Diss. ETH No. 17054

Weak Pseudorandomness and Unpredictability

A dissertation submitted to

ETH ZURICH

for the degree of
Doctor of Sciences

presented by

ULF JOHAN SJÖDIN
MSc. in Computer Science and Engineering KTH

born 22.12.1977
citizen of Sweden

accepted on the recommendation of

Prof. Dr. Ueli Maurer, examiner
Prof. Dr. Jesper Buus Nielsen, co-examiner

2007

to my parents

Acknowledgments

The research leading to this thesis was carried out under the supervision of professor Ueli Maurer. I would like to thank him for introducing me to the fascinating world of cryptography and for all competent guidance that I have received during the last years. I am also very grateful to professor Jesper Buus Nielsen for co-refereeing this thesis.

It was a great pleasure to do research with my co-authors Thomas Holenstein and Krzysztof Pietrzak. Many thanks are also due to my office mates Robert König, Bartosz Przydatek, and Stefano Tessaro (although I never got to learn any Japanese, Polish, or Italian), and to all other colleagues from the *Information Security and Cryptography* research group at ETH Zurich (including past members). A special thanks goes to Renato Renner for the many nice – but sometimes tough – jogging tours at the mountain sides of Zurich. Actually, his family name translates to "runner" (*nomen est omen*).

Finally, I thank my family and friends for their support throughout the years. You are great.

This research was partially supported by the Zurich Information Security Center (ZISC).

Abstract

To base the security of practical cryptographic schemes on weakened assumptions (which are hence more likely to hold) and to improve their efficiency are general research goals in cryptography. In this thesis we continue this quest. We focus on the most traditional problems in cryptography, namely that of assuring *privacy* and *authenticity* of data in the *symmetric* setting (where both the sender and the receiver share a secret key).

We study the Feistel-network which is a popular structure underlying many block-ciphers – e.g. DES – where the cipher is constructed from many simpler rounds, each defined by some function. In particular, we investigate the security of the Feistel-network against *chosen-plaintext-attack* (CPA) distinguishers when the only security guarantee we have for the round functions is that they are secure against *non-adaptive chosen-plaintext attacks* (nCPA). Thus the round functions have a strictly weaker security guarantee than what we would like to achieve for the whole construction. We show that in the information-theoretic setting, four rounds with nCPA-secure functions are enough and necessary to get a CPA-secure permutation. We also prove that this result unfortunately does not translate into the more practically relevant pseudorandom setting.

Further, we focus on *weak* pseudorandom functions (WPRFs), defined similarly to pseudorandom functions (PRFs) but where the distinguisher only gets to see the outputs on random inputs (and not on inputs of its choice). We propose a *chosen-ciphertext-attack* secure encryption scheme, based on any WPRF, that is superior to all previous proposed schemes given in the literature (in terms of key-material and applications of the WPRF). This is achieved by an efficient strengthening of any WPRF to a PRF and by a range-extension method for WPRFs that is optimal within a large and natural class of range extensions (especially all known today).

We also introduce a general paradigm for domain extension of *message authentication codes* and an essentially optimal extension for practical use.

Zusammenfassung

Die Sicherheit kryptographischer Verfahren auf schwächere Annahmen abzustützen (welche dann plausibler sind) und die Verbesserung der Effizienz derer sind zentrale Ziele der kryptographischen Forschung, die wir auch in dieser Dissertation verfolgen werden. Wir konzentrieren uns dabei auf die traditionellen kryptographischen Probleme der *Authentisierung* und *Verschlüsselung* von Daten im *symmetrischen* Fall (in welchem Sender und Empfänger einen gemeinsamen Schlüssel besitzen).

Genauer untersuchen wir Feistelnetzwerke. Dabei handelt es sich um eine Struktur, der zahlreiche Blockchiffren wie z.B. DES zugrunde liegen. Diese Blockchiffren sind aus mehreren einfacheren Runden aufgebaut, welche jeweils durch eine Funktion definiert sind. Insbesondere betrachten wir die Sicherheit von Feistelnetzwerken gegen *Chosen-Plaintext* Attacken (CPA) für den Fall, in welchem die Rundenfunktionen nur die Sicherheit gegen nicht-adaptive *Chosen-Plaintext* Attacken (nCPA) garantieren. Folglich bieten die Rundenfunktionen deutlich schwächere Sicherheitsgarantien als wir für die Gesamtkonstruktion erreichen möchten. Wir zeigen, dass informationstheoretisch vier Runden mit nCPA-sicheren Funktionen notwendig und hinreichend sind, um eine CPA-sichere Permutation zu erhalten. Wir beweisen zudem, dass sich dieses Resultat leider nicht auf das praktisch relevantere pseudozufällige Szenario übertragen lässt.

Weiterhin beschäftigen wir uns mit so genannten *Weak Pseudorandom Functions* (WPRFs), welche ähnlich wie gewöhnliche *Pseudorandom Functions* (PRFs) mittels eines Unterscheiders definiert sind. Der Unterscheider bekommt aber nur Ausgaben auf zufällige Eingaben (statt Ausgaben auf vom Unterscheider gewählte Eingaben) zur Verfügung gestellt. Basierend auf einer beliebigen WPRF schlagen wir ein Verschlüsselungsverfahren vor, das gegen *Chosen-Ciphertext* Attacken sicher ist und (was

die Schlüssellänge und die Anzahl der Aufrufe der WPRF betrifft) sämtliche aus der Fachliteratur bekannten Verfahren überlegen ist. Wir erreichen dies durch eine effiziente Stärkung einer beliebigen WPRF zu einer PRF und durch eine Erweiterung des Bildbereiches der WPRF, welche innerhalb einer grossen und natürlichen Klasse von Bildbereichserweiterungen (insbesondere innerhalb der Klasse aller heute bekannten Erweiterungen) optimal ist.

Zusätzlich führen wir ein allgemeines Paradigma für Definitionsbereichserweiterungen von *Message Authentication Codes* ein und geben eine optimale Erweiterung an, die sich zur praktischen Anwendung eignet.

Contents

Chapter 1

Introduction

1.1 Privacy and Authenticity

Consider a scenario where a sender (referred to as Alice) posts off a message in clear (say a postcard) to a receiver (Bob) via a courier (Eve). There is nothing that prevents Eve from reading and altering the postcard before it is handed over to Bob, i.e., violating the *privacy* and *authenticity*, respectively. How can Alice and Bob circumvent this? Throughout millennia humans have tried to give solutions to this question of which some have been more sophisticated and successful than others.[1] And although cryptography has been primarily concerned with studying these fundamental security goals for data transmissions and lots of progress have been made, it is still a hot research topic in cryptography.

The answer to the above question clearly depends on the set up. If Alice and Bob do not share any a priori knowledge, the task of assuring authenticity – over an *insecure channel* – turns out to be impossible as there is no way for Bob to distinguish Alice from Eve. Even worse, Shannon [Sha49] (and Maurer [Mau93] for the multi-message case) proved that even if the authenticity is assured, one can not achieve the privacy in the *information-theoretic setting* where Eve is computationally unbounded. However, if Alice and Bob share a secret key of the same length as the message, the latter can easily be achieved using the so-called *one-time*

[1] Mary Stuart was executed 1587 because one of her conspiracy messages ended up in (for her) wrong hands and could be decrypted.

pad, a well-known *symmetric* technique. Here, "one-time" refers to that the key is only to be used once, and "symmetric" refers to that Alice and Bob share the same key. This was a nice theoretical result but from a practical perspective not very appealing due to the large amount of key-material which indeed also is necessary in this setting [Sha49, Mau93].

