Diss. ETH No. 20907

Information-Theoretic Analyses of Symmetric-Cryptography Constructions in Idealized Models

A dissertation submitted to

ETH ZURICH

for the degree of
Doctor of Sciences

presented by

Peter Gaži
Mgr., Comenius University Bratislava

born April 12, 1983, in Bratislava
citizen of Slovak Republic

accepted on the recommendation of

Prof. Dr. Ueli Maurer, examiner
Prof. Dr. Phillip Rogaway, co-examiner

2012

Acknowledgments

First of all, I would like to thank my supervisor Ueli Maurer for being such an inspiring advisor. His dedication to research has always been contagious and I have benefited greatly from all the research discussions we had. Even though I haven't spent all my studies in Zurich, thanks to his hospitality I have always felt welcome during my research stays at his group.

I would also like to thank Phil Rogaway for kindly agreeing to serve as a co-examiner on my committee and for his extensive feedback that helped me improve this thesis.

My gratitude also goes to all the current and former members of the Information Security and Cryptography Research Group at ETH Zurich that have been my colleagues during the studies: Divesh Aggarwal, Joël Alwen, Kfir Barhum, Zuzana Beerliová, David Bernhard, Sandro Coretti, Grégory Demay, Maria Dubovitskaya, Robert Enderlein, Gian Pietro Farina, Martin Hirt, Simon Knellwolf, Christoph Lucas, Christian Matt, Jun Muramatsu, Arpita Patra, Dominik Raub, Pavel Raykov, Junji Shikata, Björn Tackmann, Stefano Tessaro, Daniel Tschudi and Vassilis Zikas. Thank you all for maintaining the relaxed atmosphere within the group and for the countless enjoyable discussions, being research-related or not. In particular, I would like to thank Stefano, Martin and Grégory for collaboration that led to some of the results in this thesis and I am also grateful to Björn for his help with the German abstract. My thank you goes also to our secretary Beate Bernhard, who was always ready to help with arranging my next visit.

Last but certainly not least, I would like to express my gratitude to all my dearest, my family and friends, for their unconditional support and understanding. Not only did they make my student life much more enjoyable, without their constant encouragement I would not have been able to finish this work.

Abstract

Viewing cryptography as a constructive discipline, most of its results can be expressed as constructions transforming some resources that are assumed to be available into a new resource that is needed for the task to be solved. Broadly, one can cluster these constructions into two categories. On one hand, there are *reductions* that transform the seemingly simpler resource that is available into a new, more complex or useful resource. On the other hand, there are *security-amplifying constructions* that transform one or more available resources with unsatisfactory security properties into a new one of the same type, however with significantly better security guarantees.

In this thesis we analyze several constructions of both these types, all belonging to the domain of symmetric cryptography. These include basic building blocks used in the design of both block ciphers and stream ciphers, various higher-level constructions that take a complete block cipher as their starting point, as well as methods to construct cryptographic hash functions from weaker primitives. All arguments presented in this thesis are purely information-theoretic, leading to results that assume no restrictions on the complexity of computations involved. Finally, we often use idealized models of cryptographic primitives (such as the random oracle, the ideal cipher, the uniformly random permutation) either as the starting points of our reductions (to focus on the security of the reductions themselves) or as the target goal for security-amplifying constructions.

The first part of the thesis studies constructions for combining several real resources to achieve amplification of their indistinguishability from an ideal resource. We build upon previous work in this area and consider the general class of so-called *neutralizing constructions* for which two separate forms of indistinguishability amplification were investigated before:

a *product theorem* and an *amplification of the distinguisher class*. In our work, we use additional insight into the structure of neutralizing constructions to unify the above results into one general statement that keeps the structure of a product theorem while also capturing the amplification of the distinguisher class, improving on previous bounds.

In the second part of the thesis, we investigate constructions for key-length extension of block ciphers in the ideal cipher model. We analyze the plain cascade construction of length ℓ and show that for $\ell \geq 3$ it provides a significant security amplification (increasing with ℓ for particular parameters) and also give a generic attack against plain cascades of arbitrary length. Then we study a modified cascade of length ℓ having XOR-randomizing steps between the encryption steps, called ℓ-XOR-cascade. We show that this construction achieves a better efficiency/security trade-off than plain cascades for the practical lengths $\ell = 2, 3, 4$; with its security also increasing with higher length ℓ. Finally, we also describe various generic attacks against classes of efficient key-length extension schemes, proving several of our security bounds tight.

Finally, the last part of the thesis is dedicated to the study of the notion of indifferentiability. Addressing recent doubts on the applicability of results in this framework, we introduce *memory-aware reducibility* as an instance of a general treatment of settings where a particular resource (in this case memory available to the adversary) has to be quantified more precisely than with the standard indifferentiability notion. This formalism allows us to lower-bound the simulator memory required for any domain extension construction for a public random function, showing that if we restrict to simulators without memory, even domain extension by a single bit becomes impossible; and for the infinite extension from an ideal compression function to a random oracle, a memory roughly linear in the length of the longest query is required. This implies irreducibility of the random oracle to the ideal cipher with respect to reductions with stateless simulators; as well as the impossibility of constructing a public random oracle from a public ideal compression function in any multi-party setting where we cannot assume a centralized adversary.

Zusammenfassung

Wenn man Kryptographie als eine konstruktive Disziplin versteht, können die meisten Resultate als Konstruktionen dargestellt werden, die gewisse als gegeben angenommene Ressourcen in neue Ressourcen transformieren. Diese Konstruktionen können grob in zwei Kategorien eingeteilt werden. Einerseits gibt es *Reduktionen*, die scheinbar einfachere gegebene Ressourcen in neue, komplexere oder besser benutzbare Ressourcen transformieren. Andererseits gibt es *sicherheitsverstärkende Konstruktionen*, die eine oder mehrere Ressourcen mit unbefriedigenden Sicherheitseigenschaften in eine neue Ressource transformieren, die zwar vom gleichen Typ ist, jedoch deutlich stärkere Sicherheitsgarantien gibt.

In dieser Arbeit analysieren wir mehrere derartige Konstruktionen, die alle in den Bereich der symmetrischen Kryptographie fallen. Einige betreffen grundlegende Bausteine, die im Design von Block- und Stromchiffren vorkommen, andere nehmen eine vollständige Blockchiffre als ihren Ausgangspunkt oder befassen sich mit Methoden, kryptographische Hash-Funktionen aus schwächeren Primitiven zu konstruieren. Alle in dieser Arbeit präsentierten Argumente sind informationstheoretisch. Wir verwenden häufig idealisierte Modelle kryptographischer Primitiven (wie Random Oracles, ideale Blockchiffren, gleichverteilt zufällige Permutationen) entweder als Ausgangspunkt unserer Reduktionen (um die Sicherheit der Reduktionen selbst zu beleuchten) oder als das Ziel für sicherheitsverstärkende Konstruktionen.

Der erste Teil der Arbeit untersucht Konstruktionen, die mehrere „reale“ Ressourcen kombinieren um eine Verstärkung der Ununterscheidbarkeit von einer idealisierten Ressource zu erreichen. Wir bauen auf früheren Arbeiten in diesem Gebiet auf und betrachten die Klasse sogenannter *neutralisierender Konstruktionen*, für die zuvor zwei unterschiedliche Arten von Verstärkungen der Ununterscheidbarkeit untersucht wurden: Ein *Produktsatz* und eine *Vergrösserung der Klasse von Unterscheidern*.

In dieser Arbeit verwenden wir weitere Einsichten bezüglich der Struktur neutralisierender Konstruktionen um die obigen Resultate zu einem allgemeineren Resultat zu vereinheitlichen, das die Struktur des Produktsatzes beibehält und gleichzeitig die Vergrösserung der Unterscheiderklasse beinhaltet, sowie die bekannten Schranken verbessert.

Im zweiten Teil der Arbeit untersuchen wir Konstruktionen zur Erweiterung der Schlüssellänge im Modell der idealen Blockchiffren. Wir analysieren die „einfache" Kaskade von Länge ℓ and zeigen, dass sie für $\ell \geq 3$ eine deutliche Verstärkung der Sicherheit erreicht (für gewisse Parameter wächst die Verstärkung mit ℓ) und zeigen auch einen generischen Angriff gegen derartige Kaskaden jeglicher Länge. Dann untersuchen wir eine modifizierte Kaskade von Länge ℓ, bei der die Werte zwischen den Chiffrierungen über XOR (exklusives Oder) mit Teilen des Schlüssels randomisiert werden, die ℓ-XOR-Kaskade. Wir zeigen, dass diese Konstruktion für praktikable Längen $\ell = 2, 3, 4$ einen besseren Ausgleich zwischen Effizienz und Sicherheit ermöglicht als die einfache Kaskade. Auch hier verbessert sich die Sicherheit für grössere Längen ℓ. Schliesslich beschreiben wir mehrere generische Angriffe gegen Klassen effizienter Schemata zur Erweiterung der Schlüssellänge und beweisen somit die Optimalität von mehreren unserer Schranken.

Der letzte Teil der Arbeit ist dem Begriff der Indifferentiability gewidmet. Wir begegnen kürzlich aufgekommenen Zweifeln an der Einsetzbarkeit dieses Frameworks indem wir *speichergewahre Reduzierbarkeit* einführen als spezifische Instanz des allgemeinen Konzepts, eine bestimmte Ressource (in diesem Fall Speicher, der dem Gegner zur Verfügung steht) präziser zu quantifizieren als beim allgemeinen Begriff der Indifferentiability. Im Falle der Bereichserweiterung für öffentliche Zufallsfunktionen können wir somit eine untere Schranke für die Speichermenge angeben, die ein entsprechender Simulator benötigt. Wenn wir insbesondere nur Simulatoren ohne Speicher betrachten, wird sogar die Erweiterung des Bereichs um ein einzelnes Bit unmöglich. Ebenso zeigen wir, dass für die unbeschränkte Erweiterung einer idealen Kompressionsfunktion zu einem Random Oracle eine Speichermenge linear in der Länge der längsten Anfrage benötigt wird. Dies impliziert, dass ein Random Oracle nicht auf eine ideale Blockchiffre reduziert werden kann, wenn man sich auf zustandslose Simulatoren beschränkt. Ebenso ist es unmöglich, ein Random Oracle in solchen Szenarien aus einer öffentlichen idealen Kompressionsfunktion zu konstruieren, in denen mehrere unabhängige Teilnehmer existieren und wir nicht von Angriffen durch einen zentralen Gegner ausgehen können.

Contents

Chapter 1

Introduction

1.1 Cryptography as a Constructive Discipline

Cryptography is a very old domain with its existence justified as far into the past as there were any secrets to communicate over an insecure environment. Being more of an art than a science, classical cryptography was concerned solely with the confidentiality of communicated messages, designing more and more complex encryption procedures, followed closely by the development of methods to analyze and break these encryptions.

This paradigm has gradually changed during the 20th century due to various reasons. First of all, on the theoretical side, several individual breakthroughs resulted into seminal contributions that allowed to base cryptography on the firm grounds of information theory [Sha48] and discover new tools that can even today puzzle a layman's intuition by their mere existence [DH76, GMR89]. On the practical side, the construction of first computers and the later proliferation of available computational power into our daily lives made the fruits of modern cryptography accessible to everyone. These new tools also led to new goals: cryptography is no longer only concerned with message confidentiality, it has evolved into a broad field studying ways to achieve authenticity, privacy, anonymity and even more complex objectives such as performing arbitrary computations among several parties securely even without mutual trust.

There is one aspect of cryptography that can be traced to its early beginnings and has become all the more relevant with the formalization

of this field: as argued in [Mau11], it is a constructive discipline. From designing simple ad-hoc encryption schemes of the past to complex provably secure protocols of today; the goal is always to develop tools suitable for solving a particular task, by starting only with resources that are assumed to be available and employing them in carefully designed constructions.

1.1.1 Reductions and Security Amplification

Irrespective of the particular constructive discipline considered, one can – loosely speaking – identify several reasons for attempting to construct a new object on top of the resources that are already available. First, if a given resource is not directly suitable for the task to be solved, it needs to be transformed into a fitting resource. Second, even if the resource at hand is already suitable for the job, one might want to improve its performance at this task, arriving at a higher-quality tool. These two types of constructive tasks translate into the context of cryptography as follows.

Reductions. The first described scenario corresponds to the fundamental notion of a *reduction*: one investigates ways how to reduce the need for a (complex) primitive $\mathbf{T}$ to the need for a (simpler) primitive $\mathbf{S}$; or, dually, how to construct $\mathbf{T}$ given only $\mathbf{S}$. To focus on the security properties of the reduction itself, one often assumes that the starting ingredients represented by $\mathbf{S}$ are in some sense *ideal* (e.g. a uniformly random string, an ideal compression function), in contrast to the building block that is then used in practice (e.g. a pseudo-random string, a real compression function).

For a reduction, the simplest setting to consider is if the resource $\mathbf{S}$ is *private*, i.e., available solely to the party applying the construction. For example, one can apply a pseudo-random generator to a private random string (used as the seed) to obtain a longer pseudo-random string. On the other hand, one often has to consider reductions in the more complex *public* setting[1] where the resource $\mathbf{S}$ is accessible by everyone (even more generally, it can of course provide different functionalities to different parties). As an illustration of this setting, consider the task of constructing a random oracle from a (publicly accessible) random compression function.

[1] Note that this distinction is unrelated to the classification of cryptography into private-key and public-key.

A very natural but often neglected requirement for all reductions is to be composable: if one devises a certain secure way to construct **T** from **S** then it is desirable that this operation is secure also as a part of a more complex construction in an arbitrary context. Such composability is usually obtained for free in the private setting, however it becomes a challenge to achieve it once we move to public primitives accessible by several parties.

Security-Amplifying Constructions. In the context of cryptography, we typically evaluate the performance of a given tool for a given task by some quantitative measure of security. In this case the second setting translates into a goal called *security amplification*. Given a particular resource (for example, a pseudo-random generator) achieving a given (potentially only moderate or even completely insufficient) security level, we might want to use it to construct a new resource of the same type with significantly better security properties.

In particular, there is a generic way to define the security measure discussed above. One can again consider an *ideal* resource for solving the task considered (this time to serve as the target to aim for) and quantify how much is the available real resource different from this ideal one. This is usually formalized by a distinguisher that interacts with either the real or the ideal system and tries to tell them apart; the performance of the best such distinguisher being the quantification of the real resource's security. For this specific measure we talk about *indistinguishability amplification* and one can then observe at least two its orthogonal forms:

- a *quantitative* form, where we construct a new primitive that performs better than the original primitive in the eyes of the same distinguishers;

- a *qualitative* form, where we obtain a primitive secure against a qualitatively different, stronger class of distinguishers than the original primitive.

Computational Restrictions. Independently of whether we consider reductions or security-amplifying constructions, there is another important defining aspect of any particular setting: restrictions on the computational power considered. On one hand, one can agree on a particular

definition of efficient computation in an attempt to capture practical feasibility and consider agents and objects that only perform such computations. On the other hand, one can decide not to make any such assumptions and consider all possible behaviors, using only arguments that are of information-theoretic nature. The resulting security notions are then broadly called *computational* and *information-theoretic* security, respectively. The relationship between these two types of results depends on the particular setting considered. Typically information-theoretic results are cleaner and require a simpler analysis, serving also as a good indication of the strength of results that one can hope for in the computational world, which is of course more realistic.

1.2 Thesis Overview

Let us now outline the contributions of this thesis and position them within the landscape described above. We follow the viewpoint of cryptography as a constructive discipline when formulating our results, focusing on the existence and properties of constructions suitable for various tasks in the settings detailed below.

First of all, the arguments presented in this thesis are information-theoretic and so are the results obtained. Naturally, it might be interesting to investigate also the translations of these results into the setting with computational restrictions. As we shall see, for some of our results this translation is straightforward, since the same information-theoretic argument can be applied also in the computational case. For other results, the implications for the computational setting remain an open question since a direct translation is clearly not possible.

Second, all our results belong to the domain of symmetric cryptography. We analyze the basic building blocks used in the design of both block ciphers and stream ciphers, as well as various higher-level constructions that take a complete block cipher as their starting point. We also understand the area of symmetric cryptography more broadly to also include keyless cryptographic hash functions (since they also introduce no asymmetry in the distribution of keys) and investigate general methods to construct them from weaker primitives.

In Chapter 2 we start by presenting the language and notation necessary for our later exposition. Most of it is dedicated to introducing the

formalism used for capturing interactions of discrete systems, since this is the stage at which all our further investigation takes place. The actual contributions of the thesis are presented in Chapters 3–5. Each of them starts by an introduction into the setting that it focuses on, a detailed summary of the contributions it contains and also an overview of prior related work. It also references publications where the results first appeared.

Chapter 3 studies constructions for combining several real resources to achieve indistinguishability amplification in the simpler, private setting. Here we take as a starting point the general treatment of information-theoretic indistinguishability amplification given by Maurer, Pietrzak and Renner in [MPR07] where the authors analyze a broad class of constructions called *neutralizing*, comprising many natural examples such as a cascade of permutations or an XOR-combination of functions. They show that any such construction provides two types of indistinguishability amplification corresponding to the notions of qualitative and quantitative amplification sketched above. We extend the random system framework used for their analysis to allow us to capture a generalized notion of distinguishing, so-called *free-start distinguishing*. This facilitates a more careful analysis of the amplification provided by neutralizing constructions, leading to a certain unification of the two forms mentioned.

In Chapter 4 we investigate constructions used for key-length extension of block ciphers. These can be seen both as reductions (since they use an underlying block cipher to construct a new block cipher with a longer key) and as quantitative indistinguishability amplification, since the new block cipher is, naturally, expected to also provide greater security than the original one. The analysis is performed in the ideal cipher model where we assume the underlying block cipher to be ideal and publicly available; and the security of the resulting construction is evaluated based on its indistinguishability — when used with a random key — from an ideal random permutation on a certain number of queries.

We first analyze the very natural and widely used key-length extending construction: the cascade of length ℓ, i.e., sequential application of ℓ block-cipher encryption steps under independently chosen keys. We show that for $\ell \geq 3$ the plain cascade provides a significant security amplification (in agreement with the result of [BR06] for triple encryption) that improves with the length of the cascade also for $\ell > 3$ for all block ciphers having smaller key length than block length. We also present a generic attack against plain cascades of arbitrary length that is a generalization

of the meet-in-the-middle attack against double encryption and of the attack by Lucks [Luc98] against triple encryption. Afterwards, we turn our attention towards a slightly modified cascade of length ℓ: we extend it by adding XOR-randomizing steps between the encryption steps, obtaining a construction referred to as ℓ-XOR-cascade. We investigate the efficiency/security trade-off achieved by such randomization and prove that ℓ-XOR-cascades are preferable to plain cascades for the practical lengths $\ell = 2, 3, 4$; with their security also increasing with higher length ℓ. These results are obtained by relating the security of XOR-cascades to the well-studied security of key-alternating ciphers in the random permutation setting. Finally, we also describe generic attacks against the class of one-query constructions (invoking the underlying block cipher only once per evaluation); against two-query constructions satisfying a certain injectivity property; and against a class of constructions of arbitrary length ℓ that are evaluated in a sequential manner. These attacks prove several of our security bounds tight. An overview of the results achieved can be found in Table 4.1 accompanied with a more detailed description.

Chapter 5 is dedicated to the study of *indifferentiability* introduced by Maurer, Renner and Holenstein [MRH04]. This notion has established itself as the right language for formalizing reductions among publicly-accessible primitives, a prominent example being the intensively-studied problem of constructing a random oracle from a public random compression function [CDMP05]. However, the widely accepted view that a construction enjoying an indifferentiability proof can replace the ideal resource in any application without compromising security has been put in question by the recent work of Ristenpart, Shacham and Shrimpton [RSS11].

We investigate these concerns and observe that certain scenarios require a more careful modelling of available resources (e.g., memory, randomness) than provided by the classical indifferentiability notion. We present a general treatment of such scenarios which leads to quantifying the simulator memory requirements for any domain-extending construction for public random functions. In particular, the following results are obtained:

- Restricting to simulators with no memory (i.e., stateless), even domain extension by a single bit is impossible.

- In the case of infinite domain extension (constructing a random oracle from a public random compression function) the memory avail-

able to the simulator has to be roughly linear in the length of the longest query asked by the distinguisher.

As a consequence of these bounds, it turns out that the random oracle model and the ideal cipher model are no longer equivalent if one restricts to reductions with stateless simulators. Interestingly, they also imply that for any multi-party setting where one cannot assume the existence of a central adversary and hence it requires to be modeled using an independent local simulator for each party, a secure construction of a public random oracle from a public ideal compression function becomes impossible.

Finally, some concluding remarks on our contributions, including the remaining open questions, are presented in Chapter 6. The thesis covers the results presented in publications [GM09, GM10, GT12, DGHM13] as well as some so far unpublished contributions.

Chapter 2

Modelling Interactions of Systems

In this chapter we summarize all the preliminaries necessary for presenting the results described in the thesis. We start by introducing the basic notational conventions in Section 2.1, which is supposed to serve as a reference, while more specific notation is reminded to the reader also later when it is used. The key parts of this chapter (Sections 2.2 and 2.3) are dedicated to the approach we employ for formalizations of various random experiments involving interactions of systems. Finally, we conclude by presenting several idealized cryptographic models used throughout the thesis in Section 2.4.

2.1 Basic Notation

Sets and Tuples. We typically denote sets by calligraphic letters or capital greek letters (e.g. $\mathcal{S}$, Σ). For a finite set $\mathcal{S}$, we denote by $|\mathcal{S}|$ the number of its elements. For a superset clear from the context, we denote the complement of a set $\mathcal{X}$ by $\overline{\mathcal{X}}$. A k-tuple is denoted as $u^k = (u_1, \ldots, u_k)$, and the set of all k-tuples of elements of $\mathcal{U}$ is denoted as $\mathcal{U}^k$. The tuples can be concatenated, which we write as $u^k v^l = (u_1, \ldots, u_k, v_1, \ldots, v_l)$.

We denote the sets of natural and real numbers by the usual symbols $\mathbb{N}$ and $\mathbb{R}$, respectively, with $\mathbb{R}_0^+$ denoting the set of all non-negative reals.

For real numbers $a < b$ we denote by (a, b) and $[a, b]$ the corresponding open and closed interval, respectively.

By $\mathrm{cs}(\cdot)$ we denote the set of all *cyclic shifts* of a given tuple, in other words,

$$\mathrm{cs}(x_1, x_2, ..., x_r) = \{(x_1, x_2, ..., x_r), (x_2, x_3, ..., x_1), ..., (x_r, x_1, ..., x_{r-1})\}.$$

The notation $\mathrm{ms}(i)$ refers to the set of monotone binary sequences of length i where zeroes are preceding ones, i.e., $\mathrm{ms}(i) = \{0^i, 0^{i-1}1, \ldots, 1^i\}$.

Functions. For a mapping f, we denote by $\mathrm{dom}(f)$ and $\mathrm{range}(f)$ its domain and range, respectively. The composition of mappings is interpreted from left to right, i.e., $f \circ g$ denotes the mapping $g(f(\cdot))$. Additionally, we let $\mathrm{Func}(m, \ell)$ be the set of all functions from $\{0,1\}^m$ to $\{0,1\}^\ell$ and $\mathrm{Perm}(n)$ be the set of all permutations of $\{0,1\}^n$. In particular, $id \in \mathrm{Perm}(n)$ represents the identity mapping when n is understood from the context. We say that a function $\varepsilon : \mathbb{N} \to \mathbb{R}_0^+$ is negligible if it vanishes faster than the inverse of any polynomial, i.e., if

$$(\forall c \in \mathbb{N})(\exists n_0 \in \mathbb{N})(\forall n > n_0) : \ \varepsilon(n) < \frac{1}{n^c}.$$

Throughout the thesis all logarithms considered are to the base 2 unless otherwise indicated. The notation $\lceil \cdot \rceil$ corresponds to the usual ceiling function, while $x^{\underline{n}}$ represents the falling factorial power, i.e., $x^{\underline{n}} = x(x-1)\cdots(x-n+1)$.

Bitstring Operations. We shall denote by $\oplus$ the usual operation of exclusive-or (XOR) on single bits and also extend it naturally on bitstrings of equal length. For the binary operation of bitstring concatenation we either omit any symbol (such as in $0^k 1^\ell$) or emphasize the concatenation by the symbol $\|$ (such as in $x\|r$).

Algorithms. We describe algorithms by using standard pseudocode notation that should not cause any confusion. Assignments are denoted $:=$ and program blocks are determined by indentation. We write $r_1, \ldots, r_m \xleftarrow{\$} \mathcal{S}$ to denote that the values $r_1, ..., r_m$ are being chosen independently and uniformly at random from the set $\mathcal{S}$. while $r_1, \ldots, r_m \xleftarrow{\$,\mathrm{d}} \mathcal{S}$ indicates uniformly random selection of *distinct* elements from $\mathcal{S}$. We typically describe algorithms representing distinguishers (see Section 2.3.2), hence we use the notation "**query** $y := f(x)$" to capture that the distinguisher queries a function f available to it (e.g. a particular interface, see

later) with a value x and stores the resulting value as y. If we describe a choice of some object $\mathcal{I}$ with particular properties, we always assume that in case more such objects exist, one is chosen in a clearly defined deterministic way (e.g., the first one with respect to some natural ordering).

Random Variables and Probability Distributions. We usually denote random variables and concrete values they can take on by capital and small letters, respectively. The complement of an event A is denoted by $\overline{A}$. Naturally, for any binary random variable B, we denote the event that it takes on the value 1 also by B and the event that it takes on the value 0 by $\overline{B}$. The symbol $p_{coll}(n,k)$ denotes the probability that k independent random variables with uniform distribution over a set of size n contain a collision, i.e., that they are not all distinct. It is well-known that $p_{coll}(n,k) < k^2/2n$.

For events A and B and random variables U and V with ranges $\mathcal{U}$ and $\mathcal{V}$, respectively, we denote by $\mathsf{P}_{UA|VB}$ the corresponding conditional probability distribution, seen as a function $\mathcal{U} \times \mathcal{V} \to [0,1]$. Here the value $\mathsf{P}_{UA|VB}(u,v)$ is well-defined for all $u \in \mathcal{U}$ and $v \in \mathcal{V}$ such that $\mathsf{P}_{VB}(v) > 0$ and undefined otherwise. Two probability distributions P_U and $\mathsf{P}_{U'}$ on the same set $\mathcal{U}$ are equal, denoted $\mathsf{P}_U = \mathsf{P}_{U'}$, if $\mathsf{P}_U(u) = \mathsf{P}_{U'}(u)$ for all $u \in \mathcal{U}$. Conditional probability distributions are equal if the equality holds for all arguments for which both of them are defined. To emphasize the random experiment $\mathcal{E}$ in consideration, we usually write it in the superscript, e.g. $\mathsf{P}^{\mathcal{E}}_{U|V}(u,v)$. By a lower-case p we denote (conditional) probability distributions that by themselves do not define a random experiment.

The statistical distance of two random variables X and X' over $\mathcal{X}$ is defined as $\delta(X,X') = \frac{1}{2}\sum_{x\in\mathcal{X}} |\mathsf{P}[X=x] - \mathsf{P}[X'=x]|$. The expected value of a discrete real-valued random variable X is denoted by

$$\mathsf{E}[X] = \sum_{x\in\mathcal{X}} (x \cdot \mathsf{P}[X=x])$$

and its variance is denoted by $\mathsf{Var}(X) = \mathsf{E}[(X - \mathsf{E}[X])^2]$. For two discrete real-valued random variables X and Y within the same random experiment, we denote their covariance by

$$\mathsf{Cov}(X,Y) = \mathsf{E}[(X - \mathsf{E}[X])(Y - \mathsf{E}[Y])].$$

Entropy Measures. For a discrete random variable X with range $\mathcal{X}$ we denote by $H(X)$ the Shannon entropy of X, i.e., $H(X) =$

$\sum_{x\in\mathcal{X}} -\mathsf{P}_X(x)\log\mathsf{P}_X(x)$ where $\mathsf{P}_X(x)$ denotes the probability that X takes on the value $x \in \mathcal{X}$. Moreover, we denote by $H(Y|X)$ the usual notion of conditional entropy of Y given X, satisfying the chain rule $H(Y|X) = H(XY) - H(X)$. For a probability $p \in [0,1]$ we also use the notion of binary entropy denoted $h(p)$ that is defined as the Shannon entropy of the binary random variable taking on the two possible values with probabilities p and $1-p$.

2.2 Abstract Systems

The Top-Down Approach. The landscape in which we formulate any statements about system interactions follows the top-down modelling approach presented in [MR11]. This advocates the premise that every possible statement should be formulated (and proven) at the highest possible level of abstraction at which this can be done, arguing that this leads to a more conceptual view and simpler proofs. Such an approach also allows for unification of results, where several seemingly independent observations turn out to be instantiations of a more general pattern that can be stated on a more abstract level.

Following the top-down approach, in this section we start by introducing a language from [Mau11] for modelling system interactions at a very abstract level, considering arbitrary systems that can be attached via their interfaces and interact with each other. Later in Section 2.3 we refine this view by giving a precise formalism for capturing systems at the level of their input-output behavior, following the work [Mau02]. This will exactly match the level of abstraction necessary and sufficient for presenting all the results in this thesis. Finally, one could imagine several lower levels of modelling of system interactions, for example taking aspects such as time into account. Since we do not need such a refinement for our exposition, we do not introduce any such model.

Abstract Systems. At the highest level of abstraction, we see a system as an object with interfaces via which it interacts with its environment (consisting of other systems). Two systems that are connected via an interface of each of them together form a new object which is again a system. Implicitly we assume each two different systems to be mutually independent.

We consider three distinct types of systems: resources, converters and distinguishers; now we describe the role of each of these classes:

Resources[2] are denoted by upper-case boldface letters (such as $\mathbf{S}$, $\mathbf{T}$) and their interfaces are labeled by elements of some index set $\mathcal{I}$. We often consider the cases of one- or two-interface resources (i.e., $|\mathcal{I}| \in \{1, 2\}$) but sometimes also model resources having a higher (finite) number of interfaces.

Converters are systems having one *inner* and one *outer* interface and are denoted by small Greek letters such as ϕ, π, σ or by sans-serif identifiers such as C or Casc. The set of all converters considered is denoted as Σ.

Distinguishers are usually denoted by $\mathbf{D}$, $\mathbf{D}^*$ or similar. A distinguisher is a system that connects to all interfaces of a resource $\mathbf{S}$ and outputs (at a separate interface) a single bit denoted W. As the name suggests, the role of a distinguisher is to differentiate between being connected to two different resources and the output bit represents its guess of which of the two systems it is connected to. Very roughly, if none of the distinguishers considered in a particular scenario can differentiate between the two systems, they can be considered similar and used interchangeably. We give a precise treatment of distinguishers and distinguishing in Section 2.3.2.

Composing Abstract Systems. We can compose a resource with a converter by attaching the converter to one of the resource's interfaces. This is denoted by $\phi^I\mathbf{S}$ where the label $I \in \mathcal{I}$ in the superscript specifies the resource's interface used: here the inner interface of ϕ is attached to the I-interface of the resource $\mathbf{S}$. The outcome of such a composition is again a resource with the same set of interfaces as $\mathbf{S}$, it just exposes the outer interface of ϕ as its I-interface; other interfaces of $\mathbf{S}$ remain unaffected. If $\hat{\phi}$ is an n-tuple of converters and $\mathbf{S}$ is an n-interface resource with a clearly understood ordering of interfaces, we sometimes write $\hat{\phi}\mathbf{S}$ to denote the application of the i-th converter in $\hat{\phi}$ to the i-th interface of $\mathbf{S}$ for all $i \in \{1, \ldots, n\}$.

Two resources $\mathbf{S}$ and $\mathbf{T}$ with the same sets of interfaces $\mathcal{I}$ can be composed in parallel, which is denoted as $\mathbf{S}\|\mathbf{T}$ and the resulting system is again a resource. It has the same set of interfaces $\mathcal{I}$; each of the interfaces $I \in \mathcal{I}$ of $\mathbf{S}\|\mathbf{T}$ allows to access the interface I of both subsystems $\mathbf{S}$ and $\mathbf{T}$.

[2] We sometimes also use the term "resources" in a more informal way to refer to computational power, memory, etc. The intended meaning should always be clear from the context.

Two converters can also be composed, both sequentially and in parallel. The sequential composition of converters ψ and ϕ is denoted by $\psi \circ \phi$, and is defined by $(\psi \circ \phi)^I\mathbf{S} = \psi^I(\phi^I\mathbf{S})$ for all resources $\mathbf{S}$. The parallel composition is denoted as $\psi|\phi$ and defined by $(\psi|\phi)^I(\mathbf{S}\|\mathbf{T}) = (\psi^I\mathbf{S})\|(\phi^I\mathbf{T})$ for all $\mathbf{S}$ and $\mathbf{T}$. In the context of converters, the term id is used to refer to the "identity converter" that forwards all inputs and outputs.

Two-Interface Resources. Since we very often consider resources with at most 2 interfaces, we simplify the algebraic notation in these special cases. For a 2-interface system $\mathbf{S}$, we understand the left and right side of the symbol $\mathbf{S}$ as representing the two interfaces of the system $\mathbf{S}$. Hence, for example, attaching a converter π to the left interface of a resource $\mathbf{S}$ results in a resource $\pi\mathbf{S}$ while attaching a converter σ to the right interface of a resource $\mathbf{T}$ results in a resource $\mathbf{T}\sigma$. For a 2-interface system $\mathbf{S}$ we sometimes denote by $[\mathbf{S}]_L$ (resp. $[\mathbf{S}]_R$) its left (resp. right) interface. In the context of indifferentiability (see Section 5.1) we also denote the left and right interfaces as private and public, respectively.

Note that all these conventions also carry over to the simplest case of a single-interface resource where we denote a converter attached to its interface by writing it in front of the resource (e.g. $\mathsf{C}\mathbf{S}$) and if the context is clear, we sometimes abuse the notation by referring to the response of $\mathbf{S}$ to a query x by $\mathbf{S}(x)$.

Finally, we introduce one additional notational shorthand that will turn out to be very practical. For 2-interface resources $\mathbf{S}$ and $\mathbf{T}$ we denote by $\mathbf{S}\lfloor\mathbf{T}$ composing these resources in parallel and applying converters such that the new resource exposes the left interface of $\mathbf{S}$ as its left interface and the right interface of $\mathbf{T}$ as its right interface, hence making the right interface of $\mathbf{S}$ and the left interface of $\mathbf{T}$ inaccessible (note that neither are these interfaces connected to each other).

2.3 Random Systems

In this section, we present in detail the random systems framework introduced in [Mau02], following some of the notational changes in [MPR07]. We employ this framework to model all the systems described in Section 2.2 at the level of abstraction where they are characterized by their input-output behavior, which will be necessary (and sufficient) for formulating all our results.

2.3.1 Input-Output Behavior of Discrete Systems

The starting point of the random-system framework is the basic observation that the input-output behavior of any (reactive) discrete system $\mathbf{S}$ with inputs in $\mathcal{X}$ and outputs in $\mathcal{Y}$ can be described by an infinite family of functions describing, for each $i \geq 1$, the probability distribution of the i-th output $Y_i \in \mathcal{Y}$ given the values of the first i inputs $X^i \in \mathcal{X}^i$ and the previous $i-1$ outputs $Y^{i-1} \in \mathcal{Y}^{i-1}$. Formally, hence, an $(\mathcal{X}, \mathcal{Y})$-*(random) system* $\mathbf{S}$ is an infinite sequence of functions $\mathsf{p}^{\mathbf{S}}_{Y_i|X^iY^{i-1}} : \mathcal{Y} \times \mathcal{X}^i \times \mathcal{Y}^{i-1} \to [0,1]$ such that $\sum_{y_i} \mathsf{p}^{\mathbf{S}}_{Y_i|X^iY^{i-1}}(y_i, x^i, y^{i-1}) = 1$ for all $i \geq 1$, $x^i \in \mathcal{X}^i$ and $y^{i-1} \in \mathcal{Y}^{i-1}$. We stress that the notation $\mathsf{p}^{\mathbf{S}}_{Y_i|X^iY^{i-1}}$, by itself, involves some abuse, as we are not considering any particular random experiment with well-defined random variables Y_i, X^i, Y^{i-1} until the system will be interacting with a distinguisher (see below), in which case the random variables will exist and take the role of the transcript. In general, we shall also typically define discrete systems by a high level description, as long as the resulting conditional probability distributions can be derived uniquely from this description. Note that similarly to the notion of a random variable, the word "random" does not imply any uniformity of distribution here.

Equivalence of Systems. We shall use boldface letters (e.g. $\mathbf{S}$) to denote both a discrete system and a random system corresponding to it. This should cause no confusion. We emphasize that all the results of this thesis stated for random systems actually hold for arbitrary systems, since the only property of a system that is relevant for them is its input-output behavior. Hence, it is reasonable to consider two discrete systems equivalent if their input-output behaviors are the same, even if their internal structure differs. Formally, we say that two systems $\mathbf{S}$ and $\mathbf{T}$ are *equivalent*, denoted $\mathbf{S} \equiv \mathbf{T}$, if they correspond to the same random system, i.e., if $\mathsf{p}^{\mathbf{S}}_{Y_i|X^iY^{i-1}} = \mathsf{p}^{\mathbf{T}}_{Y_i|X^iY^{i-1}}$ for all $i \geq 1$.

An Alternative Description. A random system can also be defined by a sequence of conditional probability distributions $\mathsf{p}^{\mathbf{S}}_{Y^i|X^i}$ for $i \geq 1$. This description is often convenient, but is not minimal: the distributions $\mathsf{p}^{\mathbf{S}}_{Y^i|X^i}$ must satisfy a consistency condition for different i. The conversion between these two forms can be described by

$$\mathsf{p}^{\mathbf{S}}_{Y^i|X^i} = \prod_{j=1}^{i} \mathsf{p}^{\mathbf{S}}_{Y_j|X^jY^{j-1}} \qquad \text{and} \qquad \mathsf{p}^{\mathbf{S}}_{Y_i|X^iY^{i-1}} = \frac{\mathsf{p}^{\mathbf{S}}_{Y^i|X^i}}{\mathsf{p}^{\mathbf{S}}_{Y^{i-1}|X^{i-1}}}. \tag{2.1}$$

Resources as Random Systems. A single-interface reactive resource can be directly modelled as a random system which precisely captures its input-output behavior. This is also true for resources with more interfaces, although some additional technical work is necessary. One can simply extend both the input alphabet $\mathcal{X}$ and the output alphabet $\mathcal{Y}$ by the cross product with the resource's index set $\mathcal{I}$, hence making every input and output contain also the description of the interface used. As already mentioned, we usually do not give the respective conditional probability distributions as long as an unambiguous high-level description of a system is possible, hence we are also absolved from the need to address this step explicitly in most of our considerations.

Special Classes and Examples. We define several special properties and classes of random systems that will turn out to be useful in our later discussion. We say that an $(\mathcal{X}, \mathcal{Y})$-random system $\mathbf{S}$ is:

- *deterministic* if it always acts in a deterministic way, i.e., if the range of the mapping $\mathsf{p}^{\mathbf{S}}_{Y_i|X^iY^{i-1}}$ is $\{0,1\}$ for all $i \geq 1$.

- *stateless* if the probability distribution of each output depends only on the current input, i.e., if there exists a conditional probability distribution $\mathsf{p}_{Y|X} : \mathcal{Y} \times \mathcal{X} \to [0,1]$ such that the system $\mathbf{S}$ satisfies $\mathsf{p}^{\mathbf{S}}_{Y_i|X^iY^{i-1}}(y_i, x^i, y^{i-1}) = \mathsf{p}_{Y|X}(y_i, x_i)$ for all $i \geq 1, x^i$ and y^i.

- a *random function* if it answers consistently, i.e., if it satisfies the condition $X_i = X_j \Rightarrow Y_i = Y_j$ for all $i, j \geq 1$ in every interaction. Note that for random functions the properties of being stateless and of being deterministic are equivalent; in general a random function does not have to be stateless.

- a *random permutation* on a set $\mathcal{X}$ if it is a random function with $\mathcal{Y} = \mathcal{X}$ mapping distinct inputs to distinct outputs, i.e., always satisfying the condition $X_i \neq X_j \Rightarrow Y_i \neq Y_j$ for all $i, j \geq 1$.

- a *cc-stateless random function* [MT09] if it corresponds to a random variable taking on as values functions tables $\mathcal{X} \to \mathcal{Y}$ (the name stands for convex-combination stateless). Again, in general a random function is not necessarily cc-stateless.

Some important examples of cc-stateless (but not stateless) random functions are the following:

- The *uniform random function (URF)* $\mathbf{R} : \{0,1\}^m \to \{0,1\}^r$ realizes a uniformly chosen function $f \in \mathrm{Func}(m,r)$, i.e., it answers every new query with an element uniformly chosen from its (finite) range. Whenever we want to emphasize the size of the domain and range of the URF, we denote it as $\mathbf{R}^{m,r}$. The system $\mathbf{R}^{m,r}$ is sometimes also called a fixed-input length random oracle.

- In contrast, we also consider a URF with the domain being the set of all bitstrings and with output length n. This is called an (arbitrary input-length) *random oracle* and denoted by $\mathbf{R}^{*,n}$.

