|X^q, Y^q}^S(1, x^q, y^q, s). \quad \Box$$
2.5 Algorithms and Complexities

2.5.1. Interactive Algorithms and Complexity

We consider interactive algorithms implementing constructions and systems: Interactive algorithms are stateful, i.e., they keep a state $\sigma$ during the execution, which is initially "empty", i.e., $\sigma := \perp$. When executed, an algorithm $A$ awaits an input at one of its interfaces, and upon such input, it is activated, possibly updates the current state, and finally produces an output at one of its interfaces. Occasionally, we run the algorithm $A$ initializing its initial state to some (compatible) value $\sigma$ other than $\perp$, and subsequently let its execution proceed accordingly: In this case, we denote the resulting algorithm as $A[\sigma]$.

Each algorithm $A$ has a well-defined input-output behavior, and can appear in expression involving systems as a placeholder for the system or construction it realizes. In particular, an algorithm $A$ implements a system $S$ if $A \equiv S$ holds.

Oracle Algorithms. Frequently, we would like to enrich the computational capabilities of an algorithms $A$ by means of an accessible system $O$, called an oracle. We denote the resulting algorithm as $A^O$. (Analogously, we may let a construction also access such an oracle.)

We remark that formally an oracle is nothing but a subsystem accessed by the algorithm (or by the construction), yet the ad-hoc notation is intended to highlight the role of the oracle as a computational support.

Complexities. Complexity-theoretic statements are always with respect to a RAM model of computation for implementing an algorithm $A$ as described above. In particular, we allow algorithms to store values with arbitrary precision and also to perfectly sample uniform elements from sets as long as these sets are sufficiently dense in some set of bit strings, and membership can efficiently be tested.\footnote{This will be tacitly assumed in the following, as we want to dispense with a more formal model to the largest extent.}

We say that an algorithm $A$ has time complexity $t_A : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ if for all $q, s \geq 0$, the sum of the description length of $A$ and of the number of steps taken by $A$ for all sequences of $q$ inputs to the main interface of the algorithm, all compatible interaction with given sub-algorithms / systems / oracles, and all compatible initial states of length $s$, is at most $t_A(q, s)$. We use the shorthand $t_A(q) := t_A(q, 0)$. Furthermore, it has
space complexity $s_A : \mathbb{N} \rightarrow \mathbb{N}$ if the size of its state is at most $s_A(q)$ for all sequences of $q$ inputs to the main interface of the algorithm and all compatible interaction with given sub-algorithms / systems / oracles.

Remark 2.1. Algorithms may also accept inputs of arbitrary length, in which case the above functions $t_A$ and $s_A$ are extended by an additional parameter, describing the total length of the inputs $\ell$, and in the definition of $t_A(q, \ell, s)$ and $s_A(q, \ell)$, the maxima are taken over all sequences of $q$ queries to the main interface of total length $\ell$.

2.5.2 Examples

We consider canonical implementations for some recurring systems.

Example 2.13. Let $R : \mathcal{X} \rightarrow \mathcal{Y}$ a URF, for finite sets $\mathcal{X}$ and $\mathcal{Y}$ (usually, $\mathcal{X} = \{0, 1\}^m$ and $\mathcal{Y} = \{0, 1\}^n$). The canonical (stateful) implementation $A_R$ keeps a list of pairs $L \subseteq \mathcal{X} \times \mathcal{Y}$, and upon a query $x \in \mathcal{X}$, it looks whether there exists $(x, y) \in L$ for some $y \in \mathcal{Y}$: If yes, it returns $y$, otherwise it picks $y' \leftarrow \mathcal{Y}$ and adds $(x, y')$ to $L$, and returns $y'$.

The space complexity is

$$s_{A_R}(q) := O(q \cdot (\log |\mathcal{X}| + \log |\mathcal{Y}|)).$$

By managing $L$ as a dictionary data structure (indexed over the first component of the pairs, according to some efficiently decidable ordering of $\mathcal{X}$), queries are answered in time $O(\log s)$ if the current state has size $s$, and therefore we have

$$t_{A_R}(q, s) := O(q \cdot \log(s + q)).$$

An analogous implementation for the case $\mathcal{X} = \{0, 1\}^*$ can be given.

Example 2.14. Similarly, an implementation of a URP $P : \mathcal{X} \rightarrow \mathcal{X}$ keeps a set $L \subseteq \mathcal{X} \times \mathcal{X}$ of input-output pairs. The only difference is that sampling of a new element $y' \leftarrow \mathcal{Y}$ is replaced by random sampling of an element $y' \leftarrow \mathcal{X} \setminus \{x' \in \mathcal{X} \ : \exists x'' : (x', x'') \in L\}$, which we assume to be perfectly realizable. This can be implemented with the same complexities as the URF case. Also, the implementation of a two-sided URP $\langle P \rangle$ with the same complexity is analogous.

2.5.3 Asymptotic Statements and Efficiency

Cryptographic statements are almost always asymptotic, i.e., they are made dependent on an open parameter $k$ of the cryptographic system of
2.5 Algorithms and Complexities

interest, called the security parameter. Examples are the key length or the output length. In other words, one consider families of systems \( \{S_k\}_{k \in \mathbb{N}} \) of distinguishers \( \{D_k\}_{k \in \mathbb{N}} \) of constructions \( \{C_k\}_{k \in \mathbb{N}} \), etc, where the corresponding elements from the families are used depending on the value of the parameter. Asymptotics will be mostly implicit in this thesis: In particular, statements will be about individual systems, and the corresponding family will only be implicit.

There are two possible approaches to an algorithmic implementation of a family of systems / constructions:

Non-uniform algorithms. A non-uniform algorithm consists of a family of algorithms \( \{A_k\}_{k \in \mathbb{N}} \), i.e., one algorithm per each value of the security parameter. Its time complexity \( t_A : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N} \) and space complexity \( s_A : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N} \) are such that \( t_A(k, \cdot, \cdot) \) and \( s_A(k, \cdot) \) are the time and space complexities of \( A_k \), respectively.

Uniform algorithms. A uniform algorithm is a single algorithm \( A \) with the value of the security parameter \( k \) as an auxiliary input, and which gives rise to a family of algorithms \( \{A_k\}_{k \in \mathbb{N}} \) where \( A_k \) is obtained by running the algorithm \( A \) on input \( k \). Also, the time and space complexities \( t_A \) and \( s_A \) are usually additionally parameterized by the parameter \( k \), i.e., \( t_A(k, \cdot, \cdot) \) and \( s_A(k, \cdot) \) are the corresponding complexities when the parameter is \( k \).

The following definition captures the notion of efficiency in the asymptotic setting.

**Definition 2.7.** A family of algorithms \( A = \{A_k\}_{k \in \mathbb{N}} \) (either non-uniform or uniform) is efficient (or polynomial-time) if for all polynomial functions \( q, s : \mathbb{N} \rightarrow \mathbb{N} \), the function \( k \mapsto t_A(k, q(k), s(k)) \) is polynomial. A system / construction is efficient (or polynomial-time, or efficiently implementable) if it supports an efficient implementation.

Non-uniform algorithms can exploit some security-parameter dependent advice (hard-coded in the algorithm description), which is not necessarily computable by a uniform algorithm. In the context of this thesis, we are mostly interested in having uniform algorithms implementing the constructions (allowing for major flexibility for the honest users), whereas we assume adversarial entities to be non-uniform (which is a worst-case assumption).

Still, this being an interesting foundational question, we will also provide proofs (or at least proof sketches) showing how to translate our main
statements to the uniform model, i.e., to work with security against uniform adversaries. This will require considerably more involved security reductions.

2.6. Pseudorandom Systems

2.6.1. Computational Indistinguishability

We let \( D_t \) be the class of distinguishers \( D \) with time complexity \( t \) (i.e., which support an implementation with time complexity \( t \)), while \( D_{t,q} \subseteq D_t \) is its restriction to distinguishers issuing at most \( q \leq t \) queries, and \( D_{t,q,\ell} \subseteq D_{t,q} \) is the additional restriction to those for which the total length of all queries is bounded by \( \ell \geq q \). Then, we define the advantages \( \Delta_t(S,T) \), \( \Delta_{t,q}(S,T) \), and \( \Delta_{t,q,\ell}(S,T) \) as the respective maximal \( \Delta^D(S,T) \) over the three above classes.\(^9\) Also, in the asymptotic setting, the quantity \( \Delta_{\text{poly}}(S,T) \) always refers to the maximum taken over all polynomial-time distinguishers.\(^10\)

2.6.2. Pseudorandom Systems and Cryptographic Functions

Pseudorandom systems are efficient computational approximations of ideal systems such as uniform random functions and uniform random permutations that only use a short random secret string (usually a secret key).

**Definition 2.8.** Let \( F(\cdot) \) be an efficiently implementable deterministic construction accessing a random variable \( S \in \mathcal{S} \) such that \( F(s) \) is a function for all \( s \in \mathcal{S} \), and let \( I \) be a system. Furthermore, for given \( t,q \in \mathbb{N} \) and \( \varepsilon \in [0,1) \),

\[
\Delta_{t,q}(F(S), I) \leq \varepsilon.
\]

Then \( F(\cdot) \) (and often, with some abuse of notation, \( F(S) \)) is called:

- A \((t,q,\varepsilon)\)-pseudorandom function (PRF) \( \mathcal{X} \to \mathcal{Y} \) if \( F(S) \) is a random function \( \mathcal{X} \to \mathcal{Y} \) and \( I \) is a URF \( R : \mathcal{X} \to \mathcal{Y} \).

- A \((t,q,\varepsilon)\)-pseudorandom permutation (PRP) on \( \mathcal{X} \) if \( F(S) \) is a random permutation \( \mathcal{X} \to \mathcal{X} \) and \( I \) is a URP \( P : \mathcal{X} \to \mathcal{X} \).

\(^9\)Despite notational overloading, it will always be clear from the context whether we bound with respect to the time complexity or to the number of queries.

\(^10\)This will usually refer to uniform distinguishers, unless otherwise explicitly stated, as we usually provide results with concrete parameters in the non-uniform setting.
2.6 Pseudorandom Systems

- A \((t, q, \varepsilon)\)-two-sided pseudorandom permutation (two-sided PRP) on \(\mathcal{X}\) if \(F(S)\) is a two-sided random permutation \(\mathcal{X} \rightarrow \mathcal{X}\) and \(I\) is a two-sided URP (PRP) : \(\mathcal{X} \rightarrow \mathcal{X}\).

- A \((t, \varepsilon)\)-pseudorandom generator (PRG) with range \(\mathcal{X}\) if \(F(S)\) is a random variable with range \(\mathcal{X}\), \(I\) is a uniform random string on \(\mathcal{X}\), and \(|\mathcal{X}| > |S|\).\(^{11}\)

\textit{Remark 2.2.} In the case of a PRF with arbitrary input set \(\mathcal{X} = \{0, 1\}^*\), we talk of a \((t, q, \ell, \varepsilon)\)-PRF if \(\Delta_{t,q,\ell}(F(S), I) \leq \varepsilon\).

In the asymptotic setting, a construction \(F(\cdot)\) is an \(\varepsilon\)-PRF if it is a \((t, q, \varepsilon)\)-PRF for all polynomially bounded \(t\) and \(q\), and it is a PRF if it is an \(\varepsilon\)-PRF for a negligible \(\varepsilon\).

We denote the evaluation of \(F(S)\) on input \(x\) as \(F(S)(x)\). Also, in most cases, pseudorandom systems are implemented as a system \(F(K, \cdot)\) from a function\(^{12}\) \(F : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}\) and \(K\) is a uniformly chosen secret key from \(\mathcal{K}\). (Such systems are cc-stateless.) We say that such a function \(F\) is a PRF / PRP / PRG if \(F(K, \cdot)\) is a PRF / PRP / PRG.

\textit{Remark 2.3.} Most results of this thesis that apply to pseudorandom systems will be usually formulated in a far more general setting, i.e., they will apply to all efficiently implementable random functions, regardless of whether they are efficiently realized from some short randomness. Also, with the exception of statements in the uniform setting, we will give all results with concrete, rather than asymptotic, parameters: Recovering the corresponding results for pseudorandom systems and/or in the asymptotic setting will be obvious.

2.6.3. Restricted Distinguishing Attacks

Indistinguishability-based security definitions can also be weakened by restricting the distinguisher’s access to the given system. For instance, the standard PRF notion considering an (adaptive) \textit{chosen-input attack} can be weakened to non-adaptive chosen-input attacks or even to \textit{(known) random-input} attacks. (Keyed functions which are secure under the latter notion are usually called \textit{weak PRFs} [NR99] in the literature, even though we will avoid this naming.\(^{13}\)) This is conveniently modeled by letting

\(^{11}\)Usually, we have \(S := \{0, 1\}^k\) and \(X := \{0, 1\}^\ell\) for \(k < \ell\).

\(^{12}\)In practice \(F\) may arise from a block cipher, or from the compression function of a hash function.

\(^{13}\)The name is slightly misleading within the context of this thesis, as we consider multiple approaches to weakening PRFs.
the distinguisher access either of \( E(F) \) and \( E(G) \), where the construction \( E(\cdot) \) enforces a particular type of access, and \( F \) and \( G \) are the random functions to be distinguished. For a chosen-input attack, the construction would just give full access to the underlying system (i.e. \( E(\cdot) \) is the identity), and the following are three additional examples:

- **Random-input attacks** against an \((X, Y)\)-system are modeled by \( K(\cdot) \) that, upon each invocation (with some dummy input), generates a fresh uniformly-chosen element \( r \in X \), makes a query with input \( r \) to the given subsystem, obtaining \( y \in Y \), and returns \( (r, y) \).

- For a quasi-group operation \( * \) on \( X \) (usually \( \oplus \)), a **random-offset attack** is modeled by a construction \( Z(\cdot) \) which initially generates a random offset \( Z \in X \), and upon each invocation with input \( x \in X \), makes a query to the given subsystem with input \( x*Z \), and outputs the returned value \( y \).

- A **non-adaptive** chosen-input attack against an \((X, Y)\)-system can be modeled by means of a construction \( N(\cdot) \) which only accepts one query consisting of a tuple of values \( (x_1, \ldots, x_q) \in X^q \), and subsequently asks all these \( q \) queries to the given system, and returns the tuple of corresponding outputs \( (y_1, \ldots, y_q) \in Y^q \).

We call an efficient construction \( F(\cdot) \) an \((E, t, q, \varepsilon)\)-PRF if

\[
\Delta_{t,q}(E(F(S)), E(R)) \leq \varepsilon
\]

for a URF \( R \). Also, we usually refer to \( K \)-PRFs as *random-input PRFs* (RI-PRFs) and to \( N \)-PRFs as *non-adaptive PRFs*. 
Chapter 3

Weakly Unpredictable Bits

Consider a pair $(X, B)$ that consists of a random variable $X$ and a correlated random bit $B$ that can be guessed with probability at most $\frac{1+\varepsilon}{2}$, for some $\varepsilon \in [0, 1)$, given the value of $X$. For $m$ independent copies $(X_1, B_1), \ldots, (X_m, B_m)$ of $(X, B)$, it is well known that the bit $B_1 \oplus \cdots \oplus B_m$ can only be guessed with probability at most $\frac{1+\varepsilon^m}{2}$ by a computationally unbounded adversary given $X_1, \ldots, X_m$. This statement can be seen as the simplest form of indistinguishability amplification, as in the case of bits, the notions of unpredictability, on the one hand, and of indistinguishability from a uniform random bit, on the other hand, essentially coincide.

The most natural approach to derive the above upper bound exploits the fundamental property that we can always adjoin an event $A$ to the choice of $(X, B)$, which occurs with probability $1 - \varepsilon$, and such that for all values $x$ in the range of $X$,

$$P_{B|X,A}(0, x) = P_{B|X,A}(1, x) = \frac{1}{2}.$$  

As long as the corresponding (independent) events occur for at least one of the pairs $(X_1, B_1), \ldots, (X_m, B_m)$ (this is true with probability $1 - \varepsilon^m$), the bit $B_1 \oplus \cdots \oplus B_m$ is truly random and can only be guessed with probability $\frac{1}{2}$. 
A *computational* analogue of this statement has been first proven by Impagliazzo [Imp95], and is usually referred to as the **Hardcore Lemma (HCL):** Under the assumption that an *efficient* adversary can only guess $B$ with probability at most $\frac{1+\varepsilon}{2}$ given $X$, the lemma shows the existence of an event $A$, occurring with probability $1 - \varepsilon$, under which it is *computationally* hard to guess $B$ from $X$ with better than negligible advantage over random guessing. Consequently, in analogy to the information-theoretic case, the HCL implies that the probability that an efficient adversary guesses the exclusive-or $B_1 \oplus \cdots \oplus B_m$ is upper bounded by $\frac{1+\varepsilon^m+\nu}{2}$, for a negligible quantity $\nu$. This is usually referred to as Yao’s **XOR Lemma** [Yao82].

**Chapter Outline and Contributions.** This chapter considers and introduces a generalization of the above setting to interactive systems: We consider pairs $(S, B)$ – called *system-bit pairs* – consisting of a system $S$ whose behavior depends on a bit $B$, and we study their properties in the context of a game where an adversary attempts to guess $B$ given access to $S$. Our main technical contributions are generalizations of both the Hardcore Lemma and Yao’s XOR Lemma, which we lift to the interactive setting of system-bit pairs.

More in detail, the contents of this chapter are organized as follows:

(i) Section 3.1 introduces the concept of a system-bit pair, and of the associated guessing advantage of an adversary. The treatment follows the lines of [MT09], where these notions were first introduced. Furthermore, we discuss basic properties of the guessing advantage, and its relation with the distinguishing advantage, in detail.

(ii) In Section 3.2.1, we first discuss characterizations of system-bit pairs in the *information-theoretic* setting. Roughly speaking, we show (Theorem 3.3) that every system-bit pair $(S, B)$ admits an MBO such that the probability of provoking the MBO to be one equals the (information-theoretic) advantage of guessing $B$ when interacting

---

1. A distinction between computationally bounded and unbounded adversaries is only meaningful provided that $X$ and $B$ are correlated, as otherwise the best predictor has constant running time (i.e., outputs the most likely outcome).

2. This result shall not be confused with another “XOR Lemma”, attributed to U.M. Vazirani, stating that a bit string is nearly uniform provided that all linear combinations of subsets of bits are nearly uniform, and which also admits a computational analogue, due to U.M. Vazirani and V.V. Vazirani. We refer to [Gol95] for an exposition.
with $S$, and such that as long as this MBO is zero, no adversary can guess the value of $B$ better than with random guessing. This result should not be considered a contribution of this thesis (while this result was never stated elsewhere, it uses very similar techniques to the main technical lemma of [MPR07]), but it will serve as a motivation for the computational results of this chapter.

(iii) Following the information-theoretic intuition, the remainder of Section 3.2 is devoted to the first main technical result of this chapter, the Hardcore Lemma (HCL) for cc-stateless system-bit pairs, a computational analogue of Theorem 3.3. We provide both a non-uniform and a uniform version of the lemma. This result has first appeared in [Tes10].

(iv) In Section 3.3, we move on to the second main result of this chapter, the XOR Lemma for cc-stateless system-bit pairs. We start by discussing a natural proof based on the HCL, which follows the information-theoretic intuition. Moreover, we prove an additional (stronger) version with a direct proof, which was originally proven in [MT09].

Related Work. The concept of computationally hard bits (given some side information) was introduced by Blum and Micali [BM82] in the context of hardcore predicates for one-way functions (cf. also Example 3.4 below for a formal definition). Hardcore predicates are essential in a number of fundamental results in cryptography as the mean to turn hardness into pseudorandomness, most notably in constructions of pseudorandom generators from one-way permutations [Yao82] and one-way functions [Lev87, GKL93, HILL99, Hol06, HHR06b, HHR06a, HRV10], as well as in the design of public-key encryptions schemes, as first proposed in the seminal work by Goldwasser and Micali [GM84]. The first universal construction of hardcore predicates for arbitrary one-way functions is due to Goldreich and Levin [GL89].

Yao [Yao82] first considered\(^3\) the question of strengthening mildly-hard bits via xor-ing the bits (hence the name “Yao’s XOR Lemma”). The first published proof of the XOR Lemma is due to Levin [Lev87] for the special case of hardcore predicates of one-way functions. An overview of proofs of the XOR Lemma was given by Goldreich, Nisan, and Wigderson [GNW95]. The simplest (and most natural) proof is given via the

\(^3\)In fact, this result is generally attributed to the oral presentation of [Yao82].
Hardcore Lemma (HCL), which was introduced and first proved by Impagliazzo [Imp95]. (A tight version, with a uniform proof, has been given by Holenstein [Hol05], which was recently further improved by Barak, Hardt, and Kale [BHK09].) A variant of the XOR Lemma in the setting of sequentially-composed interactive protocols has been proven by Halevi and Rabin [HR08].

A result closely related to the XOR Lemma, first proven by Goldreich et al. [GNW95], is the Direct Product Theorem (DPT), i.e., the statement that the probability of guessing simultaneously multiple mildly-hard bits $B_1, \ldots, B_m$ (given $X_1, \ldots, X_m$) decreases exponentially in $m$. (A simple proof of the DPT follows from the HCL.) The DPT also yields a proof of the XOR Lemma via the Goldreich-Levin Theorem [GL89], and, conversely, Unger [Ung09] recently showed how to prove the DPT from the XOR Lemma.4

Several generalization of the direct product theorem for cryptographic puzzles have been presented in the literature. In particular, the first DPT for weakly verifiable cryptographic puzzles (which for instance directly implies conventional security amplification results for one-way functions and three-round interactive arguments) has been given by Canetti, Halevi, and Steiner [CHS05]. This was further generalized to the threshold [IJK07, Jut10, CLLY10] and to the interactive [DIJK09] cases.

We also point out that a new generic approach to security amplification of puzzles and of hard bits which is amenable to the setting of sequential composition of cryptographic protocols has been recently proposed by Holenstein and Schoenebeck [HS10].

Outside the mere cryptographic context, the XOR Lemma and the DPT are both viable approaches in the problem of amplifying the hardness of a function $f : X \rightarrow \{0, 1\}$. This setting is more difficult than the cryptographic one, as pairs $(X, f(X))$ are not efficiently samplable, which hinders uniform reductions in most cases. In particular, an important parameter in this setting is the amount of non-uniformity (i.e., the length of the advice) required by the reduction in a proof of these lemmas. Substantial progress has been done in improving on Yao’s XOR Lemma by means of better results, most notably by Impagliazzo and Wigderson [IW97], Sudan, Trevisan and Vadhan [STV01], and Trevisan [Tre03, Tre05]. In particular, the connection between hardness amplification and list decoding of

4Also, [Ung09] shows how to apply his technique to the XOR Lemma presented in this chapter to obtain a DPT for cc-stateless system-bit pairs. The same result could be derived from our HCL for system-bit pairs.
error-correcting codes has been explored in many of these works, first explicitly in [Tre03]. Fully uniform product theorems (i.e., with advice taking at most polynomially many values) have been given by Impagliazzo, Jaiswal, and Kabanets [IJK06] and by Impagliazzo, Jaiswal, Kabanets, and Wigderson [IJKW10]. Finally, Shaltiel and Viola [SV08] have also investigated the complexity of black-box hardness-amplification proofs.

3.1. System-Bit Pairs

3.1.1. Definition and Examples

Throughout this chapter, we consider objects of the following type.

**Definition 3.1.** A *system-bit pair* is a pair \((S, B)\) consisting of a system \(S\) and a random bit \(B\), such that the behavior of \(S\) depends on the value of the bit \(B\).

The following are some examples of system-bit pairs.

*Example 3.1.* For a function \(f : \{0, 1\}^m \to \{0, 1\}^n\), and for independent and uniformly distributed \(m\)-bit strings \(X, R\), we can consider the system-bit pair \(((f(X), R), X \cdot R)\), where \(x \cdot r\) denotes the scalar product (modulo 2) of two \(m\)-bit strings \(x\) and \(r\).

*Example 3.2.* Another example is the pair \((R, \text{parity}(R))\), where \(R\) is a URF \(\{0, 1\}^m \to \{0, 1\}^n\) and \(\text{parity}(R) \in \{0, 1\}\) is the parity of the function table of \(R\) when interpreted as an \((n \cdot 2^m)\)-bit string.

*Example 3.3.* For two \((X, Y)\)-systems \(S\) and \(T\), and a bit \(B\) with some given distribution, not necessarily uniform, we can consider the system-bit pair \(((S, T))_{B, B}\).

We stress that the outcome of the bit \(B\) can be seen as being defined before any interaction with \(S\) has taken place. In other words, it is convenient to see the system-bit pair \((S, B)\) as a system consisting of the parallel composition of a bit \(B\) and the (correlated) system \(S\), and concepts defined for systems directly extend to system-bit pairs. For instance, if \(S\) and \(B\) can be represented as depending on an initial state \(S\), we write \((S, B) \equiv (S(S), B(S))\), and the notion of a cc-stateless system-bit pair is defined accordingly.

\*The composition operator \(\parallel\) cannot be used here, as it assumes independent systems.*
Remark 3.1. The fact that \((S, B)\) is cc-stateless implies that \(S\) is cc-stateless, but it is not always true that \((S, B)\) is cc-stateless if \(S\) is cc-stateless: Consider the system-bit pair \((S, B)\) such that \(B\) is a uniform random bit, and \(S\) has one-bit inputs and outputs: It answers the first query \(x_1\) with a random bit \(y_1\), and the second query \(x_2\) is answered by \(x_1 \oplus B\). All remaining queries are answered by independent random bits. Clearly, \(S\) by itself is even stateless (all answers are random and independent), however \((S, B)\) is not cc-stateless, since \(y_2 = x_1 \oplus B\) must hold.

3.1.2. Guessing Advantage

Every system-bit pair \((S, B)\) induces a game where an adversary \(A\) interacts with \(S(B)\) and, after a certain number of queries, tries to guess the value of the bit \(B\) by outputting a bit \(A(S(B)) \in \{0, 1\}\). In particular, an adversary, like a distinguisher, is a binary-output user. The names distinguisher and adversary are used depending on what appears more convenient for the considered setting. In particular, we will use the notations \(D_t, D_q\), and \(D_{t,q}\) in this case, rather than \(D_t, D_q\), and \(D_{t,q}\).

The following definition characterizes the quality of an adversary in winning the game.

Definition 3.2. The guessing advantage of the adversary \(A\) for the system-bit pair \((S, B)\) is defined as

\[
\text{Guess}^A(B \mid S) := 2 \cdot \text{P}[A(S(B)) = B] - 1.
\]

Furthermore, for integers \(t, q > 0\), we let \(\text{Guess}_t(B \mid S)\), \(\text{Guess}_q(B \mid S)\), and \(\text{Guess}_{t,q}(B \mid S)\) be defined as the best advantage \(\text{Guess}^A(B \mid S)\) taken over all \(A\) in \(A_t, A_q\), and \(A_{t,q}\), respectively.

Note that \(\text{Guess}^A(B \mid S) \in [-1, 1]\), where 1 means that \(A\) perfectly predicts \(B\), whereas -1 means that \(A\) is never correct.\(^6\) Furthermore \(\text{Guess}_q(B \mid S) \geq 0\) and \(\text{Guess}_{t,q}(B \mid S) \geq 0\), since we assume that the classes \(A_q\) and \(A_{t,q}\) are close under flipping the output bit. Finally, in the asymptotic setting, we also let \(\text{Guess}_{\text{poly}}(B \mid S)\) be the best advantage of a polynomial-time adversary.\(^7\)

\(^6\)However, flipping its output bit yields an adversary which is always correct.

\(^7\)As in the case of the distinguishing advantage, this usually refers to uniform adversaries.
Example 3.4. Let \( f : \{0, 1\}^m \rightarrow \{0, 1\}^n \) be a function, and let \( P : \{0, 1\}^m \rightarrow \{0, 1\} \) be a predicate. For a uniformly distributed \( X \in \{0, 1\}^m \), we consider the system-bit pair \((f(X), P(X))\): Guess_{poly}(P(X) | f(X)) being negligible is equivalent to \( P \) being a hardcore predicate of \( f \).

3.1.3. Properties of the Guessing Advantage

The following lemma establishes, in the setting of system-bit pairs, the well-known equivalence between the facts that the guessing advantage for a bit is small and the fact that the bit “looks random” given access to \( S \).

Lemma 3.1. Let \( U \) be a uniformly distributed bit (independent of the input-output behavior of any other system). Then:

(i) For all adversaries \( A \in A_{t,q} \) there exists a distinguisher \( D^* \in D_{t+O(1),q} \) such that for all system-bit pairs \((S, B)\),

\[
\left| \text{Guess}^A(B | S) \right| = 2 \cdot \Delta^D ((S, B), (S, U)).
\]

(ii) For all distinguishers \( D \in D_{t,q} \) there exists an adversary \( A^* \in A_{t+O(1),q} \) such that for all system-bit pairs \((S, B)\),

\[
\left| \text{Guess}^{A^*}(B | S) \right| = 2 \cdot \Delta^D ((S, B), (S, U)).
\]

In particular \( \text{Guess}_{t,q}(B | S) = 2 \cdot \Delta_{t+O(1),q}((S, B), (S, U)) \).

Proof. To prove (i), simply define the distinguisher \( D^* \) such that, given access to \((S, \tilde{B})\), with \( \tilde{B} \) being either \( B \) or the independent bit \( U \), lets \( A \) interact with \( S \), obtaining output \( B' \), and outputs 1 if and only if \( B' = \tilde{B} \), and outputs 0 otherwise. Note that

\[
\mathbb{P} [D^*((S, B)) = 1] = \mathbb{P} [A(S) = B] = \frac{1 + \text{Guess}^A(B | S)}{2}
\]

whereas \( \mathbb{P} [D^*(S, U) = 1] = \frac{1}{2}, \) since \( U \) is independent.

For (ii), we first observe that

\[
\Delta^D ((S, B), (S, U)) = \mathbb{P} [D((S, B)) = 1] - \mathbb{P} [D((S, U)) = 1]
\]

\[
= \mathbb{P} [D((S, B)) = 1] - \frac{1}{2} \cdot \mathbb{P} [D((S, U)) = 1 | U = B]
\]

\[
- \frac{1}{2} \cdot \mathbb{P} [D((S, U)) = 1 | U \neq B]
\]

\[
= \frac{1}{2} \cdot \mathbb{P} [D((S, U)) = 1 | U = B] - \mathbb{P} [D((S, U)) = 1 | U \neq B],
\]

since \( P[D((S, B)) = 1] = P[D((S, U)) = 1 | U = B] \). We let \( A^* \) be the adversary which samples a random bit \( U \) and computes \( B' := D((S, U)) \). If \( B' = 1 \), then it outputs \( U \), and otherwise outputs \( 1 - U \). The probability that it succeeds is

\[
P[A^*(S) = B] = P[U = B] \cdot P[D((S, U)) = 1 | U = B] \\
\quad \quad + P[U \neq B] \cdot P[D((S, U)) = 0 | U \neq B]
\]

\[
= \frac{1}{2} + \frac{1}{2} \cdot (P[D((S, U)) = 1 | U = B] - P[D((S, U)) = 1 | U \neq B]),
\]

which implies \( |\text{Guess}^{A^*}(S | B)| = 2 \cdot \Delta^D((S, B), (S, U)). \)

The following lemma expresses, in our formalism, the well-known equivalent characterization of the distinguishing advantage in terms of the advantage of guessing which one of two systems an adversary is given access to.

**Lemma 3.2.** For a uniform random bit \( B \), any two systems \( S \) and \( T \), and all distinguishers \( D \),

\[
\Delta^D(S, T) = |\text{Guess}^D(B | \langle S, T \rangle_B)|.
\]

In particular, \( \Delta_{t,q}(S, T) = \text{Guess}_{t,q}(B | \langle S, T \rangle_B) \).

**Proof.** By a straightforward calculation,

\[
\Delta^D(S, T) = |P[D(S) = 0] - P[D(T) = 0]| \\
\quad \quad = |2 \cdot (\frac{1}{2} \cdot P[D(S) = 0] + \frac{1}{2} \cdot P[D(T) = 1]) - 1| \\
\quad \quad = |2 \cdot P[D((S, T)_B) = B] - 1| = |\text{Guess}^D(B | \langle S, T \rangle_B)|.
\]

**3.2. The Hardcore Lemma for System-Bit Pairs**

This section presents the first technical contribution of this chapter, the Hardcore Lemma (HCL) for \( cc \)-stateless system-bit pairs. In Section 3.2.1, we briefly discuss an information-theoretic version of the HCL, which is of independent interest. The remaining subsections are devoted to the HCL for system-bit pairs, for which we provide proofs both in the non-uniform as well as in the uniform settings.
3.2.1. The Information-Theoretic Hardcore Lemma

Let \((X, B)\) be a pair consisting of a random variable \(X\) (with range \(\mathcal{X}\)) and a (correlated) bit \(B\) such that

\[
\text{Guess}^A(B \mid X) = \varepsilon
\]

for all adversaries \(A\). We define the event \(A\) on \((X, B)\) such that

\[
P_{XBA}(x,b) := \min\{P_{XB}(x,0), P_{XB}(x,1)\}
\]

for all \(x \in \mathcal{X}\) and \(b \in \{0, 1\}\). (Also cf. Figure 3.1 for a concrete example of a distribution \(P_{XB}\) and the corresponding construction of the event \(A\).) Clearly, dividing both sides of \(P_{XBA}(x,0) = P_{XBA}(x,1)\) by \(P_{XA}(x)\) yields

\[
P_{B \mid X,A}(0,x) = P_{B \mid X,A}(1,x)
\]

for all \(x \in \mathcal{X}\): In other words, conditioned on the event \(A\) occurring, it is impossible to guess the outcome of \(B\) from \(X\) with probability higher than \(\frac{1}{2}\). Furthermore,

\[
P[A] = \sum_{x \in \mathcal{X}, b \in \{0, 1\}} P_{XBA}(x,b)
\]

\[
= \sum_{x \in \mathcal{X}, b \in \{0, 1\}} \min\{P_{XB}(x,0), P_{XB}(x,1)\}
\]

\[
= \sum_{x \in \mathcal{X}, b \in \{0, 1\}} P_{XB}(x,b) - \sum_{x \in \mathcal{X}} |P_{XB}(x,0) - P_{XB}(x,1)|
\]

\[
= 1 - \sum_{x \in \mathcal{X}} |P_{XB}(x,0) - P_{XB}(x,1)|,
\]

and the best strategy for guessing the bit \(B\) from \(X\) (usually called the maximum-likelihood strategy) chooses, for all \(x \in \mathcal{X}, b \in \{0, 1\}\) such that \(P_{XB}(x,b) \geq P_{XB}(x,1 - b)\): It achieves success probability

\[
\frac{1 + \varepsilon}{2} = \sum_{x \in \mathcal{X}} \max\{P_{XB}(x,0), P_{XB}(x,1)\}
\]

\[
= \sum_{x \in \mathcal{X}} \frac{P_{X}(x) + |P_{XB}(x,0) - P_{XB}(x,1)|}{2}
\]

\[
= \frac{1}{2} + \frac{1}{2} \sum_{x \in \mathcal{X}} |P_{XB}(x,0) - P_{XB}(x,1)|,
\]
Figure 3.1: Illustration of the construction of the event $A$ for $(X, B)$, where $P_{XB}(0, 0) = \frac{1}{8}$, $P_{XB}(0, 1) = \frac{3}{8}$, $P_{XB}(1, 0) = \frac{1}{6}$, and $P_{XB}(1, 1) = \frac{2}{6}$. The gray area is the probability $P[A]$.

which implies $P[A] = 1 - \varepsilon$.

The above can be generalized to arbitrary system-bit pairs $(S, B)$ using techniques similar to the ones of Lemma 5 in [MPR07].\footnote{With some extra work, Theorem 3.3 and Lemma 5 in [MPR07] can be shown to be equivalent. Lemma 5 will be indeed restated (as Theorem 5.1) in Chapter 5 and proven from Theorem 3.3. Our choice of presenting Theorem 3.3 first is dictated by the structure of computational results of this thesis.} We will refer to this result as the information-theoretic hardcore lemma.

**Theorem 3.3.** Let $(S, B)$ be a system-bit pair. Then, there exists a system-bit pair $(\hat{S}, \hat{B})$ with an MBO $A_1, A_2, \ldots$ such that

(i) $(\hat{S}, \hat{B})^- \equiv (S, B)$;

(ii) $p_{Y^q, A_q=0, \hat{B}=0|X^q}(\hat{S}, \hat{B}) = p_{Y^q, A_q=0, \hat{B}=1|X^q}(\hat{S}, \hat{B})$ for all $q \geq 1$;

(iii) $\nu_q(\hat{S}) = \text{Guess}_q(B | S)$ for all $q \geq 1$.

**Proof.** We generalize the above approach from random variables to systems. For all $q \geq 1$, and for all input-output sequences $(x^q, y^q) \in X^q \times Y^q$, we define

$$m_{x^q, y^q} := \min \left\{ p_{Y^q, B|X^q}(y^q, 0, x^q), p_{Y^q, B|X^q}(y^q, 1, x^q) \right\},$$

$$\nu_q(\hat{S}) = \text{Guess}_q(B | S)$$
that for any $q$

Properties (i) and (ii) are easy to verify. For property (iii), we first note

where for the empty sequence $\perp$ we set $m_{\perp,\perp} := \min \{ p_B(0), p_B(1) \}$.

Below, we define a system-bit pair $(\widehat{S}, \widehat{B})$ which, for all $\widehat{b} \in \{0, 1\}$, for all $q \geq 1$, and all $(x^q, y^q) \in X^q \times Y^q$, satisfies

$$p_{Y \triangleright \hat{B}, A_q | X^q}(y^q, \widehat{b}, 0, x^q) = m_{x^q, y^q}$$

$$p_{Y \triangleright \hat{B}, A_q | X^q}(y^q, \widehat{b}, 1, x^q) = p_{Y \triangleright \hat{B}, \hat{S}, A_q | X^q}(y^q, \widehat{b}, x^q) - m_{x^q, y^q}.$$ 

Properties (i) and (ii) are easy to verify. For property (iii), we first note that for any $q$-query adversary $A$ interacting with $\widehat{S}$, the probability that the MBO satisfies $A_q = 0$ is

$$1 - \nu^A(\widehat{S}) = p_{A_q}(0)$$

$$= \sum_{(x^q, y^q)} p_{X \triangleright Y, A_q}(x^q, y^q, 0)$$

$$= \sum_{(x^q, y^q)} p_{X \triangleright Y, \widehat{b}, \hat{S}}(x^q, y^q-1 \cdot p_{Y \triangleright \widehat{B}, \hat{A}_q | X^q}(y^q, \widehat{b}, 0, x^q)$$

$$= \sum_{(x^q, y^q)} p_{X \triangleright Y, \widehat{b}, \hat{S}}(x^q, y^q-1 \cdot m_{x^q, y^q}$$

$$= \sum_{(x^q, y^q)} \min \left\{ p_{X \triangleright Y, \widehat{b}, \hat{S}}(x^q, y^q, 0), p_{X \triangleright Y, \widehat{b}, \hat{S}}(x^q, y^q, 1) \right\}$$

$$= \sum_{(x^q, y^q)} p_{X \triangleright Y, \widehat{b}, \hat{S}}(x^q, y^q, \widehat{b}) - S$$

$$= 1 - S,$$

where

$$S := \sum_{(x^q, y^q)} \left| p_{X \triangleright Y, \widehat{b}, \hat{S}}(x^q, y^q, 0) - p_{X \triangleright Y, \widehat{b}, \hat{S}}(x^q, y^q, 1) \right|.$$ 

Using $A$, we construct an adversary $A'$ for guessing $B$ from $S$, and then, given a transcript $(X^q, Y^q) = (x^q, y^q)$, outputs $b$ such that $p_{B \mid X \triangleright Y, \widehat{b}, \hat{S}}(b, x^q, y^q) \geq p_{B \mid X \triangleright Y, \widehat{b}, \hat{S}}(1 - b, x^q, y^q)$. It is correct with probability

$$P[A'(S) = B] = \sum_{(x^q, y^q)} \max \left[ p_{X \triangleright Y, \widehat{b}, \hat{S}}(x^q, y^q, 0), p_{X \triangleright Y, \widehat{b}, \hat{S}}(x^q, y^q, 1) \right]$$

$$= \frac{1}{2} \cdot \sum_{(x^q, y^q)} p_{X \triangleright Y, \widehat{b}, \hat{S}}(x^q, y^q) + \frac{S}{2} = \frac{1}{2} + \frac{S}{2} = \frac{1}{2} + \frac{\nu^A(\widehat{S})}{2},$$
where we have used that \( \max\{a, b\} = \frac{a+b+|a-b|}{2} \) for all \( a, b \in \mathbb{R} \). This implies \( \nu_q(\hat{S}) \leq \text{Guess}_q(B \mid S) \), and by inspection it is not hard to verify that the upper bound is indeed achieved.

It remains to show that there indeed exists a system-bit pair \((\hat{S}, \hat{B})\) with MBO \( A_1, A_2, \ldots \) for which \( p_{Y,B,A\mid X}^{(S,B)} \) is as defined above. In particular, we let \( p_{B,A_0}^{(S,B)}(\hat{b}, 0) = m_{\perp, \perp} \) and \( p_{B,A_0}^{(S,B)}(\hat{b}, 0) = p_{B}^{(S,B)}(\hat{b}) - m_{\perp, \perp} \) for all \( b \in \{0, 1\} \). (Note that \( A_0 \) is not part of the MBO, and is defined only to simplify notation.)

Furthermore, for all \( i \geq 1 \), all \( \hat{b} \in \{0, 1\} \), and all \( (x^i, y^i) \in X^i \times Y^i \), we let

\[
p_{Y,A_i \mid X_i,Y_{i-1},B_{A_{i-1}}}^{(S,B)}(y_i, 0, x^i, y^{i-1}, \hat{b}, 0^{i-1}) := \frac{m_{x^i,y^i}}{m_{x^i,y^{i-1}}}
\]

as well as

\[
p_{Y,A_i \mid X_i,Y_{i-1},B_{A_{i-1}}}^{(S,B)}(y_i, 1, x^i, y^{i-1}, \hat{b}, 0^{i-1}) := \frac{r_{x^i,y^i,\hat{b}} - m_{x^i,y^i}}{m_{x^i,y^{i-1}}},
\]

where \( r_{x^i,y^i,\hat{b}} = p_{Y_i,B_{A_{i-1}}}^{(S,B)}(y^i, \hat{b}, 0^{i-1}, x^i) \), and these values can be chosen arbitrarily, provided they satisfy \( r_{x^i,y^i,\hat{b}} \in \left[ m_{x^i,y^i}, p_{Y_i,B_i|X_i}^{(S,B)}(y^i, \hat{b}, x^i) \right] \)
and

\[
\sum_{y_i} r_{x^i,y^i,\hat{b}} = m_{x^i,y^{i-1}}.
\]

In particular, such values exist since

\[
\sum_{y_i} m_{x^i,y^{i-1}} | y_i | \leq m_{x^i,y^{i-1}} \leq p_{Y_i,B_i|X_i}^{(S,B)}(y^{i-1}, \hat{b}, x^i) = \sum_{y_i} p_{Y_i,B_i|X_i}^{(S,B)}(y^{i-1} | y_i, \hat{b}, x^i).
\]

On the other hand, for any non-zero monotone sequence \( a^{i-1} \in \{0, 1\}^{i-1} \),

\[
p_{Y,A_i \mid X_i,Y_{i-1},B_{A_{i-1}}}^{(S,B)}(y_i, 1, x^i, y^{i-1}, \hat{b}, a^{i-1}) := \frac{p_{Y_i,B_i|X_i}^{(S,B)}(y^i, \hat{b}, x^i) - r_{x^i,y^i,\hat{b}}}{p_{Y_i,B_i|X_i}^{(S,B)}(y^i, x^i, \hat{b}) - m_{x^i,y^i}}
\]

It can be verified that the given \((\hat{S}, \hat{B})\) is well defined. \(\square\)

In the remainder of this section, we will proceed in developing a computational version of Theorem 3.3.
3.2.2. Measures and State Samplers

The Hardcore Lemma for cc-stateless system-bit pairs is a computational analogue of Theorem 3.3 above. In contrast to the above information-theoretic case, however, instead of defining a MBO on the system bit pair \((S, B) \equiv (S(S), B(S))\), we will be able to define an event \(A\) on the initial state \(S\) of the cc-stateless system-bit pair (by means of a conditional probability distribution \(P_{A|S}\)) such that, conditioned on this event, guessing the associated bit is hard.

**MEASURES.** In the statement and proof of the Hardcore Lemma, it will be convenient to use the concept of a measure [Imp95] to describe the conditional probability distribution \(P_{A|S}\). In particular, we adapt this notion to cc-stateless systems: A measure \(M\) for \(S \equiv S(S)\), where \(S \in S\) is a random variable with probability distribution \(P_S\), is a mapping \(M : S \rightarrow [0, 1]\). Its *density* is defined as

\[
\mu(M) := \mathbb{E}[M(S)] = \sum_{s \in S} P_S(s) \cdot M(s).
\]

The measure \(M\) is naturally associated with a probability distribution \(P_M\) on \(S\) such that

\[
P_M(s) := \frac{P_S(s) \cdot M(s)}{\mu(M)}
\]

for all \(s \in S\). Also, we define the complement of a measure \(M\) as the measure \(\overline{M}\) such that \(\overline{M}(s) := 1 - M(s)\) for all \(s \in S\). We repeatedly abuse notation writing \(S \xleftarrow{\mu} M\) instead of \(S \xleftarrow{\mu} P_M\).

Even though, traditionally, measures are seen as “fuzzy” subsets of \(S\), if we think of \(M\) as the probability distribution \(P_{A|S}(s) := M(s)\) adjoining \(A\) to \(S\), we have \(\mu(M) = P[A]\), \(P_M = P_{S|A}\), and \(P_{\overline{M}} = P_{S|\overline{A}}\). Usually, we stick to measures for stating and proving the Hardcore Lemma, while the event-based view will be convenient when exercising these results.

**STATE SAMPLERS.** Ideally, the Hardcore Lemma for a cc-stateless system-bit pair \((S, B) \equiv (S(S), B(S))\) (for initial state \(S \in S\)) states that if

\[
\text{Guess}_{t,q}(B | S) \leq \varepsilon,
\]

then there exists a measure \(M\) for \(S\) such that

(i) \(\mu(M) \geq 1 - \varepsilon\), and
(ii) \( \text{Guess}_{t',q'}(B(S') \mid S(S')) \approx 0 \) for \( S' \xleftarrow{\$} M \) and \( t', q' \) as close as possible to \( t, q \).

Whenever \( S(S) \) is a random variable, this is equivalent to (a tight) version of the original Hardcore Lemma [Imp95, Hol06]. However, applications of the Hardcore Lemma, as for instance the proof of the XOR Lemma (Theorem 3.14) given below, often require the ability, possibly given some short advice, to efficiently simulate \( (S(S'), B(S')) \) for \( S' \xleftarrow{\$} M \) or \( (S(S''), B(S'')) \) for \( S'' \xleftarrow{\$} M \) within a security reduction.\(^9\) While in the context of random variables the advice is generally a sample of \( S' \) itself, this approach fails in the setting of interactive systems: Recall that the representation \( (S(S), B(S)) \) is possibly a mere thought experiment, and a description of \( S' \) may be of exponential size, or no efficient algorithm implementing \( (S, B) \) from \( S' \) may exist, despite the existence of an efficient stateful interactive algorithm implementing \( (S, B) \).

To overcome this issue, we introduce the notion of a state sampler for a cc-stateless system \( S \) (such as e.g. a system-bit pair) which formalizes the concept of an advice distribution.

**Definition 3.3 (State Samplers).** Let \( S \equiv S(S) \) be a cc-stateless system with implementation \( A_S \) and \( S \in S \), let \( \zeta_1, \zeta_2 \in [0, 1] \), and let \( \mathcal{M} : S \rightarrow [0, 1] \) be a measure for \( S \). A \((\zeta_1, \zeta_2)\)-(state) sampler \( O \) for \( \mathcal{M} \) and \( A_S \) with length \( \ell \) is a random process \( O \) with the following two properties:

(i) It always returns a pair \((\sigma, z)\) with \( \sigma \) being a valid state for \( A_S \) with \( |\sigma| \leq \ell \) and \( z \in [0, 1] \);

(ii) For \((\Sigma, Z) \xleftarrow{\$} O\), we have\(^{10}\)

\[
(As[\Sigma], Z) \equiv (S(S), Z'(S)),
\]

where \( Z'(S) \in [0, 1] \) is a random variable (which only depends on \( S \)) which differs from \( \mathcal{M}(S) \) by at most \( \zeta_1 \), except with probability \( \zeta_2 \), for any value taken by \( S \).

Note that \( O \) is not required to be efficiently implementable.\(^{11}\)

---

\(^9\)Then, formally, one actually needs to prove that \( \text{Guess}_{t',q'}(B(S') \mid S(S')) \approx 0 \) holds even given access to the advice: While this is implicit in the non-uniform setting (every adversary with advice can be turned in an equally good one without advice), the proof is more challenging in the uniform setting.

\(^{10}\)That is, we consider systems consisting of the parallel-composition of a system (either \( A_S[\Sigma] \) or \( S(S) \)), and a correlated \([0, 1]\)-valued random variable.

\(^{11}\)In fact, no proof in the literature provides an explicit construction of \( \mathcal{M} \), with the exception of the work by Reingold *et al.* [RTTV08], whose statement, however, is too weak to be used in our setting.
3.2 The Hardcore Lemma for System-Bit Pairs

Procedure StateSample\(^O(k)\):

```
for all \(i = 1, \ldots, k\) do
    \((\sigma_i, z_i) \leftarrow O\)
    \(c_i \leftarrow [0, 1]\)
    if \(c_i \leq z_i\) then
        return \(\sigma_i\)
return \(\bot\)
```

\(\text{// } r \in \mathbb{N}\)

Figure 3.2: The procedure StateSample, with oracle access to sampler \(O\), as described in Example 3.5

Remark 3.2. Given \(A_S\) and \(M\), it is generally not easy to decide whether a good sampler (i.e., with small \(\ell, \zeta_1\) and \(\zeta_2\)) for \(A_S\) and \(M\) exists at all. However, for all \(\varepsilon > 0\) there obviously exists a measure \(M\) with \(\mu(M) \geq 1 - \varepsilon\) and a \((0, 0)\)-sampler for \(M\) and \(A_S\) with length 0: Namely, we let \(M\) be the all-one measure (then \(P_M = P_S\)) and \(O\) simply returns the initial (void) state \(\bot\) for \(A_S\) and \(z = 1\). We will see further examples of state samplers in the proof of Theorem 3.5 below.

Using State Samplers. State samplers allow for efficient simulation of \(S(S')\) for \(S' \leftarrow M\): Given the output \((\Sigma, Z)\) sampled from a \((\zeta_1, \zeta_2)\)-sampler \(O\), we additionally flip a bit \(B\) with \(P_B(1) = Z\): Consider \(\Sigma'\) with \(P_{\Sigma'} = P_{\Sigma|B=1}\), i.e., distributed as \(\Sigma\) conditioned on the outcome of \(B\) being 1.

If \(\zeta_1 = \zeta_2 = 0\), then \(A_S[\Sigma'] \equiv S(S')\), because, first \((A_S[\Sigma], Z, B) \equiv (S(S), M(S), B')\) by definition, where \(B'\) is a bit which is 1 with probability \(M(S)\), and second, equivalence holds also conditioned on the case \(B = B' = 1\), for which we have \(P_{B'|S}(1, s) := M(s)\) and \(P_{B'|1}(1) := \sum_{s \in S} P_S(s) \cdot M(s) = \mu(M)\), and thus

\[
P_{S|B}(s, 1) = \frac{M(s) \cdot P_S(s)}{\mu(M)} = P_M(s).
\]

Of course, one can similarly simulate \(S'' \leftarrow P_M^{\Sigma'}\) as we obtain a corresponding sampler by just replacing \(z\) by \(1 - z\) in the output \((\sigma, z)\).

Example 3.5. Given a system \(S \equiv S(S)\) with a measure \(M\) and a \((\zeta_1, \zeta_2)\)-sampler \(O\) for \(M\) and an implementation \(A_S\), it is convenient to define a procedure StateSample (detailed in Figure 3.2) which, given oracle access to \(O\), adopts a reject-sampling strategy to sample a state \(\sigma\) for \(A_S\)
Weakly Unpredictable Bits

according to the above distribution $P_S$: That is, $\text{StateSample}^O(k)$ makes at most $k$ calls to $O$, and for each call outputting a pair $(\sigma, z)$, it outputs $\sigma$ with probability $z$ and stops, and otherwise it moves to the next attempt. If no attempt is successful, it outputs the initial state $\bot$.

On input $k$, the probability that all attempts fail and $\bot$ is output is at most $(1 - \mu(M) + \zeta_1 + \zeta_2)^k$.

The following lemma is useful in the case $(\zeta_1, \zeta_2) \neq (0, 0)$.

Lemma 3.4. Let $\mathcal{M}$ be a measure for $S \equiv S(S)$ and let $(S, Z(S), B(S))$ be such that $Z(s)$ is $\zeta_1$-close to $\mathcal{M}(s)$, except with probability $\zeta_2$, for all $s$, and $B(S)$ is $1$ with probability $Z(S)$. Then,

$$d(S', S'') \leq \frac{2\zeta_1}{\mu(M)} + \zeta_2$$

for $S' \overset{\mathcal{E}}{\leftarrow} \mathcal{M}$ and $S'' \overset{\mathcal{E}}{\leftarrow} P_{S,B(s)=1}$.

Proof. Let $\mathcal{E}$ be the event that $Z(S)$ is more than $\zeta_1$ away from $\mathcal{M}(S)$, then

$$d(S', S'') \leq d(P_{S'}, P_{S''|\mathcal{E}}) + P(\mathcal{E}) \leq d(P_{S'}, P_{S''|\mathcal{E}}) + \zeta_2.$$

Moreover, using the fact that $\frac{1}{1+x} \geq 1 - x$ for all $x \in \mathbb{R}_{\geq 0}$,

$$P_{S''|\mathcal{E}}(s) \geq \frac{P_{S}(s) \cdot (\mathcal{M}(s) - \zeta_1)}{P_{B(s)|\mathcal{E}}(1)} \geq \frac{P_{S}(s) \cdot (\mathcal{M}(s) - \zeta_1)}{\mu(M)} \cdot \left(1 - \frac{\zeta_1}{\mu(M)}\right) \geq \frac{P_{S}(s) \cdot (\mathcal{M}(s) - \zeta_1 - \frac{\mathcal{M}(s)\zeta_1}{\mu(M)})}{\mu(M)}.$$

Let $S^+$ be the set of values $s$ for which $P_{S'}(s) \geq P_{S''|\mathcal{E}}(s)$. Then,

$$d(P_{S'}, P_{S''|\mathcal{E}}) = \sum_{s \in S^+} P_{S'}(s) - P_{S''|\mathcal{E}}(s)$$

$$\leq \sum_{s \in S^+} P_{S}(s) \cdot \frac{\mathcal{M}(s)}{\mu(M)} - P_{S}(s) \cdot \frac{(\mathcal{M}(s) - \zeta_1 - \frac{\mathcal{M}(s)\zeta_1}{\mu(M)})}{\mu(M)}$$

$$= \frac{\zeta_1}{\mu(M)} \sum_{s \in S^+} P_{S}(s) \cdot \left(1 + \frac{\mathcal{M}(s)}{\mu(M)}\right) \leq \frac{2\zeta_1}{\mu(M)}.$$

$\square$
3.2 The Hardcore Lemma for System-Bit Pairs

Procedure EstimateWin\(^O\)\((A, \mu(M), \gamma)\): \hspace{1em} // Adv. A, \(\mu(M)\), \(\gamma \in [0, 1]\)

\[
\begin{align*}
 r &:= \left(\frac{1}{\gamma - \frac{\zeta_1}{\mu(M)} - \zeta_2}\right)^2 k \\
 \text{for all } i = 1, \ldots, r \text{ do} \\
 \sigma'_i &\leftarrow \text{StateSample}^O\left(\frac{k}{\log \left(\frac{1}{1 - \mu(M) + \zeta_1 + \zeta_2}\right)}\right) \\
 o_i &:= A(A_S[\sigma'_i]) \oplus A_B[\sigma'_i] \\
 \text{return } \frac{1}{r} \sum_{i=1}^{r} o_i
\end{align*}
\]

\[\text{Figure 3.3: The procedure EstimateWin given in Example 3.6.}\]

Example 3.6. Let \((S, B) \equiv (S(S), B(S))\) be a cc-stateless system-bit pair with efficient implementation \(A(S, B)\), and let A be an adversary: We present a procedure (called EstimateWin and illustrated in Figure 3.3) which, for all measures \(M\) for \((S, B)\), estimates the probability

\[\Pr[A(S(S')) = B(S') | S' \leftarrow M]\]

within target error \(\gamma > \frac{2\zeta_1}{\mu(M)} + \zeta_2\), except with negligible probability, given oracle access to a \((\zeta_1, \zeta_2)\)-sampler \(O\) for \(M\) and \(A(S, B)\) with polynomial length. For efficiency, we require that \(\gamma - \frac{2\zeta_1}{\mu(M)} - \zeta_2\) is noticeable, and that \(\mu(M) - \zeta_1 - \zeta_2\) is noticeable. We use the notation \((A_S[\sigma'_i], A_B[\sigma'_i])\) to denote the two components of the same instance of \((S, B)\) simulated by \(A(S, B)\). (Note that the procedure runs a new instance of \(A\) at every iteration of the for all loop.)