In an other setting, called the *computational setting*, it is assumed that certain *cryptographic primitives* (or hard to solve puzzles) exist and also that Eve is *efficient*. Here there are several efficient symmetric schemes for assuring privacy and authenticity, where Alice and Bob only need to share a secret key of fixed length (say 256 bits). One can even show that if the authenticity is assured – over the insecure channel – the fixed number of secretly shared bits for assuring privacy can be 0 (see *public-key* cryptography [DH76, RSA78, Mer78]). But unfortunately, these so-called *asymmetric* schemes are rather inefficient and therefore typically only used for sharing a secret key for some more efficient symmetric scheme. Let us also stress that in lack of provable lower bounds no proof of existence has been found for these cryptographic primitives (although they are widely believed to exist) and therefore only candidates are in use today.

The goal of this thesis has been to base the security of symmetric cryptographic systems on as weak primitives as possible. After all, a candidate for a weakened primitive is more likely to satisfy the conjecture and is potentially much more efficiently implementable. In particular, *symmetric encryption schemes* and *message authentication codes* are considered.

1.1.1 Symmetric Encryption Schemes

A symmetric encryption scheme allows parties sharing a secret-key k to assure privacy of data over an insecure channel. The scheme consists of two efficiently computable algorithms, the randomized *encryption* algorithm and the deterministic *decryption* algorithm. The encryption algorithm takes the key k and a message m as input to produce a ciphertext c. The decryption algorithm undoes encryption, i.e., on input k and a ciphertext c it returns m. Informally, it should be infeasible for any efficient adversary seeing a ciphertext to gain any information about the message – or equivalently – to decide, given two distinct plaintexts (of equal length) and the encryption c^* of one of them, which plaintext that corresponds to c^*. An encryption scheme is referred to as secure under a *chosen-ciphertext attack* (CPA) if it is secure as above even when the adversary can issue encryptions of its choice and obtain the corresponding ciphertexts. And if it does not in addition help the adversary to issue decryptions of its choice and obtain the corresponding plaintexts (except for the ciphertext c^*), it is considered secure under a *chosen-ciphertext attack* (CCA).

1.1.2 Message Authentication Codes

A well-known technique for assuring authenticity of data over an insecure channel is to use a so-called *message authentication code* (MAC). A MAC is a keyed hash function that allows parties sharing a secret key k to authenticate messages as follows. The sender computes the tag value, i.e., the value of the MAC on the message he/she wants to send, and sends it together with the message. The receiver of a message-tag pair (m, τ) recomputes the tag value of m and accepts if and only if it equals τ. Informally, it should be infeasible for any efficient forger (not in possession of k) to come up with a new valid message-tag pair (m', τ'), i.e., for which the value of the MAC on m' equals τ'. This should be the case even if the forger may query the MAC (under the key k) as it wants and possibly dependent on previous queries. A MAC is referred to as existential unforgeable under a CPA.

1.2 Contributions and Outline

1.2.1 Relaxations of Luby-Rackoff Ciphers

The Feistel-network is a popular structure underlying many block-ciphers[2] where the cipher is constructed from many simpler rounds, each defined by some function which is derived from the secret key. Typically, a *pseudorandom function* (PRF) is used as round function. Informally, a PRF is a family of functions which is efficiently computable and where a random member from the family cannot be distinguished from a uniform random function by any efficient adversary that can query the function as it wants (possibly dependent on previous queries). Luby and Rackoff [LR86] showed that the three-round Feistel-network with PRFs as round functions is a *pseudorandom permutation* (PRP) (i.e., a PRF that

[2] A block-cipher is an efficiently computable random permutation that can be transformed to an encryption scheme by some mode of operation like Cipher Block Chaining (CBC) or Electronic Code Book (ECB) (see [MvOV97] for an overview).

is a permutation), thus giving some confidence in the soundness of using a Feistel-network to design block-ciphers. In order to achieve more efficient constructions of PRPs from PRFs, many researchers have investigated the security of the Luby-Rackoff ciphers with weakened primitives as round functions. All these relaxed constructions need at least some of the round functions to be PRFs in order to get a PRP.

In order to prove that some system – which is built from *pseudorandom* components – is pseudorandom itself, it is often enough to prove the corresponding statement in the information theoretic setting where the adversary and the functions are not necessarily efficient. To be more precise, a *quasirandom* function (QRF) (analogously for a quasirandom permutation (QRP)) is defined similar to a PRF but where one does not require the distinguisher or the function to be efficient, only the number of queries the distinguisher is allowed to make is bounded. And to prove the security of (for example) the original Luby-Rackoff construction, it turns out to be enough to prove that the three round Feistel-network with QRFs as round functions is a QRP [Mau02]. The security proof then follows from a simple hybrid argument.

In Chapter 3, we investigate the security of the Feistel-network when the only security guarantee we have for the round functions are that they are secure when queried non-adaptively, i.e., when all queries of the distinguisher are chosen in advance. Although this is still a strong requirement, this was the weakest natural type of attack that we could imagine which does not make the Feistel-network trivially insecure. For example it is too weak to assume that the round functions are secure when queried on random queries.[3] In the information-theoretic (or quasirandom) setting, four rounds with QRFs secure against non-adaptive queries turns out to be sufficient (and necessary) to get a QRP. We also prove that this result (unfortunately) does not translate to the more practically relevant pseudorandom setting. This gives an other example of the phenomena that certain constructions imply quasirandomness but not pseudorandomness (see also [MP04, MPR06, Pie05]).

Further, we propose relaxations of the Luby-Rackoff cipher which, in particular, answers an open problem posed by Minematsu and Tsunoo [MT05]. The results of Chapter 3 appeared in [MOPS06a, MOPS06b].

[3]Just consider a function $\mathbf{F}$ which satisfies $\mathbf{F}(0\ldots0) = 0\ldots0$ but otherwise looks random. This $\mathbf{F}$ is secure (against random queries) as a random query is unlikely to be the all zero string. But a Feistel-network build from such functions will output $0\ldots0$ on input $0\ldots0$ and thus is easily seen not to be secure.

1.2.2 Encryption based on Weak Pseudorandomness

The problem of constructing CCA-secure symmetric encryption schemes based on any PRF has been studied extensively in the literature and several efficient and provably secure constructions have been proposed (for an overview see [Gol04]). In [NR98], Naor and Reingold posed the natural and far less studied question whether such a scheme can be efficiently constructed from a weak PRF, i.e., a PRF which is secure when queried on random inputs.[4] An example of a weak PRF is any block cipher that is secure when queried on random inputs, but it can also be derived from trapdoor one-way permutations as done in [NR99b]. But as weak PRFs can have rather strong structural properties, e.g. they can commute (i.e., $\mathsf{F}_k(\mathsf{F}_{k'}(x)) = \mathsf{F}_{k'}(\mathsf{F}_k(x))$), be self inverse (i.e., $\mathsf{F}_k(\mathsf{F}_k(x)) = x$), have small fractions of fixed points (e.g. $\mathsf{F}_k(0\ldots0) = 0\ldots0$), and have related outputs (e.g. $\mathsf{F}_k(x\|0) = F_k(x\|1)$ for all x), encryption schemes based on a PRF generally become totally insecure if the PRF is simply replaced by a weak PRF (see [DN02]).