- The *uniform random permutation (URP) on* $\{0,1\}^n$, denoted $\mathbf{P} : \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$, realizes a uniformly chosen permutation $P \in \mathrm{Perm}(n)$ allowing both forward queries of the form $(x,+)$ returning $P(x)$ as well as backward queries $(y,-)$ returning $P^{-1}(y)$. Note that (as also reminded later) in Chapter 3 we only consider URPs allowing forward queries, while still using the notation $\mathbf{P}$ for the sake of simplicity.

- The *ideal block cipher* $\mathbf{E} : \{0,1\}^\kappa \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ is a random function which realizes an independent uniform random permutation $\mathbf{E}_k \in \mathrm{Perm}(n)$ for each key $k \in \{0,1\}^\kappa$; in particular, the system allows both forward and backward queries to each $\mathbf{E}_k$. For a more detailed discussion of the ideal cipher model see Section 2.4.

Note that we often write $\mathbf{R}$, $\mathbf{P}$ or $\mathbf{E}_k$ to refer to the randomly chosen function implemented by the respective system. This notation makes sense for any system being a cc-stateless random function, since its responses do not depend on the order of the queries asked.

Finally, in contexts where this is natural (such as in the indifferentiability setting) we see all the above-mentioned cc-stateless random functions as 2-interface resources, providing access to the same function at each of them. Although such a difference in the viewpoint is purely formal, we will always explicitly mention which view we are using in respective parts of the thesis so that we avoid any confusion.

2.3.2 Distinguishers and Indistinguishability

A *distinguisher* $\mathbf{D}$ for an $(\mathcal{X},\mathcal{Y})$-random system asking q queries is a $(\mathcal{Y},\mathcal{X})$-random system which is "one query ahead:" its input-output behavior is defined by the conditional probability distributions of its

queries $\mathsf{p}^{\mathbf{D}}_{X_i|X^{i-1}Y^{i-1}}$ for all $1 \le i \le q$. (The first query of $\mathbf{D}$ is determined by $\mathsf{p}^{\mathbf{D}}_{X_1}$.) After a certain number of queries (say q), the distinguisher outputs a bit W_q depending on the transcript X^qY^q. For a random system $\mathbf{S}$ and a distinguisher $\mathbf{D}$, let $\mathbf{DS}$ be the random experiment where $\mathbf{D}$ interacts with $\mathbf{S}$. The distribution of X^qY^q in this experiment can be expressed by

$$\begin{aligned}\mathsf{P}^{\mathbf{DS}}_{X^qY^q}(x^q, y^q) &= \prod_{i=1}^{q} \mathsf{p}^{\mathbf{D}}_{X_i|X^{i-1}Y^{i-1}}(x_i, x^{i-1}, y^{i-1})\mathsf{p}^{\mathbf{S}}_{Y_i|X^iY^{i-1}}(y_i, x^i, y^{i-1}) \\ &= \mathsf{p}^{\mathbf{D}}_{X^q|Y^{q-1}}(x^q, y^{q-1}) \cdot \mathsf{p}^{\mathbf{S}}_{Y^q|X^q}(y^q, x^q) \qquad (2.2)\end{aligned}$$

where the last equality follows from (2.1).

Restricted Distinguishers. We consider two special classes of distinguishers using the following notation:

- We denote the class of all (computationally unbounded) *non-adaptive* distinguishers by NA. These distinguishers select all queries $X_1, \ldots, X_q$ in advance, i.e., independently of the outputs $Y_1, \ldots, Y_q$.
- By RI we denote the class of all (computationally unbounded) *random-input* distinguishers. These cannot select queries but are given uniformly random values $X_1, \ldots, X_q$ and the corresponding outputs $Y_1, \ldots, Y_q$.

Distinguishing Advantage and Statistical Distance. For two $(\mathcal{X}, \mathcal{Y})$-systems $\mathbf{S}$ and $\mathbf{T}$, the *distinguishing advantage* of $\mathbf{D}$ in distinguishing systems $\mathbf{S}$ and $\mathbf{T}$ by q queries is defined as

$$\Delta^{\mathbf{D}}_q(\mathbf{S}, \mathbf{T}) = \left|\mathsf{P}^{\mathbf{DS}}(W_q = 1) - \mathsf{P}^{\mathbf{DT}}(W_q = 1)\right|.$$

We shall denote by $\Delta^{\mathcal{D}}_q(\mathbf{S}, \mathbf{T})$ and $\Delta_q(\mathbf{S}, \mathbf{T})$ the maximal advantage over a certain class $\mathcal{D}$ of distinguishers and over all distinguishers issuing at most q queries, respectively. If the number of queries a particular distinguisher $\mathbf{D}$ makes is clear from the context, we sometimes simply write $\Delta^{\mathbf{D}}$ to denote its advantage.

On the other hand, we define

$$\delta^{\mathbf{D}}_q(\mathbf{S}, \mathbf{T}) := \|\mathsf{P}^{\mathbf{DS}}_{X^qY^q} - \mathsf{P}^{\mathbf{DT}}_{X^qY^q}\| = \frac{1}{2}\sum_{x^qy^q} \left|\mathsf{P}^{\mathbf{DS}}_{X^qY^q}(x^q, y^q) - \mathsf{P}^{\mathbf{DT}}_{X^qY^q}(x^q, y^q)\right|$$

to be the *statistical distance of transcripts* when $\mathbf{D}$ interacts with $\mathbf{S}$ and $\mathbf{T}$, respectively. Again, $\delta_q^{\mathcal{D}}(\mathbf{S},\mathbf{T})$ and $\delta_q(\mathbf{S},\mathbf{T})$ denote the maximal value over a class $\mathcal{D}$ of distinguishers and over all distinguishers, respectively.

The statistical distance of transcripts is closely related to the distinguishing advantage: in general we have $\Delta_q^{\mathbf{D}}(\mathbf{S},\mathbf{T}) \le \delta_q^{\mathbf{D}}(\mathbf{S},\mathbf{T})$, but for a computationally unbounded distinguisher $\mathbf{D}$ that chooses the output bit optimally, we have $\Delta_q^{\mathbf{D}}(\mathbf{S},\mathbf{T}) = \delta_q^{\mathbf{D}}(\mathbf{S},\mathbf{T})$. In particular, we have $\Delta_q(\mathbf{S},\mathbf{T}) = \delta_q(\mathbf{S},\mathbf{T})$, $\Delta_q^{\mathsf{NA}}(\mathbf{S},\mathbf{T}) = \delta_q^{\mathsf{NA}}(\mathbf{S},\mathbf{T})$ and $\Delta_q^{\mathsf{RI}}(\mathbf{S},\mathbf{T}) = \delta_q^{\mathsf{RI}}(\mathbf{S},\mathbf{T})$. Finally, using (2.2) to expand the definition of $\delta_q^{\mathbf{D}}(\mathbf{S},\mathbf{T})$, we obtain

$$\begin{aligned}\delta_q^{\mathbf{D}}(\mathbf{S},\mathbf{T}) &= \frac{1}{2}\sum_{x^q y^q} \mathsf{p}^{\mathbf{D}}_{X^q|Y^{q-1}}(x^q,y^{q-1}) \cdot \left|\mathsf{p}^{\mathbf{S}}_{Y^q|X^q}(y^q,x^q) - \mathsf{p}^{\mathbf{T}}_{Y^q|X^q}(y^q,x^q)\right| \\ &= \sum_{x^q y^q} \mathsf{p}^{\mathbf{D}}_{X^q|Y^{q-1}}(x^q,y^{q-1}) \cdot \left(\mathsf{p}^{\mathbf{S}}_{Y^q|X^q}(y^q,x^q) - \mathsf{p}^{\mathbf{T}}_{Y^q|X^q}(y^q,x^q)\right)\end{aligned} \tag{2.3}$$

where the last summation goes only over all $x^q y^q$ such that $\mathsf{p}^{\mathbf{S}}_{Y^q|X^q}(y^q,x^q) > \mathsf{p}^{\mathbf{T}}_{Y^q|X^q}(y^q,x^q)$ holds. This representation of $\delta_q^{\mathbf{D}}(\mathbf{S},\mathbf{T})$ turns out to be useful later.

Mixed Systems. For two $(\mathcal{X},\mathcal{Y})$-systems $\mathbf{S}$ and $\mathbf{T}$ and a uniform random bit B, $\langle \mathbf{S}/\mathbf{T}\rangle_B$ denotes the random system which is equal to $\mathbf{S}$ if $B = 0$ and equal to $\mathbf{T}$ otherwise. If mentioning the random variable B explicitly is not necessary, we only write $\langle \mathbf{S}/\mathbf{T}\rangle$. The following simple lemma comes from [MPR07].

Lemma 2.1 *For every distinguisher* $\mathbf{D}$ *we have:*

(i) $\Delta_q^{\mathbf{D}}(\mathbf{S},\mathbf{T}) = 2\left|\mathsf{P}^{\mathbf{D}\langle \mathbf{S}/\mathbf{T}\rangle_B}(W_q = B) - \frac{1}{2}\right|$,

(ii) $\Delta_q^{\mathbf{D}}(\mathbf{S},\langle \mathbf{S}/\mathbf{T}\rangle_B) = \frac{1}{2}\Delta_q^{\mathbf{D}}(\mathbf{S},\mathbf{T})$.

2.3.3 Constructions

Now we consider the modelling of converters at the abstraction level of their input-output behavior. Since being sufficient for our work, we will only consider converters that are always invoked by a query at the outer interface (also called *outer query*), they then issue zero or more queries to the resource attached to their inner interface (called *inner queries*) always waiting for a response; and finally produce an output at the outer interface. The underlying resource is also sometimes called a *subsystem*.

In accordance with previous literature employing the random systems framework, in various contexts we refer to these converters as *constructions*, *protocols* or *simulators*, depending on the role that they play and the naming convention in the particular setting. Note however that syntactically these are all objects of the same type and we shall apply the notation given for converters in Section 2.2 to all of them.

The notions of being deterministic and of being stateless naturally extend to constructions. We dispense with a formal definition for now, however, we point out that we allow a stateless construction to keep a state during its invocations of the underlying resource. For a precise definition of a stateless converter see Section 5.2.1 where this notion will become crucial for our investigation.

We make use of the following simple but helpful claims.

Lemma 2.2 *Let* C *and* C' *be two constructions and let* $\mathbf{F}$ *and* $\mathbf{G}$ *be (single interface) random systems.*

(i) [Mau02, Lemma 5] $\Delta_q(\mathsf{C}\mathbf{F}, \mathsf{C}\mathbf{G}) \leq \Delta_{q'}(\mathbf{F}, \mathbf{G})$*, where* q' *is the maximum number of invocations of any internal system* $\mathbf{H}$ *for any sequence of* q *queries to* $\mathsf{C}\mathbf{H}$*, if such a value is defined.*

(ii) [GM09, Lemma 3] There exists a fixed permutation $Q \in \mathrm{Perm}(n)$ *represented by a deterministic stateless system* $\mathbf{Q}$ *such that* $\Delta_q(\mathsf{C}\mathbf{P}, \mathsf{C}'\mathbf{P}) \leq \Delta_q(\mathsf{C}\mathbf{Q}, \mathsf{C}'\mathbf{Q})$.

Proof: [sketch] The first claim states the intuitive fact that interacting with the distinguished systems through an additional enveloping construction C cannot improve the distinguishing advantage. This is because if we consider the class of all information-theoretic distinguishers, the construction C can always be simulated by the distinguisher. The second claim is just an averaging argument over all the possible values taken by $\mathbf{P}$. ∎

For the sake of notational simplicity, we shall sometimes denote by $\mathsf{C}(\cdot, \cdot)$ constructions that access *two* underlying single-interface resources. This could naturally be formalized using their parallel composition. For such constructions, we let q_1 and q_2 denote the maximal possible number of queries made to the first and second subsystem, respectively, during the first q queries issued to the construction (if defined).

Moreover, if this leads to no confusion we sometimes use the symbol C also to refer to the mapping provided by the outer interface of a construction C, e.g. by writing $\mathsf{C}\colon \mathcal{X} \to \mathcal{Y}$ to denote that the outer interface of C takes inputs from the set $\mathcal{X}$ and outputs elements of the set $\mathcal{Y}$.

2.3.4 Games and Game-Winning

Games as Systems. Among random systems, we shall be in particular interested in $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-systems having a bit as a part of their output that satisfies a certain monotonicity property. Namely, if we denote the i-th output of such a system by $(Y_i, A_i) \in \mathcal{Y} \times \{0,1\}$ then we say that the sequence $A_1, A_2, \ldots$ is a *monotone binary output (MBO)* if $A_i = 1$ implies $A_j = 1$ for all $j > i$. For convenience, we define $A_0 = 0$.

The reason for studying this particular type of systems is that any one-player game can be seen as a $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-system $\mathbf{S}$ with a monotone binary output. Here the player makes moves $X_1, X_2, \ldots$ and receives game outputs $Y_1, Y_2, \ldots$ Additionally, the system after each move also outputs a monotone bit indicating whether the game has already been won. The goal of the player[3] is to provoke the change of this bit, which is initially 0. Based on this intuition, we shall refer to systems having an MBO as a part of their output as *games*. Moreover, for a game represented by a $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-system $\mathbf{S}$ we define $\mathbf{S}^-$ to be the $(\mathcal{X}, \mathcal{Y})$-system obtained from $\mathbf{S}$ by ignoring the MBO. Note that for the task of winning a game it is irrelevant whether the player can see MBO, so we can think of it interacting only with the system $\mathbf{S}^-$.

The cryptographic motivation for studying this type of systems stems from the fact that the MBO of a game can be defined to model a certain condition on the behavior of the underlying $(\mathcal{X}, \mathcal{Y})$-system. Typically, the condition captures whether or not the behavior of the system has so far violated some additional requirement (e.g. distinct outputs, consistent outputs) during the interaction. Winning the game corresponds to violating this requirement which typically implies a certain breakdown of the security of the system.

For a $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-system $\mathbf{S}$ with an MBO called A_i and for a player $\mathbf{D}$, we denote by $\nu_q^{\mathbf{D}}(\mathbf{S})$ the probability that $\mathbf{D}$ wins the game $\mathbf{S}$ within q

[3] Note that a player is formally the same type of object as a distinguisher, hence we shall use both terms, depending on the context.

queries, i.e., $\nu_q^{\mathbf{D}}(\mathbf{S}) = \mathsf{P}_{A_q}^{\mathbf{DS}}(1)$. We will usually be interested in the maximal winning probability over a class $\mathcal{D}$ of players and over all players which we denote by $\nu_q^{\mathcal{D}}(\mathbf{S})$ and $\nu_q(\mathbf{S})$, respectively.

Conditional Equivalence. Having a $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-system $\mathbf{S}$ with the MBO $\mathcal{A} = A_0, A_1, \ldots$ and a $(\mathcal{X}, \mathcal{Y})$-system $\mathbf{T}$, we say that $\mathbf{S}$ *conditioned on* $\mathcal{A}$ *is equivalent to* $\mathbf{T}$, denoted $\mathbf{S}|\mathcal{A} \equiv \mathbf{T}$, if

$$\mathsf{p}^{\mathbf{S}}_{Y_i|X^iY^{i-1}A_i=0} = \mathsf{p}^{\mathbf{T}}_{Y_i|X^iY^{i-1}}$$

for all $i \geq 1$ and for all arguments for which $\mathsf{p}^{\mathbf{S}}_{Y_i|X^iY^{i-1}A_i}$ is defined. Intuitively, this captures the fact that as long as the game $\mathbf{S}$ was not won, the underlying system $\mathbf{S}^-$ behaves the same way as $\mathbf{T}$. This is formalized by the following claim proved in [Mau02] in a slightly different notation.

Lemma 2.3 *If* $\mathbf{S}|\mathcal{A} \equiv \mathbf{T}$ *then* $\Delta_q(\mathbf{S}^-, \mathbf{T}) \leq \nu_q(\mathbf{S})$.

This claim intuitively states that if two systems are equivalent as long as some condition does not occur, then the advantage in distinguishing these systems can be upper-bounded by the best achievable probability of enforcing this condition (i.e., winning the game). An analogous type of statement for the context of the game-playing technique, known as the Fundamental Lemma of Game-Playing, was given in [BR06].

We omit the proof of Lemma 2.3 since we give the proof of a more general Lemma 2.4 below.

Blocked Systems. Let $\mathbf{S}$ be a $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-system with an MBO $\mathcal{A}$. Following [MPR07], we define $\mathbf{S}^{\perp}$ to be the $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-system which masks the $\mathcal{Y}$-output to a dummy symbol ($\perp \notin \mathcal{Y}$) as soon as the MBO turns to the value 1, i.e., as soon as the game $\mathbf{S}$ is won. More precisely, the following function is applied to the outputs of $\mathbf{S}$:

$$(y, a) \mapsto (y', a) \quad \text{where} \quad y' = \begin{cases} y & \text{if } a = 0 \\ \perp & \text{if } a = 1. \end{cases}$$

The following lemma given in [GM09] relates the optimal advantage in distinguishing two random systems to the optimal advantage in distinguishing their blocked counterparts.

Lemma 2.4 *Let* $\mathbf{S}$ *and* $\mathbf{T}$ *be two* $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$*-systems with MBOs* $\mathcal{A}$ *and* $\mathcal{B}$*, respectively. Let* $\mathbf{S}^{\perp}$ *denote the random system* $\mathbf{S}$ *blocked by* $\mathcal{A}$ *and let* $\mathbf{T}^{\perp}$ *denote* $\mathbf{T}$ *blocked by* $\mathcal{B}$*. Then for every distinguisher* $\mathbf{D}$ *we have*

$$\Delta_q^{\mathbf{D}}(\mathbf{S}^-, \mathbf{T}^-) \leq \Delta_q\left((\mathbf{S}^{\perp})^-, (\mathbf{T}^{\perp})^-\right) + \nu_q^{\mathbf{D}}(\mathbf{S}).$$

Proof: Let $\mathbf{D}$ be an arbitrary distinguisher for $(\mathcal{X}, \mathcal{Y})$-systems. Let $\mathbf{D}'$ be a distinguisher that works as follows: it simulates $\mathbf{D}$, but whenever it receives an answer $\perp$ to any of its queries, at the end of the experiment it outputs 1 instead of the output of $\mathbf{D}$. Then for any $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-system with MBO (i.e., a game) $\mathbf{G}$ we have clearly have

$$\mathsf{P}^{\mathbf{D}\mathbf{G}^-}[W_q = 1] \leq \mathsf{P}^{\mathbf{D}'(\mathbf{G}^\perp)^-}[W_q = 1] \leq \mathsf{P}^{\mathbf{D}\mathbf{G}^-}[W_q = 1] + \nu_q^{\mathbf{D}}(\mathbf{G}).$$

First, let us assume that $\mathsf{P}^{\mathbf{D}\mathbf{T}^-}[W_q = 1] \geq \mathsf{P}^{\mathbf{D}\mathbf{S}^-}[W_q = 1]$. Then, using the definition of the distinguishing advantage and the above inequalities, we get

$$\begin{aligned}
\Delta_q^{\mathbf{D}}(\mathbf{S}^-, \mathbf{T}^-) &= \left|\mathsf{P}^{\mathbf{D}\mathbf{T}^-}[W_q = 1] - \mathsf{P}^{\mathbf{D}\mathbf{S}^-}[W_q = 1]\right| \\
&= \mathsf{P}^{\mathbf{D}\mathbf{T}^-}[W_q = 1] - \mathsf{P}^{\mathbf{D}\mathbf{S}^-}[W_q = 1] \\
&\leq \mathsf{P}^{\mathbf{D}'(\mathbf{T}^\perp)^-}[W_q = 1] - (\mathsf{P}^{\mathbf{D}'(\mathbf{S}^\perp)^-}[W_q = 1] - \nu_q^{\mathbf{D}}(\mathbf{S})) \\
&\leq \Delta_q\left((\mathbf{S}^\perp)^-, (\mathbf{T}^\perp)^-\right) + \nu_q^{\mathbf{D}}(\mathbf{S})
\end{aligned}$$

which proves the lemma in this case. On the other hand, if $\mathsf{P}^{\mathbf{D}\mathbf{T}^-}[W_q = 1] < \mathsf{P}^{\mathbf{D}\mathbf{S}^-}[W_q = 1]$ then we can easily construct another distinguisher $\mathbf{D}^*$ with the same behavior as $\mathbf{D}$ and the opposite final answer bit. Then we can proceed with the argument as before and since $\Delta_q^{\mathbf{D}}(\mathbf{S}^-, \mathbf{T}^-) = \Delta_q^{\mathbf{D}^*}(\mathbf{S}^-, \mathbf{T}^-)$ and $\nu_q^{\mathbf{D}}(\mathbf{S}) = \nu_q^{\mathbf{D}^*}(\mathbf{S})$, the conclusion is valid also for the distinguisher $\mathbf{D}$. ∎

It was proved in [Mau02] that if for some $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-system $\mathbf{S}$ with MBO $\mathcal{A}$ and an $(\mathcal{X}, \mathcal{Y})$-system $\mathbf{T}$ we have $\mathbf{S}|\mathcal{A} \equiv \mathbf{T}$ then $\mathbf{T}$ can be extended into a game $\hat{\mathbf{T}}$ by adding an MBO such that $\mathbf{S}^\perp \equiv \hat{\mathbf{T}}^\perp$ which also implies $\Delta_q\left((\mathbf{S}^\perp)^-, (\hat{\mathbf{T}}^\perp)^-\right) = 0$. Hence Lemma 2.3 can be seen as a special case of Lemma 2.4 covering the situation when $\Delta_q\left((\mathbf{S}^\perp)^-, (\mathbf{T}^\perp)^-\right) = 0$, i.e., when the distinguished systems behave identically until some conditions are violated. In contrast, Lemma 2.4 is useful in the situations where the systems are not identical even while the conditions are satisfied, but their behavior is very similar, which is quantified by the term $\Delta_q\left((\mathbf{S}^\perp)^-, (\mathbf{T}^\perp)^-\right)$.

The relationship between distinguishing two systems and winning an appropriately defined game was further studied in [MPR07], where the following claim was proved.

Lemma 2.5 *For any two* $(\mathcal{X}, \mathcal{Y})$*-systems* $\mathbf{S}$ *and* $\mathbf{T}$ *there exist* $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$*-systems* $\hat{\mathbf{S}}$ *and* $\hat{\mathbf{T}}$ *such that*

(i) $\hat{\mathbf{S}}^{-} \equiv \mathbf{S}$

(ii) $\hat{\mathbf{T}}^{-} \equiv \mathbf{T}$

(iii) $\hat{\mathbf{S}}^{\perp} \equiv \hat{\mathbf{T}}^{\perp}$

(iv) $\delta_q^{\mathbf{D}}(\mathbf{S}, \mathbf{T}) = \nu_q^{\mathbf{D}}(\hat{\mathbf{S}}) = \nu_q^{\mathbf{D}}(\hat{\mathbf{T}})$ *for all* $\mathbf{D}$.

Intuitively, Lemma 2.5 states that any two systems $\mathbf{S}$ and $\mathbf{T}$ can be extended by adding an MBO to each of them that "signals" whether the system has deviated from the common behavior of both $\mathbf{S}$ and $\mathbf{T}$. The systems are equivalent as long as the MBOs are 0 and the probability that a distinguisher $\mathbf{D}$ turns one of these MBOs to 1 is equal to the statistical distance of transcripts of the experiments $\mathbf{DS}$ and $\mathbf{DT}$.

Moreover, it was proved in [MPR07] that if any $(\mathcal{X}, \mathcal{Y}\times\{0,1\})$-systems $\hat{\mathbf{S}}$ and $\hat{\mathbf{T}}$ satisfy for every $i \geq 1$ the conditions (for $\hat{\mathbf{T}}$, the conditions are analogous)

$$\begin{aligned} \mathsf{p}^{\hat{\mathbf{S}}}_{Y^i A_i | X^i}(y^i, 0, x^i) &= m^{\mathbf{S},\mathbf{T}}_{x^i, y^i} \\ \mathsf{p}^{\hat{\mathbf{S}}}_{Y^i A_i | X^i}(y^i, 1, x^i) &= \mathsf{p}^{\mathbf{S}}_{Y^i | X^i}(y^i, x^i) - m^{\mathbf{S},\mathbf{T}}_{x^i, y^i} \end{aligned} \tag{2.4}$$

where

$$m^{\mathbf{S},\mathbf{T}}_{x^i, y^i} = \min\{\mathsf{p}^{\mathbf{S}}_{Y^i | X^i}(y^i, x^i), \mathsf{p}^{\mathbf{T}}_{Y^i | X^i}(y^i, x^i)\}, \tag{2.5}$$

then they also satisfy the properties stated in Lemma 2.5. In fact, Lemma 2.5 was proved in [MPR07] by demonstrating that the systems $\hat{\mathbf{S}}$ and $\hat{\mathbf{T}}$ satisfying (2.4) can always be constructed.

2.4 Idealized Cryptographic Models

There are several idealized models of cryptographic primitives that are frequently used in various areas of provable security. The unifying idea behind these models is the following: for some real cryptographic object $\mathbf{O}$ often acting as an underlying primitive in more complex cryptographic constructions $\mathsf{C}(\mathbf{O})$, one tries to capture the desired ("ideal") properties of $\mathbf{O}$, hence arriving at its idealized version $\mathbf{I}$. One can then

focus on analyzing the security of the construction $\mathsf{C}(\cdot)$ itself by looking at it when used on top of the ideal primitive $\mathbf{I}$, i.e., one inspects the security of $\mathsf{C}(\mathbf{I})$. In case of successfully proving that $\mathsf{C}(\mathbf{I})$ achieves the desired security, this can be seen as a heuristic argument towards the security of the real construction $\mathsf{C}(\mathbf{O})$ since it excludes any possibility of an attack that would use $\mathbf{O}$ in a black-box way. In other words, any attack would have to exploit some weakness in the structure of the real primitive $\mathbf{O}$, since otherwise it could be also mounted against $\mathsf{C}(\mathbf{I})$.

One reason for the popularity of the idealized models is that typically the description of the ideal object is mathematically elegant and easy to use. This allows for much cleaner security proofs for $\mathsf{C}(\mathbf{I})$ compared to $\mathsf{C}(\mathbf{O})$ and for some constructions these are even the only security proofs known.

However, it was repeatedly observed that the idealized models are not sound, meaning that there exist constructions that can be proved secure using the ideal primitive $\mathbf{I}$ but become insecure when $\mathbf{I}$ is replaced with *any* real implementation $\mathbf{O}$.

We now discuss four important examples of idealized models that we will encounter throughout this thesis.

- The *random oracle model* (ROM) consists of seeing a real-world cryptographic hash function as a random oracle $\mathbf{R}^{*,n}$ as defined in Section 2.3.1. This methodology was explicitly introduced in [BR93] and employed in a vast amount of security proofs, e.g. [FS86, Sch91, BR95, BR96, FOPS01]. Despite being shown unsound in [CGH98, MRH04] it is still widely used and so far no practical instantiations of schemes with a security proof in the ROM have been broken.

- Descending one more level, one can also ask for secure constructions of the random oracle $\mathbf{R}^{*,n}$ from a seemingly simpler (but still ideal) primitive, the *random function* $\mathbf{R}^{m,r}$. This is the problem of (infinite) domain extension and is well-studied within the indifferentiability setting [MRH04, CDMP05].

- The *ideal cipher model* (ICM) is, in turn, used for analyzing constructions based on block ciphers and consists of the assumption that the block cipher used is the ideal cipher $\mathbf{E}$, realizing an independent uniformly random permutation for each possible key. The ICM is widely used to analyze various types of constructions

based on block ciphers (see for example [KR01, BRS02, BR06]) and was shown to be equivalent to the random oracle model [CDMP05, CPS08, HKT11] with respect to classical indifferentiability reductions. The uninstantiability argument from [CGH98, MRH04] was also adopted for the ICM in [Bla05].

- In the ICM one can again consider the question of building the ideal cipher from a simpler ideal primitive. For the special case of block ciphers following the key-alternating approach such as AES [Aes01], one can employ the *random permutation model* (RPM) to perform the analysis. These block ciphers consist of several steps applying the XOR of (a part of) the key, interleaved with applications of some key-independent permutations. In RPM these intermediate permutations are modelled as independent and uniformly random (i.e., as independent copies of the system $\mathbf{P}$ from Section 2.3.1). This approach was taken for example in [EM91, BKL+12].

There exist several other idealized models useful in other areas of cryptography, most notably the *generic group model* and the *generic ring model*. Since their application lies in the area of number-theoretic cryptography we abstain from describing these models in greater detail.

Chapter 3

Free-Start Distinguishing and Indistinguishability Amplification

3.1 Indistinguishability Amplification via Neutralizing Constructions

The central concept that we focus on throughout this chapter is the notion of *indistinguishability amplification* (IA), which we already briefly introduced in Chapter 1. Recall that it refers to the setting where we want to transform an available system $\mathbf{S}$ with certain (unsatisfactory) security properties into a new system of the same type with better security guarantees (in terms of indistinguishability from some ideal target system). We consider the special case of this scenario where we combine multiple weaker systems into the stronger one.

Neutralizing Constructions. There exist several highly efficient constructions that appear to be natural candidates for achieving indistinguishability amplification. We start by mentioning two of them (depicted in Figure 3.1) which we denote with special symbols:

- *Quasi-group combination.* For $(\mathcal{X}, \mathcal{Y})$-random systems $\mathbf{F}$ and $\mathbf{G}$ and

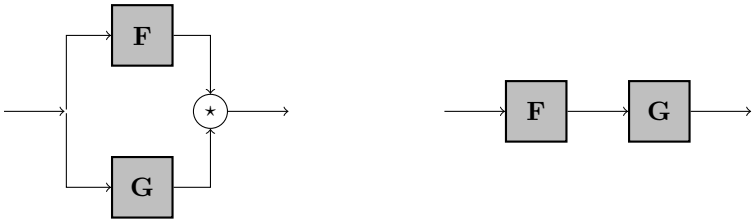

Figure 3.1: Two important examples of neutralizing constructions: $\mathbf{F} \star \mathbf{G}$ (left) and $\mathbf{F} \rhd \mathbf{G}$ (right).

for a quasi-group[4] operation $\star$ on $\mathcal{Y}$, the construction $\mathbf{F} \star \mathbf{G}$ feeds any query it receives to both subsystems and then combines their outputs using $\star$ to determine its own output.

- *Composition.* For a $(\mathcal{X}, \mathcal{Y})$-random system $\mathbf{F}$ and a $(\mathcal{Y}, \mathcal{Z})$-random system $\mathbf{G}$, $\mathbf{F} \rhd \mathbf{G}$ denotes the serial composition of systems: every input to $\mathbf{F} \rhd \mathbf{G}$ is fed to $\mathbf{F}$, its output is fed to $\mathbf{G}$ and the output of $\mathbf{G}$ is the output of $\mathbf{F} \rhd \mathbf{G}$.

Both these constructions are widely used in the design of practical cryptographic primitives such as block and stream ciphers, hence the indistinguishability amplification achieved by them deserves being studied in detail. Both (as well as other natural constructions) are examples from a general class of constructions called *neutralizing* and introduced by Maurer, Pietrzak and Renner in [MPR07].

Definition 3.1 *A construction* C *is* neutralizing *for pairs of systems* $(\mathbf{F}, \mathbf{I})$ *and* $(\mathbf{G}, \mathbf{J})$ *if* $\mathsf{C}(\mathbf{F}, \mathbf{J}) \equiv \mathsf{C}(\mathbf{I}, \mathbf{G}) \equiv \mathsf{C}(\mathbf{I}, \mathbf{J})$.

In the pairs of systems mentioned, the first one represents the real system while the second one represents its ideal counterpart. Intuitively, a construction is neutralizing if it combines the available systems in such a way that if at least one of the provided systems is ideal then the whole construction behaves as if both of them were ideal. Sometimes constructions with this property are also called combiners. One can easily see that the quasi-group combination is a neutralizing construction for any

[4] A binary operation $\star$ on $\mathcal{X}$ is a quasi-group operation if for every $a, c \in \mathcal{X}$ (every $b, c \in \mathcal{X}$) there is a unique $b \in \mathcal{X}$ ($a \in \mathcal{X}$) such that $a \star b = c$.

random functions $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{I} \equiv \mathbf{J} \equiv \mathbf{R}$, while the serial composition is neutralizing for any random permutation $\mathbf{F}$, any cc-stateless random permutation $\mathbf{G}$ and $\mathbf{I} \equiv \mathbf{J} \equiv \mathbf{P}$[5].

Previous Work on IA. Indistinguishability-amplifying constructions have been intensively studied both in the information-theoretic and the computational setting.

In the former, the most general treatment is given in the above-mentioned work [MPR07]. Two different types of indistinguishability amplification – corresponding to the quantitative and qualitative form sketched in Chapter 1 – are presented here. Both are proved for the general class of neutralizing constructions, but for clarity we also illustrate their contribution on the special case of the XOR of random functions $\mathbf{F} \oplus \mathbf{G}$.

- *Advantage Amplification.* By combining two real systems via a neutralizing construction, the best advantage in distinguishing the resulting system from the ideal one diminishes, compared to the individual distinguishing advantages of the original systems. This amplification has the form of a product theorem, stating that, for example, the advantage in distinguishing $\mathbf{F} \oplus \mathbf{G}$ from the uniform random function $\mathbf{R}$ is upper-bounded by twice the *product* of the individual distinguishing advantages for these functions from $\mathbf{R}$.

- *Distinguisher Class Amplification.* The neutralizing combination of two systems is indistinguishable from the ideal system by the general class of all distinguishers even if the individual systems used are only indistinguishable for a particular restricted subclass of distinguishers. For example, the advantage in distinguishing $\mathbf{F} \oplus \mathbf{G}$ from $\mathbf{R}$ adaptively is upper-bounded by the *sum* of advantages in distinguishing $\mathbf{F}$ and $\mathbf{G}$ from $\mathbf{R}$ non-adaptively.

This work was preceded by several others, focusing on one particular form of indistinguishability amplification or on a specific construction. A product theorem for the composition of stateless permutations was proved by Vaudenay using the decorrelation framework [Vau00, Vau03]; the amplification of the distinguisher class was proved in [MP04] for a

[5]Recall that throughout this chapter, $\mathbf{P}$ denotes a URP allowing only forward queries. Of course, since we study the general class of neutralizing constructions, our results can also be used for constructions that are neutralizing also for permutations accessible in both directions, as done in [MPR07].

particular class of constructions and in [MOPS06] also for the four-round Feistel network.

On the other hand, in the computational setting product theorems for various constructions were proved by Luby and Rackoff [LR86], Myers [Mye03] and Dodis et al. [DIJK09]. For the general case of a neutralizing construction a product theorem was proved by Maurer and Tessaro [MT09] and later a tight version for the case of a plain cascade was obtained by Tessaro [Tes11]. The second type of amplification considered above, amplification of the distinguisher class, does not in general translate to the computational setting, as observed by Pietrzak [Pie05].

3.2 Overview and Motivation

In this chapter our main goal is to unify the two aspects of information-theoretic indistinguishability amplification discussed above. Towards this aim, we first extend the random system framework described in Section 2.3 in which our analysis is performed. We introduce the concept of a system *projected to a specific state*. Loosely speaking, any properly defined discrete system $\mathbf{S}$ and a transcript t of interaction with this system together define a new system, which behaves as the original system $\mathbf{S}$ would behave after this interaction t. We refer to this new system as $\mathbf{S}$ projected to the state described by t. In particular, we already explained in Section 2.3 that any one-player game can be modelled as a special type of a discrete system. Therefore, we are also able to model the intuitive situation where a player can continue playing a given game from a specific position (where the game is not won yet) or where it can pick an arbitrary such position in the game tree and try to win the game from there.

This leads us to the central new notion we introduce, *free-start distinguishing*. Informally, the free-start distinguishing advantage of two systems is the best advantage a distinguisher can achieve, assuming that it is allowed to project both the distinguished systems to any one state consistent with both of them and then try to distinguish the resulting systems.

This concept, besides giving an interesting new viewpoint on the distinguishing of random systems, allows us to perform a more careful analysis of the indistinguishability amplification achieved by neutralizing constructions in the information-theoretic setting. We use the notion of free-start distinguishing to combine the two types of amplification described above by deriving a new bound which keeps the structure of a

product theorem, while involving also the non-adaptive distinguishing advantages, thus describing the amplification of the distinguisher class. Now we describe on an intuitive level how such a statement is obtained.

As observed in [MPR07], there is a tight correspondence between distinguishing systems and winning an appropriately defined game. Distinguishing $\mathbf{F} \oplus \mathbf{G}$ from $\mathbf{R}$ can be reduced (by a factor of 2) to winning two games constructed from $\mathbf{F}$ and $\mathbf{G}$, while obtaining only the XOR of their outputs. As long as neither of the games is won, the output of the construction is useless to the player, hence one of the games has to be won non-adaptively first. After achieving this, the player still has to win the other game, this time with access to some (possibly useful) outputs. Since winning each of these games is as hard as distinguishing the corresponding system from $\mathbf{R}$, one could conjecture a bound like

$$\Delta_q(\mathbf{F} \oplus \mathbf{G}, \mathbf{R}) \leq 2\left(\Delta_q^{\mathsf{NA}}(\mathbf{F}, \mathbf{R}) \cdot \Delta_q(\mathbf{G}, \mathbf{R}) + \Delta_q^{\mathsf{NA}}(\mathbf{G}, \mathbf{R}) \cdot \Delta_q(\mathbf{F}, \mathbf{R})\right),$$

where $\Delta_q(\mathbf{S}, \mathbf{T})$ and $\Delta_q^{\mathsf{NA}}(\mathbf{S}, \mathbf{T})$ denote the adaptive and non-adaptive advantage in distinguishing $\mathbf{S}$ from $\mathbf{T}$ with q queries, respectively.

However, this reasoning is not correct since winning the first game may involve getting the second game into a state where winning it becomes much easier than if played from the beginning. We model this by allowing the player to choose the starting position in the second game freely, with the only restriction being that the game is not won yet in the chosen position. Translated back into the language of systems distinguishing, this gives us a valid bound

$$\Delta_q(\mathbf{F} \oplus \mathbf{G}, \mathbf{R}) \leq 2\left(\Delta_q^{\mathsf{NA}}(\mathbf{F}, \mathbf{R}) \cdot \Lambda_q(\mathbf{G}, \mathbf{R}) + \Delta_q^{\mathsf{NA}}(\mathbf{G}, \mathbf{R}) \cdot \Lambda_q(\mathbf{F}, \mathbf{R})\right),$$

where $\Lambda_q(\mathbf{S}, \mathbf{T})$ denotes the free-start distinguishing advantage for systems $\mathbf{S}$ and $\mathbf{T}$ as described above. Indeed, the main result of this chapter is Theorem 3.1 representing a general statement for *all* neutralizing constructions, of which the bound above is a simple corollary. The results presented in this chapter were published in [GM10].

3.3 Projected Systems

We start by introducing the notion of projected systems. Any system $\mathbf{S}$ and a transcript of the initial part of a possible interaction with it together define a new system that simulates the behavior of $\mathbf{S}$ from the state at the end of this interaction onwards. This is formalized in the following definition.

Definition 3.2 *For an $(\mathcal{X}, \mathcal{Y})$-random system $\mathbf{S}$ and $(\overline{x}^j, \overline{y}^j) \in \mathcal{X}^j \times \mathcal{Y}^j$, let $\mathbf{S}[\overline{x}^j, \overline{y}^j]$ denote the system $\mathbf{S}$ projected to the state $\overline{x}^j\overline{y}^j$, i.e. the random system that behaves like $\mathbf{S}$ would behave after answering the first j queries $\overline{x}^j$ by $\overline{y}^j$. Formally, $\mathbf{S}[\overline{x}^j, \overline{y}^j]$ is defined by the distributions*

$$\mathsf{p}^{\mathbf{S}[\overline{x}^j,\overline{y}^j]}_{Y_i|X^iY^{i-1}}(y_i, x^i, y^{i-1}) := \mathsf{p}^{\mathbf{S}}_{Y_{j+i}|X^{j+i}Y^{j+i-1}}(y_i, \overline{x}^j x^i, \overline{y}^j y^{i-1})$$

if $\mathsf{p}^{\mathbf{S}}_{Y^j|X^j}(\overline{y}^j, \overline{x}^j) > 0$ and undefined otherwise.

This can be best illustrated if we consider a game (i.e., a special type of system with an MBO, see Section 2.3.4), where the transcript represents a position in this game. For a $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-system $\mathbf{S}$ representing a game, the MBO bits are also a part of the output, therefore we have to specify them when describing its answers to the first j queries. To denote a position where the game is not won yet, we set these bits to 0, obtaining the system $\mathbf{S}[\overline{x}^j, \overline{y}^j 0^j]$. We can now ask what is the probability of winning the game from this position.