First, by the discussion in Example 3.5, StateSample outputs a state with the wrong distribution with probability at most

\[r \cdot (1 - \mu(M) + \zeta_1 + \zeta_2)^{k/\log \left(\frac{1}{1 - \mu(M) + \zeta_1 + \zeta_2}\right)} = r \cdot 2^{-k},\]

which is negligible for the given \(r\). Assuming that the wrong distribution is never output, then \((A_S[\sigma'_i], A_B[\sigma'_i]) \equiv (S(S''), B(S''))\), with \(S''\) as in Lemma 3.4. Consequently, by Hoeffding’s inequality (Lemma 2.1), \(\sigma := \frac{1}{r} \sum_{i=1}^{r} o_i\) is an estimate of \(\Pr[A(S(S'')) = B(S'')]\) with error \(\gamma - \frac{2\zeta_1}{\mu(M)} - \zeta_2\), except with negligible probability. Moreover, by Lemma 3.4,

\[|\Pr[A(S(S')) = B(S')] - \Pr[A(S(S'')) = B(S'')]| \leq d(S', S'') \leq \frac{2\zeta_1}{\mu(M)} + \zeta_2,\]
Weakly Unpredictable Bits

as one special distinguisher for $S'$ and $S''$ outputs one if and only if $A(S(X)) = B(X)$ for $X = S'$ or $X = S''$. Therefore, the estimate $\hat{\sigma}$ is (by the triangle inequality) at most $\gamma$ away from the actual value.

Note that running $\text{EstimateWin}$ with parameter $1 - \varepsilon \leq \mu(M)$ instead of $\mu(M)$ would also produce an estimate which has at most error $\gamma$, as it can be verified by inspection.

### 3.2.3. The Hardcore Lemma for System-Bit Pairs

In the following, for understood parameters $\gamma$, $\varepsilon$, $\zeta_1$, and $\zeta_2$, we define

$$\varphi_{hc} := \frac{6400}{\gamma^2 (1 - \varepsilon)^4} \cdot \ln \left( \frac{160}{\gamma (1 - \varepsilon)^2} \right)$$

and

$$\psi_{hc} := \frac{200}{\gamma^2 (1 - \varepsilon)^4 \zeta_1} \cdot \ln \left( \frac{2}{\zeta_2} \right).$$

We now state the Hardcore Lemma for cc-stateless system-bit pairs.

**Theorem 3.5** (HCL for System-Bit Pairs). Let $(S, B) \equiv (S(S), B(S))$ be a cc-stateless system-bit pair admitting an implementation $A_{(S, B)}$ with space complexity $s_{A_{(S, B)}}$. Furthermore, for some integers $t, q > 0$ and some $\varepsilon \in [0, 1)$,

$$\text{Guess}_{t,q}(B | S) \leq \varepsilon.$$

Then, for all $0 < \zeta_1, \zeta_2 < 1$ and all $0 < \gamma \leq \frac{1}{2}$, there exists a measure $M$ for $(S, B)$ with $\mu(M) \geq 1 - \varepsilon$ such that the following two properties are satisfied:

(i) For $S' \leftarrow M$, $t' := t / \varphi_{hc}$ and $q' := q / \varphi_{hc}$,

$$\text{Guess}_{t',q'}(B(S') | S(S')) \leq \gamma.$$

(ii) There exists a $(\zeta_1, \zeta_2)$-sampler for $M$ and $A_{(S, B)}$ with length $s_{A_{(S, B)}}(q' \cdot \psi_{hc})$. If $(S(s), B(s))$ is deterministic for all $s$, then there exists a $(0, 0)$-sampler for $M$ and $A_{(S, B)}$ with length $s_{A_{(S, B)}}((7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3}) \cdot q')$.

In the remainder of this section, we outline the proof intuition and give some remarks on the above statement, while the full proof is postponed to the next section (Section 3.2.4). A uniform version of Theorem 3.5 is given below in Section 3.2.8.

**Proof Outline.** The proof, which is by contradiction, extends in a number of ways previous proofs by Impagliazzo [Imp95] and by Holenstein [Hol05]. It starts by assuming that Theorem 3.5 is wrong, that is,
that for any measure $M$ on $S$ with $\mu(M) \geq 1 - \varepsilon$ and such that a $(\zeta_1, \zeta_2)$-sampler for $M$ and $A_{(S,B)}$ with the desired length exists, there also exists an adversary $A$ with time complexity $t'$ and making $q'$ queries such that $\text{Guess}^A(B(S') | S(S')) > \gamma$ for $S' \leftarrow M$. Under this assumption, the proof builds an adversary $A'$ with time complexity $t$ and making $q$ queries such that $\text{Guess}^A(B | S) > \varepsilon$, hence contradicting the assumed hardness of guessing $B$ given access to $S$.

The proof relies on the quantity $N_A(s) := \sum_{A \in \mathcal{A}} \text{Guess}^A(B(s) | S(s))$ measuring the overall performance of a collection $\mathcal{A}$ of deterministic adversaries in guessing $B(s)$ when interacting with $S(s)$ for given a state $s \in S$. Additionally, we let $M_{A,\tau} : S \rightarrow [0,1]$ be the measure associated with $A$ and a threshold $\tau \in \mathbb{N}$ such that elements for which $A$ is worst, i.e., $N_A(s) \leq \tau$, are given high weight (i.e. $M_{A,\tau}(s) = 1$), whereas elements for which $A$ performs well, i.e., $N_A(s) \geq \tau + \frac{1}{\gamma(1-\varepsilon)}$, are not chosen ($M_{A,\tau}(s) = 0$). An intermediate measure value is assigned to states not falling into one of these two categories. In particular, $M_{\emptyset,0}$ is the all-one measure (i.e., $P_M$ equals the initial state distribution $P_S$), which has density $1 \geq 1 - \varepsilon$. A crucial property, which we show, is that $M_{A,\tau}$ always admits a $(\zeta_1, \zeta_2)$-state sampler for all $A$ and $\tau$.

The adversary $A'$ is then built as follows: Starting with $\mathcal{A} := \emptyset$ and $\tau := 0$, we iteratively find a deterministic adversary $A$ such that

$$\text{Guess}^A(B(S') | S(S')) > \gamma$$

for $S' \leftarrow M_{A,\tau}$ and add it to $\mathcal{A}$. (Such $A$ exists by the assumption that Theorem 3.5 is wrong and the fact that a $(\zeta_1, \zeta_2)$-sampler exists for the measure $M_{A,\tau}$.)

The process is repeated (possibly increasing $\tau$ to ensure that $\mu(M_{A,\tau}) \geq 1 - \varepsilon$), until either for all events $E$ defined on $S$ with $P(E) = 1 - \varepsilon$ we have $E \left[ \text{Guess}^A(B(S') | S(S')) \right] > \gamma'$ (for $A \leftarrow A$, $S' \leftarrow P_S[E]$ and $\gamma'$ slightly smaller than $\gamma$) or we have $P \left[ N_A(S) \geq \gamma \cdot |A| \right] \geq 1 - \frac{1 - \varepsilon}{4}$. For each of both cases, we give an adversary $A'$ using the obtained collection $\mathcal{A}$ and guessing $B$ from $S$ with advantage better than $\varepsilon$. Furthermore, a major part of the proof consists of proving that this procedure terminates after at most $7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3}$ iterations.

**Some Additional Remarks.** We briefly discuss some additional points concerning the statement of Theorem 3.5, which highlight the differences with the information-theoretic counterpart (Theorem 3.3).
Remark 3.3. A first natural question is whether the Hardcore Lemma can also be proven for $\mathcal{M}$ defined on the randomness of the efficient implementation $A_{(S,B)}$, hence fully dispensing with the concept of a state sampler and extend the result beyond cc-stateless systems. Yet all existing reductions in proofs of the Hardcore Lemma require repeated sequential executions of adversaries providing independent outputs, and hence inherently require a cc-stateless representation of the system-bit pair.

Remark 3.4. Let $(S, B) \equiv (S(S), B(S))$ be a system-bit pair such that $\text{Guess}_q(B|S) \leq \varepsilon$ and, additionally, $(S(s), B(s))$ is deterministic for all $s$ (or at least stateless). In both cases, even in the information-theoretic setting, we cannot define an event $A$ on $S$ with probability $1 - \varepsilon$ such that a $q$-query adversary can only guess the bit $B$ with probability $\frac{1}{2}$ under the condition that the event $A$ occurs. That is, a security loss is necessary.

We briefly illustrate this in the case $q = 1$. (Counterexamples for $q > 1$ can also be found.) For uniform independent two-valued random variables $R_{00}, R_{01} \in \{0, 1\}$, $R_{10}, R_{11} \in \{1, 2\}$, and $B \in \{0, 1\}$, consider the cc-stateless system-bit pair $(S(R_{00}, R_{01}, R_{10}, R_{11}, B), B)$. On input $x \in \{0, 1\}$, the system $S(r_{00}, r_{01}, r_{10}, r_{11}, b)$ returns $r_{bx}$. Clearly, $\text{Guess}_1(B|S) = \frac{1}{2}$: The optimal 1-query adversary $A^*$ queries 0, and outputs 0 if it obtains a value in $\{0, 1\}$, and 1 if it obtains 2.

Consider an event $A$ such that the bit $B$ can only be guessed with probability $\frac{1}{2}$ when $A$ occurs: We need

$$P_{AR_{00}R_{01}R_{10}R_{11}B}(r_{00}, r_{01}, r_{10}, r_{11}, b) = 0$$

whenever $b = 0$ and $(r_{00}, r_{01}) \neq (1, 1)$, as well as whenever $b = 1$ and $(r_{10}, r_{11}) \neq (1, 1)$, as otherwise there exists an adversary guessing $B$ with non-zero advantage conditioned on $A$ occurring. But then,

$$P[A] \leq \frac{1}{2} \cdot \frac{1}{4} + \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{4} < 1 - \frac{1}{2}.$$

Remark 3.5. It is not known whether the Hardcore Lemma can be extended beyond cc-stateless system-bit pairs. However, we can show that no such statement can hold with respect to arbitrary (but efficiently implementable) system-bit pairs. To see this, first remark that every system with MBO $\hat{S}$ can be turned into a system-bit pair $(S', B)$ such that $\text{Guess}_t(B|S') = \nu_t(\hat{S})$, where $\nu_t(\hat{S})$ denotes the probability of the MBO becoming 1 maximized over all users with time complexity $t$. Let simply $S'$ choose a secret uniform random bit $B$, and then behave as $\hat{S}$. If the MBO turns one, then $S'$ hands the bit $B$ over to the adversary. Clearly, if
3.2 The Hardcore Lemma for System-Bit Pairs

![Diagram](image)

**Figure 3.4:** Measure $M_{A,\tau}(s)$ as a function of $N_A(s)$. In particular, the gray area represents $W_{A,\tau}(s)$ given that $N_{A,\tau}(s) = N$.

the MBO is 0, then $B$ cannot be guessed with probability better than $\frac{1}{2}$, whereas if the MBO turns one, the adversary learns the bit and can guess it with probability 1.

In the setting where all parties (the adversary and the systems) share some initial random parameter (such as a public key), under reasonable assumptions there exist (cf. [BIN97, PW07]) systems\(^\text{12}\) $\hat{S}$ with MBO such that $\nu_t(\hat{S}) \leq \varepsilon$ for all polynomial $t$ and some constant $\varepsilon$, whereas $\nu_t(\hat{S}_1 \| \ldots \| \hat{S}_m) \approx \varepsilon$ holds already for very small $t'$. This yields a system-bit pair $(S, B)$ where guessing $B_1, \ldots, B_m$ for $m$ independent instances $(S_1, B_1), \ldots, (S_m, B_m)$ is not harder than guessing $B$ when interacting with $S$. This implies, for example, that the XOR Lemma, which we prove below and follows from Theorem 3.5, does not hold for such system-bit pairs.

### 3.2.4. High-Level Description of the Proof of Theorem 3.5

In this section, we present a full proof of the Hardcore Lemma for system-bit pairs (Theorem 3.5). At a high level, our proof follows Holenstein’s proof of the tight uniform\(^\text{13}\) Hardcore Lemma given in [Hol05] (which, in turn, was inspired by one of the proofs given by Impagliazzo [Imp95]). However, there are major differences (and difficulties) due to the fact that we are considering discrete interactive systems: In particular, for a given

---

\(^{12}\)Although these original results address two-party protocols, one can see the honest party as a system with MBO.

\(^{13}\)Interestingly, no simpler approach appears to work despite the statement being proven being non-uniform.
s ∈ S the behavior of the system S(s) can be randomized (but stateless). Furthermore, we need to take into account the fact that we prove the existence of a measure for which a sufficiently good state sampler exists. Also, unlike the case of random variables, it is crucial that no quantity in the proof depends on the size of states S, as we cannot generally assume them to be small. Another final difference (which turns out to be rather easy to handle at the technical level) is that, contrary to the traditional Hardcore Lemma, we allow for the distribution PS of the state S to be different than uniform.

INITIAL DEFINITIONS. For a collection A of (deterministic) adversaries, we define
\[
N_A(s) := \sum_{A \in A} \text{Guess}^A(B(s) \mid S(s)),
\]
for all s ∈ S. Note that |N_A(s)| ≤ |A|. In particular, N_A(s) provides a quantitative measure of the overall quality of the collection A in guessing B(s) given access to S(s). For instance, N_A(s)/|A| gives the advantage of a randomly chosen A ∈ A in guessing B(s) while interacting with S(s). Moreover, for a non-negative integer τ ∈ N, we define the measure M_{A,τ} : S → [0, 1] associated with A and τ such that
\[
M_{A,τ}(s) := \begin{cases} 
1 & \text{if } N_A(s) \leq τ, \\
1 - (N_A(s) - τ)γ(1 - ε) & \text{if } τ < N_A(s) < τ + \frac{1}{\gamma(1 - ε)}, \\
0 & \text{if } N_A(s) \geq τ + \frac{1}{\gamma(1 - ε)}. 
\end{cases}
\]
The value M_{A,τ}(s) is plotted as a function of N_A(s) on Figure 3.4. In other words, elements s ∈ S on which many adversaries in A are bad are given more weight by the measure, whereas those for which many adversaries are good are given no measure. Note that M_{∅,0} is the all-one measure, i.e., in particular the associated distribution P_M equals P_S. Hence, μ(M_{∅,0}) ≥ 1 − ε.

STRUCTURE OF THE PROOF. The proof is by contradiction. It starts by assuming that Theorem 3.5 is false. In other words, we assume that for some γ, ζ_1 and ζ_2 the following holds:

Assumption ¬HC. For all measures M : S → [0, 1] with μ(M) ≥ 1 − ε such that a (ζ_1, ζ_2)-sampler for M and A_{(S,B)} with length^{14} \( t := s_A_{(S,B)}(q' \cdot ψ_{hc}) \) exists, there exists an adversary A with time complexity \( t' := t/\hat{ϕ}_{hc} \), making \( q' := q/\hat{ϕ}_{hc} \)
3.2 The Hardcore Lemma for System-Bit Pairs

Procedure GoodEnough($\mathcal{A}$): // collection of adversaries $\mathcal{A}$

1: if $\mathcal{A} = \emptyset$ then return false
2: $p := \min_{\mathcal{E}, |\mathcal{E}| = 1-\varepsilon} P\left[\mathcal{A}(S(S)) = B(S) \mid \mathcal{A} \leftarrow \mathcal{A}, \mathcal{E}\right]$
3: $\rho := P[\mathcal{N}_\mathcal{A}(S) > \gamma \cdot |\mathcal{A}|]$
4: if $p \geq \frac{1}{2} + \frac{1}{4(1-\varepsilon)|\mathcal{A}|} \vee \rho \geq 1 - \frac{1-\varepsilon}{4}$ then
5: return true
6: else
7: return false

Procedure FindCollection: // no input

1: $\mathcal{A} := \emptyset$, $\tau := 0$
2: while not GoodEnough($\mathcal{A}$) do
3: if $\mu(M_{\mathcal{A},\tau}) < 1 - \varepsilon$ then
4: $\tau := \tau + 1$
5: get deterministic $\mathcal{A} \in \mathcal{A}_{\nu, q'}$ such that // see text

$$P\left[\mathcal{A}(S(S')) = B(S') \mid S' \leftarrow M_{\mathcal{A},\tau}\right] \geq \frac{1 + \gamma}{2}$$

6: $\mathcal{A} := \mathcal{A} \cup \{\mathcal{A}\}$
7: return $\mathcal{A}$

Figure 3.5: Description of the procedure FindCollection used in the proof of Theorem 3.5.

queries, and such that

$$\text{Guess}^{\mathcal{A}}(B(S') \mid S(S')) > \gamma.$$  

where $S' \leftarrow M$.

The core of proof consists of the procedure FindCollection, specified in Figure 3.5, which, under Assumption $\neg HC$, outputs a collection of deterministic adversaries $\mathcal{A}$ such that $|\mathcal{A}| \leq 7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3}$ and such that (at least) one of the following two conditions is satisfied:

(A) $P[\mathcal{N}_\mathcal{A}(S) > \gamma \cdot |\mathcal{A}|] \geq 1 - \frac{1}{2}(1 - \varepsilon)$

(B) For all events $\mathcal{E}$ (defined by a probability distribution $P_{\mathcal{E} \mid S}$) such that
\[ P[\mathcal{E}] = 1 - \varepsilon, \text{ we have} \]

\[
P \left[ A(S(S)) = B(S) \mid A \leftarrow A, \mathcal{E} \right] \geq \frac{1}{2} + \frac{1}{4 \cdot (1 - \varepsilon) \cdot |A|}. \]

We show how to derive, from a collection \( \mathcal{A} \) satisfying one of the two conditions, corresponding adversaries \( \mathcal{A}^{(A)} \) and \( \mathcal{A}^{(B)} \) with time complexity \( t \) and making \( q \) queries which guess \( B(S) \) given access to \( S(S) \) with advantage greater than \( \varepsilon \), contradicting the assumed hardness of \( (S, B) \).

Without loss of generality, throughout this proof we assume that \( B(s) \) is determined by \( s \).

**The Procedure FindCollection.** The procedure FindCollection (cf. Figure 3.5) starts with \( \mathcal{A} := \emptyset \) and parameter \( \tau = 0 \), and iteratively adds to \( \mathcal{A} \) a deterministic adversary \( \mathcal{A} \) with time complexity \( t' \) and query complexity \( q' \) such that

\[
\text{Guess}^A(B(S') \mid S(S')) > \gamma, \]

for \( S' \leftarrow \mathcal{M}_{\mathcal{A}, \tau} \). (Possibly, it first increments the value of \( \tau \) by one to ensure that \( \mu(\mathcal{M}_{\mathcal{A}, \tau}) \geq 1 - \varepsilon \).) Finally, it terminates when the collection \( \mathcal{A} \) satisfies (at least) one of conditions (A) or (B).

Note that the procedure FindCollection is to be seen as a process constructing a collection, where in particular adversaries at Line 5 are not found efficiently and explicitly, they are only guaranteed to exist by Assumption \( \neg \text{HC} \).

**Correctness of FindCollection.** In order for an adversary \( \mathcal{A} \) as required in Line 5 to exist under Assumption \( \neg \text{HC} \), two conditions must be met immediately before executing Line 5: First, the measure \( \mathcal{M}_{\mathcal{A}, \tau} \) must have density at least \( 1 - \varepsilon \). Second, there must exist a \((\zeta_1, \zeta_2)\)-sampler for \( \mathcal{M}_{\mathcal{A}, \tau} \) and \( \mathcal{A}(S, B) \) with length \( \ell \leq s_{A(S, B)}(q' \cdot \psi_{hc}) \). The following lemmas show that both conditions are always satisfied.

**Lemma 3.6.** At every execution of Line 5 in FindCollection, the condition \( \mu(\mathcal{M}_{\mathcal{A}, \tau}) \geq 1 - \varepsilon \) holds.

\[ \text{Proof.} \] The proof goes by induction. Clearly, it holds the first time Line 5 is executed, since \( \mu(\mathcal{M}_{\emptyset, \emptyset}) = 1 \geq 1 - \varepsilon \), and hence \( \tau \) is not increased.

\[ \text{\footnotesize\cite{This can be assumed without loss of generality: Given a randomized adversary, an equally good deterministic adversary can be obtained by fixing its randomness to a good value.}} \]
Assume that the claim of the lemma holds up to the $i$-th iteration of the while-loop, and consider the beginning of the $(i+1)$-st iteration: Assume that $\mu(M_{A,\tau}) < 1 - \varepsilon$. This means that there exists $A$ and $M$ such that $\mu(M_{A,\tau}) \geq 1 - \varepsilon$, and $\mu(M_{A \cup \{A\},\tau}) < 1 - \varepsilon$. Note that for all $s \in \mathcal{S}$ we have $N_{A \cup \{A\}}(s) \leq N_A(s) + 1$, which in particular implies that $M_{A \cup \{A\},\tau+1}(s) \geq M_{A,\tau}(s)$ for all $s \in \mathcal{S}$, and hence $\mu(M_{A \cup \{A\},\tau+1}) \geq \mu(M_{A,\tau}) \geq 1 - \varepsilon$.

Lemma 3.7. For all $\zeta_1, \zeta_2$, all collections of adversaries $A$ and all $\tau \in \mathbb{N}$, there exists a $(\zeta_1, \zeta_2)$-sampler for $M_{A,\tau}$ and $A(S, B)$ with length $\ell := s_A(S, B)(\psi \cdot q')$, where

$$\psi := \frac{4}{\zeta_1^2} \cdot |A|^2 \cdot \gamma^2 (1 - \varepsilon)^2 \cdot \ln \left( \frac{2}{\zeta_2} \right).$$

Furthermore, if $(S(S), B(S))$ is deterministic for every value taken by $S$, an error-free state sampler for $M_{A,\tau}$ and $A(S, B)$ with length $\ell := s_A(S, B)(|A| \cdot q')$ exists.

A proof of Lemma 3.7 is postponed to Section 3.2.5.

**Termination of FindCollection.** We prove the following lemma in Section 3.2.6: It upper bounds the number of iterations after which FindCollection returns an appropriate collection.

**Lemma 3.8.** Under the assumption that a good adversary is always found at Line 5, the procedure FindCollection terminates after at most $7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3}$ iterations of the while-loop, outputting a collection $A$ of deterministic adversaries, each with time complexity $t'$ and query complexity $q'$, such that $|A| \leq 7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3}$ and (at least) one of the two conditions (A) and (B) is satisfied.

**The Final Adversaries.** In order to conclude the proof, we need to show that such a collection yields an adversary accessing $S(B)$ and contradicting the assumed hardness of guessing $B$. This is implied by the following two lemmas, whose proofs are deferred to Section 3.2.7.

**Lemma 3.9.** Let $A$ be a set of deterministic adversaries, each with time complexity $t'$ and query complexity $q'$, satisfying Condition (A) and such that $|A| \leq 7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3}$. Then, there exists an adversary $A^{(A)}$ with time and query complexities

$$t_1 := \frac{4}{\gamma^2} \cdot \ln \left( \frac{4}{1 - \varepsilon} \right) \cdot t'$$

$$q_1 := \frac{4}{\gamma^2} \cdot \ln \left( \frac{4}{1 - \varepsilon} \right) \cdot q'$$
Weakly Unpredictable Bits

such that \( \text{Guess}^{A(A)}(B \mid \mathbf{S}) > \varepsilon \).

Lemma 3.10. Let \( \mathcal{A} \) be a set of deterministic adversaries, each with time complexity \( t' \) and query complexity \( q' \), satisfying Condition (B) such that \( |A| \leq 7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3} \). Then, there exists an adversary \( A^{(B)} \) with time and query complexities

\[
\begin{align*}
t_2 &:= \frac{6400}{\gamma^2(1 - \varepsilon)} \cdot \ln \left( \frac{160}{\gamma(1 - \varepsilon)^3} \right) \cdot t' \\
qu_2 &:= \frac{6400}{\gamma^2(1 - \varepsilon)^4} \cdot \ln \left( \frac{160}{\gamma(1 - \varepsilon)^3} \right) \cdot q'
\end{align*}
\]

such that \( \text{Guess}^{A^{(B)}}(B \mid \mathbf{S}) > \varepsilon \).

CONCLUDING THE PROOF. To conclude the proof, we observe that by Assumption \(^{-\text{HC}},\) Lemmas 3.6, 3.7, and 3.8, \( \text{FindCollection} \) outputs a collection \( A \) with \( |A| \leq 7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3} \) satisfying one of conditions (A) and (B). Then either Lemma 3.9 or Lemma 3.10 yields an adversary contradicting the assumed hardness of \( (\mathbf{S}, B) \), as by definition of \( \varphi_{\text{hc}} \), for the time complexity we have \( t_1 \leq t \) and \( t_2 \leq t \), as well as \( q_1 \leq q \) and \( q_2 \leq q \).

3.2.5. Existence of State Samplers (Proof of Lemma 3.7)

The key observation is that given the collection \( \mathcal{A} \) as well as black-box access to the system-bit pair \( (\mathbf{S}(S), B(S)) \), it is possible to compute a sufficiently good estimate of \( N_A(S) \) (and thus of \( M_{A,\tau}(S) \)) by letting (randomly chosen) adversaries from \( \mathcal{A} \) sequentially interact with \( \mathbf{S}(S) \) and check whether their outputs equal \( B \) or not. In particular, we can use the algorithm \( A_{(S,B)} \) in order to simulate \( (\mathbf{S}(S), B(S)) \). Here, for notational convenience, we assume that the algorithm \( A_{(S,B)} \) first outputs the simulated bit \( B \), and subsequently implements \( S \) accordingly.

This is summarized by the random process \( O_{A,\tau} \), which outputs a valid state of \( A_{(S,B)} \), and is described in Figure 3.6, on top. We additionally consider the idealized process \( O'_{A,\tau} \) (depicted in the lower part of Figure 3.6) which, instead of using the algorithm \( A_{(S,B)} \), uses the actual system-bit pair \( (\mathbf{S}(S), B(S)) \) for a random \( S \) \( \xleftarrow{\$} \mathcal{P}_S \), and produces an estimate \( z(S) \) of \( M_{A,\tau}(S) \) as in \( O_{A,\tau} \), and finally outputs the pair \( (S, z(S)) \).

Obviously, \( (A_{(S,B)}[\Sigma], Z) \) for \( (\Sigma, Z) \) \( \xleftarrow{\$} O_{A,\tau} \) and \( (\mathbf{S}(S), B(S), z(S)) \) for \( (S, z(S)) \) \( \xleftarrow{\$} O'_{A,\tau} \) are exactly the same, that is, the equivalence

\[
(A_{(S,B)}[\Sigma], Z) \equiv ((\mathbf{S}(S), B(S)), z(S))
\]
3.2 The Hardcore Lemma for System-Bit Pairs

**Sampler \( O_{A, \tau} \):**  // collection of adversaries \( A, \tau \in \mathbb{N} \)

\[
\begin{align*}
    & b \leftarrow A_{(S, B)} \quad // \text{run } A_{(S, B)} \text{ to get the bit } B \\
    & \sigma := \text{state of } A_{(S, B)} \text{ after outputting } B \\
    & \psi := \frac{4}{\zeta^2} \cdot |A|^2 \cdot \gamma^2 (1 - \varepsilon)^2 \cdot \ln \left( \frac{2}{\zeta^2} \right) \\
    & \text{for all } i := 1, \ldots, \psi \text{ do} \\
    & \quad A \leftarrow^s A \\
    & \quad v_i := A_{(A_{(S, B)}[\sigma])} \oplus b \oplus 1 \quad // \text{1 if correct, 0 else} \\
    & \quad \sigma := \text{last state of } A_{(S, B)} \text{ after interacting with } A \\
    & \quad \overline{N} := 2|A| \cdot \left( \frac{1}{\psi} \sum_{i=1}^{\psi} v_i - \frac{1}{2} \right) \\
    & \quad z := \max\{0, \min\{1, 1 - (\overline{N} - \tau) \gamma (1 - \varepsilon)\}\} \\
    & \text{return } (\sigma, z)
\end{align*}
\]

**Sampler \( O'_{A, \tau} \):**  // collection of adversaries \( A, \tau \in \mathbb{N} \)

\[
\begin{align*}
    & s \leftarrow^s P_S \quad // \text{collection of adversaries } A, \tau \in \mathbb{N} \\
    & \psi := \frac{4}{\zeta^2} \cdot |A|^2 \cdot \gamma^2 (1 - \varepsilon)^2 \cdot \ln \left( \frac{2}{\zeta^2} \right) \\
    & \text{for all } i := 1, \ldots, \psi \text{ do} \\
    & \quad A \leftarrow^s A \\
    & \quad v_i(s) := A(S(s)) \oplus B(s) \oplus 1 \quad // \text{1 if correct, 0 else} \\
    & \quad \overline{N}(s) := 2|A| \cdot \left( \frac{1}{\psi} \sum_{i=1}^{\psi} v_i(s) - \frac{1}{2} \right) \\
    & \quad z(s) := \max\{0, \min\{1, 1 - (\overline{N}(s) - \tau) \gamma (1 - \varepsilon)\}\} \\
    & \text{return } (s, z(s))
\end{align*}
\]

**Figure 3.6:** Top: Construction of the state sampler \( O_{A, \tau} \) for \( M_{A, \tau} \) and \( A_{(S, B)} \). Bottom: Idealized state sampler \( O'_{A, \tau} \) used in the proof of Lemma 3.7.
holds, due to the fact that $A_{(S,B)}$ implements $(S,B)$. Therefore, in order to finish the proof of property (ii) in Definition 3.3, it suffices to analyze the quality of the estimate $z(S)$ output by $O'_{A,:}$. Let us fix $s \in S$ and consider the random variable $V(s)$ obtained by sampling a random $A \leftarrow A$ and then letting $V(s) := 1$ if $A(S(s)) = B(s)$, and $V(s) := 0$ otherwise. Then, note that

$$E[V(s)] = \frac{1}{|A|} \sum_{A \in A} P[A(S(s)) = B(s)]$$

$$= \frac{1}{|A|} \sum_{A \in A} \left( \frac{1}{2} + \frac{\text{Guess}^A(B(s) \mid S(s))}{2} \right) = \frac{1}{2} + \frac{N_A(s)}{2|A|}.$$ 

Since $(S(S),B(S))$ is cc-stateless, the pair $(S(s),B(s))$ is stateless (and in particular, $S(s)$ is stateless), and the values assigned to $v_1(s), \ldots, v_\psi(s)$ are independent with the same distribution as $V(s)$. In particular, we have $E[v_i(s)] = E[V(s)]$ for all $i \in \{1, \ldots, \psi\}$. The event that $z(s)$ is a bad estimate, i.e., that $|z(s) - M_{A,:} (s)| > \zeta_1$, implies in particular

$$\left| N(s) - N_A(s) \right| > \frac{\zeta_1}{\gamma (1-\varepsilon)} \quad \text{and} \quad \left| \psi^{-1} \sum_{i=1}^\psi v_i(s) - E[V(s)] \right| > \frac{\zeta_1}{2|A|\gamma (1-\varepsilon)}.$$ 

We conclude by applying Hoeffding’s inequality (Lemma 2.1),

$$P \left[ |z(s) - M_{A,:} (s)| > \zeta_1 \right] \leq P \left[ \left| N(s) - N_A(s) \right| > \frac{\zeta_1}{\gamma (1-\varepsilon)} \right]$$

$$\leq P \left[ \left| \psi^{-1} \sum_{i=1}^\psi v_i(s) - E[V(s)] \right| > \frac{\zeta_1}{2|A|\gamma (1-\varepsilon)} \right]$$

$$\leq 2e^{-\frac{\zeta_1^2}{4|A|\gamma^2(1-\varepsilon)^2 \psi}} = \zeta_2,$$

by the choice of $\psi$.

In the case where $(S(S),B(S))$ is deterministic given $S$, then a much simpler (and error-free) state sampler can be built: Namely, it is sufficient to let all adversaries $A \in A$ interact (sequentially) with $S$ (simulated by $A_{(S,B)}$), rather than choosing them randomly, and set $v_i$ as above. Performing the analysis as above (via the ideal process $O'_{A,:}$), we see that $N(s)$ indeed equals $N_A(s)$, and thus the estimate is always correct.
3.2.6. The Procedure FindCollection Terminates (Proof of Lemma 3.8)

Similarly to the proofs in [Imp95, Hol05], we define for all collections of adversaries $A$, all $\tau \in \mathbb{N}$ and all $s \in S$ the quantity

$$W_{A,\tau}(s) := \begin{cases} \tau - N_A(s) + \frac{1}{2\gamma(1-\varepsilon)} & \text{if } N_A(s) \leq \tau, \\ \frac{1}{2} \left[ \tau + \frac{1}{\gamma(1-\varepsilon)} - N_A(s) \right] & \text{if } \tau < N_A(s) < \tau + \frac{1}{\gamma(1-\varepsilon)}, \\ 0 & \text{else.} \end{cases}$$

This quantity is to be interpreted as the area under $M_{A,\tau}(s)$, starting from $N_A(s)$, as depicted in Figure 3.4. We also define the weighted average of $W_{A,\tau}(s)$ as

$$W(A, \tau) := \mathbb{E}[W_{A,\tau}(S)] = \sum_{s \in S} P_S(s) \cdot W_{A,\tau}(s).$$

In the following, we study the evolution of this quantity in order to prove termination for the execution of FindCollection. In particular, there are two causes for a change in the value of $W$ at each iteration: First, an adversary $A$ is always added to $A$. Second, the value $\tau$ is possibly incremented by one before the adversary is added. The following two claims analyze these two cases. We point out that the analysis is more involved than the one of [Hol05], due to the different termination condition and the underlying distribution being arbitrary and not necessarily uniform.

The first claim shows that if we add an adversary to $A$ in Line 5, then $W(A \cup \{A\}, \tau)$ is smaller than $W(A, \tau)$ by at least $\frac{\gamma(1-\varepsilon)}{2}$.

**Claim 1.** Let $A$ be a collection of adversaries and let $\tau \in \mathbb{N}$ be such that $\mu(M_{A,\tau}) \geq 1 - \varepsilon$. Moreover, let $A \notin A$ be an adversary such that

$$\text{Guess}^A(B(S') \mid S(S')) > \gamma$$

for $S' \sim M_{A,\tau}$. Then $W(A \cup \{A\}, \tau) \leq W(A, \tau) - \frac{\gamma(1-\varepsilon)}{2}$.

**Proof.** For all $s \in S$, the definition of $N_A$ yields $N_{A \cup \{A\}}(s) = N_A(s) + \text{Guess}^A(B(s) \mid S(s))$, and consequently we obtain

$$W_{A \cup \{A\},\tau}(s) \leq W_{A,\tau}(s) - \text{Guess}^A(B(s) \mid S(s)) \cdot M_{A,\tau}(s) + \frac{\gamma(1-\varepsilon)}{2}.$$
Averaging over the choice of $S$ yields

$$W(A \cup \{A\}, \tau) = \sum_{s \in S} P_S(s) \cdot W_{A \cup \{A\}, \tau}(s)$$

$$\leq W(A, \tau) + \frac{\gamma(1 - \varepsilon)}{2} - \sum_{s \in S} P_S(s) \cdot \text{Guess}^{A}(B(s) \mid S(s)) \cdot M_{A, \tau}(s).$$

However, note that if $S' \overset{\$}{\leftarrow} M_{A, \tau}$, then by definition $P_{S'}(s) := P_S(s) \cdot M_{A, \tau}(s)/\mu(M_{A, \tau})$, and hence

$$\sum_{s \in S} P_S(s) \cdot \text{Guess}^{A}(B(s) \mid S(s)) \cdot M_{A, \tau}(s) =$$

$$= \mu(M_{A, \tau}) \cdot \text{Guess}^{A}(B(S') \mid S(S')),$$

and the claim follows from $\mu(M_{A, \tau}) \geq 1 - \varepsilon$ and $\text{Guess}^{A}(B(S') \mid S(S')) > \gamma$.

The following claim additionally shows bounds on the variation of $W(A, \tau)$ upon incrementing $\tau$.

**Claim 2.** Let $\eta > 0$, let $A$ be a collection of adversaries, and let $\tau \in \mathbb{N}$. Further assume that $P[N_A(S) > \eta] < 1 - \frac{1}{4}(1 - \varepsilon)$ and $\mu(M_{A, \tau}) < 1 - \varepsilon$. Then,

$$W(A, \tau + 1) \leq \begin{cases} 
W(A, \tau) + (1 - \varepsilon) + \frac{\gamma(1 - \varepsilon)}{2} - \frac{\gamma(1 - \varepsilon)^2}{8} & \text{if } \tau > \eta, \\
W(A, \tau) + (1 - \varepsilon) + \frac{\gamma(1 - \varepsilon)}{2} & \text{if } \tau \leq \eta.
\end{cases}$$

**Proof.** For all $s \in S$ we have

$$W_{A, \tau + 1}(s) \leq W_{A, \tau}(s) + M_{A, \tau}(s) + \frac{\gamma(1 - \varepsilon)}{2}.$$

In fact, if $N_A(s) \leq \tau$, then we even have $W_{A, \tau + 1}(s) \leq W_{A, \tau}(s) + M_{A, \tau}(s)$. From this we can infer

$$W(A, \tau + 1) = \sum_{s \in S} P_S(s) \cdot W_{A, \tau + 1}(s)$$

$$\leq W(A, \tau) + \sum_{s \in S} P_S(s) \cdot M_{A, \tau}(s) + \sum_{s : N_A(s) > \tau} P_S(s) \cdot \frac{\gamma(1 - \varepsilon)}{2}$$

$$\leq W(A, \tau) + (1 - \varepsilon) + P[N_A(S) > \tau] \cdot \frac{\gamma(1 - \varepsilon)}{2}.$$
3.2 The Hardcore Lemma for System-Bit Pairs

If \( \tau > \eta \), then \( P[N_A(S) > \tau] \leq P[N_A(S) > \eta] \leq 1 - \frac{1 - \rho}{4} \), and thus

\[
W(A, \tau + 1) \leq W(A, \tau) + (1 - \varepsilon) + \left(1 - \frac{1 - \varepsilon}{4}\right) \cdot \frac{\gamma(1 - \varepsilon)}{2}
\]

\[
= W(A, \tau) + (1 - \varepsilon) + \frac{\gamma(1 - \varepsilon)}{2} - \frac{\gamma(1 - \varepsilon)^2}{8}
\]

whereas if \( \tau \leq \eta \), we can only conclude that \( W(A, \tau + 1) \leq W(A, \tau) + (1 - \varepsilon) + \frac{\gamma(1 - \varepsilon)}{2} \).

\[\Box\]

In the following, let \( A(i) \) and \( \tau(i) \) be the values of \( A \) and \( \tau \) at the beginning of the \( i \)-th iteration, i.e., when \( \text{GoodEnough} \) is invoked. In particular, \( |A(i)| = i - 1 \). We now show that under the assumption\(^{16}\) that \( P[N_A(i)(S) > \gamma \cdot (i - 1)] < 1 - \frac{1}{2}(1 - \varepsilon) \) holds, then \( \text{FindCollection} \) terminates satisfying Condition (B). To this end, we define the potential function \( \pi \) such that the potential at the beginning of the \( i \)-th iteration is

\[ \pi(i) := W(A(i), \tau(i)) - \tau(i) \cdot (1 - \varepsilon). \]

Note that initially \( \pi(1) = W(\emptyset, 0) - 0 \cdot (1 - \varepsilon) = W(\emptyset, 0) = \frac{1}{2\gamma (1 - \varepsilon)} \). The above two claims imply the following for all \( i \)’s:

\[ \pi(i+1) \leq \begin{cases} 
\pi(i) - \frac{2(1 - \varepsilon)}{2} & \text{if } \tau(i) = \tau(i + 1) \\
\pi(i) & \text{if } \tau(i + 1) = \tau(i) + 1 \text{ and } \tau(i) \leq \gamma \cdot (i - 1) \\
\pi(i) - \frac{\gamma(1 - \varepsilon)^2}{8} & \text{if } \tau(i + 1) = \tau(i) + 1 \text{ and } \tau(i) > \gamma \cdot (i - 1).
\end{cases} \]

In particular, it is important to note that the value \( \pi(i) \) never increases. The following claim proves that it also decreases sufficiently fast, reaching a negative value after \( 7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3} \) iterations. Below, we show that in this case, the corresponding collection \( A \) satisfies Condition (B).

**Claim 3.** For \( \lambda := 7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3} \) we have \( \pi(\lambda + 1) < 0 \).

**Proof.** Assume, towards a contradiction, that the claim is wrong, i.e., we have an execution of \( \text{FindCollection} \) such that after \( \lambda := 7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3} \) completed iterations we have \( \pi(\lambda + 1) \geq 0 \) at the beginning of the \( (\lambda + 1) \)-st iteration. Consider the partition \( I^- \cup I^+ = \{1, \ldots, \lambda\} \), where \( I^- \) are the iterations \( i \) where \( \tau \) was not increased, whereas in \( I^+ \) the value of \( \tau \)

\[^{16}\]If this assumption is not satisfied, then \( \text{GoodEnough} \) will detect this, and return \( \text{true} \), which in particular implies termination of \( \text{FindCollection} \), returning the collection \( A(i) \) satisfying Condition (A).
is increased. We also let $\mathcal{I}^* := \{(1 - \gamma)^{-1}|\mathcal{I}| + 2, \ldots, \lambda\}$. Hence, for all $i \in \mathcal{I}^*$ we have

$$\tau(i) \geq i - 1 - |\mathcal{I}| > i - 1 - (1 - \gamma)(i - 1) = \gamma(i - 1),$$

since $(1 - \gamma)^{-1}|\mathcal{I}| < i - 1$ for all $i \in \mathcal{I}^*$. Therefore, for each iteration $i \in \mathcal{I}^*$, we have $\pi(i + 1) \leq \pi(i) - \frac{\gamma(1 - \varepsilon)^2}{8}$, and note that

$$|\mathcal{I}^*| = \lambda - \frac{|\mathcal{I}|}{1 - \lambda} - 1.$$ 

Moreover, we have $|\mathcal{I}| \leq \gamma^{-2}(1 - \varepsilon)^{-2} < \gamma^{-2}(1 - \varepsilon)^{-3}$, as otherwise this would contradict $\pi(\lambda + 1) \geq 0$ by the first claim. Using $\gamma \leq \frac{1}{2}$, we obtain $|\mathcal{I}| > 5\gamma^{-2}(1 - \varepsilon)^{-3} - 1$, which in particular implies

$$\pi(\lambda + 1) \leq \frac{1}{2\gamma(1 - \varepsilon)} - |\mathcal{I}^*| \frac{\gamma(1 - \varepsilon)^2}{8} < \frac{\gamma(1 - \varepsilon)^2}{8} - \frac{1}{8\gamma(1 - \varepsilon)} < 0,$$

which is a contradiction. \qed

Finally, we show that if $\pi(i) < 0$, then the collection $\mathcal{A}(i)$ satisfies Condition (B), and thus termination is achieved.

Claim 4. If $\pi(i) < 0$, then $\mathcal{A}(i)$ satisfies

$$\mathbb{P}\left[\mathcal{A}(S(S)) = B(S) \mid \mathcal{A} \triangleleft \mathcal{A}(i), \mathcal{E}\right] \geq \frac{1}{2} + \frac{1}{4 \cdot (1 - \varepsilon) \cdot |\mathcal{A}(i)| \cdot \gamma}$$

for all events $\mathcal{E}$ (defined by $\mathbb{P}_{\mathcal{E}|S}$) such that $\mathbb{P}[\mathcal{E}] = 1 - \varepsilon$.

Proof. Let $\tau := \tau(i)$ and $\mathcal{A} := \mathcal{A}(i)$. Also let $\mathcal{E}$ be an arbitrary event (defined by $\mathbb{P}_{\mathcal{E}|S}$) such that $p := \mathbb{P}[\mathcal{E}] = 1 - \varepsilon$. Then, by the definition of $N_A$,

$$\mathbb{P}\left[\mathcal{A}(S(S)) = B(S) \mid \mathcal{A} \triangleleft \mathcal{A}, S \in \mathcal{E}\right] = \frac{1}{2} + \frac{1}{2(1 - \varepsilon)|\mathcal{A}|} \sum_{s \in S} \mathbb{P}_{S}(s) \cdot N_A(s).$$

Note that for all $s \in S$ it is easy to verify that

$$W_{\mathcal{A}, \tau}(s) \geq \tau + \frac{1}{2\gamma(1 - \varepsilon)} - N_A(s),$$
3.2 The Hardcore Lemma for System-Bit Pairs

**Figure 3.7:** Adversary \( A^{(A)} \).

<table>
<thead>
<tr>
<th>Adversary ( A^{(A)} ): // expecting to interact with system ( S(s) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \varphi_A := \frac{4}{\gamma} \cdot \ln \left( \frac{4}{1-\varepsilon} \right) )</td>
</tr>
<tr>
<td>for all ( i := 1, \ldots, \varphi_A ) do</td>
</tr>
<tr>
<td>( A \leftarrow A )</td>
</tr>
<tr>
<td>( v_i \leftarrow A(S(s)) )</td>
</tr>
<tr>
<td>return majority ( { v_i : i \in {1, \ldots, \varphi_A } } )</td>
</tr>
</tbody>
</table>

and therefore

\[
\sum_{s \in S} P_{SE}(s) \cdot N_A(s) \geq \sum_{s \in S} P_{SE}(s) \cdot \left[ \tau + \frac{1}{2\gamma(1-\varepsilon)} - W_{A,\tau}(s) \right] \\
\geq P[E] \cdot \left( \tau + \frac{1}{2\gamma(1-\varepsilon)} \right) - \sum_{s \in S} P_{S}(s) \cdot W_{A,\tau}(s) \\
\geq (1-\varepsilon) \cdot \tau + \frac{1}{2\gamma} - W(A,\tau) \geq \frac{1}{2\gamma}.
\]

The lemma follows by substituting this into the above equation.

Therefore, after at most \( 7 \cdot \gamma^{-2} \cdot (1-\varepsilon)^{-3} \) executions of the \textbf{while}-loop, the procedure \texttt{FindCollection} outputs a set of adversaries which satisfies at least one of Condition (A) and Condition (B).

### 3.2.7. Constructing the Final Adversaries

**Proof of Lemma 3.9.** Recall that we assume that

\[
P[N_A(S) > \gamma \cdot |A|] \geq 1 - \frac{1}{4}(1-\varepsilon) > 1 - \frac{1}{2}(1-\varepsilon) = \frac{1+\varepsilon}{2}.
\]

The adversary \( A^{(A)} \) proceeds as described in Figure 3.7.

Let us fix \( s \) such that \( N_A(s) > \gamma \cdot |A| \). Then, we denote as \( I(s) \in \{0, 1\} \) the indicator random variable which is 1 if a randomly sampled \( A \leftarrow A \) interacting with \( S(s) \) is successful, outputting \( B(s) \). (Recall that \( B(s) \) is fully determined by \( s \).) Note that

\[
E[I(s)] = P[A(S(s)) = B(s) \mid A \leftarrow A] = \frac{1}{|A|} \sum_{A \in A} P[A(S(s)) = B(s)] \\
= \frac{1}{2} + \frac{1}{2 \cdot |A|} \sum_{A \in A} \text{Guess}^A(B(s) \mid S(s)) = \frac{1}{2} + \frac{N_A(s)}{2 \cdot |A|} > \frac{1+\gamma}{2}.
\]
For $i = 1, \ldots, \varphi_A$, denote as $I_i(s)$ the random variable indicating whether $v_i$ equals $B(s)$ when interacting with $S(s)$: Since $S(s)$ is stateless, the variables $I_1(s), \ldots, I_{\varphi_A}(s)$ are independent, with $E[I_i(s)] = E[I(s)]$ for all $i = 1, \ldots, \varphi_A$. We can use this (combined with Hoeffding’s inequality) to derive an upper bound on the failure probability of $A(\mathcal{A})$ given access to $S(s)$,

$$P\left[ A(\mathcal{A})(S(s)) \neq B(s) \right] = P\left[ \frac{1}{\varphi_A} \sum_{i=1}^{\varphi_A} I_i(s) < \frac{1}{2} \right]$$

$$\leq P\left[ \frac{1}{\varphi_A} \sum_{i=1}^{\varphi_A} I_i(s) < E[I(s)] - \frac{\gamma}{2} \right]$$

$$< e^{-\frac{\gamma^2}{4}\varphi_A} = \frac{1-\varepsilon}{4},$$

and therefore

$$P[A(\mathcal{A})(S(S)) = B(S)] > P[N_A(S) > \gamma \cdot |\mathcal{A}|] \cdot (1 - \frac{1-\varepsilon}{4}) \geq (1 - \frac{1-\varepsilon}{4}) \cdot (1 - \frac{1-\varepsilon}{4}) \geq 1 - \frac{1-\varepsilon}{2} = \frac{1+\varepsilon}{2}.$$ 

Note that we can fix the choice of $\varphi_A$ adversaries $A_1, \ldots, A_{\varphi_A}$ for the random choices of $\mathcal{A}$ maximizing the success probability of the resulting adversary $A(\mathcal{A})$. The time and query complexity bounds $t_1$ and $q_1$ are then clear.

**Proof of Lemma 3.10.** We define the functions $\alpha, \alpha_1 : \mathcal{S} \to [-1, 1]$ such that

$$\alpha(s) := 2 \cdot P\left[ A(S(s)) = B(s) \mid A \leftarrow \mathcal{A} \right] - 1,$$

$$\alpha_1(s) := 2 \cdot P\left[ A(S(s)) = 1 \mid A \leftarrow \mathcal{A} \right] - 1$$

for all $s \in \mathcal{S}$. We order the elements of $\mathcal{S}$ as $s_1, s_2, \ldots$ such that $\alpha(s_i) \leq \alpha(s_{i+1})$ for $i = 1, 2, \ldots$. Then, let $i^*$ be the (unique) index such that $\sum_{j=1}^{i^*-1} P_S(s_j) < 1 - \varepsilon$, but $\sum_{j=1}^{i^*} P_S(s_j) \geq 1 - \varepsilon$. We define the event $\mathcal{E}$ such that

$$P_{\mathcal{E}|S}(s_i) := \begin{cases} 
1 & \text{if } i < i^*, \\
(1-\varepsilon) - \sum_{j=1}^{i^*-1} P_S(s_j) & \text{if } i = i^*, \\
0 & \text{if } i > i^*, 
\end{cases}$$

for $s_i \in \mathcal{S}$. We can use this (combined with Hoeffding’s inequality) to derive an upper bound on the failure probability of $A(\mathcal{A})$ given access to $S(s)$.
Adversary $A^{(B)}$: \[
\varphi_B := \frac{6400 \gamma}{(1-\varepsilon)^2} \cdot \ln \left( \frac{160 \gamma}{(1-\varepsilon)^3} \right) \\
\text{for all } i := 1, \ldots, \varphi_B \text{ do} \\
A \leftarrow A \\
v_i \leftarrow A(S(s)) \\
\alpha_1 := 2 \cdot \sum_{i=1}^{\varphi_B} v_i - 1 \\
\text{return 1 with probability max } \left\{ 0, \min \left\{ 1, \frac{1}{2} + \frac{\alpha_1^* - \alpha_1(s)^*}{2(\alpha_1^* - \gamma(1-\varepsilon)^2)} \right\} \right\}
\]

Figure 3.8: Adversary $A^{(B)}$.

and define $\alpha^* := \alpha(s^*)$. It is easy to verify that $P[\mathcal{E}] = 1 - \varepsilon$, and hence (recall that $|\mathcal{A}| \leq 7 \cdot \gamma^{-2} \cdot (1-\varepsilon)^{-3}$)

\[
P \left[ A(S(S)) = B(S) \mid A \leftarrow A, \mathcal{E} \right] \geq \frac{1}{2} + \frac{1}{4(1-\varepsilon)|\mathcal{A}| \gamma} = \frac{1}{2} + \frac{\gamma(1-\varepsilon)^2}{40}.
\]

This implies $\alpha^* > \frac{\gamma(1-\varepsilon)^2}{20}$, as otherwise this would contradict the above lower bound on the guessing probability of a randomly chosen $A$ from the collection $\mathcal{A}$. We consider the adversary $A^{(B)}$ specified in Figure 3.8.

In the following, we assume that the above adversary is run on $S(s)$ for some fixed $s \in S$. Denote as $v_1(s), \ldots, v_{\varphi_B}(s)$ and $\alpha_1(s)$ the variables obtained during the interaction. Since $S(s)$ is stateless, the values assigned to $v_1(s), \ldots, v_{\varphi_B}(s)$ are independent, and distributed as the output $V(s)$ of a randomly chosen $A \leftarrow A$ accessing $S(s)$, which satisfies

\[
E[V(s)] = P \left[ A(S(s)) = 1 \mid A \leftarrow A \right] = \frac{1}{2} + \frac{\alpha_1(s)}{2}.
\]

Let $G = G(s)$ be the event that $|\alpha_1(s) - \alpha_1(s)| \leq \frac{\gamma(1-\varepsilon)^2}{40}$: By Hoeffding’s
inequality,
\[
P[\mathcal{G}] = P\left[\left|\sum_{i=1}^{\varphi_B} v_i(s) - E[V(s)]\right| > \frac{\gamma(1-\varepsilon)^2}{80}\right]
\leq 2e^{-\left(\frac{\gamma(1-\varepsilon)^2}{80}\right)^2 \varphi_B} = \frac{\gamma(1-\varepsilon)^3}{80}.
\]

For all \(s \in \mathcal{S}\), define \(\alpha(s)\) to be \(\alpha_1(s)\) if \(B(s) = 1\), and \(-\alpha_1(s)\) if \(B(s) = 0\), and note that by inspection
\[
P[A(B)(S(s)) = B(s) | \mathcal{G}] = \min \left\{1, \frac{1}{2} + \frac{\alpha(s) - \gamma(1-\varepsilon)^2/40}{2}\right\}.
\]

For any \(s\) such that \(P_{E|S}(s) > 0\), then \(\alpha(s) \geq \alpha^*\), and conditioned on the event \(\mathcal{G}\), we have \(\pi(s) \geq \alpha^* - \frac{\gamma(1-\varepsilon)^2}{40}\), and thus
\[
P[A(B)(S(s)) = B(s) | \mathcal{G}] = 1.
\]

Moreover, for every other \(s \in \mathcal{S}\), we have
\[
P[A(B)(S(s)) = B(s) | \mathcal{G}] \geq \frac{1}{2} + \frac{\alpha(s) - \gamma(1-\varepsilon)^2/40}{2}.
\]

Therefore,
\[
P[A(B)(S(S)) = B(S) | \mathcal{E}, \mathcal{G}] \geq \frac{1}{2} + \frac{\gamma(1-\varepsilon)^2}{40} - \frac{\gamma(1-\varepsilon)^2}{80} = \frac{1}{2} + \frac{\gamma(1-\varepsilon)^2}{80}.
\]

We finally conclude that
\[
P[A(B)(S(S)) = B(S)] \geq P[\mathcal{G}] \cdot P[A(B)(S(S)) = B(S) | \mathcal{G}]
\geq P[A(B)(S(S)) = B(S) | \mathcal{G}] - P[\mathcal{G}]
= P[\mathcal{E}] \cdot P[A(B)(S(S)) = B(S) | \mathcal{G}, \mathcal{E}]
+ P[\mathcal{E}^c] \cdot P[A(B)(S(S)) = B(S) | \mathcal{G}, \mathcal{E}^c] - P[\mathcal{G}]
> \varepsilon \cdot 1 + \frac{1}{2}(1-\varepsilon) + \frac{\gamma(1-\varepsilon)^3}{80} - P[\mathcal{G}] = \frac{1 + \varepsilon}{2},
\]
as we wanted to show. The time complexity $t_2$ and the query complexity $q_2$ are clear by inspection.

### 3.2.8. The Uniform Hardcore Lemma

In this section, we discuss a uniform version of Theorem 3.5 along similar lines as the one from [Hol05]. In informal terms, the uniform statement differs in that we prove that, for every polynomial-time adversary $A$ and for every noticeable function $\gamma$, there exists a good measure $\mathcal{M} = \mathcal{M}(A, \gamma)$ (with a corresponding sampler) such that guessing $B(S')$ given $S(S')$ is hard for $A$, provided $S' \leftarrow \mathcal{M}$, even when given oracle access to a sampler for $\mathcal{M}$. This will be sufficient for the applications of the lemma. (In the following, we re-define $\psi_{hc} := \frac{7200}{17(1 - \varepsilon)^4} \cdot \ln \left( \frac{17}{\varepsilon^2} \right)$ in the uniform setting.)