In an elegant work [DN02], Damgård and Nielsen proposed an efficient and provably CPA-secure symmetric encryption scheme from any weak PRF. The main ingredient of the construction is a method for range-extension of any weak PRF. They also show (using well-known techniques) how their scheme can be made CCA-secure. Their open question whether this can be done more efficiently has been the main motivation for this work.

Our results, presented in Chapter 4, are the following. First, we optimize Damgård and Nielsen's CPA-secure encryption scheme by constructing a more efficient range-extension method for weak PRFs. Our method is optimal within a large and natural class of extensions (especially all extensions that are known today). Second, we propose an efficient construction of a (regular) PRF from any weak PRF. Third, we show that these two results indeed imply a CCA-secure encryption scheme, based on any weak PRF, that is significantly more efficient than the CCA-secure scheme of Damgård and Nielsen (especially for long messages). The results of Chapter 4 can also be found in [MS07].

[4] Of course the security could be based on an even weaker primitive like any one-way function (OWF) [HILL99, GGM86]. However, such schemes are not of practical interest due to their inefficiency.

1.2.3 Domain-Extended Message Authentication Codes

An important parameter of a MAC is the message space $\mathcal{M}$. A MAC which has $\mathcal{M} = \{0,1\}^L$ for a constant L is referred to as having *fixed input length*. In most applications, however, one needs to authenticate messages of potentially *arbitrary input length*, i.e., $\mathcal{M} = \{0,1\}^*$.

In the context of constructing arbitrary-input-length MACs, domain extensions of PRFs (like CBC [BKR00]) are widely used although the security of the resulting MAC relies on the stronger PRF primitive (a PRF is a MAC, but a MAC is not necessarily a PRF). A much more natural and cautious approach, first studied by An and Bellare in [AB99], is how to extend the domain of MACs, i.e., how to construct arbitrary-input-length MACs from any fixed-input-length MAC.[5]

In Chapter 5, we investigate a general paradigm for domain extension of MACs, and give a simple and general security proof technique, applicable to a very general type of extensions. We propose a concrete, essentially optimal extension for practical use and prove its security. Our extension is superior to the best previously known extension proposed by An and Bellare [AB99]: only one rather than two secret keys is required, the efficiency is improved, and the domain is extended to arbitrary input length. The results of Chapter 5 appeared in [MS05a, MS05b].

[5]Domain extension has been studied for many other cryptographic primitives such as collision resistant hash functions [Dam89, Mer90], PRFs [BGR95, BKR00, PR00, Mau02], universal one-way hash functions [BR97, Sho00], and random oracles [CDMP05, BR06].

Chapter 2

Preliminaries

2.1 Notation

If M is a set, $\#M$ denotes its cardinality. For a sequence S of elements, $|S|$ denotes its length and S_i the sequence of its first i elements. For $N > 0$ let $\{0,1\}^{\leq N} \stackrel{\text{def}}{=} \cup_{i=1}^{N}\{0,1\}^i$. For $x,\, y \in \{0,1\}^*$, let $|x|$ denote the length of x (in bits), $x\|y$ the concatenation of x and y, $\langle n \rangle_b$ a b-bit encoding of a positive integer $n \leq 2^b$, $x[i]$ the i-th bit of x, and

$$x[i,j] \stackrel{\text{def}}{=} x[i]\big\|x[i+1]\big\|\cdots\big\|x[j]$$

for $1 \leq i \leq j \leq |x|$. For $x \in \{0,1\}^{2n}$ we denote with ${}_Lx$ and ${}_Rx$ the left and right half of x respectively, so $x = {}_Lx\|{}_Rx$. Similarly for a function f with range $\{0,1\}^{2n}$ (for some $n > 0$), we let ${}_Lf$ (${}_Rf$) denote the function one gets by ignoring the right (left) half of the output of f. A function $\mathcal{X} \to \mathcal{Y}$ is referred to as having fixed input length (FIL) if $\mathcal{X} = \{0,1\}^\ell$ for some ℓ, variable input length (VIL) if $\mathcal{X} = \{0,1\}^{\leq N}$ for some N, and arbitrary input length (AIL) if $\mathcal{X} = \{0,1\}^*$. A variable-output-length (VOL) function has an extra input specifying the length of the output. To be precise, a function $f : \mathcal{X} \times \mathbb{N} \to \{0,1\}^*$ has VOL if for all x and l

$$|f(x,l)| = l \quad \text{and} \quad f(x,l) = f(x,l+1)[1,l].$$

A quantity $\varepsilon(\gamma)$ is called negligible in γ if, for all $c > 0$, there is a constant γ_0 such that for all $\gamma > \gamma_0$ it holds that $\varepsilon(\gamma) < 1/\gamma^c$.

We use capital calligraphic letters like $\mathcal{X}$ to denote sets, capital letters like X to denote random variables and small letters like x denote concrete values. To save on notation we write X^i for $(X_1, X_2, \ldots, X_i)$. If $\mathcal{E}$ denotes an event, $\bar{\mathcal{E}}$ denotes the complementary event. By $\Pr[\Pi : \mathcal{E}]$ we denote the probability that event $\mathcal{E}$ occurs in random experiment Π.

By $s \stackrel{\$}{\leftarrow} \mathcal{S}$ we denote the operation of selecting s uniformly at random from the set $\mathcal{S}$. If D is a probability distributions over $\mathcal{S}$ then $s \leftarrow D$ denotes the operation of selecting s at random according to D. If A denotes an algorithm, we let $b \leftarrow A$ denote that b is the output of A and by $A \rightarrow b$ we denote the event that A outputs b.

2.2 Cryptographic Primitives and Reductions

PRIMITIVES. A cryptographic primitive is a family of systems – indexed by a security parameter $\gamma \in \mathbb{N}$ – with some security property, typically defined by a game between a challenger and an adversary. There is a difference between what one expects from a cryptographic primitive and what is generally considered a successful adversary. The primitive should be *uniformly* and *efficiently* computable, i.e., computable by a probabilistic uniform Turing Machine (UTM) with running time polynomial in γ, and achieve its task with *overwhelming*[6] probability to be considered useful. And it is considered *computationally* secure if no efficient non-uniform adversary, i.e., polynomial (in γ) size circuit family, exists that can violate the security property with some *non-negligible*[7] probability.[8] If the latter even holds for computationally unbounded adversaries, the primitive is referred to as *information theoretically* (or *unconditionally*) secure.

CONSTRUCTIONS. In lack of provable lower bounds hardly no progress has been made for proving the existence of computational primitives even though typically several candidates exist. Instead, there are many results showing that the existence of a certain primitive (say P_1) implies the existence of an other primitive (say P_2). This is often given in a constructive manner in the sense that given a system S of type P_1 there is an

[6] $\tau(\cdot)$ is overwhelming if $1 - \tau(\cdot)$ is negligible.

[7] I.e., the negation of negligible.

[8] Often the non-uniformity of the adversary is not explicitly stated as the security proofs also work in the uniform setting, i.e., a uniform (non-uniform) primitive implies security against efficient uniform (non-uniform) adversaries. All our security proofs also work in the uniform setting, except for Lemma 1 on page 31 for which we do not know how to prove a uniform version of.

algorithm $C(\cdot)$ (called the construction) that takes S as input and transforms it into a system $C(S)$ of type P_2. We say that the security of $C(S)$ is based on S or that $C(S)$ is *conditionally* secure.