Definition 3.3 *Let $\mathbf{S}$ be a $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-system with the MBO A_i and let $\mathbf{D}$ be a compatible player. Let $j \le q$ be non-negative integers. For any $x^j \in \mathcal{X}^j$ and $y^j \in \mathcal{Y}^j$ such that $\mathsf{p}^{\mathbf{S}}_{Y^jA_j|X^j}(y^j, 0, x^j) > 0$, we call $\nu^{\mathbf{D}}_{q-j}(\mathbf{S}[x^j, y^j 0^j])$ the probability of* $\mathbf{D}$ winning the game $\mathbf{S}$ from the position $x^j y^j$ *within the remaining $q - j$ queries. Moreover, we also define the probability of winning* $\mathbf{S}$ *within q queries* with a free start *to be*

$$\lambda_q(\mathbf{S}) := \max_{j, x^j, y^j} \nu_{q-j}(\mathbf{S}[x^j, y^j 0^j]),$$

where the maximization goes over all $j \le q, x^j, y^j$ such that the projected system $\mathbf{S}[x^j, y^j 0^j]$ is defined.

Intuitively, if a player starts playing the game $\mathbf{S}$ from the position $x^j y^j$ (assuming the game is not won yet), $\nu_{q-j}(\mathbf{S}[x^j, y^j 0^j])$ describes the probability that it wins the game within the remaining $q - j$ queries if he plays optimally from now on. On the other hand, if the player is allowed to choose *any* position in the game tree within the first q queries (where the game is not won yet) and play from that position, it can win with probability $\lambda_q(\mathbf{S})$. Clearly we have $\lambda_q(\mathbf{S}) \ge \nu_q(\mathbf{S})$, the player can always decide to play the game from the beginning. Also note that depending on the game $\mathbf{S}$, any $j \in \{0, \dots, q-1\}$ may maximize the term $\nu^{\mathbf{D}}_{q-j}(\mathbf{S}[x^j, y^j 0^j])$ in the definition of $\lambda_q(\mathbf{S})$.

Let us now consider a construction $\mathsf{C}(\mathbf{S}_1, \mathbf{S}_2)$. In this section we assume that $\mathbf{S}_1$ and $\mathbf{S}_2$ are two $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-systems (games) with MBOs A_i and B_i, respectively. Moreover, we assume that $\mathsf{C}(\mathbf{S}_1, \mathbf{S}_2)$ is a $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-construction and it combines the last binary outputs of its subsystems using the AND operation to determine its own binary output C_i. Note that although the construction may determine the number and ordering of the queries to its subsystems adaptively, we can assume that the order of the queries to the subsystems is well-defined for every run of the experiment. This justifies the following definition.

Definition 3.4 *In the experiment* $\mathbf{D}\mathsf{C}(\mathbf{S}_1, \mathbf{S}_2)$*, let* F_j^i *denote the event that the game* $\mathbf{S}_i$ *was won during the first* j *queries to* $\mathsf{C}(\mathbf{S}_1, \mathbf{S}_2)$ *and it was the first of the games* $\mathbf{S}_1$*,* $\mathbf{S}_2$ *that was won.*

Note that if both games are to be won, one of them always has to be won first. Afterwards, the adversary needs to also win the second game in order to provoke the MBO of the whole construction. This is captured by the following lemma. Recall that q_1 and q_2 denote the maximal possible number of queries made to the first and second subsystem, respectively, during the first q queries issued to the construction (if defined).

Lemma 3.1 *Let* $\mathbf{S}$ *denote the system* $\mathsf{C}(\mathbf{S}_1, \mathbf{S}_2)$ *with MBO as described above. Then we have*

$$\nu_q^{\mathbf{D}}(\mathbf{S}) \le \mathsf{P}^{\mathbf{DS}}(F_q^1) \cdot \lambda_{q_2}(\mathbf{S}_2) + \mathsf{P}^{\mathbf{DS}}(F_q^2) \cdot \lambda_{q_1}(\mathbf{S}_1).$$

Proof: Since the MBO of $\mathbf{S}$ is the AND of the MBOs of the subsystems, we have

$$\begin{aligned} \nu_q^{\mathbf{D}}(\mathbf{S}) &\le \mathsf{P}^{\mathbf{DS}}(F_q^1 \wedge B_{q_2}) + \mathsf{P}^{\mathbf{DS}}(F_q^2 \wedge A_{q_1}) \\ &= \mathsf{P}^{\mathbf{DS}}(F_q^1) \cdot \mathsf{P}^{\mathbf{DS}}(B_{q_2} | F_q^1) + \mathsf{P}^{\mathbf{DS}}(F_q^2) \cdot \mathsf{P}^{\mathbf{DS}}(A_{q_1} | F_q^2). \end{aligned}$$

It remains to upper-bound the terms $\mathsf{P}^{\mathbf{DS}}(B_{q_2}|F_q^1)$ and $\mathsf{P}^{\mathbf{DS}}(A_{q_1}|F_q^2)$. Let X_i and Y_i be the random variables corresponding to the i-th input and $\mathcal{Y}$-output of $\mathbf{S}$, respectively; and let M_i and N_i (U_i and V_i) be the random variables corresponding to the i-th input and $\mathcal{Y}$-output of $\mathbf{S}_1$ ($\mathbf{S}_2$), respectively. Let T denote the random variable corresponding to the initial part of the transcript of the experiment from its beginning until the MBO A is provoked or until the end of the experiment, whichever comes first. This transcript contains all the queries X_i to the construction, all

the corresponding answers (Y_i, C_i), as well as all the query-answer pairs $(M_i, (N_i, A_i))$ and $(U_i, (V_i, B_i))$ of the subsystems, in the order as they appeared during the execution. Conditioning over all possible values of T, we have

$$\mathsf{P}^{\mathbf{DS}}(B_{q_2}|F_q^1) = \sum_t \mathsf{P}^{\mathbf{DS}}_{T|F_q^1}(t) \cdot \mathsf{P}^{\mathbf{DS}}_{B_{q_2}|TF_q^1}(t). \tag{3.1}$$

Let now t be fixed such that $\mathsf{P}^{\mathbf{DS}}_{T|F_q^1}(t) > 0$, we need to prove $\mathsf{P}^{\mathbf{DS}}_{B_{q_2}|TF_q^1}(t) \leq \lambda_{q_2}(\mathbf{S}_2)$. Let us consider a player $\mathbf{D}'$ defined as follows: it simulates the behavior of the player $\mathbf{DC}(\mathbf{S}_1, \cdot)$. However, as long as the MBO A is not provoked, all its choices are fixed to follow the transcript t. After these "cheated" choices, as soon as the MBO A is provoked (and t ends), it simulates $\mathbf{D}$, C and $\mathbf{S}_1$ faithfully. Let j denote the number of queries issued to $\mathbf{S}_2$ in t, let u^j and v^j denote these queries and the corresponding answers, respectively. For the described player $\mathbf{D}'$, we have

$$\begin{aligned}
\mathsf{P}^{\mathbf{DS}}_{B_{q_2}|TF_q^1}(t) &= \mathsf{P}^{\mathbf{D}'\mathbf{S}_2}_{B_{q_2}|U^jV^j\overline{B_j}}(u^j, v^j) \\
&\leq \max_{\mathbf{D}} \mathsf{P}^{\mathbf{DS}_2}_{B_{q_2}|U^jV^j\overline{B_j}}(u^j, v^j) \\
&= \nu_{q_2-j}(\mathbf{S}[u^j, v^j 0^j]) \\
&\leq \lambda_{q_2}(\mathbf{S}_2),
\end{aligned}$$

and since $\sum_t \mathsf{P}^{\mathbf{DS}}_{T|F_q^1}(t) = 1$, from (5.2) we have $\mathsf{P}^{\mathbf{DS}}(B_{q_2}|F_q^1) \leq \lambda_{q_2}(\mathbf{S}_2)$. The same argument gives us a symmetric bound for $\mathsf{P}^{\mathbf{DS}}(A_{q_1}|F_q^2)$ and concludes the proof. ∎

3.4 Free-Start Distinguishing

The notion of winning a game with a free start, captured by the quantity $\lambda_q(\mathbf{S})$, has a counterpart in the language of systems indistinguishability, which we now define formally.

Definition 3.5 *For any $(\mathcal{X}, \mathcal{Y})$-random systems $\mathbf{S}$ and $\mathbf{T}$, we define the* free-start distinguishing advantage of $\mathbf{S}$ and $\mathbf{T}$ *to be*

$$\Lambda_q(\mathbf{S}, \mathbf{T}) := \max_{j, x^j, y^j} \Delta_{q-j}(\mathbf{S}[x^j, y^j], \mathbf{T}[x^j, y^j]),$$

where the maximization goes over all $j \in \{0, \ldots, q-1\}$ and all $x^j \in \mathcal{X}, y^j \in \mathcal{Y}$ such that the systems on the right side are defined.

Informally, suppose that the distinguisher is allowed to choose an arbitrary transcript $x^j y^j$ compatible with both the systems it is supposed to distinguish, project them to the states described by this transcript and then try to distinguish the resulting systems with the remaining $q - j$ queries. Then the quantity $\Lambda_q(\mathbf{S}, \mathbf{T})$ denotes the optimal advantage it can achieve.

To demonstrate the relationship between λ_q and Λ_q, we exploit the connection between distinguishing two systems and winning an appropriately defined game described in [MPR07]. Let us consider the setting with a real system $\mathbf{F}$ (e.g. a random function) and an ideal system $\mathbf{I}$ (e.g. a uniform random function). Using Lemma 2.5 (and, in particular, condition (2.4)), we can add MBOs to the systems $\mathbf{F}$ and $\mathbf{I}$ to obtain systems $\hat{\mathbf{F}}$ and $\hat{\mathbf{I}}$ such that $\nu_q(\langle\hat{\mathbf{F}}/\hat{\mathbf{I}}\rangle) = \Delta_q(\mathbf{F}, \mathbf{I})$ and the systems behave identically as long as the MBO is not provoked. Since provoking this MBO corresponds to distinguishing the systems, one can expect $\nu_{q-j}(\langle\hat{\mathbf{F}}/\hat{\mathbf{I}}\rangle[x^j, y^j 0^j])$ to be related to the advantage in distinguishing $\mathbf{F}$ and $\mathbf{I}$ projected to the state described by the transcript $x^j y^j$ on the remaining $q - j$ queries. In the following, we confirm this intuition.

Lemma 3.2 *Let* $\mathbf{F}$ *and* $\mathbf{I}$ *be two random systems, let* $\hat{\mathbf{F}}$, $\hat{\mathbf{I}}$ *be the systems obtained from* $\mathbf{F}$, $\mathbf{I}$ *by adding the MBOs according to Lemma 2.5 and condition (2.4). Then we have*

$$\nu_q(\langle\hat{\mathbf{F}}/\hat{\mathbf{I}}\rangle[\overline{x}^j, \overline{y}^j 0^j]) = \Delta_q(\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]^-, \hat{\mathbf{I}}[\overline{x}^j, \overline{y}^j 0^j]^-)$$

for any $\overline{x}^j$,$\overline{y}^j$ *such that the system on the left side is defined.*

Proof: First note that $\nu_q(\langle\hat{\mathbf{F}}/\hat{\mathbf{I}}\rangle[\overline{x}^j, \overline{y}^j 0^j]) = \nu_q(\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j])$, hence it suffices to prove $\nu_q(\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]) = \Delta_q(\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]^-, \hat{\mathbf{I}}[\overline{x}^j, \overline{y}^j 0^j]^-)$. We prove this claim by showing that the MBO of $\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]$, originally defined to capture the differences between $\mathbf{F}$ and $\mathbf{I}$, keeps the properties guaranteed by Lemma 2.5 also with respect to the systems $\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]^-$ and $\hat{\mathbf{I}}[\overline{x}^j, \overline{y}^j 0^j]^-$. We achieve this by showing that the system $\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]$ satisfies the condition (2.4) with respect to the systems $\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]^-$ and $\hat{\mathbf{I}}[\overline{x}^j, \overline{y}^j 0^j]^-$. Seeing this, the claim follows from Lemma 2.5.

Throughout the proof let p denote the probability $\mathsf{p}^{\hat{\mathbf{F}}}_{Y^j A^j|X^j}(\overline{y}^j, 0^j, \overline{x}^j) = \mathsf{p}^{\hat{\mathbf{I}}}_{Y^j A^j|X^j}(\overline{y}^j, 0^j, \overline{x}^j)$ (by the assumptions of the lemma, $p > 0$). We first show that the relevant probabilities describing the behavior of the random system $\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]$ (and $\hat{\mathbf{I}}[\overline{x}^j, \overline{y}^j 0^j]$) correspond

to the probabilities describing the original system $\hat{\mathbf{F}}$ (and $\hat{\mathbf{I}}$) scaled by the factor $1/p$. Recall from Section 2.1 that $\mathrm{ms}(i)$ denotes the set of monotone binary sequences of length i where zeroes precede ones. We have

$$\begin{aligned}
\mathsf{p}_{Y^i|X^i}^{\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]}(y^i,x^i) &= \sum_{a^i\in\mathrm{ms}(i)} \mathsf{p}_{Y^iA^i|X^i}^{\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]}(y^i,a^i,x^i) \\
&= \frac{1}{p}\cdot\sum_{a^i\in\mathrm{ms}(i)} \mathsf{p}_{Y^{j+i}A^{j+i}|X^{j+i}}^{\hat{\mathbf{F}}}(\overline{y}^j y^i,0^j a^i,\overline{x}^j x^i) \\
&= \frac{1}{p}\cdot \mathsf{p}_{Y^{j+i}A^j|X^{j+i}}^{\hat{\mathbf{F}}}(\overline{y}^j y^i,0^j,\overline{x}^j x^i)
\end{aligned}$$

and similarly $\mathsf{p}_{Y^i|X^i}^{\hat{\mathbf{I}}[\overline{x}^j,\overline{y}^j 0^j]}(y^i,x^i) = \frac{1}{p}\cdot \mathsf{p}_{Y^{j+i}A^j|X^{j+i}}^{\hat{\mathbf{I}}}(\overline{y}^j y^i,0^j,\overline{x}^j x^i)$. We can use this to express the quantity $m_{x^i,y^i}^{\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-,\hat{\mathbf{I}}[\overline{x}^j,\overline{y}^j 0^j]^-}$ (see (2.5)) as

$$\begin{aligned}
m_{x^i,y^i}^{\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-,\hat{\mathbf{I}}[\overline{x}^j,\overline{y}^j 0^j]^-} &= \min\left\{\mathsf{p}_{Y^i|X^i}^{\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-}(y^i,x^i), \mathsf{p}_{Y^i|X^i}^{\hat{\mathbf{I}}[\overline{x}^j,\overline{y}^j 0^j]^-}(y^i,x^i)\right\} \\
&= \frac{1}{p}\cdot\min\Big\{\mathsf{p}_{Y^{j+i}A^j|X^{j+i}}^{\hat{\mathbf{F}}}(\overline{y}^j y^i,0^j,\overline{x}^j x^i), \\
&\qquad\qquad \mathsf{p}_{Y^{j+i}A^j|X^{j+i}}^{\hat{\mathbf{I}}}(\overline{y}^j y^i,0^j,\overline{x}^j x^i)\Big\} \\
&= \frac{1}{p}\cdot \mathsf{p}_{Y^{j+i}A^{j+i}|X^{j+i}}^{\hat{\mathbf{F}}}(\overline{y}^j y^i,0^{j+i},\overline{x}^j x^i) \qquad (3.2) \\
&= \frac{1}{p}\cdot m_{\overline{x}^j x^i,\overline{y}^j y^i}^{\hat{\mathbf{F}},\hat{\mathbf{I}}}.
\end{aligned}$$

To justify the step (3.2), note that from the condition (2.4), which is satisfied for $\hat{\mathbf{F}}$ and $\hat{\mathbf{I}}$, we have

$$\mathsf{p}_{Y^{j+i}A^{j+i}|X^{j+i}}^{\hat{\mathbf{F}}}(\overline{y}^j y^i,0^{j+i},\overline{x}^j x^i) = \mathsf{p}_{Y^{j+i}A^{j+i}|X^{j+i}}^{\hat{\mathbf{I}}}(\overline{y}^j y^i,0^{j+i},\overline{x}^j x^i)$$

and also

$$\mathsf{p}_{Y^{j+i}A^j|X^{j+i}}(\overline{y}^j y^i,0^j,\overline{x}^j x^i) = \mathsf{p}_{Y^{j+i}A^{j+i}|X^{j+i}}(\overline{y}^j y^i,0^{j+i},\overline{x}^j x^i)$$

for at least one of the systems $\hat{\mathbf{F}}$ and $\hat{\mathbf{I}}$.

Now we can verify that the condition (2.4) is satisfied also for the system $\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]$ with respect to the systems $\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-$ and $\hat{\mathbf{I}}[\overline{x}^j,\overline{y}^j 0^j]^-$. For the first equation of (2.4), we have

$$\begin{aligned}
\mathsf{p}_{Y^iA_i|X^i}^{\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]}(y^i,0,x^i) &= \frac{1}{p}\cdot \mathsf{p}_{Y^{j+i}A_{j+i}|X^{j+i}}^{\hat{\mathbf{F}}}(\overline{y}^j y^i,0,\overline{x}^j x^i) \\
&= \frac{1}{p}\cdot m_{\overline{x}^j x^i,\overline{y}^j y^i}^{\hat{\mathbf{F}},\hat{\mathbf{I}}} = m_{x^i,y^i}^{\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-,\hat{\mathbf{I}}[\overline{x}^j,\overline{y}^j 0^j]^-}
\end{aligned}$$

and since clearly $\mathsf{p}^{\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]}_{Y^i|X^i}(y^i,x^i) = \mathsf{p}^{\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-}_{Y^i|X^i}(y^i,x^i)$, the second equation of (2.4) is satisfied as well. Therefore, by Lemma 2.5(iv), we have $\nu_q(\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]) = \Delta_q(\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-, \hat{\mathbf{I}}[\overline{x}^j,\overline{y}^j 0^j]^-)$. ∎

Lemma 3.2 involves the systems $\hat{\mathbf{F}}$ and $\hat{\mathbf{I}}$ projected to a specific state, but it is more desirable to consider projections of the original systems $\mathbf{F}$ and $\mathbf{I}$ instead. This is achieved by the following lemma.

Lemma 3.3 *In the setting described in Lemma 3.2, we have*

$$\Delta_q(\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-, \hat{\mathbf{I}}[\overline{x}^j,\overline{y}^j 0^j]^-) \le \Delta_q(\mathbf{F}[\overline{x}^j,\overline{y}^j], \mathbf{I}[\overline{x}^j,\overline{y}^j])$$

for any $\overline{x}^j$,$\overline{y}^j$ such that the systems on the left side are defined.

Proof: To prove the lemma, we show that for any distinguisher $\mathbf{D}$ we have

$$\delta^{\mathbf{D}}_q(\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-, \hat{\mathbf{I}}[\overline{x}^j,\overline{y}^j 0^j]^-) \le \delta^{\mathbf{D}}_q(\mathbf{F}[\overline{x}^j,\overline{y}^j], \mathbf{I}[\overline{x}^j,\overline{y}^j]).$$

Without loss of generality, let us assume $\mathsf{p}^{\mathbf{F}}_{Y^j|X^j}(\overline{y}^j,\overline{x}^j) \ge \mathsf{p}^{\mathbf{I}}_{Y^j|X^j}(\overline{y}^j,\overline{x}^j)$, otherwise the proof would be symmetric. This assumption implies $\hat{\mathbf{I}}[\overline{x}^j,\overline{y}^j 0^j]^- \equiv \mathbf{I}[\overline{x}^j,\overline{y}^j]$, hence it suffices to prove

$$\delta^{\mathbf{D}}_q(\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-, \mathbf{I}[\overline{x}^j,\overline{y}^j]) \le \delta^{\mathbf{D}}_q(\mathbf{F}[\overline{x}^j,\overline{y}^j], \mathbf{I}[\overline{x}^j,\overline{y}^j]).$$

Using (2.3) to express both sides of this inequality, we see that we only need to prove that for all $x^q \in \mathcal{X}^q$ and $y^q \in \mathcal{Y}^q$,

$$\begin{array}{c} \mathsf{p}^{\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-}_{Y^q|X^q}(y^q,x^q) < \mathsf{p}^{\mathbf{I}[\overline{x}^j,\overline{y}^j]}_{Y^q|X^q}(y^q,x^q) \\ \text{implies} \\ \mathsf{p}^{\mathbf{F}[\overline{x}^j,\overline{y}^j]}_{Y^q|X^q}(y^q,x^q) \le \mathsf{p}^{\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-}_{Y^q|X^q}(y^q,x^q). \end{array} \tag{3.3}$$

In the systems $\mathbf{I}[\overline{x}^j,\overline{y}^j]$, $\mathbf{F}[\overline{x}^j,\overline{y}^j]$ and $\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-$, the conditional distributions $\mathsf{p}_{Y^q|X^q}(y^q,x^q)$ are given by the following expressions, respectively:

$$\mathsf{p}^{\mathbf{I}[\overline{x}^j,\overline{y}^j]}_{Y^q|X^q}(y^q,x^q) = \frac{\mathsf{p}^{\mathbf{I}}_{Y^{j+q}|X^{j+q}}(\overline{y}^j y^q,\overline{x}^j x^q)}{\mathsf{p}^{\mathbf{I}}_{Y^j|X^j}(\overline{y}^j,\overline{x}^j)} \tag{3.4}$$

$$\mathsf{p}^{\mathbf{F}[\overline{x}^j,\overline{y}^j]}_{Y^q|X^q}(y^q,x^q) = \frac{\mathsf{p}^{\mathbf{F}}_{Y^{j+q}|X^{j+q}}(\overline{y}^j y^q,\overline{x}^j x^q)}{\mathsf{p}^{\mathbf{F}}_{Y^j|X^j}(\overline{y}^j,\overline{x}^j)} \tag{3.5}$$

$$\mathsf{p}^{\hat{\mathbf{F}}[\overline{x}^j,\overline{y}^j 0^j]^-}_{Y^q|X^q}(y^q,x^q) = \frac{\mathsf{p}^{\hat{\mathbf{F}}}_{Y^{j+q}A^j|X^{j+q}}(\overline{y}^j y^q,0^j,\overline{x}^j x^q)}{\mathsf{p}^{\hat{\mathbf{F}}}_{Y^j A^j|X^j}(\overline{y}^j,0^j,\overline{x}^j)} \tag{3.6}$$

Informally, the conditional distributions $\mathsf{p}_{Y^q|X^q}$ of the systems $\mathbf{I}[\overline{x}^j, \overline{y}^j]$, $\mathbf{F}[\overline{x}^j, \overline{y}^j]$ and $\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]^-$ are again related to the conditional distributions $\mathsf{p}_{Y^{j+q}|X^{j+q}}$ of the original systems ($\mathbf{I}$, $\mathbf{F}$, and $\hat{\mathbf{F}}$ with $A_j = 0$, respectively) by some scaling factors (the denominators in the above equations). The factor turns out to be the same for $\mathbf{I}[\overline{x}^j, \overline{y}^j]$ and $\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]^-$, however for $\mathbf{F}[\overline{x}^j, \overline{y}^j]$ it may be different. This results into a different scaling of the distributions for $\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]^-$ and $\mathbf{F}[\overline{x}^j, \overline{y}^j]$ and allows us to show that (3.3) is indeed satisfied. A more detailed argument follows.

Let us fix x^q and y^q such that $\mathsf{p}_{Y^q|X^q}^{\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]^-}(y^q, x^q) < \mathsf{p}_{Y^q|X^q}^{\mathbf{I}[\overline{x}^j, \overline{y}^j]}(y^q, x^q)$. By the definition of A_i we have $\mathsf{p}_{Y^j|X^j}^{\mathbf{I}}(\overline{y}^j, \overline{x}^j) = \mathsf{p}_{Y^j A^j|X^j}^{\hat{\mathbf{F}}}(\overline{y}^j, 0^j, \overline{x}^j)$, hence by comparing the equations (3.4) and (3.6) we get

$$\mathsf{p}_{Y^{j+q}A^j|X^{j+q}}^{\hat{\mathbf{F}}}(\overline{y}^j y^q, 0^j, \overline{x}^j x^q) < \mathsf{p}_{Y^{j+q}|X^{j+q}}^{\mathbf{I}}(\overline{y}^j y^q, \overline{x}^j x^q).$$

This in turn implies

$$\mathsf{p}_{Y^{j+q}A^{j+q}|X^{j+q}}^{\hat{\mathbf{F}}}(\overline{y}^j y^q, 0^{j+q}, \overline{x}^j x^q) < \mathsf{p}_{Y^{j+q}|X^{j+q}}^{\mathbf{I}}(\overline{y}^j y^q, \overline{x}^j x^q).$$

Now, recalling that the MBO A_i is defined to satisfy the properties (2.4), we see that $\mathsf{p}_{Y^{j+q}|X^{j+q}}^{\hat{\mathbf{F}}}(\overline{y}^j y^q, \overline{x}^j x^q) < \mathsf{p}_{Y^{j+q}|X^{j+q}}^{\mathbf{I}}(\overline{y}^j y^q, \overline{x}^j x^q)$ and therefore also $\mathsf{p}_{Y^{j+q}A^{j+q}|X^{j+q}}^{\hat{\mathbf{F}}}(\overline{y}^j y^q, 0^{j+q}, \overline{x}^j x^q) = \mathsf{p}_{Y^{j+q}|X^{j+q}}^{\mathbf{F}}(\overline{y}^j y^q, \overline{x}^j x^q)$. This in turn implies $\mathsf{p}_{Y^{j+q}A^j|X^{j+q}}^{\hat{\mathbf{F}}}(\overline{y}^j y^q, 0^j, \overline{x}^j x^q) = \mathsf{p}_{Y^{j+q}|X^{j+q}}^{\mathbf{F}}(\overline{y}^j y^q, \overline{x}^j x^q)$, hence the numerators in (3.5) and (3.6) are the same. The denominators are easy to compare, it obviously holds $\mathsf{p}_{Y^j|X^j}^{\mathbf{F}}(\overline{y}^j, \overline{x}^j) \geq \mathsf{p}_{Y^j A^j|X^j}^{\hat{\mathbf{F}}}(\overline{y}^j, 0^j, \overline{x}^j)$, hence from (3.5) and (3.6) we obtain $\mathsf{p}_{Y^q|X^q}^{\mathbf{F}[\overline{x}^j, \overline{y}^j]}(y^q, x^q) \leq \mathsf{p}_{Y^q|X^q}^{\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]^-}(y^q, x^q)$, completing the proof of the implication (3.3). ∎

Note that combining the technical Lemmas 3.2 and 3.3 gives us

$$\lambda_q(\langle \hat{\mathbf{F}}/\hat{\mathbf{I}} \rangle) = \max_{j, \overline{x}^j, \overline{y}^j} \Delta_{q-j}\left(\hat{\mathbf{F}}[\overline{x}^j, \overline{y}^j 0^j]^-, \hat{\mathbf{I}}[\overline{x}^j, \overline{y}^j 0^j]^-\right) \leq \Lambda_q(\mathbf{F}, \mathbf{I}) \qquad (3.7)$$

for the systems described above.

3.5 Connection to Indistinguishability Amplification

We are now ready to prove our main theorem. First we define some natural notation: by $\mathcal{DC}(\cdot, \mathbf{J})$ we denote the class of distinguishers obtained

by connecting any distinguisher to $\mathsf{C}(\cdot, \mathbf{J})$ and placing the system to be distinguished as the first subsystem. The class of distinguishers $\mathcal{DC}(\mathbf{I}, \cdot)$ is defined analogously.

Theorem 3.1 *Let $\mathsf{C}(\cdot,\cdot)$ be a neutralizing construction for the pairs $(\mathbf{F}, \mathbf{I})$ and $(\mathbf{G}, \mathbf{J})$ of systems. Let $\mathbf{Q}$ denote the system $\mathsf{C}(\mathbf{I}, \mathbf{J})$. Then, for all q,*

$$\begin{aligned}\Delta_q(\mathsf{C}(\mathbf{F}, \mathbf{G}), \mathbf{Q}) \quad \leq \quad & 2\left(\delta_{q_1}^{\mathcal{DC}(\cdot,\mathbf{J})}(\mathbf{F}, \mathbf{I}) \cdot \Lambda_{q_2}(\mathbf{G}, \mathbf{J}) + \right. \\ & \left. + \, \delta_{q_2}^{\mathcal{DC}(\mathbf{I},\cdot)}(\mathbf{G}, \mathbf{J}) \cdot \Lambda_{q_1}(\mathbf{F}, \mathbf{I})\right).\end{aligned}$$

Proof: We use the technique from the proof of Theorem 1 in [MPR07] to transform the task of distinguishing $\mathsf{C}(\mathbf{F}, \mathbf{G})$ from $\mathbf{Q}$ to the task of provoking the MBO of the system $\mathbf{S} := \hat{\mathbf{C}}(\langle \hat{\mathbf{F}}/\hat{\mathbf{I}} \rangle_{Z_1}, \langle \hat{\mathbf{G}}/\hat{\mathbf{J}} \rangle_{Z_2})$, where $\hat{\mathbf{F}}$, $\hat{\mathbf{I}}$ and $\hat{\mathbf{G}}$, $\hat{\mathbf{J}}$ are obtained using Lemma 2.5 from $\mathbf{F}$, $\mathbf{I}$ and $\mathbf{G}$, $\mathbf{J}$, respectively; and $\hat{\mathbf{C}}$ is the same construction as C except that it also has an MBO, which is defined as the AND of the two internal MBOs. Then we use a different approach to bound the value $\nu_q(\mathbf{S})$, exploiting the concept of free-start distinguishing.

First, by Lemma 2.1 (ii) we have

$$\Delta_q(\mathsf{C}(\mathbf{F}, \mathbf{G}), \mathbf{Q}) = 2 \cdot \Delta_q(\langle \mathsf{C}(\mathbf{F}, \mathbf{G})/\mathbf{Q} \rangle_Z, \mathbf{Q})$$

and by Lemma 2.1 (i) $\Delta_q(\langle \mathsf{C}(\mathbf{F}, \mathbf{G})/\mathbf{Q} \rangle_Z, \mathbf{Q})$ is the optimal advantage in guessing the uniform random bit Z' in the system $\langle\langle \mathsf{C}(\mathbf{F}, \mathbf{G})/\mathbf{Q} \rangle_Z/\mathbf{Q} \rangle_{Z'}$. However, thanks to the neutralizing property of $\mathsf{C}(\cdot,\cdot)$ it can be easily verified that

$$\langle\langle \mathsf{C}(\mathbf{F}, \mathbf{G})/\mathbf{Q} \rangle_Z/\mathbf{Q} \rangle_{Z'} \equiv \mathsf{C}(\langle \mathbf{F}/\mathbf{I} \rangle_{Z_1}, \langle \mathbf{G}/\mathbf{J} \rangle_{Z_2})$$

for independent uniformly random bits $Z_1 := Z$ and $Z_2 := Z \oplus Z'$. Hence, $\Delta_q(\langle \mathsf{C}(\mathbf{F}, \mathbf{G})/\mathbf{Q} \rangle_Z, \mathbf{Q})$ is also the optimal advantage in guessing the bit $Z' = Z_1 \oplus Z_2$ in $\mathsf{C}(\langle \mathbf{F}/\mathbf{I} \rangle_{Z_1}, \langle \mathbf{G}/\mathbf{J} \rangle_{Z_2})$.

We can now extend the systems $\mathbf{F}$ and $\mathbf{I}$ by adding MBOs satisfying the equations (2.4) to obtain the systems $\hat{\mathbf{F}}$ and $\hat{\mathbf{I}}$ with the properties guaranteed by Lemma 2.5. Similarly, we can extend $\mathbf{G}$ and $\mathbf{J}$ and obtain the systems $\hat{\mathbf{G}}$ and $\hat{\mathbf{J}}$. Since the MBO in $\mathbf{S}$ can always be ignored, the task of guessing $Z_1 \oplus Z_2$ can only be easier in $\mathbf{S}$ compared to $\mathsf{C}(\langle \mathbf{F}/\mathbf{I} \rangle_{Z_1}, \langle \mathbf{G}/\mathbf{J} \rangle_{Z_2})$. However, as long as one of the MBOs in the subsystems of $\mathbf{S}$ is 0, the advantage in guessing the corresponding bit Z_i is 0 and hence also the

advantage in guessing $Z_1 \oplus Z_2$ is 0. Therefore the latter advantage can be upper-bounded by $\nu_q(\mathbf{S})$.

Using Lemma 3.1, for any distinguisher $\mathbf{D}$ we have

$$\nu_q^{\mathbf{D}}(\mathbf{S}) \leq \mathsf{P}^{\mathbf{DS}}(F_q^1) \cdot \lambda_{q_2}(\langle \hat{\mathbf{G}}/\hat{\mathbf{J}} \rangle) + \mathsf{P}^{\mathbf{DS}}(F_q^2) \cdot \lambda_{q_1}(\langle \hat{\mathbf{F}}/\hat{\mathbf{I}} \rangle).$$

Let us first bound the term $\mathsf{P}^{\mathbf{DS}}(F_q^1)$. Since $\langle \hat{\mathbf{F}}/\hat{\mathbf{I}} \rangle^{\perp} \equiv \hat{\mathbf{F}}^{\perp}$ and $\langle \hat{\mathbf{G}}/\hat{\mathbf{J}} \rangle^{\perp} \equiv \hat{\mathbf{J}}^{\perp}$, we have $\mathsf{P}^{\mathbf{DS}}(F_q^1) = \mathsf{P}^{\mathbf{D}\hat{\mathbf{C}}(\hat{\mathbf{F}},\hat{\mathbf{J}})}(F_q^1)$. Moreover, $\mathsf{P}^{\mathbf{D}\hat{\mathbf{C}}(\hat{\mathbf{F}},\hat{\mathbf{J}})}(F_q^1) \leq \nu_q^{\mathbf{D}}(\mathsf{C}(\hat{\mathbf{F}},\mathbf{J}))$ since on the left side, we only consider the MBO of $\hat{\mathbf{F}}$ being provoked first, while on the right side is the probability of it being provoked at any time. Clearly $\nu_q^{\mathbf{D}}(\mathsf{C}(\hat{\mathbf{F}},\mathbf{J})) \leq \nu_{q_1}^{\mathbf{D}\mathsf{C}(\cdot,\mathbf{J})}(\hat{\mathbf{F}})$ and by Lemma 2.5 we have $\nu_{q_1}^{\mathbf{D}\mathsf{C}(\cdot,\mathbf{J})}(\hat{\mathbf{F}}) = \delta_{q_1}^{\mathbf{D}\mathsf{C}(\cdot,\mathbf{J})}(\mathbf{F},\mathbf{I})$. By a symmetric reasoning we obtain $\mathsf{P}^{\mathbf{DS}}(F_q^2) \leq \delta_{q_2}^{\mathbf{D}\mathsf{C}(\mathbf{I},\cdot)}(\mathbf{G},\mathbf{J})$.

Finally, using (3.7) we obtain the bounds $\lambda_{q_2}(\langle \hat{\mathbf{G}}/\hat{\mathbf{J}} \rangle) \leq \Lambda_{q_2}(\mathbf{G},\mathbf{J})$ and $\lambda_{q_1}(\langle \hat{\mathbf{F}}/\hat{\mathbf{I}} \rangle) \leq \Lambda_{q_1}(\mathbf{F},\mathbf{I})$, which together conclude the proof. ∎

For the two particular neutralizing constructions depicted in Figure 3.1 that motivated our analysis, we obtain the following corollaries.

Corollary 3.1 *Let* $\mathbf{F}$ *and* $\mathbf{G}$ *be* $(\mathcal{X},\mathcal{Y})$*-random functions, let* $\star$ *be a quasi-group operation on* $\mathcal{Y}$*. Then, for all* q*,*

$$\Delta_q(\mathbf{F} \star \mathbf{G}, \mathbf{R}) \leq 2\left(\Delta_q^{\mathsf{NA}}(\mathbf{F},\mathbf{R}) \cdot \Lambda_q(\mathbf{G},\mathbf{R}) + \Delta_q^{\mathsf{NA}}(\mathbf{G},\mathbf{R}) \cdot \Lambda_q(\mathbf{F},\mathbf{R})\right).$$

Proof: Applying Theorem 3.1 to the neutralizing construction $\mathbf{F} \star \mathbf{G}$, it only remains to prove that $\mathcal{D}(\cdot \star \mathbf{R})$ corresponds to the class of non-adaptive distinguishers. This is indeed the case, since any distinguisher will only receive random outputs from $\mathbf{F} \star \mathbf{R}$. It could simulate these outputs itself, ignoring the actual outputs, thus operating non-adaptively. The same holds for the class of distinguishers $\mathcal{D}(\mathbf{R} \star \cdot)$. Recalling that $\delta_q^{\mathsf{NA}}(\mathbf{S},\mathbf{T}) = \Delta_q^{\mathsf{NA}}(\mathbf{S},\mathbf{T})$ for any systems $\mathbf{S}$, $\mathbf{T}$ completes the proof. ∎

Corollary 3.2 *Let* $\mathbf{F}$ *and* $\mathbf{G}$ *be* $(\mathcal{X},\mathcal{X})$*-random permutations, let* $\mathbf{G}$ *be cc-stateless. Then, for all* q*,*

$$\Delta_q(\mathbf{F} \triangleright \mathbf{G}, \mathbf{P}) \leq 2\left(\Delta_q^{\mathsf{NA}}(\mathbf{F},\mathbf{P}) \cdot \Lambda_q(\mathbf{G},\mathbf{P}) + \Delta_q^{\mathsf{RI}}(\mathbf{G},\mathbf{P}) \cdot \Lambda_q(\mathbf{F},\mathbf{P})\right).$$

Proof: Again, when applying Theorem 3.1 to the neutralizing construction $\mathbf{F} \rhd \mathbf{G}$, we need to justify that the distinguisher classes $\mathcal{D}(\cdot \rhd \mathbf{P})$ and $\mathcal{D}(\mathbf{P} \rhd \cdot)$ correspond to NA and RI, respectively. In the first case, the distinguisher only receives random outputs, so it can again simulate them itself and hence corresponds to a non-adaptive distinguisher. In the second case, the distinguisher $\mathbf{D}(\mathbf{P} \rhd \cdot)$ can only provide random inputs to the distinguished system, with the possibility of repeating an input. However, since both $\mathbf{G}$ and $\mathbf{P}$ are cc-stateless permutations, repeated inputs will only produce repeated outputs and cannot help the distinguisher. ∎

3.6 Further Discussion

Our main theorem unifies the claims of both Theorem 1 and Theorem 2 in [MPR07] under reasonable assumptions. To see this, let us focus for example on the natural case of random functions, assuming $\mathbf{F} \equiv \mathbf{G}$ and $\mathbf{I} \equiv \mathbf{J} \equiv \mathbf{R}$. Our theorem gives a better bound than Theorem 2 in [MPR07] as long as $\Lambda_q(\mathbf{F}, \mathbf{R}) < 1/2$. It also improves the bound from Theorem 1 in [MPR07] as long as

$$\frac{\Lambda_q(\mathbf{F}, \mathbf{R})}{\Delta_q(\mathbf{F}, \mathbf{R})} < \frac{1}{2} \cdot \frac{\Delta_q(\mathbf{F}, \mathbf{R})}{\Delta_q^{\mathsf{NA}}(\mathbf{F}, \mathbf{R})}.$$

This means, loosely speaking, that the improvement occurs as long as the ratio of advantage gained from the free choice of state is smaller than the ratio of advantage gained from extending the distinguisher class.

This improvement is significant for any random function $\mathbf{F}$ that satisfies the conditions

$$\Delta_q^{\mathsf{NA}}(\mathbf{F}, \mathbf{R}) \ll \Delta_q(\mathbf{F}, \mathbf{R}) \approx \Lambda_q(\mathbf{F}, \mathbf{R}) \ll 1.$$

As an example, consider the simple cc-stateless random function $\mathbf{F}\colon \{0,1\}^n \to \{0,1\}^n$ that behaves as follows: with probability $2^{-n/2}$ it satisfies the (adaptively verifiable) condition $\mathbf{F}(\mathbf{F}(0)) = 0$ and the remaining values (including $\mathbf{F}(0)$) are chosen uniformly at random, in the rest of the cases (with probability $1 - 2^{-n/2}$) $\mathbf{F}$ behaves exactly like $\mathbf{R}$.

In general, a small $\Delta_q(\mathbf{F}, \mathbf{R})$ does not necessarily imply a small $\Lambda_q(\mathbf{F}, \mathbf{R})$, since it is easy to construct a counterexample where some specific initial transcript leads to a behavior that is easy to distinguish from

the ideal system. However, a small value of $\Lambda_q(\mathbf{F}, \mathbf{R})$ may be considered a desirable requirement for a good quasi-random function.

Chapter 4

Efficient Key-Length Extension for Block Ciphers

It is beyond question that block ciphers play a pivotal role in current cryptographic practice. Most practical constructions of secret-key primitives such as (randomized and deterministic) message encryption schemes and message authentication codes (MACs) inherently rely on a block cipher, and even in *public* key scenarios, block-cipher based symmetric encryption is widely employed in view of its efficiency once an ephemeral shared key is established.

Several practical block cipher designs have been proposed over the last decades and have been the object of extensive cryptanalytic efforts. Examples include DES [Des77], IDEA [LM90], BLOWFISH [Sch94], and the currently in-use AES [Aes01].