**Theorem 3.11 (Uniform Hardcore Lemma).** Let $(S, B) \equiv (S(S), B(S))$ be a cc-stateless system-bit pair with an efficient implementation $A(S, B)$ with space complexity $s_A(S, B)$. Furthermore, for some $\varepsilon \in [0, 1)$ (with $1 - \varepsilon$ noticeable)

$$\text{Guess}_{\text{poly}}(B | S) \leq \varepsilon.$$

For all noticeable $\zeta_1 > 0$, all $\zeta_2 = 2^{-\text{poly}(k)} > 0$, and all $0 < \gamma \leq \frac{1}{2}$ (such that $\frac{2\zeta_1}{\gamma^2} + \zeta_2 \leq \frac{1}{4}$ and $1 - \varepsilon - \zeta_1 - \zeta_2$ is noticeable), and for all polynomial-time $q'$-query (uniform) oracle adversaries $A^{(1)}$, there exists a measure $\mathcal{M}$ for $(S, B)$ with $\mu(\mathcal{M}) \geq 1 - \varepsilon$ such that the following two properties are satisfied:

(i) There exists a $(\zeta_1, \zeta_2)$-sampler $O$ for $\mathcal{M}$ and $A(S, B)$ with length $\ell := s_{A(S, B)}(q' \cdot \psi_{hc})$. In particular, if $(S(s), B(s))$ is deterministic for all $s$, then $O$ is a $(0, 0)$-sampler for $\mathcal{M}$ and $A(S, B)$ with length $s_{A(S, B)}((256 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3}) \cdot q')$.

(ii) For $S' \leftarrow \mathcal{M},$ $^{17}$

$$\text{Guess}^{A_0}(B(S') | S(S')) \leq \gamma.$$

The length of the states returned by the sampler only depends on the number of queries $q'$, but is independent of the time complexity of $A$ (and the length of oracle queries $A$ expects). Otherwise, it would be possible that the statement only holds for adversaries which cannot retrieve the output of the sampler.

$^{17}$That is, $A$ is given access to an oracle which upon each invocation ignores its input and returns a state sampled according to the state sampler.
Rather than giving a full proof, we opt for pointing out the modifications to be made to the proof of Theorem 3.5 to obtain a uniform reduction. The techniques are similar to the ones of [Hol05], yet care is required due to the interactive setting and the inability to exactly compute $P[A(S(s)) = B(s)]$. Once again, we proceed by contradiction, and the corresponding initial assumption, which we tacitly make from now on, is modified as follows:

**Assumption** $\neg HC'$. There exist noticeable $\zeta_1$, $\gamma$, and $\zeta_2 = 2^{-\text{poly}(k)}$ (with the property that $\frac{2\zeta_2}{1-\zeta_1} + \zeta_2 \leq \frac{\gamma}{2}$ and $1-\zeta_1 - \zeta_2$ is noticeable), as well as a polynomially bounded $q'$, and an efficient $q'$-query oracle adversary $A^*$ which satisfies

$$\text{Guess}^{A^*}(B(S') | S(S')) > \gamma$$

for all measures $\mathcal{M}$ for $(S, B)$ with $\mu(\mathcal{M}) \geq 1 - \varepsilon$ admitting a $(\zeta_1, \zeta_2)$ state-sampler $O$ for $A_{(S, B)}$ with length $\ell$, and $S' \overset{\varepsilon}{\sim} \mathcal{M}$.

In particular, we consider the general case where $(S(s), B(s))$ is possibly randomized. (The deterministic statement will follow in the same way as in the non-uniform case.)

The following lemma transforms $A^*$ into a more convenient form.

**Lemma 3.12.** There exists an efficient oracle adversary $B^*$ which, for all measures $\mathcal{M}$ for $(S, B)$ with $\mu(\mathcal{M}) \geq 1 - \varepsilon$ admitting a $(\zeta_1, \zeta_2)$ state-sampler $O$ for $A_{(S, B)}$ with length $\ell$, outputs with overwhelming probability, given oracle access to $O$, an efficient deterministic $q'$-query adversary $A$ (making no oracle queries) such that

$$\text{Guess}^{A^*}(B(S') | S(S')) > \gamma/6$$

for $S' \overset{\varepsilon}{\sim} \mathcal{M}$.

**Proof.** The adversary $B^*$ repeats the following, until successful: First, it chooses the random coins $r$ for $A^*$ and makes sufficiently many oracle queries in advance, returning values $(\sigma_1, z_1), (\sigma_2, z_2), \ldots$. Then, it estimates, using $\text{EstimateWin}$ with target error $\gamma/6$ (cf. Example 3.6), the advantage of $A^*$ in guessing $B(S')$ given access to $S(S')$ with fixed randomness $r$ and oracle answers $(\sigma_1, z_1), (\sigma_2, z_2), \ldots$. If the estimate is larger than $\gamma/3$, it outputs $A^*$ with hard-coded random bits and query answers.

By the property of $\text{EstimateWin}$, if an adversary is output, its advantage is larger than $\gamma/6$ with overwhelming probability. Also note that
in each attempt, random coins and oracle answers such that the resulting adversaries achieves advantage at least $\gamma/2$ are sampled with probability at least $\gamma/2$ by a Markov-like argument. In this case, an adversary is output by $B^*$, except with negligible probability. Moreover, after $\Theta\left(\frac{k}{\log(1/(1-\gamma/2))}\right)$ attempts, such good random coins and oracle answers are sampled except with probability $O(2^{-k})$.

To ensure a consistent notation with the non-uniform case, in the remainder of this proof the parameter $\gamma$ will equal $\gamma/6$ for the “actual” $\gamma$ that appears in Assumption $\neg HC'$. The security reduction implements a polynomial-time adversary $A'$ with $\text{Guess}^A(B(S) \mid S(S)) > \varepsilon$ by initially running the procedure $\text{FindCollection}$, using the oracle adversary $(B^*)^0$ at Line 5 to obtain an adversary $A$ to be added to the collection, and the state-sampler $O$ for $M_{A,T}$ accessed by $B^*$ is simulated (also in polynomial-time) using the procedure given in the proof of Lemma 3.7. The final collection $A$ output when $\text{GoodEnough}$ returns true is used according to $A(A)$ or $A(B)$ depending on whether (a variation of) Condition (A) or Condition (B) is satisfied.

However, the parameters $\rho$, $p$ (both in $\text{GoodEnough}$), and $\alpha^*$ (in $A^{(B)}$) cannot be computed exactly: The procedure $\text{GoodEnough}$ is consequently modified, as in Figure 3.9, so that these parameters are estimated and the conditions in the if-statement are adapted accordingly. The next paragraphs describe the procedures $\text{Estimate-}\rho$ and $\text{Estimate-}\rho$, as well as the resulting adaptations to $A^{(A)}$ and $A^{(B)}$. In particular, they will ensure that $\text{GoodEnough}$ returns true whenever one of Conditions (A) or (B) is satisfied, which implies that the analysis of termination for $\text{FindCollection}$ remains unchanged.

**Estimating $\rho$ and the Adversary $A^{(A)}$.** The procedure $\text{Estimate-}\rho$ is described in Figure 3.10. We assume without loss of generality that $A_{(S,B)}$ first outputs the bit $B$, and then simulates $S$. In particular, in each iteration of the outer for-loop an independent instance $(S(S_i), B(S_i))$ is simulated using $A_{(S,B)}$, for independent $S_1, \ldots, S_r$. We will therefore assume that equivalently the adversary $A_{i,j}$ interact with $S(S_i)$ for all $i$ and $j$, and the associated value $N_i$ is denoted as $N_{A}(S_i)$. Note that the expected value of the $b_j$'s (for a given $S_i = s_i$) is

$$p(s_i) := P\left[A(S(s_i)) = B(s_i) \mid A \xleftarrow{\$} A\right],$$
Procedure `GoodEnough(A)`:

1. if \( A = \emptyset \) then return false
2. \( \overline{\rho} := \text{Estimate}\_\rho(A) \)
3. \( \overline{p} := \text{Estimate}\_p(A) \)
4. if \( \overline{\rho} \geq 1 - \frac{1 - \varepsilon}{4} \lor \overline{p} \geq \frac{1}{2} + \frac{5}{32(1 - \varepsilon) |A|} \) then
   5. return true
   6. else
      7. return false

Figure 3.9: Procedure `GoodEnough` in the proof of the uniform Hardcore Lemma.

Procedure `Estimate\_\rho(A)`:

\( r := \left( \frac{12}{1 - \varepsilon} \right)^2 \cdot k \)
\( r' := \left( \frac{64}{7} \right)^2 \cdot k \)

for all \( i = 1, \ldots, r \) do
   \( b \leftarrow A(s, B)[\bot] \)
   \( \sigma_0 := \text{state of } A(s, B) \text{ after outputting } b \)
for all \( j = 1, \ldots, r' \) do
   \( A_{i,j} \leftarrow A \)
   \( b_j := A_{i,j}(A(s, B)[\sigma_{j-1}]) \oplus b \oplus 1 \)
   \( \sigma_j := \text{last state of } A(s, B)[\sigma_{j-1}] \).
\( \overline{N}_i := 2 \cdot |A| \cdot \left( \frac{1}{r'} \sum_{j=1}^{r'} b_j - \frac{1}{2} \right) \).
\( \overline{\rho} := \left\{ i : \overline{N}_i > \frac{3}{4} \cdot |A| \right\} \)
return \( \overline{\rho} \)

Figure 3.10: Procedure `Estimate\_\rho` in the proof of the uniform Hardcore Lemma.

and therefore, by the choice of \( r' \), the average \( \frac{1}{r} \sum_{j=1}^{r'} b_j \) is at most \( \gamma/8 \) away from \( p(s_i) \), except with probability \( 2e^{-k} \), by Hoeffding's inequality, which means that \( \overline{N}_A(s_i) \) is at most \( 2 |A| \gamma/8 = |A| \cdot (\gamma/4) \) away from
3.2 The Hardcore Lemma for System-Bit Pairs

\(N_{\mathcal{A}}(s_i)\). Moreover, we define

\[\overline{p}_i \colonequals \frac{1}{p} |\{i : N_{\mathcal{A}}(s_i) > \gamma \cdot |\mathcal{A}|\}|, \quad \overline{\rho}_i \colonequals \frac{1}{p} |\{i : N_{\mathcal{A}}(s_i) > (\gamma/2) \cdot |\mathcal{A}|\}|\]

and observe that \(\overline{p}_i \leq \overline{\rho} \leq \overline{\rho}_i\). Also, except with probability \(2e^{-k}\), we have

\[P \left[ N_{\mathcal{A}}(S) \geq \frac{\gamma}{2} \cdot |\mathcal{A}| \right] \geq \overline{\rho} - \frac{\gamma}{4} \geq \overline{\rho} - \frac{1 - \varepsilon}{12}.\]

In other words, if \(\overline{\rho} \geq 1 - \frac{1 - \varepsilon}{3}\), then we know that \(P \left[ N_{\mathcal{A}}(S) > \frac{\gamma}{2} \cdot |\mathcal{A}| \right] \geq 1 - \frac{1 - \varepsilon}{3} - \frac{1 - \varepsilon}{12} = 1 - \frac{5(1 - \varepsilon)}{12}\). Additionally, we also know that, except with probability \(2e^{-k}\), we have

\[\overline{p} \geq \overline{p}_i \geq P \left[ N_{\mathcal{A}}(S) > \gamma \cdot |\mathcal{A}| \right] - \frac{1 - \varepsilon}{12},\]

and thus, if Condition (A) holds, then \(\overline{p} \geq 1 - \frac{1 - \varepsilon}{4} - \frac{1 - \varepsilon}{12} = 1 - \frac{1 - \varepsilon}{3}\), and thus \(\text{GoodEnough} \) returns \(\text{true}\).

Finally, we note that \(A^{(A)}\) can simply be modified to work under the condition that \(P \left[ N_{\mathcal{A}}(S) > \frac{\gamma}{2} \cdot |\mathcal{A}| \right] \geq 1 - \frac{5(1 - \varepsilon)}{12}\) by setting \(\varphi_{\mathcal{A}}\) appropriately. Also, in the same way, we can ensure that its advantage is noticeably higher than \(\varepsilon\) in order to compensate for the negligible error in the reduction.

**Estimating \(p\) and the Adversary \(A^{(B)}\).** The estimation of \(p\) is considerably more involved, and is described in Figure 3.11. As above, we associate with each iteration of the outer for-loop an instance \((S(s_i), B(s_i))\), and denote the corresponding \(\overline{p}_i\) as \(\overline{p}(S_i)\).

Note that a function \(p : S \to \mathbb{R}\) always defines an ordering\(^{18}\) of the elements of \(S\) as \(s_1, s_2, \ldots\), such that \(p(s_i) \leq p(s_{i+1})\) for all \(i = 1, 2, \ldots\), and it is convenient to think of \(S\) as being sampled by first choosing a random variable \(R\) uniformly at random in \([0, 1)\), and then letting \(S = S(R) := s_i\) for the unique \(i\) such that \(R \in [\sum_{j=1}^{i-1} p(S(s_j)), \sum_{j=1}^{i} p(S(s_j))]\). Then, we let \(\mathcal{E}^*\) be the event that \(R \in [0, 1 - \varepsilon)\): Clearly, \(P[\mathcal{E}^*] = 1 - \varepsilon\), and it is not hard to verify that \(\mathcal{E}^*\) minimizes \(E[p(S) | \mathcal{E}]\) over all \(\mathcal{E}\) with \(P[\mathcal{E}] = 1 - \varepsilon\) defined on \(S\).\(^{19}\) In the following, we consider the function \(p\) as in the previous paragraph: Note that \(E[p(S) | \mathcal{E}] := P[\mathcal{A}(S(s)) = B(s) | A \leftarrow \mathcal{A}, \mathcal{E}]\).

\(^{18}\)In fact, there may be different possibilities, in which case we choose one arbitrarily.

\(^{19}\)If a better event \(\mathcal{E}'\) assigns probability mass to \([1 - \varepsilon, 1)\), and transferring the corresponding probability mass to the range \([0, 1 - \varepsilon)\) leads to an even better event.
Procedure \text{Estimate-}p(A):

// collection of adversaries \mathcal{A}

\begin{align*}
\gamma & := 1024 \cdot \gamma^2 \cdot |A|^2 \cdot k \\
\gamma' & := 1024 \cdot (1 - \varepsilon)^2 \cdot \gamma^2 \cdot |A|^2 \cdot k \\
\text{for all } i = 1, \ldots, r & \text{ do} \\
& b \leftarrow A(S, B)[\perp] \\
& \sigma_0 := \text{state of } A(S, B) \text{ after outputting } b \\
\text{for all } j = 1, \ldots, r' & \text{ do} \\
& A_{i,j} \leftarrow A \left( A_{i,j}(S_{\sigma_j - 1}) \right) \oplus b \oplus 1 \\
& \sigma_j := \text{last state of } A(S, B)[\sigma_j - 1] \\
& \overline{p}_i := \frac{1}{r'} \sum_{j=1}^{r'} b_j, \\
& \text{reorder } \overline{p}_1, \ldots, \overline{p}_r \text{ such that } \overline{p}_i \leq \overline{p}_{i+1} \text{ for all } i = 1, \ldots, r - 1 \\
& \overline{p}^* := (1 - \varepsilon)^r \\
& \overline{p} := \frac{1}{r - 1} \sum_{i=1}^{r'} \overline{p}_i \\
\text{return } \overline{p}
\end{align*}

Figure 3.11: Procedure \text{Estimate-}p \text{ in the proof of the uniform Hardcore Lemma.}

By the choice of \( r' \), we have \(|\overline{p}(S_i) - p(S_i)| \leq \frac{1}{32(1 - \varepsilon)|A| \gamma} \) for all \( i = 1, \ldots, r \) with overwhelming probability. In this case, by inspection, the average \( \overline{p} \) output by \text{Estimate-}p \text{ using the values } \overline{p}(S_i) \text{ is at most } \frac{1}{32(1 - \varepsilon)|A| \gamma} \text{ away from the one obtained using } p(S_i) \text{ instead of } \overline{p}(S_i) \text{ for all } i = 1, \ldots, r.

Therefore, from now on, we analyze the case where the \( p_i := p(S_i) \)'s are used. In particular, let \( R_i \in \{0, 1\} \) be associated to \( S_i \) and let \( \mathcal{E}^* \) be defined as above for the function \( p \). Using the Hoeffding’s bound twice, we can show that \(|R_i - (1 - \varepsilon)| \leq \frac{1}{32|A| \gamma}\) with overwhelming probability. First, let \( X_i \in \{0, 1\} \) (for \( i \in \{1, \ldots, r\} \)) be an the indicator random variable which is 1 if and only if \( R_i < 1 - \varepsilon - \frac{1}{32|A| \gamma} \). Then,

\[
\mathbb{P} \left[ \frac{1}{r} \sum_{i=1}^{r} X_i \geq 1 - \varepsilon \right] = \mathbb{P} \left[ \frac{1}{r} \sum_{i=1}^{r} X_i \geq \mu - \frac{1}{32|A| \gamma} \right] \leq e^{-k},
\]

which yields \( R_i \geq 1 - \varepsilon - \frac{1}{32|A| \gamma} \) with probability \( 1 - e^{-k} \), since otherwise, more than a fraction \( 1 - \varepsilon \) of the \( R_i \)'s would be smaller than \( 1 - \varepsilon - \frac{1}{32|A| \gamma} \).
Symmetrically, one can show that except with probability $e^{-k}$ we have $R_i^* \leq 1 - \varepsilon + \frac{1}{32|\mathcal{A}|\gamma}$.

We now observe that conditioned on $R_i^* \in [0, 1)$, the average $\bar{p}$ of $p(R_i)$ (for $i = 1, \ldots, i^* - 1$) is the same as the average of $p(R'_1), \ldots, p(R'_{i^* - 1})$ for $R'_1, \ldots, R'_{i^* - 1}$ uniformly distributed in $[0, R_i^*]$. Therefore, by the Hoeffding bound, we have that the average $p_1, \ldots, p_{i^* - 1}$ is at most $\frac{1}{32|\mathcal{A}|\gamma}$ away from $E[p(S) | S \in [0, R_i^*]]$, except with probability $2e^{-k}$. In the specific case where $|R_i^* - (1 - \varepsilon)| \leq \frac{1}{32|\mathcal{A}|\gamma}$, we have, using Lemma A.1,

$$|\mathbb{E}[p(S) | S \in [0, r)] - \mathbb{E}[p(S) | E^*]| \leq \frac{1}{32(1 - \varepsilon)|\mathcal{A}|\gamma},$$

since $\max R_i^*, (1 - \varepsilon) \geq 1 - \varepsilon$. Therefore, in the end we have

$$|\bar{p} - p| \leq 3 \cdot \frac{1}{32(1 - \varepsilon)|\mathcal{A}|\gamma}.$$

Thus, if the test $\bar{p} \geq \frac{1}{2} + \frac{5}{32(1 - \varepsilon)|\mathcal{A}|\gamma}$ passes, then we are guaranteed that $p \geq \frac{1}{2} + \frac{1}{16(1 - \varepsilon)|\mathcal{A}|\gamma}$ holds. Furthermore, if $p \geq \frac{1}{2} + \frac{1}{32(1 - \varepsilon)|\mathcal{A}|\gamma}$, as guaranteed by Condition (B), then the test is also successful.

Finally, we need to adapt $\mathcal{A}(B)$ to only use the guarantee that $p \geq \frac{1}{2} + \frac{1}{16(1 - \varepsilon)|\mathcal{A}|\gamma}$ (which is unproblematic), and furthermore, we need to additionally estimate $\alpha^*$: This can be done as the computation of $\bar{p}_i^*$ in Estimate-$p$, we the difference that we compute

$$\bar{p}_i := 2 \cdot \left( \frac{1}{r'} \sum_{j=1}^{r'} b_j - 1 \right),$$

and take $\alpha_i^*$ as an approximation of $\alpha^*$. Also, again we have to ensure that the advantage is noticeably higher than $\varepsilon$ to compensate for the negligible error probability. We omit the details.

### 3.3. The XOR Lemma for System-Bit Pairs

This section presents the second main result of this chapter, the XOR Lemma for cc-stateless system-bit pairs. In analogy to the first part of this chapter, we start by considering the information-theoretic case as a motivation for the computational results. Then, Section 3.3.2 presents a proof of the XOR Lemma from the Hardcore Lemma which follows the information-theoretic intuition. Finally, Section 3.3.3 is devoted to a direct proof of the XOR Lemma achieving a somewhat stronger statement.
3.3.1. Setting and Information-Theoretic Intuition

We consider the setting with \( m \) system-bit pairs \( (S_1, B_1), \ldots, (S_m, B_m) \). We are interested in upper bounding (as a function of the Guess\(_q\)(\( B_i | S_i \)) for \( i = 1, \ldots, m \)) the advantage Guess\(_q\_1,\ldots,\_q_m\)(\( B_1 \oplus \cdots \oplus B_m | S_1 \cdots S_m \)) of guessing the bit \( B_1 \oplus \cdots \oplus B_m \) given parallel access to \( S_1, \ldots, S_m \), where at most \( q_i \) queries to each system \( S_i \) are allowed. That is, we consider the most general attack where the adversary can query each subsystem \( S_i \) individually at most \( q_i \) times, adaptively depending on the answers of queries to other subsystems.

The following theorem shows that the advantage is upper bounded by the product of the individual advantages Guess\(_q\)(\( B_i | S_i \)).

**Theorem 3.13.** For all \( q_1, \ldots, q_m > 0 \),

\[
\text{Guess}_{q_1, \ldots, q_m}(B_1 \oplus \cdots \oplus B_m | S_1 \cdots S_m) \leq \prod_{i=1}^{m} \text{Guess}_{q_i}(B_i | S_i).
\]

**Proof.** For all \( i = 1, \ldots, m \), define for \( (S_i, B_i) \) the corresponding system-bit pair with MBO \( (\hat{S}_i, \hat{B}_i) \) which is guaranteed to exist by Theorem 3.3. Clearly,

\[
\text{Guess}_{q_1, \ldots, q_m}(B_1 \oplus \cdots \oplus B_m | S_1 \cdots S_m) = \text{Guess}_{q_1, \ldots, q_m}(\hat{B}_1 \oplus \cdots \oplus \hat{B}_m | \hat{S}_1 \cdots \hat{S}_m).
\]

However, as long as there exists an \( i \) such that the MBO of \( (\hat{S}_i, \hat{B}_i) \) is 0, Property (ii) of Theorem 3.3 implies that the bit \( \hat{B}_i \) is uniformly distributed, and therefore the exclusive-or \( \hat{B}_1 \oplus \cdots \oplus \hat{B}_m \) cannot be guessed, except with probability \( \frac{1}{2} \). Therefore, for all adversaries \( A \), with \( \nu := \nu_{q_1, \ldots, q_m}(\hat{S}_1 \cdots \hat{S}_m) \),

\[
P[A(\hat{S}_1 \cdots \hat{S}_m) = \hat{B}_1 \oplus \cdots \oplus \hat{B}_m] \leq (1 - \nu) \cdot \frac{1}{2} + \nu \cdot 1 \leq \frac{1 + \nu}{2},
\]

and by Theorem 2.4,

\[
\nu_{q_1, \ldots, q_m}(\hat{S}_1 \cdots \hat{S}_m) \leq \prod_{i=1}^{m} \nu_{q_i}(\hat{S}_i) = \prod_{i=1}^{m} \text{Guess}_{q_i}(B_i | S_i). \quad \Box
\]
3.3.2. XOR Lemma from the Hardcore Lemma

In this section we prove a generalization of Yao’s XOR Lemma [Yao82, GNW95] to system-bit pairs, which is a computational analogue of Theorem 3.13: We are given $m$ instances $\langle S_1, B_1 \rangle, \ldots, \langle S_m, B_m \rangle$ of a cc-stateless system-bit pair $(S, B) \equiv (S(S), B(S))$ with the property that $\text{Guess}_{t',q'}(B | S) \leq \varepsilon$ and admitting an implementation $A_{(S,B)}$ with time and space complexities $t_{A_{(S,B)}}$ and $s_{A_{(S,B)}}$, respectively. We prove that, given access to all of $S_1, \ldots, S_m$ in parallel, the XOR $B_1 \oplus \cdots \oplus B_m$ can only be guessed with advantage at most $\varepsilon m + \gamma$ by an adversary with related time complexity $t$ and making at most $q$ queries to each subsystem $S_i$, where $\gamma$ can be made arbitrarily small at the cost of reducing $t$ and $q$ with respect to $t'$ and $q'$.

**Theorem 3.14 (XOR Lemma).** For all integers $t, q, \varepsilon > 0$, all $\varepsilon \in (0, 1)$, $\zeta_1, \zeta_2 \in (0, 1)$, and $\gamma \in (0, \frac{1}{2}]$, if

$$\text{Guess}_{t',q'}(B | S) \leq \varepsilon,$$

then,

$$\text{Guess}_{t,q,\ldots,q}(B_1 \oplus \cdots \oplus B_m | S_1 \parallel \ldots \parallel S_m) \leq \varepsilon m + 2m \cdot (\zeta_1 + \zeta_2) + \gamma,$$

where $t' = \varphi_{hc} \cdot \left[ t + O ((m - 1) \cdot (l + t_{A_{(S,B)}}(q,l))) \right]$ and $q' = \varphi_{hc} \cdot q$, with $l := s_{A_{(S,B)}}(\psi_{hc} \cdot q)$.

**Proof.** Let $A$ be an adversary with time complexity $t$ issuing $q$ queries to each subsystem $S_1, \ldots, S_m$ and outputting a guess $B'$ for $B_1 \oplus \cdots \oplus B_m$. Further, using the Hardcore Lemma (Theorem 3.5), let $M$ with $\mu(M) \geq 1 - \varepsilon$ be such that

$$\text{Guess}_{t'/\varphi_{hc},q}(B(S') | S(S')) \leq \gamma$$

(3.1)

for $S' \leftarrow M$, and let $O$ be the corresponding $(\zeta_1, \zeta_2)$-sampler for $M$ and $A_{(S,B)}$ with length $l = s_{A_{(S,B)}}(\psi_{hc} \cdot q)$.

---

20 An analogous statement can easily be obtained if the instances are from different system-bit pairs. However, we will prove such a general statement in Theorem 3.15 below, and for this version, opt for simplicity in order to illustrate an example application of the Hardcore Lemma.

21 As pointed out in [GNW95], referring to a note by Steven Rudich, such a trade-off is unavoidable. A quantitative study of this trade-off follows from the work of Shaltiel and Viola [SV08].
We consider an adversary $A'$ which, given access to $S(S')$, first samples $m$ pairs $(\Sigma_1, Z_1), \ldots, (\Sigma_m, Z_m)$ using $O$, and subsequently flips independent bits $U_1, \ldots, U_m \in \{0, 1\}$, where $P_{U_j}(1) := Z_j$ for $j = 1, \ldots, m$. Then, it picks an $i \in \{1, \ldots, m\}$ (provided it exists) such that $U_i = 1$, and simulates the interaction of $A$ with

$$A_S[\Sigma_1] \parallel \cdots \parallel A_S[\Sigma_i-1] \parallel S(S') \parallel A_S[\Sigma_i+1] \parallel \cdots \parallel A_S[\Sigma_m],$$

where $A_S[\Sigma_j]$ is the instance of $S$ simulated by $A_{(S,B)}[\Sigma_j]$ (with $B'_j$ being the associated bit), for $j = 1, \ldots, m$, $j \neq i$. When $A$ outputs a bit $B'_i$, $A'$ outputs $B' \oplus \bigoplus_{j \neq i} B'_j$. If no such $i$ exists, $A'$ terminates directly by returning a random bit. Clearly, $A'$ can equivalently be seen as sampling $m$ independent pairs $(S_1, Z(S_1)), \ldots, (S_m, Z(S_m))$, letting $U_j$ be one with probability $Z(S_j)$, and replacing $A_S[\Sigma_j]$ and $B'_j$ with $S(S_j)$ and $B(S_j)$ for all $j = 1, \ldots, m$, $j \neq i$. This does not alter the guessing advantage.

We also consider the adversary $\tilde{A}'$ where $Z(S_j)$ is replaced by $M(S_j)$ for all $j = 1, \ldots, m$, and denote as $\tilde{U}_1, \ldots, \tilde{U}_m$ the corresponding independent bits. (Remark that $P_{\tilde{U}_j}(1) = \mu(M) \geq 1 - \epsilon$.) Analogously, in the experiment where $A$ interacts with $S_1 \parallel \cdots \parallel S_m$, define the bits $U_j$ accordingly. In both experiments, let $E$ be the event that $\tilde{U}_1 = \cdots = \tilde{U}_m = 0$. Note that $P[E] \leq \epsilon^m$. Furthermore,

$$P \left[ \tilde{A}'(S(S')) = B(S') \mid E \right] = P \left[ A(S_1) \parallel \cdots \parallel S_m = B_1 \oplus \cdots \oplus B_m \mid E \right],$$

from which, using the facts that $P \left[ A(S_1) \parallel \cdots \parallel S_m = B_1 \oplus \cdots \oplus B_m \mid \overline{E} \right] \leq 1$ and that $P \left[ \tilde{A}'(S(S')) = B(S') \mid \overline{E} \right] = \frac{1}{2}$, we can upper bound $p := P \left[ A(S_1) \parallel \cdots \parallel S_m = B_1 \oplus \cdots \oplus B_m \right]$ as

$$p \leq P[E] + (1 - P[E]) \cdot P \left[ \tilde{A}'(S(S')) = B(S') \mid \overline{E} \right] \leq \frac{P[E]}{2} + P[E] \cdot \frac{1}{2} + (1 - P[E]) \cdot P \left[ \tilde{A}'(S(S')) = B(S') \mid \overline{E} \right] \leq \frac{\epsilon^m}{2} + P \left[ \tilde{A}'(S(S')) = B(S') \right],$$

or, in other words

$$\text{Guess}^A(B_1 \oplus \cdots \oplus B_m \mid S_1 \parallel \cdots \parallel S_m) \leq \epsilon^m + \text{Guess}^{\tilde{A}'}(B(S') \mid S(S')).$$

(3.2)

Furthermore, note that for all $j = 1, \ldots, m$ and $b \in \{0, 1\}$

$$d(\tilde{U}_j, U_j) = \left| P_{\tilde{U}_j}(b) - P_{U_j}(b) \right| \leq \zeta_1 + P[E] \leq \zeta_1 + \zeta_2,$$
3.3 The XOR Lemma for System-Bit Pairs

where $\mathcal{E}'$ is the event that an error larger than $\zeta_1$ occurs in the estimate. Then, it is not hard to verify that

$$\text{Guess}^{\tilde{A}}(B(S') | S(S')) \leq \text{Guess}^{A'}(B(S') | S(S'))$$

$$+ 2 \cdot d((U_1, \ldots, U_m), (\tilde{U}_1, \ldots, \tilde{U}_m))$$

$$\leq \text{Guess}^{A'}(B(S') | S(S')) + 2m(\zeta_1 + \zeta_2).$$

To conclude the proof, we notice that an adversary at least as good as $A'$ can be implemented with complexity $t + O((m - 1) \cdot (l + t_{A_{S,B_i}}(q, l))) \leq t'/\varphi_{he}$ (for an optimal choice of the samples from $O$ and of the associated bits $U_1, \ldots, U_m$), and by Equation 3.1 this adversary achieves advantage at most $\gamma$. □

3.3.3. A Direct Proof via the Isolation Technique

In the following, we present an improved version of Theorem 3.14, whose proof is slightly more involved, and which does not make use of the Hardcore Lemma, but relies on the so-called isolation technique, which was previously used in a number of hardness amplification results, including Levin’s proof of the XOR Lemma [Lev87, GNW95], Myers’ indistinguishability amplification result for PRFs [Mye03], and hardness amplification of weakly-verifiable puzzles [CHS05].

Here, in contrast to Theorem 3.14, we consider the setting with $m$ (possibly different) system-bit pairs $(S_1, B_1), \ldots, (S_m, B_m)$, where all but one of them are cc-stateless. We define the quantity

$$\varphi_{\oplus} := 2 \left( \frac{24m}{\gamma} \right)^2 \cdot \ln \left( \frac{7m}{\gamma} \right)$$

for understood $m$ and $\gamma$. Also, $t_{A_{S_i}}$ and $s_{A_{S_i}}$ are the time and space complexities of some implementation $A_{S_i}$ of the system $S_i$, whereas $t_{A_{(S_i, B_i)}}$ is the time complexity of an implementation $A_{(S_i, B_i)}$ of the pair $(S_i, B_i)$. (Note that an efficient implementation of the latter implies one for the former, but we allow for this distinction.) For all $i$, we denote $l_i := s_{A_{S_i}}(q_i \cdot \varphi_{\oplus})$ and $l_{<i} := \sum_{j=1}^{i-1} l_j$ (for understood $q_1, \ldots, q_{m-1}$).

**Theorem 3.15 (XOR Lemma).** Let $(S_1, B_1), \ldots, (S_{m-1}, B_{m-1})$ be all cc-stateless system-bit pairs, and let $(S_m, B_m)$ be an arbitrary system-bit pair. For all $t, q_1, \ldots, q_m, \gamma > 0$,

$$\text{Guess}_{t, q_1, \ldots, q_m}(B_1 \oplus \cdots \oplus B_m | S_1 \| \ldots \| S_m) \leq \prod_{i=1}^{m} \text{Guess}_{t_i, q_i}(B_i | S_i) + \gamma,$$
where
\[
    t'_i = l_{<i} + \varphi_{\oplus} \cdot \left[ t + \mathcal{O} \left( \sum_{j=1}^{i-1} t_{A_{\delta_j}}(q_j, l_j) + \sum_{j=i+1}^{m} t_{A_{(A_j, u_j)}}(q_j) \right) \right]
\]
and \( q'_i = \varphi_{\oplus} \cdot q_i \) for \( i = 1, \ldots, m-1 \), whereas
\[
    t_m = l_{<m} + t + \mathcal{O} \left( \sum_{j=1}^{m-1} t_{A_{\delta_j}}(q_j, l_j) \right), \quad q'_m = q_m.
\]

The asymmetry of our proof technique allows \((S_m, B_m)\) to be fully stateful. Furthermore, both \( t'_m \) and \( q'_m \) are much smaller than the corresponding terms \( t'_i \) and \( q'_i \) for \( i = 1, \ldots, m-1 \). The full proof is given in the next section.

### 3.3.4. Proof of Theorem 3.15

The proof of Theorem 3.15 relies on the so-called Isolation Lemma, which reduces the situation where an adversary obtains an advantage \( \delta \cdot \varepsilon \) in guessing the XOR of the bits \( B_1 \oplus \cdots \oplus B_m \) for \( m-i \) system-bit pairs to either an adversary guessing \( B_i \) from \( S_i \) with advantage at least \( \varepsilon \), or to the situation where an adversary guesses \( B_{i+1} \oplus \cdots \oplus B_m \) for the system-bit pairs \((S_{i+1}, B_{i+1}), \ldots, (S_m, B_m)\) with advantage at least \( \delta \). In contrast to similar lemmas in the literature, we give a more concrete statement which will be useful later on.

For the remainder of this section, we define, for an understood parameter \( \gamma > 0 \),
\[
    \varphi_{\text{isol}} = 2 \left( \frac{24}{\gamma} \right)^2 \cdot \ln \left( \frac{7}{\gamma} \right).
\]
Note that the value \( \varphi_{\oplus} \) used above corresponds to the special case where \( \gamma := \gamma/m \) for the understood parameter \( \gamma \). The values \( l_1, l_2, \ldots \) are defined as above (before the statement of Theorem 3.15) with respect to \( \varphi_{\text{isol}} \). Finally, we introduce the shorthands (for \( i \leq j \))
\[
    B_{[i,j]} := B_i \oplus \cdots \oplus B_j,
    S_{[i,j]} := S_i \| \cdots \| S_j,
    A_{S_{[i,j]}[\sigma_i, \ldots, \sigma_j]} := A_{S_i[\sigma_i]} \| \cdots \| A_{S_j[\sigma_j]}.
\]

\textsuperscript{22} An incomparable generalization of the XOR Lemma for stateful interactive systems was proposed by Halevi and Rabin [HR08]. However, it relies on sequential (rather than parallel) access to the systems \( S_1, \ldots, S_m \), which is not sufficient for the applications of this thesis.
Lemma 3.16 (Isolation Lemma). Let \( i \in \{1, \ldots, m - 1\} \), let \( \tau > 0 \), and let \( A \) be an adversary with complexity \( t \) making \( q_j \) queries to \( S_j \) for \( j = 1, \ldots, m \). Moreover, let \( \sigma_1, \ldots, \sigma_{i-1} \) be valid states for \( A_{S_1}, \ldots, A_{S_{i-1}} \), respectively, with \( |\sigma_j| \leq l_j \) (for \( j = 1, \ldots, i-1 \)), and let \( b_{[1,i-1]}^{(i-1)} \in \{0,1\} \). Assume that

\[
\text{Guess}^A(b_{[1,i-1]}^{(i-1)} \oplus B_{[i,m]} | A_{S_{[1,i-1]}}[\sigma_1, \ldots, \sigma_{i-1}]|S_{[i,m]}) > \delta \cdot \varepsilon + \tau.
\]

Then, at least one of the following two statements is true:

(i) There exists a valid state \( \sigma_i \) for \( AS_i \) with \( |\sigma_i| \leq l_i \) and a bit \( b_{[1,i]}^{(i)} \in \{0,1\} \) such that

\[
\text{Guess}^A(b_{[1,i]}^{(i)} \oplus B_{[i+1,m]} | A_{S_{[1,i]}}[\sigma_1, \ldots, \sigma_i]|S_{[i+1,m]}) > \delta;
\]

(ii) There exists an adversary \( A_i' \) with running time \( t' \) making \( q' \) queries such that \( \text{Guess}^{A_i'}(B_i | S_i) \geq \varepsilon \), and

\[
t' = t \cdot \phi_{\text{isol}} \cdot \left[ t + O(\sum_{j=1}^{i-1} t_{A_{S_j}}(q_j, l_j) + \sum_{j=i+1}^{m} t_{A_{S_j,B_j}}(q_j)) \right]
\]

and \( q' = \phi_{\text{isol}} \cdot q_i \).

The proof of the Isolation Lemma is postponed to Section 3.3.5. The next paragraph shows how the full XOR Lemma can be obtained from the Isolation Lemma. Note that we do not put priority on optimizing values (such as the function \( \phi_{\text{isol}} \) defined above), but rather on having simpler expressions, which are in particular independent of the upper bounds on the underlying advantages.

From the Isolation Lemma to the XOR Lemma. In the following, fix some \( t, q_1, \ldots, q_m, \gamma > 0 \) and set \( \tau := \gamma / (m - 1) \). Define \( t_i, q_i \) as in the statement of Theorem 3.15, and let \( \varepsilon_i := \text{Guess}^{t_i,q_i}(B_i | S_i) \).

From now on, let \( A \) be the adversary with running time \( t \) making \( q_1, \ldots, q_m \) queries to the respective systems, and such that

\[
\text{Guess}^A(B_{[1,m]} | S_{[1,m]}) \geq \varepsilon_1 \cdots \varepsilon_m + \gamma.
\]

Define \( \gamma_i := (m - 1 - i) \cdot \tau \) for \( i = 0, \ldots, m - 1 \) and consider the statements \( \text{STAT}_i(\sigma_1, \ldots, \sigma_i, b) \) for a bit \( b \in \{0,1\} \) and valid states \( \sigma_1, \ldots, \sigma_i \).
of $A_{\sigma_1}, \ldots, A_{\sigma_i}$ respectively, with $|\sigma_j| \leq l_j$ for $j = 1, \ldots, i$ which holds if and only if

$$\text{Guess}^A(b \oplus B_{[i+1, m]} | A_{[i, \sigma_i]}[\sigma_1, \ldots, \sigma_i] || S_{[i+1, m]}) \geq \varepsilon_{i+1} \cdots \varepsilon_m + \gamma_i$$

$$\geq \varepsilon_{i+1} \cdot (\varepsilon_{i+2} \cdots \varepsilon_m + \gamma_{i+1}) + \gamma,$$

where we have used the facts that $\gamma_i := \gamma_i - 1 + \gamma$ and $\varepsilon_{i+1} \leq 1$. In particular, by our assumption $\text{STAT}_0(0)$ holds. Furthermore, given that the assumption $\text{STAT}_i(\sigma_1, \ldots, \sigma_{i-1}, b)$ holds for some $i$ we apply the Isolation Lemma (Lemma 3.16). Note that condition (ii) cannot hold, as otherwise there exists an adversary $A'_i$ such that $\text{Guess}^{A'_i}(B_i | S_i) > \varepsilon_i$, but this contradicts the assumed hardness of $(S_i, B_i)$, as we have defined $\varepsilon_i$ with respect to the maximal running time of a constructed adversary $A'_i$. Therefore, condition (i) must hold: In other words, we have shown that for all $i = 0, \ldots, m - 2$

$$\text{STAT}_i(\sigma_1, \ldots, \sigma_i, b) \Rightarrow \exists \sigma_{i+1}, b' : \text{STAT}_i(\sigma_1, \ldots, \sigma_i, \sigma_{i+1}, b \oplus b'),$$

where all the states $\sigma_1, \sigma_2, \ldots$ are valid for their respective implementations and are such that $|\sigma_i| \leq l_i$. Iterating the argument we obtain that there exist $\sigma_1, \ldots, \sigma_{m-1}$ and a bit $b$ such that

$$\text{Guess}^A(b \oplus B_m | A_{[1, m-1]}[\sigma_1, \ldots, \sigma_{m-1}] || S_m) > \varepsilon_m + \gamma_{m-1} = \varepsilon_m.$$

However, we can now consider the adversary

$$A' := A(A_{[1, m-1]}[\sigma_1, \ldots, \sigma_{m-1}] || \cdot) \oplus b$$

which, given access to $S_m$, simulates the adversary $A$ interacting with the system $A_{[1, m-1]}[\sigma_1, \ldots, \sigma_{m-1}] || S_m$, obtaining output $b'$, and finally outputs $b \oplus b'$. Such an adversary has advantage larger than $> \varepsilon_m$, contradicting the assumed hardness of $(S_m, B_m)$.

### 3.3.5. Proof of the Isolation Lemma

Throughout the proof, let $(S_i(\cdot), B_i(\cdot))$ and $S \in \mathcal{S}$ be such that we have $(S_i(S), B_i(S)) \equiv (S_i, B_i)$. Note that such a representation of $(S_i, B_i)$ exists since the system-bit pair is assumed to be cc-stateless (as $i \leq m - 1$). In particular, $B_i(s)$ depends (without loss of generality) deterministically on the input $s$. For the remainder of this proof, it is convenient to define $\gamma'_i := \gamma_i$ and for the system-bit pairs $(S_{i+1}, B_{i+1}), \ldots, (S_m, B_m)$ (and the
Note that these are only notational shorthands. In particular, the system given system, and adds \( b \) to the right-hand side) we see that for all \( s \in S \)

\[
\alpha_1(s) := 2 \cdot P[A(M(S_i(s))) \oplus b_{[i,i-1]} \oplus B_{[i+1,m]} = 1] - 1,
\]

\[
\alpha(s) := 2 \cdot P[A(M(S_i(s))) \oplus b_{[i,i-1]} \oplus B_{[i+1,m]} = B_i(s)] - 1.
\]

By rearranging terms in the definition of \( \alpha \) (i.e. by moving \( b_{[i,i-1]} \oplus B_{[i+1,m]} \) to the right-hand side) we see that \( E[\alpha(S)] > \delta' + \varepsilon + \gamma \) by the assumption of the lemma, and also \( \alpha(s) = \alpha_1(s) \) if \( B_i(s) = 1 \), and \( \alpha(s) = -\alpha_1(s) \) if \( B_i(s) = 0 \).

The remainder of the proof of the Isolation Lemma is subdivided into the following two technical lemmas.

**Lemma 3.17.** If \( P[|\alpha_1(S)| > \delta + \gamma'] > \gamma' \), then there exists \( \sigma_i \) for \( A_{S_i} \) with \(|\sigma_i| \leq l_i \) and \( b_i \in \{0,1\} \) such that

\[
\text{Guess}^{A_i}(b_{[1,i-1]} \oplus b_i \oplus B_{[i+1,m]} | M(A_{S_i} | \sigma_i)) > \delta.
\]

**Lemma 3.18.** If \( P[|\alpha_1(S)| > \delta + \gamma'] \leq \gamma' \), then there exists an adversary \( A'_i \) with running time \( t'_i \) and making \( q'_i \) queries such that \( \text{Guess}^{A'_i}(B_i | S_i) > \varepsilon \).

The Isolation Lemma is implied by these two lemmas, since either \( P[|\alpha_1(S)| > \delta + \gamma'] > \gamma' \) holds (and in this case Lemma 3.17 implies statement (i)), or \( P[|\alpha_1(S)| > \delta + \gamma'] \leq \gamma' \) holds, which yields (ii).

The remainder of this section is devoted to proving the above two lemmas.

**Proof of Lemma 3.17.** We define a random process \( \text{SAMPLE} \) (which is depicted on the left-hand side of Figure 3.12) putting out a pair \((\sigma, b)\), where \( \sigma \) is a valid state for \( A_{S_i} \), and \( b \in \{0,1\} \) is a bit. The process \( \text{SAMPLE} \) lets \( \varphi_{\text{isol}} \) independent instances of \( \overline{A}(\cdot) := A(M(\cdot)) \oplus b_{[1,i-1]} \oplus B_{[i+1,m]} \)\(^2\) sequentially interact with the same instance of \( S_i \), producing

\[2\text{i.e. the adversary which for new independent instances of } (S_{i+1}, B_{i+1}), \ldots, (S_m, B_m) \text{ simulates an interaction of } A \text{ with } A_{S_{[i,i-1]}}[\sigma_1, \ldots, \sigma_{i-1}] || S || S_{[i+1,m]} \text{, where } S \text{ is the given system, and adds } b_{[1,i-1]} \oplus B_{[i+1,m]} \text{ to its output to obtain the actual output.} \]
outputs \(o_1, \ldots, o_\varphi\|\text{isol}\) (and denote by \(\bar{\sigma}\) their average). The instance of \(S_i\) is simulated using \(A_{S_i}\) by letting, for \(j = 1, \ldots, \varphi\|\text{isol}\), the \(j\)-th independent instance of \(\bar{A}\) interact with \(A_{S_i}[\sigma_j-1]\), and setting \(\sigma_j\) to be the final state of \(A_{S_i}\) after such interaction (and \(\sigma_0 := \bot\) is the empty, initial state for \(A_{S_i}\)).

Also, it sets \(\sigma := \sigma_{\varphi\|\text{isol}}\). The process \(\text{SAMPLE}\) computes \(\bar{\sigma} := 2 \cdot \bar{\sigma} - 1\). If \(\bar{\sigma}_1\) is higher than \(\delta\), we expect \(\text{Guess}^A(b_{[1,i-1]} \oplus 1 \oplus B_{[i+1,m]} | M(A_{S_i}[\sigma])) > \delta\) (and the process hence outputs \((\sigma, 1)\)), whereas if it is lower than \(-\delta\), we expect \(\text{Guess}^A(b_{[1,i-1]} \oplus 0 \oplus B_{[i+1,m]} | M(A_{S_i}[\sigma])) > \delta\) (and \(\text{SAMPLE}\) outputs \((\sigma, 0)\)).

We consider a random experiment where \(\text{SAMPLE}\) samples a pair \((\sigma, b)\), and we subsequently compute \(A(M(A_{S_i}[\sigma])) \oplus b_{[1,i-1]} \oplus B_{[i+1,m]}\).

We define the event \(T\) that one of the \texttt{if}-statements within process \(\text{SAMPLE}\) is executed, and we show that under the assumption that \(P[|\alpha_1(S)| > \delta + \bar{\sigma}_1] > \bar{\sigma}\) we have \(P[T] > 0\) and

\[
\bar{\sigma}_1 \geq \frac{1 + \delta}{2},
\]

for

\[
\bar{\sigma}_1 := P[(\sigma, b) \leftarrow \text{SAMPLE} : \text{A}(M(A_{S_i}[\sigma])) \oplus b_{[1,i-1]} \oplus B_{[i+1,m]} = b | T].
\]

Note that this is sufficient to obtain the statement of Lemma 3.17, as we can choose a pair \((\sigma, b)\) output by \(\text{SAMPLE}\) conditioned on the event \(T\) (note that this yields a well-defined distribution of pairs \((\sigma, b)\) maximizing the above probability, and set \(\sigma_i := \sigma, b_{[1,i-1]} := b_{[1,i-1]} \oplus b\). Since \(A\) issues at most \(q_i\) queries, and the above process is repeated \(\varphi\|\text{isol}\) times, we have \(|\sigma_i| \leq s_{A_{S_i}}(q_i \cdot \varphi\|\text{isol}) = l_i\).

In order to prove inequality (3.3), we consider a second sampling process (called \(\text{SAMPLE}'\) and depicted on the right-hand side of Figure 3.12) which samples \(s\) according to \(P_S\) and subsequently computes the values \(o_1, \ldots, o_\varphi\|\text{isol}\) by letting \(\varphi\|\text{isol}\) independent instances of \(\bar{A}\) interact with \(S_i(s)\). The process finally outputs the pair \((s, b)\) with \(b\) being computed as in \(\text{SAMPLE}\). In particular, we also denote here as \(T\) the event that one of the two conditions in the \texttt{if}-statement is satisfied, and we define

\[
\pi_2 := P[(s, b) \leftarrow \text{SAMPLE}' : \text{A}(M(S_i(s))) \oplus b_{[1,i-1]} \oplus B_{[i+1,m]} = b | T].
\]

A crucial observation is that the sampling processes \(\text{SAMPLE}\) and \(\text{SAMPLE}'\) compute the values \(o_1, \ldots, o_\varphi\|\text{isol}\) by letting multiple, independent, instances of \(\bar{A}\) interact with the same instance of \(S_i\), and furthermore, we are interested in the probabilities \(\pi_1\) and \(\pi_2\) that a further, independent, instance of \(\bar{A}\) is successful in guessing the bit \(b\) when interacting
3.3 The XOR Lemma for System-Bit Pairs

Process SAMPLE:

\[
\begin{align*}
  b &:= 0; \\
  \sigma_0 &:= \text{initial state of } A_S \\
  \text{for all } j &= 1, \ldots, \varphi_{\text{isol}} \text{ do} \\
  o_j &:= A(M(A_S[\sigma_{j-1}]) \oplus b_{[1,i-1]} \oplus B_{[i+1,m]}) \\
  \sigma_j &:= \text{final state of } A_S \\
  \alpha_1 &:= 2 \cdot \frac{1}{\varphi_{\text{isol}}} \cdot \sum_{j=1}^{\varphi_{\text{isol}}} o_j - 1 \\
  \text{if } \alpha_1 > \delta + (2/3)^{\gamma'} &\text{ then} \\
  b &:= 1 \\
  \text{else if } \alpha_1 < -\delta - (2/3)^{\gamma'} &\text{ then} \\
  b &:= 0 \\
  \text{return } (\sigma, b)
\end{align*}
\]

Process SAMPLE':

\[
\begin{align*}
  b &:= 0 \\
  s &\leftarrow P_{S_i} \\
  \text{for all } j &= 1, \ldots, \varphi_{\text{isol}} \text{ do} \\
  o_j &:= A(M(S_i(s)) \oplus b_{[1,i-1]} \oplus B_{[j]}) \\
  \alpha_1 &:= 2 \cdot \frac{1}{\varphi_{\text{isol}}} \cdot \sum_{j=1}^{\varphi_{\text{isol}}} o_j - 1 \\
  \text{if } \alpha_1 > \delta + (2/3)^{\gamma'} &\text{ then} \\
  b &:= 1 \\
  \text{else if } \alpha_1 < -\delta - (2/3)^{\gamma'} &\text{ then} \\
  b &:= 0 \\
  \text{return } (s, b)
\end{align*}
\]

Figure 3.12: Sampling processes SAMPLE and SAMPLE' used in the proof of Lemma 3.17.
(once again) with the same instance of $S_i$. In the first random experiment, the instance of $S_i$ is simulated by using the algorithm $A_{S_i}$. In the second case, this is done by choosing a random $s$ and running $S_i(s)$. However, in both cases, all probabilities (including the event $T$) only depend on the input-output behavior of $S_i$, and not on the way this instance is simulated, and hence we have $\pi_1 = \pi_2$, and $T$ occurs with the same probability in both experiments.\footnote{Informally, this means that if we simulate a system $S = S(S)$ which is used only through black-box access, and we do this by means of an algorithm $G$, we implicitly sample a corresponding $S$ during the interaction even though such $S$ does not actually exist.}

In the remainder of the proof we thus show that $\pi_2 \geq 1 + \frac{1 + \delta}{2}$, which is much simpler than working with the original random experiment, as for each value $s$ the system $S_i(s)$ is stateless and we can conveniently upper-bound the error in the estimate of $\pi_T$ (using e.g. Hoeffding’s inequality) due to the fact that the random variables $o_1, \ldots, o_{\varphi_{\text{isol}}}$ conditioned on some fixed value $s$, are statistically independent. More formally, we denote the values $\pi_T$ and $o_j$ associated with a particular choice of $s$ as $\pi_T(s)$ and $o_j(s)$, respectively. We additionally define (for every $\eta \geq 0$) the set

$$G_{\eta} := \{ s \in S \mid |o_1(s)| > \delta + \eta \}.$$

Note that $G_{\eta} \subseteq G_{\eta'}$ for all $\eta > \eta'$ and $P[s \in G_{\pi_T}] > \pi_T$ by the above assumption. Furthermore, with a slight abuse of notation, we denote by $G_{\eta}$ the event $s \in G_{\eta}$. In the following, we show that $P[G_{\pi_T/3} \land T]$ is overwhelming.

Note that for a fixed $s$ (because of the fact that $S_i(s)$ is stateless) the random variables $o_j(s)$ (which here we exceptionally denote by lowercase letters) are independent binary variables, with

$$E[o_j(s)] = p_1(s) = P[A(M(S_i(s))) \oplus b_{[1,i-1]} \oplus B_{[i+1,m]} = 1]$$

for all $j = 1, \ldots, \varphi_{\text{isol}}$. First, by Hoeffding’s bound, and because of the factor two in the definition of $\alpha_1$, we have for every fixed $s \in S$,

$$P[|\alpha_1(s) - \overline{\pi_T}(s)| > \pi_T/3] \leq P \left[ \left| \frac{1}{\varphi_{\text{isol}}} \sum_{i=1}^{\varphi_{\text{isol}}} o_i(s) - p_1(s) \right| > \pi_T/6 \right]$$

$$\leq 2 \cdot e^{-\varphi_{\text{isol}}(\pi_T/6)^2}.$$

This in particular implies that for the randomly chosen $s$,

$$P[G_{\pi_T/3} \land T] \leq P[T \mid G_{\pi_T/3}]$$

$$\leq P[|\alpha_1(s) - \overline{\pi_T}(s)| > \pi_T/3 \mid s \notin G_{\pi_T/3}] \leq 2 \cdot e^{-\varphi_{\text{isol}}(\pi_T/6)^2}.$$
3.3 The XOR Lemma for System-Bit Pairs

This yields
\[
P[G_{\gamma'/3} | T] = \frac{P[G_{\gamma'/3} \land T]}{P[G_{\gamma'/3} \land T] + P[G_{\gamma'/3} \land T]}
\]
\[
\geq \frac{P[G_{\gamma'/3} \land T]}{P[G_{\gamma'/3} \land T] + 2 \cdot e^{-\varphi_{\text{isol}}(\gamma'/6)^2}}.
\]

As the function \( x \mapsto \frac{x}{1+x} \) is non-decreasing over \([0,1]\) for every constant \( c \in [0,1]\), it is now sufficient to find a non-trivial lower bound for \( P[G_{\gamma'/3} \land T] \): Since \( G_{\gamma'} \subseteq G_{\gamma'/3} \), we have
\[
P[G_{\gamma'/3} \land T] \geq P[G_{\gamma'} \land T] \geq P[G_{\gamma'}] \cdot P[T | G_{\gamma'}]
\]
\[
\geq \gamma' \cdot (1 - 2 \cdot e^{-\varphi_{\text{isol}}(\gamma'/6)^2}) \geq \gamma' - 2 \cdot e^{-\varphi_{\text{isol}}(\gamma'/6)^2},
\]

since \( P[T | G_{\gamma'}] \leq P[|\alpha_1(S) - \alpha_T(S)| > \gamma'/3 | s \in G_{\gamma'}] \). (Note that the above inequality implies in particular \( P[T | T] > 0 \).) Plugging this into (3.4) yields
\[
P[G_{\gamma'/3} | T] \geq 1 - \frac{2}{\gamma'} \cdot e^{-\varphi_{\text{isol}}(\gamma'/6)^2},
\]
and thus we conclude
\[
\pi_1 = \pi_2
\]
\[
\geq P[G_{\gamma'/3} | T] \cdot P[A(M(S_i(S))) = b_{[1, i-1]} \oplus b \oplus B_{[i+1, m]} | S \in G_{\gamma'/3} \land T] \geq \left( 1 - \frac{2}{\gamma'} \cdot e^{-\varphi_{\text{isol}}(\gamma'/6)^2} \right) \cdot \left( 1 + \delta + \gamma'/3 \right) \geq \frac{1 + \delta}{2} + \frac{\gamma'}{6} \cdot e^{-\varphi_{\text{isol}}(\gamma'/6)^2} > \frac{1 + \delta}{2},
\]

by the definition of \( \varphi_{\text{isol}} \).