BLACK-BOX CONSTRUCTION. Most constructions only make use of the input-output behavior of the system to be transformed and in particular do not make use of the implementation or code describing the given system. Such constructions are called black-box constructions since it is as if the system was put inside of a black-box. Such constructions are often denoted by $C^{\langle\cdot\rangle}$, where the dot is a place holder for the system to be transformed.

BLACK-BOX PROOF. To show a result of the form "$C(\cdot)$ *transforms a system* S *of type* P_1 *into* $C(S)$ *of type* P_2", one shows that the existence of an adversary A_2 that breaks $C(S)$ (in the sense of P_2) implies the existence of an adversary A_1 that breaks S (in the sense of P_1). These proofs are mostly black-box, in the sense that A_1 only uses the input-output behavior of A_2 and S, i.e., $A_1^{A_2,S}$ breaks S if A_2 breaks $C(S)$.

BLACK-BOX REDUCTION. Putting things together, a black-box reduction is a black-box construction with a black-box proof. To be more precise, a black-box reduction consists of two algorithms $C^{\langle\cdot\rangle}$ and $A_1^{\langle\cdot\rangle,\langle\cdot\rangle}$ satisfying the following properties. Let S denote a system of type P_1, then C^S is a system of type P_2 and for all adversaries A_2 that break C^S (in the sense of P_2), $A_1^{A_2,S}$ breaks S (in the sense of P_1).

WEAKENED PRIMITIVES. As mentioned previously, the research goal of this thesis has been to base the security of cryptographic systems on as weak primitives as possible. Put differently, for some primitive P_2 we try to find a possibly weak primitive P_1 and an efficient construction $C^{\langle\cdot\rangle}$ such that if S is of type P_1 then C^S is of type P_2. Our results are presented in a concrete security framework and in particular we are interested in how efficient the constructions are (e.g. the number of queries $C^{\langle\cdot\rangle}$ issues to S).

2.3 Random Systems

Many results in this thesis are stated and proven in the random systems framework of Maurer [Mau02]. A *random system* is a system which takes inputs $X_1, X_2, \ldots$ and generates, for each new input X_i, an output Y_i which depends probabilistically on the inputs and outputs seen so far.

We define random systems in terms of the distribution of the outputs Y_i conditioned on X^iY^{i-1} (i.e., the actual query X_i and all previous input/output pairs $X_1Y_1, \ldots, X_{i-1}Y_{i-1}$).

Definition 1 (Random systems). *An* $(\mathcal{X}, \mathcal{Y})$-random system $\mathbf{F}$ *is a sequence of conditional probability distributions* $\mathsf{P}^{\mathbf{F}}_{Y_i|X^iY^{i-1}}$ *for* $i \geq 1$. *Here we denote by* $\mathsf{P}^{\mathbf{F}}_{Y_i|X^iY^{i-1}}(y_i, x^i, y^{i-1})$ *the probability that* $\mathbf{F}$ *will output* y_i *on input* x_i *conditioned on the fact that* $\mathbf{F}$ *did output* y_j *on input* x_j *for* $j = 1, \ldots, i-1$.

As special classes of random systems we will consider *random functions* (which are exactly the stateless random systems) and *random permutations*.

Definition 2 (Random functions and permutations). *A* random function $\mathcal{X} \rightarrow \mathcal{Y}$ *(*random permutation *on* $\mathcal{X}$*) is a random variable which takes as values functions* $\mathcal{X} \rightarrow \mathcal{Y}$ *(permutations on* $\mathcal{X}$*).*

A uniform random function (URF) $\mathbf{R} : \mathcal{X} \rightarrow \mathcal{Y}$ *(A* uniform random permutation (URP) $\mathbf{P}$ *on* $\mathcal{X}$*) is a random function with uniform distribution over all functions from* $\mathcal{X}$ *to* $\mathcal{Y}$ *(permutations on* $\mathcal{X}$*).*

A uniform random VOL-function $\mathbf{R} : \mathcal{X} \times \mathbb{N} \rightarrow \{0,1\}^*$ *is a VOL-function for which* $\mathbf{R}(\cdot, l)$ *is a URF* $\mathcal{X} \rightarrow \{0,1\}^l$ *for all* l.

Throughout, $\mathbf{P}$ and $\mathbf{R}$ are used for functions as defined above (when $\mathcal{X}$, $\mathcal{Y}$ are to be understood and also whether the function has VOL or not). To be explicit, we sometimes let $\mathbf{R}_{L,\ell}$, $\mathbf{R}_{\leq L,\ell}$, $\mathbf{R}_{*,\ell}$ denote uniform random functions where $\mathcal{Y} = \{0,1\}^\ell$ and $\mathcal{X}$ equals $\{0,1\}^L$, $\{0,1\}^{\leq L}$, and $\{0,1\}^*$, respectively. By $\mathbf{R}_{L,*}$ we denote a uniform VOL-function with $\mathcal{X} = \{0,1\}^L$ (and $\mathcal{Y} = \{0,1\}^*$).

A pair x, x' of distinct inputs for a function $\mathbf{F}$ satisfying $\mathbf{F}(x) = \mathbf{F}(x')$ is referred to as a *non-trivial* collision for $\mathbf{F}$.

Definition 3. *For a (randomized) function* $\mathbf{F}$ *we denote with* $\mathsf{coll}_q(\mathbf{F})$ *the collision probability of any fixed q-tuple of distinct inputs, i.e.,*

$$\mathsf{coll}_q(\mathbf{F}) \stackrel{def}{=} \max_{x_1,\ldots,x_q} \Pr[\exists i, j \in \{1, \ldots, q\}, i \neq j, \mathbf{F}(x_i) = \mathbf{F}(x_j)]. \tag{2.1}$$

For a random function this gives the so-called *birthday bound* (by applying the union bound):

$$\mathsf{coll}_q(\mathbf{R}_{L,\ell}) \leq q(q-1)/2^{\ell+1}. \tag{2.2}$$

We will often compose some $(\mathcal{X}, \mathcal{Y})$-random system $\mathbf{F}(.)$ with some $(\mathcal{Y}, \mathcal{Z})$-random system $\mathbf{G}(.)$.

Definition 4. *With* $\mathbf{F} \triangleright \mathbf{G}(.) \stackrel{\text{def}}{=} \mathbf{G}(\mathbf{F}(.))$ *we denote the sequential composition of* $\mathbf{F}$ *and* $\mathbf{G}$.[9]

The following well-known method for transforming any function to a permutation is illustrated in Figure 2.1.