Formally, a block cipher with keyspace $\{0,1\}^\kappa$ and message space $\{0,1\}^n$ is nothing but a family of efficiently computable (and invertible) permutations E_k on the set of n-bit strings indexed by a κ-bit key k. One can see it as a mapping $E\colon \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ such that for each $k \in \{0,1\}^\kappa$, $E(k,\cdot)$ is a permutation on the set $\{0,1\}^n$. Typically $E_k(x)$ is written instead of $E(k,x)$ and $E_k^{-1}(\cdot)$ refers to the inverse of the permutation $E_k(\cdot)$. We usually refer to such a block cipher as a (κ, n)-block cipher. For example, $n = 64$ and $\kappa = 56$ for DES, and $n = 128$ and $\kappa \in \{128, 192, 256\}$ for AES.

Within applications, we typically demand that a block cipher is a good *pseudorandom permutation* (PRP), i.e., when used with a random secret

key, in the eyes of a computationally bounded attacker it behaves as a permutation chosen uniformly at random. For instance, PRP security of the underlying block cipher is necessary to infer security of all modes of operations for message encryption (such as counter-mode and CBC encryption [BDJR97]) as well as of message authentication codes like CBC-MAC [BKR94] and PMAC [BR02].

4.1 The Key-Length Extension Problem

In practice, we define the PRP security level of a block cipher as the complexity required to distinguish it from a random permutation with non-negligible advantage. The *key length* κ of a block cipher crucially limits the achievable security level, since the secret key K can be recovered given black-box access to $E(K,\cdot)$ and evaluating $E(\cdot,\cdot)$ approximately 2^κ times by the following trivial brute-force attacker:

1. compute $y := E_K(0^n)$ via the provided oracle E_K
2. for all $k \in \{0,1\}^\kappa$ compute $y^{(k)} := E_k(0^n)$ yourself
3. for every match $y = y^{(k)}$, verify with a different input

Obviously, this key-recovery attack also yields a PRP distinguishing adversary with equal complexity. The weakness represented by this attack is *generic*, in the sense that it only depends on κ, and no amount of ingenuity put into the design of a particular block cipher can prevent it. In contrast, no such immediate dependency exists between security and the block length n of a block cipher.

Key Length Extension. Due to the above reason, the continuous increase of the availability of computing resources makes the key length a crucial security parameter of every block cipher. Key lengths of say fewer than 64 bits are no longer sufficient to ensure security, making key recovery a matter of a few hours even on modest architectures. This is a serious problem for legacy designs such as DES which have very short keys of length 56 bits, but which otherwise do not seem to present significant non-generic security weaknesses. Constructions based on DES also remain very attractive because of its short block length $n = 64$ which allows enciphering short inputs. This is for example a compelling reason for using DES in current applications in the financial industry, such

as the EMV standard [EMV08], where the block cipher is applied to PIN numbers, which are very short.

The above described situation motivates the problem of *key-length extension*, which is the main topic of this chapter: We seek for very efficient constructions provably transforming any (κ, n)-block cipher E into a (κ', n)-block cipher E' with both $\kappa' > \kappa$ and higher PRP security, i.e., the PRP security of E' should be higher than 2^κ whenever E does not exhibit any non-generic weaknesses. We aim both at providing very efficient approaches to key length extension and at understanding the optimal security achievable by such constructions.

Before turning to our results, we discuss known results and constructions for key-length extension to place our work into perspective.

4.1.1 Existing Approaches

The short key length $\kappa = 56$ of DES has constituted the main motivation behind previous work on key-length extension. However, we stress that all previous constructions are generic, and can be applied to *any* block cipher with short keys, hence extending the applicability of these results (as well as the results of our own) way beyond the specific case of DES.

Key Whitening. A first proposal called DESX (due to Rivest) stretches the key length of DES by employing a technique called *key whitening* (this approach was later used by Even and Mansour [EM91]). The construction DESX takes one 56-bit encyption key k and two 64-bit whitening keys k_i and k_o and for a message block $m \in \{0,1\}^{64}$ its operation is defined by

$$\mathrm{DESX}_{k_i,k_o,k}(m) = k_o \oplus \mathrm{DES}_k(k_i \oplus m).$$

DESX can be generalized to a generic transformation from a (κ, n)-block cipher to a $(\kappa + 2n, n)$-block cipher whose security was studied by Kilian and Rogaway [KR01]: They proved that any successful PRP distinguishing attack requires $2^{\frac{\kappa+n}{2}}$ queries. (Their result is in fact more fine-grained, as they show that 2^ρ construction and $2^{\kappa+n-\rho}$ ideal block cipher queries, respectively, are necessary for all integers ρ. While different bounds for both query types are sometimes justified, here we adopt the standard worst-case approach only bounding the *sum* of both query numbers. Nevertheless, we comment on possible trade-offs between types of queries where applicable.) In [KR01] it is also observed that the same key may be used in both whitening steps (i.e., $k_i = k_o$) and an attack using $2^{\max\{\kappa,n\}}$ queries is provided.

Cascading. An alternative to the approach of key whitening is *cascading* (or cascade encryption). This refers to sequentially composing ℓ block-cipher calls, typically with independently chosen keys. (Such a construction is called a cascade of length ℓ or an ℓ-cascade.)

The security properties of a cascade of different ciphers was studied by Even and Goldreich [EG85] who show that a cascade of ciphers is at least as strong as the *strongest* of the ciphers against attacks that are restricted to operating on full blocks. In contrast, Maurer and Massey [MM93] show that for the most general attack model, where it is for example possible that an attacker might obtain only half the ciphertext block for a chosen message block, the cascade is only at least as strong as the *first* cipher of the cascade.

It is well known that a cascade of length two does not substantially increase security due to the meet-in-the-middle attack [DH77], even though a security increase in terms of distinguishing advantage is achieved for low attack complexities [ABCV98]. This makes triple encryption the shortest cascade with a potential for significant security gain and indeed it has found widespread usage as *Triple-DES* (3DES) [3DE98, 3DE99, 3DE04]. Given keys $k_1, k_2, k_3 \in \{0,1\}^{56}$, 3DES encrypts a 64-bit message m as

$$\mathrm{3DES}_{k_1,k_2,k_3}(m) = \mathrm{DES}_{k_3}(\mathrm{DES}_{k_2}(\mathrm{DES}_{k_1}(m))) \ .$$

Sometimes a variant with shorter keys is used, which we will denote by 3DES′ and is defined by

$$\mathrm{3DES}'_{k_1,k_2}(m) = \mathrm{DES}_{k_1}(\mathrm{DES}^{-1}_{k_2}(\mathrm{DES}_{k_1}(m))).$$

The security of 3DES (and a variant of 3DES′ with independent keys) was formally studied by Bellare and Rogaway [BR06], showing its security up to roughly $2^{\kappa+\min\{n,\kappa\}/2}$ queries when DES is replaced by an ideal block cipher. For the case of DES parameters, their result gives concretely security up to the threshold value of 2^{78} queries, whereas the best known attack due to Lucks [Luc98] shows that no security better than 2^{90} can be expected.

We emphasize that despite the availability of modern designs with built-in larger keys (e.g., $\kappa \in \{128, 192, 256\}$ for AES), Triple-DES remains nowadays popular, not only because of backwards compatibility, but also because its short block size ($n = 64$ vs. $n \geq 128$ for AES) is well suited to applications enciphering short inputs such as personal-identification numbers (PINs). For example, it is the basis of the EMV standard for

PIN-based authentication of debit and credit card transactions [EMV08]. However, the use of three calls per processed message block is widely considered a drawback within applications. One of the constructions proposed in this chapter can be seen as a solution to this problem.

Other Related Work. A large body of work is devoted to studying cascading-based security amplification of block ciphers only assumed to satisfy weaker forms of PRP security, both in the information-theoretic [Vau00, Vau03, MP04, MPR07] as well as in the computational settings [LR86, MT09, Tes11, DDKS12] (see also discussion in Section 3.1). These results however consider an orthogonal model to ours and are hence incomparable.

4.1.2 Formalization in the Ideal Cipher Model

Since our goal is to assess the security level achieved by various key-length extension constructions, in our proofs we assume the absence of generic weaknesses of the underlying block cipher. We achieve this by utilizing the ideal cipher model as described in Section 2.4. The value of a security proof for a key-length extending construction in the ideal cipher model is that it rules out the existence of any generic attacks, treating the underlying cipher as a black-box.

The Setting. We consider a distinguisher $\mathbf{D}$ that is allowed to issue the following two types of queries:

- *block cipher queries* that allow the distinguisher to evaluate the block cipher $\mathbf{E}$ under any key of its choice and on any message block (either in the encryption or the decryption direction).
- *construction queries* that allow the distinguisher to query either the evaluated key-length extending construction C used with the block cipher $\mathbf{E}$ and a random secret key K; or a random permutation $\mathbf{P}$ independent of $\mathbf{E}$. Again, both query directions are allowed. The goal of the distinguisher is to decide which of these two cases occurred.

To capture this setting formally, unless stated otherwise we consider all the resources mentioned in this chapter (such as $\mathbf{P}$ and $\mathbf{E}$) as having two identical interfaces[6] both providing access to the same instance of the

[6] left and right, see discussion of 2-interface resources in Section 2.2

respective random function. We can then capture the advantage achieved by a distinguisher $\mathbf{D}$ in distinguishing the two settings described above as $\Delta^{\mathbf{D}}(\mathsf{C}_K\mathbf{E}, \mathbf{P}\lfloor\mathbf{E})$. Recall from Section 2.2 that the symbol "$\lfloor$" stands for combining the two 2-interface resources involved in such a way that $\mathbf{P}\lfloor\mathbf{E}$ exposes the left interface of $\mathbf{P}$ as its left interface and the right interface of $\mathbf{E}$ as its right interface, while the other interfaces of both systems are simply "blinded". Hence, the queries to the left interface correspond to the construction queries, while the queries to the right interface correspond to the block cipher queries. This is also true for the system $\mathsf{C}_K\mathbf{E}$, here we naturally randomize also over the choice of the key K.

Performance Measure. The distinguisher's complexity is measured solely in terms of the number of its queries, hence our results are of information-theoretic nature. Note that the security of *any* key-length extension construction in our model can be upper-bounded by $2^{\kappa+n}$ which corresponds to the trivial attack asking all possible block cipher and construction queries.

We remark that this way to capture the security of a key-length extending construction was already used in [KR01, BR06], while the work [ABCV98] uses an analogous model where they only allow forward construction queries.

Relation to Indifferentiability. The type of statements we sketched above (and we make throughout this chapter) can be seen also as an indifferentiability reduction statement, constructing a private random permutation for the honest parties out of a public ideal block cipher (and a secret key possessed only by the honest parties). For a detailed description of the indifferentiability setting, see Section 5.1. Proving an indistinguishability statement of the form above is equivalent to stating the indifferentiability of the construction $\mathsf{C}_K\mathbf{E}$ from the ideal resource $\mathbf{P}\lfloor$ (accessible only via the left interface), with the simulator having a particular form: it simulates the real resource $\mathbf{E}$ by plain lazy sampling, not having any access to the ideal resource at all.

4.1.3 Overview of Our Results

Our contribution to the topic of key-length extension consists of two flavours of results. First, we give security lower bounds, showing that for some particular construction, the distinguishing problem described in Section 4.1.2 is hard for any distinguisher issuing less than some necessary amount of queries. Second, we present generic attacks that give

an upper bound on the security of certain general classes of constructions in our model and constitute a natural counterpart to the results of the first type. For some of the constructions we achieve a tight bound on its security by obtaining matching lower and upper bounds, as detailed below.

Cascades. We start our investigation by looking at the case of a plain cascade construction. In Section 4.2 we address the security of cascades of the general length ℓ, which is pointed out as an open problem in the previously mentioned work [BR06]. The lower bound on security that we achieve improves with the length ℓ of the cascade for all block ciphers where $\kappa < n$. With increasing cascade length, the bound approaches very roughly the value $2^{\kappa+\min\{n/2,\kappa\}}$ (the exact formula can be found in Theorem 4.1 on page 66). The condition $\kappa < n$ is satisfied for example for the prominent example of the DES block cipher, where the length of the key is 56 bits and the length of one block is 64 bits. For these parameters, the result from [BR06] (in which we fix some minor technical issues along the way) proves that the triple encryption is secure up to 2^{78} queries, but our result shows that a cascade of length 5 is secure up to 2^{83} queries. The larger the difference $n - \kappa$, the more a longer cascade can help.

As a complement to the above-mentioned positive result, in Section 4.3 we present a general attack on ℓ-cascade in our model that requires roughly $2^{\kappa+\frac{\ell-2}{\ell}n}$ queries ($2^{\kappa+\frac{\ell-1}{\ell+1}n}$ queries) for even (odd) ℓ. The well-known meet-in-the-middle attack and the attack of Lucks [Luc98] turn out to be special cases of our attack for $\ell = 2$ and $\ell = 3$, respectively.

XOR-Cascades. After exploring the security of the seemingly simplest possible construction — the cascade — we turn our attention to constructions that are a little more involved. We shall consider the so-called ℓ-XOR-cascades that, loosely speaking, consist of ℓ cascaded encryption steps interleaved with key-whitening steps using the XOR operation (see Figure 4.3 on page 79). The justification of several design decisions for this construction (such as no whitening step after the last encryption, potential dependency of the keys used) follows from the security proofs and is deferred to later sections, here we just summarize the obtained results.

In Section 4.4.2 we give a general method to reduce the security of XOR-cascades in our model to the security of so-called key-alternating ciphers (described in Section 4.4.1) in the random permutation model. We then employ this approach to prove that 2-XOR-cascade is secure up to $2^{\kappa+n/2}$ queries. By applying existing recent results [BKL+12, Ste12, LPS12]

on the security of key-alternating ciphers we also obtain that 3-XOR-cascade and 4-XOR-cascade are secure up to $2^{\kappa+\frac{2}{3}n}$ and $2^{\kappa+\frac{3}{4}n}$ queries, respectively; and finally, that a general ℓ-XOR-cascade of odd (even) length is secure up to $2^{\kappa+\frac{\ell-1}{\ell+1}n}$ queries ($2^{\kappa+\frac{\ell-2}{\ell}n}$ queries), respectively. Contrasting these results with the generic attacks on plain cascades, we see for example that in our model the 3-XOR-cascade is provably at least as secure as a 6-cascade and a 4-XOR-cascade is at least as secure as an 8-cascade, while providing much better efficiency.

Generic Attacks. Motivated by the question of tightness of the above-mentioned bounds, we proceed by investigating generic attacks on several classes of key-length extending constructions. In Section 4.5 we prove that for constructions issuing at most one block cipher call per invocation, regardless of the amount of key material employed an attack with query complexity $2^{\max\{\kappa,n\}}$ always exists (using memory $2^{\max\{\kappa,n\}}$),[7] showing the optimality of DESX-like constructions in the case $\kappa = n$.

We then turn to two-call constructions, which are necessary to achieve higher security: Here, we prove that any construction for which distinct inputs map to distinct first queries and distinct answers from the first call imply distinct inputs to the second call admits a distinguishing attack making $2^{\kappa+n/2}$ ideal block cipher queries and 2^n construction queries.

Proceeding to the general case of constructions issuing ℓ queries to the block cipher, we restrict ourselves to those that work in a sequential way: they consist of interleaved applications of block cipher steps and applications of some permutations that only depend on the key used. For this class of constructions that we call *sequential* we exhibit an attack requiring $2^{\kappa+\frac{\ell-1}{\ell}n}$ queries. Note that the above-discussed XOR-cascades belong to the class of sequential constructions, hence we see that ℓ-XOR-cascade cannot be secure beyond $2^{\kappa+\frac{\ell-1}{\ell}n}$ queries. This shows that the obtained security bounds for $\ell \in \{2,3,4\}$ are tight and moreover, the ℓ-XOR-cascades of this length are optimally secure among the class of all sequential constructions.

Practicality of 2-XOR-Cascade. Having this understanding of the landscape of key-length extending constructions, it seems that one its noteworthy implication is the impressive security/efficiency ratio of the 2-XOR-cascade construction studied in Section 4.4.3. It provides a comparable or higher security than triple encryption (such as in the widely-used

[7] More precisely, our attack requires roughly 2^κ ideal block cipher queries and 2^n construction queries.

3DES) while being extremely efficient — it only requires two block cipher calls and $\kappa + n$ key bits. While being optimally secure within the class of injective 2-call constructions, we also show other supporting arguments in favour of this construction in Section 4.5.4. Namely, we prove that simpler randomization methods for 2-cascades admit distinguishing attacks with even lower complexity. For example, randomizing the cascade of length two as $E_{k_2}(E_{k_1}(m \oplus z_1)) \oplus z_2$ instead of using our approach yields a simple $2^{\max\{\kappa,n\}}$ meet-in-the middle attack. This shows an interesting feature of the design of the 2-XOR-cascade, namely that while targeting CCA security (i.e., we allow for forward and backward queries to the construction), its design is asymmetric.

Final Remarks. Table 4.1 summarizes the results of this chapter in the context of previously known results. To serve as an overview, some bounds are presented in a simplified form.

Finally, note that all generic attacks presented in this chapter can be mounted even if the distinguisher is only allowed to ask forward construction queries. Moreover, these queries can be chosen arbitrarily, giving us again a *known-plaintext attack*. In contrast, our security proofs are valid also with respect to an adversary allowed to ask inverse construction queries (CCA adversary).

This chapter covers the contributions given in the publications [GM09, GT12] as well as some so far unpublished results. Note that the security analysis of the plain cascade given in Section 4.2 was already presented in a different cast in [Gaž10] and is given here for completeness.

4.2 Security of the Cascade Construction

The main result in this section is Theorem 4.1 which proves the plain cascade construction of length ℓ to be secure up to roughly $\min\left\{2^{\frac{2\ell\kappa}{\ell+1}}, 2^{\kappa+\frac{n}{2}}\right\}$ queries for odd values of ℓ, which also implies a similar bound for one-step longer even-length cascades, as discussed later.

We start by describing how we formalize the cascaded encryption. Let $\mathsf{Casc}_\ell\colon (\{0,1\}^\kappa)^\ell \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ denote a (deterministic stateless) construction which on its inner interface expects a subsystem $\mathbf{E}\colon \{0,1\}^\kappa \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ realizing a block cipher. Casc_ℓ then realizes cascaded encryption of length ℓ using the

ℓ	ℓ-cascade		ℓ-XOR-cascade	sequential ℓ-query construction
	security	attack	security	attack
2	κ	κ	$\kappa + \frac{n}{2}$	$\kappa + \frac{n}{2}$
3	$\kappa + \min\left\{\frac{\kappa}{2}, \frac{n}{2}\right\}$	$\kappa + \frac{n}{2}$	$\kappa + \frac{2}{3}n$	$\kappa + \frac{2}{3}n$
4	$\kappa + \min\left\{\frac{\kappa}{2}, \frac{n}{2}\right\}$	$\kappa + \frac{n}{2}$	$\kappa + \frac{3}{4}n$	$\kappa + \frac{3}{4}n$
odd ≥ 5	$\min\left\{\frac{2\ell\kappa}{\ell+1}, \kappa + \frac{n}{2}\right\}$	$\kappa + \frac{\ell-1}{\ell+1}n$	$\kappa + \frac{\ell-1}{\ell+1}n$	$\kappa + \frac{\ell-1}{\ell}n$
even	$\min\left\{\frac{2(\ell-1)\kappa}{\ell}, \kappa + \frac{n}{2}\right\}$	$\kappa + \frac{\ell-2}{\ell}n$	$\kappa + \frac{\ell-2}{\ell}n$	

Table 4.1: Security lower bounds and best known attacks for various key-length extension schemes. Each given term is a logarithm of the respective number of queries and is parameterized by the key length κ and block size n of the underlying block cipher. References and further details to all depicted bounds are given in the text.

block cipher $\mathbf{E}$ and the keys given, i.e., Casc_ℓ answers each forward query $(k_1, \ldots, k_\ell, x, +)$ by $\mathbf{E}_{k_\ell}(\cdots \mathbf{E}_{k_1}(x) \cdots)$ and each backward query $(k_1, \ldots, k_\ell, y, -)$ by $\mathbf{E}_{k_1}^{-1}(\cdots \mathbf{E}_{k_\ell}^{-1}(y) \cdots)$. For a randomly chosen (secret) key tuple $\bar{K} = (K_1, \ldots, K_\ell) \xleftarrow{\$} (\{0,1\}^\kappa)^\ell$ (and a randomly chosen tuple of distinct keys $\widetilde{K} = (K_1, \ldots, K_\ell) \xleftarrow{\$,\mathrm{d}} (\{0,1\}^\kappa)^\ell$), we let $\mathsf{Casc}_{\ell,\bar{K}}$ (resp. $\mathsf{Casc}_{\ell,\widetilde{K}}$) be the construction which gives bidirectional access to the permutation given by Casc_ℓ under key $\bar{K}$ (resp. $\widetilde{K}$). In other words, both $\mathsf{Casc}_{\ell,\bar{K}}$ and $\mathsf{Casc}_{\ell,\widetilde{K}}$ take outer queries from $\{0,1\}^n \times \{+,-\}$. The evaluation of the construction $\mathsf{Casc}_{\ell,\bar{K}}$ is depicted in Figure 4.1.

4.2.1 Block Cipher Structure and Chains

We introduce some additional notions related to the cascade encryption setting. Our terminology follows and extends that in [BR06].

A block cipher E can be seen as a directed graph consisting of 2^n vertices representing the message space and $2^{n+\kappa}$ edges. Each vertex x

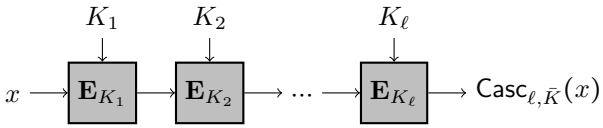

Figure 4.1: Evaluation of the ℓ-cascade construction by the left interface of $\mathsf{Casc}_{\ell,\bar{K}}\mathbf{E}$.

has 2^κ outgoing edges pointing to the encryptions of the message x using all possible keys. Each of the edges is labeled by the respective key. For a fixed block cipher E, we denote by

$$w(E) = \max_{x,y} |\{k \mid E_k(x) = y\}|$$

the maximal number of distinct keys mapping the plaintext x onto the ciphertext y, the maximum taken over all pairs of blocks (x, y). Intuitively, $w(E)$ is the weight of the heaviest edge in the graph corresponding to E. This also naturally defines a random variable $w(\mathbf{E})$ for the random system $\mathbf{E}$ realizing the ideal block cipher.

If a distinguisher makes queries to a block cipher E, let $x \xrightarrow{k} y$ denote the fact that it either made a query $E_k(x)$ and received the encryption y or made a query $E_k^{-1}(y)$ and received the decryption x. An *r-chain for keys* $(k_1, \ldots, k_r)$ is an $(r+1)$-tuple $(x_0, k_1, \ldots, k_r)$ for which there exist $x_1, \ldots, x_r$ such that $x_0 \xrightarrow{k_1} x_1 \xrightarrow{k_2} \cdots \xrightarrow{k_r} x_r$ holds. Similarly, if a fixed permutation Q is given and $1 \le i < r$, then an *i-disconnected r-chain for keys* $(k_1, \ldots, k_r)$ with respect to Q is an $(r+1)$-tuple $(x_0, k_1, \ldots, k_r)$ for which there exist $x_1, \ldots, x_r$ such that we have both

$$x_0 \xrightarrow{k_{r-i+1}} x_1 \xrightarrow{k_{r-i+2}} \cdots \xrightarrow{k_r} x_i \text{ and } Q^{-1}(x_i) \xrightarrow{k_1} x_{i+1} \xrightarrow{k_2} \cdots \xrightarrow{k_{r-i}} x_r.$$

When describing chains, we sometimes explicitly refer to the permutations instead of the keys that define them. For disconnected chains, we sometimes omit the reference to the permutation Q if it is clear from the context. The purpose of the following definition will be clear from the proof of Theorem 4.1.

Definition 4.1 *Let Q be a fixed permutation. A distinguisher* examines *the key tuple $(k_1, k_2, \ldots, k_r)$ w.r.t. Q if it creates either an r-chain or an i-disconnected r-chain w.r.t. Q for $(k_1, k_2, \ldots, k_r)$ for any $i \in \{1, \ldots, r-1\}$.*

4.2.2 The Main Argument

Now we take the key step in establishing the lower bound on the security of the plain cascade encryption of length ℓ. Our approach is inspired by that used in [BR06] to analyze triple encryption, with the necessary modifications required by the more general setting. Using Lemma 2.4 we also gain an improvement by a constant factor of 2 over the result in [BR06] for triple cascades; however, in order to fix minor technical problems in the proof in [BR06] (as detailed in [GM09]) a new factor ℓ appears in the security bound.

Although Theorem 4.1 only explicitly states the security of cascades with odd length, we point out that a simple reduction argument proves that longer cascades cannot be less secure than shorter ones, except for a negligible term $\ell/2^k$. Therefore, our result also implicitly proves any even cascade to be at least as secure as a one step shorter odd-length cascade.

In order to make the argument as modular as possible, we start here by addressing the overall structure of the proof, which makes us use two technical lemmas without proof (Lemmas 4.1 and 4.2). These lemmas are then proved in later subsections.

Recall from Section 4.1.2 that $\mathbf{P}$ and $\mathbf{E}$ denote the uniform random permutation and the ideal block cipher, respectively; viewed as 2-interface resources. The following theorem bounds $\Delta_q(\mathsf{Casc}_{\ell,\bar{K}}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E})$, the advantage in distinguishing cascade encryption of length ℓ from a random permutation, given access to the underlying block cipher. The notation $x^{\underline{n}}$ represents the falling factorial power as defined in Section 2.1.

Theorem 4.1 *Let $\ell \geq 3$ be an odd integer. For the construction $\mathsf{Casc}_{\ell,\bar{K}}$ and random systems $\mathbf{E}$, $\mathbf{P}$ defined as above we have*

$$\Delta_q(\mathsf{Casc}_{\ell,\bar{K}}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E}) \leq 2\ell\alpha^{\lfloor\ell/2\rfloor}\frac{q^{\lceil\ell/2\rceil}}{(2^\kappa)^{\underline{\ell}}} + 1.9\left(\frac{\ell q}{2^{\kappa+n/2}}\right)^{2/3} + \frac{\ell^2}{2^{\kappa+1}},$$

where $\alpha = \max\{2e2^{\kappa-n}, 2n + \kappa\lfloor\ell/2\rfloor\}$.

Proof: First, it is easy to see that

$$\Delta_q(\mathsf{Casc}_{\ell,\bar{K}}\mathbf{E}, \mathsf{Casc}_{\ell,\widetilde{K}}\mathbf{E}) \leq p_{coll}(2^\kappa, \ell) < \ell^2/2^{\kappa+1}$$

and hence we have

$$\Delta_q(\mathsf{Casc}_{\ell,\bar{K}}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E}) \leq \Delta_q(\mathsf{Casc}_{\ell,\widetilde{K}}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E}) + \ell^2/2^{\kappa+1}.$$

However, note that the system $\mathsf{Casc}_{\ell,\widetilde{K}}\mathbf{E}$ simply can be seen as providing an interface to query $2^\kappa + 1$ (dependent) permutations

$$\mathbf{E}_{k_1}, \mathbf{E}_{k_2}, \ldots, \mathbf{E}_{k_{2^\kappa}} \text{ and } \mathbf{E}_{K_\ell}\left(\cdots \mathbf{E}_{K_2}\left(\mathbf{E}_{K_1}\left(x\right)\right)\cdots\right) ,$$

each both in forward and backward direction, where $k_1, k_2, \ldots, k_{2^\kappa}$ is an enumeration of the κ-bit strings. By the group structure of $\mathrm{Perm}(n)$ under composition, the joint distribution of these permutations does not change if we start by choosing the last permutation uniformly at random, i.e., we replace it by $\mathbf{P}$, then choose random distinct keys $\widetilde{K}$ and finally choose the permutations of the block cipher independently and randomly except for the one corresponding to the key K_ℓ, which we set to

$$x \mapsto \mathbf{P}\left(\mathbf{E}_{K_1}^{-1}\left(\cdots \mathbf{E}_{K_{\ell-2}}^{-1}\left(\mathbf{E}_{K_{\ell-1}}^{-1}\left(x\right)\right)\right)\right).$$

Hence, let G be a construction that expects access to a single permutation at its inner interface (let us denote it P) and at the outer interface it provides access to a block cipher (let us denote it G). It answers queries to G in the following way: in advance, it chooses random distinct keys $\widetilde{K}$ and then generates random independent permutations for G used with any key except K_ℓ. For K_ℓ, G realizes the permutation

$$x \mapsto P\left(G_{K_1}^{-1}\left(\cdots G_{K_{\ell-2}}^{-1}\left(G_{K_{\ell-1}}^{-1}\left(x\right)\right)\cdots\right)\right),$$

querying P for any necessary values. Then the above argument shows that $\mathsf{Casc}_{\ell,\widetilde{K}}\mathbf{E} \equiv \mathbf{P}\mathsf{G}$ and hence we obtain

$$\Delta_q\left(\mathsf{Casc}_{\ell,\widetilde{K}}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E}\right) = \Delta_q\left(\mathbf{P}\mathsf{G}, \mathbf{P}\lfloor\mathbf{E}\right) \leq \Delta_q\left(\mathbf{Q}\mathsf{G}, \mathbf{Q}\lfloor\mathbf{E}\right),$$

where the last inequality follows from claim (ii) in Lemma 2.2 and $\mathbf{Q}$ denotes the fixed permutation whose existence is guaranteed by this claim. Since $\mathbf{Q}$ is fixed and hence known to the distinguisher, it makes no sense to query it and thus it remains to bound the advantage in distinguishing the right interfaces of $\mathbf{Q}\mathsf{G}$ and $\mathbf{E}$ for any permutation $\mathbf{Q}$. To simplify the notation, from now on we denote the single-interface system providing access to the right interface of $\mathbf{Q}\mathsf{G}$ by $\mathbf{G}$ and see $\mathbf{E}$ as single-interface, too. Note that this concludes the reduction to single-interface resources that we will use in a slightly modified way also in the proof of Theorem 4.4.

We shall refer to all forward or backward queries to $\mathbf{G}$ involving the permutations indexed by any of the keys $K^{(1)}, \ldots, K^{(\ell)}$ as *relevant*. Similarly, the system $\mathbf{E}$ can be seen as also choosing some random distinct keys $\widetilde{K} = (K_1, \ldots, K_\ell)$, this just does not affect its behavior, and we can hence define relevant queries for $\mathbf{E}$ in an analogous way. Let us now add MBOs $\mathcal{A}^h$ and $\mathcal{B}^h$ to the systems $\mathbf{E}$ and $\mathbf{G}$ respectively, obtaining games $\hat{\mathbf{E}}$ and $\hat{\mathbf{G}}$. The MBOs are defined as follows:

- $\mathcal{A}^h_q = 1$ (i.e., the game $\hat{\mathbf{E}}$ is won after q rounds) if the keys $(K_1, K_2, \ldots, K_\ell)$ get examined w.r.t. $\mathbf{Q}$ (in the sense of Definition 4.1) by the first q queries or if more than h of these q queries were relevant.
- $B^h_q = 1$ if either the first q queries formed a chain for the tuple $(K_1, K_2, \ldots, K_\ell)$ or if more than h of these queries were relevant.

The parameter h will be chosen optimally at the end of the proof. Let $\hat{\mathbf{E}}^\perp$ and $\hat{\mathbf{G}}^\perp$ denote the systems $\hat{\mathbf{E}}$ and $\hat{\mathbf{G}}$ blocked by $\mathcal{A}^h$ and $\mathcal{B}^h$, respectively. Then we can apply Lemma 2.4 to obtain

$$\Delta_q(\mathbf{G}, \mathbf{E}) \le \Delta_q\left((\hat{\mathbf{G}}^\perp)^-, (\hat{\mathbf{E}}^\perp)^-\right) + \nu_q(\hat{\mathbf{E}}) \,.$$

Let us first bound the quantity $\nu_q(\hat{\mathbf{E}})$. We can write the bit A^h_q as $U_q \vee V^h_q$, where $U_q = 1$ if the first q queries examined the tuple of keys $(K_1, K_2, \ldots, K_\ell)$ and $V^h_q = 1$ if more than h of the first q queries were relevant. Note that both boolean sequences $\mathcal{U} = U_0, U_1, \ldots$ and $\mathcal{V}^h = V^h_0, V^h_1, \ldots$ are monotone, hence we can "split" the game $\hat{\mathbf{E}}$ into two games $\hat{\mathbf{E}}_1$ and $\hat{\mathbf{E}}_2$ such that they both only differ from $\hat{\mathbf{E}}$ in the MBO part of the output, using $\mathcal{U}$ and $\mathcal{V}^h$ as their MBOs, respectively. Applying the union bound then gives us $\nu_q(\hat{\mathbf{E}}) \le \nu_q(\hat{\mathbf{E}}_1) + \nu_q(\hat{\mathbf{E}}_2)$.

We prove in Lemma 4.1 that $\nu_q(\hat{\mathbf{E}}_1) \le 2\ell\alpha^{\lfloor \ell/2 \rfloor} q^{\lceil \ell/2 \rceil}/(2^\kappa)^{\underline{\ell}}$. For the second part, since the keys $K_1, \ldots, K_\ell$ do not affect the outputs of $\mathbf{E}$, adaptivity does not help when trying to win the game $\hat{\mathbf{E}}_2$, therefore we can restrict our analysis to non-adaptive strategies. The probability that a given query is relevant is $\ell/2^\kappa$, hence the expected number of relevant queries among the first q queries is $\ell q/2^\kappa$ and by Markov inequality we have $\nu_q(\hat{\mathbf{E}}_2) \le \ell q/h2^\kappa$. All put together, we have

$$\nu_q(\hat{\mathbf{E}}) \le 2\ell\alpha^{\lfloor \ell/2 \rfloor} q^{\lceil \ell/2 \rceil}/(2^\kappa)^{\underline{\ell}} + \ell q/h2^\kappa.$$

It remains to bound $\Delta_q\left((\hat{\mathbf{G}}^{\perp})^{-}, (\hat{\mathbf{E}}^{\perp})^{-}\right)$. These systems only differ in their behavior for the first h relevant queries, so let us make this difference explicit. For the definition of the set of cyclic shifts $\mathrm{cs}(\cdot)$ see Section 2.1.

- Let $\mathbf{G}_{\mathrm{r}}$ be a random system that allows queries to ℓ independent random permutations $\pi_1, \pi_2, \ldots, \pi_\ell$, but returns $\perp$ once the queries create an ℓ-chain for any tuple in the set of cyclic shifts $\mathrm{cs}(\pi_1, \pi_2, \ldots, \pi_\ell)$.

- Let $\mathbf{E}_{\mathrm{r}}$ be a random system that allows queries to ℓ random permutations $\pi_1, \pi_2, \ldots, \pi_\ell$ such that $\pi_1 \circ \pi_2 \circ \ldots \circ \pi_\ell = \mathsf{id}$, but returns $\perp$ once the queries create an ℓ-chain for the tuple $(\pi_1, \pi_2, \ldots, \pi_\ell)$.

- Let $\mathsf{C}_{h,\mathbf{Q}}$ be a construction (parametrized by h and the permutation $\mathbf{Q}$ from above) that realizes a block cipher at its outer interface, let us denote it by E. In advance, it picks ℓ random distinct keys $K_1, K_2, \ldots, K_\ell$. Then it realizes the queries to $E_{K_1}, E_{K_2}, \ldots, E_{K_\ell}$ as π_1, $\pi_2, \ldots, \pi_{\ell-1}$ and $\pi_\ell \circ \mathbf{Q}$ respectively, where the permutations π_i for $i \in \{1, \ldots, \ell\}$ are provided by the subsystem attached to its inner interface. E_K for all other keys K are realized by $\mathsf{C}_{h,\mathbf{Q}}$ as random permutations. However, $\mathsf{C}_{h,\mathbf{Q}}$ only redirects the first h relevant queries to the subsystem, after this number is exceeded, it responds to all queries by $\perp$.

Intuitively, the subsystem used is responsible for the answers to the first h relevant queries (hence the subscript "r"). Since the disconnected chains in $\mathsf{C}_{h,\mathbf{Q}}$ correspond exactly to the ordinary chains in $\mathbf{G}_{\mathrm{r}}$, we have $\mathsf{C}_{h,\mathbf{Q}}\mathbf{G}_{\mathrm{r}} \equiv (\hat{\mathbf{G}}^{\perp})^{-}$ and $\mathsf{C}_{h,\mathbf{Q}}\mathbf{E}_{\mathrm{r}} \equiv (\hat{\mathbf{E}}^{\perp})^{-}$. According to claim (i) in Lemma 2.2 and Lemma 4.2 given below, we have

$$\Delta_q\left((\hat{\mathbf{G}}^{\perp})^{-}, (\hat{\mathbf{E}}^{\perp})^{-}\right) \leq \Delta_h(\mathbf{G}_{\mathrm{r}}, \mathbf{E}_{\mathrm{r}}) \leq h^2/2^n.$$

Now we can optimize the choice of the constant h. The part of the advantage that depends on h is $f(h) = \ell q/h2^{\kappa} + h^2/2^n$. This term is minimal for $h^* = (\ell q 2^{n-\kappa-1})^{1/3}$ and we get $f(h^*) < 1.9\left(\frac{\ell q}{2^{\kappa+n/2}}\right)^{2/3}$. This completes the proof. ∎

4.2.3 Examining the Relevant Keys

Here we analyze the probability that the adversary examines the relevant keys $(K_1, \ldots, K_\ell)$ with respect to the permutation $\mathbf{Q}$ during its interaction with the right interface of $\mathbf{Q} \lfloor \mathbf{E}$. This is a generalization of Lemma 7 from [BR06] to longer cascades, also taking disconnected chains into account.

Lemma 4.1 *Let the game $\hat{\mathbf{E}}_1$ be defined as in the proof of Theorem 4.1, with the number of keys ℓ being odd. Then we have $\nu_q(\hat{\mathbf{E}}_1) \le 2\ell\alpha^{\lfloor \ell/2 \rfloor} q^{\lceil \ell/2 \rceil} / (2^\kappa)^{\underline{\ell}}$, where $\alpha = \max\{2e2^{\kappa-n}, 2n + \kappa\lfloor \ell/2 \rfloor\}$.*

Proof: Recall that the relevant keys $K_1, \ldots, K_\ell$ are examined by the distinguisher if it creates either an ℓ-chain or an i-disconnected ℓ-chain for the tuple $(K_1, K_2, \ldots, K_\ell)$ for any $i \in \{1, \ldots, \ell - 1\}$.

Let $i \in \{1, \ldots, \ell - 1\}$ be fixed. We first bound the probability that the distinguisher creates an i-disconnected ℓ-chain. Since the relevant keys do not affect the behavior of the system $\mathbf{E}$, this probability is equal to the number of ℓ-tuples of distinct keys for which an i-disconnected ℓ-chain was created, divided by the number of all ℓ-tuples of distinct keys, which is $(2^\kappa)^{\underline{\ell}}$. The numerator can be upper bounded by the number of all i-disconnected ℓ-chains that were created (here we also count those created for non-distinct key tuples). Hence, let $\mathsf{Ch}^E_{i,\ell,q}$ denote the maximum number of i-disconnected ℓ-chains any distinguisher can create by issuing q queries to a fixed block cipher E and let $\mathsf{Ch}^{\mathbf{E}}_{i,\ell,q}$ denote the expected value of $\mathsf{Ch}^E_{i,\ell,q}$ with respect to the choice of E by $\mathbf{E}$.

Let G be a directed graph corresponding to a block cipher E, as described in Section 4.2.3. Let H be the spanning subgraph of G containing only the edges that were queried by the distinguisher. Any i-disconnected ℓ-chain consists of ℓ edges in H, let us denote them as $e_1, e_2, \ldots, e_\ell$, following the order in which they appear in the chain. Then for each of the odd edges $e_1, e_3, \ldots, e_\ell$ there are q possibilities to choose which of the queries corresponds to this edge. Once the odd edges are fixed, they uniquely determine the vertices $x_0, x_1, \ldots, x_\ell$ such that e_j is $x_{j-1} \to x_j$ for $j \in \{1, 3, \ldots, \ell\} \setminus \{i+1\}$ and e_{i+1} is $Q^{-1}(x_i) \to x_{i+1}$ if i is even. Since there are at most $w(E)$ possible edges to connect any pair of vertices in G, there are now at most $w(E)$ possibilities to choose each of the even edges $e_2, e_4, \ldots, e_{\ell-1}$ so that e_j is $x_{j-1} \to x_j$ for $j \in \{2, 4, \ldots, \ell - 1\} \setminus \{i + 1\}$ and e_{i+1} is $Q^{-1}(x_i) \to x_{i+1}$ if i is odd. Hence, $\mathsf{Ch}^E_{i,\ell,q} \le w(E)^{\lfloor \ell/2 \rfloor} q^{\lceil \ell/2 \rceil}$ and $\mathsf{Ch}^{\mathbf{E}}_{i,\ell,q} \le w(\mathbf{E})^{\lfloor \ell/2 \rfloor} q^{\lceil \ell/2 \rceil}$.