**Proof of Lemma 3.18.** We construct the adversary \( A'_i \) for predicting the bit \( B_i \) given access to \( S_i \) as described in Figure 3.13: It estimates the value \( \alpha_1 \) by repeatedly simulating the execution of \( A(M(\cdot)) \oplus b_{[1, i-1]} \oplus B_{[i+1, m]} \) with the system \( S_i \), and finally outputs 1 with probability \( \frac{1}{2} + \frac{\pi_1}{2(\gamma + \gamma')} \).

As above, we assume that \( (S_i, B_i) \) is instantiated by first sampling \( s \in S \) according to \( P_S \) and then behaving as \( (S_i(s), B_i(s)) \). This allows us to analyze the behavior of \( A'_i \) conditioned on each value \( s \in S \). In particular, we denote by \( \pi_T(s) \) the value obtained when run on \( S(s) \).

We first consider the event \( \mathcal{E} \) that for the chosen \( S = s \) we have \( |\alpha_1(s) - \alpha_1(s)| > \gamma' \). Note that the individual runs of the independent instances give independent outputs because \( S_i(s) \) is stateless. Hence, by
Adversary $A'_i$: // given access to $S_i$

for all $j := 1, \ldots, \varphi_{\text{isol}}$ do
  $o_j := A(M(S_i)) \oplus b_{i,i-1} \oplus B_{i+1}$
  $\alpha_1 := 2 \cdot \frac{1}{\varphi_{\text{isol}}} \cdot \sum_{j=1}^{\varphi_{\text{isol}}} o_j - 1$
return 1 with probability $\min\{\frac{1}{2} + \frac{\varphi_{\text{isol}}}{2(\varepsilon + \gamma')}, 0\}, 1}$

**Figure 3.13:** Adversary $A'_i$ in the proof of Lemma 3.18.

Hoeffding’s bound, $P[\mathcal{E}] \leq 2 \cdot e^{-\varphi_{\text{isol}}(\gamma/8)^2}$. Also as above, we let $\mathcal{G}_{\gamma'}$ the set of values $s \in S$ such that $|\alpha_1(s)| \leq \delta + \gamma'$. (In particular, $P[\mathcal{G}_{\gamma'}] \leq \gamma'$, by the assumption of the lemma.) For all $s \in \mathcal{G}_{\gamma'}$, we have

$$P[A'_i(S_i(s)) = B_i(s) \land \mathcal{E}] = P[A'_i(S_i(s)) = B_i(s) \mid \mathcal{E}]$$

as well as

$$P[A'_i(S_i(s)) = B_i(s) \mid \mathcal{E}] \geq \frac{1}{2} + \frac{\alpha(s) - \gamma'}{2(\delta + \gamma')}.$$  (3.5)

Moreover,

$$\sum_{s \in \mathcal{G}_{\gamma'}} P_S(s) \cdot \alpha(s) = \sum_{s \in \mathcal{S}} P_S(s) \cdot \alpha(s) - \sum_{s \in \mathcal{G}_{\gamma'}} P_S(s) \cdot \alpha(s)$$

$$\geq E[\alpha(s)] - P[\mathcal{G}_{\gamma'}] \geq \delta \cdot \varepsilon + \gamma - \gamma',$$  (3.7)

which finally yields

$$P[A'_i(S_i(s)) = B_i(s) \land s \in \mathcal{G} \mid \mathcal{E}] = \sum_{s \in \mathcal{G}} P_S(s) \cdot P[A'_i(S_i(s)) = B_i(s) \mid \mathcal{E}]$$

$$\geq \frac{1}{2} + \frac{\sum_{s \in \mathcal{G}} P_S(s) \cdot \alpha(s) - \gamma'}{2(\delta + \gamma')}$$

$$\geq \frac{1}{2} + \frac{\delta \cdot \varepsilon + \gamma - 2\gamma'}{2(\delta + \gamma')}$$

$$= \frac{1}{2} + \frac{\varepsilon \cdot (\delta + \gamma') + \gamma - \delta \cdot \gamma' - 2\gamma'}{2(\delta + \gamma')}$$

$$\geq \frac{1}{2} + \frac{\gamma'}{2(\delta + \gamma')} \geq \frac{1}{2} + \frac{\gamma'}{2} + \frac{\gamma'}{4}.$$
3.3 The XOR Lemma for System-Bit Pairs

Hence, using Equation 3.5 we conclude that

\[ P[A'_i(S_i) = B_i] \geq \frac{1 + \varepsilon}{2} + \frac{\gamma}{4} - 2e^{-\varphi_{\text{isol}}(\gamma/8)} = \frac{1 + \varepsilon}{2} \]

for the given value of \( \varphi_{\text{isol}} \).

As for the complexity of the adversary, we note that the adversary \( A' \) needs to hard-code a description of the states \( \sigma_1, \ldots, \sigma_{i-1} \) (needing at most \( l_i \) bits, which are hence counted only once in the time complexity) and runs \( \varphi_{\text{isol}} \) copies of \( A \), making each time \( q_i \) queries to \( S_i \), and we additionally need to simulate \( A_{S_j}[\sigma_j] \) for all \( j = 1, \ldots, i - 1 \) (which takes time \( t_{A_{S_j}}(q_j, l_j) \)), and need to simulate new instances of \( (S_j, B_j) \) for \( j = i + 1, \ldots, m \) (which takes time \( t_{A_{S_j, B_j}}(q_j) \)).

3.3.6. The Uniform XOR Lemma

This section presents a uniform version of the XOR Lemma (Theorem 3.15) using fairly standard techniques (cf. e.g. [GNW95]), and we thus dispense with a quantitatively precise analysis of the reduction: We only discuss the main modifications to the proof of Theorem 3.15. In particular, we only consider the case \( (S_1, B_1) \equiv \cdots \equiv (S_m, B_m) \equiv (S, B) \).

**Theorem 3.19** (XOR Lemma). Let \( (S_1, B_1), \ldots, (S_m, B_m) \) be independent instances of an efficient cc-stateless system-bit pair \( (S, B) \) with

\[ \text{Guess}_{\text{poly}}(B \mid S) \leq \varepsilon. \]

Then,

\[ \text{Guess}_{\text{poly}}(B_1 \oplus \cdots \oplus B_m \mid S_1 \parallel \cdots \parallel S_m) \leq \varepsilon^m + \nu \]

for a negligible function \( \nu \).

**Proof (Sketch).** The proof is by contradiction, and starts by assuming the existence of an efficient uniform adversary \( A \) such that

\[ \text{Guess}_{\text{poly}}(B_1 \oplus \cdots \oplus B_m \mid S_1 \parallel \cdots \parallel S_m) > \varepsilon^m + \gamma \]

for a noticeable function \( \gamma \) and infinitely many values of the security parameter \( k \). Given black box access to the adversary \( A \), we build a new adversary \( A' \) for predicting \( B \) given access to \( S \) operating as follows:

(1) Initially, it sets \( b_{[1,0]} := 0 \).
(2) For each $i = 1, \ldots, m - 1$, given states $\sigma_1, \ldots, \sigma_{i-1}$, it repeatedly and independently runs process SAMPLE, without the final return statement. We consider two cases:

(a) If the if-statement is satisfied in some run of SAMPLE (which would have output $(\sigma, b)$), then we set $\sigma_i := \sigma$ and $b_{[1,i]} := b \oplus b_{[1,i-1]}$, and move to the next index $i + 1$.

(b) If after a number $i_{\text{max}} := -k / \log(1 - \lambda'(k))$ of runs with the same index $i$ of SAMPLE the if-statement was never satisfied, we run the adversary $A'_i$ with the found states $\sigma_1, \ldots, \sigma_{i-1}$ and $b_{[1,i-1]}$ as above against the given system $S$ to guess $B$.

(3) If the for loop is terminated (i.e. Case (b) was never met, and we end up with $i = m$), then states $\sigma_1, \ldots, \sigma_{m-1}$ and a bit $b_{[1,m-1]}$ have been found. We use them to run the adversary $A(G[\sigma_1] || \cdots || G[\sigma_{m-1}] || \cdot)$ on $S$ and add $b_{[1,m-1]}$ to the resulting output.

The main idea is that for given $i$, in the case where $|\alpha_1(S)| > \varepsilon^{m-i} + \tau'$ holds with probability at least $\tau'$, then a good state $\sigma_i$ is found with overwhelming probability within the given number of iterations $i_{\text{max}}$, and we can move to the next $i$. If for some $i$ the if-statement is never satisfied, then with overwhelming probability $|\alpha_1(S)| \leq \varepsilon^{m-i} + \tau'$, and thus the attacker $A'_i$ is successful.
Chapter 4

Security Amplification for Neutralizing Constructions

The XOR Lemma (Theorem 3.15) solves the problem of amplifying the distinguishing advantage for weakly pseudorandom bits: This is a special case of the general problem where real systems $S_1, \ldots, S_m$ and ideal systems $I_1, \ldots, I_m$ are given, and we look for a construction $C(\cdot)$ for which $\Delta_{t,q}(C(S_1, \ldots, S_m), C(I_1, \ldots, I_m))$ is much smaller than all of the $\Delta_{t',q'}(S_i, T_i)$, and, ideally, $t$ and $q$ are as close as possible to $t'$ and $q'$.

In this chapter, we consider a large class of constructions $C(\cdot)$, which we refer to as neutralizing, with the property that

$$C(T_1, \ldots, T_m) \equiv C(I_1, \ldots, I_m)$$

whenever there exists $T_i$ such that $T_i \equiv I_i$. These constructions abstract the notion of a combiner [Her05] for computational indistinguishability: It suffices that one system $T_i$ is (computationally) indistinguishable from the corresponding ideal system $I_i$ (and thus is “secure”) in order for $C(T_1, \ldots, T_m)$ to be computationally indistinguishable from the ideal system $C(I_1, \ldots, I_m)$.

We first prove that any such combiner ensures a mild form of computational indistinguishability amplification, whereas for a subclass of these constructions we show nearly optimal computational indistinguishability amplification.
Chapter Outline and Contributions. The results of this chapter have first appeared in [MT09], and their presentation is structured as follows:

(i) Section 4.1 reviews the concept of neutralizing constructions, first introduced by Maurer, Pietrzak, and Renner [MPR07], as well as a few examples. Moreover, the information-theoretic product theorem for neutralizing constructions of [MPR07] is re-stated (Theorem 4.1) in Section 4.2.1, and we recast its proof in terms of an explicit reduction to the XOR Lemma for system-bit pairs.

A similar technique is employed, in Section 4.2.2, to prove Theorem 4.2, which is our first main result and a computational analogue of Theorem 4.1: Broadly speaking, it implies that any neutralizing construction achieves computational indistinguishability amplification with respect to the distinguishing advantage as long as the (computational) indistinguishability advantage in distinguishing every subsystem $S_i$ from the corresponding ideal system $I_i$ is smaller than $\frac{1}{2}$.

(ii) Direct corollaries of Theorem 4.2 are discussed in Section 4.3, such as a tight product theorem for PRF composition under quasi-group operations (like XOR), that improves on previous work by Dodis et al. [DIJK09], as well as a simple proof that the cascade of random permutations amplifies the security of the underlying permutations. While the latter result will be improved in the next chapter, the first result yields tightness of Theorem 4.2 at this level of generality.

(iii) Section 4.4 considers a large class of randomized neutralizing constructions, ensuring a property called self independence (of the corresponding ideal systems), for which we prove a strong product theorem (Theorem 4.6), i.e., such construction amplifies computational indistinguishability in a nearly optimal fashion even if the underlying computational advantages $\Delta_{v,q}(S_i, I_i)$ are close to 1. Theorem 4.6 is then exercised for different constructions in Section 4.5:

- Our main corollary implies that a small modification of the cascade of random permutations, where two random offsets are added at both ends, suffices to achieve strong security amplification.
- We also prove that in the case of random-input security, combining random functions with a quasi-group operation is sufficient to achieve strong security amplification, in contrast with the chosen-input setting.
4.1 Neutralizing Constructions

Finally, in the case of chosen-input security, strong security amplification can be achieved by adding a secret random offset to the input. This improves on a previous construction by Myers [Mye99, Mye03].

The results of this chapter are presented in the non-uniform setting only. The modifications to obtain uniform statements are fairly straightforward, and follow the same approach as in the proof of the uniform XOR Lemma (Theorem 3.19).

Related Work. The concept of a neutralizing construction was introduced by Maurer, Pietrzak and Renner [MPR07], allowing a unified treatment of indistinguishability amplification with respect to the information-theoretic distinguishing advantage. Previous results in this setting have been obtained by Vaudenay [Vau98, Vau99] within the framework of decorrelation theory.

In contrast, only limited work has addressed amplification results with respect to the computational distinguishing advantage. The first product theorem is due to Luby and Rackoff [LR86], who proved that the cascade of two PRPs is strictly stronger than both of the individual components. Yet, extensions of this result to longer cascades (as done by Myers [Mye99]) only apply to constant-length cascades, due to the high complexity of the resulting reduction: This approach thus falls short of proving amplification from a non-negligible advantage to a negligible one, which is our ultimate goal. The first product theorem for PRFs (which follows as a corollary from Theorem 4.6) appeared in [Mye99] (the result was subsequently published in [Mye03]). Only more recently, Dodis et al. [DIJK09] provided the first product theorem for PRFs via simple XOR composition, which we improve quantitatively as an application of Theorem 4.2.

We additionally stress that our work on cascades is in contrast with a previous line of work focusing on generic attacks against cascades [EG85, MM93, BR06, GM09a], where one only assumes black-box access to the underlying block cipher, usually modeled as an ideal cipher.

4.1. Neutralizing Constructions

We start with the following definition, which differs slightly from the one given in [MPR07].
Definition 4.1. A construction \( C(\cdot) \) is called neutralizing for system classes \( S_1, \ldots, S_m \) and ideal systems \( I_1, \ldots, I_m \) (with \( I_i \in S_i \) for all \( i = 1, \ldots, m \)) if, for all \( S_1, \ldots, S_m \) such that \( S_i \in S_i \) (for \( i = 1, \ldots, m \)), we have
\[
C(S_1, \ldots, S_m) \equiv C(I_1, \ldots, I_m)
\]
whenever there exists some \( i \) with \( S_i \equiv I_i \).

Neutralizing constructions capture the notion of a combiner \cite{Her05} for computational indistinguishability properties: Whenever at least one system \( S_i \in S_i \) is computationally indistinguishable from \( I_i \), then the systems \( C(S_1, \ldots, S_m) \) and \( C(I_1, \ldots, I_m) \) are computationally indistinguishable. Typical examples for the classes \( S_i \) are the classes of random variables, random functions, and random permutations. Without loss of generality, we assume that each such system class \( S \) satisfies the natural property that if \( S \in S \), then fixing (even partially) the random choices in some implementation of \( S \) yields a system in \( S \) as well. In particular, it is convenient to say that \( C(\cdot) \) is neutralizing for \( S_1, \ldots, S_m \) and \( I_1, \ldots, I_m \) if it is neutralizing for some classes \( S_1, \ldots, S_m \) with \( S_i \in S_i \) and \( I_i \in S_i \) for \( i = 1, \ldots, m \).

We provide two examples of neutralizing constructions with two subsystems. (Both constructions extend naturally to an arbitrary number of subsystems.)

Example 4.1. Every quasi-group operation\(^1 \) \( \ast \) on a finite set \( \mathcal{Y} \) induces a construction \( C(\cdot) \) such that \( C(F, G) := F \ast G \) which is neutralizing for any two random functions \( F \) and \( G : \mathcal{X} \to \mathcal{Y} \) and ideal systems \( I \) and \( J \) being independent URPs. In particular, \( I \ast J \) is also a URF. As a special case, this result holds for random variables \( X, Y \) over \( \mathcal{Y} \), the ideal systems being uniform random elements of \( \mathcal{Y} \).

Example 4.2. The cascade operator \( \triangleright \) induces a construction \( C'(\cdot) \) with \( C'(Q_1, Q_2) := Q_1 \triangleright Q_2 \) which is neutralizing for any two cc-stateless random permutations \( Q_1 \) and \( Q_2 : \mathcal{X} \to \mathcal{X} \) (in fact \( Q_1 \) can possibly be stateful) where the ideal systems \( I \) and \( J \) are both URPs \( \mathcal{X} \to \mathcal{X} \). In particular, \( I \triangleright J \) is also a URP. If \( Q_1 \) is cc-stateless, then the same result holds even in the two-sided case for \( \langle Q_1 \rangle \) and \( \langle Q_2 \rangle \) (with ideal system \( \langle P \rangle \) for a URP \( P \)).

\(^1\)That is, given \( a, c \in \mathcal{Y} \) (or \( b, c \in \mathcal{Y} \)) there exists a unique \( b \) (a) such that \( a \ast b = c \). An example is bit-wise XOR \( \oplus \) for \( \mathcal{Y} = \{0, 1\}^n \), and any group operation is a quasi-group operation as well.
4.2. Product Theorem for Neutralizing Constructions

4.2.1. The Information-Theoretic Product Theorem

Throughout this and the following subsections, let $C(\cdot)$ be a neutralizing construction for the real systems $S_1, \ldots, S_m$ and the corresponding ideal systems $I_1, \ldots, I_m$. We re-state (and give a proof in slightly different terms) the general product theorem upper bounding the distinguishing advantage $\Delta_q(C(S_1, \ldots, S_m), C(I_1, \ldots, I_m))$ in terms of the individual advantages $\Delta_{q_i}(F_i, I_i)$, with $q_i$ being the number of queries $C(\cdot)$ makes to $i$-th subsystem when queried $q$ times. The proof is illustrative for conveying the main ideas underlying the proof of Theorem 4.2 below.

**Theorem 4.1.** Let $C(\cdot)$ be as above, and let $q > 0$ be such that $C(\cdot)$ makes $q_i$ queries to its $i$-th subsystem when invoked $q$ times. Then,

$$\Delta_q(C(S_1, \ldots, S_m), C(I_1, \ldots, I_m)) \leq 2^{m-1} \cdot \prod_{i=1}^m \Delta_{q_i}(S_i, I_i).$$

**Proof.** The core of the proof is a generic argument (i.e. it holds for all distinguishers) reducing the task of upper bounding the distinguishing advantage $\Delta_q(C(S_1, \ldots, S_m), C(I_1, \ldots, I_m))$ for a neutralizing construction $C(\cdot)$ to an instance of the XOR-lemma. Define

$$S := (S_1, \ldots, S_m) \quad \text{and} \quad I := (I_1, \ldots, I_m),$$

and let $B$ a bit such that $P_B(0) = \frac{1}{2^{m-1}}$. It is easy to verify that for all distinguishers $D$,

$$\Delta^D(C(S), C(I)) = 2^{m-1} \cdot \Delta^D(\langle C(S), C(I) \rangle_B, C(I))$$

$$= 2^{m-1} \cdot \left[ \text{Guess}^D(B' | \langle C(S), C(I) \rangle_B, C(I))_{B'} \right].$$

$B'$ is a uniform random bit, independent of $B$. Note that conditioned on $B' = 0$, the system $\langle C(S), C(I) \rangle_B, C(I))_{B'}$ behaves as $C(S)$ with probability $\frac{1}{2^{m-1}}$, and as $C(I)$ otherwise (i.e., with probability $1 - \frac{1}{2^{m-1}}$). On the other hand, conditioned on $B' = 1$ it always behaves as $C(I)$. In particular, this implies that (for independent uniform random bits $B_1, \ldots, B_m$), the following two system-bit pairs are equivalent

$$\langle \langle C(S), C(I) \rangle_B, C(I) \rangle_{B'}, B' \rangle \equiv \langle (S_1, I_1)_{B_1}, \ldots, (S_m, I_m)_{B_m} \rangle, B_1 \oplus \cdots \oplus B_m,$$

\[\text{This only appears implicitly in the information-theoretic product theorem of [MPR07].}\]
because of the neutralizing property. We thus obtain

\[ \text{Guess}^D(B' | \langle C(S), C(I) \rangle_{B_i}, C(I)) = \]
\[ = \text{Guess}^D(B_1 \oplus \cdots \oplus B_m | C(S_1, I_1), \ldots, (S_m, I_m)) \]
\[ = \text{Guess}^D(C(B_1 \oplus \cdots \oplus B_m | \langle S_1, I_1 \rangle_{B_i}, \ldots, \langle S_m, I_m \rangle_{B_i})) \leq \text{Guess}_{q_1, \ldots, q_m}(B_1 \oplus \cdots \oplus B_m | \langle S_1, I_1 \rangle_{B_i}, \ldots, \langle S_m, I_m \rangle_{B_i}), \]

where we have “absorbed” the computation of \( C(\cdot) \) into \( D \), clearly without modifying the advantage. Using the (information theoretic) XOR Lemma (Theorem 3.13), we have

\[ \Delta_q(C(S), C(I)) \leq 2^{m-1} \cdot \prod_{i=1}^{m} \text{Guess}_{q_i}(B_i | \langle S_i, I_i \rangle_{B_i}) \]
\[ = 2^{m-1} \cdot \prod_{i=1}^{m} \Delta_{q_i}(S_i, I_i). \]

A computational version of Theorem 4.1 seems easy to obtain, as we can simply replace the use of the information-theoretic XOR lemma by the computational one. Yet, a major issue is due to the presence of the factor \( 2^{m-1} \): Namely, when using the computational XOR lemma, the additive term \( \gamma \) is also multiplied by \( 2^{m-1} \), and the term \( 2^{m-1} \cdot \gamma \) is generally too large (or, in other words, in order to make it sufficiently small, the resulting \( \gamma \) would make the reduction inefficient). Below, we overcome this issue by considering the case \( m = 2 \) first, and subsequently use this as an “Isolation Lemma” to obtain the full theorem.

### 4.2.2. The Computational Product Theorem

From now on, assume that all systems, except for possibly \( S_m \) and \( I_m \) are cc-stateless. The computational version of Theorem 4.1 relies on the canonical implementation \( A_{\langle S_i, I_i \rangle_{B_i}} \) of \( \langle S_i, I_i \rangle_{B_i} \) which chooses a random bit \( B_i \in \{0,1\} \) and answers each query using the implementations \( A_{S_i} \) and \( A_{I_i} \) (with respective complexities \( t_{A_{S_i}} \) and \( t_{A_{I_i}} \)) of \( S_i \) or of \( I_i \), respectively, depending on the value of \( B_i \). It can be implemented with complexity \( t_{A_{\langle S_i, I_i \rangle_{B_i}}(q, s)} = \max\{t_{A_{S_i}}(q, s), t_{A_{I_i}}(q, s)\} + O(1) \). This also yields an implementation of \( \langle S_i, I_i \rangle_{B_i}, B_i \) with the same complexity (by additionally outputting the bit \( B_i \)). We let \( \varphi_S \) be as in Section 3.3.3, and for all \( i \), we denote \( l_i := s_{A_{\langle S_i, I_i \rangle_{B_i}}(q_i \cdot \varphi_S)} \) and \( l_{<i} := \sum_{j=1}^{i-1} l_j \) (for understood \( q_1, \ldots, q_{m-1} \)). Finally, we let \( t_{A_C} \) be the time complexity of an efficient implementation \( A_C \) of \( C(\cdot) \).
4.2 Product Theorem for Neutralizing Constructions

Theorem 4.2. For all \( t, q > 0 \) and \( \gamma > 0 \), if \( \Delta_{t', q'}(S_i, I_i) \leq \frac{1}{2} \) for all \( i = 1, \ldots, m - 1 \), then

\[
\Delta_{t, q}(C(S_1, \ldots, S_m), C(I_1, \ldots, I_m)) \leq 2^{m-1} \cdot \prod_{i=1}^{m} \Delta_{t', q'}(S_i, I_i) + 2\gamma,
\]

where

\[
t'_i := l_{<i} + \varphi_{\text{isol}} \cdot \left[ t + t_{A_{\mathcal{C}}}(q) + O\left( \sum_{j=1}^{i-1} t_{A_{\mathcal{S}, I_{\mathcal{B}}}}(q_j, l_j) + \sum_{j=i+1}^{m} t_{A_{\mathcal{S}, I_{\mathcal{B}}}}(q_j) \right) \right]
\]

and \( q'_i := \varphi_{\text{isol}} \cdot q_i \) for all \( i = 1, \ldots, m - 1 \), whereas

\[
t'_m := l_{<m} + t + t_{C}(q) + O\left( \sum_{j=1}^{m-1} t_{A_{\mathcal{S}, I_{\mathcal{B}}}}(q_j, l_j) \right) \quad \text{and} \quad q'_m := q_m.
\]

In the remainder of this section, we provide a full proof of the theorem, whereas Section 4.3 presents some applications.

The key technique used in the proof of Theorem 4.2 is a version of the Isolation Lemma (Lemma 3.16) in the setting of computational indistinguishability amplification. While this follows the same lines as the proof of Theorem 4.1 given above for the case \( m = 2 \), we rely on the stronger statement given by the Isolation Lemma (Lemma 3.16) in order to prove the result for the most general setting where no additional structural requirements are made on the construction \( C(\cdot) \).

**Isolation Lemma.** For the sake of simplifying notation, for all \( i = 1, \ldots, m \), we denote as \( A_i := A_{\{S_i, I_i\}_{\mathcal{B}}} \) the implementation of the system \( \langle S_i, I_i \rangle_{\mathcal{B}} \). It is also convenient to write \( C(S_1 \parallel \ldots \parallel S_m) \) instead of \( C(S_1, \ldots, S_m) \), i.e., we see the construction as accessing the parallel composition of all subsystems, rather than the individual subsystems. Furthermore, we define (for \( i \leq j \)) the shorthands

\[
S_{[i,j]} := S_i \parallel \ldots \parallel S_j \\
I_{[i,j]} := I_i \parallel \ldots \parallel I_j \\
A_{[i,j]}[\sigma_1, \ldots, \sigma_j] := A_i[\sigma_i] \ldots \parallel A_j[\sigma_j].
\]

Also, let \( \varphi_{\text{isol}} \) be as in Section 3.3.4.
Lemma 4.3. Let $i \in \{1, \ldots, m-1\}$, let $\gamma > 0$, and let $D \in \mathcal{D}_{t, \gamma}$ be a distinguisher. Moreover, let $\Sigma_i, \ldots, \Sigma_{i-1}$ be valid states for $A_1, \ldots, A_{i-1}$, respectively, with $|\sigma_j| \leq l_j$ (for $j = 1, \ldots, i-1$), and let $b_{[i,i-1]} \in \{0, 1\}$ be a binary value. Also, assume that for a uniform bit $B$,

$$\text{Guess}^D(b_{[i,i-1]} \oplus B \mid C(A_{[i,i-1]}[\sigma_1, \ldots, \sigma_{i-1}])\langle S_{[i,m]}, I_{[i,m]} \rangle_B) > 2\gamma + 2\delta.$$  

Then, at least one of the following two statements is true:

(i) There exists a valid state $\sigma_i$ for $A_i$ with $|\sigma_i| \leq l_i$ and a bit $b_{[i]} \in \{0, 1\}$ such that

$$\text{Guess}^D(b_{[i,i]} \oplus B \mid C(A_{[i,i]}[\sigma_1, \ldots, \sigma_i])\langle S_{[i+1,m]}, I_{[i+1,m]} \rangle_B) > \delta;$$

(ii) There exists a distinguisher $D' \in \mathcal{D}_{t', q'_i}$ such that

$$\Delta^D'(S_i, I_i) \geq \text{Guess}^D'(B_i \mid \langle S_i, I_i \rangle_{B_i}) > \varepsilon,$$

and

$$t'_i = t_i + \varphi_{\text{isol}} \cdot \left[ t + tA(c)(q) + O \left( \sum_{j=1}^{i-1} tA_j(q_j, l_j) + \sum_{j=i+1}^{m} tA_j(q_j) \right) \right]$$

and $q'_i = \varphi_{\text{isol}} \cdot q_i$.

Proof: A straightforward calculation yields (for an additional uniform random bit $B'$ independent of $B$, and for $A := A_{[i,i-1]}[\sigma_1, \ldots, \sigma_{i-1}]$)

$$\text{Guess}^D(b_{[i,i-1]} \oplus B \mid C(A)\langle S_{[i,m]}, I_{[i,m]} \rangle_B) =$$

$$= 2 \cdot \text{Guess}^D(b_{[i,i-1]} \oplus B \mid C(A)\langle S_{[i,m]}, I_{[i,m]} \rangle_{B'}I_{[i,m]} \rangle_B). \quad (4.1)$$

Also, as $C(\cdot)$ is neutralizing for $S_1, \ldots, S_m$ and $I_1, \ldots, I_m$, it is also neutralizing for $\langle S_1, I_1 \rangle_{B_1}, \ldots, \langle S_{i-1}, I_{i-1} \rangle_{B_{i-1}}, S_i, \ldots, S_m$ and $I_1, \ldots, I_m$. In turn, because $\langle S_1, I_1 \rangle_{B_1}, \ldots, \langle S_{i-1}, I_{i-1} \rangle_{B_{i-1}}$ are cc-stateless systems, the construction $C(\cdot)$ is also neutralizing for $A_1[\sigma_1], \ldots, A_{i-1}[\sigma_{i-1}], S_i, \ldots, S_m$ and $I_1, \ldots, I_m$. This yields

$$C(A)\langle S_{[i,m]}, I_{[i+1,m]} \rangle \equiv C(A)\langle I_{[i,m]} \rangle S_{[i+1,m]} \rangle \equiv C(A)\langle I_{[i,m]} \rangle \equiv C(I_1, \ldots, I_m),$$

from which we infer the equivalence

$$\langle C(A)\langle S_{[i,m]}, I_{[i,m]} \rangle_{B'}I_{[i,m]} \rangle_B, b_{[i,i-1]} \oplus B \rangle \equiv$$

$$\equiv \langle C(A)\langle S_{[i,m]}, I_{[i,m]} \rangle_{B} \langle S_{[i+1,m]}, I_{[i+1,m]} \rangle_{B''}, b_{[i,i-1]} \oplus B \oplus B'' \rangle \quad (4.2)$$
for independent uniform random bits $B_i$ and $B''_i$, since both system-bit pairs behave as $C(I_1, \ldots, I_m)$ whenever the bit is $1 - b_{1,i-1}$, whereas otherwise they behave as $C(I_1, \ldots, I_m)$ or $C(A||S_{i,m})$ with probability $1/2$ each. Therefore, combining (4.1) and (4.2), the assumption of the lemma can equivalently be reformulated as

$$\text{Guess}^{DC}(b_{1,i-1} \oplus B_i \oplus B''_i| A||S_i, I_i||S_{i+1,m}, I_{i+1,m}|B') > \delta \epsilon + \tau,$$

We can now apply the Isolation Lemma (Lemma 3.16) to obtain the desired statement, and note that the running time to implement the system-bit pair $(S_{i+1,m}, I_{i+1,m})B', B')$ is of the order $\sum_{j=i+1}^{m} t_{A(S_j, I_j)B_j}(q_j)$ by construction of $A(S_j, I_j)B_j$. 

**From the Isolation Lemma to the Product Theorem.** For given $t$, $q$ and $\gamma$, and $\tau := \gamma/(m-1)$, we let $t'_1, \ldots, t'_m$ and $q'_1, \ldots, q'_m$ be as in the statement of Theorem 4.2, and define $\epsilon_i := \Delta_{i,q'_i}(F_i, I_i)$. Furthermore, recall that $\epsilon_i \leq \frac{\epsilon}{2}$ for all $i = 1, \ldots, m-1$. 

From now on, assume that Theorem 4.2 is false, and let $D \in \mathcal{D}_{t,q}$ be the distinguisher such that

$$\Delta^D(C(S_1, \ldots, S_m), C(I_1, \ldots, I_m)) = \text{Guess}^D(B||C(F_{1,m}, I_{1,m}|B)) \geq 2^{m-1} \cdot \epsilon_1 \cdots \epsilon_m + 2 \gamma$$

for a uniform random bit $B$. Define $\gamma_i := (m-1-i) \cdot \tau$ for $i = 0, \ldots, m-1$ and consider the statements $\mathcal{STAT}_i(\sigma_1, \ldots, \sigma_i, b)$ for a bit $b \in \{0, 1\}$ and valid states $\sigma_1, \ldots, \sigma_i$ of $A_1, \ldots, A_i$ respectively, with $|\sigma_j| \leq l_j$ for $j = 1, \ldots, i$ which holds if and only if

$$\text{Guess}^D(b \oplus B'| C(A|1,i)[\sigma_1, \ldots, \sigma_i||S_{i+1,m}, I_{i+1,m}|B']) \geq 2^{m-1-i} \cdot \epsilon_{i+1} \cdots \epsilon_m + \gamma_i$$

where we have used the facts that $\gamma_i := \gamma_{i-1} + \tau$ and $2 \cdot \epsilon_{i+1} \leq 1$. In particular, by our assumption $\mathcal{STAT}_0(0)$ holds. Furthermore, given $\mathcal{STAT}_i(\sigma_1, \ldots, \sigma_{i-1}, b)$ holds for some $i$ we apply the Isolation Lemma (Lemma 4.3). Note that condition (ii) cannot hold, as otherwise there exists a distinguisher $D'_i$ such that $\Delta^D_i(S_i, I_i) > \epsilon_i$, but this contradicts the assumed indistinguishability of $S_i$ and $I_i$, as we have defined $\epsilon_i$ with respect to the maximal running time of a constructed distinguisher $D'_i$. 


Therefore, condition (i) must hold: In other words, we have shown that for all $i = 0, \ldots, m - 2$

$$\text{STAT}_i(\sigma_1, \ldots, \sigma_i, b) \implies \exists \sigma_{i+1}, b': \text{STAT}_i(\sigma_1, \ldots, \sigma_{i+1}, b \oplus b'),$$

where all the states $\sigma_1, \sigma_2, \ldots$ are valid for their respective implementations and are such that $|\sigma_i| \leq l_i$. Iterating the argument we obtain that there exist $\sigma_1, \ldots, \sigma_{m-1}$ and a bit $b$ such that

$$\text{Guess}^D(b \oplus B_m | C(A_{[1,m-1]}[\sigma_1, \ldots, \sigma_{m-1}] || (S_m, I_m) B_m)) > \epsilon_m + \tilde{\gamma}_{m-1} = \epsilon_m.$$ 

Hence, the distinguisher

$$D' := D(C(A_{[1,m-1]}[\sigma_1, \ldots, \sigma_{m-1}] || \cdot)) \oplus b$$

obtains advantage larger than $\epsilon_m$, contradicting the assumed indistinguishability of $S_m$ and $I_m$.

### 4.3. Applications of the Product Theorem

#### 4.3.1. Sums of PRFs and Tightness

Let $F_1, \ldots, F_m : \mathcal{X} \to \mathcal{Y}$ be random functions, with respective algorithm implementations $A_{F_i}$, where all but $F_m$ are required to be cc-stateless, and let $\star$ be a quasi-group operation on $\mathcal{Y}$. The operator $\star$ is neutralizing, as discussed in Section 4.1, for $F_1, \ldots, F_m$ and ideal systems $I_1 = \cdots = I_m = R$, where $R : \mathcal{X} \to \mathcal{Y}$ is a URF. We assume that elements of $\mathcal{Y}$ can be encoded using $\ell \approx \log |\mathcal{Y}|$ bits. Recall that the canonical implementation of $R$ keeps a linearly-growing state of size $s = \mathcal{O}(q \cdot \ell)$ after $q$ queries, and answers each query in time $\mathcal{O}((\log(s))$. Therefore, with $t_{A_{F_i} R_i B_i}(q, s) = \mathcal{O}(\max\{t_{A_{F_i}}(q, s), q \cdot \log(s + q)\})$ and $l_i := \mathcal{O}(\max\{s_{A_{F_i}}(\varphi_\oplus q), \varphi_\oplus q \cdot \ell\})$, we apply Theorem 4.2 to obtain the following result (we tacitly assume that all advantages are bounded by $\frac{1}{2}$):

**Corollary 4.4.** For all $t, q > 0$ and $\gamma > 0$,

$$\Delta_{t,q}(F_1 \star \cdots \star F_m, R) \leq 2^{m-1} \cdot \prod_{i=1}^{m} \Delta_{t,q_i}^{\star i}(F_i, R) + 2\gamma.$$
4.3 Applications of the Product Theorem

We remark that the analogous result for PRGs follows as a special case, since a PRG can be seen as a one-input PRF.

A weaker version of this result was shown by Dodis et al. [DIJK09] for the special case $\ast = \oplus$: Their bounds depend in particular on the number of queries, and is hence far away from our bound, which we prove to be tight up to the additive term $\gamma$.

Remark 4.1 (Tightness). Consider the cc-stateless random function $F$ that behaves as a URF $R : \{0, 1\}^n \rightarrow \{0, 1\}^n$, with the exception that the first bit of $F(0^n)$ is one with probability $\frac{1}{2} + \varepsilon$.

Note that $\Delta_t(R, P) \leq \varepsilon$ for all $D$, as the function table of $F$ has statistical distance $\varepsilon$ from a randomly chosen function. However, on input $0^n$, the first output bit of the XOR $F_1 \oplus \cdots \oplus F_m$ is one with probability $\frac{1}{2} + 2^{m-1} \varepsilon^m$: Therefore, the constant-time distinguisher $D^*$ that returns the first output bit on input $0^n$ achieves advantage $2^{m-1} \varepsilon^m$.

Regardless of the security loss we are ready to accept, the bound in Corollary 4.4 is consequently tight up to the term $\gamma$, and Theorem 4.2 cannot be improved at this level of generality.

4.3.2. Cascade of PRPs

Let $P : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a URP and let $Q_1, \ldots, Q_m : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be cc-stateless random permutations, where $Q_i$ and $\langle Q_i \rangle$ are implemented by $A_{Q_i}$ and $A_{\langle Q_i \rangle}$, respectively. Recall that the $\triangleright$ operator is neutralizing for $Q_1, \ldots, Q_m$ (all with ideal system $P$), as well as for $\langle Q_1 \rangle, \ldots, \langle Q_m \rangle$ (all with ideal system $\langle P \rangle$). Recall that simulating the URP $P$ (as well as the two-sided URP $\langle P \rangle$) requires the same complexity as implementing a URF. Therefore, similarly to the PRF case above, we have $t_{A_{Q_i}, n, s} (q, s) = O(\max\{t_{A_{Q_i}}(q, s), q \cdot \log(s + q)\})$, as well as $l_i = O(\max\{s_{A_{Q_i}}(\varphi \cdot q), \varphi \cdot q \cdot n\})$ (and analogously for $\langle Q_i \rangle$), and Theorem 4.2 yields the following corollary:

Corollary 4.5. For all $t, q > 0$ and $\gamma > 0$,

$$\Delta_{t, q}(Q_1 \triangleright \cdots \triangleright Q_m, P) \leq 2^{m-1} \cdot \prod_{i=1}^{m} \Delta_{t_i, q_i}(Q_i, P) + 2\gamma,$$

This function is efficiently implementable by a stateful algorithm, whereas a keyed function approximating $F$ can be built from any PRF: Just add sufficiently many bits to the key to simulate the first output bit on input $0^n$, and behave otherwise as the original PRF.
and

\[
\Delta_{t,q}(\langle Q_1 \rangle \triangleright \cdots \triangleright \langle Q_m \rangle, \langle P \rangle) \leq 2^{m-1} \cdot \prod_{i=1}^{m} \Delta_{t_i,q_i}(\langle Q_i \rangle, \langle P \rangle) + 2\gamma.
\]

Furthermore, we note that \(Q_1\) is allowed to be stateful in the one-sided case, as Theorem 4.2 allows one system to be stateful: In fact, \(\triangleright\) is not necessarily neutralizing whenever at least two permutations are stateful.

At time of publication (in [MT09]), this was the first result considering two-sided PRPs, and even in the one-sided setting only the case \(m = 2\) was considered by Luby and Rackoff [LR86], and subsequently extended to any constant \(m\) by Myers [Mye99]. (Also, a slightly weaker result for the case \(m = O(\log \log n)\) is given in [Mye99].) However, all of these results fall short of achieving security amplification, as they are not able to transform an \(\varepsilon\)-PRP into a fully secure PRP. However, contrary to the PRF case above, in this case the bound is not tight, and indeed we will show in Chapter 5 how it can be improved to achieve amplification well beyond advantage \(\frac{1}{2}\).

### 4.4. A Strong Product Theorem

Since Theorem 4.2 holds for arbitrary neutralizing constructions, one cannot avoid the factor \(2^{m-1}\) in the bound by Remark 4.1. This section shows a strong product theorem for subclass of neutralizing constructions satisfying a simple information-theoretic property, i.e., the obtained upper bound is roughly the product of the individual advantages. Contrary to Theorem 4.2, this result is not a computational analogue of an existing information-theoretic statement: Instead, we take a proof-centric approach, identifying properties that need to be satisfied by neutralizing construction \(C(\cdot)\) in order to obtain a strong product theorem.

#### 4.4.1. Self-Independence

We start with some intuition. For a neutralizing construction \(C(\cdot)\) for \(cc\)-stateless systems \(S_1, \ldots, S_m\) and \(I_1, \ldots, I_m\) with \(\Delta_{t_i,q_i}(S_i, I_i) \leq \varepsilon_i\), we want to prove that

\[
\Delta_{t,q}(C(S_1, \ldots, S_m), I) \leq \prod_{i=1}^{m} \varepsilon_i + \gamma,
\]

...
where $I := C(I_1, \ldots, I_m)$. One possible proof approach, that we will follow, makes use of the isolation technique, as in the proof of Theorem 3.15: By a counting argument, it is not hard to show that given initial states $s_1, \ldots, s_{i-1}$ for $S_1, \ldots, S_{i-1}$ as well as a distinguisher $D$ such that

$$
\Delta^D(C(S_1(s_1), \ldots, S_{i-1}(s_{i-1}), S_i, \ldots, S_m), I) > \prod_{j=i}^{m} \varepsilon_j + \gamma
$$

then either there exists $s_i$ for $S_i$ such that

$$
\Delta^D(C(S_1(s_1), \ldots, S_{i-1}(s_{i-1}), S_i, \ldots, S_m), I) > \prod_{j=i+1}^{m} \varepsilon_j + \gamma,
$$

or, for a fraction $\varepsilon_i$ of the $s_i$’s,

$$
\Delta^D(C(S_1(s_1), \ldots, S_i(s_i), S_{i+1}, \ldots, S_m), I) = \Delta^D C_i(S_i(s_i), I_i) > \gamma
$$

where (recall that $C(\cdot)$ is neutralizing)

$$
C_i(\cdot) := C(S_1(s_1), \ldots, S_{i-1}(s_{i-1}), \cdot, S_{i+1}, \ldots, S_m).
$$

In the latter case, we would like to build a distinguisher $D'_i$ out of $D_i := D C_i$ which contradicts indistinguishability of $S_i$ and $I_i$. The simplest strategy is to let independent instances of $D_i$ sequentially interact with the given system $T$, resulting in output bits $b_1, b_2, \ldots$, where either $T = S_i(s_i)$ for some state $s_i$ or $T = I_i$, producing an estimate $\overline{p}$ of the probability $P[D_i(T) = 1]$ as the average of the bits $b_1, b_2, \ldots$. If $\overline{p}$ is sufficiently far from $P[D_i(I_i) = 1]$ (say at least $\gamma/2$), then $D'_i$ outputs 1, otherwise it outputs 0. It is not hard to verify that if the estimate is within an error of at most $\gamma/2$ from the actual probability, $D'_i$ achieves advantage larger than $\varepsilon_i$.

Note that $T = S_i(s_i)$ is stateless, and hence the bits $b_1, b_2, \ldots$ are independent when interacting with $T$: By Hoeffding’s inequality (Lemma 2.1), $\overline{p}$ is indeed a good estimate of $P[D_i(S_i(s_i)) = 1]$. However, in the case where $T = I_i$, the $b_i$’s are not necessarily independent, and hence no statement about the quality of $\overline{p}$ can be made in general.

We introduce a condition which is sufficient for (almost) independence to be achieved, which we refer to as self-independence of an ideal

---

\footnote{This is because it outputs 1 on the $\varepsilon_i$-fraction of the $s_i$’s considered above, and outputs 0 when interacting with $I_i$.}
system $I$ under a construction $C(\cdot)$: It captures the property that a computationally unbounded distinguisher is unable to tell apart the scenario where the same instance of $I$ is accessed through independent instances of $C(\cdot)$ from the setting where each instance of $C(\cdot)$ accesses an independent instance of $I$. Should $C_i(\cdot)$ satisfy this property for $I_i$ above, then it would be easy to show that a good estimate is produced also in the case $T = I_i$.

**Definition 4.2.** The system $I$ is $\eta$-self-independent under $C(\cdot)$ for a function $\eta : \mathbb{N} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$, if for all $q, \lambda > 0$, the best (information-theoretic) distinguishing advantage when allowing $q$ queries to each subsystem satisfies

$$\Delta_{q,\ldots,q}(C_1(I)||\ldots||C_\lambda(I), C_1(I_1)||\ldots||C_\lambda(I_\lambda)) \leq \eta(q, \lambda),$$

where $C_1(\cdot), \ldots, C_\lambda(\cdot)$ and $I_1, \ldots, I_\lambda$ are independent copies of $C(\cdot)$ and $I$, respectively.

As an example, consider the construction $C(\cdot)$ which generates a (secret) random $n$-bit offset $Z$, and given access to a random function $F : \{0, 1\}^n \to \{0, 1\}^n$, $C(F)$ returns $F(x \oplus Z)$ upon each query $x$. It is not hard to show that a URF $R : \{0, 1\}^n \to \{0, 1\}^n$ is $\eta$-self-independent under $C(\cdot)$ for $\eta(q, \lambda) \leq 2^{\lambda^2} \cdot 2^{-n}$, i.e., the probability that for some distinct $i \neq j$ the instances $C_i(\cdot)$ and $C_j(\cdot)$ invoke $R$ with the same input.

### 4.4.2. A Product Theorem from Self-Independence

In the following, let $C(\cdot)$ be a neutralizing construction for the real systems $S_1, \ldots, S_m$ and ideal system $I_1, \ldots, I_m$, all of which (with the possible exception of $S_m$ and $I_m$) are cc-stateless. Furthermore, we assume that $S_i(\cdot)$ is efficiently implementable for all $i = 1, \ldots, m-1$, and the corresponding (short) random variable $S_i$ is drawn from the set $S_i$. Also, we let $E(\cdot)$ be a construction restricting access to $S_i$ and $I_i$, in the sense of Section 2.6.3. Finally, for $i = 1, \ldots, m$, and for $s_1 \in S_1, \ldots, s_{i-1} \in S_{i-1}$ we define

$$C_{s_1, \ldots, s_{i-1}}(\cdot) := C(S_1(s_1), \ldots, S_{i-1}(s_{i-1}), \cdot, S_{i+1}, \ldots, S_m)$$

and consider the following two properties:

---

5While the same techniques as in the proof of Theorem 3.15 could be used to address general cc-stateless systems where $S_i(\cdot)$ is not necessarily efficient, this will not be necessary for our applications: In particular, for the application for Theorem 4.2, we always need to consider systems with no such efficient implementations, such as URFs and URPs, but this is not needed.
4.4 A Strong Product Theorem

(i) For all $i = 1, \ldots, m - 1$ (the property is not necessary for $i = m$) and all $s_1 \in S_1, \ldots, s_{i-1} \in S_{i-1}$, the ideal system $I_i$ is $\eta$-self-independent under the construction $C_{s_1, \ldots, s_{i-1}}^{(i)}(\cdot)$ for some small function $\eta$.

(ii) For all $i = 1, \ldots, m$ and $s_1 \in S_1, \ldots, s_{i-1} \in S_{i-1}$, there exists a construction $T_{s_1, \ldots, s_{i-1}}^{(i)}(\cdot)$ with the property that for independent instances $T_1(\cdot), \ldots, T_\lambda(\cdot)$ and $C_1(\cdot), \ldots, C_\lambda(\cdot)$ of $T_{s_1, \ldots, s_{i-1}}^{(i)}(\cdot)$ and $C_{s_1, \ldots, s_{i-1}}^{(i)}(\cdot)$, respectively, and all compatible systems $S$,

$$T_1(E(S)) \| \cdots \| T_\lambda(E(S)) \equiv C_1(S) \| \cdots \| C_\lambda(S).$$

We define $t_{T_i}$ as the maximal complexity (taken over all $s_1, \ldots, s_{i-1}$) for implementing the construction $T_{s_1, \ldots, s_{i-1}}^{(i)}(\cdot)$.

In the following, we define $\lambda := \left(\frac{4m}{\gamma}\right)^2 \ln \left(\frac{4m}{\gamma}\right)$, for understood $m$ and $\gamma$.

**Theorem 4.6.** Let $q > 0$, $C(\cdot)$, and $E(\cdot)$ be as above satisfying conditions (i) and (ii), and assume that upon $q$ queries, $C(\cdot)$ makes at most $q_i$ queries to the $i$-th subsystem. Then, for all $t > 0$ and $\gamma > 0$,

$$\Delta_{t,q}(C(S_1, \ldots, S_m), C(I_1, \ldots, I_m)) \leq \prod_{i=1}^{m} \Delta_{t', q'_i}(E(S_i), E(I_i)) + \sum_{i=1}^{m-1} \eta(q_i, \lambda) + \gamma,$$

where

$$t'_i := \lambda \cdot (t + O(t_{T_i}(q_i))) \quad \text{and} \quad q'_i := \lambda \cdot q_i$$

for all $i = 1, \ldots, m - 1$, whereas $t'_m := t + O(t_{T_m}(q))$ and $q'_m := q_m$.

The full proof of Theorem 4.6 is deferred to Section 4.4.3. Furthermore, in Section 4.5 we provide application examples of Theorem 4.6.

### 4.4.3. Proof of the Strong Product Theorem

This section presents a proof of Theorem 4.6. As in the cases of Theorems 3.15 and 4.2, the proof is based on the isolation technique. It abstracts and generalizes the proof technique used by Myers [Mye03].
(which was in turn based on Levin’s proof of the XOR-lemma [Lev87, GNW95]). Throughout this section, we let

\[ S_{[i,j]} := S_i, \ldots, S_j, \]

\[ S_{[i,j]}(s_i, \ldots, s_j) := S_i(s_i), \ldots, S_j(s_j), \]

\[ I_{[i,j]} := I_i, \ldots, I_j. \]

**The Isolation Lemma.** In the following, we let \( I := C(I_1, \ldots, I_m) \) and define \( \lambda = \lambda(m, \gamma) := \left( \frac{\gamma}{4} \right)^2 \cdot \ln \left( \frac{\gamma}{\delta} \right) \). Also, in the following, let \( \lambda \) be defined as above with respect to some understood \( \gamma \), whereas with respect to \( \gamma \) we define \( \lambda := \left( \frac{\gamma}{4} \right)^2 \cdot \ln \left( \frac{\gamma}{\delta} \right) \).

**Lemma 4.7.** Let \( i \in \{1, \ldots, m - 1\}, \zeta > 0, \) and \( \tau > 0 \). Assume that \( D \) makes \( q \) queries to \( C(\cdot) \) resulting in \( q_j \) queries to each of the \( m \) subsystems \( S_j \), and that for values \( s_1 \in S_1, \ldots, s_{i-1} \in S_{i-1} \) we have

\[ P[D(C(S_{[1,i-1]}(s_1, \ldots, s_{i-1}), I_{[i,m]})) = 1] - P[D(I) = 1] > \delta \cdot \epsilon + \zeta + \tau, \]

then at least one of the two following statements holds:

(i) There exists a value \( s_i \in S_i \) such that

\[ P[D(C(S_{[1,i]}(s_1, \ldots, s_i), S_{[i+1,m]})) = 1] - P[D(I) = 1] > \delta + \zeta + \tau. \]

(ii) There exists a distinguisher \( D'_i \) such that \( \Delta_{D'_i}(E(S_i), E(I_i)) > \epsilon \) which makes \( q'_i := \lambda \cdot q_i \) queries and has time complexity \( t'_i := \lambda \cdot (t + t_T(q_i)) \).

**From the Isolation Lemma to Theorem 4.6.** Similarly to the cases of Theorems 3.15 and 4.2, we apply the isolation lemma iteratively to obtain Theorem 4.6. Fix \( t, q, \gamma > 0 \) (and let \( \tau := \gamma/(m - 1) \)) such that upon \( q \) queries \( C(\cdot) \) makes \( q_1, \ldots, q_m \) queries to its subsystems. For all \( i = 1, \ldots, m \), let \( t'_i, q'_i \) as in the statement of Theorem 4.6, and let \( \epsilon_i := \Delta_{t'_i,q'_i}(S_i, B_i) \).

Assume towards a contradiction that there exists a distinguisher \( D \) with running time \( t \) making \( q \) queries to \( C \) such that

\[ \Delta_D(C(S_{[1,m]}), I) \geq P[D(C(S_{[1,m]})) = 1] - P[D(I) = 1] \]

\[ \geq \epsilon_1 \cdots \epsilon_m + \sum_{i=1}^{m-1} \eta(q_i, \lambda) + \gamma. \]
and define $\gamma_i := (m - 1 - i) \cdot \eta$ for $i = 0, \ldots, m - 1$ and consider the statements $\text{STAT}_i(s_1, \ldots, s_i)$ which holds if and only if

\[
P[D(C(S_{[1,i]}(s_1, \ldots, s_i), S_{[i+1,m]}))] = 1] - P[D(I) = 1] \geq \\
\geq \varepsilon_{i+1} \cdot \varepsilon_m + \sum_{j=i+1}^{m-1} \eta(q_j, \lambda) + \gamma_i \\
\geq \varepsilon_{i+1} \cdot (\varepsilon_{i+2} \cdot \varepsilon_m + \gamma_{i+1}) + \sum_{j=i+1}^{m-1} \eta(q_j, \lambda) + \eta.
\]

In particular, by our assumption $\text{STAT}_0$ holds. By the Isolation Lemma (Lemma 4.7) and the fact that $\Delta_{i, s_i}'(s_{i+1}, I_{i+1}) \leq \varepsilon_{i+1}$, whenever the condition $\text{STAT}_i(s_1, \ldots, s_i)$ holds for some $s_1, \ldots, s_i$, then condition (ii) cannot hold, and hence condition (i) must hold, i.e., there must exists $s_{i+1}$ such that $\text{STAT}_{i+1}(s_1, \ldots, s_{i+1})$ holds. In other words, we have shown that for all $i = 0, \ldots, m - 2$

\[
\text{STAT}_i(s_1, \ldots, s_i) \implies \exists s_{i+1} : \text{STAT}_{i+1}(s_1, \ldots, s_{i+1}).
\]

Iterating the argument we obtain that there exist $s_1, \ldots, s_{m-1}$ such that $\text{STAT}_{m-1}(s_1, \ldots, s_{m-1})$ holds, i.e.

\[
P[D(C(S_{[1,m-1]}(s_1, \ldots, s_{m-1}), S_{m-1})] = 1] - P[D(I) = 1] \geq \varepsilon_m.
\]

This, however, gives rise to a distinguisher $D'_m := D(T_{m-1,s_1,\ldots,s_{m-1}})$ with advantage $\varepsilon_m$ and the given complexity $t'_m$, contradicting the indistinguishability assumption on $S_m$ and $I_m$.

**Proof of the Isolation Lemma (Lemma 4.7).** We fix $i \in \{1, \ldots, m\}$ as in the statement of the lemma. It is convenient to denote $\eta := \eta(q_i, \lambda)$, as well as $C^{(i)}(\cdot) := C_{s_1,\ldots,s_{i-1}}^{(i)}(\cdot)$. Note that since $C(\cdot)$ is neutralizing, we have $C^{(i)}(I_i) \equiv I$. Furthermore, we also define the function $\alpha : S_i \rightarrow [-1, 1]$ such that

\[
\alpha(s) := P[D(C^{(i)}(S(s)))] = 1) - P[D(I) = 1].
\]

Clearly, by the assumption of the lemma, we have $E[\alpha(S_i)] > \delta \cdot \varepsilon + \zeta + \eta$. Furthermore, if there existed an $s_i$ such that $\alpha(s_i) > \delta + \zeta - \eta$, the first statement of the lemma would be directly true. Therefore, we assume in the following that $\alpha(s) \leq \delta + \zeta - \eta$ for all $s \in S_i$. Moreover, we define the set

\[
\mathcal{G} := \{s \in S_i \mid \alpha(s) > \eta/2\}.
\]
The following claim gives a lower bound on the probability that $S_i$ is in the set $G$.