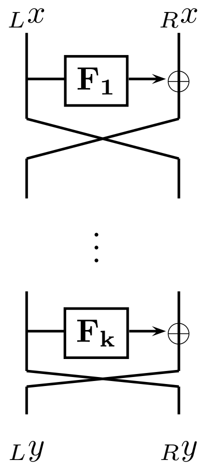


Figure 2.1: The k-round Feistel-network

Definition 5 (Feistel-network)**.** *The (one-round) Feistel-network* $\psi[\mathbf{F}] : \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ *is a permutation based on a function* $\mathbf{F} : \{0,1\}^n \rightarrow \{0,1\}^n$ *and is defined as*

$$\psi[\mathbf{F}](x) \stackrel{\text{def}}{=} (\mathbf{F}({}_Lx) \oplus {}_Rx) \| {}_Lx \tag{2.3}$$

(where we sometimes write ψ_{2n} *instead of* ψ *to make the size of the input explicit). With*

$$\psi[\mathbf{F}_1 \cdots \mathbf{F}_k] \stackrel{\text{def}}{=} \psi[\mathbf{F}_1] \triangleright \psi[\mathbf{F}_2] \triangleright \cdots \triangleright \psi[\mathbf{F}_k]$$

we denote the k*-round Feistel-network based on (randomized) round functions* $\mathbf{F}_1, \ldots, \mathbf{F}_k$*, here the randomness used by any function is always assumed to be independent of the randomness of the other round functions. The* k*-round Feistel-network where the same instantiation of a function* $\mathbf{F}$ *is used for all rounds is denoted by* $\psi[\mathbf{F}^k] \stackrel{\text{def}}{=} \psi[\underbrace{\mathbf{F} \cdots \mathbf{F}}_{k \text{ times}}]$.

[9] Note that $\mathbf{F} \triangleright \mathbf{G}$ is usually denoted with $\mathbf{G} \circ \mathbf{F}$, if $\mathbf{F}$ and $\mathbf{G}$ are functions.

In the sequel, we will consider random systems that interact. For this we define the concept of a random system which is one query ahead.

Definition 6. *A $(\mathcal{Y},\mathcal{X})$-random system $\mathbf{D}$ which is* one query ahead *is defined by $\mathsf{P}^{\mathbf{A}}_{X_i|Y^{i-1}X^{i-1}}$ for all i.*

In particular, the first output $\mathsf{P}^{\mathbf{D}}_{X_1}$ is defined before $\mathbf{D}$ is fed with any input.

We can now consider the random experiment where a $(\mathcal{Y},\mathcal{X})$-random system which is one query ahead queries a $(\mathcal{X},\mathcal{Y})$-random system

Definition 7. *With $\mathbf{D}\diamond\mathbf{F}$ we denote the random experiment where a random system $\mathbf{D}$ which is one query ahead interactively queries a compatible random system $\mathbf{F}$.*

2.4 Indistinguishability

2.4.1 Quasirandom Setting

RANDOM VARIABLES. A distinguisher $\mathbf{D}$ for two distributions D_1, D_2 over some set $\mathcal{X}$ (or equivalently for the corresponding random variables) is a $\mathcal{X}\to\{0,1\}$ random system.

Definition 8. *The advantage of a distinguisher $\mathbf{D}$ for two probability distributions D_1, D_2 over a finite set is*

$$\Delta^{\mathbf{D}}(D_1,D_2) \stackrel{\text{def}}{=} \Big|\Pr\left[s\leftarrow D_1 : \mathbf{D}(s)\to 1\right] - \Pr\left[s\leftarrow D_2 : \mathbf{D}(s)\to 1\right]\Big|,$$

and the maximal distinguishing advantage for D_1, D_2 is

$$\Delta(D_1,D_2) \stackrel{\text{def}}{=} \max_{\mathbf{D}} \Delta^{\mathbf{D}}(D_1,D_2).$$

$\Delta(D_1,D_2)$ is also referred to as the statistical distance of D_1 and D_2.

RANDOM SYSTEMS. It is more intricate to define what we mean by the indistinguishability of random systems as here one must specify how the systems can be accessed.

Definition 9. *A $(\mathcal{Y},\mathcal{X})$-*distinguisher *is a $(\mathcal{Y},\mathcal{X})$-random system which is one query ahead.*

We divide distinguishers into classes by posing restrictions on how the distinguisher can produce its queries. In particular the following attacks will be of interest to us

- CPA (Adaptive Chosen-Plaintext Attack): here the distinguisher can choose the i-th query after receiving the $(i-1)$-th output.
- nCPA (Non-Adaptive Chosen-Plaintext Attack): the distinguisher must choose all queries in advance.
- KPA (Known-Plaintext Attack): the distinguisher must choose all queries as specified by a third party which chooses the queries uniformly at random.[10]

If $\mathbf{F}$ is a permutation, its inverse $\mathbf{F}^{-1}$ is well defined and we can consider the attacks

- CCA (Adaptive Chosen-Ciphertext Attack)
- nCCA (Non-Adaptive Chosen-Ciphertext Attack)

which are defined like a CPA and nCPA, respectively, but where the attacker can additionally make queries from the inverse direction.

Definition 10. *For $q \geq 1$, the two random experiments $\mathbf{D}\diamond\mathbf{F}$ and $\mathbf{D}\diamond\mathbf{G}$ define a distribution over $\mathcal{X}^q \times \mathcal{Y}^q$. The* advantage *of $\mathbf{D}$ after q queries in distinguishing $\mathbf{F}$ from $\mathbf{G}$, denoted $\Delta_q^{\mathbf{D}}(\mathbf{F},\mathbf{G})$, is the statistical difference between those distributions*

$$\Delta_q^{\mathbf{D}}(\mathbf{F},\mathbf{G}) \stackrel{\text{def}}{=} \frac{1}{2} \sum_{\mathcal{X}^q \times \mathcal{Y}^q} \left| \mathsf{P}^{\mathbf{D}\diamond\mathbf{F}}_{X^q Y^q} - \mathsf{P}^{\mathbf{D}\diamond\mathbf{G}}_{X^q Y^q} \right| . \tag{2.4}$$

The maximal advantage of an ATK*-distinguisher making q queries for $\mathbf{F}$ and $\mathbf{G}$ is*

$$\Delta_q^{\mathsf{ATK}}(\mathbf{F},\mathbf{G}) \stackrel{\text{def}}{=} \max_{\mathsf{ATK}\text{-}\textit{distinguisher }\mathbf{D}} \Delta_q^{\mathbf{D}}(\mathbf{F},\mathbf{G}). \tag{2.5}$$

Informally, a family of random functions indexed by a security parameter ($\gamma \in \mathbb{N}$) is an ATK-secure quasirandom function (QRF), if for any polynomial $p(.)$ the distinguishing advantage $\Delta_{p(\gamma)}^{\mathsf{ATK}}(\mathbf{F},\mathbf{R})$ is negligible (in γ). A quasirandom permutation (QRP) is defined similarly but using $\mathbf{P}$ instead of $\mathbf{R}$, and where we additionally require that $\mathbf{F}$ (for any value of the security parameter) is a permutation.

[10] There is a crucial difference if the adversary chooses the random queries itself, see [PS06].

Remark 1. *To be more general, we could consider distinguishers that output a decision bit after each query (say D_i after the i-th query). It is easy to verify that (2.5) is equivalent to*

$$\max_{\mathsf{ATK}\text{-}\textit{distinguisher}\ \mathbf{D}} \left| \Pr\left[\mathbf{D}\diamond\mathbf{F} : D_q = 1\right] - \Pr\left[\mathbf{D}\diamond\mathbf{G} : D_q = 1\right] \right|,$$

but it is not the necessarily the case that (2.4) is equivalent to

$$\left| \Pr\left[\mathbf{D}\diamond\mathbf{F} : D_q = 1\right] - \Pr\left[\mathbf{D}\diamond\mathbf{G} : D_q = 1\right] \right|$$

as $\mathbf{D}$ may always output 0. However, if $\mathbf{D}$ makes optimal choices for D_i (based on the information at hand) this turns out to be the same.