It remains to bound the value $w(\mathbf{E})$. For this, we use the bound from [BR06], where the inequality $\mathsf{P}[w(\mathbf{E}) \geq \beta] < 2^{2n+1-\beta}$ is proved for any $\beta \geq 2e2^{\kappa-n}$. Using this inequality gives us

$$\begin{aligned}\mathsf{Ch}^{\mathbf{E}}_{i,\ell,q} &\leq \mathsf{E}[\mathsf{Ch}^{E}_{i,\ell,q} \mid w(E) < \alpha] + \mathsf{E}[\mathsf{Ch}^{E}_{i,\ell,q} \mid w(E) \geq \alpha] \cdot 2^{2n+1-\alpha} \\ &\leq \alpha^{\lfloor \ell/2 \rfloor} q^{\lceil \ell/2 \rceil} + 2^{\kappa\lfloor \ell/2 \rfloor} q^{\lceil \ell/2 \rceil} 2^{2n+1-\alpha} \leq 2\alpha^{\lfloor l/2 \rfloor} q^{\lceil \ell/2 \rceil},\end{aligned}$$

where the last two inequalities hold since $w(E) \leq 2^{\kappa}$ and $\alpha \geq 2n + \kappa\lfloor \ell/2 \rfloor \geq 2$.

Putting all together, we get that the probability of forming an i-disconnected ℓ-chain for the keys $(K_1, K_2, \ldots, K_\ell)$ can be upper bounded by $2\alpha^{\lfloor \ell/2 \rfloor} q^{\lceil \ell/2 \rceil}/(2^k)^{\underline{\ell}}$. Since this holds for each $i \in \{1, 2, \ldots, \ell - 1\}$ and the probability of creating an ℓ-chain for the keys $(K_1, \ldots, K_\ell)$ can be bounded in the same way, by the union bound we get $\nu_q(\hat{\mathbf{E}}_1) \leq 2\ell\alpha^{\lfloor \ell/2 \rfloor} q^{\lceil \ell/2 \rceil}/(2^{\kappa})^{\underline{\ell}}$. ∎

4.2.4 Recognizing Permutation Dependence without Chains

To conclude the proof of Theorem 4.1 we generalize the bound on $\Delta_h(\mathbf{G}_{\mathrm{r}}, \mathbf{E}_{\mathrm{r}})$ stated by Lemma 9 in [BR06] to apply to the general case of ℓ-cascade encryption. Recall from Theorem 4.1 that $\mathbf{G}_{\mathrm{r}}$ is a random system that provides an interface to query ℓ random independent permutations[8] $\pi_1, \ldots, \pi_\ell$ in both directions. However, if the queries of the distinguisher form an ℓ-chain for any tuple of permutations in $\mathrm{cs}(\pi_1, \ldots, \pi_\ell)$, the system $\mathbf{G}_{\mathrm{r}}$ becomes blocked and answers all subsequent queries (including the one that formed the chain) with the symbol $\perp$. On the other hand, $\mathbf{E}_{\mathrm{r}}$ is a random system that provides an interface to query ℓ random permutations $\pi_1, \ldots, \pi_\ell$ such that $\pi_1 \circ \cdots \circ \pi_\ell = id$, again in both directions. Similarly, if an ℓ-chain is created for any tuple in $\mathrm{cs}(\pi_1, \ldots, \pi_\ell)$ (which is in this case equivalent to creating an ℓ-chain for $(\pi_1, \ldots, \pi_\ell)$), $\mathbf{E}_{\mathrm{r}}$ answers all subsequent queries with the symbol $\perp$. Therefore, the value $\Delta_h(\mathbf{G}_{\mathrm{r}}, \mathbf{E}_{\mathrm{r}})$ denotes the best possible advantage in distinguishing ℓ independent random permutations from ℓ random permutations dependent in the described way, without forming an ℓ-chain.

Lemma 4.2 *Let $\mathbf{G}_{\mathrm{r}}$ and $\mathbf{E}_{\mathrm{r}}$ be the random systems defined in the proof of Theorem 4.1. Then $\Delta_h(\mathbf{G}_{\mathrm{r}}, \mathbf{E}_{\mathrm{r}}) \leq h^2/2^n$.*

[8] All permutations considered here are defined on the set $\{0,1\}^n$.

Proof: First, let us introduce some notation. In any experiment where the permutations $\pi_1, \ldots, \pi_\ell$ are queried, let $\mathrm{dom}_j(\pi_i)$ denote the set of all $x \in \{0,1\}^n$ such that among the first j queries, the query $\pi_i(x)$ was already answered or some query $\pi_i^{-1}(y)$ was answered by x. Similarly, let $\mathrm{range}_j(\pi_i)$ be the set of all $y \in \{0,1\}^n$ such that among the first j queries, the query $\pi_i^{-1}(y)$ was already answered or some query $\pi_i(x)$ was answered by y. In other words, $\mathrm{dom}_j(\pi_i)$ and $\mathrm{range}_j(\pi_i)$ denote the domain and range of the partial function π_i defined by the first j answers. For each pair of consecutive permutations[9] π_i and π_{i+1}, let $\mathcal{X}_i^{(j)}$ denote the set $\{0,1\}^n \setminus (\mathrm{range}_j(\pi_i) \cup \mathrm{dom}_j(\pi_{i+1}))$ of fresh, unused values. If $x \xrightarrow{\pi_i} y$ then we call the queries $\pi_i(x)$ and $\pi_i^{-1}(y)$ *trivial* and the queries $\pi_{i+1}(y)$ and $\pi_{i-1}^{-1}(x)$ are said to *extend* a chain if they are not trivial too.

Now we introduce an intermediate random system $\mathbf{U}$ and show how both $\mathbf{G}_\mathrm{r}$ and $\mathbf{E}_\mathrm{r}$ are conditionally equivalent to $\mathbf{U}$. This allows us to use Lemma 2.3 to bound the advantage in distinguishing $\mathbf{G}_\mathrm{r}$ and $\mathbf{E}_\mathrm{r}$. The system $\mathbf{U}$ also provides an interface to query ℓ permutations $\pi_1, \ldots, \pi_\ell$. It works as follows: it answers any non-trivial forward query $\pi_i(x)$ with a value chosen uniformly from the set $\mathcal{X}_i^{(j-1)}$ and any non-trivial backward query $\pi_i^{-1}(x)$ with a value chosen uniformly from the set $\mathcal{X}_{i-1}^{(j-1)}$ (assuming it is the j^{th} query). Any trivial queries are answered consistently with previous answers. Moreover, if the queries form an ℓ-chain for any tuple in $\mathrm{cs}(\pi_1, \ldots, \pi_\ell)$, $\mathbf{U}$ also gets blocked and responds with $\perp$ to any further queries. Note that $\mathbf{U}$ is only defined as long as $|\mathcal{X}_i^{(j-1)}| \geq 0$, but if this is not true, we have $h \geq 2^n$ and the lemma holds trivially.

Let us now consider the j^{th} query that does not extend an $(\ell-1)$-chain (otherwise both $\mathbf{G}_\mathrm{r}$ and $\mathbf{U}$ get blocked). Then the system $\mathbf{G}_\mathrm{r}$ answers any non-trivial forward query $\pi_i(x)$ by a random element uniformly chosen from $\{0,1\}^n \setminus \mathrm{range}_{j-1}(\pi_i)$ or gets blocked if this answer would create an ℓ-chain by connecting two shorter chains. On the other hand, the system $\mathbf{U}$ answers with a random element uniformly chosen from $\mathcal{X}_i^{(j-1)}$, which is a subset of $\{0,1\}^n \setminus \mathrm{range}_{j-1}(\pi_i)$. The situation for backward queries is analogous. Therefore, let us add an MBO $\mathcal{K}$ to $\mathbf{G}_\mathrm{r}$, obtaining a game $\hat{\mathbf{G}}_\mathrm{r}$: $K_j = 0$ if $K_{j-1} = 0$ and the answer to the j^{th} query was picked from the set $\mathcal{X}_i^{(j-1)}$ if it was a non-trivial forward query $\pi_i(x)$ or from the set $\mathcal{X}_{i-1}^{(j-1)}$ if it was a non-trivial backward query $\pi_i^{-1}(y)$. Note that as long as $\hat{\mathbf{G}}_\mathrm{r}$ is not won, no ℓ-chain can emerge by connecting two shorter chains.

[9] The indexing of permutations is cyclic, e.g. $\pi_{\ell+1}$ denotes the permutation π_1.

By the previous observations and the definition of $\mathcal{K}$, we have $\hat{\mathbf{G}}_{\mathrm{r}}|\mathcal{K} \equiv \mathbf{U}$ which by Lemma 2.3 implies $\Delta_h(\mathbf{G}_{\mathrm{r}}, \mathbf{U}) \leq \nu_h(\hat{\mathbf{G}}_{\mathrm{r}})$. The probability that $\mathcal{K}$ is set to 1 by the j^{th} answer is

$$\frac{|\mathrm{dom}_{j-1}(\pi_{i+1}) \setminus \mathrm{range}_{j-1}(\pi_i)|}{|\{0,1\}^n \setminus \mathrm{range}_{j-1}(\pi_i)|} \leq \frac{|\{0,1\}^n \setminus \mathcal{X}_i^{(j-1)}|}{|\{0,1\}^n|} \leq \frac{j-1}{2^n},$$

which gives us $\nu_h(\hat{\mathbf{G}}_{\mathrm{r}}) \leq \sum_{j=1}^{h}(j-1)/2^n \leq h^2/2^{n+1}$.

In the system $\mathbf{E}_{\mathrm{r}}$, the permutations $\pi_1, \ldots, \pi_\ell$ can be seen as 2^n cycles of length ℓ, each of which is formed by the edges connecting the vertices

$$x, \pi_1(x), \ldots, \pi_{\ell-1}(\cdots\pi_1(x)\cdots), x$$

for some $x \in \{0,1\}^n$ and labeled by the respective permutations. We shall call such a cycle *used* if at least one of its edges was queried in either direction[10], otherwise we call it *unused*. Let us now add an MBO $\mathcal{L}$ to $\mathbf{E}_{\mathrm{r}}$, obtaining a game $\hat{\mathbf{E}}_{\mathrm{r}}$: $L_j = 0$ if and only if during the first j queries, any non-trivial query which did not extend an existing chain queried an unused cycle.

We claim that $\hat{\mathbf{E}}_{\mathrm{r}}|\mathcal{L} \equiv \mathbf{U}$. To see this, let us consider all possible types of queries. If the j^{th} query $\pi_i(x)$ is trivial or it extends an $(\ell-1)$-chain, both systems behave identically. Otherwise, the system $\mathbf{E}_{\mathrm{r}}$ answers with a value y, where $y \notin \mathrm{range}_{j-1}(\pi_i)$ (because π_i is a permutation) and $y \notin \mathrm{dom}_{j-1}(\pi_{i+1})$, since that would mean that $\mathcal{L}$ was set to 1 either earlier (if this query extends an existing chain) or now (if it starts a new chain). All values from $\mathcal{X}_i^{(j-1)}$ have the same probability of being y, because for any $y_1, y_2 \in \mathcal{X}_i^{(j-1)}$, there exists a straightforward bijective mapping between the arrangement of the cycles consistent with $\pi_i(x) = y_1$ or $\pi_i(x) = y_2$ (and all previous answers). Therefore, $\mathbf{E}_{\mathrm{r}}$ answers with an uniformly chosen element from $\mathcal{X}_i^{(j-1)}$ and so does $\mathbf{U}$. For backward queries, the situation is analogous. By Lemma 2.3 this gives us $\Delta_h(\mathbf{E}_{\mathrm{r}}, \mathbf{U}) \leq \nu_h(\hat{\mathbf{E}}_{\mathrm{r}})$.

Let the j^{th} query be a non-trivial forward query $\pi_i(x)$ that does not extend a chain, i.e., $x \in \mathcal{X}_{i-1}^{(j-1)}$. Let u denote the number of elements in $\mathcal{X}_{i-1}^{(j-1)}$ that are in a used cycle on the position between π_{i-1} and π_i. Then since every element in $\mathcal{X}_{i-1}^{(j-1)}$ has the same probability of having

[10] We consider a separate edge connecting two vertices for each cycle in which they follow each other, hence each query creates at most one used cycle.

this property (for the same reason as above), this query wins the game $\hat{\mathbf{E}}_{\mathrm{r}}$ with probability

$$\frac{u}{|\mathcal{X}_{i-1}^{(j-1)}|} \leq \frac{u + |\mathrm{range}_{j-1}(\pi_{i-1}) \cup \mathrm{dom}_{j-1}(\pi_i)|}{2^n} \leq \frac{j-1}{2^n}.$$

Hence $\nu_h(\hat{\mathbf{E}}_{\mathrm{r}}) \leq \sum_{j=1}^{h}(j-1)/2^n \leq h^2/2^{n+1}$.

Putting everything together, we have

$$\Delta_h(\mathbf{G}_{\mathrm{r}}, \mathbf{E}_{\mathrm{r}}) \leq \Delta_h(\mathbf{G}_{\mathrm{r}}, \mathbf{U}) + \Delta_h(\mathbf{U}, \mathbf{E}_{\mathrm{r}}) \leq h^2/2^n$$

which completes the proof. ∎

4.3 A Generic Attack on Plain Cascades

Having a lower bound on the security of the plain cascade, we naturally proceed to approach the question from the opposite direction and explore generic attacks on the cascade construction in our model. In this section we describe such an attack for the general case of ℓ-cascade. It shows that, roughly speaking, plain cascade of length ℓ can be attacked in $2^{\kappa+\frac{\ell-2}{\ell}n}$ queries ($2^{\kappa+\frac{\ell-1}{\ell+1}n}$ queries) for even (odd) ℓ.

Interestingly, for $\ell = 2$ our attack corresponds to the well-known meet-in-the-middle attack against double encryption [DH77] and for $\ell = 3$ it corresponds to one of the attacks given in [Luc98].

Theorem 4.2 *For the cascade construction* Casc_ℓ *of even length* ℓ*, for uniformly random (and independent) keys* $\bar{K} = (K_1, \ldots, K_\ell) \in (\{0,1\}^\kappa)^\ell$ *and for a parameter* $0 < t < 2^{2n/\ell-1}$*, there exists a distinguisher* $\mathbf{D}$ *such that*

$$\Delta^{\mathbf{D}}\left(\mathsf{Casc}_{\ell,\bar{K}}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E}\right) \geq 1 - \frac{2}{t} - 2^{\ell\kappa - t(n-1)}$$

and $\mathbf{D}$ *asks at most* $\ell \cdot 2^{\kappa+\frac{\ell-2}{\ell}n}$ *queries to the right interface and* $2t \cdot 2^{\frac{\ell-2}{\ell}n}$ *forward queries to the left interface. For odd-length cascades* $\mathbf{D}$ *requires at most* $\ell \cdot 2^{\kappa+\frac{\ell-1}{\ell+1}n}$ *queries to the right interface and* $2t \cdot 2^{\frac{\ell-1}{\ell+1}n}$ *forward queries to the left interface.*

Proof: Let us assume that ℓ is even, we give the description of the distinguisher $\mathbf{D}$ interacting with a system $\mathbf{S}$ in Figure 4.2. It first independently chooses random sets $\mathcal{S}_0, \mathcal{S}_2, \ldots, \mathcal{S}_{\ell-2} \subseteq \{0,1\}^n$ of the given sizes and issues $2t \cdot 2^{\frac{\ell-2}{\ell}n}$ queries to the left interface (the cascade or the random permutation) to obtain $\mathcal{S}_\ell := [\mathbf{S}]_L(\mathcal{S}_0)$. Each $\mathcal{S}_i$ will represent the subset of values $\{0,1\}^n$ that $\mathbf{D}$ "cares about" after i steps of the cascade. Then $\mathbf{D}$ issues $\ell \cdot 2^{\kappa+\frac{\ell-2}{\ell}n}$ block cipher queries to the right interface to obtain all the values

$$\mathbf{E}_k(\mathcal{S}_0), \mathbf{E}_k^{-1}(\mathcal{S}_2), \mathbf{E}_k(\mathcal{S}_2), \ldots, \mathbf{E}_k^{-1}(\mathcal{S}_{\ell-2}), \mathbf{E}_k(\mathcal{S}_{\ell-2}), \mathbf{E}_k^{-1}(\mathcal{S}_\ell)$$

with all possible keys $k \in \{0,1\}^\kappa$. These are all the queries $\mathbf{D}$ makes, it remains to justify that they are sufficient to expect that there is a constant number of values $x \in \{0,1\}^n$ that, in case the correct keys are guessed, can be traced through the whole cascade only with the information obtained above. Each such path then allows us to compare its endpoint with $[\mathbf{S}]_L(x)$ which will most probably only match if $\mathbf{S} = \mathsf{Casc}_{\ell.\bar{K}}\mathbf{E}$.

Let us analyze the probability that the set $\mathcal{I}$ is found on line 14 in the setting where $\mathbf{S} = \mathsf{Casc}_{\ell.\bar{K}}\mathbf{E}$ and the inspected key is the correct one, i.e., $\bar{k} = \bar{K}$. Consider the sets

$$\begin{aligned}
\mathcal{P}_0 &= \mathcal{S}_0 \\
\mathcal{P}_2 &= (\mathbf{E}_{k_1}^{-1}(\mathbf{E}_{k_2}^{-1}(\mathcal{S}_2)) \\
&\vdots \\
\mathcal{P}_{2\ell-2} &= \mathbf{E}_{k_1}^{-1}(\cdots \mathbf{E}_{k_{\ell-3}}^{-1}(\mathbf{E}_{k_{\ell-2}}^{-1}(\mathcal{S}_{\ell-2}))\cdots)
\end{aligned}$$

i.e., $\mathcal{P}_{2i}$ is the subset of the plaintext space $\{0,1\}^n$ that gets mapped to $\mathcal{S}_{2i}$ after applying the first $2i$ steps of the cascade with the correct keys. Since the sets $\mathcal{S}_{2i}$ were chosen independently at random, we can invoke Lemma 4.3 (proven separately below) to obtain that for $\mathcal{P} = \bigcap_{i=0}^{\ell/2-1} \mathcal{P}_{2i}$ we have

$$\mathsf{E}(|\mathcal{P}|) = \frac{\prod_{i=0}^{\ell/2-1} |\mathcal{P}_{2i}|}{2^{n(\frac{\ell}{2}-1)}} = \frac{\prod_{i=0}^{\ell/2-1} |\mathcal{S}_{2i}|}{2^{n(\frac{\ell}{2}-1)}} = 2t$$

and similarly $\mathsf{Var}(|\mathcal{P}|) \leq 2t$. Using Chebyshev inequality, this gives us $\mathsf{P}(|\mathcal{P}| < t) \leq 2/t$. If this does not occur (i.e., if $|\mathcal{P}| \geq t$) then the set $\mathcal{P}$ clearly satisfies all requirements imposed on the set $\mathcal{I}$ on lines 14 and 15, hence the desired $\mathcal{I}$ exists and $\mathbf{D}$ will output 1 in this case. Overall, this gives us that $\mathbf{D}(\mathsf{Casc}_{\ell.\bar{K}}\mathbf{E})$ outputs 1 with probability at least $1 - 2/t$.

Distinguisher $\mathbf{D}(\mathbf{S})$: // $\mathbf{S} \in \{\mathsf{Casc}_{\ell.\bar{K}}\mathbf{E}, \mathbf{P}\text{⊥}\mathbf{E}\}$

1: **choose uniformly at random** $\mathcal{S}_0 \subseteq \{0,1\}^n$ **s.t.** $|\mathcal{S}_0| = 2t \cdot 2^{\frac{\ell-2}{\ell}n}$
2: **for** $i := 1$ **to** $\ell/2 - 1$ **do**
3: **choose uniformly at random** $\mathcal{S}_{2i} \subseteq \{0,1\}^n$ **s.t.** $|\mathcal{S}_{2i}| = 2^{\frac{\ell-2}{\ell}n}$
4: **for all** $x \in \mathcal{S}_0$ **do**
5: **query** $y(x) := [\mathbf{S}]_L(x, +)$
6: $\mathcal{S}_\ell := \{y(x) \mid x \in \mathcal{S}_0\}$
7: **for all** $x \in \mathcal{S}_0 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_{\ell-2}$ **do**
8: **for all** $k \in \{0,1\}^\kappa$ **do**
9: **query** $e_k(x) := [\mathbf{S}]_R(k, x, +)$
10: **for all** $y \in \mathcal{S}_2 \cup \mathcal{S}_4 \cup \cdots \cup \mathcal{S}_\ell$ **do**
11: **for all** $k \in \{0,1\}^\kappa$ **do**
12: **query** $e_k^{-1}(y) := [\mathbf{S}]_R(k, y, -)$
13: **for all** $\bar{k} = (k_1, \ldots, k_\ell) \in (\{0,1\}^\kappa)^\ell$ **do**
14: **choose** $\mathcal{I} \subseteq \mathcal{S}_0$ **s.t.** $|\mathcal{I}| = t$ **and** $\forall x \in \mathcal{I}, \forall i \in \{1, \ldots, \ell\}$: $e_{k_i}(\cdots e_{k_1}(x))$ **is known from lines** 9 **and** 12
15: **if** $\mathcal{I}$ exists $\wedge\ \forall x \in \mathcal{I} : y(x) = e_{k_\ell}(\cdots e_{k_1}(x))$ **then**
16: **return** 1
17: **return** 0

Figure 4.2: Distinguisher $\mathbf{D}$ for the proof of Theorem 4.2 for the case of ℓ being even.

On the other hand, if $\mathbf{S} = \mathbf{P}\text{⊥}\mathbf{E}$ then for each $\bar{k}$ the condition on line 15 can only be satisfied with probability at most $2^{-t(n-1)}$, hence by union bound $\mathbf{D}(\mathbf{P}\text{⊥}\mathbf{E})$ outputs 1 with probability at most $2^{\ell\kappa - t(n-1)}$, which concludes the proof for the case of even ℓ.

For odd ℓ we just start by choosing $\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_3, \ldots, \mathcal{S}_{\ell-2} \subseteq \{0,1\}^n$ with $|\mathcal{S}_0| = 2t \cdot 2^{\frac{\ell-1}{\ell+1}n}$ and each of the remaining sets having size $2^{\frac{\ell-1}{\ell+1}n}$. The rest of the attack and its analysis is analogous. ∎

Note that there is a trade-off between the number of construction queries and block cipher queries required for the attack presented in Theorem 4.2. The attack can be generalized to require a lower number $2tm$ of construction queries and $2^{\kappa+n-\frac{2\log m}{\ell-2}}$ block cipher queries.

4.3.1 Estimating Intermediate Set Sizes

The following technical lemma is used in the proof of Theorem 4.2 and we present it separately since it will also turn out to be useful later. It is a generalization of Lemma 6 given in the full version of [GT12].

Let E, Var and Cov denote the usual notions of expected value, variance and covariance, respectively.

Lemma 4.3 *Let $\mathcal{U}$ be a set such that $|\mathcal{U}| = N$ and for $m \in \mathbb{N}$ let $\mathcal{A}_1, \cdots, \mathcal{A}_m$ be sets of size $a_1, \ldots, a_m$ respectively, such that each A_i for $i \geq 2$ is chosen independently uniformly at random from all subsets of $\mathcal{U}$ having a_i elements; A_1 may be chosen arbitrarily. If the random variable X denotes the number of elements of the intersection $\mathcal{A}_1 \cap \cdots \cap \mathcal{A}_m$ then we have $\mathsf{E}(X) = (\prod_{i=1}^{m} a_i)/N^{m-1}$ and $\mathsf{Var}(X) \leq (\prod_{i=1}^{m} a_i)/N^{m-1}$.*

Proof: It is easy to see that X can be expressed as $\sum_{i=1}^{a_1} X_i$ where X_i is the indicator random variable equal to 1 iff $e_i \in \mathcal{A}_2 \cap \cdots \cap \mathcal{A}_m$ (e_i being the i-th element of $\mathcal{A}_1$ in some ordering). For the expected value we clearly have $\mathsf{E}(X_i) = \prod_{i=2}^{m}(a_i/N)$ due to the independent random choice of the sets $\mathcal{A}_2, \ldots, \mathcal{A}_m$. By linearity of expectation this gives us $\mathsf{E}(X) = \sum_{i=1}^{a_1} \mathsf{E}(X_i) = (\prod_{i=1}^{m} a_i)/N^{m-1}$. We then obtain the variance as

$$\mathsf{Var}(X) = \sum_{i=1}^{a_1} \mathsf{Var}(X_i) + 2 \cdot \sum_{1 \leq i < j \leq a_1} \mathsf{Cov}(X_i, X_j)$$

and by bounding the terms in this equation by

$$\begin{aligned} \mathsf{Var}(X_i) &= \mathsf{E}(X_i^2) - (\mathsf{E}(X_i))^2 = \prod_{i=2}^{m} \frac{a_i}{N} - \left(\prod_{i=2}^{m} \frac{a_i}{N}\right)^2 \leq \frac{\prod_{i=2}^{m} a_i}{N^{m-1}} \\ \mathsf{Cov}(X_i, X_j) &= \mathsf{E}(X_i \cdot X_j) - \mathsf{E}(X_i) \cdot \mathsf{E}(X_j) < 0 \end{aligned}$$

we obtain the desired result. ∎

4.4 Security of the XOR-Cascade Construction

We now turn to investigate the so-called XOR-cascades that, loosely speaking, consist of several encryption steps interleaved with key-whitening steps via the XOR operation.

This design paradigm still offers several degrees of freedom: the addition or omission of the key-whitening step at the beginning and at the end; as well as repetition or dependence of keys across the encryption and whitening steps. We resolve the first choice by including the first XOR operation and omitting the last one, see Figure 4.3 and the formal definition below. Keeping exactly one of these two XOR steps is necessary for the presented proof in the case of repeated whitening keys (see below), although we remark that introducing an XOR operation at the end of the construction *instead* of the one at the beginning would result in a symmetric proof also in this case.

In the choice of key-scheduling we consider the variant that derives all keys used in the encryption steps from a single one in a fixed deterministic way such that they are distinct. This is safe thanks to the properties of the ideal cipher model that we are working in that postulates the independence of the permutations realized for each key by the block cipher. In order to weaken this assumption, one could also consider independent keys for each of the encryption steps.

Finally, in the general case we assume the whitening keys to be random and independent, although in the specific case of 2-XOR-cascade we show that the same key might be used for both XOR operations (see Section 4.4.3). A formal definition of the ℓ-XOR-cascade construction follows.

Let us fix a deterministic way to derive ℓ distinct κ-bit keys $(k^{(1)}, \ldots, k^{(\ell)})$ out of a given κ-bit key k in such a way that each mapping $k \mapsto k^{(i)}$ is a bijection. For example, if we assume $\ell \leq \kappa$ then we can simply set $k^{(i)} := k \oplus 0^{i-1}10^{\kappa-i}$, i.e., $k^{(i)}$ will differ from k in the i-th bit. The definition extends naturally to random variables $K^{(1)}, \ldots, K^{(\ell)}$ derived from a uniformly random key K.

In the following discussion, let us model the XOR-Cascade of length ℓ by a (deterministic stateless) construction X_ℓ which on the inner interface expects to access a subsystem $\mathbf{E}$: $\{0,1\}^\kappa \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ realizing a block cipher and on the outer interface it then provides a function $\{0,1\}^\kappa \times (\{0,1\}^n)^{\ell+1} \times \{+,-\} \to \{0,1\}^n$ by answering each forward query $(k, z_1, \ldots, z_\ell, x, +)$ by

$$\mathbf{E}_{k^{(\ell)}}\left(\cdots \mathbf{E}_{k^{(2)}}\left(\mathbf{E}_{k^{(1)}}\left(x \oplus z_1\right) \oplus z_2\right) \cdots \oplus z_\ell\right)$$

and each backward query $(k, z_1, \ldots, z_\ell, y, -)$ by

$$\mathbf{E}^{-1}_{k^{(1)}}\left(\cdots \mathbf{E}^{-1}_{k^{(\ell-1)}}\left(\mathbf{E}^{-1}_{k^{(\ell)}}\left(y\right) \oplus z_\ell\right) \oplus z_{\ell-1} \cdots\right) \oplus z_1.$$

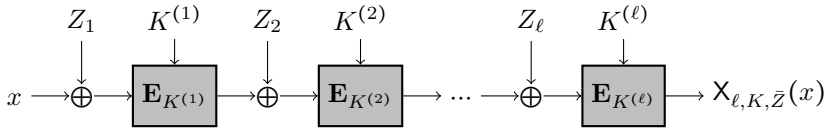

Figure 4.3: Evaluation of the ℓ-XOR-Cascade construction by the left interface of $\mathsf{X}_{\ell,K,\bar{Z}}\mathbf{E}$.

For randomly chosen (secret) keys $(K, \bar{Z}) \in \{0,1\}^\kappa \times (\{0,1\}^n)^\ell$ where $\bar{Z} = (Z_1, \ldots, Z_\ell)$, we let $\mathsf{X}_{\ell,K,\bar{Z}}$ be the construction which gives bidirectional access to the permutation obtained from X_ℓ by fixing the key inputs to $K, \bar{Z}$ (i.e., takes inputs from $\{0,1\}^n \times \{+,-\}$). The evaluation of the construction $\mathsf{X}_{\ell,K,\bar{Z}}$ is depicted in Fig. 4.3.

4.4.1 Key-Alternating Ciphers

Before turning to our results, we introduce the notion of *key-alternating ciphers*. This concept, studied for example in [EM91, BKL+12, Ste12, LPS12], is surprisingly close to our notion of XOR-cascades, however introduced with a very different motivation. It refers to a construction of a block cipher by alternating two types of steps: an XOR of a secret key and an application of a publicly known permutation (see Figure 4.4 and the formal definition below). A prominent example of a block cipher having this structure is the current standard AES [Aes01]. This approach to block-cipher construction is then typically studied in the random permutation model (see Section 2.4) where one assumes that the permutation steps consist of applications of uniformly random and independent, publicly accessible permutations. Below we model the key-alternating ciphers under this assumption. Note that in this setting it is natural to consider constructions that both start and end with the XOR operation.

In the following, let $\bar{\mathbf{P}}_i$ denote a resource providing bidirectional access to i independent uniformly random permutations $\mathbf{P}_1, \ldots, \mathbf{P}_i$, using some fixed addressing mechanism for the queries. As usual throughout this chapter, we see $\bar{\mathbf{P}}_i$ as having two interfaces and providing access to the same set of i permutations via each of them.

Let A_ℓ be a construction which on its inner interface expects to access some subsystem $\hat{\mathbf{P}}_\ell$ giving bidirectional access to ℓ arbitrary permutations denoted $P_1, \ldots, P_\ell$ (e.g., $\bar{\mathbf{P}}_\ell$ is such a subsystem). On the outer

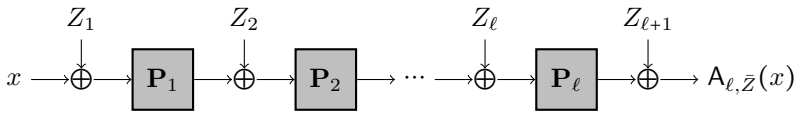

Figure 4.4: Evaluation of the key-alternating construction by the left interface of $\mathsf{A}_{\ell,\bar{Z}}\bar{\mathbf{P}}_\ell$.

interface it then provides a function $(\{0,1\}^n)^{\ell+2} \times \{+,-\} \to \{0,1\}^n$ by answering each forward query $(z_1, \ldots, z_{\ell+1}, x, +)$ by

$$P_\ell\left(\cdots P_2\left(P_1\left(x \oplus z_1\right) \oplus z_2\right) \cdots \oplus z_\ell\right) \oplus z_{\ell+1}$$

and each backward query $(z_1, \ldots, z_{\ell+1}, y, -)$ by

$$P_1^{-1}\left(\cdots P_{\ell-1}^{-1}\left(P_\ell^{-1}\left(y \oplus z_{\ell+1}\right) \oplus z_\ell\right) \oplus z_{\ell-1} \cdots\right) \oplus z_1.$$

Again, for randomly chosen keys $\bar{Z} \in (\{0,1\}^n)^{\ell+1}$ where $\bar{Z} = (Z_1, \ldots, Z_{\ell+1})$, we let $\mathsf{A}_{\ell,\bar{Z}}$ be the construction which gives bidirectional access to the permutation obtained from A_ℓ by fixing the first $\ell + 1$ inputs to $\bar{Z}$ (i.e., it takes inputs from $\{0,1\}^n \times \{+,-\}$). The evaluation of the construction $\mathsf{A}_{\ell,\bar{Z}}$ is depicted in Fig. 4.4.

We now present several bounds recently proved for the security of key-alternating ciphers, recast into our formalism:

Theorem 4.3 *Let $\mathsf{A}_{\ell,\bar{Z}}$ denote the key-alternating cipher of length ℓ as described above.*

1. *[BKL+12] For any $q < 2^n/100$ we have*

$$\Delta_q(\mathsf{A}_{2,\bar{Z}}\bar{\mathbf{P}}_2, \mathbf{P} \lfloor \bar{\mathbf{P}}_2) \le \frac{8.6q^3}{2^{2n}} + \frac{3q^2}{2^{\frac{4}{3}n}}.$$

2. *[Ste12] For any $\ell \ge 1$ and $q < 2^n/100$ we have*

$$\Delta_q(\mathsf{A}_{\ell,\bar{Z}}\bar{\mathbf{P}}_\ell, \mathbf{P} \lfloor \bar{\mathbf{P}}_\ell) \le 3\ell \cdot \frac{q^2}{2^{\frac{3}{2}n}} + (\ell+1) \cdot \frac{q^\ell}{2^{\frac{\ell^2}{\ell+1}n}}.$$

3. *[LPS12] For any even $\ell \ge 1$ we have*

$$\Delta_q(\mathsf{A}_{\ell,\bar{Z}}\bar{\mathbf{P}}_\ell, \mathbf{P} \lfloor \bar{\mathbf{P}}_\ell) \le 2^{\frac{\ell}{4}+3} \cdot \left(\frac{q^{\ell+2}}{2^{\ell n}}\right)^{\frac{1}{4}}.$$

4.4.2 Reduction to the Random Permutation Model

We now reduce our original problem — the security of XOR-cascades in the ideal cipher model — to the problem of the security of one step shorter key-alternating ciphers in the random permutation model. Such a reduction allows us to analyze the problem in a simpler setting without considering the block cipher keys, as well as invoke existing results on key-alternating ciphers. This step follows a pattern similar to the one used in Section 4.2.2 to analyze the security of plain cascades, with some necessary modifications to take the key-whitening steps into account; our description focuses on these differences.

Theorem 4.4 *Let $\ell \geq 2$ and let $(K, \bar{Z}) \in \{0,1\}^\kappa \times (\{0,1\}^n)^\ell$ be uniformly chosen keys. For the constructions $\mathsf{X}_{\ell,K,\bar{Z}}$ and $\mathsf{A}_{\ell-1,\bar{Z}}$ defined as above, and for every distinguisher $\mathbf{D}$ making q queries to the right interface (block cipher queries),*

$$\Delta^{\mathbf{D}}\left(\mathsf{X}_{\ell,K,\bar{Z}}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E}\right) \leq \min_h \left\{ \frac{\ell q}{h 2^\kappa} + \Delta_h\left(\mathsf{A}_{\ell-1,\bar{Z}}\bar{\mathbf{P}}_{\ell-1}, \mathbf{P}\lfloor\bar{\mathbf{P}}_{\ell-1}\right)\right\} .$$

In particular, $\mathbf{D}$ can make arbitrarily many queries to the left interface (construction queries).

Proof: We start by applying a reduction of the distinguished systems to a single interface, similarly to the proof of Theorem 4.1. The system $\mathsf{X}_{\ell,K,\bar{Z}}\mathbf{E}$ can again can be seen as providing an interface to query $2^\kappa + 1$ (dependent) permutations

$$\mathbf{E}_{k_1}, \mathbf{E}_{k_2}, \ldots, \mathbf{E}_{k_{2^\kappa}} \text{ and } \mathbf{E}_{K^{(\ell)}}\left(\cdots \mathbf{E}_{K^{(2)}}\left(\mathbf{E}_{K^{(1)}}\left(x \oplus Z_1\right) \oplus Z_2\right) \cdots \oplus Z_\ell\right)$$

in both directions. Let G denote the construction that expects access to a single permutation (denoted by P) at its inner interface and it provides access to a block cipher (denoted by G) at the outer interface. G chooses in advance random keys $(K, \bar{Z})$ and then it generates random independent permutations to serve as G used with any key except $K^{(\ell)}$. For $K^{(\ell)}$, G realizes the permutation

$$x \mapsto P\left(G^{-1}_{K^{(1)}}\left(\cdots G^{-1}_{K^{(\ell-2)}}\left(G^{-1}_{K^{(\ell-1)}}\left(x \oplus Z_\ell\right) \oplus Z_{\ell-1}\right)\cdots\right) \oplus Z_1\right),$$

querying P for any necessary values. Then again, thanks to the group structure of $\mathrm{Perm}(n)$ under composition and the fact that the XOR operation for a fixed offset is also a permutation, we have $\mathsf{X}_{\ell,K,\bar{Z}}\mathbf{E} \equiv \mathbf{P}\mathsf{G}$ and hence

$$\Delta_q\left(\mathsf{X}_{\ell,K,\bar{Z}}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E}\right) \quad = \quad \Delta_q\left(\mathbf{P}\mathsf{G}, \mathbf{P}\lfloor\mathbf{E}\right) \leq \Delta_q\left(\mathbf{Q}\mathsf{G}, \mathbf{Q}\lfloor\mathbf{E}\right)$$

where the last inequality follows from claim (ii) in Lemma 2.2 and $\mathbf{Q}$ denotes the fixed permutation from this claim. Since now only queries to the right interface make sense for the distinguisher, we shall denote the single-interface systems providing access to these right interfaces by $\mathbf{G}$ and $\mathbf{E}$, respectively.

We can again let the system $\mathbf{E}$ also choose a random key K (and hence also all $K^{(i)}$) and then refer to all queries involving the keys $K^{(1)}, \ldots, K^{(\ell)}$ in both $\mathbf{E}$ and $\mathbf{G}$ as *relevant*. We then extend the systems $\mathbf{E}$ and $\mathbf{G}$ by adding MBOs $\mathcal{A}^h$ and $\mathcal{B}^h$, respectively, and obtaining games $\hat{\mathbf{E}}$ and $\hat{\mathbf{G}}$. This time the MBOs only take care of the number of relevant queries: each of the games $\hat{\mathbf{E}}$ and $\hat{\mathbf{G}}$ is won as soon as more than h of the queries asked so far were relevant (for a parameter h to be determined later).

It is easy to upper-bound the probability of asking more than h relevant queries in $\hat{\mathbf{E}}$: since the choice of the keys $K^{(i)}$ does not affect the responses of the system (and therefore the behavior is also independent of the associated MBO), we only have to consider non-adaptive strategies. Hence, for any distinguisher $\mathbf{D}$ asking q queries, the expected number of relevant queries among them is $\ell q \cdot 2^{-\kappa}$ and using Markov inequality we obtain $\nu_q(\hat{\mathbf{E}}) \leq \ell q/h2^{\kappa}$.

Let $\hat{\mathbf{E}}^{\perp}$ and $\hat{\mathbf{G}}^{\perp}$ denote the systems $\hat{\mathbf{E}}$ and $\hat{\mathbf{G}}$ blocked by $\mathcal{A}^h$ and $\mathcal{B}^h$, respectively. Then we can apply Lemma 2.4 to obtain

$$\Delta_q(\mathbf{G}, \mathbf{E}) \leq \Delta_q\left((\hat{\mathbf{G}}^{\perp})^{-}, (\hat{\mathbf{E}}^{\perp})^{-}\right) + \nu_q(\hat{\mathbf{E}}) \leq \Delta_q\left((\hat{\mathbf{G}}^{\perp})^{-}, (\hat{\mathbf{E}}^{\perp})^{-}\right) + \ell q/h2^{\kappa} \; .$$

As before, the systems $(\hat{\mathbf{G}}^{\perp})^{-}$ and $(\hat{\mathbf{E}}^{\perp})^{-}$ only differ in a small part. Moreover, this time it corresponds to the systems considered in the security definition of key-alternating ciphers in the random permutation model. More specifically, we have $(\hat{\mathbf{G}}^{\perp})^{-} = \mathsf{C}\mathbf{S}$ and $(\hat{\mathbf{E}}^{\perp})^{-} = \mathsf{C}\mathbf{T}$, where:

- $\mathbf{S} := \mathsf{A}_{\ell-1,\bar{Z}}\bar{\mathbf{P}}_{\ell-1}$ is a system that chooses $\bar{Z} \in (\{0,1\}^n)^{\ell}$ at random and provides access (by means of both forward and backward queries) to ℓ randomly chosen permutations $\pi_1, \ldots, \pi_{\ell} \in \mathrm{Perm}(n)$ such that they satisfy the equation

$$\pi_{\ell}^{-1}(\pi_{\ell-1}(\cdots\pi_2(\pi_1(\cdot \oplus Z_1) \oplus Z_2) \oplus Z_3\cdots) \oplus Z_{\ell}) = id;$$

- $\mathbf{T} := \mathbf{P} \perp \bar{\mathbf{P}}_{\ell-1}$ is a system providing access (by means of both forward and backward queries) to ℓ uniformly random permutations $\pi_1, \ldots, \pi_{\ell} \in \mathrm{Perm}(n)$ that are independent;

- C is a (randomized) construction that on its inner interface expects a subsystem which provides ℓ permutations $\pi_1, \ldots, \pi_\ell$. C itself provides access to a block cipher C at the outer interface as follows: it chooses a uniformly random key K and sets $C_{K^{(i)}} := \pi_i$ for all $i \in \{1, \ldots, \ell - 1\}$ and $C_{K^{(\ell)}}(\cdot) := \mathbf{Q}(\pi_\ell^{-1}(\cdot))$. Recall that $\mathbf{Q}$ is s fixed permutation and note that C only queries its subsystem once it is asked a relevant query. The permutations for all other keys are chosen independently at random by C. Moreover, C only allows h relevant queries, after that it returns $\perp$ without issuing any inner queries.