**Claim 5.** $P[S_i \in G] > \varepsilon + \eta + \gamma/2$.

**Proof.** Assume, towards a contradiction, that $P[S_i \in G] \leq \varepsilon + \gamma/2 + \eta$. Then,

\[
E[\alpha(S_i)] = P[S_i \in G] \cdot E[\alpha(S_i)|S_i \in G] + P[S_i \notin G] \cdot E[\alpha(S_i)|S_i \notin G]\]

\[
\leq (\varepsilon + \eta + \gamma/2) \cdot (\delta + \zeta - \eta) + \gamma/2 \]

\[
\leq \varepsilon \delta + \zeta - \eta + \eta + \gamma/2 + \gamma/2 = \varepsilon \delta + \zeta + \gamma.
\]

which is in contradiction with the assumption on $D$. \(\Box\)

We construct the distinguisher $D'_i$ for distinguishing $E(S_i)$ from $E(I_i)$ as specified on the left-hand-side of Figure 4.1: With $\pi := P[D(I) = 1]$, $D'_i$ simulates $\lambda$ independent instances of $D(T_{s_1, \ldots, s_{i-1}}(\cdot))$ sequentially interacting with the given system $S$, and computes the average $\overline{\sigma}$ of the $\lambda$ bits output by these instances, and finally outputs $1$ if $\overline{\sigma} > \pi/4$, and $0$ otherwise. Note that by the property of $T_{s_1, \ldots, s_{i-1}}(\cdot)$ we can equivalently consider a distinguisher $D''_i$ for $S_i$ and $I_i$ (depicted on the right-hand-side of Figure 4.1) which computes the bits $o_1, \ldots, o_\lambda$ by letting independent instances of $D(C^{(i)}(\cdot))$ interact with the given system. Clearly, $\Delta^{D'_i}(E(S_i), E(I_i)) = \Delta^{D''_i}(S_i, I_i)$, and we thus focus on the latter advantage.

The remainder of the proof consists of the following two lemmas.
4.4 A Strong Product Theorem

Lemma 4.8. \( P[D_i''(S_i(s)) = 1] > 1 - e^{-\lambda(\tau/4)^2} \) for all \( s \in G \).

Lemma 4.9. \( P[D_i''(I_i) = 1] \leq e^{-\lambda(\tau/4)^2} + \eta(q_i, \lambda) \).

Before we turn to the proofs of the two lemmas, we note that they suffice to obtain the isolation lemma, since by Claim 5

\[
\Delta D_i''(S_i, I_i) \geq P[S_i \in G] \cdot P[D_i''(S_i) = 1|S_i \in G] - P[D_i''(I_i) = 1]
\]

\[
> (\varepsilon + \gamma/2 + \eta)(1 - e^{-\lambda(\tau/4)^2}) - e^{-\lambda(\tau/4)^2} - \eta
\]

\[
\geq \varepsilon + \gamma/2 + \eta - 2e^{-\lambda(\tau/4)^2} - \eta = \varepsilon + \gamma/2 - 2e^{-\lambda(\tau/4)^2} = \varepsilon
\]

for the chosen value of \( \lambda \).

Proof of Lemma 4.8. Since the system \( S_i(s) \) is stateless, note that for a fixed \( s \in G \) the random variables \( o_1, \ldots, o_\lambda \) are independent binary variables with

\[
p(s) := P[o_j = 1] = P[D(C^{(i)}(S_i(s))) = 1],
\]

and thus \( E[\sigma] = p(s) \). We know that \( p(s) - \pi > \gamma/2 \), since \( s \in G \), and thus \( \sigma - \pi \leq \gamma/4 \) implies that \( \sigma < p(s) - \gamma/4 \). Therefore, by Hoeffding’s bound, we have

\[
P[D_i''(S_i(s)) = 0] = P[\sigma - \pi \leq \gamma/4] \leq P[\sigma < p(s) - \gamma/4] < e^{-\lambda(\tau/4)^2}. \quad \square
\]

Proof of Lemma 4.9. Consider a distinguisher \( D_i'' \) which given access to the parallel composition \( S_1 \parallel \ldots \parallel S_\lambda \) of \( \lambda \) systems \( S_1, \ldots, S_\lambda \), computes \( o_j := D(S_j) \) for all \( j = 1, \ldots, \lambda \), and then outputs its decision bit as \( D_i'' \).

Clearly, for independent instances \( C_1^{(i)}(\cdot), \ldots, C_\lambda^{(i)}(\cdot) \) of \( C^{(i)}(\cdot) \) we have

\[
P[D_i''(I_i) = 1] = P[D_i''(C_1^{(i)}(I_i)) \ldots C_\lambda^{(i)}(I_i)) = 1],
\]

i.e., where every system in the parallel composition accesses the same instance \( I_i \). Note that because of the \( \eta \)-self-independence of \( I_i \) under \( C^{(i)} \) we have, for independent instances \( I_{i,1}, \ldots, I_{i,\lambda} \) of \( I_i \),

\[
\Delta D_i''(C_1^{(i)}(I_i)) \ldots C_\lambda^{(i)}(I_i), C_1^{(i)}(I_{i,1}) \ldots C_\lambda^{(i)}(I_{i,\lambda})) \leq \eta(q_i, \lambda),
\]

from which we directly infer

\[
P[D_i''(I_i) = 1] \leq \eta(q_i, \lambda) + P[D_i''(C_1^{(i)}(I_{i,1})) \ldots C_\lambda^{(i)}(I_{i,\lambda})) = 1]. \quad (4.3)
\]
We hence upper bound the probability on the right-hand side. Note that, in this case, the variables $o_1, \ldots, o_\lambda$ are assigned independent equally-distributed values, with $P[o_j = 1] = \pi$ all $j = 1, \ldots, \lambda$ (by the neutralizing property of $C(\cdot)$), and thus $E[\pi] = \pi$ holds as well. Then, by Hoeffding’s inequality,

$$P[D''_i(C^{(i)}(I_{i,\lambda})) = 1] = P[\sigma - \pi > \gamma/4] < e^{-\lambda(\gamma/4)^2},$$

which combined with (4.3) yields the desired upper bound. □

4.5. Applications of the Strong Product Theorem

This section presents some applications of Theorem 4.6. Throughout this section, let $Q_1, \ldots, Q_m : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be cc-stateless random permutations, and let $F_1, \ldots, F_m : \{0, 1\}^n \rightarrow \{0, 1\}^\ell$ be cc-stateless random functions. In particular, we let $Q_i \equiv Q_i(S_i)$ for some random variable $S_i$ over a set $S_i$, for all $i = 1, \ldots, m$, and analogously we have $F_i \equiv F_i(S_i)$.

Furthermore, let $P : \{0, 1\}^n \rightarrow \{0, 1\}^n$ and $R : \{0, 1\}^n \rightarrow \{0, 1\}^\ell$ be a URP and URF, respectively. Assume that $Q_i(s)(x)$ (and $Q_i^{-1}(s)(y)$) and $F_i(s)(x)$ can be computed in time $t_Q$ and $t_F$, respectively, for all $s, x,$ and $y$.

4.5.1. Randomized Cascade of PRPs

Our first application is a strong product theorem for (two-sided) PRPs. We modify the (two-sided) cascade $\langle Q_1 \rangle \triangleright \cdots \triangleright \langle Q_m \rangle$ by choosing two independent random offsets that are added to the inputs and the outputs, i.e., we consider $\langle \oplus_{Z_1} \rangle \triangleright \langle Q_1 \rangle \triangleright \cdots \triangleright \langle Q_m \rangle \triangleright \langle \oplus_{Z_2} \rangle$ for two independent uniform $n$-bit strings $Z_1, Z_2$, where for some $z \in \{0, 1\}^n$ the system $\langle \oplus_{z} \rangle$ is the two-sided mapping which answers a forward query $(x, +)$ with $x \oplus z$ and a backward query $(y, -)$ with $y \oplus z$. The computational overhead is minimal compared to the regular cascade, and requires only additional storage for two $n$-bit strings (which are to be seen as part of the secret key). 6 Clearly the neutralizing property of the original cascade is preserved.

The central observation is that the two additional random offsets ensure property (i) and allow for applying Theorem 4.6. More formally, for some $i \in \{1, \ldots, m\}$ and $s_1 \in S_1, \ldots, s_{i-1} \in S_{i-1}$ we define (for a

---

6The idea of adding random offsets at both ends of a permutation was already used by Even and Mansour [EM97], though in a completely different context.
system $S$ implementing a two-sided permutation from $n$ bits to $n$ bits)
the construction $N(\cdot)$ as choosing two offsets $Z_1, Z_2$ independently and
uniformly at random, as well as $S_{i+1}, \ldots, S_m$, and such that
\[
N(S) := \langle \oplus_{Z_1} \rangle \triangleright \langle Q_1(s_1) \rangle \triangleright \cdots \triangleright \langle Q_{i-1}(s_{i-1}) \rangle \triangleright S \triangleright \langle Q_i(S_{i+1}) \rangle \triangleright \cdots \triangleright \langle Q_m(S_m) \rangle \triangleright \langle \oplus_{Z_2} \rangle.
\]
The following lemma states the self-independence of a (two-sided) URP
\langle P \rangle$ under $N(\cdot)$.

**Lemma 4.10.** Let $N_1(\cdot), \ldots, N_\lambda(\cdot)$ be independent instances of $N$, and let
$P_1, \ldots, P_\lambda, P : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be independent URPs. Then
\[
\Delta_{q, \ldots, q}(N_1(\langle P \rangle)) \| \cdots \| N_\lambda(\langle P \rangle)) \| N_1(\langle P_1 \rangle)) \cdots \| N_\lambda(\langle P_\lambda \rangle)) \leq \lambda^2 \cdot q^2 \cdot 2^{-n}.
\]
In other words, $\langle P \rangle$ is $\eta$-self-independent under $N(\cdot)$ for $\eta(q, \lambda) := \lambda^2 q^2 2^{-n}$.

**Proof.** For notational simplicity, we define two systems $\hat{H}_0$ and $\hat{H}_1$ with
MBOs such that
\[
\hat{H}_0 := N_1(\langle P \rangle) \| \cdots \| N_\lambda(\langle P \rangle) \quad \text{and} \quad \hat{H}_1 := N_1(\langle P_1 \rangle) \| \cdots \| N_\lambda(\langle P_\lambda \rangle).
\]
For both systems, all $t = 1, \ldots, \lambda$, and all $i \geq 1$, let $\lambda_i^{(t)}$ contain those $n$-bit
strings $x$ for which, when processing the first $i$ queries, $N_i(\cdot)$ has issued
a forward query $(x, +)$, or alternatively a backward query $(y, -)$ of $N_i(\cdot)$
was answered with the value $x$ by the given permutation. Analogously,
let $\gamma_i^{(t)}$ contain those $n$-bit strings $y$ for which when processing the first $i$
queries $N_i(\cdot)$ has issued a backward query $(y, -)$, or alternatively a forward
query $(x, +)$ of $N_i(\cdot)$ was answered by $y$.

For both systems $\hat{H}_0$ and $\hat{H}_1$, the respective MBOs $A_1, A_2, \ldots$ and
$B_1, B_2, \ldots$ are such that $A_t = 1$ (or $B_t = 1$) if for some $t \neq t'$ we have
$\lambda_i^{(t)} \cap \lambda_i^{(t')} \neq \emptyset$ or $\gamma_i^{(t)} \cap \gamma_i^{(t')} \neq \emptyset$.

We note that for all $i \geq 1$,
\[
p_{\hat{H}_0, Y | X^i Y^{i-1} A_i = 0} = p_{\hat{H}_1, Y | X^i Y^{i-1} B_i = 0},
\]
because for both systems, if the MBO is 0, the query to $\langle P \rangle$ / $\langle P_t \rangle$
associated with a (fresh) forward query to $N_t(\langle P \rangle)$ / $N_t(\langle P_t \rangle)$ returns a
uniform value from $\{0, 1\}^n \setminus \bigcup_{t'} Y_t^{(t')}$. (The case of a backward query is
symmetric.) Furthermore,
\[
p_{\hat{H}_0, A_i = 0 | X^i Y^{i-1} A_i = 0} \geq p_{\hat{H}_1, B_i = 0 | X^i Y^{i-1} B_i = 0},
\]
because, for system $\hat{H}_0$, the condition $\lambda_i^{(t)} \cap \lambda_i^{(t')} = \emptyset$ holds for all $t \neq t'$ if and only if $\mathcal{Y}_i^{(t)} \cap \mathcal{Y}_i^{(t')} = \emptyset$ for all $t \neq t'$. Therefore, by Lemma 2.6, we obtain
\[
\Delta^D(\hat{H}_0^-, \hat{H}_1^-) \leq \nu^D(\hat{H}_1)
\]
for any distinguisher $D$ issuing at most $q$ queries to each subsystem and thus making totally at most $\lambda \cdot q$ queries.

We thus focus on upper bounding the probability $\nu^D(\hat{H}_1)$ that $D$ provokes $B_{\lambda q} = 1$. More specifically, let $Z_1^{(t)}, Z_2^{(t)}, S_1^{(t)}, \ldots, S_m^{(t)}$ be the values taken by $Z_1, Z_2, S_{i+1}, \ldots, S_m$ in $N_i(\cdot)$ for $t = 1, \ldots, \lambda$. Note that for all values $z_1^{(t)}, z_2^{(t)}$ and $s_i^{(t)}, \ldots, s_m^{(t)}$ taken by these variables, the system $\hat{H}_1$ behaves as the parallel composition of $\lambda$ independent (two-sided) random permutations. We can thus apply Lemma 2.9. We assume that we are given input-output pairs $(u_1^{(t)}, v_1^{(t)}), \ldots, (u_q^{(t)}, v_q^{(t)})$ for $N_i((P_i))$ for $t = 1, \ldots, \lambda$, i.e., the $i$-th query to $N_i((P_i))$ has been either a forward query $u_i^{(t)}$ with output $v_i^{(t)}$ or a backward query $v_i^{(t)}$ with output $u_i^{(t)}$. Moreover, we define for all $i = 1, \ldots, q$ and $t = 1, \ldots, \lambda$
\[
P_i^{(t)} := (Q_1(s_1) \triangleright \cdots \triangleright Q_{i-1}(s_{i-1}))(u_i^{(t)} \oplus Z_1^{(t)})
\]
\[
Q_i^{(t)} := (Q_m^{-1}(s_m^{(t)}) \triangleright \cdots \triangleright Q_{i+1}^{-1}(s_{i+1}^{(t)}))(v_i^{(t)} \oplus Z_2^{(t)}),
\]
\[i.e., these are the corresponding input-output pairs for the underlying permutations $(P_1), \ldots, (P_\lambda)$. Note that all of these random variables are (individually) uniformly distributed and furthermore, given $t \neq t'$, $i, j$ we have $P_i^{(t)} = P_i^{(t')}$ and $Q_i^{(t)} = Q_i^{(t')} = 2^{-n}$. Therefore, $B_{\lambda q} = 1$ implies that there exist distinct $t, t'$ and (not necessarily distinct) $i, j$ such that $P_i^{(t)} = P_j^{(t')}$ or $Q_i^{(t)} = Q_j^{(t')}$. By the union bound we conclude that
\[
\nu^D(\hat{H}_1) \leq 2 \cdot \left(\frac{\lambda}{2}\right) \cdot q^2 \cdot 2^{-n} \leq \lambda^2 \cdot q^2 \cdot 2^{-n}.
\]
\[\square\]

Therefore, we can combine Theorem 4.6 (with $E(\cdot)$ being the identity) and Lemma 4.10 to obtain the following result.

**Corollary 4.11.** For all $t, q, \gamma > 0$, and independent uniform $n$-bit strings $Z_1, Z_2$, 
\[
\Delta_{t, q}(\oplus_{Z_1} \triangleright Q_1 \triangleright \cdots \triangleright Q_m \triangleright \oplus_{Z_2}, (P)) \leq \prod_{i=1}^m \Delta_{t, q_i}(Q_i, (P)) + \frac{mq^2\lambda^2}{2^n} + \gamma.
\]
where \( t'_i := \lambda \cdot \left( t + O(q \cdot \sum_{j \neq i} t_{Q_j}) \right) \) and \( q'_i := \lambda \cdot q \) for all \( i = 1, \ldots, m - 1 \), whereas \( t'_m := t + O(q \cdot \sum_{j=1}^{m-1} t_{Q_j}) \) and \( q'_m := q \).

The result can be used to obtain an \((\varepsilon^m + \nu)\)-two-sided PRP from any \(\varepsilon\)-two-sided PRP, for a negligible function \(\nu\). (Note that the \(\eta\)-dependent term is negligible for polynomial \(t, q\) and any \(\gamma\) which is the inverse of a polynomial.) It can be shown that the second random offset \(Z_2\) is superfluous in the one-sided case.

### 4.5.2. Sum of Random-Input PRFs

The construction \(K(F_1 \oplus \cdots \oplus F_m)\) (i.e. the XOR of the functions accessed in a random-input attack) is clearly neutralizing (the ideal system being \(K(R)\)). However, the fact that \(F_1 \oplus \cdots \oplus F_m\) and \(R\) are invoked on random inputs only allows for proving a much stronger result using Theorem 4.6, since it satisfies property (i) needed by the theorem.

More precisely, let us fix \(i \in \{1, \ldots, m\}\) and \(s_1 \in S_1, \ldots, s_i-1 \in S_{i-1}\), and define for a random function \(S : \{0, 1\}^n \rightarrow \{0, 1\}^\ell\) the construction \(W(\cdot)\) as

\[
W(S) := K(F_1(s_1) \oplus \cdots \oplus F_{i-1}(s_{i-1}) \oplus S \oplus F_{i+1} \oplus \cdots \oplus F_m).
\]

Then, we can show the following.

**Lemma 4.12.** Let \(R, R_1, \ldots, R_\lambda : \{0, 1\}^n \rightarrow \{0, 1\}^\ell\) be independent URFs, and let \(W_1(\cdot), \ldots, W_\lambda(\cdot)\) be independent instances of \(W(\cdot)\), then

\[
\Delta_{q,\ldots,q}(W_1(R)) \cdots \|W_\lambda(R), W_1(R_1)) \cdots \|W_\lambda(R_\lambda)) \leq \frac{q^2\lambda^2}{2}2^{-n}.
\]

In other words, the URF \(R\) is \(\eta\)-self-independent under \(W\) for \(\eta(q, \lambda) \leq \frac{q^2\lambda^2}{2} \cdot 2^{-n}\).

**Proof.** Consider the system \(\hat{H}_0\) with an MBO \(A_1, A_2\ldots\) such that

\[
\hat{H}_0 := W_1(R) \cdots \|W_\lambda(R).
\]

Each query \(i\) to the subsystem \(t\) is associated with a random input \(r_{i,t}\) at which the XOR is evaluated, and let \(A_i := 0\) if within the first \(i\) queries there exists no two random inputs \(r_{j,t} = r_{k,t'}\) for distinct \(t \neq t'\). Moreover, let \(H_1 := W_1(R_1) \cdots \|W_\lambda(R_\lambda)\).
As long as $A_i = 0$ holds in $\hat{H}_0$, the random function $R$ is never evaluated at the same input by two constructions $W_t(\cdot)$ and $W_t(\cdot')$, and thus as long as $A_i$ is 0, the system $\hat{H}_0$ behaves as $H_1$: Corollary 2.7 and Lemma 2.8 imply that for all distinguishers $D$ issuing at most $q$ queries to each of the $\lambda$ subsystems we have

$$\Delta^D(\hat{H}_0, H_1) \leq \nu^D(\hat{H}_0) \leq \left(\frac{\lambda}{2}\right)q^2 2^{-n}. \quad \Box$$

Moreover, for all $i$ and keys $s_1 \in S_1, \ldots, s_{i-1} \in S_{i-1}$, the appropriate construction $T_{s_1, \ldots, s_{i-1}}(\cdot)$ generates random keys $S_i, \ldots, S_m$ and whenever invoked, it issues a query to $K(S)$, obtaining $(r, y)$, and outputs the pair

$$\left(r, \bigoplus_{j=1}^{i-1} F_j(s_j)(r) \oplus y \oplus \bigoplus_{j=i+1}^{m} F_j(S_j)(r)\right).$$

It is easy to see that these constructions satisfy property (ii), since $K(\cdot)$ evaluates the given function at a fresh random input upon each invocation. Therefore, Theorem 4.6 yields the following result.

**Corollary 4.13.** For all $t, q, \gamma > 0$,

$$\Delta_{t, q}(K(F_1 \oplus \cdots \oplus F_m), K(R)) \leq \prod_{i=1}^{m} \Delta_{t'_i, q'_i}(K(F_i), K(R))$$

$$+ \frac{(m - 1)q^2 \lambda^2}{2^{n+1}} + \gamma,$$

where $t'_i := \lambda(t + O(q \cdot \sum_{j \neq i} t_{F_j}))$ and $q'_i := \lambda \cdot q$ for all $i = 1, \ldots, m - 1$, whereas $t'_m := t + O(q \cdot \sum_{j=1}^{m-1} t_{F_j})$ and $q'_m := q$.

The result holds for any other quasi-group operation. It is remarkable that XOR satisfies much stronger indistinguishability amplification properties under random-input attacks than under chosen-input attacks.

**4.5.3. Randomized XOR of PRFs**

The first product theorem for PRFs, due to Myers [Mye03], considered the neutralizing composition $Z_1(F_1) \oplus \cdots \oplus Z_m(F_m)$ for independent instances of $Z(\cdot)$ (which we assume to be instantiated with $\oplus$, but any
other quasi-group operation would work). We show that this result is implied by Theorem 4.6. In fact, the same result holds for the construction \(Z(F_1 \oplus \cdots \oplus F_m)\) using the same offset for all invocations.

More precisely, fix \(i \in \{1, \ldots, m\}\) and \(s_1 \in S_1, \ldots, s_{i-1} \in S_i\), and define the constructions \(M(\cdot)\) and \(M'(\cdot)\) such that for any random function \(S : \{0,1\}^n \to \{0,1\}^{\ell'}\),

\[M(S) := Z(F_1(s_1) \oplus \cdots \oplus F_{i-1}(s_{i-1}) \oplus S \oplus F_{i+1} \oplus \cdots \oplus F_m).\]

and

\[M'(S) := Z_1(F_1(s_1)) \oplus \cdots \oplus Z_{i-1}(F_{i-1}(s_{i-1})) \oplus Z_i(S) \oplus Z_{i+1}(F_{i+1}) \oplus \cdots \oplus Z_m(F_m),\]

**Lemma 4.14.** Let \(R, R_1, \ldots, R_\lambda : \{0,1\}^n \to \{0,1\}^\ell\) be independent URFs, and let \(M_1(\cdot), \ldots, M_\lambda(\cdot)\) be independent instances of \(M(\cdot)\), then

\[\Delta_{q, \ldots, q}(M_1(R) \ldots M_\lambda(R), M_1(R_1) \cdots M_\lambda(R_\lambda)) \leq \frac{q^2 \cdot \lambda^2}{4} \cdot 2^{-n}\]

and

\[\Delta_{q, \ldots, q}(M_1(R) \ldots M_\lambda(R), M_1(R_1) \cdots M_\lambda(R_\lambda)) \leq \frac{q^2 \cdot \lambda^2}{4} \cdot 2^{-n}.\]

In other words, the URF \(R\) is \(\eta\)-self-independent under \(M(\cdot)\) and \(M'(\cdot)\) for \(\eta(q, \lambda) \leq \frac{q^2 \lambda^2}{2} \cdot 2^{-n}\).

**Proof.** We just prove the statement for \(M(\cdot)\), as the proof for \(M'(\cdot)\) is fully analogous. We define the system \(\tilde{H}_0\) with MBO \(A_1, A_2, \ldots\) such that

\[\tilde{H}_0^- := M_1(R) \cdots M_\lambda(R)\]

To define the MBO, let \(Z_1, \ldots, Z_\lambda\) be the random offsets chosen by \(Z(\cdot)\) in \(M_1, \ldots, M_\lambda\), respectively. We let \(A_i := 0\) if and only if among the first \(i\) queries there exists no two queries \(X_j, X_k, j \neq k\) to distinct subsystems \(t, t' \in \{1, \ldots, \lambda\}\) such that \(X_i \oplus Z_i = X_j \oplus Z_j\), and \(A_i := 1\) otherwise. Clearly, as long as this does not hold, every query (regardless of the systems) returns an independent random output, i.e., behaves as \(H_1 := M_1(R_1) \cdots M_\lambda(R_\lambda)\), and it is easy to verify that we can apply both Corollary 2.7 and Lemma 2.8, and for all distinguishers \(D\) issuing at most \(q\) queries to each of the \(\lambda\) subsystems we have

\[\Delta^D(\tilde{H}_0^-, H_1) \leq \max_{x, x' : x \neq x'} p_{\tilde{H}_0^-, X, \lambda}(1, x^\lambda).\]
Security Amplification for Neutralizing Constructions

where $x^{\lambda q}$ is a sequence of queries with $q$ queries to each subsystem, and for each such sequence and for each pair of systems $t \neq t'$ there are at most $q^2$ pairs of possibly colliding queries $X_i \oplus Z_{t_i}, X_j \oplus Z_{t'_j}$, and the probability that they collide is exactly $2^{-n}$. By the union bound we have

$$p_{A_{\lambda q}|X^{\lambda q}}(1, x^{\lambda q}) \leq \left( \frac{\lambda}{2} \right) \cdot q^2 \cdot 2^{-n}. \quad \square$$

A major advantage of Myers’ original construction (which was unobserved so far) is that independent instances of the construction can be simulated even when only given access to $Z(S)$ (with $S \in \{F_i, R \}$). The relevance of this observation is due to the fact that the best advantage under $Z(\cdot)$ can be significantly smaller than under direct access: Consider e.g. a good PRF which is modified to have the additional property of outputting the zero string when evaluated at some fixed known input, regardless of the key.

In order to apply Theorem 4.6, the construction $T_{s_1, \ldots, s_{i-1}}(\cdot)$ chooses independent instances $F_{i+1}, \ldots, F_m, Z_1(\cdot), \ldots, Z_{i-1}(\cdot), Z_{i+1}(\cdot), \ldots, Z_m(\cdot)$, and a random $n$-bit string $Z$, and on input $x$ queries $x \oplus Z$ to $Z(S)$, obtaining $y \in \{0, 1\}^\ell$, and outputs

$$y \oplus \bigoplus_{j=1}^{i-1} Z_j(F_j(s_j))(x) \oplus \bigoplus_{j=i+1}^{m} Z_j(F_j)(x),$$

where $Z_j(F_j)(x)$ is the result of invoking the system $Z_j(F_j)$ on input $x$.

Once again, condition (ii) is easily verified by the fact that access through $Z(\cdot)$ can be re-randomized by simply adding a fresh random offset to all inputs. Thus, Theorem 4.6 yields the following strengthened version of the main result of [Mye03].

**Corollary 4.15.** Let $Z_1(\cdot), \ldots, Z_m(\cdot)$ be independent instances of $Z(\cdot)$. For all $t, q, \gamma > 0$,

$$\Delta_{t,q}(Z_1(F_1) \oplus \cdots \oplus Z_m(F_m), R) \leq \prod_{i=1}^{m} \Delta_{t'_i,q'_i}(Z(F_i), Z(R)) + \frac{(m-1)q^2\lambda^2}{2^{n+1}} + \gamma,$$

where $t'_i := \lambda(t + O(q \cdot \sum_{j \neq i} t_{F_j}))$ and $q'_i := \lambda \cdot q$ for all $i = 1, \ldots, m - 1$, whereas $t'_m := t + O(q \cdot \sum_{j=1}^{m-1} t_{F_j})$ and $q'_m := q$. 
Despite the generality of the results of Chapter 4, Theorem 4.2 only yields a mild form of advantage amplification, whereas Theorem 4.6 implies optimal amplification for a restricted class of randomized constructions only. Therefore, there exist examples, such as the (plain) cascade, for which only Theorem 4.2 applies, although one hopes for better bounds. To bridge this gap, this chapter introduces a class of constructions — called extracting constructions — for which we prove a strong computational indistinguishability amplification theorem.

Our paradigm is best illustrated by considering two finite random variables $X$ and $Y$ with statistical distance $d(X, Y) = \varepsilon$. By Lemma 2.2, it is always possible to adjoin events $A$ and $B$ to them such that:

(i) $P[A] = P[B] = 1 - \varepsilon = 1 - d(X, Y)$;

(ii) $P_{X|A} = P_{Y|B}$, i.e, conditioned on both events occurring, $X$ and $Y$ have the same distribution.

If $Y = U$ is uniformly distributed on $S$, these two properties imply that the random variable $X$, with probability $1 - \varepsilon$, has high min-entropy

$$H_\infty(X|A) = H_\infty(Y|B) \geq \log |S| - \log \left(\frac{1}{1 - \varepsilon}\right),$$
i.e., which is at most \( \log \left( (1 - \varepsilon)^{-1} \right) \) away from the maximum achievable value \( \log |S| \).

The core of our paradigm consists of a computational analogue of this property which applies to any (cc-stateless) random function \( F \) such that, for all polynomially-bounded \( t \) and \( q \),

\[
\Delta_{t, q}(F, J) \leq \varepsilon,
\]

where \( J \) realizes a function chosen uniformly from a set \( \mathcal{F} \). Informally, we show that with probability \( 1 - \varepsilon \), the random function \( F \) is computationally indistinguishable from a random function \( J' \) whose function table has min-entropy at least \( \log |\mathcal{F}| - \log \left( (1 - \varepsilon)^{-1} \right) \).

This reduces the task of proving computational indistinguishability amplification for a construction \( C(\cdot) \) to the substantially simpler problem of showing that \( C(\cdot) \) implements a target ideal system \( I \) whenever a sufficiently large fraction of its subsystems implements a random functions whose function table has sufficiently large min-entropy: We call such constructions \( C(\cdot) \) extracting, in view of their obvious similarity with the concept of randomness extractors. The cascade of random permutations will turn out to be such a construction.

**Chapter Outline and Contributions.** We provide a more detailed road map of the contents of this chapter:

(i) We start, in Section 5.1.1, by restating (Theorem 5.1) the main technical lemma of [MPR07]. It generalizes the above characterization of the distance of random variables by showing that one can define MBOs on any two systems \( S \) and \( T \) such that (i) both systems behave in the same way as long as the MBOs are 0, and (ii) the probability that each of the MBOs becomes 1 after \( q \) queries equals the distinguishing advantage \( \Delta_{q}(S, T) \).

We provide a proof based on the information-theoretic Hardcore Lemma for system-bit pairs (Theorem 3.3) in order to introduce the techniques used to prove the computational results of this chapter.

(ii) In Section 5.1.2, we present the first main technical result of this chapter, which we refer to as the Hardcore Lemma (HCL) for computational indistinguishability and which is a computational analogue of Theorem 5.1. Informally, it states that for any two cc-stateless systems \( S \equiv S(S) \) and \( T \equiv T(T) \) with \( \Delta_{\text{poly}}(S, T) \leq \varepsilon \) there exist events defined on \( S \) and \( T \), each occurring with probability \( 1 - \varepsilon \), such that
conditioned on these events, \( S(S) \) and \( T(T) \) are computationally indistinguishable.

In analogy to the information-theoretic case, the proof relies on the Hardcore Lemma for system-bit pairs, and it is given both in the non-uniform (Section 5.1.2) and in the uniform (Section 5.1.3) settings.

(iii) Section 5.2 introduces the class of extracting constructions, for which we prove a general security amplification theorem (both in the non-uniform and in the uniform settings) based on the HCL for computational indistinguishability, and as an application, we present a number of new results:

- In Section 5.3, we present a new paradigm for security amplification of \( \varepsilon \)-PRGs for any \( \varepsilon < 1 \), which improves on previous techniques (illustrated in Section 4.3 as an application of Theorem 4.2) requiring \( \varepsilon < \frac{1}{2} \). Our construction can be instantiated to be fully deterministic.

- Our main application, presented in Section 5.5, is a proof that the cascade of \( m \varepsilon \)-PRPs is a \( (\varepsilon^m (m - (m - 1)\varepsilon) + \nu) \)-PRP (for a negligible function \( \nu \)), and we show this bound to be tight. (A slightly weaker bound is obtained in the uniform case.) This bound implies efficient amplification via cascading for all \( \varepsilon < 1 \) such that \( 1 - \varepsilon \) is noticeable, without the need of random offsets as in the randomized cascade (cf. Section 4.5.1). This settles an open problem due to Luby and Rackoff [LR86].

- Section 5.6 presents alternative (and somewhat more natural) security amplification proofs for the randomized cascade and the random-offset constructions presented in Sections 4.5.1 and 4.5.3.

(iv) A central tool in proving the extracting property is a characterization of the input-output behavior of cc-stateless random functions with high min-entropy function tables, which generalizes a previous result by Unruh [Unr07], and which is proven in Section 5.4.

We remark that by means of the notion of an extracting construction, this chapter presents a generalized and unified exposition of the contents of [MT10] and [Tes10]. The former paper introduced the concept of the HCL for computational indistinguishability in the case of random variables
and applied this to security amplification of PRGs, whereas the latter paper subsequently extended the HCL to interactive systems to obtain tight bounds for the cascade of PRPs.

Related Work. To the best of our knowledge, the only result in the literature employing a technique which is a special case of an extracting construction is the so-called extraction lemma, used within PRG constructions from one-way functions [HILL99, Hol06] (a similar result was also used by Sudan et al. [STV01]): One proves that the concatenation of sufficiently many mildly unpredictable bits (in the sense of Chapter 3) contains enough computational entropy which can be extracted by an appropriate randomness extractor. However, this technique only applies to bits, rather than arbitrary random variables.

Advantage amplification for the cascade of PRPs was first studied by Luby and Rackoff [LR86] for the case of two permutations, and their results was further generalized by Myers [Mye99] to longer cascade. While Luby and Rackoff’s original bound was tight (and matches our bound for the case \(m = 2\)), this is not true for Myers’ generalization, which also only applies to constant-length cascades, and hence falls short of solving the problem of security amplification for arbitrarily weak PRPs.

A related (but orthogonal) line of work to ours considers so-called dense subsets of pseudorandom sets. Roughly speaking, for a PRG \(G : \{0,1\}^k \to \{0,1\}^\ell\) and a parameter \(t\) (such as \(t = \text{poly}(\log k)\)), Reingold et al. [RTTV08] show that for any \(k\)-bit distribution \(U'_k\) of min-entropy at least \(k - t\), there exists an \(\ell\)-bit random variable \(U'_\ell\) with min-entropy at least \(\ell - t\) such that \(G(U'_k)\) and \(U'_\ell\) are computationally indistinguishable. This result has been independently proven by Dziembowski and Pietrzak [DP08], who used it in order to build secure cryptosystems resilient to leakage of information about their secret values.

5.1. Hardcore Lemmas for Computational Indistinguishability

5.1.1. The Information-Theoretic Case: Maximal MBOs

Recall (see Lemma 2.2) that there exist events \(A\) and \(B\) on any two random variables \(X\) and \(Y\) with equal range \(S\) such that \(\Pr_{X,A} = \Pr_{Y,B}\) and \(\Pr[A] = \Pr[B] = 1 - d(X,Y)\). A generalization to arbitrary interactive sys-
5.1 Hardcore Lemmas for Computational Indistinguishability

For any two \((X, Y)\)-systems \(S\) and \(T\) we can define \((X, Y)\)-systems \(\hat{S}\) and \(\hat{T}\) with MBOs such that

(i) \(\hat{S}^- \equiv S\) and \(\hat{T}^- \equiv T\),

(ii) \(\hat{S}^+ \equiv \hat{T}^+\),

(iii) \(\nu_q(\hat{S}) = \nu_q(\hat{T}) = \Delta_q(S, T)\) for all \(q \geq 1\).

We give a simple proof based on the information-theoretic HCL for system-bit pairs (Theorem 3.3): It illustrates the technique used, below, to derive the HCL for computational indistinguishability from the (computational) HCL for system-bit pairs.

The key step is to consider the system-bit pair \((F, B) := (\langle S, T \rangle_B, B)\) for a uniform random bit \(B\). By Lemma 3.2,

\[
\Delta_q(S, T) = \text{Guess}_q(B | \langle S, T \rangle_B).
\]

Additionally, by Theorem 3.3, there exists a system-bit pair \((\hat{F}, \hat{B})\) with an MBO \(C_1, C_2, \ldots\) such that \((\hat{F}, \hat{B})^- \equiv (F, B)\) and, for all \(i \geq 1\),

\[
p_{Y_i, C_i=0, B=0 | X_i}^{(\hat{F}, \hat{B})} = p_{Y_i, C_i=0, B=1 | X_i}^{(F, B)}.
\]

We define two systems \(\hat{S}\) with MBO \(A_1, A_2, \ldots\) and \(\hat{T}\) with MBO \(B_1, B_2, \ldots\) such that, for all \(i \geq 1\),

\[
\hat{p}_{A_i, Y_i | X_i, Y_{i+1}=1, A_{i-1}}^{(\hat{F}, \hat{B})} := p_{C_i, Y_i | X_i, Y_{i+1}=1, C_{i-1}, B=0}^{(\hat{F}, \hat{B})},
\]

\[
\hat{p}_{B_i, Y_i | X_i, Y_{i+1}=1, B_{i-1}}^{(\hat{F}, \hat{B})} := p_{C_i, Y_i | X_i, Y_{i+1}=1, C_{i-1}, B=1}^{(\hat{F}, \hat{B})}.
\]

It is easy to verify that \(\hat{S}^- \equiv S\) and \(\hat{T}^- \equiv T\). Furthermore, we have \(S^+ \equiv T^+\), since for all \(i \geq 1\),

\[
p_{A_i=0 Y_i | X_i}^{\hat{S}} = p_{C_i=0 Y_i | X_i, B=0}^{(\hat{F}, \hat{B})} = 2 \cdot p_{C_i=0 B=0 | Y_i | X_i}^{(\hat{F}, \hat{B})} = 2 \cdot p_{C_i=0 B=1 | Y_i | X_i}^{(\hat{F}, \hat{B})} = p_{B_i=1 Y_i | X_i}^{\hat{T}} = p_{B_i=0 Y_i | X_i}^{\hat{T}}.
\]

This in particular implies \(\nu_q(\hat{S}) = \nu_q(\hat{T})\), and we conclude by noticing that given \(\hat{B} = 0\), provoking \(C_q = 1\) in \((\hat{F}, \hat{B})\) is equivalent to provoking

\(^1\)A weaker version was previously proven by Maurer and Pietrzak [MP04].
A_q = 1 in \( \hat{S} \), and symmetrically, given \( \hat{B} = 1 \), provoking \( C_q = 1 \) in \(( \hat{F}, \hat{B} )\) is equivalent to provoking \( B_q = 1 \) in \( \hat{T} \). That is,

\[
\Delta_q(S, T) = \text{Guess}_q(B \mid F) = \frac{1}{2} \left( \nu_q(S) + \nu_q(T) \right),
\]

which implies (iii).

To summarize, we have gone through three steps: First, we define the system-bit pair \(( F, B )\). Second, we compute a maximal MBO for \(( F, B )\), and finally, we “distribute” the part of the MBO corresponding to the case \( B = 0 \) to \( S \), and the one corresponding to \( B = 1 \) to \( T \). We use a similar approach in the computational case below.

5.1.2. The Non-Uniform HCL for Computational Indistinguishability

Given two cc-stateless systems \( S \equiv S(S) \) and \( T \equiv T(T) \) such that, for some given \( t \) and \( q \),

\[
\Delta_{t,q}(S, T) \leq \varepsilon,
\]

we prove that, for all \( \gamma > 0 \), there exist events \( A \) and \( B \), both with probability at least \( 1 - \varepsilon \), such that

\[
\Delta_{t',q'}(S(S'), T(T')) \leq \gamma
\]

for \( S' \xleftarrow{\$} P_{S \mid A} \) and \( T' \xleftarrow{\$} P_{T \mid B} \), where \( t' < t \) and \( q' < q \). In contrast to Theorem 5.1 (but in analogy to Theorem 3.5), this statement does not define MBOs, but instead provide events defined on the initial state only, which is the strongest statement one could expect.

Once again, we use the language of measures, rather than events, to simplify notation in the proof. Also, we let \( \varphi_{hc} \) and \( \psi_{hc} \) be as defined for Theorem 3.5. A uniform version of the theorem is discussed and proven in the next section.

Theorem 5.2 (Hardcore Lemma for Computational Indistinguishability).

Let \( S \equiv S(S) \) and \( T \equiv T(T) \) be cc-stateless systems, with respective implementations \( A_S \) (with space complexity \( s_{A_S} \)) and \( A_T \) (with space complexity \( s_{A_T} \)). Furthermore, for some integers \( t, q > 0 \) and some \( \varepsilon \in (0, 1) \),

\[
\Delta_{t,q}(S, T) \leq \varepsilon.
\]

Then, for all \( 0 < \zeta_1, \zeta_2 < 1 \) and all \( 0 < \gamma \leq \frac{1}{2} \), there exist measures \( \mathcal{M}_S \) and \( \mathcal{M}_T \) such that \( \mu(\mathcal{M}_S) \geq 1 - \varepsilon \) and \( \mu(\mathcal{M}_T) \geq 1 - \varepsilon \) and the following properties hold:
5.1 Hardcore Lemmas for Computational Indistinguishability

(i) For \( S' \leftarrow M_S, T' \leftarrow M_T, t' := t/\varphi_{hc} \) and \( q' := q/\varphi_{hc} \), we have 
\[ \Delta_{t',q'}(S(S'), T(T')) \leq 2\gamma; \]

(ii) There exist a \((\zeta_1, \zeta_2)\)-sampler \( O_S \) for \( M_S \) and \( A_S \) with length \( s_{A_S}(\psi \cdot q') \) and a \((\zeta_1, \zeta_2)\)-sampler \( O_T \) for \( M_T \) and \( A_T \) with length \( s_{A_T}(\psi \cdot q') \). Furthermore, if both \( S \) and \( T \) are random functions, then both samplers can be made error-less with lengths \( s_{A_S}(\psi \cdot q') \) and \( s_{A_T}(\psi \cdot q') \), where \( \psi := T \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3} \).

Proof. Let \( S \) and \( T \) be the respective ranges of \( S \) and \( T \). Along the lines of Theorem 5.1, \((F, B)\) is the cc-stateless system-bit pair with a uniform random bit \( B \) and where \( F \) behaves as \( S \) if \( B = 0 \) and as \( T \) if \( B = 1 \). In particular, \((F, B)\) has initial state \((x, b) \in (S \times \{0\}) \cup (T \times \{1\})\), sampled by first letting \( b \leftarrow \{0, 1\} \), and then choosing \( x \leftarrow P_S \) if \( b = 0 \) and \( x \leftarrow P_T \) otherwise, and
\[ (F(x, b), B(x, b)) = \begin{cases} (S(x), 0) & \text{if } b = 0, \\ (T(x), 1) & \text{if } b = 1. \end{cases} \]

The canonical implementation \( A_{(F, B)} \) of \((F, B)\) is obtained by first choosing the random bit \( b \leftarrow \{0, 1\} \), and then using \( A_S \) or \( A_T \) to simulate the respective system: The state of \( A_{(S, B)} \) is a pair \((\sigma, b)\) consisting of the bit \( b \) and the current state \( \sigma \) of \( A_S \) or \( A_T \).

By Lemma 3.2,
\[ \Delta_{t,q}(S, T) = \text{Guess}_{t,q}(B \mid F) \leq \varepsilon, \]
and Theorem 3.5 thus implies that there exists a measure \( M \) on \((S \times \{0\}) \cup (T \times \{1\})\) such that \( \mu(M) \geq 1 - \varepsilon \), and
\[ \text{Guess}_{t,q}(B' \mid F(X')) \leq \gamma, \tag{5.1} \]
where \((X', B') \leftarrow M, t' := t/\varphi_{hc} \) and \( q' := q/\varphi_{hc} \). Furthermore, a \((\zeta_1, \zeta_2)\)-sampler \( O \) is associated with the measure \( M \) and \( A_{(F, B)} \).

First, note that \( P_{B'}(0) \in \left[ \frac{1 - \gamma}{2}, \frac{1 + \gamma}{2} \right] \), since otherwise there exists a fixed value \( b' \in \{0, 1\} \) which can be output by an adversary (with constant time complexity and making no query) to achieve advantage higher than \( \gamma \), contradicting (5.1). In the following, we assume without loss of generality that \( \frac{1 - \gamma}{2} \leq P_{B'}(0) \leq \frac{1}{2} \), and \( \frac{1}{2} \leq P_{B'}(1) \leq \frac{1 + \gamma}{2} \). (The other case is symmetric.)
We define the measures $\mathcal{M}_0 : \mathcal{S} \to [0, 1]$ and $\mathcal{M}_1 : \mathcal{T} \to [0, 1]$ such that

$$\mathcal{M}_0(s) := M(s, 0) \quad \text{and} \quad \mathcal{M}_1(t) := M(t, 1)$$

for all $s \in \mathcal{S}$ and all $t \in \mathcal{T}$. Then note that

$$\mu(\mathcal{M}_0) = \sum_{s \in \mathcal{S}} P_S(s) \cdot M_0(s) = 2\mu(M) \cdot P_B(0),$$

$$\mu(\mathcal{M}_1) = \sum_{t \in \mathcal{T}} P_T(t) \cdot M_1(t) = 2\mu(M) \cdot P_B(1),$$

which implies $(1 - \gamma) \cdot \mu(M) \leq \mu(\mathcal{M}_0) \leq \mu(M)$ and $\mu(M) \leq \mu(\mathcal{M}_1) \leq (1 + \gamma) \cdot \mu(M)$. Moreover, we set $\mathcal{M}_T := \mathcal{M}_1$, and, if $\mu(\mathcal{M}_0) < 1 - \varepsilon$, we define $\mathcal{M}_S$ such that

$$\mathcal{M}_S(s) := \frac{\varepsilon}{1 - \mu(\mathcal{M}_0)} \cdot M_0(s) + \frac{1 - \varepsilon - \mu(\mathcal{M}_0)}{1 - \mu(\mathcal{M}_0)}$$

for all $s \in \mathcal{S}$. (If $\mu(\mathcal{M}_0) \geq 1 - \varepsilon$, then $\mathcal{M}_S := \mathcal{M}_0$.) Observe that we always have $M_0(s) \leq M_S(s) \leq 1$, and $1 - \varepsilon \leq \mu(M_S) \leq \mu(M)$, since if $\mu(\mathcal{M}_0) < 1 - \varepsilon$, then

$$\mu(\mathcal{M}_S) = \sum_{s \in \mathcal{S}} P_S(s) \cdot M_S(s)$$

$$= \frac{\varepsilon}{1 - \mu(\mathcal{M}_0)} \cdot \mu(M_0) + \frac{1 - \varepsilon - \mu(\mathcal{M}_0)}{1 - \mu(\mathcal{M}_0)} = 1 - \varepsilon.$$

We now consider the distribution $(X'', B'')$ which chooses $B''$ uniformly at random, and then $X''$ according to $\mathcal{M}_S$ or $\mathcal{M}_T$ depending on whether $B''$ equals 0 or 1. With $S' \xleftarrow{\$} \mathcal{M}_S$ and $T' \xleftarrow{\$} \mathcal{M}_T$,

$$\Delta_{\varepsilon', \varepsilon''}(S(S'), T(T')) = \text{Guess}_{\varepsilon', \varepsilon''}(B'' \mid F(X''))$$

$$\leq \text{Guess}_{\varepsilon', \varepsilon''}(B' \mid F(X')) + 2 \cdot d((B', X'), (B'', X''))$$

$$\leq \gamma + 2 \cdot d((B', X'), (B'', X'')),$$

since

$$\text{Guess}_A(B' \mid F(X')) - \text{Guess}_A(B'' \mid F(X'')) \leq 2 \cdot d((X', B'), (X'', B''))$$

for all adversaries $A$. \footnote{To see this, note that the (inefficient) distinguisher which given a pair $(X, B)$ simulates the interaction of $A$ with $F(X)$ and outputs 1 if and only if its output equals $B$ achieves advantage $\frac{1}{2} [\text{Guess}_A(B' \mid F(X')) - \text{Guess}_A(B'' \mid F(X''))] \leq d((X', B'), (X'', B''))$.} The following claim yields the desired upper bound.
Claim 6. $d((X', B'), (X'', B'')) \leq \frac{\gamma}{2}$.

Proof. On the one hand, we first note that for all $t \in T$ we have

$$P_{X'' B''}(t, 1) = \frac{1}{2 \mu(M_1)} \cdot P_T(t) \cdot M(t, 1)$$

where we have used the fact that $\mu(M_1) \geq \mu(M)$. On the other hand, for all $s \in S$ we have (using $\mu(M_S) \leq \mu(M)$ and $M_0(s) \leq M_S(s)$)

$$P_{X'' B''}(s, 0) = \frac{1}{2 \mu(M_S)} \cdot P_S(s) \cdot M_S(s)$$

Therefore,

$$d((X', B'), (X'', B'')) = \sum_{t \in T} P_{X' B'}(t, 1) - P_{X'' B''}(t, 1)$$

For all $t \in T$ we have

$$\leq \frac{1}{2 \mu(M)} \cdot P_T(t) \cdot M(t, 1) = P_{X' B'}(t, 1),$$

where

$$= \frac{1}{2} \sum_{t \in T} P_T(t) \cdot M(t, 1) \cdot \left( \frac{1}{\mu(M)} - \frac{1}{\mu(M_1)} \right)$$

$$= \frac{1}{2} \left( \frac{\mu(M_1)}{\mu(M)} - 1 \right) \leq \frac{\gamma}{2}. \quad \Box$$

To conclude the proof, note the state sampler $O$ for $M$ and $A_{(F, B)}$ with length $l = s_{A_{(F, B)}}(q' \cdot \psi)$ outputs a triple $(b, \sigma, z)$, where $\sigma$ is a state for $A_S$ if $b = 0$ and a state of $A_T$ when $b = 1$. In particular, for $(\Sigma, B, Z) \xleftarrow{} O$,

$$(A_B[\Sigma], B, Z) \equiv (F(X, B), B, Z(X, B)),$$

where $A_0 := A_S$, $A_1 := A_T$, and $Z(X, B)$ is at most $\zeta_1$ off $M(X, B)$, except with probability $\zeta_2$, for all values $x, b$ taken by $X, B$.

Consequently, the state sampler $O_T$ for $M_T$ outputs $(\sigma, z)$ sampled according to the output distribution $O$ conditioned on $B = 1$. The sampler $O_S$ outputs $(\sigma, z')$ by taking $(\sigma, z)$ sampled from $O$ conditioned on $B = 0$, and then sets $z' := z$ if $\mu(M_0) \geq 1 - \varepsilon$, and otherwise sets

$$z' := \frac{\varepsilon}{1 - \mu(M_0)} \cdot z + \frac{1 - \varepsilon - \mu(M_0)}{1 - \mu(M_0)}.$$
Clearly, it remains a \((\zeta_1, \zeta_2)\)-sampler, due to \(\mu (\mathcal{M}_S) \leq 1\).

Note that Theorem 3.5 only guarantees that the lengths of the states output by both samplers are bounded by \(\max\{s_{A_S}(\psi_{hc} \cdot q'), s_{A_T}(\psi_{hc} \cdot q')\}\). The fact that the individual bounds hold for each of the states can be inferred from a careful analysis of the proof of Theorem 3.5. (This more precise statement is in general not necessary in the following, but will produce slightly nicer statements.)

Also, if both \(S\) and \(T\) are cc-stateless random functions, then \((F, B)\) has deterministic behavior for any value of its initial state, and thus \(O\) is error-less, which implies that \(O_S\) and \(O_T\) are also error-less.

5.1.3. The Uniform HCL for Computational Indistinguishability

The above proof of Theorem 5.2 is inherently non-uniform, as it uses quantities which cannot be determined exactly in a uniform reduction. However, we can provide a uniform reduction that follows the above lines to obtain a uniform version based Theorem 3.11. (In the uniform case, recall that we set \(\psi_{hc} := \frac{7200}{\zeta(1-\epsilon)} \cdot \ln \left(\frac{2}{\epsilon^2}\right)\).

**Theorem 5.3.** Let \(S \equiv S(S)\) and \(T \equiv T(T)\) be cc-stateless systems, with respective efficient implementations \(A_S\) and \(A_T\). Furthermore, assume that for some \(\epsilon \in [0, 1)\) (such that \(1 - \epsilon\) is noticeable),

\[
\Delta_{poly}(S, T) \leq \epsilon.
\]

For all noticeable \(\zeta_1 > 0,\) all \(\zeta_2 = 2^{-\text{poly}(k)} > 0,\) and all \(0 < \gamma \leq \frac{1}{2}\) (such that \(\frac{2\zeta_1}{\zeta_2^2} + \zeta_2 \leq \frac{1}{2}\) and \(1 - \epsilon - \zeta_1 - \zeta_2\) is noticeable), and for all polynomial-time \(q'\)-query (uniform) oracle distinguishers \(D(\cdot, \cdot)\), there exist measures \(\mathcal{M}_S\) and \(\mathcal{M}_T\) such that \(\mu(\mathcal{M}_S) \geq 1 - \epsilon\) and \(\mu(\mathcal{M}_T) \geq 1 - \epsilon\) and the following properties hold:

(i) There exist \((\zeta_1, \zeta_2)\)-samplers \(O_S\) and \(O_T\) for \(\mathcal{M}_S\) and \(A_S\) as well as for \(\mathcal{M}_T\) and \(A_T\), respectively, with length \(\ell := \max\{s_{A_S}(\psi_{hc} \cdot q'), s_{A_T}(\psi_{hc} \cdot q')\}\). Furthermore, if both \(S\) and \(T\) are random functions, then both samplers can be made error-less with length \(\ell := \max\{s_{A_S}(\psi \cdot q'), s_{A_T}(\psi \cdot q')\}\) for \(\psi := 256 \cdot \gamma^{-2} \cdot (1 - \epsilon)^{-3}\).

(ii) For \(S' \xleftarrow{\$} \mathcal{M}_S\) and \(T' \xleftarrow{\$} \mathcal{M}_T\),

\[
\Delta^{D_{O_S \cdot O_T}}(S(S'), T(T')) \leq 13\gamma.
\]
5.1 Hardcore Lemmas for Computational Indistinguishability

Proof. Assume that Theorem 5.3 is false, that is:

There exist noticeable \( \zeta_1 > 0, \zeta_2 = 2^{-\text{poly}(k)} > 0, \) and \( 0 < \gamma \leq \frac{1}{2} \) (such that \( \frac{2\zeta_1}{\zeta_2} + \zeta_2 \leq \frac{1}{2} \) and \( 1 - \varepsilon - \zeta_1 - \zeta_2 \) is noticeable),
a polynomially bounded \( q' \), and an efficient \( q' \)-query oracle distinguisher \( D^* = (D^*)^{(\cdot)} \) which satisfies

\[
\Delta^{(D^*)^{(S,T)}} (S(S'), T(T')) > 13 \gamma
\]

for all measures \( M_S \) and \( M_T \), both with density at least \( 1 - \varepsilon \),

admitting respective \( (\zeta_1, \zeta_2) \)-state-samplers \( O_S \) and \( O_T \) with length \( \ell \) for the corresponding implementations \( A_S \) and \( A_T \),

and for \( S' \sim M_S, T' \sim M_T \).

Then, we prove that for the cc-stateless system-bit pair \( (F, B) \) defined as in the proof of Theorem 5.2, there exists an efficient \( q' \)-query oracle adversary \( A^* = (A^*)^{(\cdot)} \) such that for all measures \( M \) for \( (F, B) \) with density at least \( 1 - \varepsilon \) admitting a \( (\zeta_1, \zeta_2) \)-state-sampler \( O \) for \( A_{(F,B)} \) with length \( \ell \),

\[
\text{Guess}^{(A^*)^{(\cdot)}} (B' | F(X', B')) > \gamma
\]

for \( (X', B') \sim M \), except with negligible probability. As this suffices to contradict the uniform Hardcore Lemma for the system-bit pair \( (F, B) \) (Theorem 3.11),

there exists a uniform polynomial-time distinguisher \( D \) such that

\[
\Delta^{D} (S, T) \geq \text{Guess}^{D} (B | F) > \varepsilon,
\]

contradicting the indistinguishability of \( S \) and \( T \).