2.4.2 Pseudorandom Setting

In the pseudorandom setting, we typically put a restriction on the efficiency (or size) of random systems by modeling them as circuits. For a circuit D, we let $|\mathsf{D}|$ denote the size.

RANDOM VARIABLES. A circuit distinguisher D for two distributions D_1, D_2 over some set $\mathcal{X}$ (or equivalently for the corresponding random variables) is a $\mathcal{X} \to \{0,1\}$ random system (modeled as a circuit).

Definition 11. *The advantage of a circuit distinguisher D for two probability distributions D_1, D_2 over some finite set is*

$$\mathbf{Adv}^{\mathsf{D}}(D_1, D_2) \stackrel{\text{def}}{=} \left| \Pr\left[x \leftarrow D_1 : \mathsf{D}(x) \to 1\right] - \Pr\left[x \leftarrow D_2 : \mathsf{D}(x) \to 1\right] \right|,$$

and the maximal distinguishing advantage is

$$\mathbf{Adv}_t(D_1, D_2) \stackrel{\text{def}}{=} \max_{\mathsf{D}, |\mathsf{D}| \le t} \mathbf{Adv}^{\mathsf{D}}(D_1, D_2).$$

As special cases of indistinguishability of distributions, the following well-known distributions, generated by some cyclic group G and some generator g of G, will be of interest to us.

Definition 12 (Decisional Diffie Hellman (DDH) [DH76])**.** *For a cyclic group G of order ρ and a generator g of G*

$$\begin{aligned} &\mathbf{Adv}_s^{DDH}(G, g) \\ \stackrel{\text{def}}{=}\ &\max_{\mathsf{D}, |\mathsf{D}| \le s} \left| \Pr_{a,b}\left[\mathsf{D}(g, g^a, g^b, g^{ab}) \to 1\right] - \Pr_{a,b,c}\left[\mathsf{D}(g, g^a, g^b, g^c) \to 1\right] \right|, \end{aligned}$$

where the probability is over the random choice of $a, b, c \in \mathbb{Z}_\rho$.

If the group G in addition has prime order, we also consider the following distribution.

Definition 13 (Inverse Decisional Diffie Hellman (IDDH) [BDZ03])**.** *For a cyclic group G of prime order ρ and a generator g of G*

$$\begin{aligned}&\mathbf{Adv}_t^{IDDH}(G,g)\\ \stackrel{\text{def}}{=}\ &\max_{\mathsf{D},|\mathsf{D}|\leq t}\left|\Pr_a\left[\mathsf{D}(g,g^a,g^{a^{-1}})\to 1\right]-\Pr_{a,b}\left[\mathsf{D}(g,g^a,g^b)\to 1\right]\right|,\end{aligned}$$

where the probability is over the random choice of $a, b \in \mathbb{Z}_\rho$.

Let $\mathcal{G}$ denote an efficiently computable family of groups indexed by a security parameter $\gamma \in \mathbb{N}$. By efficiently computable we mean that one can efficiently (i.e., in time polynomial in γ by a UTM) sample a group (together with a generator) from the family, and efficiently compute the group operations therein. Abusing notation we denote with $(G, g) = \mathcal{G}(\gamma)$ any group/generator pair for security parameter γ. And $\mathcal{G}$ is referred to as a DDH and an IDDH group if for any polynomial $p(.)$ the distinguishing advantages $\mathbf{Adv}_{p(\gamma)}^{DDH}(\mathcal{G}(\gamma))$ and $\mathbf{Adv}_{p(\gamma)}^{IDDH}(\mathcal{G}(\gamma))$ are negligible (in γ), respectively.[11]

Next, we give an example of a group G in which the DDH and IDDH distinguishing problem is conjectured to be hard (for an overview of other such groups see [Bon98]); let ρ, φ be "large" primes (say $\lceil \log_2(\rho)\rceil = 1024$ and $\log_2(\varphi) \geq 160$) such that φ divides $\rho - 1$ and then let G be the subgroup of order φ in $\mathbb{Z}_\rho^*$.

RANDOM SYSTEMS. A circuit distinguisher for random systems has (as in the quasirandom setting) access to a random system $\mathbf{S}$. After some number of queries to $\mathbf{S}$ it outputs a decision bit. This process is (as before) denoted by $\mathsf{D}\diamond\mathbf{S}$.

Definition 14. *The advantage of a circuit distinguisher D for $\mathbf{F}$ and $\mathbf{G}$ is*

$$\mathbf{Adv}^{\mathsf{D}}(\mathbf{F},\mathbf{G}) \stackrel{\text{def}}{=} |\Pr[\,\mathsf{D}\diamond\mathbf{F} : \mathsf{D}\to 1] - \Pr[\,\mathsf{D}\diamond\mathbf{G} : \mathsf{D}\to 1]|.$$

The maximal advantage of an ATK-distinguisher of size at most t that makes at most q queries is[12]

$$\mathbf{Adv}_{t,q}^{\mathsf{ATK}}(\mathbf{F},\mathbf{G}) \stackrel{\text{def}}{=} \max_{\substack{\mathsf{ATK}\text{-distinguisher } \mathsf{D},\\ |\mathsf{D}|\leq t,\ \#\text{ of queries} \leq q}} \mathbf{Adv}^{\mathsf{D}}(\mathbf{F},\mathbf{G}).$$

[11] It is easy to show that an IDDH group is a DDH group, but it is an open question whether a DDH group of prime order is an IDDH group.

[12] In particular $\mathbf{Adv}_{\infty,q}^{\mathsf{ATK}}(\mathbf{F},\mathbf{G}) = \Delta_q^{\mathsf{ATK}}(\mathbf{F},\mathbf{G})$.

A family of keyed functions F indexed by a security parameter $\gamma \in \mathbb{N}$ is an ATK-secure pseudorandom function (PRF) if F (with security parameter γ) is computable in polynomial (in γ) time by a UTM and the maximal distinguishing advantage $\mathbf{Adv}^{\mathsf{ATK}}_{p(\gamma),p(\gamma)}(\mathsf{F}, \mathbf{R})$ is negligible in γ for any polynomial $p(.)$ (and uniformly at random chosen key). Pseudorandom permutations (PRP) are defined similarly but using $\mathbf{P}$ instead of $\mathbf{R}$, and where we additionally require that F (for any value of the security parameter and key) is a permutation.

Similarly to Definition 14, we consider the maximal ATK-advantage for distinguishing random VOL-functions. The only difference is that we have an extra parameter specifying the total output length and that the distinguisher also specifies the output length in its queries.

Definition 15. *The maximal advantage of a (circuit)* ATK*-distinguisher* D *of size at most t that makes at most q queries*[13] *of total output length at most μ for the VOL-functions* $\mathbf{F}$ *and* $\mathbf{G}$ *is*

$$\mathbf{Adv}^{\mathsf{VOL\text{-}ATK}}_{t,q,\mu}(\mathbf{F}, \mathbf{G}) \stackrel{\text{def}}{=} \max_{\mathsf{D}} \mathbf{Adv}^{\mathsf{D}}(\mathbf{F}, \mathbf{G}),$$

where the maximum is taken over all distinguishers D *with the above resources.*

A $\mathsf{VOL\text{-}ATK}$-secure PRF F is a family of keyed VOL functions, indexed by a security parameter $\gamma \in \mathbb{N}$, for which F (with security parameter γ) is computable in polynomial (in γ and the desired output-length) time by a UTM and the maximal advantage $\mathbf{Adv}^{\mathsf{VOL\text{-}ATK}}_{p(\gamma),p(\gamma),p(\gamma)}(\mathsf{F}, \mathbf{R})$ is negligible in γ for any polynomial $p(.)$ (and key chosen uniformly at random).