By Lemma 2.2(i), the observation that $(\hat{\mathbf{G}}^{\perp})^{-} = \mathsf{C}\mathbf{S}$ and $(\hat{\mathbf{E}}^{\perp})^{-} = \mathsf{C}\mathbf{T}$ gives us

$$\Delta_q\left((\hat{\mathbf{G}}^{\perp})^{-}, (\hat{\mathbf{E}}^{\perp})^{-}\right) \leq \Delta_h(\mathbf{S}, \mathbf{T}) = \Delta_h\left(\mathsf{A}_{\ell-1,\bar{Z}}\bar{\mathbf{P}}_{\ell-1}, \mathbf{P}\lfloor\bar{\mathbf{P}}_{\ell-1}\right).$$

Since the whole argument holds for any h, we can minimize over it to conclude the proof of the theorem. ∎

4.4.3 The Double XOR-Cascade

As a first application of Theorem 4.4 we invoke it to analyze the security of a variant of 2-XOR-cascade, using the same key Z in both whitening steps.

More precisely, given a (κ, n)-block cipher E, we define the $(\kappa + n, n)$-block cipher $2\mathrm{XOR}^E$ such that

$$2\mathrm{XOR}^E_{k,z}(m) = E_{k^{(2)}}(E_{k^{(1)}}(m \oplus z) \oplus z)$$

for all $k \in \{0,1\}^\kappa$, $z, m \in \{0,1\}^n$. To formally model this construction, let us accept a small modification of the notation given at the beginning of Section 4.4 by writing $\mathsf{X}_{2,K,Z}$ to refer to the construction $\mathsf{X}_{2,K,\bar{Z}}$ where $\bar{Z} = (Z, Z)$, i.e., both whitening keys are the same. The notation $\mathsf{A}_{1,Z}$ will be understood in the analogous way.

We show that even with the above-mentioned savings on key material, 2-XOR-cascade is still secure up to $2^{\kappa+n/2}$ queries in our model. The remaining technical work to be done after applying the reduction from Theorem 4.4 is a security proof for the one-step key-alternating cipher with the same whitening keys. We present the proof as given in [GT12], although it could be also obtained by adapting the arguments in [EM91].

Theorem 4.5 *Let* $\mathbf{P}$ *and* $\mathbf{E}$ *denote a URP on* $\{0,1\}^n$ *and an ideal* (κ, n)*-block cipher, respectively; let* $(K, Z) \in \{0,1\}^\kappa \times \{0,1\}^n$ *be uniformly chosen keys. For the construction* $\mathsf{X}_{2,K,Z}$ *defined as above, and for every distinguisher* $\mathbf{D}$ *making* q *queries to the right interface (block cipher queries),*

$$\Delta^{\mathbf{D}}\left(\mathsf{X}_{2,K,Z}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E}\right) \leq 4 \cdot \left(\frac{q}{2^{\kappa+n/2}}\right)^{2/3} .$$

In particular, $\mathbf{D}$ *can make arbitrarily many queries to the left interface (construction queries).*

Proof: We start by invoking Theorem 4.4 which gives us

$$\Delta^{\mathbf{D}}\left(\mathsf{X}_{2,K,Z}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E}\right) \leq \min_h \left\{ \frac{2q}{h2^\kappa} + \Delta_h\left(\mathsf{A}_{1,Z}\mathbf{P}, \mathbf{P}\lfloor\mathbf{P}'\right) \right\} ,$$

where $\mathbf{P}'$ denotes an independent instance of a URP $\mathbf{P}$ to avoid confusion. It now remains to bound $\Delta_h\left(\mathsf{A}_{1,Z}\mathbf{P}, \mathbf{P}\lfloor\mathbf{P}'\right)$. To simplify the notation, let us denote the distinguished systems as $\mathbf{S}$ and $\mathbf{T}$ respectively; note that:

- $\mathbf{S} := \mathsf{A}_{1,Z}\mathbf{P}$ is a system that chooses $Z \in \{0,1\}^n$ at random and provides access (by means of both forward and backward queries) to two randomly chosen permutations π_1, π_2 on $\{0,1\}^n$ such that they satisfy the equation $\pi_2(\pi_1(\cdot \oplus Z) \oplus Z) = id$;
- $\mathbf{T} := \mathbf{P} \lfloor \mathbf{P}'$ is a system providing access (by means of both forward and backward queries) to two uniformly random permutations $\pi_1, \pi_2 \in \mathrm{Perm}(n)$ that are independent.

We proceed by taking a different view of the internal workings of the system $\mathbf{S}$. Once the values Z, π_1, π_2 are chosen, the internal state of $\mathbf{S}$ can be represented by a set $\mathcal{T}$ of 2^n 4-tuples (x_1, y_1, x_2, y_2) such that $\pi_1(x_1) = y_1$ and $\pi_2(x_2) = y_2$, and $x_2 = y_1 \oplus Z$ and $x_1 = y_2 \oplus Z$. For any $\mathcal{I} \subseteq \{1, \ldots, 4\}$, let $\mathcal{T}_\mathcal{I}$ be the projection of $\mathcal{T}$ on the components in $\mathcal{I}$. Then note that for any two distinct tuples $(x_1, y_1, x_2, y_2), (x_1', y_1', x_2', y_2') \in \mathcal{T}$ we have $x_1 \neq x_1'$, $y_1 \neq y_1'$, $x_2 \neq x_2'$, and $y_2 \neq y_2'$, in other words $\mathcal{T}_{\{i\}} = \{0,1\}^n$ for every $i \in \{1, \ldots, 4\}$.

Equivalently, it is not hard to verify that $\mathbf{S}$ can be implemented using lazy-sampling to set up $\mathcal{T}$: Initially, $\mathcal{T} = \varnothing$ and Z is a uniform n-bit string. Then, $\mathbf{S}$ answers queries as follows:

- Upon a query $\pi_1(x)$, it returns y if $(x, y) \in \mathcal{T}_{\{1,2\}}$ for some y. Otherwise, it returns a random $y \in \{0,1\}^n \setminus \mathcal{T}_{\{2\}}$ and adds $(x, y, y \oplus Z, x \oplus Z)$ to $\mathcal{T}$.

- Upon a query $\pi_1^{-1}(y)$, it returns x if $(x, y) \in \mathcal{T}_{\{1,2\}}$ for some x. Otherwise, it returns a random $x \in \{0,1\}^n \setminus \mathcal{T}_{\{1\}}$ and adds $(x, y, y \oplus Z, x \oplus Z)$ to $\mathcal{T}$.

- Upon a query $\pi_2(x)$, it returns y if $(x, y) \in \mathcal{T}_{\{3,4\}}$ for some y. Otherwise, it returns a random $y \in \{0,1\}^n \setminus \mathcal{T}_{\{4\}}$ and adds $(y \oplus Z, x \oplus Z, x, y)$ to $\mathcal{T}$.

- Upon a query $\pi_2^{-1}(y)$, it returns x if $(x, y) \in \mathcal{T}_{\{3,4\}}$ for some x. Otherwise, it returns a random $x \in \{0,1\}^n \setminus \mathcal{T}_{\{3\}}$ and adds $(y \oplus Z, x \oplus Z, x, y)$ to $\mathcal{T}$.

We consider an intermediate system $\mathbf{S}'$ obtained from $\mathbf{S}$: In addition to $\mathcal{T}$, it also keeps track of sets $\mathcal{P}_1$ and $\mathcal{P}_2$, both consisting of ordered pairs of n-bit strings. (Again $\mathcal{P}_{i,1}$ and $\mathcal{P}_{i,2}$ denote the strings appearing as first and second component in $\mathcal{P}_i$, respectively.) Initially each $\mathcal{P}_i$ is empty and during the experiment, $\mathcal{P}_i$ keeps track of input-output pairs for π_i which were already defined by directly answering a π_i query in either direction (as opposed to those that were defined internally by $\mathbf{S}'$ when answering a π_{3-i} query). Concretely, $\mathbf{S}'$ answers a query $\pi_1(x)$ by y if $(x, y) \in \mathcal{T}_{\{1,2\}} \cup \mathcal{P}_1$ for some y. Otherwise, it returns a uniformly chosen $y \in \{0,1\}^n \setminus \mathcal{P}_{1,2}$ and adds (x, y) to $\mathcal{P}_1$. Moreover, if $y \notin \mathcal{T}_{\{2\}}$, it also adds the tuple $(x, y, y \oplus Z, x \oplus Z)$ to $\mathcal{T}$. Queries $\pi_1^{-1}(y)$, $\pi_2(x)$, and $\pi_2^{-1}(y)$ are answered in a symmetric fashion. Having this description of $\mathbf{S}'$, note that we obtain the system $\mathbf{T}$ if a query $\pi_1(x)$ is answered by some given y only if $(x, y) \in \mathcal{P}_1$, and otherwise a fresh random output is generated (but the 4-tuples are still added to $\mathcal{T}$ as above).

We now extend the system $\mathbf{S}'$ by adding one of two different MBOs $\mathcal{A}$ and $\mathcal{B}$, obtaining games $\mathbf{S}_1$ and $\mathbf{S}_2$, respectively:

- The MBO $\mathcal{A} = A_0, A_1, \ldots$ turns to 1 at the first query $\pi_i(x)$ answered by a random y which satisfies $y \in \mathcal{T}_{\{2(i-1)+2\}}$, or $\pi_i^{-1}(y)$ answered by a random x such that $x \in \mathcal{T}_{\{2(i-1)+1\}}$.

- The MBO $\mathcal{B} = B_0, B_1, \ldots$ turns to 1 at the first query $\pi_i(x)$ such that there exists y which satisfies $(x, y) \in \mathcal{T}_{\{2(i-1)+1,2(i-1)+2\}} \setminus \mathcal{P}_i$, or $\pi_i^{-1}(y)$ such that there exists x satisfying $(x, y) \in \mathcal{T}_{\{2(i-1)+1,2(i-1)+2\}} \setminus \mathcal{P}_i$.

By the above representations of $\mathbf{S}$ and $\mathbf{T}$, one can easily verify that $\mathbf{S}_1|\mathcal{A} \equiv \mathbf{S}$ and $\mathbf{S}_2|\mathcal{B} \equiv \mathbf{T}$. Therefore, by the triangle inequality and by Lemma 2.3,

$$\Delta_h(\mathbf{S}, \mathbf{T}) \leq \Delta_h(\mathbf{S}, \mathbf{S}') + \Delta_h(\mathbf{S}', \mathbf{T}) \leq \nu_h(\mathbf{S}_1) + \nu_h(\mathbf{S}_2).$$

To upper bound $\nu_h(\mathbf{S}_1)$, note that each time a fresh random value is chosen from $\{0,1\}^n \setminus \mathcal{P}_{i,j}$ when answering the i^{th} query, it is in $\mathcal{T}_{2(i-1)+j}$ with probability at most $\frac{i-1}{2^n - i} \leq 2\frac{i-1}{2^n}$, hence the union bound gives us $\nu_h(\mathbf{S}_1) \leq \frac{h^2}{2^n}$.

In order to bound $\nu_h(\mathbf{S}_2)$, let us construct a game $\hat{\mathbf{T}}$ by adding an MBO $\mathcal{C} = C_0, C_1, \ldots$ to $\mathbf{T}$ which turns to 1 under the same circumstances as $\mathcal{B}$ in $\mathbf{S}_1$ (note that this can be done since $\mathbf{T}$ also keeps track of the sets $\mathcal{T}$ and $\mathcal{P}_i$). As a consequence of these equivalent definitions and the fact that the behaviors of $\mathbf{S}'$ and $\mathbf{T}$ are the same as long as the respective games are not won, we have $\nu_h(\mathbf{S}_2) = \nu_h(\hat{\mathbf{T}})$. However, the input-output behavior of $\mathbf{T}$ is independent of Z (and $\mathcal{C}$), and hence we can equivalently postpone the sampling of Z to the end of the interaction, go through the generated transcript to construct $\mathcal{T}$, and upper bound the probability that $\mathcal{C}$ is set to 1 at some query. This implies that for the choice of Z, one query must have been *bad* in the following sense:

- A query $\pi_1(x)$ is preceded by a π_2-query resulting in an input-output pair (x', y') such that $y' \oplus Z = x$;
- A query $\pi_1^{-1}(y)$ is preceded by a π_2-query resulting in an input-output pair (x', y') such that $x' \oplus Z = y$;
- A query $\pi_2(x')$ is preceded by a π_1-query resulting in an input-output pair (x, y) such that $y \oplus Z = x'$;
- A query $\pi_2^{-1}(y')$ is preceded by a π_1-query resulting in an input-output pair (x, y) such that $x \oplus Z = y'$.

Given the transcript, and for randomly chosen Z, the i^{th} query is bad with probability at most $(i-1)/2^n$, and the probability that at least one query is bad is thus at most $\frac{h^2}{2^{n+1}}$ by the union bound.

Putting all the obtained terms together, the part of the distinguisher's advantage that depends on h is $f(h) = q/h2^{\kappa-1} + 3h^2/2^{n+1}$. This term is minimal for $h^* = (\frac{1}{3}q2^{n-\kappa+1})^{1/3}$ which gives us $f(h^*) < 4 \cdot \left(\frac{q}{2^{\kappa+n/2}}\right)^{2/3}$ as desired. ∎

Intuitively, 2XOR requires some mild form of related-key security [BK03] which we obtain for free when the underlying block cipher is ideal, but may be a concern in practice. However, it should be noted that an analogous security statement to the one above could be also proved for the construction 2XOR' that differs only by using two independently chosen keys for the block cipher calls. Hence, 2XOR' achieves the same security level as 2XOR at the cost of a longer $(2\kappa+n)$-bit key, which is for instance still shorter than in DESX with independent whitening keys (for DES parameters).

We stress that the double-randomization in 2XOR is crucial for its security: omitting one of the randomization steps, as well as adding a third randomization step for the same Z, would all result in invalidating the above argument. In particular, Section 4.5.4 provides some extra intuition as for why other simpler randomization methods for double cascade fail to provide the required security level.

4.4.4 Longer XOR-Cascades

We can also combine Theorem 4.4 with the recent results on the security of key-alternating ciphers in the random permutation model cited in Section 4.4.1. This gives us security statements for longer XOR-cascades, as outlined in the following corollary.

Corollary 4.1 *Let $(K,\bar{Z}) \in \{0,1\}^\kappa \times (\{0,1\}^n)^\ell$ be uniformly chosen keys and let $\mathsf{X}_{\ell,K,\bar{Z}}$ denote the ℓ-XOR-cascade construction as defined above. We have:*

1. *[BKL+12] 3-XOR-cascade is secure up to $2^{\kappa+\frac{2}{3}n}$ queries; more precisely, for $n \geq 20$ we have*

$$\Delta_q\left(\mathsf{X}_{3,K,\bar{Z}}\mathbf{E},\mathbf{P}\text{⊥}\mathbf{E}\right) \leq 3\cdot\left(\frac{q}{2^{\kappa+\frac{2}{3}n}}\right)^{\frac{1}{2}} + 9\cdot\left(\frac{q}{2^{\kappa+\frac{2}{3}n}}\right)^{\frac{3}{2}} + 3\cdot\frac{q}{2^{\kappa+\frac{2}{3}n}}\ .$$

2. *[Ste12] ℓ-XOR-cascade is secure up to $2^{\kappa+\frac{3}{4}n}$ queries for $\ell \geq 4$; more precisely, for $n \geq 27$ we have*

$$\Delta_q\left(\mathsf{X}_{\ell,K,\bar{Z}}\mathbf{E},\mathbf{P}\text{⊥}\mathbf{E}\right) \leq \ell\cdot\left(\frac{q}{2^{\kappa+\frac{3}{4}n}}\right)^{\frac{1}{2}} + 9\cdot\frac{q}{2^{\kappa+\frac{3}{4}n}} + 4\cdot\left(\frac{q}{2^{\kappa+\frac{3}{4}n}}\right)^{\frac{3}{2}}\ .$$

3. *[LPS12] ℓ-XOR-cascade is secure up to $2^{\kappa+\frac{\ell-1}{\ell+1}n}$ queries for odd ℓ; more precisely, we have*

$$\Delta_q\left(\mathsf{X}_{\ell,K,\bar{Z}}\mathbf{E}, \mathbf{P}\bot\mathbf{E}\right) \leq (\ell+1)\left(\frac{q}{2^{\kappa+\frac{\ell-1}{\ell+1}n}}\right)^{\frac{1}{2}} + 2^{\frac{\ell+11}{4}}\left(\frac{q}{2^{\kappa+\frac{\ell-1}{\ell+1}n}}\right)^{\frac{\ell+1}{8}} .$$

For even ℓ one can prove the same security as for one step shorter odd-length XOR-cascade.

Proof: [sketch] We simply combine the statement of Theorem 4.4 with the bounds on the security of the key-alternating cipher achieved in the referenced papers, choosing the value h to be $q^{\frac{1}{2}}2^{\frac{n}{3}-\frac{\kappa}{2}}$, $q^{\frac{1}{2}}2^{\frac{3n}{8}-\frac{\kappa}{2}}$ and $q^{\frac{1}{2}}2^{\frac{(\ell-1)n}{2(\ell+1)}-\frac{\kappa}{2}}$ in the three cases above, respectively. The statements for constructions with more rounds follow from the fact that

$$\Delta_h\left(\mathsf{A}_{\ell,\bar{Z}}\bar{\mathbf{P}}_\ell, \mathbf{P}\bot\bar{\mathbf{P}}_\ell\right) \leq \Delta_h\left(\mathsf{A}_{\ell-1,\bar{Z}}\bar{\mathbf{P}}_{\ell-1}, \mathbf{P}\bot\bar{\mathbf{P}}_{\ell-1}\right)$$

which can be shown by a straightforward reduction. ∎

4.5 Generic Attacks on Efficient Key-Length Extension Schemes

After obtaining several security bounds for XOR-cascades, it is natural to ask the following questions:

1. Are these bounds tight?
2. Can better security be achieved by more sophisticated but comparably efficient constructions?
3. Can comparable security levels be achieved by much simpler or more efficient constructions?

These are the questions that motivate our investigation in this section and lead to generic distinguishing attacks against several classes of constructions. As already mentioned, all generic attacks presented in this section can also be mounted in the KPA setting.

To approach the question formally, we consider stateless and deterministic (keyed) constructions C invoking an ideal cipher $\mathbf{E} : \{0,1\}^\kappa \times$

$\{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ to implement a function $\{0,1\}^{\kappa'} \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ to serve as a block cipher with key length κ'. We assume that the construction C when applied to a block cipher realizes a permutation for each $k' \in \{0,1\}^{\kappa'}$ and hence it also provides the interface for inverse queries as indicated. Consequently, for a random (secret) κ'-bit string K', we let $\mathsf{C}_{K'}\mathbf{E}$ denote the system which on its left interface only gives bidirectional access to the permutation obtained by invoking C with the key fixed to K'. (i.e., it takes inputs from $\{0,1\}^n \times \{+,-\}$). Although we allow it, in fact none of the attacks in this section will require backward queries.

4.5.1 One-Query Constructions

Throughout this section, we assume that $\mathsf{C}_{k'}$, to evaluate input $(x,+)$ for $x \in \{0,1\}^n$ under a key $k' \in \{0,1\}^{\kappa'}$, makes exactly one query to the underlying subsystem, and we denote this query as $q(k',x)$. We consider two different cases, depending on the structure of $q(\cdot,\cdot)$, before deriving the final attack.

The Injective Case. We first consider the case where the mapping $x \mapsto q(k',x)$ is injective for each k'. We shall denote this as a *one-injective-query construction*. In this case, distinct outer queries to $\mathsf{C}_{k'}$ lead to distinct inner queries to $\mathbf{E}$ and hence if the distinguisher queries both $\mathsf{C}_{K'}$ (via the left interface) and $\mathbf{E}$ (via the right interface) at sufficiently many random positions, one can expect that during the evaluation of the outer queries, $\mathsf{C}_{K'}$ asks $\mathbf{E}$ for a value that was also asked by the distinguisher. If this occurs, the distinguisher can, while trying all possible keys k', evaluate $\mathsf{C}_{k'}$ on its own by simulating C and using the response from $\mathbf{E}$; and by comparing the outcomes it can distinguish the construction from a truly random permutation. This is the main idea behind Corollary 4.2.

However, to capture the full generality of our argument, we obtain Corollary 4.2 by proving a slightly more general Lemma 4.4. Instead of an ideal block cipher $\mathbf{E}$, we consider an *arbitrary* random function $\mathbf{F}\colon \{0,1\}^d \to \{0,1\}^*$ as the public resource available to the construction (for convenience, we again model it as having two interfaces). The presented proof technique also works in this more general setting, which might be of independent interest.

Note that since the construction C itself is deterministic and stateless, any adversary with access to $\mathbf{F}$ can compute the output of $[\mathsf{C}\mathbf{F}]_L$ on any

input from $\{0,1\}^{\kappa'} \times \{0,1\}^n \times \{+,-\}$ by simulating C on its own and querying $\mathbf{F}$ when necessary.

Lemma 4.4 *Let C be a one-injective-query construction as above, let $\mathbf{P}$ be a URP on $\{0,1\}^n$, and let $K' \in \{0,1\}^{\kappa'}$ be a random key. Then, for all parameters $0 < t < 2^{\min\{n,d/2\}-1}$, there exists a distinguisher $\mathbf{D}$ such that for all random functions $\mathbf{F} : \{0,1\}^d \to \{0,1\}^*$,*

$$\Delta^{\mathbf{D}}(\mathsf{C}_{K'}\mathbf{F}, \mathbf{P}\lfloor\mathbf{F}) \geq 1 - 2/t - 2^{\kappa'-t\cdot(n-1)} ,$$

where $\mathbf{D}$ makes $2t \cdot \max\left\{2^{d/2}, 2^{d-n}\right\}$ queries to the right interface as well as $\min\left\{2^{d/2}, 2^n\right\}$ forward queries to the left interface.

Proof: We start by fixing the parameter t from the given interval. The distinguisher $\mathbf{D}$ is described in Figure 4.5, where

$$q_{\mathbf{S}} := \min\left\{2^{d/2}, 2^n\right\} \quad \text{and} \quad q_{\mathbf{F}} := 2t \cdot \max\left\{2^{d/2}, 2^{d-n}\right\} ,$$

$x_1, \ldots, x_{q_{\mathbf{S}}} \in \{0,1\}^n$ are arbitrarily chosen distinct fixed strings and $\mathcal{R}$ is a randomly chosen subset of $\{0,1\}^d$ of size $q_{\mathbf{F}}$, as indicated in Figure 4.5.

For the analysis, assume that $\mathbf{S} = \mathsf{C}_{K'}\mathbf{F}$ and the key K' has been chosen to take the value k'. Then, the expected size of the set $\mathcal{S} = \mathcal{R} \cap \{q(k', x_1), \ldots, q(k', x_{q_{\mathbf{S}}})\}$ is at least $2t$, with variance upper-bounded also by $2t$ by Lemma 4.3 in Section 4.3.1. Therefore, by Chebyshev's inequality, the probability that $|\mathcal{S}| < t$ is upper-bounded by $2/t$. If $|\mathcal{S}| \geq t$ then the set $\mathcal{I}$ is determined on Line 8 and the condition on Line 12 is satisfied and $\mathbf{D}$ outputs 1. (Note that this condition is verified by simulating the construction $\mathsf{C}_{k'}$ and answering its inner queries using the previously obtained $\mathbf{F}$-values.) On the other hand, if $\mathbf{S} = \mathbf{P}\lfloor\mathbf{F}$, then for every fixed value k' the condition on Line 12 is only satisfied with probability $(2^n - t)!/2^n! \leq 2^{-t\cdot(n-1)}$. Hence, by the union bound, the probability that this condition is satisfied for *any* key k' is upper-bounded by $2^{\kappa'-t(n-1)}$. Putting all of this together, we see that $\mathbf{D}(\mathsf{C}_{K'}\mathbf{F})$ outputs 1 with probability at least $1 - 2/t$, whereas $\mathbf{D}(\mathbf{P}\lfloor\mathbf{F})$ outputs 1 with probability at most $2^{\kappa'-t(n-1)}$, which yields the statement of the lemma. ∎

As our primary interest lies in block-cipher based constructions operating on the same domain as the underlying cipher, the following corollary restates Lemma 4.4 for this setting. Recall that the ideal block cipher $\mathbf{E} : \{0,1\}^\kappa \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ can be seen as a random function with input length $\kappa + n + 1$ and output length n.

Distinguisher $\mathbf{D}(\mathbf{S})$: // $\mathbf{S} \in \{\mathsf{C}_{K'}\mathbf{F}, \mathbf{P}\lfloor\mathbf{F}\}$

choose arbitrary distinct $x_1, \ldots, x_{q_\mathbf{S}} \in \{0,1\}^n$
for $i := 1$ **to** $q_\mathbf{S}$ **do**
  query $y_i := [\mathbf{S}]_L(x_i, +)$
choose uniformly at random $\mathcal{R} \subseteq \{0,1\}^d$ **s.t.** $|\mathcal{R}| = q_\mathbf{F}$
for all $x \in \mathcal{R}$ **do**
  query $f(x) := [\mathbf{S}]_R(x)$
for all $k' \in \{0,1\}^{\kappa'}$ **do**
  choose $\mathcal{I} \subseteq \{1, \ldots, q_\mathbf{S}\}$ **s.t.** $|\{q(k', x_i) : i \in \mathcal{I}\}| = t$ **and** $\forall i \in \mathcal{I} : q(k', x_i) \in \mathcal{R}$
  if $\mathcal{I}$ exists **then**
    for all $i \in \mathcal{I}$ **do**
      simulate C **to evaluate** $z_i := \mathsf{C}(k', x_i, +)$ **answering inner queries by** $f(\cdot)$
    if $\forall i \in \mathcal{I} : y_i = z_i$ **then**
      return 1
return 0

Figure 4.5: Distinguisher $\mathbf{D}$ for the proofs of Lemma 4.4 and Theorem 4.6.

Corollary 4.2 *Let* $\mathbf{E}: \{0,1\}^\kappa \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ *be an ideal block cipher, let* $\mathsf{C}: \{0,1\}^{\kappa'} \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ *be a one-injective-query construction* [11] *and let* $\mathbf{P}$ *be a URP on* $\{0,1\}^n$*. Then, for a random key* $K' \in \{0,1\}^{\kappa'}$ *and every parameter* $0 < t < 2^{\min\{n,k\}-1}$,[12] *there exists a distinguisher* $\mathbf{D}$ *such that*

$$\Delta^{\mathbf{D}}(\mathsf{C}_{K'}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E}) \geq 1 - 2/t - 2^{\kappa'-t\cdot(n-1)} ,$$

and which makes at most $4t \cdot 2^{\max\{(\kappa+n)/2,\kappa\}}$ *queries to the right interface as well as at most* $2 \cdot 2^{\min\{(\kappa+n)/2,n\}}$ *forward queries to the left interface.*

The above statement covers most of the natural one-query constructions, since these typically satisfy the injectivity requirement (e.g. the

[11] Note that here $q(k', x)$ denotes the complete input consisting of a key k, a block m, and an indication of whether this is a forward or a backward query, totalling hence $\kappa + n + 1$ input bits.

[12] Roughly speaking, higher t increases the advantage but also the required number of queries; we obtain the desired bound using a constant t. For a first impression, consider e.g. $t = 4$ and $\kappa' \approx 2n$.

DESX construction). In the following we see that constructions asking non-injective queries do not achieve any improvement in security.

Non-Injective Queries. We now permit that the construction C might, for some key k', invoke the underlying ideal cipher in a *non-injective* way, i.e., $q(k', \cdot)$ is not an injective map. We prove that, roughly speaking, the permutation obtained by such a construction $\mathsf{C}_{K'}\mathbf{E}$ might be distinguishable from a URP $\mathbf{P}$ based solely on an entropy argument. The intuitive reasoning is that if C allows on average (over the choice of the key k') that too many queries x map to the same $q(k', x)$, then it also does not manage to obtain sufficient amount of randomness from the underlying random function to simulate $\mathbf{P}$ convincingly, opening the door to a distinguishing attack. In the following, let $q(k') = |\{q(k', x) : x \in \{0,1\}^n\}|$ for all $k' \in \{0,1\}^\kappa$.

Lemma 4.5 is again given in the more general setting with the underlying random function $\mathbf{F}$, which we here require to have a finite range $\{0,1\}^r$.

Lemma 4.5 *Let* $\mathsf{C}\colon \{0,1\}^{\kappa'} \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ *be a one-query construction, let* $\mathbf{P}$ *be a URP on* $\{0,1\}^n$ *and let* $\mathbf{F}\colon \{0,1\}^d \to \{0,1\}^r$ *be a random function. Also, let* $K' \in \{0,1\}^{\kappa'}$ *be a random key, and assume that there exists* q^* *such that* $q(K') \le q^*$ *with probability at least* $\frac{1}{2}$*. Then, there exists a distinguisher* $\mathbf{D}$ *asking* 2^n *forward queries to the left interface only, such that*

$$\Delta^{\mathbf{D}}\left(\mathsf{C}_{K'}\mathbf{F}, \mathbf{P}\lfloor\mathbf{F}\right) \ge \tfrac{1}{2} - 2^{\kappa'+r\cdot q^* - \log(2^n!)} \ .$$

Proof: Let $\mathcal{K}^*$ be the set of keys k' for which $q(k') \le q^*$. By our assumption, $\mathsf{P}[K' \in \mathcal{K}^*] \ge \frac{1}{2}$. Also, let $\mathcal{M}$ be the set of mappings which can be implemented by the left interface of $\mathsf{C}_{K'}\mathbf{F}$ given $K' \in \mathcal{K}^*$. We have $|\mathcal{M}| \le |\mathcal{K}^*| \cdot 2^{r\cdot q^*} \le 2^{\kappa'+r\cdot q^*}$ since given $K' \in \mathcal{K}^*$, $\mathsf{C}_{K'}$ obtains at most $r \cdot q^*$ output bits from $\mathbf{F}$. Note that $\mathcal{M}$ only depends on C and the description of $\mathbf{F}$, and can hence be computed by $\mathbf{D}$.

The distinguisher $\mathbf{D}$ queries the left interface for all values in $\{0,1\}^n \times \{+\}$ and returns 1 if and only if the mapping obtained is in $\mathcal{M}$, and 0 otherwise. Obviously, $\mathbf{D}$ outputs 1 with probability at least $\frac{1}{2}$ when interacting with $\mathsf{C}_{K'}\mathbf{F}$. However, if $\mathbf{D}$ interacts with $\mathbf{P}\lfloor\mathbf{F}$, the probability that it observes a mapping from $\mathcal{M}$ is upper bounded by $2^{\kappa'+r\cdot q^*} \cdot 2^{-\log(2^n!)}$, which concludes the proof. ∎

We can again consider the ideal block cipher $\mathbf{E}$ as a special case of the random function $\mathbf{F}$ in the setting above, which then gives us a distinguisher $\mathbf{D}$ such that $\Delta^{\mathbf{D}}\left(\mathsf{C}_{K'}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E}\right) \ge \frac{1}{2} - 2^{\kappa'+n\cdot q^* - \log(2^n!)}$.

Putting the Pieces Together. We can combine the techniques used to prove Lemma 4.4 (somewhat relaxing the injectivity requirement) and Lemma 4.5 to obtain the following final theorem yielding an attack for arbitrary one-query block-cipher constructions.

Theorem 4.6 *Let $n \geq 6$ and $\kappa' \leq 2^n - 1$, let $\mathbf{E}\colon \{0,1\}^\kappa \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ be an ideal block cipher, let $\mathsf{C}\colon \{0,1\}^{\kappa'} \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ be a one-query construction, and let $\mathbf{P}$ be a URP on $\{0,1\}^n$. Then, for a random key $K' \in \{0,1\}^{\kappa'}$ and for all parameters $0 < t < 2^{n-2}$, there exists a distinguisher $\mathbf{D}$ such that*

$$\Delta^{\mathbf{D}}\left(\mathsf{C}_{K'}\mathbf{E}, \mathbf{P}\mathord{\perp}\mathbf{E}\right) \geq \min\left\{\frac{1}{4}, \frac{1}{2} - \frac{2}{t} - 2^{\kappa'-t\cdot(n-1)}\right\}$$

which asks at most $8t \cdot 2^\kappa$ queries to the right interface and 2^n forward queries to the left interface.

Proof: Throughout the proof, fix $q^* = \left((n-2)\cdot 2^n - \kappa' - 1\right)/n$. Assume first that $q(K') \leq q^*$ with probability at least $\frac{1}{2}$. Then we can use Lemma 4.5 to obtain a distinguisher $\mathbf{D}$ asking 2^n queries such that

$$\Delta^{\mathbf{D}}\left(\mathsf{C}_{K'}\mathbf{E}, \mathbf{P}\mathord{\perp}\mathbf{E}\right) \geq \tfrac{1}{2} - 2^{\kappa'+n\cdot q^* - \log(2^n!)} \geq \tfrac{1}{2} - 2^{(n-2)\cdot 2^n - \log(2^n!) - 1}\,.$$

Applying the bound $\ln x! \geq x\ln(x/e) + 1$ we get $\log(2^n!) \geq (n-2)\cdot 2^n + 1$ and therefore $\Delta^{\mathbf{D}}\left(\mathsf{C}_{K'}\mathbf{E}, \mathbf{P}\mathord{\perp}\mathbf{E}\right) \geq 1/4$.

Let us address the complementary case that $q(K') > q^*$ with probability at least $\frac{1}{2}$. Since we assume $\kappa' \leq 2^n - 1$ and $n \geq 6$, we note that $q^* \geq 2^n \cdot (n-3)/n \geq 2^{n-1}$. Hence, roughly speaking, for at least half of the keys k' the mapping $q(k', \cdot)$ is "almost injective" and we can use the same distinguisher as in the proof of Lemma 4.4 with different parameters.

More precisely, we use the distinguisher $\mathbf{D}$ depicted in Figure 4.5 with $q_{\mathbf{S}} := 2^n$ (we query the entire domain) and $q_{\mathbf{F}} := 8t\cdot 2^\kappa$ (recall that the ideal cipher $\mathbf{E}$ has domain size $\kappa + n + 1$). If k' is chosen such that $q(k') > 2^{n-1}$ (which is true with probability at least $\frac{1}{2}$), then the expected value and the variance of the cardinality of $\mathcal{S} := \{q(k', x) \mid x \in \{0,1\}^n\} \cap \mathcal{R}$ are at least $2t$ (again by Lemma 4.3) and hence Chebyshev's inequality yields that $|\mathcal{S}| \geq t$ except with probability at most $2/t$. Therefore $\mathbf{D}(\mathsf{C}_{K'}\mathbf{E})$ outputs 1 with probability at least $\frac{1}{2} - 2/t$. However, as in the proof of Lemma 4.4, $\mathbf{D}(\mathbf{P}\mathord{\perp}\mathbf{E})$ outputs 1 with probability at most $2^{\kappa'-t\cdot(n-1)}$. This concludes the proof. ∎

Theorem 4.6 shows that no one-query construction can achieve security beyond $2^{\max\{\kappa,n\}}$ queries, hence in the search for efficient key-length

extension schemes one has to consider constructions issuing at least two queries.

4.5.2 Injective Two-Query Constructions

We now consider deterministic stateless constructions $\mathsf{C} : \{0,1\}^{\kappa'} \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ that make exactly two queries to the block cipher connected to their inner interface to evaluate each outer query. In the following, these constructions shall be referred to as *two-query constructions*. We denote by $q_1(k',x) \in \{0,1\}^{\kappa} \times \{0,1\}^n \times \{+,-\}$ the first query C asks its subsystem when it is itself being asked a forward query $(k',x,+)$. Moreover, we denote by $q_2(k',x,s) \in \{0,1\}^{\kappa} \times \{0,1\}^n \times \{+,-\}$ the second query it asks when it is itself being asked a forward query $(k',x,+)$ and the answer to the first query $q_1(k',x)$ was $s \in \{0,1\}^n$. Since C is deterministic and stateless, both q_1 and q_2 are well-defined mappings.

Theorem 4.7 *Let* $\mathsf{C}\colon \{0,1\}^{\kappa'} \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ *be a two-query construction satisfying the following two conditions:*

1. *for every* $k' \in \{0,1\}^{\kappa'}$ *the mapping* $q_1(k',\cdot)$ *is injective,*
2. *distinct answers to the first query imply distinct second queries, i.e., for every* $k' \in \{0,1\}^{\kappa'}$ *and every* $x,x' \in \{0,1\}^n$ *if* $s \neq s'$ *then* $q_2(k',x,s) \neq q_2(k',x',s')$.

Then for a random key $K' \in \{0,1\}^{\kappa'}$*, for a URP* $\mathbf{P}$ *on* $\{0,1\}^n$ *and for every parameter* $0 < t < 2^{n/2-1}$*, there exists a distinguisher* $\mathbf{D}$ *such that*

$$\Delta^{\mathbf{D}}(\mathsf{C}_{K'}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E}) \geq 1 - 2/t - 13 \cdot 2^{-\frac{n}{2}} - 2^{\kappa'-t\cdot(n-1)},$$

where $\mathbf{D}$ *makes at most* $2(t+4)\cdot 2^{\kappa+n/2}$ *block cipher queries as well as* 2^n *forward construction queries.*

Proof: The distinguisher $\mathbf{D}$ is described in Figure 4.6, with the parameters $q_{\mathbf{E},1} := 8 \cdot 2^{\kappa+n/2}$ and $q_{\mathbf{E},2} := 2t \cdot 2^{\kappa+n/2}$.

Let us analyze the interaction of this distinguisher with $\mathsf{C}_{K'}\mathbf{E}$. For every fixed key k', the size of the set $\{q_1(k',x) \mid x \in \{0,1\}^n\}$ is 2^n due to the injectivity of the mapping $q_1(k',\cdot)$. Hence the expected size of the

Distinguisher $\mathbf{D}(\mathbf{S})$: // $\mathbf{S} \in \{\mathsf{C}_{K'}\mathbf{E}, \mathbf{P}\lfloor\mathbf{E}\}$

for all $x \in \{0,1\}^n$ **do**
  query $y(x) := [\mathbf{S}]_L(x, +)$
choose uniformly at random $\mathcal{R} \subseteq \{0,1\}^\kappa \times \{0,1\}^n \times \{+,-\}$ **such that** $|\mathcal{R}| = q_{\mathbf{E},1}$
choose uniformly at random $\mathcal{U} \subseteq \{0,1\}^\kappa \times \{0,1\}^n \times \{+,-\}$ **such that** $|\mathcal{U}| = q_{\mathbf{E},2}$
for all $r \in \mathcal{R} \cup \mathcal{U}$ **do**
  query $e(r) := [\mathbf{S}]_R(r)$
for all $k' \in \{0,1\}^{\kappa'}$ **do**
  choose $\mathcal{I} \subseteq \{0,1\}^n$ **s.t.** $|\mathcal{I}| = t$ **and** $\forall x \in \mathcal{I}$:
    - $q_1(k', x) \in \mathcal{R}$
    - $q_2(k', x, e(q_1(k', x))) \in \mathcal{U}$
  if $\mathcal{I}$ exists **then**
    for all $x \in \mathcal{I}$ **do**
      simulate C **to evaluate** $z(x) := \mathsf{C}(k', x, +)$ **answering inner queries by** $e(\cdot)$
    if $\forall x \in \mathcal{I} : y(x) = z(x)$ **then**
      return 1
return 0

Figure 4.6: Distinguisher $\mathbf{D}$ for the proof of Theorem 4.7.

set $\mathcal{S} := \{q_1(k', x) \mid x \in \{0,1\}^n\} \cap \mathcal{R}$ is $4 \cdot 2^{n/2}$ with the variance upper-bounded by $4 \cdot 2^{n/2}$ by Lemma 4.3. Chebyshev's inequality then yields $\mathsf{P}[|\mathcal{S}| < 3 \cdot 2^{n/2}] \leq 2^{2-n/2}$.