The adversary \( A^* \), given oracle access to the sampler \( O \) for \( M \), proceeds as detailed in Figure 5.1. It first produces an estimate \( \overline{p}_0 \) of the probability \( p_0 \) that \( B' = 0 \). Then, if \( B' \) takes a value \( b \in \{0, 1\} \) with probability substantially larger than \( \frac{1}{2} \), it outputs \( b \). Otherwise, it simply lets the distinguisher \( D^* \) interact with the given system \( F(X', B') \), with oracles \( O_S \) and \( O_T \) which are \( (\zeta_1, \zeta_2) \)-samplers for the measures \( M_S \) and \( M_T \) defined as

\[
M_S(s) := \frac{\varepsilon}{\varepsilon + 3\gamma(1 - \varepsilon)} \cdot M_0(s) + \frac{3\gamma(1 - \varepsilon)}{\varepsilon + 3\gamma(1 - \varepsilon)}
\]

\[
M_T(t) := \frac{\varepsilon}{\varepsilon + 3\gamma(1 - \varepsilon)} \cdot M_1(t) + \frac{3\gamma(1 - \varepsilon)}{\varepsilon + 3\gamma(1 - \varepsilon)}
\]

\[\footnote{As discussed in the proof of Theorem 3.11, the final adversary can be modified to compensate for a negligible error probability.}\]
132 Security Amplification via Extracting Constructions

Adversary \((A^*)^O(F)\): // For system-bit pair \((F, B) \equiv (F(X, B), B)\)
\[
\begin{align*}
  r & := \frac{16k}{\gamma} \\
  \sigma_1, b_1, \ldots, \sigma_r, b_r & \leftarrow \text{StateSample}^O(k \cdot \log\left(\frac{1}{\varepsilon + \xi_1 + \xi_2}\right)) \\
  p_0 & := \frac{1}{r} \\
  \text{if } p_0 > \frac{1}{2} + \gamma & \text{ then return 0} \\
  \text{else if } p_0 < \frac{1}{2} - \gamma & \text{ then return 1} \\
  \text{else} \\
  \delta & := \text{EstimateWin}^O((D^*)^O_s, O_t, \gamma, 1 - \varepsilon) > \frac{1}{2} \text{ then 0} \\
  \text{else} \\
  \delta & := 1 \\
  \text{return } (D^*)^O_s, O_t (F) \oplus \delta
\end{align*}
\]

Figure 5.1: Oracle adversary \(A^*\) in the proof of Theorem 5.3. The value \(k\) is the (implicit) security parameter, and the oracles \(O_s\) and \(O_t\) are simulated from \(O\) as described in the text.

for all \(s \in S\) and \(t \in T\), where \(M_0\) and \(M_1\) are defined as in the proof of Theorem 5.2. The state sampler \(O_s\) is implemented by sampling pairs \(((\sigma_1, b_1), z_1), ((\sigma_2, b_2), z_2), \ldots\) from \(O\) until \(b_i = 0\) is satisfied, and then \((\sigma_i, \varepsilon + 3\gamma(1 - \varepsilon) z_i + \frac{3\gamma(1 - \varepsilon)}{\varepsilon + 2\delta(1 - \varepsilon)} z_i)\) is output. If after \(k\) attempts no pair with \(b_i = 0\) is returned, then it outputs \((\perp, 1)\). (For \(M_T\) the sampler \(O_T\) symmetrically uses \(b_i = 1\).) This yields, along the same lines as in the discussion at the end of the proof of Theorem 5.2, \((\zeta_1, \zeta_2)\)-samplers for both measures, provided that a pair with \(b_i = 0\) (for \(O_s\)) or \(b_i = 1\) (for \(O_T\)) occurs within the \(k\) samples from \(O\).

Finally, \(A^*\) uses \(\text{EstimateWin}\) (cf. Example 3.6) to determine whether the probability that \((D^*)^O_s, O_T\) outputs the correct bit is at least \(\frac{1}{2}\). In the affirmative case, \(A^*\) returns the output bit of \((D^*)^O_s, O_T\), whereas otherwise its output bit is flipped before its returned.

**Analysis.** In the following, we assume that \(\text{StateSample}\) always samples correctly. Note that for the probability \(\tilde{p}_0\) that \(b = 0\) for a pair \((\sigma, b)\)
5.1 Hardcore Lemmas for Computational Indistinguishability

Return by \( \pi \), we have, by Lemma 3.4

\[
|\tilde{\pi}_0 - p_0| \leq \frac{2\zeta_1}{\mu(M)} + \zeta_2 \leq \frac{2\zeta_1}{1 - \varepsilon} + \zeta_2.
\]

By the choice of \( r \), applying Hoeffding’s inequality yields

\[
|\tilde{\pi}_0 - p_0| \leq |\tilde{\pi}_0 - \bar{\pi}_0| + |\bar{\pi}_0 - p_0| \leq \gamma/4 + 2\zeta_1 + \zeta_2 \leq \gamma/2.
\]

except with negligible probability. We now consider two different cases.

CASE 1: \( |\bar{\pi}_0 - \frac{1}{2}| > \gamma \). Assume that \( \bar{\pi}_0 > \frac{1}{2} + \gamma \) (dealing with the second if-statement is symmetric), then \( p_0 > \frac{1}{2} + \gamma \), and outputting 0 achieves advantage \( \gamma \).

CASE 2: \( |\bar{\pi}_0 - \frac{1}{2}| \leq \gamma \). This ensures that \( p_0 = \frac{\mu_0}{\mu(M)} \geq \frac{1 - 3\gamma}{2} \), which in particular implies that \( \pi_S \) and \( \pi_T \) are correct,\(^4\) except with negligible probability. Moreover, from this we also infer that

\[
\mu(M_0) = \sum_{s \in S} P_S(s) \cdot M(s, 0) = 2\mu_0 \geq (1 - 3\gamma) \cdot \mu(M).
\]

Similarly, we can derive \( \mu(M_1) \geq (1 - 3\gamma) \cdot \mu(M) \).

Note that we have \( M_S(s) \geq M_0(s) \) for all \( s \in S \), as well as \( M_T(t) \geq M_1(t) \) for all \( t \in T \), and furthermore

\[
\mu(M_S) = \sum_{s \in S} P_S(s) \cdot M_S(s)
= \varepsilon + 3\gamma(1 - \varepsilon) \cdot \mu(M_0) + \frac{3\gamma(1 - \varepsilon)}{\varepsilon + 3\gamma(1 - \varepsilon)} = 1 - \varepsilon,
\]

and the same obviously holds for \( \mu(M_T) \).

As in the non-uniform proof, we consider \( (X'', B'') \) such that \( B'' \) is uniform and \( X'' \) is sampled according to \( M_S \) if \( B'' = 0 \), and as \( M_T \) otherwise, whereas \( (X', B') \) are sampled according to \( \pi \). The following claim is proven in Appendix A.2.

Claim 7: \( d((X', B'), (X'', B'')) \leq 6\gamma \)

\(^4\)Since \( b_i = 0 \) or \( b_i = 1 \) happens with probability at most \( \frac{3}{4} + \zeta_1 + \zeta_2 \leq \frac{3}{4} + \frac{1}{8} = \frac{7}{8} \).
Our initial assumption implies $\left| \text{Guess}^{(D^*)^{08,0\cdot T}}(B'' | F(X'', B'')) - \frac{1}{2} \right| > 13\gamma$, and by the claim this implies $\left| \text{Guess}^{(D^*)^{08,0\cdot T}}(B' | F(X', B'')) - \frac{1}{2} \right| > \gamma$. To conclude the proof, note that EstimateWin outputs the correct $\delta$, except with negligible probability, and thus we can remove the absolute values.

5.2. Extracting Constructions

5.2.1. Motivation and Definition

Throughout this section, let $\mathcal{F}$ be a set consisting of functions $X \to Y$. Examples are the sets of all functions $X \to Y$, of all permutations on $X = Y$, or of random variables with range $Y$.\(^5\) Also, we let $F \leftarrow F(F)$ and $J \leftarrow J(J)$ be cc-stateless random functions, where the function table $J$ of $J$ is chosen uniformly from $\mathcal{F}$, and such that for some $t, q > 0$, $\Delta_{t,q}(F, J) \leq \varepsilon$.

By Theorem 5.2, there exist events $A$ and $B$, both occurring with probability at least $1 - \varepsilon$, such that $\Delta_{t,q}(F', J(J')) \leq 2\gamma$ for $F' \overset{\epsilon}{\leftarrow} P_{F|A}$ and $J' \overset{\epsilon}{\leftarrow} P_{J|B}$, and the function table $J'$ has min-entropy

$$H_\infty(J') \geq \log |\mathcal{F}| - \log \left( \frac{1}{1 - \varepsilon} \right),$$

since

$$P_{J'}(f) = \frac{P_{J|B}(f)}{P[B]} \leq \frac{P_J(f)}{1 - \varepsilon} = \frac{1}{|\mathcal{F}| \cdot (1 - \varepsilon - 1)}.$$ 

Informally, this is to be interpreted as follows: With probability $1 - \varepsilon$, the random function $F$ is computationally indistinguishable from a random function $J'$ whose function table has min-entropy at least $\log |\mathcal{F}| - \log ((1 - \varepsilon)^{-1})$.

This observation motivates a new class of constructions combining random functions and which are required to be “close” to some target ideal system $I$ whenever a sufficiently large fraction of the tables of its subsystems have large min-entropy.

\(^5\)Such a random variable can be seen as a random function $\{\bot\} \to Y$ with single-input domain.
Definition 5.1. For \( \varepsilon \in [0, 1) \), \( \tau \in \{1, \ldots, m\} \) and \( \delta : \mathbb{N} \to \mathbb{R}_{\geq 0} \), an \( m \)-subsystem construction \( C(\cdot) \) is \((\varepsilon, \tau, \delta)\)-extracting for \( F \) and \( I \), if for all cc-stateless random functions \( F_1, \ldots, F_m \) with tables \( F_1, \ldots, F_m \in F \),

\[
\Delta_q(C(F_1, \ldots, F_m), I) \leq \delta(q)
\]

whenever there exists \( I \subseteq \{1, \ldots, m\} \) with \( |I| \geq \tau \) such that, for all \( i \in I \),

\[
H_\infty(F_i) \geq \log |F| - \log \left( \frac{1}{1 - \varepsilon} \right).
\]

Ideally, we would like \( \delta(q) \) to be negligible (in \( m \)), for \( \varepsilon < 1 - \frac{1}{\text{poly}} \) and for all polynomially bounded \( q \). Also note that a slightly more general definition would be possible, where instead of the parameter \( \varepsilon \), entropy lower bounds for each subsystem are given. However, the current definition will be convenient in order to reduce parameters in all applications of extracting constructions we encounter in this chapter.

5.2.2. The Product Theorem

For convenience, define the shorthand \( p(m, \tau, \varepsilon) := \sum_{i=0}^{\tau-1} \binom{m}{i} (1 - \varepsilon)^i \varepsilon^{m-i} \), i.e., the probability that strictly less than a threshold \( \tau \) out of \( m \) independent events \( A_1, \ldots, A_m \) with \( \Pr[A_i] = 1 - \varepsilon \) occur.

In the following, let \( C(\cdot) \) be an \((\varepsilon, \tau, \delta)\)-extracting construction\(^6\) for function set \( F \) and ideal system \( I \), and which can be implemented with complexity \( t_{AC} \). Furthermore, assume that upon \( q \) queries, the construction \( C(\cdot) \) makes \( q_i \) queries to the \( i \)-th subsystem.

Let \( F \) and \( J \) be cc-stateless random functions, with implementations \( A_F \) and \( A_J \), such that \( J \) implements a uniformly chosen function table from \( F \). For independent copies \( F_1, \ldots, F_m \) of \( F \), the following lemma bounds the advantage \( \Delta_{t,q}(C(F_1, \ldots, F_m), I) \).

Lemma 5.4. For all distinguishers \( D \in \mathcal{D}_{t,q} \) and for all \( k \in \mathbb{N} \), there exists an oracle distinguisher \( D' = D^{(\cdot)} \) such that for all measures \( M \) for \( F \) and \( N \) for \( J \), both with density at least \( 1 - \varepsilon \), and admitting respective error-less state samplers \( O_M \) and \( O_N \) for \( A_F \) and \( A_J \) (with lengths \( l_F \) and \( l_J \)),

\[
\Delta^D(C(F_1, \ldots, F_m), I) \leq m \cdot \Delta^{D^{O_M, O_N}}(F', J') + p(m, \tau, \varepsilon) + \delta(q) + 2^{-k},
\]

\(^6\)where, in the uniform case, we assume that \( \delta \) and \( \tau \) are efficiently computable functions.
Distinguisher $D^{'OM,ON}(S)$: 

\[ \begin{align*}
\mathcal{G} & := \emptyset \\
\text{for all } i = 1, \ldots, m \text{ do} \\
& (\sigma_i, z_i) \xleftarrow{\$} O_M \\
& \mathcal{G} := \mathcal{G} \cup \{i\} \text{ with probability } z_i \\
& i^* \xleftarrow{\$} \{1, \ldots, m\} \\
\text{for all } i = 1, \ldots, m \text{ do} \\
& \quad \text{if } i \in \mathcal{G} \text{ and } i < i^* \text{ then} \\
& \quad \quad \sigma'_i \xleftarrow{\$} \text{StateSample}_{ON}((k + \log(m))/\log(0.5)) \\
& \quad \quad S_i := A_F[\sigma'_i] \\
& \quad \text{else} \\
& \quad \quad S_i := A_J[\sigma_i] \\
& \quad \text{if } i^* \in \mathcal{G} \text{ then} \\
& \quad \quad \text{return } D(C(S_1, \ldots, S_{i^*-1}, S, S_{i^*+1}, \ldots, S_m)) \\
& \quad \text{else} \\
& \quad \quad \text{return } D(C(S_1, \ldots, S_m))
\end{align*} \]

Figure 5.2: The distinguisher $D^{'(\cdot, \cdot)}$ in the proof of Lemma 5.4.

where $F'$ and $J'$ have their function tables sampled according to $M$ and $N'$, respectively. Moreover, $D'$ has time complexity

\[ t' = t + t_{AC}(q) + O\left(\sum_{i=1}^{m} \max\{t_{AF}(q_i, l_F), t_{AJ}(q_i, l_J)\} + \right. \]

\[ \left. + \frac{m(k + \log(m))}{\log(1/\varepsilon)}\right), \]

and makes at most $\max_{i \in \{1, \ldots, m\}} q_i$ queries to the given system.

Proof. The distinguisher $D'$, given access to a system $S = I'$ or $S = J'$, operates as described in Figure 5.2. It starts by setting $\mathcal{G} := \emptyset$, issuing $m$ queries to $O_M$, obtaining pairs $(\sigma_i, z_i)$, and adds each $i \in \{1, \ldots, m\}$ to $\mathcal{G}$ with probability $z_i$. Furthermore, it picks an index $i^*$ uniformly from $\{1, \ldots, m\}$. Then, if $i^* \in \mathcal{G}$, it behaves as

$D(C(S_1, \ldots, S_{i^*-1}, S, S_{i^*+1}, \ldots, S_m))$

and otherwise just outputs $D(C(S_1, \ldots, S_m))$, where, for all $i \in \{1, \ldots, m\}$, we have $S_i := J'(J' \xleftarrow{\$} P_N \text{ if } i \in \mathcal{G} \text{ and } i < i^*, \text{ and } S_i := A_F[\sigma_i] \text{ otherwise.}$
In particular, \( J(J') \) is simulated by running \( A_J[\sigma'_i] \), where \( \sigma'_i \) is sampled using \texttt{StateSample}, as described in Example 3.5. Note that by the chosen parameter, this procedure outputs a state with the correct distribution, except with probability \( \frac{2^{-k}}{m} \), and hence the probability that it ever outputs a state with the wrong distribution is at most \( 2^{-k} \).

In the following, we assume that the outputs of \texttt{StateSample} always have the right distribution: This clearly gives an advantage which is at most \( 2^{-k} \) larger than the one with possible aborts. We use the shorthands

\[
P[D'(S) \mid g] := P[D'(S) = 1 \mid |G| = g] \\
P[D'(S) \mid g, i] := P[D'(S) = 1 \mid |G| = g \land i^* = i]
\]

to denote the conditional probability of \( D' \) outputting 1 when interacting with \( S \) given that \( |G| = g \in \{0, 1, \ldots, m\} \) (possibly additionally conditioned on \( i^* = i \)). Then, simple conditioning yields

\[
\Delta^{D'}(F', J') = |P[D'(F') = 1] - P[D'(J') = 1]| \\
= \left| \sum_{g=0}^{m} P_{|G|(g)} \cdot \left[ P[D'(F') \mid g] - P[D'(J') \mid g] \right] \right| \\
= \left| \sum_{g=0}^{m} P_{|G|(g)} \cdot \frac{1}{m} \sum_{i^* = 1}^{m} \left[ P[D'(F') \mid g, i^*] - P[D'(J') \mid g, i^*] \right] \right|.
\]

By construction, we also observe that

\[
P[D'(F') \mid g, i^*] = P[D'(J') \mid g, i^* - 1] \tag{5.2}
\]

for all \( g \in \{1, \ldots, m\} \) and \( i^* \in \{2, \ldots, m\} \), as in both cases the output is \( D'(C(S_1, \ldots, S_m)) \) where \( S_i \equiv J' \) for all \( i \in \mathcal{G} \cap \{1, \ldots, i^* - 1\} \), whereas \( S_i \equiv F' \) for all \( i \in \mathcal{G} \cap \{i^*, \ldots, m\} \). Using this in the above, we obtain

\[
\Delta^{D'}(F', J') = \frac{1}{m} \left| \sum_{g=0}^{m} P_{|G|(g)} \cdot \left( P[D'(F') \mid g, 1] - P[D'(J') \mid g, m] \right) \right| \\
\geq \frac{1}{m} \left| \sum_{g=0}^{m} P_{|G|(g)} \cdot P[D'(F') \mid g, 1] - P[D(I) = 1] \right| - \\
- \frac{1}{m} \left| P[D(I) = 1] - \sum_{g=0}^{m} P_{|G|(g)} \cdot P[D'(J') \mid g, m] \right|.
\]
On the one hand, we now remark that
\[ \sum_{g=0}^{m} P_{|G|}(g) \cdot P[D'(F') \mid g, 1] = P[D(C(F_1, \ldots, F_m)) = 1]. \]

On the other hand, because \( \mu(N) \geq 1 - \varepsilon \), whenever \( g \geq \tau \),
\[ |P[D'(J') \mid g, m] - P[D(I) = 1]| \leq \delta(q), \]
and therefore
\[
\frac{1}{m} \left| \sum_{g=0}^{m} P_{|G|}(g) \cdot P[D'(J') \mid g, m] - P[D(I) = 1] \right| \leq \\
\leq \sum_{g=0}^{\tau-1} P_{|G|}(g) + \sum_{g=\tau}^{m} P_{|G|}(g) \cdot |P[D'(J') \mid g, m] - P[D(I) = 1]| \\
\leq p(m, \tau, \varepsilon) + \delta(q),
\]
and hence
\[
\Delta_{D'}(F', J') \geq \frac{1}{m} \cdot \left[ \Delta_{D}(C(F_1, \ldots, F_m), I) - \delta(q) - p(\tau, m, \varepsilon) \right],
\]
which concludes the proof.

Remark 5.1. Both terms \( 2^{-k} \) (in the advantage) as well as \( m \cdot (k + \log(m)) \cdot \log \left( \frac{1}{\varepsilon} \right) \) (in the complexity \( t' \)) are superfluous in the non-uniform setting, as a sufficiently good state sampled according to \( P_N \) can be fixed, and we thus dispense with calls to \texttt{StateSample}.

By combining Lemma 5.4 and Remark 5.1 with the good measures \( M = M_F \) and \( N = M_J \) provided by Theorems 5.2 and 5.3, the following two theorems are straightforward.

**Theorem 5.5 (Non-Uniform Product Theorem).** For all integers \( t, q > 0 \) and for all \( \gamma \in (0, \frac{1}{2}] \), if
\[ \Delta_{t', q'}(F, J) \leq \varepsilon, \]
then,
\[ \Delta_{t, q}(C(F_1, \ldots, F_m), I) \leq 2m \cdot \gamma + p(\tau, m, \varepsilon) + \delta(q), \]
where \( q \) is maximal such that \( q_i \cdot \varphi_{he} \leq q' \) for all \( i = 1, \ldots, m \), and
\[ t' = \varphi_{he} \cdot \left[ t + t_{AC}(q) + O \left( \sum_{i=1}^{m} \max \{ t_F + t_{AF}(q_i, l_F), I_J + t_{AJ}(q_i, l_J) \} \right) \right] \]
for \( \psi := 7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3} \) and \( \varphi_{hc} \) as in Theorem 5.2, \( l_F := s_{A_F}(q \cdot \psi) \), and \( l_J := s_{A_J}(q \cdot \psi) \).

**Theorem 5.6** (Uniform Product Theorem). If \( 1 - \varepsilon \) is noticeable,
\[
\Delta_{\text{poly}}(F, J) \leq \varepsilon,
\]
and, in addition, \( \delta(q) \) is negligible for all polynomial \( q \), and \( F, J, \) and \( C(\cdot) \) are efficiently implementable, then,
\[
\Delta_{\text{poly}}(C(F_1, \ldots, F_m), I) \leq p(m, \tau, \varepsilon) + \nu
\]
for a negligible function \( \nu \).

In the following, we limit ourselves to discuss applications using the non-uniform product theorem, even though a uniform result can be obtained analogously by applying the uniform product theorem.

### 5.3. Security Amplification for PRGs

#### 5.3.1. Motivation

In Section 4.3, it was shown that if an \( \ell \)-bit random variable \( X \) satisfies \( \Delta_t(X, U_\ell) \leq \varepsilon \) for all polynomially bounded \( t \) and for a uniform \( \ell \)-bit string \( U_\ell \) (for instance, \( X \) may be the output \( G(U_k) \) of a weak PRG \( G : \{0, 1\}^k \rightarrow \{0, 1\}^\ell \)), then
\[
\Delta_t(X_1 \oplus \cdots \oplus X_m, U_\ell) \leq 2^{m-1} \cdot \varepsilon^m + \nu
\]
for all polynomially bounded \( t \) and a negligible function \( \nu \).

This approach suffers from two major disadvantages: First, computational indistinguishability amplification is inherently limited to the case \( \varepsilon < \frac{1}{2} \). For instance, the security of a weak PRG \( G : \{0, 1\}^k \rightarrow \{0, 1\}^\ell \) with a very large stretch \( \ell >> k \), and with one single constant output bit, is not amplified by the \( \oplus \)-operator, even if all other output bits are pseudorandom. Second, the construction is expanding only when \( \ell > k \cdot m \). Note that this issue cannot be overcome by extending the output size of the weak PRG, due to the security loss that would quickly yield an \( \varepsilon' \)-PRG with \( \varepsilon' \) close to one, violating the requirement that \( \varepsilon' < \frac{1}{2} \).

Using Theorem 5.5, we provide the first solution which amplifies the security of an \( \varepsilon \)-PRG \( G : \{0, 1\}^k \rightarrow \{0, 1\}^\ell \) for any \( \varepsilon < 1 \). First, we need to review some basic concepts on randomness extractors.
5.3.2. Randomness Extractors

A source \( S \) is a set of probability distributions, and an \( \varepsilon \)-extractor for \( S \) is an efficiently computable function \( \text{Ext} : \{0,1\}^m \times \{0,1\}^d \to \{0,1\}^n \) such that for a uniformly distributed \( d \)-bit string \( R \), we have \( d(\text{Ext}(X,R),U_n) \leq \varepsilon \) for all \( m \)-bit random variables \( X \) with \( P_X \in S \) and a uniformly distributed \( n \)-bit string \( U_n \). Furthermore, the extractor is called strong if the stronger condition \( d((\text{Ext}(X,R),R),(U_n,R))) \leq \varepsilon \) holds.

A two-parameter function \( h : \{0,1\}^m \times \{0,1\}^d \to \{0,1\}^n \) is called two-universal if \( P[h(x,K) = h(x',K)] = 2^{-n} \) for any two distinct \( m \)-bit \( x \) and \( x' \) and a uniform \( d \)-bit \( K \). An example with \( d = m \) is the function \( h(x,k) := (x \odot k) \mid_n \), where \( \odot \) is the multiplication of binary strings interpreted as elements of \( GF(2^m) \), and \( \mid_n \) outputs the first \( n \) bits of a given string. Two-universal hash functions are good extractors:

**Lemma 5.7 (Leftover Hash Lemma [BBR88, ILL89]).** For any \( \varepsilon > 0 \), every two-universal hash function \( h : \{0,1\}^m \times \{0,1\}^d \to \{0,1\}^n \) is a strong \( \varepsilon \)-extractor for the source of \( m \)-bit random variables with min-entropy at least \( n + 2 \log \left( \frac{1}{\varepsilon} \right) \).

We also note that extractors with smaller seed exist for the source of random variables with guaranteed min-entropy. We refer the reader to [Sha02] for a survey.

**Deterministic Extractors.** An extractor is deterministic if \( d = 0 \), i.e., no additional randomness is needed. (Note that such extractors are vacuously strong.) A class of sources allowing for deterministic extraction are so-called \((m,\ell,k)\)-total-entropy independent sources [KRVZ06], consisting of random variables of the form \((X_1,\ldots,X_m)\), where \( X_1,\ldots,X_m \) are independent \( \ell \)-bit strings, and the total min-entropy of \((X_1,\ldots,X_m)\) is at least \( k \).\(^7\) In particular, the following extractor from [KRVZ06] will be useful for our purposes. (Unconditional constructions requiring a higher entropy rate \( \delta \) are also given in [KRVZ06].)

**Theorem 5.8 ([KRVZ06]).** Under the assumption that primes with length in \([\tau,2\tau]\) can be found in time \( \text{poly}(\tau) \), there is a constant \( \eta \) such that for all \( m,\ell \in \mathbb{N} \) and all \( \delta > \zeta > (m\ell)^{-9} \), there exists a polynomial-time computable \( \varepsilon \)-extractor \( \text{Ext} : \{0,1\}^\ell \mid m \to \{0,1\}^n \) for \((m,\ell,\delta \cdot m\ell)\)-total-entropy independent sources, where \( n = (\delta - \zeta)m\ell \) and \( \varepsilon = e^{-\left(m\ell\right)^{\Omega(1)}} \).

\(^7\)Note that in this case \( H_\infty(X_1,\ldots,X_m) = \sum_{i=1}^m H_\infty(X_i) \).
5.3 Security Amplification for PRGs

5.3.3. The Concatenate-And-Extract Construction

This section presents, as an application of Theorem 5.5, the first construction achieving security amplification of arbitrarily weak PRGs.

Let \( \text{Ext} : \{0,1\}^{m\ell} \times \{0,1\}^{d} \rightarrow \{0,1\}^{n} \) be an efficiently computable function. The Concatenate-and-Extract (CaE) construction is a function \( \{0,1\}^{m\ell + d} \rightarrow \{0,1\}^{n + d} \) such that

\[
\text{CaE}(x_1 \parallel \cdots \parallel x_m \parallel r) := \text{Ext}(x_1 \parallel \cdots \parallel x_m, r) \parallel r.
\]

for all \( x_1, \ldots, x_m \in \{0,1\}^{\ell} \) and \( r \in \{0,1\}^{d} \).

Let \( S_{\ell,m,\varepsilon,\eta} \) be the set of random variables \( X_1 \parallel \cdots \parallel X_m \in \{0,1\}^{m\ell} \) where \( X_1, \ldots, X_m \in \{0,1\}^{\ell} \), and \( (1 - \varepsilon - \eta)m \) of the random variables \( X_i \) have min-entropy \( H_\infty(X_i) \geq \ell - \log((1 - \varepsilon) - 1) \). The following statement follows directly from the definition of an extractor.

Lemma 5.9. For all \( \varepsilon \in [0,1) \) and \( \eta \in [0,1 - \varepsilon] \), if \( \text{Ext} \) is a strong \( \delta \)-extractor for \( S_{\ell,m,\varepsilon,\eta} \), the construction \( \text{CaE}(\cdot, U_d) \) (for a uniform \( d \)-bit string \( U_d \)) is \( (\varepsilon, \tau, \delta) \)-extracting for the set of \( \ell \)-bit random variables and ideal system \( I := U_n \) for

\[
\delta(q) := \delta \quad \text{and} \quad \tau := (1 - \varepsilon - \eta)m.
\]

Note that the reason we look at \( \text{CaE} \) as deterministic, with its randomness \( U_d \) being an explicit parameter, is that we want to interpret \( \text{CaE} \) as an efficiently computable function to be used as a PRG. Using Theorem 5.5 (and Hoeffding’s inequality to upper bound \( p(m, m(1 - \varepsilon - \eta), \varepsilon) \)), we obtain the following corollary.

Corollary 5.10. Let \( X \) be an \( \ell \)-bit random variable, and let \( X_1, \ldots, X_m \) be independent copies of \( X \). For all integers \( t > 0 \), and for all \( \varepsilon \in [0,1) \), all \( \gamma \in (0, \frac{1}{2}] \), and all \( \eta \in (0, 1 - \varepsilon) \), if

\[
\Delta_t(X, U_t) \leq \varepsilon
\]

and \( \text{Ext} \) is a strong \( \delta \)-extractor (computable in time \( t_{\text{Ext}} \)) for \( S_{\ell,m,\varepsilon,\eta} \), then

\[
\Delta_t(\text{CaE}(X_1, \ldots, X_m, R), U_{n+d}) \leq 2m \cdot \gamma + 2 \cdot e^{-\eta \ell m} + \delta
\]

where \( t' = \phi_{HE}[t + t_{\text{Ext}} + O(m \cdot \ell)] \), and \( U_t, U_d, \) and \( U_{n+d} \) are uniform \( \ell, d, \) and \( (n + d) \)-bit strings, respectively.

If \( X = G(U_k) \) for an \( \varepsilon \)-PRG \( G : \{0,1\}^k \rightarrow \{0,1\}^\ell \), the resulting construction is expanding if \( n/mk > 1 \), and for an optimal extractor this
ratio is roughly \((1 - \varepsilon) \left[ \ell - \log \left( \frac{1}{1 - \varepsilon} \right) \right] / k\) (we ignore the entropy loss of the extractor for simplicity), or, turned around, our construction expands if the underlying \(\varepsilon\)-PRG satisfies \(\ell/k > \frac{1}{1 - \varepsilon} + \log \left( \frac{1}{1 - \varepsilon} \right) / k\). In particular, this value is independent of \(m\). In Section 5.3.4, we show that for a large class of natural constructions this is essentially optimal. For example, for \(\varepsilon = \frac{1}{2}\), the output length \(\ell\) of the given generator \(G\) needs to be slightly larger than \(2k\) in order to achieve expansion. For comparison, by XOR-ing, expansion is achieved if \(\ell/k > m\), where \(m = \omega(k/\log(1/\varepsilon)) = \omega(k)\) in order for the construction to be secure, and thus \(\ell = \omega(k^2)\) is required.

Also, the fact that all \(\ell\)-bit blocks are independent allows for using deterministic extractors in the \textbf{CaE} construction, such as the one given by Theorem 5.8, as long as \((1 - \varepsilon) \left[ 1 - \frac{1}{\ell} \log \left( \frac{1}{1 - \varepsilon} \right) \right] \) is bounded from below by \((m\ell)^{-\eta}\).

5.3.4. Optimality of the Output Length

We address the optimality of the output length of the \textbf{CaE} construction with respect to the following general class of constructions which operate by combining a number of independent outputs.

**Definition 5.2.** Let \(U_\ell\), \(U_d\), and \(U_h\) be uniformly distributed \(\ell\)-, \(d\)-, and \(h\)-bit random strings, respectively. A function \(C : (\{0,1\}^\ell)^m \times \{0,1\}^d \rightarrow \{0,1\}^h\) is a relativizing \((\varepsilon, \nu)\)-indistinguishability amplifier if for all polynomial-time oracle distinguishers \(D\) and for independent random variables \(X_1, \ldots, X_m\) with \(P_{X_i} = P_X\), we have

\[
\Delta^O_D (X, U_\ell) \leq \varepsilon
\]

for all polynomial-time oracle distinguishers \(D^{(1)}\), we have

\[
\Delta^O_D (C(X_1, \ldots, X_m, U_d), U_h) \leq \nu,
\]

for all polynomial-time oracle distinguishers \(D^{(1)}\) and for independent random variables \(X_1, \ldots, X_m\) with \(P_{X_i} = P_X\).

All constructions admitting a black-box security reduction (such as the \textbf{CaE} construction) are relativizing (cf. also [RTV04] for an overview of reduction types). The following theorem shows that the output length achieved by the \textbf{CaE} construction is essentially optimal for such constructions.

\footnote{Possibly with help of oracle \(O\).}
5.3 Security Amplification for PRGs

Theorem 5.11. For all $\varepsilon \in [0, 1)$ and all $\nu \in \left(0, \frac{1}{4}\right]$, all relativizing $(\varepsilon, \nu)$-indistinguishability amplifiers $C(\cdot) : \left(\{0, 1\}^\ell \times \{0, 1\}^d \rightarrow \{0, 1\}^h\right)$ satisfy

$$h \leq \left(m \cdot (1 - \varepsilon) \cdot \left[\ell - \log \left(\frac{1}{1 - \varepsilon}\right)\right] + d - 2\nu \log(2\nu)\right) \cdot (1 - 2\nu)^{-1}.$$ 

Proof. In the following, assume for simplicity that $\ell - \log \left(\frac{1}{1 - \varepsilon}\right)$ is an integer. Consider the $\ell$-bit random variable $X$ which, with probability $\varepsilon$, takes the value $1^\ell$, whereas with probability $1 - \varepsilon$, it takes as value an $\ell$-bit string $z \parallel 0^{\log \left(\frac{1}{1 - \varepsilon}\right)}$, where $z$ is a uniformly distributed $(\ell - \log(1/(1 - \varepsilon)))$-bit string. It is easy to verify that

$$\Delta D^O(X, U_\ell) \leq d(X, U_\ell) \leq \varepsilon,$$

for all distinguishers $D(\cdot)$ and oracles $O$. Furthermore,

$$H(X) = \varepsilon \cdot 0 + (1 - \varepsilon) \cdot \left[\ell - \log \left(\frac{1}{1 - \varepsilon}\right)\right],$$

which in turn implies

$$H(C(X_1, \ldots, X_m, U_d)) \leq H(X_1 \ldots X_m U_d) = m \cdot (1 - \varepsilon) \cdot \left[\ell - \log \left(\frac{1}{1 - \varepsilon}\right)\right] + d.$$

Assume now, towards a contradiction, that

$$h > (1 - \varepsilon) \cdot m \cdot \left[\ell - \log \left(\frac{1}{1 - \varepsilon}\right)\right] + d + 2\nu(h - \log(2\nu)),$$

then

$$H(U_h) - H(C(X_1, \ldots, X_m, U_d)) > 2\nu \cdot (h - \log(2\nu)),$$

and by Equation (2.3) (cf. Section 2.2.4), this implies

$$d(C(X_1, \ldots, X_m, U_d), U_h) > \nu.$$

To finish the proof, we note that if we let $O$ be an oracle allowing efficient solution of any $\mathcal{PSPACE}$-problem, then there exists a polynomial-time distinguisher $D^O$, which, on input $y \in \{0, 1\}^h$, uses the oracle to determine the probability that $C(X_1, \ldots, X_m, U_d) = y$, and outputs 1 if and only if this probability is smaller than $2^{-h}$. This distinguisher achieves exactly advantage $d(C(X_1, \ldots, X_m, U_d), U_h)$, contradicting the assumption on $C$. 

5.4. High-Min-Entropy Function Tables

5.4.1. Introduction

Let \( F \) be a set of functions and let \( F \equiv F(F) \) be a cc-stateless random function with function table \( F \in F \) such that
\[
H_\infty(F) \geq \log |F| - \delta
\]
for some \( \delta \geq 0 \). If we let \( F \) be the set of all functions \( \{0,1\}^n \rightarrow \{0,1\} \) and \( \delta \in \mathbb{N} \), an example is the function table \( F \) for which (when seen as an \( 2^n \)-bit string) the first \( 2^n - \delta \) bits are uniform and independent, whereas the remaining \( \delta \) bits are all zero. In the general case where nothing but a lower bound on the entropy of \( F \) is known, understanding the input-output behavior of \( F \) is a non-trivial task, as complex correlations among output values may occur.

In the following, we present a general and useful result for describing the input-output behavior of such a random function \( F \). It applies to different function sets \( F \), and generalizes an earlier lemma by Unruh [Unr07] for the special case where \( F \) is the set of all functions \( X \rightarrow Y \).

Our proof only relies on the weaker assumption that the Shannon entropy satisfies \( H(F) \geq \log |F| - \delta \), and we have not been able to prove a stronger statement under the assumption that the min-entropy is large, which remains an open problem.

5.4.2. Entropy Regular Function Sets

We start with some technical definitions characterizing a set \( F \) of functions \( X \rightarrow Y \).

**Definition 5.3.** A query \( x_i \in X \) is redundant for \( F \) and history \( (x_{i-1}, y_{i-1}) \in X^{i-1} \times Y^{i-1} \) if for all random variables \( F \) with range \( F \) such that
\[
P \left[ \forall j \in \{1, \ldots, i-1\} : F(x_j) = y_j \right] > 0,
\]
the value \( F(x_i) \) is fully determined by \( (x_{i-1}, y_{i-1}) \). Otherwise, the query is non-redundant. In particular, \( (x^q, y^{q-1}) \in X^q \times Y^{q-1} \) or \( (x^q, y^q) \in X^q \times Y^q \) is non-redundant if \( x_i \) is non-redundant for \( (x_{i-1}, y_{i-1}) \) for all \( i = 1, \ldots, q \).

**Example 5.1.** If \( F \) contains all functions \( X \rightarrow Y \), then of course \( x \) is redundant for \( (x_{i-1}, y_{i-1}) \) if and only if there exists \( j \in \{1, \ldots, i-1\} \) such that \( x_j = x \).
Example 5.2. Let $\mathcal{F}$ be the set of all mappings $q : \mathcal{X} \times \{+, -\} \to \mathcal{X}$ such that $q(\cdot, +)$ is a permutation $\mathcal{X} \to \mathcal{X}$, and $q(\cdot, -)$ its inverse. (Such functions model two-sided permutations.) Then, for all $(x^{i-1}, y^{i-1})$ the query $(x, +)$ is redundant if there exists $x_j = (x, +) \in \{x_1, \ldots, x_{i-1}\}$, or if there exists $j \in \{1, \ldots, i-1\}$ and $y$ such that $x_j = (y, -)$ and $y_j = x$.

Definition 5.4. The set $\mathcal{F}$ is (entropy) regular if both the following holds:

(i) There exists some value $N$ with the property that for all $q < N$ and non-redundant $(x^q, y^q) \in \mathcal{X}^q \times \mathcal{Y}^q$, there exists $x^q_{q+1}$ which is not redundant for $(x^q, y^q)$, whereas every $(x^q, y^q)$ for $q > N$ contains a redundant query.

(ii) For $J$ uniformly distributed on $\mathcal{F}$, there exists a function $h : \{1, \ldots, N\} \to \mathbb{R} \geq 0$ such that for all $1 \leq i \leq |\mathcal{X}|$, and all $i$-query non-redundant interactions $(x^i, y^{i-1}) \in \mathcal{X}^i \times \mathcal{Y}^{i-1}$ with

$$P \left[ \forall j \in \{1, \ldots, i-1\} : J(x_j) = y_j \right] > 0$$

we have,

$$P \left[ J(x_i) = y_i \mid J(x_1) = y_1, \ldots, J(x_{i-1}) = y_{i-1} \right] \in \left\{ 0, 2^{-h(i)} \right\}$$

for all $y_i \in \mathcal{Y}$.

Also, we let $H : \{1, \ldots, N\} \to \mathbb{R}_{\geq 0}$ be such that $H(q) := \sum_{i=1}^q h(i)$.

Example 5.3. The set of functions $\mathcal{X} \to \mathcal{Y}$ is regular with $h(i) := \log |\mathcal{Y}|$ for all $i$ and $N := |\mathcal{X}|$. Analogously, the set of permutations $\mathcal{X} \to \mathcal{X}$ and the set of two-sided permutations $\mathcal{X} \to \mathcal{X}$ are both regular with $h(i) := \log(|\mathcal{X}| - i + 1)$ and $N := |\mathcal{X}| - 1$.

Example 5.4. The simplest example of a function set which is not regular, for $\mathcal{X} = \mathcal{Y} = \{0, 1\}$, consists of the three functions $f_1, f_2, f_3$ with

$$f_1(0) = 0, \quad f_1(1) = 0,$$
$$f_2(0) = 0, \quad f_2(1) = 1,$$
$$f_3(0) = 1, \quad f_3(1) = 1.$$

Note that in particular Property (i) does not hold.

The following lemma will be useful in the following.
Lemma 5.12. Let $F$ be regular (with associated function $h$), and $F$ be a random variable on $\mathcal{F}$. Then, for all $i$, and all non-redundant $(x^i, y^{i-1}) \in \mathcal{X}^i \times \mathcal{Y}^{i-1}$,

$$H(F(x_i) | F(x_1) = y_1, \ldots, F(x_{i-1}) = y_{i-1}) \leq h(i).$$

Proof. Let $J$ be uniformly distributed on $\mathcal{F}$. If $P_{F(x_i) | F(x_1) = y_1, \ldots, F(x_{i-1}) = y_{i-1}}(J(x_i)) > 0$, then also $P_{J(x_i) | J(x_1) = y_1, \ldots, J(x_{i-1}) = y_{i-1}}(J(x_i)) > 0$. This implies that the number of values taken by $F(x_i)$ is at most $2^{h(i)}$.

5.4.3. Main Technical Lemma

Let us fix a regular set $F$ of functions $\mathcal{X} \to \mathcal{Y}$: For every user $U$ and for a cc-stateless random function $F$ with function table $F \in F$, we define the random function $F_U$ which is initialized by letting $U$ interact with $F$: If $(x_1, y_1), \ldots, (x_\Lambda, y_\Lambda)$ is the resulting interaction, then $F_U$ answers according to a uniformly chosen function $F' \in F$ constrained to $F'(x_i) = y_i$ for all $i \in \{1, \ldots, \Lambda\}$.

The following lemma states that if a cc-stateless random function behaves according to a function table with large Shannon entropy, than there always exists a good deterministic $U$ such that $F$ and $F_U$ are indistinguishable.

Lemma 5.13. Let $F$ be a regular set of functions $\mathcal{X} \to \mathcal{Y}$ (with associated parameter $N$), and let $F$ be a cc-stateless random functions with function table $F \in F$ such that $H(F) \geq \log(|F|) - \delta$. For all $q, \Lambda \in \mathbb{N}$ with $q + \Lambda \leq N$, there exists a deterministic user $U$ making at most $\Lambda$ queries such that

$$\Delta_q(F, F_U) \leq \left( \frac{q \delta}{\Lambda} \right)^{\frac{1}{2}}.$$

Proof. In the following, let $N$, $h$, and $H$ be as in Definition 5.4. For a deterministic user $U$ making $i$ non-redundant queries $X_1, \ldots, X_i \in \mathcal{X}$ to $F$, it is convenient to define the quantity

$$H^U(F) := H(Y_1, Y_2, \ldots, Y_i),$$

where $Y_1, Y_2, \ldots, Y_i \in \mathcal{Y}$ are the answers of these queries made by $U$ (in the order they are issued). As all queries are non-redundant, $i \leq N$. Furthermore, note that the quantity $H^U(F)$ only makes sense if $U$ is
deterministic, as otherwise additional randomness can be injected by \( U \) itself. Additionally, we define the function \( d : \{1, \ldots, N\} \to \mathbb{R} \) such that

\[
d(i) := \max_U \left[ H(i) - H^U(F) \right],
\]

where the maximum is taken over all \( U \)'s as above making exactly \( i \) non-redundant queries, i.e., \( d(i) \) measures the best deviation from the maximal possible entropy \( H(i) \) achieved by such an \( i \)-query \( U \). (Hence, \( d(i) \geq 0 \).) The following properties hold for \( d \).

**Claim 8.** The function \( d \) satisfies the following two properties:

(i) \( d(N) \leq \delta \).

(ii) \( d(i) \leq d(i + 1) \) for all \( i \in \{0, \ldots, N - 1\} \).

**Proof.** For (i), fix an arbitrary \( N \)-query deterministic \( U \) interacting with \( F \). Note that with knowledge of \( U \), given answers \( Y_1, \ldots, Y_N \) we can uniquely reconstruct the function table \( F \), since we can simulate an execution of \( U \) answering each query \( X_i \) with \( Y_i \) to obtain the corresponding queries \( X_1, \ldots, X_N \), and from these \( F \) is fully determined by the assumption that they are non-redundant. Also given \( F \), the output values \( Y_1, \ldots, Y_q \) are fully determined. Hence,

\[
H^U(F) = H(Y_1, Y_2, \ldots, Y_N) = H(F) \geq \log |F| - \delta = H(N) - \delta,
\]

which implies \( d(N) \leq \delta \).

To prove (ii), fix some \( i > 0 \), and let \( U \) be such that it makes \( i \) queries \( X_1, \ldots, X_i \), and maximizes \( H(i) - H^U(F) \). Then, for any \( U' \) which acts as \( U \) for its first \( i \) queries, and makes an additional non-redundant query \( X_{i+1} \notin \{X_1, \ldots, X_i\} \) we have (with \( Y_1, \ldots, Y_{i+1} \in \mathcal{Y} \) being the corresponding answers)

\[
d(i + 1) \geq H(i + 1) - H(Y_1, \ldots, Y_i, Y_{i+1})
= \underbrace{H(i) - H(Y_1, \ldots, Y_i)}_{= d(i)} + h(i + 1) - H(Y_{i+1} | Y_1, \ldots, Y_i),
\]

since \( H(Y_{i+1} | Y_1, \ldots, Y_i) \leq h(i) \) by Lemma 5.12. \( \square \)

The following claim shows that since \( d \) is monotone and grows overall by at most \( \delta \) on its domain, then there must be a portion of length \( q \) in \( \{1, \ldots, \Lambda + q\} \) where \( d \) is almost flat.
Claim 9. There exists \( i^* \in \{0, \ldots, \Lambda \} \) such that

\[
d(i^* + q) - d(i^*) \leq \frac{q\delta}{\Lambda}.
\]

Proof. Define \( \lambda := \left\lfloor \frac{\Lambda}{q} \right\rfloor \) as well as the set \( S := \{i \cdot q : i = 0, \ldots, \lambda \} \subseteq \{0, \ldots, \Lambda\} \). Clearly, \( d(\lambda \cdot q + q) \leq d(N) \leq \delta \), and hence there must be an \( i^* \) in \( S \) such that \( d(i^* + q) - d(i^*) \leq \frac{\delta}{\lambda + 1} \), as otherwise this would contradict \( d(\lambda q + q) \leq \delta \). The claim follow by the fact that \( \lambda + 1 \geq \frac{\Lambda}{q} \).

In the following, let \( i^* \) be as guaranteed to exist by Claim 9, and take \( \bar{U} \) maximizing \( H(i^* \cdot q) - H(U(Y_1, \ldots, Y_{i^*})) \). Consider the random function \( F \rightarrow J \) which behaves as \( F \) for the first \( i^* \) queries, but then behaves as a randomly chosen function \( J \in \mathcal{F} \) consistent with the input and outputs of the first \( i^* \) queries. Additionally, let \( D \) be an arbitrary \( q \)-query deterministic distinguisher issuing non-redundant queries: We define \( \bar{U} \rightarrow D \) to be the the distinguisher making \( i^* + q \) non-redundant queries, which runs \( \bar{U} \) and then runs \( D \), but does not repeat queries for which the answers is known due to a query of \( \bar{U} \). In particular, it possibly asks extra (dummy) queries in order to ensure that \( i^* + q \) queries are always asked.

Claim 10. For all deterministic \( q \)-query distinguishers \( D \),

\[
\Delta^D(F, F_{\bar{U}}) \leq d\left( P^{(1)}_{Y_{i^*+q}}, P^{(2)}_{Y_{i^*+q}} \right),
\]

where \( P^{(1)}_{Y_{i^*+q}} \) and \( P^{(2)}_{Y_{i^*+q}}(F \rightarrow J) \) are the distributions of the answers of the queries of \( \bar{U} \rightarrow D \) when interacting with the systems \( F \) and \( F \rightarrow J \), respectively.

Proof. Given the value of \( Y_{i^*+q} \), one possible strategy to distinguish samples from \( P^{(1)}_{Y_{i^*+q}(F \rightarrow J)} \) and \( P^{(2)}_{Y_{i^*+q}(F \rightarrow J)} \) is to mimic the behavior of \( D \) in the corresponding interaction with \( F \) and \( F \rightarrow J \), respectively.

Using the shorthands \( P^{(1)} \) and \( P^{(2)} \) for \( P^{(1)}_{\bar{U} \rightarrow D} \) and \( P^{(2)}_{\bar{U} \rightarrow D}(F \rightarrow J) \), respectively, as well as

\[
Y_{i^*} := [Y_1, \ldots, Y_{i^*}] \text{ and } Y^q := [Y_{i^*+1}, \ldots, Y_q],
\]
we have

\[ D \left( P_{Y^*}^{(1)} \| P_{Y^*}^{(2)} \right) = \sum_{y^*+q} P_{Y^*}^{(1)}(y^*+q) \cdot \log \left( \frac{P_{Y^*}^{(1)}(y^*+q)}{P_{Y^*}^{(2)}(y^*+q)} \right) \]

\[ = \sum_{y^*} P_{Y^*}(y^*) \cdot \sum_{y^*} P_{Y^*}(y^*, y^*) \cdot \log \left( \frac{P_{Y^*}(y^*)}{P_{Y^*}(y^*, y^*)} \right) \]

where we have used the fact that \( P_{Y^*}^{(1)} = P_{Y^*}^{(2)} \), since the first \( i^* \) queries are answered by \( F \) in both cases. Furthermore, we have

\[ -\log(P_{Y^*|Y^*}(y^*, y^*)) = \sum_{j=i^*+1}^{i^*+q} h(j) = H(i^* + q) - H(i^*) \]

by the regularity property of \( F \). Also note that

\[ \sum_{y^*} P_{Y^*}(y^*) \cdot \sum_{y^*} P_{Y^*}(y^*, y^*) \log \left( P_{Y^*}(y^*) \right) = -H(Y^*|Y^*) \]

where the Shannon entropy is computed in the first experiment: In particular, using the fact that

\[ H(Y^*|Y^*) = H(Y^{i^*+q}) - H(Y^*) \]

and replacing everything into the above,

\[ D \left( P_{Y^*+q}^{(1)} \| P_{Y^*+q}^{(2)} \right) = H(i^* + q) - H(i^*) - H(Y^*|Y^*) \]

\[ = H(i^* + q) - H(Y^{i^*+q}) - (H(i^*) - H(Y^*)) \]

\[ \leq d(i^* + q) - d(i^*) \leq \frac{q\delta}{\Lambda}. \]

Pinsker’s inequality (cf. Equation (2.2) in Section 2.2.4), combined with Claim 10, implies that \( \Delta D(F, F_U) \leq \sqrt{\frac{q\delta}{\Lambda}} \) for all deterministic \( q \)-query \( D \), and the final bound follows since a randomized distinguisher never outperforms a deterministic one. \( \square \)

### 5.5 Cascade of Weak Permutations

In this section, we show that the cascade \( \varepsilon \)-PRPs on a set \( X \) is security amplifying for essentially all \( \varepsilon < 1 - \frac{1}{|X|} \). We note that no better result
can be expected, as a \((1 - \frac{1}{N^2})\)-PRP with a fixed point can be given, and the cascade preserves such a fixed point.\(^9\)

The core of the result consists of proving that the cascade of cc-stateless random permutations is an \((\varepsilon, 2, \delta)\)-extracting construction, for all \(\varepsilon\) and a sufficiently small \(\delta\).

### 5.5.1. Cascades as Extracting Constructions

Let \(Q_1\) and \(Q_2\) be two independent cc-stateless random permutations on the set \(X\) (with \(N := |X|\)) with the property that the min-entropies of their respective function tables \(Q_1\) and \(Q_2\) satisfy \(H_\infty(Q_1) \geq \log(N!) - \log \left(\left(1 - \varepsilon\right)^{-1}\right)\) and \(H_\infty(Q_2) \geq \log(N!) - \log \left(\left(1 - \varepsilon\right)^{-1}\right)\) for some \(\varepsilon \in [0, 1 - \frac{1}{N}]\). We prove that the cascade \(Q_1 \triangleright Q_2\) is indistinguishable from a URP \(P\) for computationally unbounded distinguishers, both in the one- and in the two-sided cases.

**Lemma 5.14.** For all \(q, \Lambda \geq 1\),

\[
\Delta_q(\langle Q_1 \triangleright Q_2 \rangle, (P)) \leq \frac{4q\Lambda}{N} + \frac{2\Lambda(q + \Lambda)}{(1 - \varepsilon)N} + 2\left(\frac{q\log \left(\left(1 - \varepsilon\right)^{-1}\right)}{\Lambda}\right)^{\frac{3}{2}}.
\]

Furthermore, \(\Delta_q(\langle Q_1 \triangleright Q_2 \rangle, (P)) \leq \Delta_q(\langle Q_1 \triangleright Q_2 \rangle, (P))\)

Before we proceed to a proof of Lemma 5.14, we remark that the same bound applies to any cascade \(Q'_1 \triangleright \cdots \triangleright Q'_m\) of independent cc-stateless random permutations with \(i < j\) such that \(Q'_i \equiv Q_1\) and \(Q'_j \equiv Q_2\), as it can be seen as the cascade of two permutations \(Q_1 := Q'_1 \triangleright \cdots \triangleright Q'_i\) and \(Q_2 := Q'_{i+1} \triangleright \cdots \triangleright Q'_m\) with the same min-entropy guarantees on their function tables. This yields the following corollary.

**Corollary 5.15.** For all \(\varepsilon \in [0, 1]\), \(\Lambda \geq 1\) and a finite set \(X\) with \(|X| = N\), the cascade of \(m\) random permutations on \(X\) is \((\varepsilon, 2, \delta)\)-extracting for the set of permutations \(X \rightarrow X\) and a URP \(P : X \rightarrow X\), where

\[
\delta(q) := \frac{4q\Lambda}{N} + \frac{2\Lambda(q + \Lambda)}{(1 - \varepsilon)N} + 2\left(\frac{q\log \left(\left(1 - \varepsilon\right)^{-1}\right)}{\Lambda}\right)^{\frac{3}{2}}.
\]

Furthermore, the statement also holds in the two-sided case, with a two-sided URP \(\langle P \rangle\) as the associated ideal system.

\(^9\)This restriction is irrelevant, since in order for the reduction to be efficient, we need \(\varepsilon < 1 - \frac{1}{\text{poly}}\), and \(|X|\) is super-polynomial in the security parameter.
Note that we are allowed free choice of $\Lambda$: For our purposes, it suffices to set $\Lambda := (\log N)^{\zeta}$ (for a slowly growing $\zeta = \omega(1)$, where $\log N$ is the security parameter) to achieve indistinguishability for any polynomial number of queries $q$ and any $\varepsilon \leq 1 - \frac{(\log N)^{3\zeta}}{N}$.

\textbf{Proof (Of Lemma 5.14).} Throughout this section, we let $N := |X|$ and for an integer $N$, we define $N^{(i)} := N \cdot (N - 1) \cdots (N - i + 1)$. In particular, $N^{(N)} = N!$. We also fix parameters $\varepsilon$ and $\Lambda$.

For every user $\mathbf{U}$ issuing $\Lambda$ queries and for a cc-stateless two-sided random permutation $\langle Q \rangle$ with domain $X$, we define the two-sided random permutation $\langle Q \rangle_{\mathbf{U}}$ with domain $X$ which is initialized by letting $\mathbf{U}$ interact with $Q$: If $(x_1, y_1), \ldots, (x_\Lambda, y_\Lambda)$ are the resulting input-output pairs, i.e., for all $i$ either a forward query $(x_i, +)$ returned $y_i$ or a backward query $(y_i, -)$ returned $x_i$, then $\langle Q \rangle_{\mathbf{U}}$ answers backward and forward queries according to a randomly chosen permutation $Q$ constrained to $Q(x_i) = y_i$ for all $i \in \{1, \ldots, \Lambda\}$. Lemma 5.13 implies that there exist users $\mathbf{U}_1$ and $\mathbf{U}_2$ such that for $i = 1, 2$,

$$\Delta_q(\langle Q_i \rangle, \langle Q_i \rangle_{\mathbf{U}_i}) \leq \left(\frac{q \delta}{\Lambda}\right)^{\frac{1}{2}}.$$ 

Let $\langle Q'_i \rangle := \langle \langle Q_i \rangle_{\mathbf{U}_i} \rangle$ for $i = 1, 2$ be the two corresponding systems.

\textbf{The Hybrid Systems $\langle H \rangle$.} We introduce a hybrid two-sided (stateful) random permutation $\langle H \rangle$, which simulates the cascade $\langle Q'_1 \triangleright Q'_2 \rangle$, with some differences. The two-sided permutation $\langle H \rangle$ is initialized by running $\mathbf{U}_1$ and $\mathbf{U}_2$ on $Q_1$ and $Q_2$, respectively: Let

$$\mathcal{T}^{(1)} := \left\{ (x_1^{(1)}, y_1^{(1)}), \ldots, (x_\Lambda^{(1)}, y_\Lambda^{(1)}) \right\}$$
$$\mathcal{T}^{(2)} := \left\{ (y_1^{(2)}, z_1^{(2)}), \ldots, (y_\Lambda^{(2)}, z_\Lambda^{(2)}) \right\}$$

be the resulting input-output pairs. We also use the shorthands

$$\mathcal{X}^{(1)} := \{x_1^{(1)}, \ldots, x_\Lambda^{(1)}\}, \quad \mathcal{Y}^{(1)} := \{y_1^{(1)}, \ldots, y_\Lambda^{(1)}\},$$
$$\mathcal{Y}^{(2)} := \{y_1^{(2)}, \ldots, y_\Lambda^{(2)}\}, \quad \mathcal{Z}^{(2)} := \{z_1^{(2)}, \ldots, z_\Lambda^{(2)}\}.$$ 

Additionally $\langle H \rangle$ maintains a set of triples $Q$ (which is initially empty). A triple $(x, y, z) \in Q$ means that the first permutation in the cascade simulated by $\langle H \rangle$ maps $x$ to $y$, whereas the second one maps $y$ to $z$. For notational convenience, we let $Q_1$ be the set of elements $x$ which appear
as first component of a triple in $Q$. Similarly, we define $Q_2$ and $Q_3$ as the projections on the second and third components, respectively.