To simplify the notation, we will frequently refer to a CPA-secure PRF simply as a PRF, a KPA-secure PRF as a weak PRF (WPRF), and a KPA-secure VOL-PRF as a VOL-WPRF.

2.5 Unpredictability

A random function is considered unpredictable or unforgeable if it is infeasible for computationally bounded (or sometimes unbounded) adversaries to predict the output for any input x. This should be the case even if the adversary gets to see the evaluation of the function on several inputs (different from x). This is formalized by giving the adversary access to the function prior to the prediction.

[13] A KPA-distinguisher may in advance specify the output length of each query.

Definition 16. *An adversary (or forger)* $\mathbf{A}$ *for a random function* $\mathcal{X} \to \mathcal{Y}$ *has access to the function and outputs (after some number of queries to the function) a value in* $\mathcal{X} \times \mathcal{Y}$ *(called the forgery).*

As for indistinguishability, we divide the forgers into attack classes by posing restrictions on how the forger can produce its queries. We consider the attack classes CPA and KPA as previously defined.

Here, we consider the pseudorandom setting and hence put a restriction on the efficiency (or size) of the forger by modeling it as a circuit A (which outputs a forgery). To be precise, in the random experiment $\mathsf{A}\diamond\mathbf{F}$ (for some random function $\mathbf{F}$), the forger A queries $\mathbf{F}$ some number of times and finally outputs a forgery.

Definition 17. *The success probability of a (circuit) forger* A *in forging* $\mathbf{F}$ *is defined as follows. In the random experiment* $\mathsf{A}\diamond\mathbf{F}$, *let* $x_1, \ldots, x_q$ *denote the oracle queries issued by* A*. Then*

$$\mathbf{Succ}^{\mathsf{A}}(\mathbf{F}) \stackrel{\text{def}}{=} \Pr[\, \mathsf{A}\diamond\mathbf{F}, (x,y) \leftarrow \mathsf{A} : y = \mathbf{F}(x),\ x \notin \{x_1, \ldots, x_q\}],$$

where (x, y) *denotes the forgery of* A*. The maximal success probability is*

$$\mathbf{InSec}^{\mathsf{UF\text{-}ATK}}_{t,q,\mu}(\mathbf{F}) \stackrel{\text{def}}{=} \max_{\mathsf{A}} \mathbf{Succ}^{\mathsf{A}}(\mathbf{F}),$$

where the maximum is taken over all ATK*-forger of size at most* t *that issues at most* q *queries (including the forgery) to* $\mathbf{F}$ *of total input length at most* μ *bits (including the length of the forgery input part* x*).*

If $\mathbf{F}$ has FIL, we drop the parameter μ as it is given by the number of queries q and the input length of $\mathbf{F}$.

Informally, a family of keyed functions F indexed by a security parameter $\gamma \in \mathbb{N}$ is an ATK-secure *message authentication code* (MAC) if F (with security parameter γ) is computable by a UTM in polynomial (in γ and the input length) time, and for any polynomial $p(.)$ the maximal success probability $\mathbf{InSec}^{\mathsf{UF\text{-}ATK}}_{p(\gamma),p(\gamma),p(\gamma)}(\mathsf{F})$ is negligible in γ (for a key chosen uniformly at random). For simplicity, we will refer to a CPA-secure MAC simply as a MAC and a KPA-secure MAC as a weak MAC (WMAC). Furthermore, an ATK-forger A of size at most t, that makes at most q queries (including the forgery) to its oracle of total length at most μ (including the length of the forgery message) and that has success probability at least ε (i.e., $\varepsilon \leq \mathbf{Succ}^{\mathsf{UF\text{-}CPA}}_{t,q,\mu}(\mathsf{F})$), is denoted as a

$$(t, q, \mu, \varepsilon)_{\mathsf{ATK}}\text{-forger},$$

where we drop ATK and μ if they are implicitly given.

2.6 Encryption

A symmetric encryption scheme $\mathcal{SE} = (\mathsf{Enc}, \mathsf{Dec})$ consists of two efficient algorithms.[14] The randomized encryption algorithm Enc is a $(\mathcal{K} \times \mathcal{M}, \mathcal{C})$-random system and the deterministic decryption algorithm is of the form

$$\mathsf{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$$

(where $\perp$ stands for invalid ciphertext). We require that for any key $k \in \mathcal{K}$ and message $m \in \mathcal{M}$ we have that $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$. There are several notions for privacy and integrity of $\mathcal{SE}$ (see [BN00, KY00, BDJR97]).

2.6.1 Privacy

We use the privacy notion IND-Px-Cy (for some $x, y \in \{0, 1, 2\}$) of $\mathcal{SE}$ as introduced in [KY00]. It formalizes an adversary's inability, given certain oracle access to the encryption Enc_k and decryption oracle Dec_k, to distinguish the ciphertexts of two chosen plaintexts (of the same length). This concept is modeled with help of the *left-or-right encryption oracle* $\mathsf{Enc}_k(\mathcal{LR}(\cdot, \cdot, b))$, defined as

$$\mathsf{Enc}_k(\mathcal{LR}(m_0, m_1, b)) \stackrel{\text{def}}{=} \mathsf{Enc}_k(m_b),$$

where $b \in \{0, 1\}$ (and $k \in \mathcal{K}$). To be precise:

Definition 18. *An adversary* A *for an encryption scheme $\mathcal{SE}$ has access to a $(\mathcal{M}, \mathcal{C})$-random system (the encryption oracle) and a $(\mathcal{C}, \mathcal{M})$-random system (the decryption oracle) for which it can provide as inputs any messages and ciphertexts, respectively.* A *can also issue a single query to a $(\mathcal{M} \times \mathcal{M}, \mathcal{C})$-random system (the left-or-right encryption oracle) with input messages of equal size. Finally,* A *returns a bit.*

We divide the adversary into classes by posing restrictions on when the distinguisher can query the encryption oracle Enc_k and the decryption oracle Dec_k, respectively. Let

- P0 denote that Enc_k may never be invoked.
- P1 denote that Enc_k may be invoked until the left-or-right encryption oracle $\mathsf{Enc}_k(\mathcal{LR}(\cdot, \cdot, b))$ is invoked but not thereafter.

[14] We assume that it is efficient to sample keys from $\mathcal{K}$.

- P2 denote that Enc_k may always be invoked.

Similarly, let

- C0 denote that Dec_k may never be invoked.
- C1 denote that Dec_k may be invoked until the left-or-right encryption oracle $\mathsf{Enc}_k(\mathcal{LR}(\cdot,\cdot,b))$ is invoked but not thereafter.
- C2 denote that Dec_k may always be invoked but not on the value returned from $\mathsf{Enc}_k(\mathcal{LR}(\cdot,\cdot,b))$ (after $\mathsf{Enc}_k(\mathcal{LR}(\cdot,\cdot,b))$ is invoked).

By $\mathsf{A}\diamond[\mathsf{Enc}_k, \mathsf{Dec}_k, \mathsf{Enc}_k(\mathcal{LR}(\cdot,\cdot,b))]$ we denote the process in which A queries the random systems Enc_k, Dec_k, and $\mathsf{Enc}_k(\mathcal{LR}(\cdot,\cdot,b))$, and then outputs a bit $\hat{b}$ (i.e., a prediction for b). An adversary A whose queries satisfies Px and Cy (for some $x, y \in \{0,1,2\}$) is referred to as a IND-Px-Cy-adversary.