Let us assume that $|\mathcal{S}| \geq 3 \cdot 2^{n/2}$: We need to show that the image $\mathbf{E}(\mathcal{S})$ of $\mathcal{S}$ under $\mathbf{E}$ is also sufficiently large. Indeed, note that the set $\mathcal{S}$ is independent of $\mathbf{E}$: If an ideal block cipher is queried at $3 \cdot 2^{n/2}$ arbitrary distinct triples $(k, x, \sigma) \in \{0,1\}^\kappa \times \{0,1\}^n \times \{-,+\}$, then for any two such triples (k, x, σ) and (k', x', σ'), the probability that the query outputs collide under $\mathbf{E}$ is at most $\frac{2}{2^n}$. (The case with the highest probability is when $k = k'$, $x \neq x'$, $\sigma = -$, and $\sigma' = +$.) The expected number of collisions is upper-bounded by $\frac{9 \cdot 2^n}{2} \cdot \frac{2}{2^n} = 9$. Therefore, by Markov's inequality no more than $2^{n/2}$ collisions (among the values $\mathbf{E}(s)$ for $s \in \mathcal{S}$) occur, except with probability at most $9 \cdot 2^{-n/2}$, and if at most $2^{n/2}$ collisions occur,

then $|\mathbf{E}(\mathcal{S})| \geq 3 \cdot 2^{n/2} - 2^{n/2} = 2^{n/2+1}$. Due to the assumed property of q_2, this also means that the set $\mathcal{T} := \{q_2(k', x, e(q_1(k', x))) : q_1(k', x) \in \mathcal{S}\}$ has cardinality $|\mathcal{T}| \geq 2^{n/2+1}$.

Similar to above, the estimated size of the set $\mathcal{V} := \mathcal{T} \cap \mathcal{U}$ is at least $2^{n/2+1} \cdot 2t \cdot 2^{\kappa+n/2} / 2^{\kappa+n+1} = 2t$ with variance at most $2t$ (again by Lemma 4.3) and hence the probability that $|\mathcal{V}| < t$ can again be upper-bounded by $2/t$ using the Chebyshev inequality.

Note that if $|\mathcal{V}| \geq t$ then a set $\mathcal{I}$ will be found on Line 8 when going through the right chosen key k', which implies that the test on Line 12 succeeds, and we are hence guaranteed that $\mathbf{D}$ outputs 1, unless one of the previously mentioned bad events happens, for which the probability is bounded by $13 \cdot 2^{-n/2} + 2/t$. On the other hand, if $\mathbf{D}$ interacts with $\mathbf{P} \lfloor \mathbf{E}$, the probability that this test is satisfied for any of the keys $k' \in \{0,1\}^{\kappa'}$ can be upper-bounded by $2^{\kappa' - t \cdot (n-1)}$ as in the proof of Lemma 4.4. ∎

Hence, no two-query construction from the large class described in the above theorem (which we loosely call "injective constructions") can achieve security beyond $2^{\kappa+n/2}$ queries. This together with the claim of Theorem 4.5 implies that 2-XOR-cascade is optimally secure within this class.

4.5.3 Sequential Constructions

An attempt to generalize the attack from Section 4.5.2 to constructions issuing more block cipher queries per invocation runs into a technical problem. Roughly speaking, when analyzing the number of plaintexts that we can follow through multiple applications of the block cipher, the positions where the block cipher is queried are (after the first round) no longer independent of its internal randomness. Hence this approach cannot be generalized to the case of constructions issuing more block cipher queries if we are only assuming a property analogous to the injectivity considered in Section 4.5.2.

Thus, when extending our perspective to constructions issuing an arbitrary number ℓ of queries, we at the same time restrict ourselves by the following natural assumption: we only consider constructions with *input-independent key scheduling*. Informally, this means that the keys used for each block cipher call (and its direction) is determined solely by the key K' provided to the construction and does not depend on the plaintext

x to be encrypted. It is easy to see that input-independent key scheduling, together with the injective property considered in Section 4.5.2, together imply that the construction must have the particular form which we call *sequential* and formally describe below.

Definition 4.2 *We call a construction* $\mathsf{C}\colon \{0,1\}^{\kappa'} \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ sequential *if, given an underlying block cipher* $\mathbf{E}$ *and with the first part of its input fixed to a key* $K' \in \{0,1\}^{\kappa'}$*, it provides bidirectional access to the permutation*

$$x \mapsto P_\ell\left(\mathbf{E}_{K_\ell}\left(P_{\ell-1}\left(\cdots \mathbf{E}_{K_2}\left(P_1\left(\mathbf{E}_{K_1}\left(P_0(x)\right)\right)\right)\cdots\right)\right)\right)$$

such that the keys K_i *and the permutations* P_i *are derived from the overall key* K' *in a deterministic way.*

Note that the ℓ-XOR-cascade construction is an example of a sequential construction. For sequential ℓ-query constructions, the attack from Section 4.5.2 can be generalized without any difficulties, hence we only present the resulting statement. Note that this attack can also be seen as a lifting of an attack presented in [BKL+12] into the ideal block cipher setting.

Theorem 4.8 *Let* $\mathsf{C}\colon \{0,1\}^{\kappa'} \times \{0,1\}^n \times \{+,-\} \to \{0,1\}^n$ *be a sequential* ℓ*-query construction. Then for a random key* $K' \in \{0,1\}^{\kappa'}$*, for a URP* $\mathbf{P}$ *on* $\{0,1\}^n$ *and for every parameter* $0 < t < 2^{n/\ell-2}$*, there exists a distinguisher* $\mathbf{D}$ *such that*

$$\Delta^{\mathbf{D}}(\mathsf{C}_{K'}\mathbf{E}, \mathbf{P} \lfloor \mathbf{E}) \geq 1 - 2/t - 3 \cdot 2^{-\frac{n}{\ell}} - 2^{\kappa'-t(n-1)},$$

where $\mathbf{D}$ *makes at most* $2(\ell+t) \cdot 2^{\kappa+\frac{\ell-1}{\ell}n}$ *block cipher queries as well as* 2^n *forward construction queries.*

Again, a trade-off between the number of construction queries and block cipher queries is possible: an analogous attack can be mounted with a lower number m of construction queries and at most $2(\ell+t) \cdot 2^{\kappa+n-\frac{\log m}{\ell}}$ block cipher queries. Also here the construction queries can be arbitrary, obtaining a known-plaintext attack.

4.5.4 Various Randomized Double Cascades

To justify the design of the construction 2XOR presented in Section 4.4.3, we conclude the section on generic attacks by briefly discussing some related constructions and their security: Namely, we observe that for various simpler key-dependent randomizations of the double cascade, meet-in-the-middle attacks issuing at most $2^{\max\{\kappa,n\}}$ queries can be mounted.

Single Randomization. We start with constructions with one single randomization step. Consider the following mappings C_1, C_2, and C_3 parametrized by the underlying block cipher E:

$$C_{1,k_1,k_2,z}(m) = E_{k_2}(E_{k_1}(m \oplus z))$$
$$C_{2,k_1,k_2,z}(m) = E_{k_2}(E_{k_1}(m) \oplus z)$$
$$C_{3,k_1,k_2,z}(m) = E_{k_2}(E_{k_1}(m)) \oplus z$$

and the respective constructions $\mathsf{C}_{i,K_1,K_2,Z}$ providing bidirectional access to the permutation $C_{i,K_1,K_2,Z}$ at the outer interface (for random keys K_1, K_2, Z), while accessing the underlying block cipher at the inner one.

We distinguish $\mathsf{C}_{2,K_1,K_2,Z}\mathbf{E}$ and $\mathbf{P}\lfloor\mathbf{E}$ with advantage $\frac{1}{2}$ using $\mathcal{O}(2^\kappa)$ queries as follows: Let $x_1, x_2, \ldots, x_t$ be fixed distinct n-bit strings where $t = \lceil \frac{2\kappa+n}{n-1} \rceil$. For all $i = 1, \ldots, t$ and $k_1', k_2' \in \{0,1\}^\kappa$, we first compute $u_i(k_1')$ and $v_i(k_2')$ by querying $(k_1', x_i, +)$ and $(k_2', \pi(x_i), -)$ to the right interface (the block cipher $\mathbf{E}$), where π is the permutation implemented by the left interface (the construction or a random permutation). Finally, we output 1 if there exist $z' \in \{0,1\}^n$ and $k_1', k_2' \in \{0,1\}^\kappa$ with $u_i(k_1') \oplus v_i(k_2') = z$ for all $i = 1, \ldots, t$, and 0 otherwise.

Clearly, we will always output 1 when interacting with $\mathsf{C}_{2,K_1,K_2,Z}\mathbf{E}$. However, if we interact with $\mathbf{P}\lfloor\mathbf{E}$ then for each k_1', k_2' and z the sequence $v_i(k_2')$ is a uniformly random t-tuple of distinct n-bit strings. Hence, the probability that $u_i(k_1') \oplus v_i(k_2') = z$ for all $i = 1, \ldots, t$ is at most

$$(2^n - t)!/2^n! < 2^{-t(n-1)}$$

and the probability that there exist such k_1', k_2' and z is hence at most $2^{2\kappa+n-t(n-1)} \leq \frac{1}{2}$ by the union bound.

Similarly, for C_1 one modifies the attack to compute $u_i(k_1')$ as the result of querying $\mathbf{E}$ with input $(k_1', x_i, -)$, whereas $v_i(k_2')$ is obtained by applying π^{-1} to the output of the query $(k_2', x_i, +)$. To attack C_3, we proceed symmetrically.

Double Randomization. It is slightly more complicated to attack the construction realizing the mapping with two randomization steps

$$C_{4,k_1,k_2,z_1,z_2}(m) = E_{k_2}(E_{k_1}(m \oplus z_1)) \oplus z_2 \ .$$

Here we show an attack requiring $\mathcal{O}(2^\kappa)$ block cipher queries and 2^n construction queries. Following on the above notation, we let $u_i(k_1')$ and $v_i(k_2')$ be the results of querying $\mathbf{E}$ with $(k_1', x_i, -)$ and $(k_2', x_i, +)$ for all $k_1', k_2' \in \{0,1\}^\kappa$ and $i = 1, \ldots, m$, but also query all of the given permutation π (this costs us 2^n queries). Then, as above we check for the existence of $k_1', k_2' \in \{0,1\}^\kappa$ and $z_1, z_2 \in \{0,1\}^n$ such that $\pi(z_1 \oplus u_i(k_1')) = v_i(k_2') \oplus z_2$ for all $i = 1, \ldots, t$. Note that the probability that this is achieved when π is chosen randomly is at most $2^{2\kappa+2n} \cdot \frac{(2^n-t)!}{2^n!} \leq \frac{1}{2}$ if we set $t = \lceil \frac{2k+2n}{n-1} \rceil$. A similar attack can be given when $z_1 = z_2$.

For completeness, note that for the remaining variant of doubly-randomized double cascade, i.e., for

$$C_{5,k_1,k_2,z_1,z_2}(m) = E_{k_2}(E_{k_1}(m) \oplus z_1) \oplus z_2$$

one could easily modify the presented proof for 2XOR to arrive at the same security guarantees as stated in Theorem 4.5 also for this construction.

Chapter 5

Resource-Restricted Indifferentiability

5.1 The Indifferentiability Framework

Certain cryptographic constructions are used (and hence have to be analyzed) in settings where some access to their internal state is available publicly, making it also within reach of any potential attacker. The notion of *indifferentiability* was introduced in [MRH04] to serve as a generalization of indistinguishability tailored for analyzing exactly this type of constructions. It also comes with a composition theorem formalizing the important property that an ideal primitive can be replaced by an indifferentiable construction in any context.

The indifferentiability framework found its most important application in the analysis of hash function constructions. Many existing cryptographic constructions were proven to be secure in the random oracle model (ROM) described in Section 2.4. However, it was shown in [CGH98] that there exist constructions that are secure in ROM, but become completely insecure once the random oracle is instantiated by *any* real hash function. Therefore, if we instantiate the random oracle by an existing cryptographic hash function H, the proof in the ROM can then only be taken as a heuristic argument towards the security of the overall construction. However, if one uses a hash function construction H^f that was proven indifferentiable from a random oracle when using an ideal

compression function f, this excludes any possible attacks exploiting the structure of H and reduces the security of the construction to the security of the underlying compression function f [CDMP05]. This is a much more compact object and is comparatively simpler to analyze. As a consequence, an indifferentiability proof in the setting with an ideal compression function is generally considered an important argument towards the security of a practical hash function design and many of the SHA-3 candidates (including the winner Keccak [BDPVA08a]) enjoy such a proof (see e.g. [BDPVA08b, CN08, DRRS09, DRS09, AMP10]). Moreover, the indifferentiability framework offers a precise formalism for studying the security of reductions of the random oracle to the ideal compression function, which is an interesting theoretical question by itself. We now present the formal definitions of two variants of this notion.

Classical (Weak) Indifferentiability. In the classical indifferentiability defined in [MRH04] one restricts only to resources having two interfaces. The first one, referred to as *private*, is meant to model the access to the resource by all honest parties. On the other hand, the second interface is called *public* and is present to model the adversarial access to the internal state of the resource. Since we focus on the indifferentiability setting throughout this chapter, unless stated otherwise we see all resources as also having 2 interfaces and refer to them as described above. Later, in Section 5.4 we also consider resources with more interfaces.

Let $\mathbf{S}$ and $\mathbf{T}$ be such 2-interface resources. For given sets Σ and $\mathcal{D}$ of converters and distinguishers, respectively, we define $\mathbf{T}$ *being ε-reducible to* $\mathbf{S}$ *in the sense of weak indifferentiability* (denoted $\mathbf{S} \xrightarrow[\text{wi}]{\varepsilon} \mathbf{T}$) as

$$\mathbf{S} \xrightarrow[\text{wi}]{\varepsilon} \mathbf{T} \overset{\text{def.}}{\Leftrightarrow} (\exists \pi \in \Sigma)(\forall \mathbf{D} \in \mathcal{D})(\exists \sigma \in \Sigma) : \Delta^{\mathbf{D}}(\pi\mathbf{S}, \mathbf{T}\sigma) \leq \varepsilon.$$

Usually we call $\mathbf{S}$ the real and $\mathbf{T}$ the ideal resource; hence also the random experiment of $\mathbf{D}$ interacting with $\pi\mathbf{S}$ (resp. $\mathbf{T}\sigma$) is called the real (resp. ideal) experiment.

The converters π and σ are typically referred to as the protocol and the simulator, respectively. This is because the converter π captures the protocol that the honest parties should apply to the real resource $\mathbf{S}$ in order to construct the resource $\mathbf{T}$ while σ simulates to the adversary in the ideal experiment an interface that is the same as the one he would be granted access to in the real experiment. The two settings distinguished are depicted in Fig. 5.1.

Note that by choosing the sets Σ and $\mathcal{D}$, this definition covers both information-theoretic and computational indifferentiability; moreover,

Figure 5.1: The real (left) and the ideal (right) setting considered in the definition of indifferentiability.

one could also easily derive an asymptotic definition. These remarks are also true for all other reducibility notions presented below.

Strong Indifferentiability. For given sets Σ and $\mathcal{D}$ we define $\mathbf{T}$ *being* ε*-reducible to* $\mathbf{S}$ *in the sense of strong indifferentiability* (denoted $\mathbf{S} \xrightarrow[\text{si}]{\varepsilon} \mathbf{T}$) as

$$\mathbf{S} \xrightarrow[\text{si}]{\varepsilon} \mathbf{T} \overset{\text{def.}}{\Leftrightarrow} (\exists \pi, \sigma \in \Sigma)(\forall \mathbf{D} \in \mathcal{D}) : \Delta^{\mathbf{D}}(\pi\mathbf{S}, \mathbf{T}\sigma) \leq \varepsilon.$$

It is easy to see that the definitions of strong and weak indifferentiability differ in the order of the quantifiers: while strong indifferentiability requires the existence of a single simulator that works for all considered distinguishers, for weak indifferentiability it suffices to exhibit a different simulator for each distinguisher. Clearly reducibility under strong indifferentiability implies reducibility under the weak one and moreover, positive indifferentiability results (such as those in [CDMP05] showing security of MD-variants) typically prove this stronger type of statement by exhibiting a simulator that does not depend on the distinguisher. A detailed discussion of the interesting relationship between these two forms of simulatability in various formalisms can be found in [HU05, Can01], here we only remark that both notions are composable in the spirit of Theorem 5.1 (see below).

Domain Extension for Hash Functions. As already discussed, the typical application of indifferentiability that we will have in mind throughout this chapter is the analysis of domain-extending constructions for hash functions. As an example, we briefly introduce the domain extension construction chop-MD from [CDMP05]. Let $f\colon\{0,1\}^{r+d} \to \{0,1\}^r$ be a compression function. The function chop-MD$^f\colon\{0,1\}^* \to \{0,1\}^{r/2}$ is defined as follows:

function chop-MD$^f(m)$
  $m' \leftarrow \mathsf{Pad}(m)$
  parse m' **as** $m_1\|\cdots\|m_b$ **for** $m_i \in \{0,1\}^d$
  $y_0 \leftarrow 0^r$ (or any fixed initialization vector)
  for $i = 1$ **to** b **do** $y_i \leftarrow f(m_i\|y_{i-1})$
  return first $r/2$ **bits of** y_b

The role of the function Pad is to append the length of the message and a padding in a decodable way to obtain m' with length being a multiple of d bits. It will not be relevant for our discussion.

5.1.1 Limitations of Classical Indifferentiability

Indifferentiability as well as indistinguishability comes in two flavours, namely *information-theoretic* and *computational*. In the information-theoretic variant, there are no restrictions posed on the resources available to the distinguisher and the simulator, while in the computational variant, they are both bound to be efficient. The typical specification of an efficient algorithm is by probabilistic polynomial time (PPT), which implies also polynomial memory and randomness limitations. This way of restricting adversarial resources is also present in other simulation-based security notions such as universal composability [Can01] and reactive simulatability [BPW04].

These two degrees of resource restrictions are sufficient for most natural settings and hence indifferentiability results have a wide scope of applicability. However, there are practical settings where this rough approach is not sufficient and a more fine-grained analysis is needed. One such scenario was recently exhibited in [RSS11] in the context of auditable storage. Let us briefly review this scenario, since it will serve as a starting point for our work.

Storage-Auditing Scenario from [RSS11]. The example put forward in [RSS11] is a two-party protocol for verification of storage. Its goal is to allow the first party (the user) to verify that the second party (the server – e.g. a storage service) is properly storing a certain piece of data that the user has provided earlier.

The challenge-response protocol for this task that is analyzed in [RSS11] works as follows: the user sends the server a random challenge c and the server is required to respond with the value $H(m\|c)$ for a cryptographically secure hash function H. As argued in [RSS11], this protocol is clearly secure in the random oracle model. However, if the hash function is instantiated by the Merkle-Damgård-chop construction $\mathsf{chop\text{-}MD}^f$ (described in Section 5.1) using an ideal compression function f, the protocol becomes completely insecure. A malicious server can simply parse the data m into blocks $m_1, \ldots, m_\ell$ and compute the value

$$r := f(f(\cdots f(f(IV, m_1), m_2)\cdots), m_\ell).$$

Now it can discard the data m and store only the value r, allowing it to respond correctly to any challenge c by returning the first half of the bits of $f(r, c)$. [13]

Since the construction chop-MD was shown indifferentiable from a random oracle in [CDMP05], this example contradicts the common understanding that an ideal primitive can be replaced by an indifferentiable construction in any context without compromising the security. The goal of the rest of this section is to explain this seemingly inconsistent situation. In our view, the best way to understand it is by explicitly taking into account the memory requirements of the simulator that is used to prove indifferentiability of the chop-MD construction from a random oracle.

First, let us see why this is a relevant aspect. The role of a simulator in an indifferentiability reduction statement is to capture the fact that anything that the adversary would be able to obtain from its interaction with the real resource, she could also obtain when interacting with the ideal one. This is since she could perform all the tasks embodied in the simulator on her own. However, such argumentation is only valid if it is feasible for the adversary to incorporate the simulator into itself, not violating its own complexity limitations.

This is clearly not the case in the scenario described above. If the goal of the adversary is to pass the challenge without actually storing the data m, it cannot afford to perform the job of the simulator itself. This is because the simulator proving indifferentiability of chop-MD from a random oracle presented in [CDMP05] actually remembers all the queried values during the interaction. Hence if we take the adversary that is able to cheat to pass the verification protocol with chop-MD without remembering m and try to modify it to work in the random oracle setting, it would additionally need to perform the simulator's job, remembering the whole data m. This invalidates the reduction proof.

5.1.2 Contributions of This Chapter

We show a general approach to treating indifferentiability scenarios comprising resource restrictions such as the one given in [RSS11]. Choosing memory as the particular resource to consider, we model indifferentiability statements with greater focus on memory requirements of the simulators.

[13] For simplicity, we assumed that the length of m is divisible by the block length and the challenge consists of a single block.

Memory-Aware Reducibility. In Section 5.2 we introduce the notion of *memory-aware reducibility* that is derived from reducibility in the classical indifferentiability setting as given in [MRH04, MR11], but does not allow the memory requirements of the simulator to be "swept under the rug", requiring only that they are polynomial. It requires any memory necessary for the simulator to be explicitly modeled as a part of the ideal primitive; with the intuitive meaning that the real construction is provably as good as the ideal one as long as we assume that the adversary has the necessary amount of memory available. We also give a composition theorem for this new notion.

Note that this does not render the original notion of indifferentiability incorrect or obsolete. A (computational) indifferentiability statement never aspired to give any guarantees on the memory requirements of the simulator beyond its efficiency (and hence polynomial memory). Therefore, in situations where such a guarantee is not sufficient (as in the scenario from [RSS11]), one has to resort to a more detailed modeling of the resources involved, as we demonstrate here.

An independent approach to analyzing the complexity of the simulator in an indifferentiability statement appeared recently in [DRST12], where the authors focus on the number of queries the simulator issues per one invocation. To the best of our knowledge, our work is the first one pointing out the importance of the simulator's memory requirements.

Simulator Memory for Domain Extension. In Section 5.3 we look at the most important application of indifferentiability: the question of *domain extension* for public random functions. More precisely, we consider constructions that can be used to obtain an arbitrary input-length random oracle $\mathbf{R}^{*,n}\colon\{0,1\}^* \to \{0,1\}^n$ from an ideal compression function $\mathbf{R}^{m,r}\colon\{0,1\}^m \to \{0,1\}^r$ in an indifferentiable way, such as the various variants of the Merkle-Damgård construction proposed in [CDMP05]. We also consider the question of finite domain extension, i.e., constructing $\mathbf{R}^{\ell,r}$ from $\mathbf{R}^{m,r}$ for $\ell > m$.

The formalism of memory-aware reducibility allows us to investigate the minimal necessary memory requirements of the simulator for *any* such domain-extension construction. We prove two lower bounds on the memory required by the simulator, with the following consequences (see Section 5.3 for the precise bounds):

1. With stateless simulators (i.e., without any memory) even domain extension by a single bit (i.e., $\ell = m + 1$) is impossible.

2. For a natural class of simulators issuing at most one query to the ideal resource per invocation, any simulator for a domain extension by d bits (i.e., $\ell - m = d$) requires at least d bits of memory.

These bounds hold for both the information-theoretic and the computational setting. They naturally imply analogous impossibility results for constructing an arbitrary input-length random oracle, with the obvious transition of ℓ denoting the length of the longest query issued to the random oracle. This answers negatively the open question of the existence of such a construction using no simulator memory asked in [RSS11]. To appreciate the relevance of the memory requirements of a simulator in a reduction statement, we again point to the example given in Section 5.1.1.

As another consequence, we also obtain the irreducibility of the random oracle to the ideal cipher with respect to stateless simulators, in contrast to the equivalence of these two ideal primitives with respect to classical indifferentiability [CDMP05, CPS08, HKT11].

Random Oracles Used by Multiple Parties. The impossibility results described above have some intriguing consequences for the setting where a random oracle is being used in a protocol by multiple parties, if we consider that several of these parties might deviate from the prescribed protocol in a potentially non-coordinated way (for example due to conflicting goals). According to the abstract cryptography framework [MR11], a security notion for such a situation has to involve local simulators for each of the parties that deviate from the protocol (see Section 5.4.1 for details). Clearly, if a distinguisher is allowed to access two such simulators (for two of the parties) in the ideal world, these have to be stateless as otherwise they would produce inconsistent results when brought to different states. On the other hand, our results described above imply that also for this setting, no stateless simulator can exist. Hence, roughly speaking, one can conclude that for settings where one cannot assume a central adversary coordinating all the actions of the misbehaving parties, no secure construction of a random oracle from an ideal compression function exists. This might be relevant in the contexts of *rational cryptography* [HT04], *incoercible computation* [CG96], *receipt-free voting* [BT94] or *collusion-free computation* [LMs05, AKL+09] and its recent composable variants [AKMZ12, CV12]. We formalize the above argument in Section 5.4 as an illustration of the impact of our results.

This chapter covers results that are presented in the paper [DGHM13].

5.2 Memory-Aware Reducibility

5.2.1 Stateless Simulators

To formally define memory-aware reducibility, we will need to consider the class of *stateless converters* in the following sense. A stateless converter is a converter that uses no memory between answering outer queries, i.e., its (possibly randomized) behavior for a particular query depends only on the query itself and the ongoing interaction at the inner interface, not on previous outer queries and the transcript of the interaction during their evaluation. A formal definition follows.

Definition 5.1 *A converter ϕ is stateless if there exists a sequence of conditional probability distributions $\mathsf{p}^{\phi}_{IX_{j+1}|X_1\ldots X_jY_1\ldots Y_jQ}$ for $j \geq 0$ such that whenever ϕ received a query q at the outer interface and has then issued the sequence of queries $x_1, \ldots, x_j$ to the inner interface, obtaining responses $y_1, \ldots, y_j$, then $\mathsf{p}^{\phi}_{IX_{j+1}|X_1\ldots X_jY_1\ldots Y_jQ}(i, x_{j+1}, x_1, \ldots, x_j, y_1, \ldots, y_j, q)$ determines the probability that its next action will be to output the value x_{j+1} at interface $i \in \{\mathsf{inner}, \mathsf{outer}\}$. For a set of converters Σ we denote by Σ_{s} the set of all stateless converters from Σ.*

For example, the converter accessing an ideal compression function and realizing a Merkle-Damgård construction on top of it would be stateless according to the above definition.

5.2.2 Quantifying the Memory Requirements of the Simulator

Let $\mathbf{M}_s$ denote a resource that provides a dummy private interface and at the public (adversarial) interface, it provides the functionality of s-bit memory, i.e., allows efficient storage and retrieval of arbitrary information such that its size is in every point in time upper-bounded by s bits. Note that the behavior of $\mathbf{M}_s$ can be unambiguously captured at the abstraction level of random systems. This, together with the definition of a stateless converter by its input-output behavior, leads to expressing all our considerations at this abstraction level.

To quantify the memory requirements of the simulator in a reducibility statement we shall use the following formalism, which we broadly denote as *memory-aware reducibility*.

Definition 5.2 *For given sets Σ and $\mathcal{D}$ of converters and distinguishers, respectively, we define* $\mathbf{T}$ being ε-reducible to $\mathbf{S}$ in the presence of s bits of adversarial memory *(denoted* $\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon,s} \mathbf{T}$*) as*

$$\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon,s} \mathbf{T} \overset{\text{def.}}{\Leftrightarrow} (\exists \pi \in \Sigma)(\forall \mathbf{D} \in \mathcal{D})(\exists \sigma \in \Sigma_{\mathsf{s}}) : \Delta^{\mathbf{D}}(\pi\mathbf{S}, [\mathbf{T}||\mathbf{M}_s]\sigma) \leq \varepsilon.$$

Informally speaking, the statement $\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon,s} \mathbf{T}$ indicates that $\mathbf{T}$ can be constructed securely from $\mathbf{S}$ within error ε in an environment where the adversary has s bits of memory available. In other words, whatever the adversary can achieve in the real world she could also achieve in the ideal world, but it might need up to s more bits of memory to do so. Evaluating whether this is acceptable depends on the context in which we want to use $\mathbf{S}$ instead of $\mathbf{T}$.

As before, by specifying the sets of converters and distinguishers to be considered, this definition covers both computational and information-theoretic memory-aware reducibility. As in the case of classical indifferentiability, the transition to an asymptotic definition would be straightforward.

In the special case of no memory (i.e., $s = 0$) the notion of memory-aware reducibility $\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon,0} \mathbf{T}$ collapses to the notion of reducibility with stateless simulators. If we refer to this situation, we usually omit the 0 and simply write $\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon} \mathbf{T}$. Technically, this special case is equivalent to the notion of reset indifferentiability introduced in [RSS11]: First, if the simulator is stateless, then it can be used also in the scenario with resets with the same outcome. On the other hand, any simulator that satisfies the requirements of reset indifferentiability must be able to simulate successfully even in presence of an adversary that resets it before every query, hence there also exists an equivalent stateless simulator. However, our motivation to introduce stateless simulators is completely different. We do not put it forward as a security notion by itself, but only as a tool for modeling the memory requirements of the simulator explicitly.

5.2.3 Composability

The formalism of memory-aware reducibility given above leads to statements that are composable under some natural closure assumptions on the sets of converters and distinguishers considered. These assumptions correspond to the notion of a cryptographic algebra [MR11, Def. 14] and of a compatible distinguisher class [MR11, Def. 16] applied to the more

specific setting of discrete systems. Here we only state these assumptions (and consequently also the theorem) in an informal way to sketch the composability achieved.

Theorem 5.1 (informal) *Let us assume that the considered set of converters Σ is closed under both sequential composition $\circ$ and parallel composition $|$ and the considered set of distinguishers $\mathcal{D}$ is closed under the emulation of a converter and the emulation of a resource. Let $\mathbf{S}$, $\mathbf{T}$, and $\mathbf{V}$ be resources such that $\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon_1, s_1} \mathbf{T}$ and $\mathbf{T} \xrightarrow[\mathrm{m}]{\varepsilon_2, s_2} \mathbf{V}$. Then:*

1. *For any resource $\mathbf{U}$ we have $\mathbf{S}\|\mathbf{U} \xrightarrow[\mathrm{m}]{\varepsilon_1, s_1} \mathbf{T}\|\mathbf{U}$ and $\mathbf{U}\|\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon_1, s_1} \mathbf{U}\|\mathbf{T}$.*
2. *We have $\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon_1+\varepsilon_2, s_1+s_2} \mathbf{V}$.*

5.3 Lower Bounds on Simulator Memory for Any Domain-Extending Construction

We now investigate the amount of memory that we must assume to be available to the adversary in order to be able to conclude the security of classical domain extension constructions for hash functions.

5.3.1 Fixed Input-Length Random Oracles

The following theorem upper-bounds the achievable domain extension for fixed input-length random oracles, given a bound on the memory available to the simulator. In particular, it implies that without simulator memory, even domain extension by a single bit becomes impossible, thus solving an open problem introduced in [RSS11]. In our proof we use the following inequality given by Fano in [Fan61]; recall that $H(\cdot)$ and $h(\cdot)$ denote the Shannon entropy and binary entropy functions as defined in Section 2.1.

Lemma 5.1 (Fano's inequality) *Let X and Y be random variables. For the error probability p_e of any algorithm reconstructing X given Y we have*

$$h(p_e) + p_e \cdot \log(|\mathcal{X}| - 1) \geq H(X|Y)$$

where $\mathcal{X}$ is the support of X.

If we assume X to be distributed over bitstrings of a certain length, one can easily derive a corollary of the above inequality that lower-bounds the error probability of reconstruction of a randomly chosen bit of X.

Corollary 5.1 *Let X, Y be random variables such that $X \in \{0,1\}^n$, i.e., $X = (X_1, X_2, \ldots, X_n)$ for $X_i \in \{0,1\}$. Moreover, let us consider any algorithm reconstructing X given Y and denote by $\tilde{X}_i$ the reconstruction of X_i. Then for the probability of error in a uniformly chosen random bit $\bar{p}_e := \frac{1}{n}\sum_{i=1}^{n} \mathsf{P}[\tilde{X}_i \neq X_i]$ we have*

$$h(\bar{p}_e) \geq \frac{1}{n} H(X|Y).$$

Proof: We have

$$\begin{aligned} h(\bar{p}_e) &= h\left(\frac{1}{n}\sum_{i=1}^{n} \mathsf{P}[\tilde{X}_i \neq X_i]\right) \geq \frac{1}{n}\sum_{i=1}^{n} h(\mathsf{P}[\tilde{X}_i \neq X_i]) \\ &\geq \frac{1}{n}\sum_{i=1}^{n} H(X_i|Y) \geq \frac{1}{n} H(X|Y) \end{aligned}$$

where we applied the Jensen inequality, the Fano inequality for the case $|\mathcal{X}| = 2$ and the basic properties of the entropy function. ∎

Now we are ready to prove the first of the key results of this chapter.

Theorem 5.2 *Assume that for any $\pi \in \Sigma$, the distinguisher $\mathbf{D}$ constructed from π according to Fig. 5.2 is present in $\mathcal{D}$. Then any reduction $\mathbf{R}^{m,r} \xrightarrow[\mathrm{m}]{\varepsilon,s} \mathbf{R}^{\ell,r}$ with*[14] *$r \geq 2$ and $\varepsilon \leq 0.04$ satisfies*[15]

$$\ell - m \leq s + \lceil \log(\min\{s,t\}) \rceil \tag{5.1}$$

where $t \geq 1$ denotes an upper bound on the number of queries the simulator issues to the ideal primitive $\mathbf{R}^{\ell,r}$ to answer a single query.

Proof: Recalling Def. 5.2, let us denote by π the protocol performing the reduction from the statement and let us consider a distinguisher $\mathbf{D}$ interacting with either $\pi\mathbf{R}^{m,r}$ or $[\mathbf{R}^{\ell,r} \| \mathbf{M}_s]\sigma$, where σ is the stateless simulator corresponding to $\mathbf{D}$.

[14] The bound degrades gracefully for smaller r and bigger ε. In particular, for the same ε and $r = 1$ with no memory ($s = 0$) domain extension by a single bit is still impossible.

[15] To avoid handling the special case $s = 0$ separately we use the notational convention $\log 0 = 0$ throughout this section.

Distinguisher $\mathbf{D}(\mathbf{S})$: // $\mathbf{S} \in \{\pi\mathbf{R}^{m,r}, [\mathbf{R}^{\ell,r} \| \mathbf{M}_s]\sigma\}$

1: $X \xleftarrow{\$} \{0,1\}^\ell$
2: **query** $Y := [\mathbf{S}]_L(X)$
3: **simulate** π **to evaluate** $\hat{Y} := \pi(X)$
   answer new inner queries by querying $[\mathbf{S}]_R$
   answer repeated inner queries consistently
4: **if** $Y = \hat{Y}$ **then**
5: **return** 1
6: **return** 0

Figure 5.2: The distinguisher $\mathbf{D}$ for the proof of Theorem 5.2.

Overview. In our proof we only consider the trivial distinguisher $\mathbf{D}$ given in Fig. 5.2 that chooses a random input $X \in \{0,1\}^\ell$ and then evaluates the $\{0,1\}^\ell$-domain function on the input X in two different ways. First it queries the private (left) interface for the whole input X; second it simulates the protocol π on X on its own and uses the public (right) interface to answer the $\{0,1\}^m$-queries issued by π. Moreover, it never repeats a query to the right interface: in case the simulated protocol π would issue a repeated query, it is answered as before. We will refer to this modified (simulated) protocol π as π'; note that $\mathbf{D}$ is capable of this modification since it can keep the history of query-answer pairs in its state. Finally, $\mathbf{D}$ outputs 1 if and only if the two values obtained from these evaluations are equal. The distinguisher $\mathbf{D}$ participating in both the real and the ideal setting is depicted in Fig. 5.3. Note that it is natural to expect that this simple distinguisher $\mathbf{D}$ is present in any reasonable distinguisher class.

Clearly if $\mathbf{D}$ interacts with $\pi\mathbf{R}^{m,r}$ it always outputs 1. It remains to analyze the probability of $\mathbf{D}$ outputting 1 when interacting with $[\mathbf{R}^{\ell,r} \| \mathbf{M}_s]\sigma$. To this end, we consider the ideal setting depicted at the bottom part of Fig. 5.3 and upper-bound the probability that the output of the protocol π' simulated by $\mathbf{D}$ will be the correct value $\mathbf{R}^{\ell,r}(X)$. Informally speaking, we do this by upper-bounding the amount of useful information that π' can obtain about the actual values of $\mathbf{R}^{\ell,r}$ and show that it is not enough to recover $\mathbf{R}^{\ell,r}(X)$ with sufficient probability.

We use two separate approaches to bound this amount, each proving the above claim for one of the values in the minimum term in (5.1). In the first approach, we upper-bound the number of distinct queries the

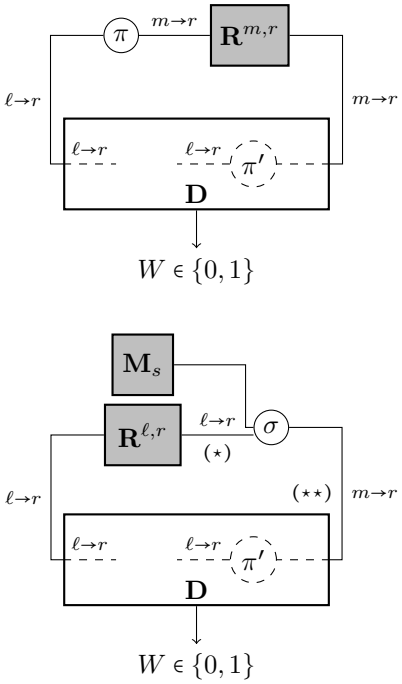

Figure 5.3: The real and the ideal setting for the proof of Theorem 5.2. The notation $i \to o$ describes an interface that accepts queries from $\{0,1\}^i$ and responds with elements from $\{0,1\}^o$.

simulator σ is able to issue to $\mathbf{R}^{\ell,r}$ in any of its possible configurations (determined by the query it is answering and the state of its memory), thus using the channel denoted $(\star)$ in Fig. 5.3 as the "bottle-neck" to be considered. On the other hand, in the second approach we upper-bound the information provided by σ to π', this time the channel $(\star\star)$ acting as the "bottle-neck". We now give the details of both approaches.

First Approach: The Channel $(\star)$. To capture the randomness involved in the ideal distinguishing experiment, we denote by R_w the (fresh, independent) internal randomness used by σ when it is answering an outer

query $w \in \{0,1\}^m$ *for the first time*[16] and let $R_\sigma := \{R_w\}_{w\in\{0,1\}^m}$. Moreover, let $R_{\mathbf{R}}$ denote the overall randomness of the ideal resource $\mathbf{R}^{\ell,r}$, i.e., its function table. For a fixed randomness $R_\sigma = r_\sigma$ and $R_{\mathbf{R}} = r_{\mathbf{R}}$ where $r_\sigma = \{r_w\}_{w\in\{0,1\}^m}$, let us denote by $f(w,z,r_w,r_{\mathbf{R}}) \subseteq \{0,1\}^\ell$ the set of all queries that the (stateless) simulator σ issues to the random oracle $\mathbf{R}^{\ell,r}$ while evaluating an outer query w with the available memory $\mathbf{M}_s$ containing value $z \in \{0,1\}^s$, using randomness r_w while the responses from $\mathbf{R}^{\ell,r}$ are determined by $r_{\mathbf{R}}$. Since the random variables R_w and $R_{\mathbf{R}}$ represent the only sources of randomness in this evaluation, f is a well-defined deterministic mapping and by our assumption $|f(w,z,r_w,r_{\mathbf{R}})| \le t$ for all possible inputs. Let us define $\mathcal{S}_{r_\sigma,r_{\mathbf{R}}}$ to be the set of all possible queries under all inputs (w,z) for this fixed randomness $(r_\sigma,r_{\mathbf{R}})$, i.e.,

$$\mathcal{S}_{r_\sigma,r_{\mathbf{R}}} := \bigcup_{\substack{w\in\{0,1\}^m\\ z\in\{0,1\}^s}} f(w,z,r_w,r_{\mathbf{R}}),$$

then we have $|\mathcal{S}_{r_\sigma,r_{\mathbf{R}}}| \le 2^{m+s+\log t}$ for any $(r_\sigma,r_{\mathbf{R}})$. Since $X \in \{0,1\}^\ell$ was chosen at random and independently from $\mathcal{S}_{R_\sigma,R_{\mathbf{R}}}$, we obtain $\mathsf{P}(X \in \mathcal{S}_{R_\sigma,R_{\mathbf{R}}}) \le 2^{m+s+\log t}/2^\ell = 2^{m+s+\log t-\ell}$. Hence, if $\ell - m > s + \lceil \log t \rceil$ then $X \notin \mathcal{S}_{R_\sigma,R_{\mathbf{R}}}$ with probability at least $1/2$. However, if $X \notin \mathcal{S}_{R_\sigma,R_{\mathbf{R}}}$ then π' has no information about $\mathbf{R}^{\ell,r}(X)$ and hence can only guess it successfully with negligible probability. Therefore, any proper simulation requires $\ell - m \le s + \lceil \log t \rceil$.