Also, at every point in time in the interaction, let $\tilde{T}^{(1)} \subseteq T^{(1)}$ be the subset of pairs which do not share a component with $Q_1$ and $Q_2$, and let $\tilde{T}^{(2)}$ be defined analogously. Also, let $\tilde{X}^{(1)}$, $\tilde{Y}^{(1)}$, $\tilde{Y}^{(2)}$ and $\tilde{Z}^{(2)}$ be the associated projections.

On input $(x, +)$ with $x \notin Q_1$, a new triple $(x, y, z)$ with $y \notin Q_2$ and $z \notin Q_3$ is added as follows:

(A) If there exists $(x, y') \in \tilde{T}^{(1)}$, then we set $y := y'$. Furthermore, if there exists $(y, z') \in \tilde{T}^{(2)}$, then we set $z := z'$. Otherwise we choose $z \leftarrow X \setminus Q_3$.

(B) Otherwise, we set $y \leftarrow X \setminus Q_2$ (i.e., the set of all values which have not been used yet). Furthermore, if some $(y, z') \in \tilde{T}^{(2)}$, then $z := z'$. Otherwise, we set $z \leftarrow X \setminus (Q_3 \cup \tilde{Z}^{(2)})$.

Backward queries $(y, -)$ are answered symmetrically.

Bounding $\Delta_q(\langle Q'_1 \triangleright Q'_2 \rangle, (H))$. The only difference between $\langle H \rangle$ and $\langle Q'_1 \triangleright Q'_2 \rangle$ occurs in (A) when sampling $z \leftarrow X \setminus Q_3$ rather than $z \leftarrow X \setminus (Q_3 \cup \tilde{Z}^{(2)})$ and in (B) when $y \leftarrow X \setminus Q_2$ rather than $y \leftarrow X \setminus (Q_2 \cup \tilde{Y}^{(1)})$. Also, the symmetrical statement holds for backward queries.

Consequently, we define an MBO $A_1, A_2, \ldots$ on $\langle H \rangle$ (obtaining $\tilde{H}$) which turns to one as soon as an element $z \in \tilde{Z}^{(2)}$, or $y \in \tilde{Y}^{(1)}$ (in a forward query), or $x \in \tilde{X}^{(1)}$, or $y \in \tilde{Y}^{(2)}$ (in a backward query) is sampled. Then, conditioned on the MBO not turning one, the system $\langle \tilde{H} \rangle$ behaves as $\langle Q'_1 \triangleright Q'_2 \rangle$, i.e., and by Corollary 2.7,

$$\Delta_q(\langle H \rangle, \langle Q'_1 \triangleright Q'_2 \rangle) \leq \nu_q(\langle H \rangle).$$

To upper bound $\nu_q(\langle H \rangle)$, we think of the four assignments of interest to be equivalently executed as follows: Initially, three independent lists $x'_1, \ldots, x'_p$, $y'_1, \ldots, y'_p$, and $z'_1, \ldots, z'_p$, consisting each of uniform independent random distinct elements of $X$, are sampled. Whenever we have to assign a value $x'$, $y'$, or $z'$ in one of the four cases of interest, we pick the first element from the corresponding list, and delete it from the list. Also, whenever $\langle H \rangle$ adds a triple $(x', y', z')$ to $Q$, the values $x'$, $y'$, and $z'$ are removed from the corresponding lists, if they appear in them, after the query is answered.
The probability that \( A_q = 1 \) occurs is upper bounded by the probability that one of \( \mathcal{X}^{(1)} \cap \{ x_1', \ldots, x_q' \}, \mathcal{Y}^{(1)} \cap \{ y_1', \ldots, y_q' \}, \mathcal{Y}^{(2)} \cap \{ y_1', \ldots, y_q' \}, \) or \( \mathcal{Z}^{(2)} \cap \{ z_1', \ldots, z_q' \} \) is non-empty. As every element of the three lists is, individually, uniformly distributed, the union bound yields

\[
\Delta_q(\langle \mathcal{H} \rangle, (Q'_1 \triangleright Q'_2)) \leq \nu_q(\hat{\langle \mathcal{H} \rangle}) \leq \frac{4q\Lambda}{N}.
\]

**Bounding \( \Delta_q(\langle \mathcal{H} \rangle, (P)) \).** In the following, we consider the sets \( \mathcal{X}^{(1)}_\text{in} \subseteq \mathcal{X}^{(1)} \) and \( \mathcal{Y}^{(1)}_\text{in} \subseteq \mathcal{Y}^{(1)} \) of inputs of forward and backward queries of \( U_1 \), respectively. Analogously, we define \( \mathcal{Y}^{(2)}_\text{in} \) and \( \mathcal{Z}^{(2)}_\text{in} \). Furthermore, we let

\[
\mathcal{X}^{(1)}_\text{out} := \mathcal{X}^{(1)} \setminus \mathcal{X}^{(1)}_\text{in}, \quad \mathcal{Y}^{(1)}_\text{out} := \mathcal{Y}^{(1)} \setminus \mathcal{Y}^{(1)}_\text{in}, \quad \mathcal{Y}^{(2)}_\text{out} := \mathcal{Y}^{(2)} \setminus \mathcal{Y}^{(2)}_\text{in}, \quad \mathcal{Z}^{(2)}_\text{out} := \mathcal{Z}^{(2)} \setminus \mathcal{Z}^{(2)}_\text{in}.
\]

Then, we define the MBO \( B_1, B_2, \ldots \) on \( \langle \mathcal{H} \rangle \) (obtaining \( \hat{\langle \mathcal{H} \rangle} \)) such that \( B_i = 1 \) if at initialization \( \mathcal{Y}^{(1)}_\text{out} \cap \mathcal{Y}^{(2)} \neq \emptyset \), or \( \mathcal{Y}^{(2)}_\text{out} \cap \mathcal{Y}^{(1)} \neq \emptyset \) occur, or one of the following is true:

- A forward query \( X_j = (x,+) \) (with \( j \in \{1,\ldots,i\} \)) with \( x \in \mathcal{X}^{(1)}_\text{out} \) is issued;

- A backward query \( X_j = (z,-) \) (with \( j \in \{1,\ldots,i\} \)) with \( z \in \mathcal{Z}^{(2)}_\text{out} \) is issued.

Assume that the \( i \)-th query is a forward query \( (x,+) \). (For a backward query, the argument is symmetric.) Then, given that \( B_i = 0 \), we consider two cases:

- If (A) occurs, then the fact that \( B_i = 0 \) yields that \( x = x^{(1)}_j \in \mathcal{X}^{(1)}_\text{in} \) (for some \( j \in \{1,\ldots,\Lambda\} \)), and that \( y^{(1)}_j \notin \mathcal{Y}^{(2)} \), which in turn implies \( y^{(1)}_j \notin \hat{\mathcal{Y}}^{(2)} \). Thus a random value from \( \mathcal{X} \setminus Q_3 \) is returned.

- Otherwise, if (B) occurs, consider a particular value \( z \in \mathcal{X} \setminus Q_3 \). If \( z = z^{(2)}_j \in \hat{\mathcal{Z}}^{(2)} \) (for some \( j \in \{1,\ldots,\Lambda\} \)), the probability that this value is returned is \( \frac{1}{|\mathcal{X} \setminus Q_3|} = \frac{1}{|\mathcal{X} \setminus Q_3|} \), i.e., the probability that \( y^{(2)}_j \in \hat{\mathcal{Y}}^{(2)} \) is chosen.
On the other hand, if $z \notin \overline{\mathcal{Z}}^{(2)}$, the probability that $z$ is returned equals

$$\left| \mathcal{X} \setminus (\mathcal{Q}_2 \cup \overline{\mathcal{Y}}^{(2)}) \right| \left| \mathcal{X} \setminus \mathcal{Q}_2 \right| = \left| \mathcal{X} \setminus (\mathcal{Q}_3 \cup \overline{\mathcal{Z}}^{(2)}) \right| = 1 - \frac{1}{|\mathcal{X} \setminus \mathcal{Q}_3|},$$

since $|\mathcal{Q}_2| = |\mathcal{Q}_3|$, $\overline{\mathcal{Y}}^{(2)} \cap \mathcal{Q}_2 = \emptyset$, $\overline{\mathcal{Z}}^{(2)} \cap \mathcal{Q}_3 = \emptyset$, and $|\overline{\mathcal{Y}}^{(2)}| = |\overline{\mathcal{Z}}^{(2)}|$. We have just shown that $\widetilde{(H)}$ behaves as $\langle P \rangle$ as long as $B_i = 0$ (even if $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ are known), and Corollary 2.7 yields

$$\Delta_q(\langle H \rangle, \langle P \rangle) \leq \nu_q(\widetilde{(H)}).$$

Also, with the state $S$ consisting of a description of $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$, note that $B_i$ only depends on $S$ and $X^i, Y^{i-1}$, and the conditions for applying Lemma 2.9 are fulfilled with $T$ being a URP.$^{10}$ Given input-output pairs $(x_1, z_1), \ldots, (x_q, z_q)$, we sample

$$(X_1^{(1)}, Y_1^{(1)}), \ldots, (X^\Lambda_1, Y^\Lambda_1) \text{ and } (Y_1^{(2)}, Z_1^{(2)}), \ldots, (Y^\Lambda_2, Z^\Lambda_2)$$

by letting $\overline{U}_1$ and $\overline{U}_2$ interact with $Q_1$ and $Q_2$ independently, and upper bound the probability that $B_q = 1$ by $P[E_1 \lor E_2] \leq P[E_1] + P[E_2]$, where $E_1$ is the event that any of the outputs of the queries of $\overline{U}_1$ is in $\mathcal{Y}^{(2)} \cup \mathcal{X}^q$ and $E_2$ is the event that any of the outputs of the queries of $\overline{U}_2$ is in $\mathcal{Z}^q \cup \mathcal{Y}^{(1)}$. Note that the sequence of outputs of queries of $\overline{U}_1$ and $\overline{U}_2$ have both min-entropy at least $\log (N^{(\Lambda)}) - \log ((1 - \varepsilon)^{-1})$, i.e., each such sequence occurs with probability at most $\frac{1}{(1 - \varepsilon)^N}$. On the other hand, since $|\mathcal{X}^i \cup \mathcal{Y}^{(2)}| \leq q + \Lambda$ and $|\mathcal{Z}^i \cup \mathcal{Y}^{(1)}| \leq q + \Lambda$, there are (by a union bound) at most $\Lambda(q + \Lambda) \cdot (N - 1)^{(\Lambda - 1)}$ sequences provoking $E_1$ and $E_2$. Thus, we conclude that

$$\nu_q(\widetilde{(H)}) \leq 2 \cdot \frac{\Lambda(q + \Lambda)}{(1 - \varepsilon)N}.$$
**5.5 Cascade of Weak Permutations**

WRAPPING UP. We can now collect the previous bounds:

\[
\Delta_q((Q_1 \triangleright Q_2), (P)) \leq \Delta_q((Q_1 \triangleright Q_2), (Q_1 \triangleright Q_2))
\]

\[
+ \Delta_q((Q_1 \triangleright Q_2), (Q'_1 \triangleright Q'_2)) + \Delta_q((Q'_1 \triangleright Q'_2), (P))
\]

\[
\leq \Delta_q((Q_1), (Q'_1)) + \Delta_q((Q_2), (Q'_2)) + \Delta_q((Q'_1 \triangleright Q'_2), (P))
\]

\[
\leq 2 \cdot \left( \frac{q \log((1-\varepsilon)^{-1})}{A} \right) + \Delta_q((Q'_1 \triangleright Q'_2), (P)),
\]

where we used the fact that removing the first (or the second) permutation only makes distinguishing easier. Furthermore,

\[
\Delta_q((Q'_1 \triangleright Q'_2), (P)) \leq \Delta_q((Q'_1 \triangleright Q'_2), (H)) + \Delta_q((H), (P))
\]

\[
\leq \frac{4qA}{N} + \frac{2A(q + \Lambda)}{(1-\varepsilon)N},
\]

which concludes the proof.

\[\square\]

**5.5.2. Main Security Statement**

Let \(Q\) be a cc-stateless random permutation with domain \(X'\) (for \(N := |X'| = 2^n\), where \(n\) is the security parameter) such that \(\langle Q \rangle\) is implemented by the algorithm \(A_{\langle Q \rangle}\) with time complexity \(t_{A_{\langle Q \rangle}}\) and space complexity \(s_{A_{\langle Q \rangle}}\). We also consider the canonical (efficient) implementation of a two-sided URP \(\langle P \rangle\) discussed in Section 2.5.2, i.e., each query is answered in time logarithmic in the number of table entries, and after \(q\) queries, the table has \(q\) entries and size \(s = O(q \cdot n)\). Moreover, we let \(Q_1, \ldots, Q_n\) be independent instances of \(Q\).

We could now simply apply Theorem 5.5 with Corollary 5.15 to prove security amplification, and this is indeed the best we can obtain for the uniform setting. Yet, for the non-uniform case, we are able to prove the following result, which is slightly stronger.

**Theorem 5.16.** For all integers \(t, q > 0\), for all \(\varepsilon \in [0, 1)\) and \(\gamma \in \left(0, \frac{1}{2}\right]\), if

\[
\Delta_{t,q'}((Q), (P)) \leq \varepsilon,
\]

then, for all \(\Lambda > 0\),

\[
\Delta_{t,q}((Q_1 \triangleright \cdots \triangleright Q_m), (P)) \leq (m - (m - 1)\varepsilon) \cdot \varepsilon^m + \frac{4q\Lambda}{N} + \frac{2A(q + \Lambda)}{(1-\varepsilon)N}
\]

\[
+ 2 \left( \frac{q \log((1-\varepsilon)^{-1})}{A} \right) + (2m + 2)\gamma
\]
where

\[ t' = \varphi_{\text{hc}} \cdot \left[ t + O \left( m \cdot \max \left\{ l + t_{A_{(q)}}(q, l), q \cdot (\psi n + \log(q \cdot (\psi + 1))) \right\} \right) \right] \]

and \( q' = \varphi_{\text{hc}} \cdot q \), for \( q := 7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3} \) and \( \varphi_{\text{hc}} \) is as in Theorem 5.2, and \( l := s_{A_{(q)}}(q \cdot \psi) \).

The same bounds apply to the single-sided case, and we postpone a proof of tightness to Section 5.5.3.

The proof of Theorem 5.16, given below, follows the same high-level approach as Theorem 5.5: It exploits the fact that with very high probability at least two permutations in the cascade are “good”, i.e., they are computationally indistinguishable from random permutations with high entropy. In this case, the extracting property of the cascade implies that the real system is close to a URP. However, in order to improve the bound even further, we consider what happens in the case that at most one permutation is good: While Theorem 5.5 bounds the advantage by one in this case, this can be improved. For example, when exactly one permutation is good, the entire cascade is computationally indistinguishable from a permutation with min-entropy \( \log(N!) - \log((1 - \varepsilon)^{-1}) \), and it is easy to show that such a permutation can be distinguished with advantage at most \( \varepsilon \) from URP. We are unaware of how this technique can be applied to the uniform setting.

**Proof.** Theorem 5.2 implies that we can define (two-sided) random permutations \( \langle Q' \rangle, \langle Q'' \rangle \), and \( \langle P' \rangle \) such that the following three properties hold for some \( p \leq \varepsilon \): (i) The function table of \( \langle P' \rangle \) has min-entropy at least \( \log(N!) - \log((1 - \varepsilon)^{-1}) \), (ii) \( \langle Q \rangle \) behaves as \( \langle Q' \rangle \) with probability \( 1 - p \) and as \( \langle Q'' \rangle \) with probability \( p \), and (iii) \( \Delta_{v, \varphi_{\text{hc}}}(\langle Q' \rangle, \langle P' \rangle) \leq 2\gamma \) for \( t'' = t'/\varphi_{\text{hc}} \) and \( q'' = q'/\varphi_{\text{hc}} = q \). Furthermore, \( \langle Q' \rangle \) and \( \langle Q'' \rangle \) can both be perfectly implemented using \( A_{(q)} \) initialized with some appropriately distributed state of length at most \( s_{A_{(q)}}(q \cdot \psi) \) given as advice. Similarly, \( \langle P' \rangle \) can be simulated by running the above algorithm initialized with an appropriate state of length \( O(q \cdot \psi \cdot n) \). (See the discussion in Section 3.2.2.)

Additionally, for all \( \mathcal{I} \subseteq \{1, \ldots, m\} \), let \( A_i \) be the event that \( \langle Q_i \rangle \) behaves as \( \langle Q \rangle \) for all \( i \in \mathcal{I} \) whereas \( \langle Q_i \rangle \) behaves as \( \langle Q'' \rangle \) for all \( i \notin \mathcal{I} \). Likewise, for independent instances \( \langle Q' \rangle \) and \( \langle Q'' \rangle \) (for \( i = 1, \ldots, m \)) of \( \langle Q' \rangle \) and \( \langle Q'' \rangle \), respectively, let \( Q_I := S_1 \vartriangleright \cdots \vartriangleright S_m \), where \( S_i := Q'_i \) for all \( i \in \mathcal{I} \) and \( S_i := Q''_i \) for all \( i \notin \mathcal{I} \).

We now fix some distinguisher \( D \in D_{t, q} \), and observe that

\[
\delta^D((Q_1 \vartriangleright \cdots \vartriangleright Q_m), \langle P \rangle) = \sum_{\mathcal{I} \subseteq \{1, \ldots, m\}} q_{\mathcal{I}} \cdot \delta^D(\langle Q_I \rangle, \langle P \rangle),
\]  

(5.3)
where \( \delta^D(F, G) := P[D(F) = 1] - P[D(G) = 1] \) and \( q_I := P[A_I] = (1 - p)^{|I|} \cdot p^{m-|I|} \).

We first upper bound the summands corresponding to sets \( I \) with at most one element. To this end, for all \( i = 1, \ldots, m \), we consider the distinguisher \( D_i \), which, given access to a two-sided random permutation \( \langle S \rangle \), outputs \( D(\langle Q''_1 \succ \cdots \succ Q''_{i-1} \succ S \succ Q''_{i+1} \succ \cdots \succ Q''_m \rangle) \): Note that \( D_i \) can be implemented with time complexity \( t + (m - 1) \cdot \lceil l + t_{A,q_i}(q,l) \rceil \leq t'' \) given the appropriate advice.

We have \( \delta'_i := \delta^D(\langle Q' \rangle, \langle P \rangle) = \delta^D_i(\langle Q' \rangle, \langle P' \rangle) + \delta^D(\langle P' \rangle, \langle P \rangle) \leq 2\gamma + \varepsilon \), where the bound on the first term follows from the Hardcore Lemma (for every fixed value of the advice), whereas the bound on the second term follows from the fact that \( \delta^D(\langle P' \rangle, \langle P \rangle) \leq \varepsilon \). Additionally,

\[
\delta^D_i(\langle Q' \rangle, \langle P \rangle) = (1 - p) \cdot \delta'_i + p \cdot \delta''_i \leq \varepsilon
\]

with \( \delta''_i := \delta^D(\langle Q'' \rangle, \langle P \rangle) \). Using the fact that

\[
\langle Q''_1 \succ \cdots \succ Q''_{i-1} \succ P \succ Q''_{i+1} \succ \cdots \succ Q''_m \rangle \equiv \langle P \rangle,
\]

we obtain \( \delta^D(\langle Q'_i \rangle, \langle P \rangle) = \delta''_i \) and \( \delta^D(\langle Q'' \rangle, \langle P \rangle) = \delta'_i \) for all \( i = 1, \ldots, m \), and thus

\[
\sum_{|I| \leq 1} q_I \cdot \delta^D(\langle Q_I \rangle, \langle P \rangle) = \sum_{i=1}^m \frac{1}{m} \cdot p^m \cdot \delta''_i + p^{m-1}(1 - p) \cdot \delta'_i \leq \max_{i \in \{1, \ldots, m\}} \{ p^m \cdot \delta''_i + m \cdot p^{m-1} \cdot (1 - p) \cdot \delta'_i \}.
\]

However, for all \( i \in \{1, \ldots, m\} \), we combine all of the above observations to obtain

\[
p^m \delta''_i + mp^{m-1}(1 - p) \delta'_i = p^{m-1}(p \delta''_i + (1 - p) \delta'_i) + (m - 1)p^{m-1}(1 - p) \delta'_i \leq p^{m-1} \varepsilon + (m - 1)p^{m-1}(1 - p) \varepsilon + 2\gamma \leq \varepsilon^m + (m - 1)\varepsilon^m(1 - \varepsilon) + 2\gamma = \varepsilon^m (m - (m - 1)\varepsilon) + 2\gamma,
\]

where we also have used \( p \leq \varepsilon \) and the fact that \( p^m + (m - 1)p^{m-1}(1 - p) \) grows monotonically for \( p \in [0, 1] \).

To bound the remaining summands of Equation (5.3), we use a standard hybrid argument. Let \( \langle P'_1 \rangle, \ldots, \langle P'_{m} \rangle \) be independent instances of \( \langle P' \rangle \). For given \( I \) (with size at least two) we define \( Q_{x,i} \) (for \( i \in \{0, \ldots, m\} \))
which is the cascade $S_1 \triangleright \cdots \triangleright S_m$ where $S_j = Q''_j$ if $j \notin \mathcal{I}$, $S_j = P'_j$ if $j \in \mathcal{I}$ and $j \leq i$ and $S_j := Q'_j$ if $j \in \mathcal{I}$ and $j > i$. Then,

$$\delta^D((Q_I), (P)) = \sum_{i=0}^{m-1} \delta^D((Q_{I,i}), (Q_{I,i+1})) + \delta^D((Q_{I,m}), (P)).$$

For $i = 1, \ldots, m$, let $D'_i := D((S_1 \triangleright \cdots \triangleright S_{i-1} \triangleright \cdots \triangleright S_{m}))$, with $S_j$ defined as above, but simulated using the corresponding implementations of $(Q)$ and $(P)$, as well as appropriate advice: In particular, it can be implemented with time complexity $t''$ and thus, for all $i = 1, \ldots, m$,

$$\delta^D((Q_{I,i-1}), (Q_{I,i})) = \delta^{D'_i}((Q'_i), (P')) \leq 2\gamma$$

by the Hardcore Lemma and the fact that this holds for all values of the advice. Finally, as for all $j \in \mathcal{I}$ the function table of $(Q'_j)$ in the cascade $(Q_{I,m})$ has min-entropy at least $\log(N) - \log((1 - \varepsilon)^{-1})$, we can bound $\delta^D((Q_{I,m}), (P))$ using Corollary 5.15 since $|\mathcal{I}| \geq 2$. The fact that $\delta_{t,q}((Q_1 \triangleright \cdots \triangleright Q_m), (P)) = \Delta_{t,q}((Q_1 \triangleright \cdots \triangleright Q_m), (P))$ concludes the proof. 

5.5.3. Tightness of the Bounds

In the following, let $\varepsilon < 1 - 2^{-n}$ such that $\log((1 - \varepsilon)^{-1})$ is an integer, and consider the cc-stateless random permutation $Q : \{0, 1\}^n \to \{0, 1\}^n$ which initially flips a bit $B \in \{0, 1\}$ taking value 0 with probability $\varepsilon$. If $B = 0$, then $Q$ behaves as the identity permutation $\text{id}$, whereas if $B = 1$ it behaves as a randomly chosen permutation $Q'$ with the constraint that the first $\log((1 - \varepsilon)^{-1})$ bits of $Q'(0^n)$ are all equal to 0. Clearly, it is possible to give an efficient stateful algorithm implementing $Q$ (or $(Q)$) by using lazy sampling. We prove the following for $Q$.

Lemma 5.17. Let $P : \{0, 1\}^n \to \{0, 1\}^n$ be a LRP, and let $Q_1, \ldots, Q_m$ be independent instances of $Q$. Then:

(i) For all distinguishers $D$ we have

$$\Delta^D((Q), (P)) \leq \varepsilon,$$

regardless of their computing power.
(ii) There exists a constant-time distinguisher $D^*$ making one single forward query such that
\[
\Delta^{D^*}(Q_1 \vartriangleright \cdots \vartriangleright Q_m, P) = \Delta^{D^*}((Q_1 \vartriangleright \cdots \vartriangleright Q_m), \langle P \rangle) \\
\geq (m - (m - 1)\varepsilon)\varepsilon^m - \frac{1}{2^n}.
\]

This implies that our bound cannot be substantially improved, even if allowing a huge security loss in the reduction. This in particular extends a previous tightness result given by Myers [Mye99] for the special case $m = 2$ to arbitrary $m \geq 2$.

**Proof.** In the following, let $Q$ and $P$ be random variables representing the distributions of the permutation tables of $Q$ and $P$, respectively. There are $(1 - \varepsilon)(2^n)!$ permutations $\pi$ for which the last $\log((1 - \varepsilon)^{-1})$ bits of $\pi(0^n)$ all equal to 0, and the identity $id$ is one such permutation. Hence,

$$P_Q(id) = \varepsilon + (1 - \varepsilon) \cdot \frac{1}{(1 - \varepsilon)(2^n)!} = \varepsilon + \frac{1}{2^n} = P_P(id).$$

For all $\pi \neq id$, we have $P_Q(\pi) \leq (1 - \varepsilon) \cdot \frac{1}{(1 - \varepsilon)(2^n)!} = \frac{1}{2^n} = P_P(\pi)$. This yields $\Delta^D((P), (Q)) \leq d(P, Q) = P_Q(id) - P_P(id) = \varepsilon$ for all distinguishers $D$.

For the second claim, we define $D^*$ as the distinguisher querying $0^n$ and outputting 1 if and only if the first $\log((1 - \varepsilon)^{-1})$ bits of the resulting output are all 0, and outputting 0 otherwise. It is easy to verify that $P[D^*(P) = 1] = 2^{-\log((1 - \varepsilon)^{-1})} = 1 - \varepsilon$, as the output of $P$ on input $0^n$ is a uniformly distributed $n$-bit string.

Denote as $B_i$ the bit $B$ associated with the $i$-th instance $Q_i$, and let $A_\mathcal{I}$ for $\mathcal{I} \subseteq \{1, \ldots, m\}$ be the event that $B_i = 1$ for all $i \in \mathcal{I}$ and $B_i = 0$ for all $i \notin \mathcal{I}$. Furthermore, let $\mathcal{E}$ be the event that $A_\mathcal{I}$ occurs for some $\mathcal{I}$ with $|\mathcal{I}| \leq 1$. Clearly, $P[\mathcal{E}] = \varepsilon^{m} + m(1 - \varepsilon)\varepsilon^{m-1}$ and

$$P[D^*(Q_1 \vartriangleright \cdots \vartriangleright Q_m) = 1 \mid \mathcal{E}] = 1,$$

since $Q_1 \vartriangleright \cdots \vartriangleright Q_m$ under $\mathcal{E}$ behaves either as the identity or as $Q'$, and in both cases the first $\log((1 - \varepsilon)^{-1})$ bits are all 0.

Assume that $A_\mathcal{I}$ occurs for $\mathcal{I}$ with $k := |\mathcal{I}| \geq 2$, and let $Q'_1, \ldots, Q'_k$ be independent random permutations answering according to $Q'$. Then

$$P[D^*(Q_1 \vartriangleright \cdots \vartriangleright Q_m) = 1 \mid A_\mathcal{I}] = P[D^*(Q'_1 \vartriangleright \cdots \vartriangleright Q'_k) = 1].$$
Note that for any input \( x \neq 0^n \) the probability that the first \( \log \left( (1 - \epsilon)^{-1} \right) \) output bits of \( Q'_k(x) \) are all 0 is exactly \( 1 - \epsilon \), whereas the probability that \( Q'_k \) is invoked on \( 0^n \) is at most \( \frac{1}{(1 - \epsilon)2^n} \) (as regardless of the input, the output \( Q'_{k-1} \) is uniformly distributed on a set of at least size \( (1 - \epsilon)2^n \)), and therefore

\[
P\left[ D^*(Q'_1 \supset \cdots \supset Q'_k) = 1 \right] \geq \left( 1 - \frac{1}{(1 - \epsilon)2^n} \right) \cdot (1 - \epsilon) = 1 - \epsilon - \frac{1}{2^n},
\]

and therefore we also have \( P\left[ D^*(Q'_1 \supset \cdots \supset Q'_m) = 1 \middle| \mathcal{E} \right] \geq 1 - \epsilon - \frac{1}{2^n} \), from which we conclude (using \( Q_{\mathcal{E}} := Q_1 \supset \cdots \supset Q_m \))

\[
\Delta D^*(Q_{\mathcal{E}}, P) \geq P\left[ D^*(Q_{\mathcal{E}}) = 1 \right] - P\left[ D^*(P) = 1 \right]
= P\left[ \mathcal{E} \right] \cdot P\left[ D^*(Q_{\mathcal{E}}) = 1 \middle| \mathcal{E} \right] + P\left[ \mathcal{E}^c \right] \cdot P\left[ D^*(Q_{\mathcal{E}}) = 1 \middle| \mathcal{E}^c \right] - (1 - \epsilon)
\geq P\left[ \mathcal{E} \right] \cdot \epsilon - \frac{1}{2^n}
\geq \epsilon^{m+1} + (1 - \epsilon)\epsilon^m - \frac{1}{2^n}
= (m - (m - 1)\epsilon)\epsilon^m - \frac{1}{2^n}.
\]

\[\square\]

Remark 5.2. The above Q can efficiently be implemented by a stateful algorithm. This suffices to prove tightness of Theorem 5.16, as it applies to such a general setting (Q is cc-stateless), yet it leaves open the question of whether the bound is tight in the setting of weak PRPs, where we are given a keyed permutation only. That this is indeed the case can be seen as follows: Assume we are given a two-sided PRP \( P : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n \). Then, we construct \( P' : \{0,1\}^{k+\log((1-\epsilon)^{-1})} \times \{0,1\}^n \to \{0,1\}^n \) such that, for all \( K \in \{0,1\}^k \) and \( K' \in \{0,1\}^{\log((1-\epsilon)^{-1})} \),

\[
P'(K||K', x) := \begin{cases} 
  x & \text{if } K' \neq 0^{\log((1-\epsilon)^{-1})}, \\
  P(K, x) \oplus P(K, 0^n)|_{\log((1-\epsilon)^{-1})} & \text{else},
\end{cases}
\]

where \( x|_{\log((1-\epsilon)^{-1})} \) is such that the first \( \log((1-\epsilon)^{-1}) \) bits equal the corresponding bits of \( x \), and the last \( n - \log((1-\epsilon)^{-1}) \) bits are 0. It can be shown that \( P' \) is an \((\epsilon + \nu)\)-PRP for a negligible function \( \nu \). Also, it is
easy to verify that given a random key, \( P' \) implements the identity with probability \( \epsilon \), and with probability \( 1 - \epsilon \) implements a permutation such that, when queried on input \( 0^n \), the first \( \log ((1 - \epsilon)^{-1}) \) bits are all zero, and thus the above distinguisher \( D^* \) also applies, from which we infer tightness.

### 5.6. Randomized Cascade and Randomized XOR

We conclude this chapter by providing alternative security amplification proofs for the randomized cascade (Section 4.5.1) and the randomized XOR of random functions (Section 4.5.3), by seeing them as extracting constructions. We note that by setting \( \Lambda := \frac{2^n}{2} \), the bounds are slightly superior to the ones obtained in Chapter 4.

#### 5.6.1. The Randomized Cascade

In the following, let \( \langle Q \rangle \) be a two-sided cc-stateless random permutation on the \( n \)-bit strings (with implementation \( A_{\langle Q \rangle} \)), and let \( \langle Q_1 \rangle, \ldots, \langle Q_m \rangle \) be independent instances of \( \langle Q \rangle \). Recall that the randomized cascade (cf. Section 4.5.1) is the construction realizing

\[
\langle \oplus Z_1 \rangle \triangleright \langle Q_1 \rangle \triangleright \cdots \triangleright \langle Q_m \rangle \triangleright \langle \oplus Z_1 \rangle,
\]

from \( \langle Q_1 \rangle, \ldots, \langle Q_m \rangle \), where \( Z_1 \) and \( Z_2 \) are independent uniform \( n \)-bit strings.

In the following, we show that the randomized cascade construction is \((\epsilon, 1, \delta)\)-extracting, for all \( \epsilon \in [0, 1) \) and some sufficiently small \( \delta \). The key step is the following lemma.

**Lemma 5.18.** Let \( \overline{Q} : \{0, 1\}^n \rightarrow \{0, 1\}^n \) be a cc-stateless random permutation whose function table has min-entropy at least \( \log(2^n!) - \log ((1 - \epsilon)^{-1}) \). Then, for independent uniform \( n \)-bit strings \( Z_1 \) and \( Z_2 \), and for all \( \Lambda > 0 \),

\[
\Delta_q(\langle \oplus Z_1 \triangleright \overline{Q} \triangleright \oplus Z_2 \rangle, \langle P \rangle) \leq \left( \frac{q \log ((1 - \epsilon)^{-1})}{\Lambda} \right)^{\frac{1}{2}} + \frac{2q\Lambda}{2^n},
\]

where \( \langle P \rangle \) is a two-sided \( n \)-bit URP.
Proof. Lemma 5.13 guarantees the existence of a user $U$ such that

$$
\Delta_q(\langle \oplus Z_1 \triangleright Q \triangleright \oplus Z_2 \rangle, \langle \oplus Z_1 \triangleright (Q \cup \oplus Z_2) \rangle) \leq \Delta_q(\langle Q \rangle, \langle (Q \cup Z_2) \rangle) \\
\leq \left( \frac{q \log \left( \frac{1 - \varepsilon}{\Lambda} \right) - 1}{\Lambda} \right)^{\frac{1}{2}}.
$$

Let $\hat{H}$ be the system with an MBO $A_1, A_2, \ldots$ behaving as $\langle \oplus Z_1 \triangleright (Q \cup \oplus Z_2) \rangle$, with $A_i = 1$ if one of the first $i$ queries to $\hat{H}$ results in a query to $\langle Q \rangle$ which is part of the input-output pairs obtained by $\hat{U}$. Also, consider the system $\hat{P}$ with MBO $B_1, B_2, \ldots$ which initially lets $U$ interact with $\langle Q \rangle$, and then, independently of the resulting input-output pairs, behaves as $\langle \oplus Z_1 \triangleright P \triangleright \oplus Z_2 \rangle$. Moreover, $B_i = 1$ whenever, among the first $i$ queries, some input-output pair resulting from a query to the underlying $\langle P \rangle$ has common input or output with an input-output pair obtained by $\hat{U}$ at initialization.

Then, we clearly have $\hat{H} \equiv \langle \oplus Z_1 \triangleright (Q \cup \oplus Z_2) \rangle$, $\hat{P} \equiv \langle P \rangle$. Moreover, $\hat{H}$ and $\hat{P}$ have the same behavior conditioned on the MBOs being $0$, and the probability that the MBO turns one is clearly larger in $\hat{P}$: We can hence apply Lemma 2.6,

$$
\Delta_q(\langle \oplus Z_1 \triangleright (Q \cup \oplus Z_2) \rangle, \langle P \rangle) \leq \nu_q(\hat{P}).
$$

To bound the latter probability, we apply Lemma 2.9, since the behavior of $\hat{P}$ is independent of the outcomes of $Z_1$ and $Z_2$, and of $U$’s interaction: For any $q$ inputs and $q$ outputs, the probability that random $n$-bit $Z_1$ and $Z_2$ provoke $B_i = 1$ is at most $\frac{2q\Lambda}{2n}$ by the union bound. \qed

Under the assumption that there exists an $i \in \{1, \ldots, m\}$ such that the function table of $Q_i$ has min-entropy at least $\log(2^n! - \log (1 - \varepsilon)^{-1})$, the table of $\overline{Q} := Q_1 \triangleright \cdots \triangleright Q_m$ has also min-entropy at least $\log(2^n! - \log (1 - \varepsilon)^{-1})$, and Lemma 5.18 implies that the randomized cascade is $(\varepsilon, 1, \delta)$-extracting for

$$
\delta(q) = \left( \frac{q \log \left( \frac{1 - \varepsilon}{\Lambda} \right)}{\Lambda} \right)^{\frac{1}{2}} + \frac{2q\Lambda}{2n}
$$

for all $\varepsilon \in [0, 1)$. For a two-sided URP $\langle P \rangle$ on the $n$-bit strings, this yields the following corollary by Theorem 5.5.
Corollary 5.19. For all integers $t, q > 0$, for all $\varepsilon \in [0, \frac{1}{2}]$, if
\[
\Delta_{t, q}((Q), (P)) \leq \varepsilon,
\]
then, for all $\Lambda > 0$,
\[
\Delta_{t, q}(\oplus_{Z_1} \oplus (Q_1) \oplus \cdots \oplus (Q_m) \oplus Z_1, (P)) \leq \varepsilon^m + \left(\frac{q \log \left((1 - \varepsilon)^{-1}\right)}{\Lambda}\right) \frac{1}{2} + \frac{2q\Lambda}{2^\psi} + 2m\gamma,
\]
where
\[
t' = \varphi_{hc} \cdot [t + O\left(m \cdot \max\{t + t_{A, Q}, q \cdot (\psi + \log(q(\psi + 1)))\}\right)]
\]
and $q' = q \cdot \varphi_{hc}$ for $\psi := 7 \cdot \gamma^{-2} \cdot (1 - \varepsilon)^{-3}$ and $\varphi_{hc}$ as in Theorem 5.2, and $l := s_{A, Q}(q \cdot \psi)$.

5.6.2. Randomized XOR of PRFs

Let $F : \{0, 1\}^n \rightarrow \{0, 1\}^\ell$ be a cc-stateless random function (with implementation $A_F$), and let $F_1, \ldots, F_m$ be independent instances of $F$. Then, as in Section 4.5.3, we consider the construction which realizes
\[
\oplus_{Z} \oplus (F_1 \oplus \cdots \oplus F_m),
\]
from $F_1, \ldots, F_m$, where $Z$ is a uniform $n$-bit string. We show the following.

Lemma 5.20. Let $F : \{0, 1\}^n \rightarrow \{0, 1\}^\ell$ be a cc-stateless random function whose function table has min-entropy at least $\ell 2^n - \log ((1 - \varepsilon)^{-1})$. Then, for a uniform $n$-bit string $Z$ and for all $\Lambda > 0$,
\[
\Delta_q(\oplus_{Z} \oplus F, R) \leq \left(\frac{q \log \left((1 - \varepsilon)^{-1}\right)}{\Lambda}\right) \frac{1}{2} + \frac{q\Lambda}{2^\psi},
\]
for a URF $R : \{0, 1\}^n \rightarrow \{0, 1\}^\ell$.

Proof. Let $U$ be as guaranteed by Lemma 5.13, i.e., such that
\[
\Delta_q(\oplus_{Z} \oplus F, \oplus_{Z} \oplus F_U) \leq \Delta_q(F, F_U) \leq \left(\frac{q \log \left((1 - \varepsilon)^{-1}\right)}{\Lambda}\right) \frac{1}{2}.
\]
Consider $\hat{H}$ which behaves as $\oplus_Z \triangleright F_U$ with an MBO $A_1, A_2, \ldots$ such that $A_i = 1$ if and only if one of the inputs $x \oplus Z$ among the first $i$ queries belongs to $U$’s queries. Note that as long as $A_i = 0$, all outputs are independent and uniform, and we can use both Corollary 2.7 and Lemma 2.8 to obtain

$$\Delta_q (\oplus_Z \triangleright F_U, R) \leq \nu_q (\hat{H}) \leq \max_{x^q} p_{\hat{H}}^X (1, x^q) \leq \frac{q \Lambda}{2^n},$$

since for each $i$, the probability that $x_i \oplus Z$ is among $U$’s queries is $\frac{\Lambda}{2^n}$. \[\square\]

If there exists an $i \in \{1, \ldots, m\}$ such that the function table of $F_i$ has min-entropy at least $\ell 2^n - \log ((1 - \epsilon)^{-1})$, then the same holds for the function table of $F := F_1 \oplus \cdots \oplus F_m$, and hence Lemma 5.20 implies that the randomized-xor construction is $(\epsilon, 1, \delta)$-extracting for

$$\delta(q) \leq \left( \frac{q \log ((1 - \epsilon)^{-1})}{\Lambda} \right)^{\frac{1}{2}} + \frac{q \Lambda}{2^n},$$

and all $\epsilon \in [0, 1)$. Combined with Theorem 5.5, this implies the following corollary, for a URF $R : \{0, 1\}^n \to \{0, 1\}^\ell$.

**Corollary 5.21.** For all integers $t, q > 0$, for all $\epsilon \in [0, 1)$ and $\gamma \in (0, \frac{1}{2}]$, if $\Delta_{t', q'} (F, R) \leq \epsilon$, then, for all $\Lambda > 0$,

$$\Delta_{t, q} (\oplus Z \triangleright (F_1 \oplus \cdots \oplus F_m), R) \leq \epsilon^m + \left( \frac{q \log ((1 - \epsilon)^{-1})}{\Lambda} \right)^{\frac{1}{2}} + \frac{q \Lambda}{2^n} + 2m \gamma,$$

where

$$t' = \varphi_{hc} \cdot \left[ t + O \left( m \cdot \max \{ l + t_{Ap} (q, l), q \cdot \psi n + \log (q (\psi + 1)) \} \right) \right]$$

and $q' = q \cdot \varphi_{hc}$ for $\psi := 7 \cdot \gamma^{-2} \cdot (1 - \epsilon)^{-3}$ and $\varphi_{hc}$ as in Theorem 5.2, and $l := s_{Ap} (q \cdot \psi)$. 

Chapter 6

From Random-Input to Chosen-Input Indistinguishability

This final chapter is devoted to computational indistinguishability amplification with respect to the distinguisher class. More specifically, we consider PRF constructions from random-input PRFs (RI-PRFs), for which indistinguishability is only guaranteed against distinguishers of the form $DK$, for an efficient distinguisher $D$ and the construction $K$ enforcing random-input access (cf. Section 2.6.3). In particular, we address a further relaxation of this notion, called an $s$-RI-PRF, where the number of distinguisher queries is bounded by a small quantity $s$ such as a constant or a small growing function.$^1$

For small $s$, the $s$-RI-PRF notion is significantly weaker than the concept of a full-fledged RI-PRF. For instance, the outputs associated with $s + 1$ distinct random inputs may satisfy an easily verifiable relation with no impact on the pseudorandomness of the function. Also, from a practical standpoint, being an $s$-RI-PRF is a very realistic assumption on a cryptographic function such as a block cipher or a compression function.

$^1$Note that a $s$-RI-PRF $\{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^\ell$ is only interesting if the key length satisfies $\kappa < s \cdot \ell$. For $\kappa \geq s \cdot \ell$, such functions can be constructed unconditionally, e.g. using $s$-wise independent functions. (However, optimal unconditional constructions with $\kappa = s \cdot n$ are not known for all parameters $m$.)
of a hash function (like SHA-1), even in the presence of very powerful cryptanalytic attacks.

**Chapter Outline and Contributions.** We provide PRF constructions that outperform, in terms of efficiency, existing construction from RI-PRFs (with an unrestricted amount of queries), while only requiring the underlying function to be an $s$-RI-PRF, for $s$ as low as two. Furthermore, our constructions are iterated and can process inputs of arbitrary length. This structure makes them well suited to be derived from properly keyed hash functions with very weak compression functions.

More in detail, the contents of this chapter consist of the following results, which have first appeared in [MT08]:

- In Section 6.1, we present our first construction (called the RC construction) of an arbitrary-input-length PRF from any $s$-RI-PRF (for $s \geq 2$) for which output and key lengths are equal. As a special case, one obtains a fixed-input-length PRF which, for input length $\ell$, makes $\approx \frac{\ell}{\log s}$ calls to the underlying $s$-RI-PRF per evaluation.

- In Section 6.1.4, we show that careful instantiation of the RC construction yields an efficient symmetric encryption scheme relying on the sole assumption of an $s$-RI-PRF (for some $s \geq 2$), while requiring (on average) only $1 + \frac{1}{s-1}$ calls to $F$ per encrypted data block, and minimal storage overhead. Furthermore, in Section 6.1.5 we also show that the RC construction yields constructions of efficient PRGs from $s$-RI-PRFs.

- Section 6.2 presents a further construction, called the NRC construction, which improves the throughput of the RC construction for long messages making a novel use of pairwise independence, while still solely relying on the underlying function being an $s$-RI-PRF.

- Finally, Section 6.3 addresses the problem of deriving our constructions by keying iterated hash functions (such as SHA-1 or MD5) whose compression function is an $s$-RI-PRF: If minimal (and natural) regularity properties are additionally guaranteed by the compression function, the keying can be done in an entirely black-box way.

We point out that being an $s$-RI-PRF is, to the best of our knowledge, the weakest assumption on (keyed) cryptographic functions for which efficient modes of operations have ever been considered.
**Related Work.** The notion of a RI-PRF was first introduced by Naor and Reingold [NR99] under the name “weak PRF”. The use of RI-PRFs in efficient cryptographic constructions was studied by Damgård and Nielsen [DN02], who considered constructions for range extension of RI-PRFs, a question further studied by Maurer and Sjödin [MS07] and by Pietrzak and Sjödin [PS07].

The first construction of a PRF from a RI-PRF is also due to Naor and Reingold [NR99], and a further construction was later proposed by Maurer and Sjödin [MS07]. Both assume a length-preserving underlying function $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ (which can be obtained e.g. from a block cipher) and realize a keyed function mapping $\ell$-bit strings to $n$-bit strings (for a fixed input length $\ell$).

**The Naor-Reingold Construction [NR99].** The construction $\text{NR}_\ell$ takes an $\ell$-bit input (with $\ell$ being a power of two) and its secret key consists of $2\ell n$-bit strings $k_{1,0}, k_{1,1}, \ldots, k_{\ell,0}, k_{\ell,1}$. The computation on input $x = (x_1, \ldots, x_\ell)$ proceeds as follows: First, we define $y_i^{(\log \ell+1)} := k_{i,x_i}$ for all $i = 1, \ldots, \ell$. Then, for all $j = \log \ell, \ldots, 1$ we compute $y_i^{(j)} := F(y_{2i-1}^{(j+1)}, y_{2i}^{(j+1)})$ for all $i = 1, \ldots, 2j-1$ and finally output $y_1^{(1)}$. In other words, the elements of the key corresponding to the individual input bits are chosen as the values of the $\ell$ leaves of a complete binary tree which is evaluated in a bottom-up fashion by computing the value of each inner vertex as $F(y_l, y_r)$, where $y_l$ and $y_r$ are the values of its children, and finally outputting the value of the root. Hence, one evaluation of the construction needs $\ell 2^{\log \ell} + \ell 2^{\log \ell-1} + \cdots + \ell 2^0 = \ell - 1$ calls to the underlying function $F$. A more involved construction (which we call $\text{NR}_{s,\ell}$) by the same authors uses a key consisting of $s n$-bit values and improves the total number of calls to roughly $\ell / \log s$ per evaluation, but only accepts $\ell$ and $\log s$ to have the form $2^j + 2$ for some $j \geq 0$. (For both constructions, other input lengths can be achieved through appropriate paddings.)

**The IC Construction [MS07].** The construction $\text{IC}_\ell$ takes a $(\kappa + 2n)$-bit key consisting of three values $k_1 \in \{0,1\}^\kappa$ and $r, r' \in \{0,1\}^n$. (The value $r'$ can even be made public.) It first precomputes the values $k_i := F(k_{i-1}, r')$ for all $i = 2, \ldots, \ell$. On input $x = (x_1, \ldots, x_\ell) \in \{0,1\}^\ell$, it sets $y_0 := r$, and for all $j = 1, \ldots, \ell$, computes $y_j := F(k_j, y_{j-1})$ if $x_j = 1$, and $y_j := y_{j-1}$ else. Finally, it outputs $y_\ell$. The construction $\text{IC}_\ell$ requires

---

2The constructions of [NR99] rely on an intermediate primitive, called a synthesizer, but in fact a RI-PRF is a synthesizer.
$w(x)$ calls to $F$ when evaluated on input $x$, where $w(x) \leq \ell$ is the Hamming weight of $x$. If memory restrictions do not allow storage of the keys $k_2, \ldots, k_\ell$, their values have to be computed at each evaluation and thus the construction requires $(\ell - 1) + w(x)$ calls to $F$ per evaluation, which can be as high as $2\ell - 1$.

In order to prove security of all the aforementioned constructions with respect to adversaries issuing $q$ queries, the underlying RI-PRF must also be secure when evaluated at (at least) $q$ random inputs. Moreover, one application of our constructions is to employ them as iterated constructions of PRFs where candidates for RI-PRFs may arise from (keyed) compression functions of hash functions, which have the form $F : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^\kappa$ (where e.g. $\kappa = 160$ and $n = 512$ for SHA-1). The above constructions can all be extended in a straightforward way to handle such functions as well, but for the same input length $\ell$ the number of calls would increase considerably if $n > \kappa$ (roughly, by a factor of $\lceil n/\kappa \rceil$ with respect to the case $n = \kappa$, which is e.g. 4 for SHA-1). This holds even if we just want $\kappa$-bit outputs. Hence, this calls for a construction for which the condition $n > \kappa$ does not have a negative impact on the efficiency of the construction.

Finally, we point out that non-adaptively secure PRFs are in particular RI-PRFs, and that even under this stronger assumption, constructions of full-fledged PRFs with better efficiency are not known. In fact, while very simple operations are sufficient in order to achieve adaptive security in the information-theoretic setting [MP04, MPR07, GM09b], most of these have been shown not to work in the computational setting [Mye04, Pie05], unless public-key encryption does not exist [Pie06, CLO10].

6.1. The Randomized Cascade Construction

6.1.1. Description of the Construction

In this section, we present the first construction of this chapter: It is reminiscent of the cascade construction of Bellare et al. [BCK96b], but only

---

3The security statements of [NR99] are asymptotic, but more concrete statements can be obtained by a closer inspection of the proofs.

4One can simply base the above constructions on the function $F' : (k_1 || \ldots || k_c, r) \mapsto F(k_1, r) || \ldots || F(k_c, r)$ (possibly chopping some bits) where $c = \lceil n/\kappa \rceil$ (the function $F'$ can be shown to be a RI-PRF). Note that more involved range-extension techniques (such as those from [DN02, MS07, PS07]) do not work here, as they require a length-preserving function beforehand.
requires the underlying function $F : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^\kappa$ to be an $s$-RI-PRF with $s \geq 2$ being a parameter of the construction.

Recall that, for a set $X$, an efficiently computable injective function $\text{ENC} : X \rightarrow \{1,\ldots,s\}^+$ is a prefix-free encoding scheme if $\text{ENC}(x)$ is not a prefix of the sequence $\text{ENC}(x')$ for all distinct $x, x' \in X$. It is generally desirable that $\text{ENC}$ operates in an on-line manner. (We refer the reader to the recent work by Dodis et al. [DPT10] for the state of the art in building such codes.) If $X = \{0,1\}^\ell$ for some fixed $\ell$, then prefix-freeness is achieved “for free” by encoding all inputs as sequences in $\{1,\ldots,s\}^*$ of equal length $\left\lceil \frac{\ell}{\log_2 s} \right\rceil$.

For a function $F : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^\kappa$, its cascade $F^* : \{0,1\}^\kappa \times (\{0,1\}^n)^+ \rightarrow \{0,1\}^\kappa$ is such that, for key $k$ and input $(x_1, \ldots, x_\ell) \in (\{0,1\}^n)^+$, it outputs $\sigma_t$, where $\sigma_0 := k$, and $\sigma_i := F(\sigma_{i-1}, x_i)$ for all $i = 1, \ldots, \ell$. The randomized cascade construction with parameter $s$ and input set $X$ (where usually either $X = \{0,1\}^s$ or $X = \{0,1\}^\ell$ for a fixed $\ell$) for the function $F$ and prefix-free encoding scheme $\text{ENC}$, denoted $\text{RC}_{s,X,\text{ENC}}(F)$, is a mapping $\{0,1\}^\kappa \times \{0,1\}^{sn} \times X \rightarrow \{0,1\}^\kappa$: As its inputs, it takes a key consisting of a $\kappa$-bit private part $k$ and an $sn$-bit long public part, which is interpreted as the concatenation of $s$ $n$-bit strings $r_1, \ldots, r_s$, as well as a value $x \in X$, and outputs $F^*(k, (r_{m_1}, \ldots, r_{m_s}))$, where $\text{ENC}(x) = (m_1, \ldots, m_s)$.

For notational convenience, we use the shorthands $\text{RC}_{s,\text{ENC}}$ for $X = \{0,1\}^s$ (and omit the prefix-free encoding when it is generally understood from the context), as well as $\text{RC}_{s,\ell}$ for $X = \{0,1\}^\ell$ (where the canonical encoding described above is used). We also generically refer to the construction as the RC construction. The RC construction is depicted in Figure 6.1 for the special case $s = 2$.

**EFFICIENCY COMPARISONS.** A fair comparison between the RC construction and previous results can be undertaken for the fixed-input-length construction $\text{RC}_{s,\ell}$ only. In the length-preserving case ($\kappa = n$), the construction $\text{RC}_{\ell,s}$ is comparable to (for the case $s = 2$) the NR- and the IC constructions in terms of calls to $F$, and outperforms them for $s > 2$. Furthermore, we obtain the same space-time trade-off of the NR$_{s,\ell}$ construction, but we allow for all possible values of $s$. Our construction also limits the effects of possibly very long input paddings in the NR- and NR constructions. The efficiency improvement of our construction is however more evident in the case where $n > \kappa$, as even if $s = 2$, the number of calls to $F$ of (the extended versions of) all other constructions is larger at least by a factor $\left\lceil \frac{\ell}{s} \right\rceil$ (the factor is e.g. 4 when instantiating $F$ with the
compression function of SHA-1). Finally, because of the iterated structure, efficient sequential evaluation of $\text{RC}_{s,\ell}$ requires (beside sufficient storage for the key material) $\kappa$ bits only to store the “chaining value”.

6.1.2. Security of the RC Construction

Every prefix-free encoding $\text{ENC}$ is conveniently described in terms of a (possibly infinite) directed tree $T = (V, E)$: Its vertex set $V$ consists of all sequences $(m_1, \ldots, m_j)$ which are a prefix of $\text{ENC}(x)$ for some input $x$ (in particular, including the encodings themselves and the empty sequence $\bot$), and for each $(m_1, \ldots, m_j) \in V$ there exists a directed edge to $(m_1, \ldots, m_j, m_{j+1})$ for all $m_{j+1} \in \{1, \ldots, s\}$ such that $(m_1, \ldots, m_{j+1}) \in V$. It is easy to see that $\bot$ is the root of the directed tree and its leaves are exactly the encodings of the inputs. We provide two examples of such trees in Figure 6.2.

Every sequence of queries to the RC construction thus defines a subtree of $T$ consisting of the paths from the root to the encodings of the queries: For notational convenience, we define the shorthand $L(x_1, \ldots, x_q)$ to be the number of inner vertices (i.e. vertices which are not leaves) of the subtree induced by the $q$ inputs $x_1, \ldots, x_q$. It is easy to verify that $L(x_1, \ldots, x_q) \leq 1 + q \left( \left\lceil \frac{\ell}{\log s} \right\rceil - 1 \right)$ for $\text{RC}_{s,\ell}$. Also, we define $L(q, \ell) := \max_{x_1, \ldots, x_q:|x_i| \leq \ell} L(x_1, \ldots, x_q)$ for the case where the inputs are strings with arbitrary length.

Consequently, one can see an interaction with the RC construction as a process where the tree $T = (V, E)$ defined by $\text{ENC}$ is traversed and $\kappa$-bit values are assigned to all visited vertices: While the root $\bot$ is assigned a random $\kappa$-bit value, the value of each visited vertex $(m_1, \ldots, m_j)$ is set to $F(z, r_{m_j})$, with $z$ being the value of the parent vertex $(m_1, \ldots, m_{j-1})$. 
A query with input $x$ is answered with the value at the corresponding leaf $\text{ENC}(x)$. By the definition of an $s$-RI-PRF, it is easy to see that evaluating $F$ under some given (pseudo-)random secret key at $s$ independent random inputs produces $s$ pseudorandom outputs, and hence intuitively the above process sets the values of all visited vertices to pseudorandom values (and in particular this holds for the leaves). However, to formalize this intuition, we have to show that it is indeed possible to recycle the same values $r_1, \ldots, r_s$ for each inner vertex.