Second Approach: The Channel ($\star\star$). In this case, let us denote by $\sigma_z(w)$ the response of σ to a query $w \in \{0,1\}^m$ with the available memory set to the value $z \in \{0,1\}^s$ and let us denote by $Z'(w,z) \in \{0,1\}^s$ the new contents of the memory after this invocation of σ. Note that since σ is stateless, both $\sigma_z(w)$ and $Z'(w,z)$ are random variables fully determined by the function table of $\mathbf{R}^{\ell,r}$ and the internal randomness of σ used during this invocation. We can now define T to be the table containing a sample of $\sigma_z(w)$ and $Z'(w,z)$ for all possible w and z, formally

$$T := \{(\sigma_z(w), Z'(w,z))\}_{(w,z)\in\{0,1\}^m\times\{0,1\}^s}.$$

Then T can be seen as a random variable distributed over $\{0,1\}^{(r+s)\cdot 2^{m+s}}$ and is again determined by the function table of $\mathbf{R}^{\ell,r}$ and the randomness used by σ.

[16] Formally, one can imagine σ being replaced by a stateful simulator that chooses all random variables R_w at the beginning and then uses it when the query w arrives for the first time. This view does not change the outcomes of the experiment.

We now consider a different protocol ρ instead of π' which we allow to be stateful, but we only provide it with access to T, not σ (which we denote by ρ^T). We claim that the probability of the best such ρ in reconstructing $\mathbf{R}^{\ell,r}(X)$ given access to T is not smaller than the same probability for π' given access to the right interface of $[\mathbf{R}^{\ell,r} \| \mathbf{M}_s]\sigma$, i.e., we have

$$\max_{\rho} \mathsf{P}[\rho^T(X) = \mathbf{R}^{\ell,r}(X)] \geq \mathsf{P}\left[[[\mathbf{R}^{\ell,r} \| \mathbf{M}_s]\sigma\pi']_R(X) = \mathbf{R}^{\ell,r}(X)\right]. \quad (5.2)$$

This is because one possible ρ to be considered on the left side of (5.2) is the following: it simulates π' and answers each of its queries to σ using the respective value from T instead (recall that π' asks each query at most once). It also keeps track of the memory contents in its own state, updating it after each answered query according to the value given in T. This ρ clearly achieves equality in (5.2).

Now, since any ρ as described above only has access to T, we can upper-bound the probability of ρ successfully reconstructing $\mathbf{R}^{\ell,r}(X)$ based on T. To simplify the notation, we shall denote by F the whole function table of $\mathbf{R}^{\ell,r}$ seen as a random variable (uniformly distributed over $\{0,1\}^{r2^\ell}$). The value X is chosen independently at random, hence we can apply Corollary 5.1 to obtain a lower bound on the probability $\bar{p}_e$ of error in a randomly chosen bit of $\mathbf{R}^{\ell,r}(X)$:

$$\begin{aligned} h(\bar{p}_e) &\geq \frac{1}{r2^\ell} H(F|T) = \frac{1}{r2^\ell}\left(H(FT) - H(T)\right) \geq \frac{1}{r2^\ell}\left(H(F) - H(T)\right) \\ &\geq \frac{1}{r2^\ell}\left(r2^\ell - (r+s)2^{m+s}\right) = 1 - 2^{m+s-\ell} - \left(\frac{s}{r}\right)2^{m+s-\ell} \end{aligned}$$

If $\ell - m > s + \lceil \log s \rceil$ then since m, s, ℓ are integers we get $2^{m+s-\ell} \leq 1/2$ and $(s/r) \cdot 2^{m+s-\ell} \leq 1/2r$, hence $h(\bar{p}_e) \geq 1/2 - 1/2r$, resulting in $\bar{p}_e \geq 0.04$ for $r \geq 2$. Therefore any simulator successful beyond 96% has to satisfy $\ell - m \leq s + \lceil \log s \rceil$ as desired. ∎

Before we apply our result also to other contexts, note that our argument above is completely information-theoretic and hence the bound applies to both information-theoretic *and* computational memory-aware reducibility.

5.3.2 Arbitrary Input-Length Random Oracles

Seen from a different perspective, the above theorem also imposes a lower bound on the required simulator memory for any reduction of an

arbitrary input-length random oracle to a fixed input-length random oracle (i.e., an ideal compression function) as a function of the lengths of hashed messages. We now investigate this setting.

In the statement below we shall again consider the distinguisher given in Fig. 5.2, this time for the setting of the reduction $\mathbf{R}^{m,r} \xrightarrow[\mathrm{m}]{\varepsilon,s} \mathbf{R}^{*,r}$. To emphasize that it chooses the value X from the set $\{0,1\}^\ell \subseteq \{0,1\}^*$, we shall denote it $\mathbf{D}_\ell$, note that it again implicitly depends on a protocol π. One could give a similar statement also for a distinguisher asking several private queries and using the public interface to evaluate the protocol π on the longest one.

Corollary 5.2 *If for every $\pi \in \Sigma$ the distinguisher $\mathbf{D}_\ell$ described above is present in $\mathcal{D}$ then any reduction $\mathbf{R}^{m,r} \xrightarrow[\mathrm{m}]{\varepsilon,s} \mathbf{R}^{*,r}$ with $r \geq 2$ and $\varepsilon \leq 0.04$ satisfies*

$$s \geq \ell - m - \lceil \log(\min\{s,t\}) \rceil$$

where $t \geq 1$ denotes an upper bound on the number of queries the simulator itself issues to the ideal primitive to answer a single query. For the more general case $\mathbf{R}^{m,r} \xrightarrow[\mathrm{m}]{\varepsilon,s} \mathbf{R}^{*,n}$ we still have $s \geq \ell - m - \lceil \log t \rceil$ under the same assumptions.

Proof: The argument is analogous to the proof of Theorem 5.2 with a trivial modification to account for $\mathbf{R}^{*,r}$ as the ideal resource: in part $(\star\star)$ the random variable F now only stands for the part of the function table of $\mathbf{R}^{*,r}$ corresponding to inputs of length ℓ. The second claim holds since the analysis of the case $(\star)$ in the proof of Theorem 5.2 does not depend on the range of the ideal resource. ∎

Discussion. To illustrate the meaning of the above statement, let us consider the domain extension construction chop-MD described in Section 5.1. The simulator presented in [CDMP05] to show its indifferentiability from a random oracle would use (without optimizations) roughly $(1 + r/m) \cdot \ell$ bits of memory to answer all queries of the distinguisher $\mathbf{D}_\ell$ considered in Corollary 5.2, while always asking at most one query to the ideal primitive to answer a single query itself. Our result implies that for any indifferentiable domain extension construction, if the respective simulator is of this single-query form then it needs at least $\ell - m$ bits of memory. Since typically $\ell \gg m$, this implies that the simulator given in [CDMP05] has essentially optimal memory requirements within this class (i.e., linear in ℓ).

5.3.3 Random Oracle vs. Ideal Cipher

Our proof of Theorem 5.2 relies on information-theoretic arguments that remain valid also after introducing additional permutation structure into the real resource. Hence, as a side result, we also obtain the impossibility of reducing an arbitrary input-length random oracle to an ideal cipher with respect to stateless simulators. This is in contrast to the results of [CDMP05] that demonstrate the possibility of such reduction with respect to stateful simulators.

We denote by $\mathbf{E}^{k,n}$ the ideal cipher with key length k and block length n. Recall that $\mathbf{E}^{k,n}$ allows both encryption and decryption queries under an arbitrary key, hence taking an input of $k + n + 1$ bits. Taking this into account, the argument is analogous to the part $(\star\star)$ in the proof of Theorem 5.2 and is hence omitted.

Corollary 5.3 *If for every $\pi \in \Sigma$ and for $\ell = k + n + \lceil \log(n/r) \rceil + 2$ the distinguisher $\mathbf{D}_\ell$ considered in Corollary 5.2 is present in $\mathcal{D}$, then any reduction $\mathbf{E}^{k,n} \xrightarrow[\mathrm{m}]{\varepsilon} \mathbf{R}^{*,r}$ has to satisfy $\varepsilon \geq 0.1$.*

5.4 Domain Extension is Impossible in a General Multi-Party Setting

As a particular application of our results, in this section we present some interesting consequences of the lower bound on simulator memory for domain extension of random functions established in the previous section.

Multiple Parties. The approach taken in any indifferentiability analysis is to model the system in question as having two interfaces: the private one and the public one, as described in Section 5.1. However, we often consider the constructed primitives to be used in an environment or protocol involving multiple parties. For example, a random oracle is typically understood to be available to all entities participating in a protocol (or possibly many concurrent protocols) that use it. The generic translation of an indifferentiability result into a security guarantee for such a setting is then tacitly assumed. Namely, we view *all* the honest parties as accessing identical copies of the private interface of the real primitive, each party running a local copy of the protocol π realizing the reduction.

On the other hand, *all* the misbehaving parties are allowed to access the internals of the construction via identical copies of the public interface.

This implicit reasoning step imposes some requirements on the reduction used. First of all, the protocol π used by all the honest parties must be stateless in the sense of Definition 5.1. This is intuitively easy to see, since an inherently stateful protocol could lead to inconsistent behavior observed by different honest parties. For example, when constructing a random oracle from an ideal compression function, in the ideal world the resource (a random oracle) is stateful, with a single state (its function table) accessible to all honest parties. If in the real world a part of this state was stored by the protocol, different parties running different instances of the protocol could obtain different function values for the same query. Naturally, typical protocols constructing a random oracle from an ideal compression function such as the variants of the Merkle-Damgård construction proposed in [CDMP05] are indeed designed to be stateless.

It turns out that for a generic transition from an indifferentiability statement to a security guarantee in a setting with multiple parties, using stateless protocols is by itself *not* sufficient. However, before we can formally approach this question, we first have to describe how we formulate security requirements in the multi-party setting. For this task we use the approach of abstract cryptography (AC) of Maurer and Renner.

5.4.1 AC Reducibility

Here we only give a very brief introduction to the AC framework required for our exposition, further details and the justification of the framework are given in [MR11]. The framework introduces a strong notion of isomorphism given at a very abstract level that, when applied to the particular setting of abstract systems, gives rise to the security notion described below. Its main technical difference compared to other simulation-based security definitions (e.g. [Can01, BPW04]) relevant for our discussion is that it requires the existence of a *local* simulator for each of the parties.

From now on, we will be discussing more general resources having n interfaces labeled $1, \ldots, n$. Recall that if $\hat{\phi} = (\phi_1, \ldots, \phi_n)$ is an n-tuple of converters and $\mathbf{S}$ is an n-interface resource, we write $\hat{\phi}\mathbf{S}$ to denote the resource $\mathbf{S}$ with the converter ϕ_i applied to its i-th interface for all $i \in \{1, \ldots, n\}$. For a subset $\mathcal{P} \subseteq \{1, \ldots, n\}$ and an n-tuple of converters

$\hat{\phi} = (\phi_1, \ldots, \phi_n)$ let us denote by $\hat{\phi}_{\mathcal{P}}$ the n-tuple of converters that is obtained from $\hat{\phi}$ by replacing all converters on positions *not* in $\mathcal{P}$ by the identity converter id. Hence, for two n-interface resources $\mathbf{S}$ and $\mathbf{T}$, the notation $\hat{\pi}_{\mathcal{P}}\mathbf{S}$ below denotes the system $\mathbf{S}$ with a protocol from $\hat{\pi}$ connected to every interface in $\mathcal{P}$ while $\hat{\sigma}_{\overline{\mathcal{P}}}\mathbf{T}$ denotes $\mathbf{T}$ with a simulator from $\hat{\sigma}$ connected to every interface *not* in $\mathcal{P}$.

Let $\mathbf{S}$ and $\mathbf{T}$ be n-interface resources. For some understood Σ and $\mathcal{D}$, we say that $\mathbf{T}$ is ε-reducible to $\mathbf{S}$ in the sense of AC (denoted $\mathbf{S} \xrightarrow[\text{AC}]{\varepsilon} \mathbf{T}$) if there exist two n-tuples of converters $\hat{\pi} = (\pi_1, \ldots, \pi_n)$ and $\hat{\sigma} = (\sigma_1, \ldots, \sigma_n)$ such that for every subset $\mathcal{P}$ of indices $\{1, \ldots, n\}$ and every distinguisher $\mathbf{D} \in \mathcal{D}$ we have $\Delta^{\mathbf{D}}(\hat{\pi}_{\mathcal{P}}\mathbf{S}, \hat{\sigma}_{\overline{\mathcal{P}}}\mathbf{T}) \leq \varepsilon$, i.e.:

$$\mathbf{S} \xrightarrow[\text{AC}]{\varepsilon} \mathbf{T} \overset{\text{def.}}{\Leftrightarrow} (\exists \hat{\pi}, \hat{\sigma} \in \Sigma^n)(\forall \mathcal{P} \subseteq \{1, \ldots, n\})(\forall \mathbf{D} \in \mathcal{D}) : \Delta^{\mathbf{D}}(\hat{\pi}_{\mathcal{P}}\mathbf{S}, \hat{\sigma}_{\overline{\mathcal{P}}}\mathbf{T}) \leq \varepsilon. \tag{5.3}$$

For a 1-interface resource $\mathbf{S}$, let us denote by $\hat{\mathbf{S}}_n$ the n-interface resource that provides access to the same internal copy of $\mathbf{S}$ on each of its interfaces. For $\mathcal{P} \subseteq \{1, \ldots, n\}$ and a distinguisher $\mathbf{D}$ from the class $\mathcal{D}$ let $\mathsf{Proj}_{\mathcal{P}}(\mathbf{D})$ denote a new distinguisher for the 2-interface indifferentiability setting that works exactly as $\mathbf{D}$ does but asks all $\mathbf{D}$'s queries to interfaces in $\mathcal{P}$ at the private interface instead and all $\mathbf{D}$'s queries to interfaces in $\overline{\mathcal{P}}$ at the public interface instead. Moreover, let $\mathsf{Proj}_{\mathcal{P}}(\mathcal{D}) := \{\mathsf{Proj}_{\mathcal{P}}(\mathbf{D}) \mid \mathbf{D} \in \mathcal{D}\}$.

5.4.2 Generic Transition to the n-Party Setting

Now we are ready to state a theorem that formalizes the above-mentioned generic transition from any indifferentiability statement to a more meaningful statement in the multi-party AC setting under certain assumptions. Since the isomorphism notion introduced in the AC framework requires us to make statements where the simulators are chosen independently of the distinguisher (such as in (5.3)), to relate indifferentiability to AC we make use of its strong version described in Section 5.1.

Theorem 5.3 *Let $\mathbf{S}$, $\mathbf{T}$ be 1-interface resources and let $n \in \mathbb{N}$. If $\hat{\mathbf{S}}_2 \xrightarrow[\text{si}]{\varepsilon} \hat{\mathbf{T}}_2$ for a class of converters Σ and distinguishers $\mathcal{D}$, and both the protocol π and the simulator σ used in this reduction are stateless, then we have $\hat{\mathbf{S}}_n \xrightarrow[\text{AC}]{\varepsilon} \hat{\mathbf{T}}_n$ for the class of converters Σ and any class of distinguishers $\mathcal{D}'$ such that $\mathsf{Proj}_{\mathcal{P}}(\mathcal{D}') \subseteq \mathcal{D}$ for all $\mathcal{P} \subseteq \{1, \ldots, n\}$.*

Proof: To obtain the n-tuples of protocols and simulators (denoted $\hat{\pi}$ and $\hat{\sigma}$, respectively) required to prove the AC statement, one can simply use n independent copies of the protocol π and the simulator σ, respectively. Let us consider a distinguisher $\mathbf{D} \in \mathcal{D}$ for some fixed set $P \subseteq \{1, \ldots, n\}$ interacting with either $\hat{\pi}_{\mathcal{P}}\hat{\mathbf{S}}_n$ or $\hat{\sigma}_{\overline{\mathcal{P}}}\hat{\mathbf{T}}_n$ and relate it to the distinguisher $\mathsf{Proj}_{\mathcal{P}}(\mathbf{D})$ interacting with either $\pi\hat{\mathbf{S}}_2$ or $\hat{\mathbf{T}}_2\sigma$. Since both π and σ are stateless, clearly whenever $\mathbf{D}$ issues a query to an interface in $\mathcal{P}$, the distribution of the answer will be the same as if $\mathsf{Proj}_{\mathcal{P}}(\mathbf{D})$ issued the same query to the private interface. Analogously, any query issued by $\mathbf{D}$ to an interface in $\overline{\mathcal{P}}$ corresponds in the same way to a query issued by $\mathsf{Proj}_{\mathcal{P}}(\mathbf{D})$ to the public interface. Hence $\Delta^{\mathbf{D}}(\hat{\pi}_{\mathcal{P}}\hat{\mathbf{S}}_n, \hat{\sigma}_{\overline{\mathcal{P}}}\hat{\mathbf{T}}_n) = \Delta^{\mathsf{Proj}_{\mathcal{P}}(\mathbf{D})}(\pi\hat{\mathbf{S}}_2, \hat{\mathbf{T}}_2\sigma)$ and the theorem follows. ∎

5.4.3 Impossibility of Domain Extension

Let us now consider the specific case of the domain extension for random functions in the n-party case[17] (i.e., the reduction $\hat{\mathbf{R}}_n^{m,r} \xrightarrow[\mathrm{AC}]{\varepsilon} \hat{\mathbf{R}}_n^{\ell,r}$ with $\ell > m$). In this case using inherently stateful simulators σ_i would also lead to inconsistencies, for the same reason as described for the protocols π_i. Note that we cannot claim that such a reduction cannot be achieved using a stateful simulator, since its stateful behavior might not manifest in the distinguishing experiment. However, any such stateful simulator could be replaced by a stateless one without significant impact, as formalized in Lemma 5.2 below. Later we observe that the simulators cannot be stateless (for the same reason as in the indifferentiability case), leading to the impossibility result.

For the statement of Lemma 5.2, we will assume that the set of distinguishers $\mathcal{D}$ satisfies a simple closure property. For any $\mathbf{D} \in \mathcal{D}$ asking queries only to interfaces 1 and 2 let us consider a derived distinguisher $\mathbf{D}_{(i)}$ that proceeds in the same way as $\mathbf{D}$ but it also counts its queries to interface 2 and as soon as its i-th such query occurs, it asks the same query also to interface 3. At the end, $\mathbf{D}_{(i)}$ will output 1 if and only if the response to its i-th query to interface 2 was distinct from the response to the same query to interface 3. We assume $\mathbf{D}_{(i)} \in \mathcal{D}$ for all $\mathbf{D} \in \mathcal{D}$ and all $1 \leq i \leq q$ where q is an upper bound on the number of $\mathbf{D}$'s queries to interface 2.

[17] In the rest of the section we will use symbols such as $\mathbf{R}^{m,r}$ to refer to the single-interface resource and use the introduced notation to explicitly state the number of interfaces we want to consider (e.g., $\hat{\mathbf{R}}_n^{m,r}$).

Lemma 5.2 *Consider some fixed $n \geq 3$, $\ell > m$ and some fixed sets of converters Σ and distinguishers $\mathcal{D}$ satisfying the property given above. Assume that there exists a reduction $\hat{\mathbf{R}}_n^{m,r} \xrightarrow[\text{AC}]{\varepsilon} \hat{\mathbf{R}}_n^{\ell,r}$ via a tuple of protocols $\hat{\pi} = (\pi_1, \ldots, \pi_n)$ and simulators $\hat{\sigma} = (\sigma_1, \ldots, \sigma_n)$. Then there also exists a tuple of simulators $\hat{\sigma}' = (\sigma_1, \sigma_2', \sigma_3, \ldots, \sigma_n)$ such that σ_2' is stateless and for every distinguisher $\mathbf{D} \in \mathcal{D}$ accessing only interfaces 1 and 2 we have $\Delta^{\mathbf{D}}(\hat{\pi}_{\{1\}}\hat{\mathbf{R}}_n^{m,r}, \hat{\sigma}'_{\{1\}}\hat{\mathbf{R}}_n^{\ell,r}) \leq (q+1)\varepsilon$ where q is an upper bound on the number of its queries to interface 2.*

Proof: Let us consider the case in equation (5.3) where $\mathcal{P} = \{1\}$, hence any distinguisher would in the ideal experiment interact with a local simulator σ_i on each interface $i > 1$. Let $\mathbf{D}$ be a distinguisher that only asks queries to interfaces 1 and 2. Informally, we show that if we replace σ_2 by a simulator σ_2' that treats each query in the same way as the simulator σ_3 would treat its *first* query, this change will most likely remain unnoticed by any such $\mathbf{D}$. Moreover, since the behavior of this σ_2' on a particular query does not depend on any previous interaction, it is stateless in the sense of Definition 5.1. Note that we do not assume that this σ_2' belongs to the set Σ of converters considered for our reductions, but this will not be required by the further use of Lemma 5.2.

For any $i \in \{1, \ldots, q\}$ let us consider the derived distinguisher $\mathbf{D}_{(i)}$ described above. It will never output 1 in the real experiment, since there both interfaces 2 and 3 provide access to the same function. Therefore, the advantage achieved by $\mathbf{D}_{(i)}$ is equal to the probability that the two responses it compares in the ideal experiment are distinct. Since $\mathbf{D}_{(i)} \in \mathcal{D}$ we know that its advantage is at most ε, hence in the ideal experiment this inconsistency can occur with probability at most ε. This means that for each $i \in \{1, \ldots, q\}$, if the i-th response of σ_2 was replaced by a response generated by σ_2' instead, the outcome of the whole distinguishing experiment would change with probability at most ε. Applying the union bound, we get that if we replace *all* responses of σ_2 by those of σ_2', the transcript of the whole distinguishing experiment will change with probability at most $q\varepsilon$. Finally, since the advantage achieved by $\mathbf{D}$ before this change was at most ε, it will be at most $(q+1)\varepsilon$ afterwards. ∎

Let us now denote by $\hat{\mathbf{D}}$ the distinguisher given in Fig. 5.2 (implicitly parametrized by a converter $\pi \in \Sigma$) modified into the n-interface setting as follows: it uses interface 1 for all its (originally) private-interface queries, while using interface 2 for all public-interface queries. If $\hat{\mathbf{D}} \in \mathcal{D}$ then the upper bound given in Lemma 5.2 applies

to $\Delta^{\hat{\mathbf{D}}}(\hat{\pi}_{\{1\}}\hat{\mathbf{R}}_n^{m,r}, \hat{\sigma}'_{\overline{\{1\}}}\hat{\mathbf{R}}_n^{\ell,r})$. On the other hand, since $\ell > m$ and σ'_2 uses no memory, following the proof of Theorem 5.2 we also obtain that $\Delta^{\hat{\mathbf{D}}}(\hat{\pi}_{\{1\}}\hat{\mathbf{R}}_n^{m,r}, \hat{\sigma}'_{\overline{\{1\}}}\hat{\mathbf{R}}_n^{\ell,r}) > 0.04$. Combining these observations we get the following corollary.

Corollary 5.4 *Consider some fixed $n \geq 3$, $r \geq 2$, $\ell > m$ and sets of converters Σ and distinguishers $\mathcal{D}$ satisfying the properties required in Lemma 5.2 and additionally such that for each $\pi \in \Sigma$ the respective $\hat{\mathbf{D}}$ is in $\mathcal{D}$. If there exists a reduction $\hat{\mathbf{R}}_n^{m,r} \xrightarrow[\text{AC}]{\varepsilon} \hat{\mathbf{R}}_n^{\ell,r}$ via a tuple of protocols $\hat{\pi} = (\pi_1, \ldots, \pi_n)$ then $\varepsilon > 0.04/(p+1)$ where p is an upper bound on the number of $\{0,1\}^m$-queries the protocol π_1 used for this reduction needs to evaluate on one $\{0,1\}^\ell$-input.*

Discussion. By the above result it is impossible to extend the domain of a public random function even by a single bit in a multi-party environment where the parties must be modeled as possibly having conflicting goals or deviating from the protocol in an uncoordinated manner (or, technically speaking, in any scenario where a proper modeling requires the use of local simulators). This is in contrast to the two-party indifferentiability setting (with several constructions that achieve this transformation) where one implicitly makes the assumption that all dishonest parties are coordinated by a hypothetical central adversary. This seems to be a very strong assumption in particular for random oracles that are typically thought of as being used by many different parties in many different applications. Of course, a particular use of a construction proven secure in the 2-party scenario within a multi-party setting as discussed above might still be secure under some additional assumptions, however our result indicates that such use should always be explicitly justified.

Chapter 6

Concluding Remarks

Throughout the thesis, we have investigated the security provided by constructions in various sub-areas of symmetric cryptography. We conclude by emphasizing some of the outcomes of our study and mentioning several remaining open questions in each of the investigated areas.

For the topic of indistinguishability amplification, a natural question is whether our results from Chapter 3 have any counterpart in the computational world and how would it be positioned within the existing literature on computational indistinguishability amplification. Although it is not difficult to define the concept of free-start distinguishing in the computational setting, our main result does not translate to this setting. This is because such a translation would imply that for example composition of non-adaptively secure pseudo-random permutations is adaptively secure, which would contradict the results in [Pie05] under standard assumptions. Nevertheless, our insights into the structure of neutralizing constructions might still be of some use in the computational setting, which we leave as an open question.

Our investigation of key-length extending constructions in Chapter 4 suggests that, generally speaking, extending the cascade construction by the computationally very cheap key-whitening steps might pay off very well in terms of security increase. In particular, if it wasn't for legacy reasons, the 2-XOR-cascade construction using the same key for both randomization steps would seem to represent a more efficient replacement for the widely used triple cascade, maintaining a comparable level of security. On the other hand, the landscape of presented results also

leaves several questions open. First of all, it would be interesting to know whether the plain cascade construction approaches the highest achievable security level in our model (up to roughly $2^{\kappa+n}$ queries) with increasing length as the XOR-cascade does, even if at a slower pace. Moreover, in the analysis of XOR-cascades we have considered the optimization achieved by key repetition in the whitening steps only in the special case of length 2, however similar optimizations might be possible also for longer XOR-cascades. One could investigate the existing results on key-alternating ciphers from this perspective. Finally, on the side of attacks, our results serve more as an indication of tightness of the achieved security results than as a guideline for practical attacks since with increasing ℓ our measure of attack complexity becomes less realistic. It would therefore be valuable to study attacks on randomized cascades in a more realistic model, as was recently done for plain cascades in [DDKS12].

The results given in Chapter 5 present a general treatment of reductions in settings where a certain resource has to be modelled in detail, however we only illustrate it on the particular example of memory. One could also consider other resources such as randomness, hand in hand with identifying scenarios where such a precision might be necessary for their proper understanding. Also, the impossibility result for domain extension of a public random function in a general multi-party environment suggests that some basic and natural tasks and constructions might become problematic as soon as one does not assume the existence of a centralized adversary or has other reasons to consider local simulators. There are certainly more examples of this phenomenon and we find them worth a further study.

Bibliography

[3DE98] ANSI X9.52: Triple Data Encryption Algorithm Modes of Operation, 1998.

[3DE99] FIPS PUB 46-3: Data Encryption Standard (DES). National Institute of Standards and Technology, 1999.

[3DE04] Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher. National Institute of Standards and Technology, Special Publication 800-67, 2004.

[ABCV98] William Aiello, Mihir Bellare, Giovanni Di Crescenzo, and Ramarathnam Venkatesan. Security amplification by composition: The case of doubly-iterated, ideal ciphers. In *Advances in Cryptology — CRYPTO '98*, volume 1462 of *LNCS*, pages 390–407. Springer Berlin Heidelberg, 1998.

[Aes01] Advanced encryption standard. In *FIPS PUB 197, Federal Information Processing Standards Publication*, 2001.

[AKL+09] Joël Alwen, Jonathan Katz, Yehuda Lindell, Giuseppe Persiano, Abhi Shelat, and Ivan Visconti. Collusion-free multiparty computation in the mediated model. In Shai Halevi, editor, *Advances in Cryptology — CRYPTO 2009*, volume 5677 of *LNCS*, pages 524–540. Springer Berlin Heidelberg, 2009.

[AKMZ12] Joël Alwen, Jonathan Katz, Ueli Maurer, and Vassilis Zikas. Collusion-preserving computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology — CRYPTO 2012*, volume 7417 of *LNCS*, pages 124–143. Springer Berlin Heidelberg, 2012.

[AMP10] Elena Andreeva, Bart Mennink, and Bart Preneel. On the Indifferentiability of the Grostl Hash Function. In Juan Garay and Roberto De Prisco, editors, *Security and Cryptography for Networks*, volume 6280 of *LNCS*, pages 88–105. Springer Berlin Heidelberg, 2010.

[BDJR97] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *FOCS '97: Proceedings of the 38th IEEE Annual Symposium on Foundations of Computer Science*, pages 394–403, 1997.

[BDPVA08a] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak specifications. Submission to NIST (Round 1), 2008.

[BDPVA08b] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In Nigel Smart, editor, *Advances in Cryptology — EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 181–197. Springer Berlin Heidelberg, 2008.

[BK03] Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In *Advances in Cryptology — EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 491–506. Springer Berlin Heidelberg, 2003.

[BKL+12] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Francois-Xavier Standaert, John Steinberger, and Elmar Tischhauser. Key-alternating ciphers in a provable setting: encryption using a small number of public permutations. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology — EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 45–62. Springer Berlin Heidelberg, 2012.

[BKR94] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining message authentication code. In *CRYPTO '94: Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, LNCS, pages 341–358. Springer Berlin Heidelberg, 1994.

[Bla05] John Black. The ideal-cipher model, revisited: An uninstantiable blockcipher-based hash function. Cryptology ePrint

Archive, Report 2005/210, 2005. `http://eprint.iacr.org/`.

[BPW04] Michael Backes, Birgit Pfitzmann, and Michael Waidner. A general composition theorem for secure reactive systems. In Moni Naor, editor, *Theory of Cryptography — TCC 2004*, volume 2951 of *LNCS*, pages 336–354. Springer Berlin Heidelberg, 2004.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[BR95] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo Santis, editor, *Advances in Cryptology — EUROCRYPT '94*, volume 950 of *LNCS*, pages 92–111. Springer Berlin Heidelberg, 1995.

[BR96] Mihir Bellare and Phillip Rogaway. The Exact Security of Digital Signatures – How to Sign with RSA and Rabin. In Ueli Maurer, editor, *Advances in Cryptology — EUROCRYPT '96*, volume 1070 of *LNCS*, pages 399–416. Springer Berlin Heidelberg, 1996.

[BR02] John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 384–397. Springer Berlin Heidelberg, 2002.

[BR06] Mihir Bellare and Phillip Rogaway. Code-based game-playing proofs and the security of triple encryption. In *Advances in Cryptology — EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer Berlin Heidelberg, 2006. Full version at `http://eprint.iacr.org/2004/331`.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, volume 2442 of *LNCS*, pages 320–335. Springer Berlin Heidelberg, 2002.

[BT94] Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections. In *STOC '94: Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 544–553, New York, NY, USA, 1994. ACM.

[Can01] Ron Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *FOCS '01: Proceedings of the 42nd IEEE Annual Symposium on Foundations of Computer Science*, pages 136–145. IEEE Computer Society Press, Oct. 2001. Full version at `http://eprint.iacr.org/2000/067`.

[CDMP05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård Revisited: How to Construct a Hash Function. In Victor Shoup, editor, *Advances in Cryptology — CRYPTO 2005*, volume 3621 of *LNCS*, pages 430–448. Springer Berlin Heidelberg, 2005.

[CG96] Ran Canetti and Rosario Gennaro. Incoercible multiparty computation. In *FOCS '96: Proceedings of the 37th IEEE Annual Symposium on Foundations of Computer Science*, pages 504–513. IEEE Computer Society, 1996.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In *STOC '98: Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 209–218. ACM, 1998.

[CN08] Donghoon Chang and Mridul Nandi. Improved Indifferentiability Security Analysis of chopMD Hash Function. In Kaisa Nyberg, editor, *Fast Software Encryption*, volume 5086 of *LNCS*, pages 429–443. Springer Berlin Heidelberg, 2008.

[CPS08] Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In *Advances in Cryptology — CRYPTO 2008*, volume 5157 of *LNCS*, pages 1–20. Springer Berlin Heidelberg, 2008.

[CV12] Ran Canetti and Margarita Vald. Universally composable security with local adversaries. In Ivan Visconti and Roberto De Prisco, editors, *SCN*, volume 7485 of *LNCS*, pages 281–301. Springer Berlin Heidelberg, 2012.

[DDKS12] Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. Efficient dissection of composite problems, with applications to cryptanalysis, knapsacks, and combinatorial search problems. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology — CRYPTO 2012*, volume 7417 of *LNCS*, pages 719–740. Springer Berlin Heidelberg, 2012.

[Des77] Data encryption standard. In *In FIPS PUB 46, Federal Information Processing Standards Publication*, 1977.

[DGHM13] Grégory Demay, Peter Gaži, Martin Hirt, and Ueli Maurer. Resource-restricted indifferentiability. In *Advances in Cryptology — EUROCRYPT 2013*, LNCS. Springer Berlin Heidelberg, 2013. To appear.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[DH77] W. Diffie and M. E. Hellman. Exhaustive Cryptanalysis of the NBS Data Encryption Standard. *Computer*, 10(6):74–84, 1977.

[DIJK09] Yevgeniy Dodis, Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. Security amplification for interactive cryptographic primitives. In *Theory of Cryptography — TCC 2009*, volume 5444 of *LNCS*, pages 128–145. Springer Berlin Heidelberg, 2009.

[DRRS09] Yevgeniy Dodis, Leonid Reyzin, Ronald Rivest, and Emily Shen. Indifferentiability of Permutation-Based Compression Functions and Tree-Based Modes of Operation, with Applications to MD6. In Orr Dunkelman, editor, *Fast Software Encryption*, volume 5665 of *LNCS*, pages 104–121. Springer Berlin Heidelberg, 2009.

[DRS09] Yevgeniy Dodis, Thomas Ristenpart, and Thomas Shrimpton. Salvaging Merkle-Damgård for Practical Applications. In Antoine Joux, editor, *Advances in Cryptology — EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 371–388. Springer Berlin Heidelberg, 2009.

[DRST12] Yevgeniy Dodis, Thomas Ristenpart, John Steinberger, and Stefano Tessaro. To Hash or Not to Hash Again?

(In)Differentiability Results for H 2 and HMAC. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology — CRYPTO 2012*, volume 7417 of *LNCS*, pages 348–366. Springer Berlin Heidelberg, 2012.

[EG85] S. Even and O. Goldreich. On the power of cascade ciphers. *ACM Trans. Comput. Syst.*, 3(2):108–116, 1985.

[EM91] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. In *Journal of Cryptology*, pages 151–161. Springer Berlin Heidelberg, 1991.

[EMV08] *EMV Integrated Circuit Card Specification for Payment Systems, Book 2: Security and Key Management, v.4.2*. June 2008.

[Fan61] Robert Fano. *Transmission of Information: A Statistical Theory of Communications*. The MIT Press, Cambridge, MA, 1961.

[FOPS01] Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. RSA-OAEP Is Secure under the RSA Assumption. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, volume 2139 of *LNCS*, pages 260–274. Springer Berlin Heidelberg, 2001.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology — CRYPTO '86*, volume 263 of *LNCS*, pages 186–194, 1986.

[Gaž10] Peter Gaži. *Methods in Provable Security*. PhD thesis, Comenius University Bratislava, September 2010.

[GM09] Peter Gaži and Ueli Maurer. Cascade encryption revisited. In M. Matsui, editor, *Advances in Cryptology — ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 37–51. Springer Berlin Heidelberg, December 2009.

[GM10] Peter Gaži and Ueli Maurer. Free-start distinguishing: Combining two types of indistinguishability amplification. In K. Kurosawa, editor, *The 4th International Conference on Information Theoretic Security - ICITS 2009*, volume 5973 of *LNCS*, pages 28–44. Springer Berlin Heidelberg, 2010.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GT12] Peter Gaži and Stefano Tessaro. Efficient and optimally secure key-length extension for block ciphers via randomized cascading. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology — EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 63–80. Springer Berlin Heidelberg, 2012.

[HKT11] Thomas Holenstein, Robin Künzler, and Stefano Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11: Proceedings of the 43rd Annual ACM Symposium on Theory of Computing, pages 89–98, New York, NY, USA, 2011. ACM.

[HT04] Joseph Halpern and Vanessa Teague. Rational secret sharing and multiparty computation. In *STOC '04: Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 623–632, New York, NY, USA, 2004. ACM.

[HU05] Dennis Hofheinz and Dominique Unruh. Comparing two notions of simulatability. In Joe Kilian, editor, *Theory of Cryptography — TCC 2005*, volume 3378 of *LNCS*, pages 86–103. Springer Berlin Heidelberg, 2005.

[KR01] Joe Kilian and Phillip Rogaway. How to Protect DES Against Exhaustive Key Search (an Analysis of DESX). *Journal of Cryptology*, 14:17–35, 2001.

[LM90] Xuejia Lai and James L. Massey. A proposal for a new block encryption standard. In *Advances in Cryptology — EUROCRYPT '90*, LNCS, pages 389–404. Springer Berlin Heidelberg, 1990.

[LMs05] Matt Lepinksi, Silvio Micali, and abhi shelat. Collusion-free protocols. In *STOC '05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 543–552, New York, NY, USA, 2005. ACM.

[LPS12] Rodolphe Lampe, Jacques Patarin, and Yannick Seurin. An Asymptotically Tight Security Analysis of the Iterated

Even-Mansour Cipher. To appear at ASIACRYPT 2012, 2012.

[LR86] M Luby and C Rackoff. Pseudo-random permutation generators and cryptographic composition. In *STOC '86: Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 356–363, New York, NY, USA, 1986. ACM.

[Luc98] Stefan Lucks. Attacking triple encryption. In Serge Vaudenay, editor, *Fast Software Encryption*, volume 1372 of *LNCS*, pages 239–253. Springer Berlin Heidelberg, 1998.

[Mau02] Ueli Maurer. Indistinguishability of random systems. In Lars Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 110–132. Springer Berlin Heidelberg, May 2002.

[Mau11] Ueli Maurer. Constructive cryptography – a new paradigm for security definitions and proofs. In S. Moedersheim and C. Palamidessi, editors, *Theory of Security and Applications (TOSCA 2011)*, volume 6993 of *LNCS*, pages 33–56. Springer Berlin Heidelberg, April 2011.

[MM93] Ueli Maurer and James L. Massey. Cascade ciphers: The importance of being first. *Journal of Cryptology*, 6(1):55–61, 1993.

[MOPS06] Ueli Maurer, Yvonne Anne Oswald, Krzysztof Pietrzak, and Johan Sjödin. Luby-Rackoff ciphers from weak round functions? In *Advances in Cryptology — EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 391–408. Springer Berlin Heidelberg, May 2006.

[MP04] Ueli Maurer and Krzysztof Pietrzak. Composition of random systems: When two weak make one strong. In Moni Naor, editor, *Theory of Cryptography — TCC 2004*, volume 2951 of *LNCS*, pages 410–427. Springer Berlin Heidelberg, February 2004.

[MPR07] Ueli Maurer, Krzysztof Pietrzak, and Renato Renner. Indistinguishability amplification. In Alfred Menezes, editor, *Advances in Cryptology — CRYPTO 2007*, volume 4622 of

LNCS, pages 130–149. Springer Berlin Heidelberg, August 2007.

[MR11] Ueli Maurer and Renato Renner. Abstract cryptography. In Bernard Chazelle, editor, *The Second Symposium on Innovations in Computer Science ICS 2011*, pages 1–21. Tsinghua University Press, January 2011.

[MRH04] Ueli Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *Theory of Cryptography — TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer Berlin Heidelberg, February 2004.

[MT09] Ueli Maurer and Stefano Tessaro. Computational indistinguishability amplification: Tight product theorems for system composition. In Shai Halevi, editor, *Advances in Cryptology — CRYPTO 2009*, volume 5677 of *LNCS*, pages 350–368. Springer Berlin Heidelberg, August 2009.

[Mye03] Steven Myers. Efficient amplification of the security of weak pseudo-random function generators. *Journal of Cryptology*, 16(1):1–24, 2003.

[Pie05] Krzysztof Pietrzak. Composition does not imply adaptive security. In *Advances in Cryptology — CRYPTO 2005*, volume 3621 of *LNCS*, pages 55–65. Springer Berlin Heidelberg, August 2005.

[RSS11] Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In Kenneth Paterson, editor, *Advances in Cryptology — EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 487–506. Springer Berlin Heidelberg, 2011.

[Sch91] C.P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4:161–174, 1991.

[Sch94] Bruce Schneier. Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish). In *Fast Software Encryption – FSE '93*, volume 809 of *LNCS*, pages 191–204. Springer Berlin Heidelberg, 1994.

[Sha48] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 625–56, 1948.

[Ste12] John Steinberger. Improved Security Bounds for Key-Alternating Ciphers via Hellinger Distance. Cryptology ePrint Archive, Report 2012/481, 2012. `http://eprint.iacr.org/`.

[Tes11] Stefano Tessaro. Security amplification for the cascade of arbitrarily weak PRPs: Tight bounds via the interactive Hardcore Lemma. In *Theory of Cryptography — TCC 2011*, volume 6597 of *LNCS*, pages 37–54. Springer Berlin Heidelberg, 2011.

[Vau00] Serge Vaudenay. Adaptive-attack norm for decorrelation and super-pseudorandomness. In *SAC '99: Proceedings of the 6th Annual International Workshop on Selected Areas in Cryptography*, LNCS, pages 49–61. Springer Berlin Heidelberg, 2000.

[Vau03] Serge Vaudenay. Decorrelation: a theory for block cipher security. *Journal of Cryptology*, 16(4):249–286, 2003.