From now on, we fix $s$ and a prefix-free encoding scheme $\text{ENC} : \mathcal{X} \to \{1, \ldots, s\}^+$, a function $F : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^\kappa$, and we let $\mathcal{R} : \mathcal{X} \to \{0,1\}^\kappa$ and $\mathcal{R}' : \{0,1\}^n \to \{0,1\}^\kappa$ be URFs. Moreover, we let $\mathcal{R}_C = \mathcal{R}_{C,\mathcal{X},\text{ENC}}(F)$. The following theorem formally captures the main security statement for the $\mathcal{R}_C$ construction (for input set $\mathcal{X}$). Its proof can be found in the next section.

**Theorem 6.1** (Security of the $\mathcal{R}_C$ construction). For all $L, t > 0$ and all distinguishers $D \in \mathcal{D}_e$ with $L(x_1, x_2, \ldots) \leq L$ for all query sequences $x_1, x_2, \ldots \in \mathcal{X}$, there exists a distinguisher $D'$ such that

$$\Delta^D((\mathcal{R}_C(K, R, \cdot, R), (R, R))) \leq L \cdot [\Delta^{D'}(F(K, \cdot), R') + s^2 \cdot 2^{-(n+1)}],$$

where $K$ and $R$ are uniformly distributed $\kappa$- and $sn$-bit strings, respectively, and $D'$ makes $s$ queries and has running time $t' = t + O(L \cdot t_F)$, with $t_F$ being the time needed to evaluate $F$.

---

5Except in the case where two of the random inputs $r_1, \ldots, r_s$ collide, which happens with small probability only.
Recall that $K(\cdot)$ stands for the construction allowing for a random-input attack defined in Section 2.6.3. The notation $(\mathbf{RC}(K, R, \cdot), R)$ indicates that the value $R$ is additionally given to the distinguisher, i.e., it does not need to be secret. We also remark that the term $s^2 2^{-(n+1)}$ is generally negligible, as $s$ is assumed to be small (e.g., a constant).

The theorem can be restated in the following more compact form in the arbitrary-input-length case.

**Corollary 6.2.** For all $t, q, \ell \geq 0$,

$$
\Delta_{t, q, \ell}((\mathbf{RC}(K, R, \cdot), R), (R, R)) \leq L(q, \ell) \cdot \left[ \Delta_{t', s}(K(F(K, \cdot)), K(R')) + s^2 2^{-(n+1)} \right],
$$

with $t' = t + O(L(q, \ell) \cdot t_F)$.

In comparison with earlier constructions from RI-PRFs, all variants of the $\mathbf{RC}$ construction require $F$ to be only an $s$-RI-PRF. A minor positive aspect of the randomized cascade construction (if compared with other constructions) is the absence of any $q$-dependent birthday-like term in the above bound. Furthermore, if we assume that $F$ is indeed secure against $q$ queries, the security of the $\mathbf{RC}_{s, \ell}$ construction is comparable to the one of the $\mathbf{IC}_{\ell}$ construction if we assume (in fact, very optimistically) that the best RI-PRF distinguishing advantage grows linearly in the number of queries, i.e.

$$
\Delta_{t, q}(K(F(K, \cdot)), K(R')) = \Theta(q \cdot \Delta_{t, s}(K(F(K, \cdot)), K(R'))).
$$

**Remark 6.1.** It is easy to increase the output size of the $\mathbf{RC}$ construction (if needed) with the addition of a minor number of invocations of $F$ per evaluation, which is independent of the input length: To obtain a construction $\overline{\mathbf{RC}}(F) : \{0, 1\}^\kappa \times \{0, 1\}^{sn} \times \mathcal{X} \to \{0, 1\}^{\phi \kappa}$ with output size $\phi \cdot \kappa$, we fix $\phi$ distinct strings $a_1, \ldots, a_\phi \in \mathcal{X}$ such that $L(a_1, \ldots, a_\phi)$ is minimal. Then, given key with private part $k$ and public part $r_1, \ldots, r_s$, on input $x \in \mathcal{X}$, to compute $\overline{\mathbf{RC}}(F)(k, r_1||\ldots||r_s, x)$ we first let

$$
k' := \mathbf{RC}(F)(k, r_1||\ldots||r_s, x)
$$

and finally output

$$
\mathbf{RC}(F)(k', r_1||\ldots||r_s, a_1)||\ldots||\mathbf{RC}(F)(k', r_1||\ldots||r_s, a_\phi).
$$

Security of this construction can be inferred by the fact that evaluating it at input $x$ accounts to evaluating at inputs $(x, a_1), \ldots, (x, a_\phi)$ a variant of the $\mathbf{RC}$ construction with input set $\mathcal{X} \times \{a_1, \ldots, a_\phi\}$ and the prefix-free encoding $\overline{\mathbf{ENC}}(x, a) := \overline{\mathbf{ENC}}(x) || \overline{\mathbf{ENC}}(a)$. 
6.1.3. Proof of Theorem 6.1

The proof relies on \( L \) hybrid constructions \( H_0(\cdot), \ldots, H_{L-1}(\cdot) \) that access \( K(F) \), for a random function \( F : \{0,1\}^n \rightarrow \{0,1\}^k \). For all \( i = 0, \ldots, L-1 \), the construction \( H_i(\cdot) \) first makes \( s \) queries to \( K(F) \), which return \((r_1, y_1), \ldots, (r_s, y_s) \in \{0,1\}^n \times \{0,1\}^k \) \( s \) queries to \( K(F) \), and outputs \( r_1, \ldots, r_s \in \{0,1\}^n \). It subsequently implements a function \( X \rightarrow \{0,1\}^k \). In particular, it keeps track of vertices of the subtree of \( T = (V, E) \) (defined as above) induced by the queries to the implemented function, assigning to all internal vertices \( v \) of the subtree increasing integer values \( \text{label}(v) \) when they are visited for the first time. Furthermore, \( H_i(K(F)) \) associates \( \kappa \)-bit values \( \text{value}(v) \) with all visited vertices in the following way: Each query \( x \in X \) (with \( \text{ENC}(x) = (m_1, \ldots, m_\lambda) \)) is answered by looking for the highest \( \lambda^* \) such that \( \text{value}(m_1, \ldots, m_\lambda) \) is defined (we abuse notation setting \((m_1, m_0) = \perp\)), and for all \( j = \lambda^* + 1, \ldots, \lambda \) defining \( \text{value}(m_1, \ldots, m_j) \) to a random value if the label of \((m_1, \ldots, m_{j-1})\) is smaller than \( i \), to \( y_m \) if it equals \( i \), and otherwise \( \text{value}(m_1, \ldots, m_j) := F(\text{value}(m_1, \ldots, m_{j-1}), r_m) \).

Finally, the value of \((m_1, \ldots, m_\lambda)\) is returned.

We can easily see that, for a beacon \( B \) with \( \kappa \)-bit output,

\[
\begin{align*}
H_0(K(F(K,\cdot)))) &\equiv (\text{RC}(K, R, \cdot), R), \\
D(H_{L-1}(K(B))) &\equiv D(R, R).
\end{align*}
\]

While the first equivalence is obvious by inspection, the second one follows from the observation that for the given distinguisher \( D \) the labels assigned to vertices are bounded by \( L - 1 \), and thus in this case all values returned by queries are independent and uniformly distributed (as, in particular, the values \( y_1, \ldots, y_s \) are also independent from any other value and from \( r_1, \ldots, r_s \)). Moreover, for all \( i = 0, \ldots, L - 2 \)

\[
H_i(K(B)) \equiv H_{i+1}(K(F(K,\cdot)))),
\]

because in \( H_i(K(B)) \) all values of vertices whose parent has label not larger than \( i \) (this includes the vertex with label \( i+1 \)) are independent and uniform, whereas the values of all other vertices are computed by using the function \( F \) accordingly. However, if we use an externally given \( K \) as the value of the vertex with label \( i+1 \), rather than an internally generated random value, the behavior does not change, as \( K \) is never output by the prefix-freeness of \( \text{ENC} \), and we obtain exactly \( H_{i+1}(K(F(K,\cdot)))) \).
To conclude, we define the distinguisher $D'$ that picks an index $i \in \{0, \ldots, L-1\}$ uniformly at random and runs $D'_i := D_i$ for a uniformly-distributed $\kappa$-bit key $K$, and with $F := K(F(K, \cdot))$ and $I := K(B)$,

$$\Delta D'(F, I) = \frac{1}{L} \left| \sum_{i=0}^{L-1} P[D'_i(F) = 1] - P[D'_i(I) = 1] \right|$$

$$= \frac{1}{L} \left| \sum_{i=0}^{L-1} P[D(H_i(F)) = 1] - P[D(H_i(I)) = 1] \right|$$

$$= \frac{1}{L} \left| P[D(H_0(F)) = 1] - P[D(H_{L-1}(I)) = 1] \right|$$

$$= \frac{1}{L} \Delta D((RC(K, R, \cdot), R), (R, R)).$$

Moreover, by the triangle inequality,

$$\Delta D'K(F(K, \cdot), B) \leq \Delta D'(F(K, \cdot), R') + \Delta D'(R', B)$$

and it is easy to show that $\Delta D'(R', B) \leq \binom{s}{2} \cdot 2^{-n}$ by the fact that $D'$ queries the given system at $s$ independent $n$-bit random inputs, and the advantage is bounded by the probability that two such inputs collide.

It also easy to verify that the distinguisher $D'$ can be implemented with the given running time. (We assume that the running time of $ENC$ is minimal and can hence be neglected.)

### 6.1.4. Applications I: Efficient Symmetric Encryption

This section presents a construction of a (symmetric) encryption scheme from an $s$-RI-PRF which is (already for minimal values of $s$) nearly as efficient as PRF-based encryption schemes (such as counter-mode and CBC-encryption). This result shows the feasibility of practically efficient symmetric cryptography from very weak assumptions.

First, we briefly recall the definition of a symmetric encryption scheme.

**Definition 6.1.** For a function $\mu : \mathbb{N} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$, a symmetric encryption scheme (SES) with completeness $\mu$, key space $K$, and plaintext space $M$, is a pair of efficiently computable constructions $(ENC(\cdot), DEC(\cdot))$ such that, for any $q, \ell \in \mathbb{N}$, for a randomly chosen key $K \in K$, and any sequence of plaintexts $M_1, \ldots, M_q \in M$ with length at most $\ell$ input to $ENC(K)$, resulting in ciphertexts $C_1, \ldots, C_q$, we have

$$P[\forall i \in \{1, \ldots, q\} : DEC(K)(C_i) = M_i] \geq \mu(q, \ell).$$
In general, one considers symmetric encryption schemes with completeness 1. Furthermore, note that ENC is allowed to be randomized, and to even keep a state, whereas DEC is generally assumed to be stateless, even though sometimes a stateful implementation may lead to better efficiency.

The following is a standard security notion for symmetric encryption schemes.

**Definition 6.2.** A SES \((\text{ENC}(\cdot), \text{DEC}(\cdot))\) is \((t, q, \ell, \varepsilon)\)-indistinguishable under chosen-plaintext attacks (IND-CPA) if for a uniform random key \(K \in \mathcal{K}\),

\[
\Delta_{t, q, \ell}(\text{SEL}_0 \text{ENC}(K), \text{SEL}_1 \text{ENC}(K)) \leq \varepsilon(t, q, \ell),
\]

where \(\text{SEL}_b\), for \(b \in \{0, 1\}\), takes a pair of plaintexts \((m_0, m_1)\) with the same length \(|m_0| = |m_1|\) as input at the outer interface, forwards \(m_b\) to the inner interface, and forwards every input at the inner interface to the outer interface.

A scheme is usually called IND-CPA secure if it is \((t, q, \ell, \varepsilon)\)-IND-CPA for all polynomials \(t\) and \(q\) and some negligible function \(\varepsilon\).

For all \(s \geq 2\), given a function \(F : \{0, 1\}^\kappa \times \{0, 1\}^n \rightarrow \{0, 1\}^\kappa\), we consider the (stateful) encryption scheme \((\text{ENC}_s(\cdot), \text{DEC}_s(\cdot))\) with perfect completeness and plaintext space \(\mathcal{M} = (\{0, 1\}^\kappa)^+\), where \(\text{ENC}(\cdot)\) operates as in Figure 6.3. It only requires \(\kappa + sn\) bits of memory (to keep \(\sigma\) and \(r_1, \ldots, r_s\), we neglect the space needed to store the counter) and approximately \(1 + \frac{1}{\varepsilon-1}\) calls to \(F\) per \(\kappa\)-bit block of encrypted data. An efficient stateful decryption algorithm is obtained by essentially running \(\text{ENC}(\cdot)\) on the ciphertext, where the counter component \(\text{cnt}\) is used to detect out of order messages.\(^7\)

The scheme is better understood as an instance of so-called countermode encryption: Given a PRF \(F' : \{0, 1\}^{\kappa'} \times \mathbb{N} \rightarrow \{0, 1\}^\kappa\), one obtains an efficient stateful IND-CPA encryption scheme (cf. [BDJR97] for a security proof) for arbitrary-length messages by using \(F'\) in so-called countermode, i.e. given a secret key \(k\), we keep a counter \(\text{ctr}\) (initially 0), and the encryption of a plaintext \(m\) (which we assume for simplicity to have

---

\(^6\)Many variants of this notion exist. We refer the reader to the works of Bellare et al. [BDJR97] and Katz and Yung [KY00] for further details on security notions for symmetric encryption.

\(^7\)A drawback of our approach is that efficient decryption also requires a stateful algorithm, which one would possibly want to avoid. We point out that other modes of operation, such as CBC encryption, also require stateful decryption.
Construction $\text{ENC}_s(k, r_1, \ldots, r_s)$:

$F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$, $k \in \{0, 1\}^n$, $r_1, \ldots, r_s \in \{0, 1\}^n$

\begin{verbatim}
upon initialization do:
    σ := k // setting up state
    cnt := 0 // counter
upon receiving a plaintext $m \in (\{0, 1\}^\kappa)^+$ do: // encryption of $m$
    for all $i = 1, \ldots, |m|/\kappa$ do
        $y_i := F(σ, r_{2+(\text{cnt mod } (s-1))})$
        cnt := cnt + 1
        if cnt ≡ 0 (mod s - 1) then
            $σ := F(σ, r_1)$
    return [cnt, $m \oplus y_1 \parallel \ldots \parallel y_{|m|/\kappa}$]
\end{verbatim}

Figure 6.3: Encryption procedure for the scheme $\text{ENC}(\cdot)$ and $\text{DEC}(\cdot)$ based on an $s$-RI-PRF $F$.

length $|m|$ multiple of $\kappa$ is the ciphertext

$[\text{ctr}, m \oplus (F'(k, \text{ctr})\|F'(k, \text{ctr} + 1)\| \ldots \| F'(k, \text{ctr} + |m|/\kappa - 1))]$

(and $\text{ctr}$ is increased by $|m|/\kappa$). Note in particular that we need one call to $F'$ for each $\kappa$-bit block of encrypted data.\(^8\) Security of our scheme follows by observing that it corresponds to using the RC construction for $F'$ with the prefix-free encoding scheme $\text{CTRENC} : \mathbb{N} \rightarrow \{1, \ldots, s\}^+$ defined as

$\text{CTRENC}(i) := i \text{ div } s - 1 || (2 + (i \text{ mod } s - 1))$.

The tree arising from this encoding scheme is illustrated in Figure 6.2.

Furthermore, the values $r_1, \ldots, r_s$ can be chosen publicly by one communicating party (provided an authenticated channel is available), hence

\(^8\)Variants of randomized stateless counter-mode encryption (where one chooses a fresh random counter at every encryption instead of keeping a state) based on any RI-PRF $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ were presented in [DN02, MS07]. As with a full PRF, these schemes only require one call per $n$-bit block of encrypted data, but the underlying RI-PRF must be secure against as many queries as the amount of encrypted message blocks.
reducing the cost of key establishment to the generation of the \(\kappa\)-bit private part of the key. Extension to security against (adaptive) chosen-ciphertext attacks based on any \(s\)-RI-PRF can be then obtained by standard techniques appending a MAC of the ciphertext [BN00] (e.g. using any of the PRF constructions presented in this chapter).

6.1.5. Application II: Pseudorandom Generators from \(s\)-RI-PRFs

We show that the RC construction yields PRG constructions from RI-PRFs which improve on previous results. We omit the proofs of the technical claims (which are mostly corollaries of Theorem 6.1 or are based on standard techniques). Surprisingly, constructing a good PRG from a RI-PRF (or an \(s\)-RI-PRF) turns out not to be a straightforward task: In contrast to PRFs, a RI-PRF \(F\) does not generally allow to find few “good” inputs \(x_1, \ldots, x_t\) such that the mapping

\[ k \mapsto F(k, x_1) \parallel \ldots \parallel F(k, x_t) \]

is a PRG. However, one can use this approach employing the RC construction as the underlying PRF: For any \(t\) fixed inputs \(x_1, \ldots, x_t\) (\(t > 2\)), we define the mapping \(G(F) : \{0, 1\}^{s \cdot n + t \cdot \kappa} \rightarrow \{0, 1\}^{s \cdot n + t \cdot \kappa}\) such that

\[ G(F)(k, r_0, r_1) \parallel \cdots \parallel G(F)(k, r_s, k) \]

is a PRG if \(F\) is an \(s\)-RI-PRF. (The order of the strings in the concatenation is irrelevant.) Note that an important advantage is that the \(s\) \(n\)-bit strings \(r_1, \ldots, r_s\) can be output as well. For example, given a 2-RI-PRF \(F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n\), the mapping \(G(F) : \{0, 1\}^{3n} \rightarrow \{0, 1\}^{6n}\) such that \(G(F)(k, r_0, r_1)\)

\[ r_0 \parallel F(F(k, r_0), r_0) \parallel F(F(k, r_0), r_1) \parallel F(F(k, r_1), r_0) \parallel F(F(k, r_1), r_1) \parallel r_1 \]

is a length-doubling PRG which requires 6 calls to \(F\). In particular, 3 calls are necessary in order to input only one both halves of the output. This improves a construction given in [MS07], which needed 3 and 4 calls, respectively.

An alternative approach to building a PRF from an \(s\)-RI-PRF \(F\) would consist of first constructing a length-doubling PRG \(G\) from \(F\), and subsequently using the well-known GGM construction [GGM84] to build a PRF with a \(\kappa\)-bit key and \(\ell\)-bit inputs by outputting, on input \(x = (x_1, \ldots, x_{\ell-1}, x_\ell) \in \{0, 1\}^\ell\) and key \(k\), the \(\kappa\)-bit value

\[ G_{x_\ell}(G_{x_{\ell-1}}(\cdots G_{x_1}(k) \cdots)), \]
where \( G_i(k) \) for \( i = 0, 1 \) gives the first and the second half of the output of \( G \), respectively. However, it is not hard to see that all constructions following this approach turn out to be less efficient than using the RC construction directly (e.g., using the above PRG one needs 3 calls of \( F \) per input bit).

### 6.2. The Nested Randomized Cascade Construction

Even though the RC construction can be practically efficient in special instantiation scenarios discussed earlier, its throughput is a major bottleneck in the case where the construction is used as a PRF (or a MAC) which is invoked at arbitrary inputs with variable lengths. Furthermore, the prefix-free encoding can be a limiting factor in the arbitrary-input-length case. One possibility to mitigate this for long messages (i.e., longer than \( \kappa \) bits) is to use domain extension techniques where initially an almost-universal hash function [CW79, Sti91] is applied to the message, and the output is processed by a fixed-input-length variant of the RC construction. This section presents a construction following this approach, where the almost-universal hash function is also implemented from the same underlying \( \kappa \)-RI-PRF. Its core ingredient is a novel use of pairwise independence.

**PAIRWISE-INDEPENDENT MAPPINGS.** First recall that a mapping\(^9\) \( M : \{0, 1\}^\kappa \times \{0, 1\}^m \to \{0, 1\}^n \) is pairwise independent if, under a uniform random \( \kappa \)-bit key \( K \), the values \( M(K, x) \) and \( M(K, x') \) are independent and uniformly distributed for all distinct \( x, x' \in \{0, 1\}^m \). Most pairwise-independent mappings satisfy the following property, which will be central in our construction.

**Definition 6.3.** A pairwise-independent mapping \( M : \{0, 1\}^\kappa \times \{0, 1\}^m \to \{0, 1\}^n \) is **key programmable** if there exists a (possibly randomized) algorithm SAMPLE which on input \((x, x', y, y')\) (where possibly \( x = x', y = y' \)) returns a uniformly chosen element from the set

\[
\{ k \mid M(k, x) = y, M(k, x') = y' \}.
\]

\(^9\)Of course, we could just replace the first phase by some existing construction of a (almost) universal hash function. However, our goal is to present a solution which only relies on a compressing RI-PRF, and which is amenable to being instantiated from existing primitives such as hash functions, as we discuss below.

\(^{10}\)We use the word mapping, rather than hash function, to stress the fact that \( m = n \) may also hold.
If $M$ is key programmable, the following two random experiments are equivalent to sampling a random $\kappa$-bit key $K$: (i) For some $n$-bit string $x$, sample $Y$ as a uniform random $n$-bit string and $K \overset{\$}{\leftarrow} \text{SAMPLE}(x, x, Y, Y)$; and (ii) For $n$-bit strings $x \neq x'$, sample $Y, Y'$ as independent random $n$-bit strings and $K \overset{\$}{\leftarrow} \text{SAMPLE}(x, x', Y, Y')$. Both the last two sampling strategies are used to ensure that $M(K, x) = Y$ (and possibly $M(K, x') = Y'$) for values $Y, Y' \in \{0, 1\}^n$ which, although uniform and independent, are provided externally.

We provide two examples of key-programmable pairwise-independent mappings.

**Example 6.1.** Let $M$ be such that given $k_1, k_2 \in \{0, 1\}^n$ and the input $x \in \{0, 1\}^n$, the output $M(k_1||k_2, x)$ equals $k_1 \oplus (k_2 \odot x)$, where $\oplus$ and $\odot$ are addition and multiplication of $n$-bit strings interpreted as elements of the extension field $GF(2^n)$. The unique $k_1||k_2$ such that $M(k_1||k_2, x) = y$ and $M(k_1||k_2, x') = y'$ (with $x \neq x'$) can efficiently be found solving the corresponding system of two equalities. Is only a single constraint $M(k_1||k_2, x) = y$ given, one chooses a random $n$-bit string $k_2$ and sets $k_1 := (k_2 \odot x) \oplus y$.

**Example 6.2.** An alternative is the mapping $M'$ whose $(nm + n)$-bit key consists of an $(m \times n)$-binary matrix $A$ and of an $n$-dimensional binary column vector $b$, and on input $x$ the output is $Ax + b$, where $x$ is interpreted as an $nm$-dimensional column vector, and addition and multiplications are modulo 2. The function $M'$ needs a larger key than $M$ described above, but avoids finite-field multiplications.

**Universal Hashing.** Also recall that for $\delta : \mathbb{N} \rightarrow \mathbb{R}^+$, we say that a function $H : \{0, 1\}^{\kappa} \times \{0, 1\}^* \rightarrow \{0, 1\}^n$ is $\delta$-almost universal (\(\delta\text{-AU}\)) if

$$p_H^{\text{COLL}}(x, x') := P[H(K, x) = H(K, x')] \leq \delta(\max\{|x|, |x'|\})$$

for all distinct $x, x' \in \{0, 1\}^*$, where $K$ is a uniformly chosen $\kappa$-bit key.\(^{11}\) The following lemma will be useful. (We omit its fairly standard proof.)

**Lemma 6.3.** Let $H : \{0, 1\}^{\kappa_1} \times \{0, 1\}^* \rightarrow \{0, 1\}^\kappa$ be $\delta$-AU, and let $F' : \{0, 1\}^{\kappa_2} \times \{0, 1\}^\kappa \rightarrow \{0, 1\}^\kappa$ be a keyed function. For all distinguishers $D \in \mathcal{D}_{\ell, n, \ell}$, there exists a distinguisher $D'$ such that

$$\Delta^D(H(K_1, \cdot) \triangleright F'(K_2, \cdot), R) \leq \Delta^{D'}(F'(K_2, \cdot), R') + \frac{1}{2} \cdot q^2 \cdot \delta(\ell),$$

\(^{11}\)This extends the standard notion [CW79, Sti91] to deal with arbitrary input lengths by letting $\delta$ be a function of the message length. Alternatively, one could use the notion of a cAU-hash function from [Bel06], but this will not be necessary here.
and where $K_1$ and $K_2$ are independent uniform $\kappa_1$- and $\kappa_2$-bit keys, respectively, whereas $D : \{0, 1\}^* \rightarrow \{0, 1\}^\kappa$ and $R : \{0, 1\}^\kappa \rightarrow \{0, 1\}^\kappa$ are URFs. Moreover, $D'$ makes $q$ queries and has running time $t' = t + q \cdot t_H(\ell)$, where $t_H(\ell)$ is the time needed to evaluate $H$ on inputs of length at most $\ell$.

**Construction.** The main idea of the nested RC construction (NRC, for short) is to combine an iterated phase where blocks are processed at a higher rate (but which satisfies a property weaker than pseudorandomness) with a second phase where the $RC_{s,\kappa}$ construction (for fixed input length $\kappa$ and a parameter $s$) is invoked on the output of the first phase (with independent key material).

Let $M : \{0, 1\}^{\kappa'} \times \{0, 1\}^m \rightarrow \{0, 1\}^n$ be a key-programmable pairwise-independent mapping, and let $F : \{0, 1\}^\kappa \times \{0, 1\}^n \rightarrow \{0, 1\}^\kappa$. The construction $PI_M(F) : \{0, 1\}^{\kappa + \kappa'} \times \{0, 1\}^* \rightarrow \{0, 1\}^\kappa$ takes a key $k \parallel k'$, where $k \in \{0, 1\}^\kappa$ and $k' \in \{0, 1\}^{\kappa'}$. On input $x \in \{0, 1\}^*$, it pads $x$ as $(x_1, \ldots, x_\lambda)$, where $x_1, \ldots, x_\lambda \in \{0, 1\}^m$, and outputs

$$F^*(k, (M(k', x_1), \ldots, M(k', x_\lambda))).$$

Moreover, given the additional parameter $s$, we define the nested construction $NRC_{M,s}(F) : \{0, 1\}^{2\kappa + \kappa'} \times \{0, 1\}^{sn} \times \{0, 1\}^* \rightarrow \{0, 1\}^\kappa$ such that

$$NRC_{M,s}(F)(k_1 \parallel k_2 \parallel k', r_1 \parallel \ldots \parallel r_s, x) := RC_{s,\kappa}(F)(k_1, r_1 \parallel \ldots \parallel r_s, PI_M(F)(k_2 \parallel k', x)).$$

It is easy to verify that in order to process a message $x$, the construction needs totally $\lceil \frac{|x| + 1}{m} \rceil + \lceil \frac{\kappa}{\log s} \rceil$ calls to the underlying function $F$.

It is tempting to increase the throughput of the construction by choosing a mapping $M$ with $m$ much larger than $n$. However, all known constructions of pairwise-independent hash functions (in particular key-programmable ones) require keys twice as long as the input (rather than the output), and hence such an approach would entail a much longer key. In fact, we believe the length-preserving mapping $M$ presented above to be a viable practically efficient solution: This special case of the construction is depicted in Figure 6.4.

---

---

12 According to the canonical padding which pads a string $x$ to have length being a multiple of $m$ by appending a 1 and sufficiently many 0’s: The resulting padded string consists hence of $\left\lceil \frac{|x| + 1}{m} \right\rceil$ $m$-bit blocks.
6.2 The Nested Randomized Cascade Construction

Figure 6.4: The construction $\text{NRC}_{M,s}^F$ for the special case $M(k_a\|k_b, x) = (k_a \odot x) \oplus k_b$.

**Security.** We fix $s \geq 2$, a key-programmable pairwise-independent mapping $M : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}^n$, and a function $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$. Also, we let $\text{NRC} = \text{NRC}_{M,s}(F)$ and $\text{PI} = \text{PI}_M(F)$. In addition, $\mathbf{R} : \{0, 1\}^* \rightarrow \{0, 1\}^*$ and $\mathbf{R}' : \{0, 1\}^n \rightarrow \{0, 1\}^n$ are both URFs. The following theorem precisely quantifies the security of the NRC construction.

**Theorem 6.4.** For all distinguishers $D \in \mathcal{D}_{t,q,\ell}$, there exist distinguishers $D'$ and $D''$ such that

$$
\Delta^D(\text{NRC}(K', R, \cdot), \mathbf{R}) \leq \left(1 + q \left(\left\lceil \frac{\kappa}{\log s} \right\rceil - 1\right)\right) \cdot \left(\Delta^D \mathbf{K}(F(K', \cdot), \mathbf{R'}) + s^2 \cdot 2^{-(n+1)}\right) + \left\lceil \frac{t+1}{m} \right\rceil \cdot q^2 \cdot \left(\Delta^{D''} \mathbf{K}(F(K', \cdot), \mathbf{R'}) + 2^{-n}\right) + q^2 \cdot 2^{-(\kappa+1)},
$$

where $K'$ is a random $(2\kappa + \kappa')$-bit key, $R$ is a random $s\kappa$-bit key, and $K$ is a random $\kappa$-bit key. Furthermore, $D'$ makes $s$ queries and has running time $t' = t + \mathcal{O}(q(\frac{\kappa}{m} + \frac{\kappa}{\log s}) \cdot t_F)$, whereas $D''$ makes two queries and has running time $t'' = \mathcal{O}(\frac{q^2 \kappa}{m} \cdot t_F)$. (Here $t_F$ denotes the time needed for an evaluation of $F$.)

The core of the proof consists of showing that whenever $F$ is a RI-PRF for two-query adversaries, the PI construction is $\delta$-AUs for a suitable function $\delta$. Given $x$ and $x'$ with corresponding padded strings $(x_1, \ldots, x_\lambda)$ and $(x_1', \ldots, x_{\lambda'})$ (where without loss of generality $\lambda < \lambda'$), let $\lambda^*$ be maximal with the property that $x_1 = x_1', \ldots, x_{\lambda} = x_{\lambda}'$, (in particular, $\lambda^* := 0$ if $x_1 \neq x_1'$), and define the quantity $\Lambda(x, x')$ as $\lambda + \lambda' - \lambda^* - 1$ if $(x_1, \ldots, x_{\lambda})$ is not a prefix of $(x_1', \ldots, x_{\lambda'})$, and as $\lambda + 1$ otherwise. Note that $\Lambda(x, x') \leq \lambda + \lambda' \leq 2\max\{\lambda, \lambda'\} \leq 2\left\lceil \frac{\kappa+1}{m} \right\rceil$ if $|x|, |x'| \leq \ell$. 
The following lemma provides a precise upper bound on the collision probability of the PI construction in terms of the RI-PRF distinguishing advantage of a distinguisher \( D_{x,x'} \) (which in particular only depends on \( x \) and \( x' \)) for \( F \). Its proof is deferred to Section 6.2.1.

**Lemma 6.5.** For all distinct inputs \( x, x' \in \{0,1\}^* \), there exists a two-query distinguisher \( D'_{x,x'} \) such that
\[
P_{\Pr}^{\text{COLL}}(x,x') \leq \Lambda(x,x') \cdot \left( \Delta_{D'_{x,x'}}^{\Sigma}(F(K,\cdot),R') + 2^{-n} \right) + 2^{-\kappa},
\]
where \( D'_{x,x'} \) has running time \( O(\Lambda(x,x') \cdot t_F) \) and makes two queries.

In particular, given some \( \ell \), let \( x, x' \) be strings with \( |x|, |x'| \leq \ell \) maximizing \( \Delta_{D'_{x,x'}}^{\Sigma}(F(K,\cdot),R') \), and set \( D'' := D'_{x,x'} \). Then \( D'' \) has running time \( t'' = O\left(\frac{2\ell+1}{m} \cdot t_F\right) \). We define
\[
\delta(\ell) := 2 \left\lceil \frac{\ell+1}{m} \right\rceil \cdot (\Delta_{D'_{x,x'}}^{\Sigma}(F(K,\cdot),R') + 2^{-n}) + 2^{-n}.
\]

The function \( \Pi \) is \( \delta \)-universal by Lemma 6.5, and this implies Theorem 6.4 using Lemma 6.3 and Theorem 6.1.

### 6.2.1. Proof of Lemma 6.5

The proof follows the lines of [Bel06], with some extra difficulties due to the use of RI-PRFs instead of PRFs. We fix two inputs \( x, x' \in \{0,1\}^* \) and we let \( (x_1, \ldots, x_\lambda) \) and \( (x'_1, \ldots, x'_{\lambda'}) \) be their corresponding paddings. Without loss of generality, assume that \( \lambda \leq \lambda' \), and let \( \Lambda := \Lambda(x,x') \) and \( \lambda^* \) be defined as above. The proof considers two distinct cases:

1. \((x_1, \ldots, x_\lambda)\) is not a prefix of \((x'_1, \ldots, x'_{\lambda'})\),

2. \((x_1, \ldots, x_\lambda)\) is a prefix of \((x'_1, \ldots, x'_{\lambda'})\).

For the proof, we also neglect the time complexity of \( \text{SAMPLE} \) (as it is minor with respect to evaluating \( F \)). Also for convenience, we denote \( \text{SAMPLE}(x,x,y,y) \) as \( \text{SAMPLE}(x,y) \).

**Case 1.** Similarly to Theorem 6.1, one proves that both outputs are pseudorandom, and hence can only collide with low probability. It is convenient to define the tuple \((m_1, \ldots, m_{\lambda+\lambda'-\lambda^*})\) such that
\[
m_i := \begin{cases} 
x_i & \text{if } i \leq \lambda \\
x'_{i-\lambda+\lambda^*} & \text{if } i > \lambda
\end{cases}
\]
for all $i = 1, \ldots, \lambda + \lambda' - \lambda^*$. Moreover, we associate $\Lambda$ integer labels 
$\text{label}(m_1, \ldots, m_j) \in \{0, \ldots, \Lambda - 1\}$ with all prefixes $(m_1, \ldots, m_j)$, with 
$j \in \{0, \ldots, \lambda + \lambda' - \lambda^* - 1\} \setminus \{\lambda\}$ in increasing order. Furthermore, for all 
$j = 1, \ldots, \lambda + \lambda' - \lambda^*$ we let 
\[
\text{parent}(m_1, \ldots, m_j) := \begin{cases} 
(m_1, \ldots, m_{j-1}) & \text{if } j \neq \lambda + 1 \\
(m_1, \ldots, m_{\lambda'}) & \text{if } j = \lambda + 1.
\end{cases}
\]

We define $\Lambda$ hybrid constructions $H_0(\cdot), H_1(\cdot), \ldots, H_{\Lambda-1}(\cdot)$ where, for all $i = 0, \ldots, \Lambda - 1$, given access to $K(F)$ (for a random function $F : \{0, 1\}^n \to \{0, 1\}^n$), the system $H_i(K(F))$ first sets $k'$ as follows: If $i = \lambda^*$, 
it samples $k' \leftarrow \text{SAMPLE}(m_{i+1}, m_{\lambda+1}, r_1, r_2)$ and lets $y(m_{i+1}) := y_1$, 
y(m_{\lambda+1}) := y_2$, whereas if $i \neq \lambda^*$, then it just sets $k' \leftarrow \text{SAMPLE}(m_{i+1}, r_1)$ 
and $y(m_{i+1}) := y_1$. Then, it assigns $\kappa$-bit values $\text{value}(m_1, \ldots, m_j)$ to all 
$(m_1, \ldots, m_j)$ for $j = 0, \ldots, \lambda + \lambda' - \lambda^*$. First, $\text{value}(\bot)$ is set to a uniform 
random string, and otherwise $\text{value}(m_1, \ldots, m_j)$ is set depending on the label $l$ of $\text{parent}(m_1, \ldots, m_j)$. 
If $l < i$, it is assigned a uniform random value, if $l > i$, it is assigned $F(v, M(k', m_j))$ (with $v$ being the value of 
the parent), and if $l = i$, it is assigned $y(m_j)$. Finally, the system returns 
$(\text{value}(m_1, \ldots, m_\lambda), \text{value}(m_1, \ldots, m_{\lambda+\lambda'-\lambda^*}))$

By inspection, we see that, for a $\kappa$-bit random $K$ and a $\kappa'$-random $K'$
\[
H_0(K(F(K, \cdot))) \equiv (\text{PI}(K\|K', x), \text{PI}(K\|K', x')) \\
H_{\Lambda-1}(K(B)) \equiv (R_1, R_2),
\]

where $B$ is a $\kappa$-bit beacon, and $R_1, R_2$ are independent uniform $\kappa$-bit 
strings. The first equivalence holds because the sampling of $k'$ through 
the sampling algorithm (and with $r_1, r_2$ being uniform and independent) 
is equivalent to sampling a random $k'$ and letting $y(m_{i+1}) := M(k', m_{i+1})$ 
(and possibly $y(m_{\lambda'+1}) := M(k', m_{\lambda'+1})$). Moreover, for the second 
equivalence, all assigned values are independent and uniform. In 
addition, as in the proof of Theorem 6.1, we can infer that 
\[
H_i(K(B)) \equiv H_{i+1}(K(F(K, \cdot))).
\]

Consider now the distinguisher $D_{\text{coll}}$ which, given two $\kappa$-bit strings $(y_1, y_2)$ 
outputs 1 if and only if $y_1 = y_2$, and 0 otherwise. Furthermore, let 
$D'_i := D_{\text{coll}} H_i(\cdot)$ for $i = 0, \ldots, \Lambda - 1$. Clearly, by the above, 
\[
P [D'_0(K(F(K, \cdot))) = 1] = p^{\text{coll}}_{\text{PI}}(x, x') \\
P [D'_{\Lambda-1}(K(B)) = 1] = 2^{-\kappa}.
Finally, we define $D'_{x,x'}$ as the distinguisher running $D'_i$ for a uniformly chosen $i \in \{0, \ldots, \Lambda - 1\}$. This yields (for $F := K(F(K, \cdot))$ and $I := K(B)$)

$$\Delta^{D'_{x,x'}}(F, I) = \left| P[D'_{x,x'}(F) = 1] - P[D_{x,x'}(I) = 1] \right|$$

$$= \frac{1}{\Lambda} \sum_{i=0}^{\Lambda-1} \left| P[D'_i(F) = 1] - P[D'_i(I) = 1] \right|$$

$$= \frac{1}{\Lambda} \left| P[D'_0(K(F(K, \cdot))) = 1] - P[D'_{\Lambda-1}(K(B)) = 1] \right|$$

$$\geq \frac{1}{\Lambda} \left[ p_{\text{coll}}^{\Lambda}(x,x') - 2^{-n} \right],$$

and the desired bound follows from the fact that

$$\Delta^{D'_{x,x'}}(F, I) \leq \Delta^{D'_{x,x'}}(K(F(K, \cdot)), K(R')) + \Delta_2(R', B),$$

where the last summand is bounded by $2^{-n}$.

**Case 2.** For all $i = 1, \ldots, \lambda + 1$ we let $H_i(\cdot)$ be the hybrid construction that first obtains two pairs $(r_1, y_1), (r_2, y_2)$ from the given system $K(F)$, and samples $k' \leftarrow \text{SAMPLE}(x'_i, r_1)$. Then, it sets $\text{value}(x'_1, \ldots, x'_{i-1}) := \perp$, $\text{value}(x'_1, \ldots, x'_i) := y_1$, and

$$\text{value}(x'_1, \ldots, x'_j) := F(\text{value}(x'_1, \ldots, x'_{j-1}), M(k', x'_j))$$

for all $j = i + 1, \ldots, \lambda'$. Finally, it outputs

$$(\text{value}(x'_1, \ldots, x'_{\lambda'}), \text{value}(x'_1, \ldots, x'_{\lambda'}), (r_2, y_2)).$$

We let $D'_\text{coll}$ be such that, given a triple $(y, y', (r_2, y_2))$, it returns 1 if and only if $y = y'$. Then, define $D'_i := D'_\text{coll} H_i$ for all $i = 1, \ldots, \lambda$. By inspection,

$$P[D'_i(K(F(K, \cdot))) = 1] = p_{\text{coll}}^{\lambda}(x, x'),$$

and additionally for all $i = 1, \ldots, \lambda - 1$,

$$H_i(K(B)) = H_{i+1}(K(F(K, \cdot))),$$

which implies $P[D'_i(K(B)) = 1] = P[D'_{\lambda+1}(K(F(K, \cdot))) = 1]$.

Also let $D'_{\text{col}}$ be the distinguisher which on input $(y, y', (r_2, y_2))$ computes $F(y', r_2)$ and returns 1 if and only if it equals $y_2$, and 0 otherwise. We define $D'_{\lambda+1} := D'_{\text{coll}} H_{\lambda+1}$. Then,

$$P[D'_{\lambda+1}(K(B)) = 1] = 2^{-\kappa},$$
since the value $y_2$ is independent of $y'$ and $r_2$, and hence also of $F(y', r_2)$. Finally, we have
\[
P[D'_{\lambda}(K(B)) = 1] \leq P[D'_{\lambda+1}(K(F(K, \cdot))) = 1],
\]
as the probability that $y = y'$ in $H_{\lambda}(K(B))$ is the same as the probability that $K = y'$ in $H_{\lambda+1}(K(F(K, \cdot)))$, and this latter event implies that $D'_{\lambda+1}$ outputs one.

The final distinguisher $D'_{x, x'}$ picks $i \overset{\$}{\leftarrow} \{1, \ldots, \lambda + 1\}$, and runs $D'_i$. Then, with $F = K(F(K, \cdot))$ and $I = K(B)$,
\[
\Delta^{D'_{x, x'}}(F, I) = |P[D_{x, x'}(F) = 1] - P[D_{x, x'}(I) = 1]|
\geq \frac{1}{\lambda + 1} \sum_{i=1}^{\lambda+1} P[D'_i(F) = 1] - P[D'_i(I) = 1]
\geq \frac{1}{\lambda + 1} \left[p_{\text{COLL}}(x, x') - 2^{-\kappa}\right].
\]
Once again, the final bound is obtained using the triangle inequality and the fact that $\Delta_2(R', B) \leq 2^{-n}$.

### 6.3. Black-Box Keying of Iterated Hash Functions

In this final section, we turn to some practical aspects of the constructions of this chapter.

#### 6.3.1. Iterated Hash Functions and MACs

Recall that an iterated hash function [Mer89, Dam89] $H : \{0, 1\}^* \rightarrow \{0, 1\}^\kappa$ with underlying compression function $F : \{0, 1\}^\kappa \times \{0, 1\}^n \rightarrow \{0, 1\}^\kappa$ ($n$ is generally called the block length) and initialization value $IV \in \{0, 1\}^\kappa$ is defined such that every input $x \in \{0, 1\}^*$ is first padded as $(x_1, \ldots, x_\lambda) \in (\{0, 1\}^n)^+$ and subsequently the value $F^\kappa(IV, (x_1, \ldots, x_\lambda))$ is output. In general, the last block $x_\lambda$ contains some padding bits as well as the length of the message (the so-called MD-strengthening) to preserve collision resistance of the compression function. Examples of such functions are those from the MD and the SHA families.

Bellare et al. [BCK96a] proposed two constructions of message authentication codes\footnote{Informally, a MAC is a cryptographic function with a secret key which is “unpredictable” (also see below for a formal definition). While a PRF is a MAC, the latter notion is strictly weaker than the former.} (MACs), called \textbf{HMAC} and \textbf{NMAC}, which are obtained
by appropriately keying an iterated hash function $H : \{0,1\}^\kappa \times \{0,1\}^* \rightarrow \{0,1\}^\kappa$ (where the first input is the initialization value) as

$$\text{HMAC}(H)(k_1 \parallel k_2, x) := H(IV, k_2 \parallel H(IV, k_1 \parallel x))$$

for a fixed known $IV$ and $|k_1|, |k_2|$ both equal to the block length of $H$, and as

$$\text{NMAC}(H)(k_1 \parallel k_2, x) := H(k_2, H(k_1, x)),$$

respectively.\(^\text{14}\) (Note that HMAC only requires black-box usage of $H$.) Despite alternative designs of MACs (such as CBC-MAC [BKR00] and UMAC [BHK+99]), these constructions have enjoyed widespread usage due to the large availability of hash function implementations (both in hardware and in software). From the theoretical standpoint, security of HMAC/NMAC has been first proved [BCK96a] under the assumption that the compression function of $H$ is a PRF (when keyed through the chaining value), and that $H$ is weakly collision resistant, i.e. it is hard to find two distinct messages $x, x'$ with $H(K, x) = H(K, x')$ for a secret key $K$ (given oracle access to $H(K, \cdot)$). Bellare [Bel06] subsequently proved HMAC/NMAC to be an arbitrary-input-length PRF under the sole assumption of the compression function being a PRF. We point out that the cascade construction by Bellare et al. [BCK96b] can also be seen as a way to key a hash function with a single key to obtain a PRF under the same assumption, at the expense of using a prefix-free encoding of the inputs. More recently, Fischlin [Fis08] presented security proofs for HMAC/NMAC (when used as a MAC rather than as a PRF) relying on non-malleability properties of the underlying compression function. A further recent line of research [HPY07, Yas07] has been concerned with increasing the efficiency of the HMAC/NMAC constructions by imposing slightly stronger requirements on the underlying compression function (i.e. pseudorandomness under mild types of related-key attacks).

The bottom line is that in order to deploy one of these constructions in practice, it is relevant to assess the level of confidence one is willing to put in the given compression function, but in view of continuous cryptanalytic achievements this is far from being a simple task. The results of this chapter can be seen as taking steps in the opposite direction: We have raised the question of constructing iterated MACs (and PRFs) with very low requirements on the given compression function, while guaranteeing limited impact on the performance when compared with constructions with stronger underlying security assumptions. In particular, we

\(^{14}\)Practical implementations usually consider single-keyed versions which, for simplicity, are not discussed here.
consider constructions which only require the underlying compression function to be an $s$-RI-PRF (for $s$ as small as two).

### 6.3.2. Instantiations from Hash Functions

The iterated structure of the RC and the NRC constructions makes compression functions ideal candidates for instantiating the underlying $s$-RI-PRF. In general, however, we may be constrained to only have black-box access to an implementation of an iterated hash function $H : \{0,1\}^* \rightarrow \{0,1\}^\kappa$ with direct access neither to the initialization value $IV$ nor to the underlying compression function $F : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^\kappa$. To overcome this obstacle, we encode (as in HMAC) an $n$-bit key as the first block of the input to the hash function $H$. More precisely, given the prefix-free encoding scheme $ENC : \{0,1\}^* \rightarrow \{1,\ldots,s\}^+$, we consider the construction $HRC_{s,ENC}$ which takes a key with private part $k \in \{0,1\}^n$ and public parts $r_1,\ldots,r_s \in \{0,1\}^n$, and on input $x$ such that $ENC(x) = (m_1,\ldots,m_\lambda)$ outputs the value

$$HRC_{s,ENC}(H)(k, r_1 \parallel \ldots \parallel r_s, x) := H(k || r_{m_1} || \ldots || r_{m_\lambda}),$$

and analogously we define $HRC_{s,\ell}$ for inputs of fixed-length $\ell$ (using the canonical encoding to the base $s$). Furthermore, with $M : \{0,1\}^{\kappa'} \times \{0,1\}^m \rightarrow \{0,1\}^n$ being a key-programmable pairwise-independent mapping, we consider the construction $HNRC_{M,s}$ which takes a key with private part $k_1, k_2 \in \{0,1\}^n$, $k' \in \{0,1\}^{\kappa'}$ and public parts $r_1,\ldots,r_s$. On input input $x$ (padded as $(x_1,\ldots,x_\lambda)$) it outputs

$$HNRC_{M,s}(H)(k_1 \parallel k_2 \parallel k', r_1 \parallel \ldots \parallel r_s, x) := HRC_{s,\kappa'}(H)(k_1, r_1 \parallel \ldots \parallel r_s, H(k_2 \parallel M(k', x_1) \parallel \ldots \parallel M(k', x_\lambda))).$$

In order to lift the security statements of the RC and the NRC constructions to both the HRC and HNRC constructions, the assumption that $F$ is an $s$-RI-PRF is not sufficient: First, it is necessary that the $\kappa$-bit output $F(IV,K)$ is computationally indistinguishable from a uniformly distributed random string of length $\kappa$ (under a secret random $K$); This guarantees that the chaining value obtained after the first evaluation of $F$ is pseudorandom and can be used as the “key” for the RC- or the PI construction. A further problem is due to the fact that we generally cannot enforce the last $n$-bit block processed by $F$ to be random because of the padding introduced by $H$, and this issue should not destroy the pseudorandomness of the outputs. To our rescue, however, comes the fact
that each such block is processed keying $F$ with a fresh pseudorandom value: It is hence enough to additionally guarantee that for an arbitrary fixed $n$-bit string $x$ and a random secret $\kappa$-bit string $K$, the string $F(K, x)$ is computationally indistinguishable from a random $\kappa$-bit string.

Both these extra properties are very weak requirements: In fact, a good compression function should satisfy them even unconditionally. It is sufficient, for example, that $F(IV, \cdot)$ and $F(\cdot, x)$ (for all $x \in \{0, 1\}^n$) are all (nearly-)regular functions. (We refer the reader to [BK04] for a discussion on regularity-properties of hash functions.). With these two additional assumptions on the compression function $F$ of $H$, the security bounds of the RC and the NRC construction can be lifted to their black-box counterparts. We omit the proofs (which are very similar to the ones of the original constructions).
In this thesis, we have provided a general treatment of several facets of the problem of computational indistinguishability amplification.

With respect to the question of advantage amplification, our results yield an in-depth quantitative understanding of several constructions combining cryptographic systems. As an application of these results, many approaches to security amplification of computational-indistinguishability based symmetric-key primitives, such as PRGs, PRFs, and PRPs, are now fully understood. We also expect a number of further applications of our results to other settings and constructions.

Our bounds are, in most cases, tight, up to some additive negligible terms, for which we believe there is room for quantitative improvement. For example, for several results in Chapter 5, these terms arise from the application of Lemma 5.13. Although this lemma is very useful, in that it characterizes the input-output behavior of high-entropy random functions, it would not be surprising if much better results could be obtained by ad-hoc construction-specific analyses.

We also remark that the restriction to cc-stateless systems seems to be an inherent limitation of our proof techniques. This class is sufficiently general to comprise many cryptographic systems of interest, but
it is well conceivable that similar product theorems may hold for a larger class of systems, even though a proof may require considerably new techniques. We also point out that a somewhat orthogonal approach is taken in the study of parallel composition of interactive protocols [BIN97, PV07, PW07, Hai09, CL10, HPWP10], where the security reduction is limited to interaction in one single protocol instance with a pre-determined number of rounds, but these techniques do not apply to the study of composition of weak secret-key primitives. An important open problem is to unify these two lines of work to the greatest extent, developing a general understanding of secure composition of cryptographic systems.

Finally, even though an existing information-theoretic statement has often inspired us to study a corresponding computational analogue, we are not aware of an approach to translate the results of Chapter 5 back to the information-theoretic setting, with a tight security reduction.\(^1\) Our results heavily rely on relating computational indistinguishability to a system behaving as a high-entropy function table with good probability, but it is known that this approach cannot yield a tight information-theoretic reduction.

As for computational indistinguishability for the distinguisher class, we have shown that efficient arbitrary-input-length PRFs (and consequently MACs and encryption schemes) can be constructed under very weak assumptions, i.e., PRFs where indistinguishability only holds under a limited number of random queries. This feasibility of practical symmetric cryptographic under very weak assumptions.

We believe that our construction from RI-PRFs is optimal in terms of calls to underlying function. It is an interesting open problem to confirm this intuition by means of a lower bound proof for black-box reductions. In sharp contrast to the information-theoretic case [MP04, MPR07], we conjecture that no better construction exists even under the stronger assumption of a PRF which is indistinguishable under non-adaptive queries. This is partially confirmed by known impossibility results [Mye04, Pie05] for some natural constructions.

\(^1\)Of course, the computational result itself also applies to the information-theoretic setting. Yet, as in all other examples, we would expect a much better security reduction.


A.1. A Technical Lemma on Conditional Expectations

**Lemma A.1.** Let $f : [0, 1] \to [0, B]$ be monotone increasing, and let $R$ be uniformly distributed in $[0, 1]$. Let $\mathcal{E}_v$ be the event that $R \leq v$. Then for all $\alpha, \beta \in [0, 1],\n
|E[f(R) | \mathcal{E}_\alpha] - E[f(R) | \mathcal{E}_\beta]| \leq B \cdot \frac{\lvert \alpha - \beta \rvert}{\max \{\alpha, \beta\}}.

**Proof.** Assume without loss of generality that $\alpha \leq \beta$. Then, due to the monotonicity of $f$, we have $E[f(R) | \mathcal{E}_\alpha] \leq E[f(R) | \mathcal{E}_\beta]$, and

$$E[f(R) | \mathcal{E}_\beta] - E[f(R) | \mathcal{E}_\alpha] = \frac{1}{\beta} \int_0^\beta f(r) \, dr - \frac{1}{\alpha} \int_0^\alpha f(r) \, dr$$

$$= \frac{1}{\beta} \int_0^\beta f(r) \, dr + \left( \frac{1}{\beta} - \frac{1}{\alpha} \right) \int_0^\alpha f(r) \, dr$$

$$\leq \frac{1}{\beta} \int_0^\beta f(r) \, dr$$

$$\leq B \frac{\beta - \alpha}{\beta},$$

since $\frac{1}{\beta} - \frac{1}{\alpha} \leq 0$. \qed
A.2. Details of the Proof of Theorem 5.3

Proof of Claim 7. Let $S^+ \subseteq S$ be defined as the set of $s \in S$ such that $P_{X'B'}(s, 0) \leq P_{X''B''}(s, 0)$. Analogously, let $T^+ \subseteq T$ be the set of $t \in T$ such that $P_{X'B'}(t, 1) \leq P_{X''B''}(t, 1)$. Recall that

$$d((X', B'), (X'', B'')) = \sum_{s \in S^+} P_{X''B''}(s, 0) - P_{X'B'}(s, 0) + \sum_{t \in T^+} P_{X''B''}(t, 1) - P_{X'B'}(t, 1). \quad (A.1)$$

We only upper bound the first sum (i.e., over all $s \in S^+$), as the other sum can be bounded symmetrically. First, for all $s \in S$,

$$P_{X''B''}(s, 0) - P_{X'B'}(s, 0) = \frac{1}{2} \cdot P_S(s) \cdot \left[ \frac{M_S(s)}{\mu(M_S)} - \frac{M_0(s)}{\mu(M)} \right].$$

Moreover, it is convenient to rearrange terms as

$$M_S(s) = \frac{\varepsilon}{\varepsilon + 3\gamma \cdot (1 - \varepsilon)} M_0(s) + \frac{3\gamma \cdot (1 - \varepsilon)}{\varepsilon + 3\gamma \cdot (1 - \varepsilon)}$$

$$= M_0(s) + \frac{3\gamma \cdot (1 - \varepsilon)}{\varepsilon + 3\gamma \cdot (1 - \varepsilon)} \cdot (1 - M_0(s)).$$

Using $\mu(M_S) \geq \mu(M_0)$ and $\mu(M) \leq \frac{\mu(M_0)}{1 - 3\delta}$,

$$\frac{M_S(s)}{\mu(M_S)} - \frac{M_0(s)}{\mu(M)} \leq \frac{3\gamma}{\varepsilon + 3\gamma (1 - \varepsilon)} \cdot \frac{1 - M_0(s)}{\mu(M_S)} \leq \frac{3\gamma}{\varepsilon + 3\gamma (1 - \varepsilon)} \cdot (1 - M_0(s))$$

for all $s \in S$, where we have also used $\mu(M_S) \geq 1 - \varepsilon$ for the last inequality. Note in particular that the upper bound is always non-negative, and
this can be used to conclude that

\[
\sum_{s \in S^+} P_{X'Y'}(s, 0) - P_{X'B'}(s, 0) = \frac{1}{2} \sum_{s \in S^+} P_S(s) \cdot \left[ \frac{M_S(s)}{\mu(M_S)} - \frac{M_0(s)}{\mu(M_0)} \right]
\]

\[
\leq \frac{3\gamma}{2} \sum_{s \in S^+} P_S(s) \cdot \left[ \frac{M_0(s)}{\mu(M_0)} + \frac{1 - M_0(s)}{\varepsilon + 3\gamma(1 - \varepsilon)} \right]
\]

\[
\leq \frac{3\gamma}{2} \sum_{s \in S} P_S(s) \cdot \left[ \frac{M_0(s)}{\mu(M_0)} + \frac{1 - M_0(s)}{\varepsilon + 3\gamma(1 - \varepsilon)} \right]
\]

\[
= \frac{3\gamma}{2} \left[ 1 + \frac{1 - \mu(M_0)}{\varepsilon + 3\gamma(1 - \varepsilon)} \right] \leq 3\gamma,
\]

since

\[
1 - \mu(M_0) \leq 1 - (1 - 3\gamma)(1 - \varepsilon) = \varepsilon + 3\gamma(1 - \varepsilon).
\]