Definition 19. *The advantage of a (circuit) adversary* A *for an encryption scheme* $\mathcal{SE} = (\mathsf{Enc}, \mathsf{Dec})$ *is defined as*

$$\mathbf{Adv}^{\mathsf{A}}(\mathcal{SE}) \stackrel{\text{def}}{=} 2\cdot\Pr\left[k \stackrel{\$}{\leftarrow} \mathcal{K}, b \stackrel{\$}{\leftarrow} \{0,1\}, \mathsf{A}\diamond[\mathsf{Enc}_k, \mathsf{Dec}_k, \mathsf{Enc}_k(\mathcal{LR}(\cdot,\cdot,b))] : \mathsf{A} \to b\right] - 1.$$

For $x, y \in \{0,1,2\}$, the maximal advantage of an IND-Px-Cy*-adversary* A *for* $\mathcal{SE}$ *is*

$$\mathbf{Adv}^{\mathsf{IND\text{-}P}x\text{-}\mathsf{C}y}_{t,q,\mu,q',\mu'}(\mathcal{SE}) \stackrel{\text{def}}{=} \max_{\mathsf{A}} \mathbf{Adv}^{\mathsf{A}}(\mathcal{SE}),$$

where the maximum is taken over all IND-Px-Cy*-adversaries* A *of size at most t that makes at most $q-1$ queries to* Enc_k *of total input length at most $\mu - |m_0|$ (where m_0 denotes one of* A*'s inputs to* $\mathsf{Enc}_k(\mathcal{LR}(\cdot,\cdot,b))$*), and q' queries to* Dec_k *of total input length μ'.*

An encryption scheme $\mathcal{SE} = (\mathsf{Enc}, \mathsf{Dec})$ indexed by a security parameter $\gamma \in \mathbb{N}$ is IND-Px-Cy-secure if Enc and Dec (with security parameter γ) are computable in polynomial (in γ and the input length) time by a UTM, and for any polynomial $p(.)$ the maximal advantage $\mathbf{Adv}^{\mathsf{IND\text{-}P}x\text{-}\mathsf{C}y}_{p(\gamma),p(\gamma),p(\gamma),p(\gamma),p(\gamma)}(\mathcal{SE})$ is negligible in γ. To simplify the notation, we often refer to the IND-P2-C0-, IND-P1-C1-, and IND-P2-C2-security notions as CPA-, nCCA-, and CCA-security, respectively.[15]

[15] As shown in [KY00], IND-P1-Cy implies IND-P2-Cy for $y \in \{0,1,2\}$. Furthermore, in the language of [BDJR97], IND-P2-C0 and IND-P2-C2 are equivalent to FTG-CPA and FTG-CCA, respectively, and FTG implies ROR, LOR, and SEM.

2.6.2 Integrity

The strongest integrity notion for an encryption scheme is *integrity of ciphertexts* ($\mathsf{INT\text{-}CTXT}$) [BN00]. It formalizes the infeasibility of any efficient adversary A given oracle access to the encryption oracle Enc_k to come up with a valid ciphertext c different from the outputs of Enc_k. This is modeled with help of a *verification oracle* $\mathsf{Dec}_k^* : \mathcal{C} \to \{0,1\}$ defined as

$$\mathsf{Dec}_k^*(c) = \begin{cases} 1 & \mathsf{Dec}_k(c) \neq \perp \\ 0 & \text{otherwise.} \end{cases}$$

To be precise:

Definition 20. *An* $\mathsf{INT\text{-}CTXT}$ *adversary* A *for an encryption scheme* $\mathcal{SE}$ *has access to a* $(\mathcal{M}, \mathcal{C})$*-random system (the encryption oracle) for which it can provide as input any message.* A *also has access to a* $(\mathcal{C}, \{0,1\})$*-random system* Dec_k^* *(the verification oracle) for which it can provide any input different from the outputs from the encryption oracle.*

Let $\mathsf{A}\diamond[\mathsf{Enc}_k, \mathsf{Dec}_k^*]$ denote the process where A interacts with Enc_k and Dec_k^*.

Definition 21 ($\mathsf{INT\text{-}CTXT}$)**.** [BN00] *The success probability of a (circuit) adversary* A *to violate the integrity of ciphertexts of an encryption scheme* $\mathcal{SE} = (\mathsf{Enc}, \mathsf{Dec})$ *is*

$$\mathbf{Succ}^{\mathsf{A}}(\mathcal{SE}) \stackrel{def}{=} \Pr\left[k \stackrel{\$}{\leftarrow} \mathcal{K}, \mathsf{A}\diamond[\mathsf{Enc}_k, \mathsf{Dec}_k^*] : \mathsf{Dec}_k^* \to 1\right],$$

where $\mathsf{Dec}_k^* \to 1$ *denotes the event that some output of* Dec_k^* *equals* 1*. The maximal success probability of an* $\mathsf{INT\text{-}CTXT}$*-adversary is defined as*

$$\mathbf{InSec}^{\mathsf{INT\text{-}CTXT}}_{t,q,\mu,q',\mu'}(\mathcal{SE}) \stackrel{def}{=} \max_{\mathsf{A}} \mathbf{Succ}^{\mathsf{A}}(\mathcal{SE}),$$

where the maximum is taken over all $\mathsf{INT\text{-}CTXT}$*-adversaries* A *of size at most* t *that makes at most* q *queries to* Enc_k *of total length at most* μ *bits and at most* q' *queries to* D_k^* *of total length at most* μ' *bits.*

We say that $\mathcal{SE} = (\mathsf{Enc}, \mathsf{Dec})$, indexed by a security parameter $\gamma \in \mathbb{N}$, assures $\mathsf{INT\text{-}CTXT}$ if Enc and Dec (with security parameter γ) is computable in polynomial (in γ and the input length) time by a UTM and for any polynomial $p(.)$ the $\mathbf{InSec}^{\mathsf{INT\text{-}CTXT}}_{p(\gamma),p(\gamma),p(\gamma),p(\gamma),p(\gamma)}(\mathcal{SE})$ is negligible in γ.

Chapter 3

Luby-Rackoff Ciphers from Weak Round Functions

In this chapter, we propose various relaxations of the Luby-Rackoff ciphers. In particular, we investigate for the first time – to the best of our knowledge – the CPA-security of the permutation one gets by a Feistel-network where none of the round functions is guaranteed to be CPA-secure. The results of this chapter appeared in [MOPS06a, MOPS06b].

3.1 Background

LUBY-RACKOFF CIPHERS. In their celebrated paper [LR86] Luby and Rackoff prove that the three-round Feistel-network is a CPA-secure PRP (or block-cipher) if each round is instantiated with an independent CPA-secure PRF, and with one extra round even CCA-security is achieved. Besides reducing PRPs to PRFs, this result also gives some confidence in the soundness of using a Feistel-network to design block-ciphers. In order to prove that some system – which is built from *pseudorandom* components – is pseudorandom itself, it is often enough to prove it to be *quasirandom* when the components are replaced by the corresponding ideal systems. In particular, to prove the security of the original three-round Luby-Rackoff cipher it is enough to prove the purely information-theoretic result that the three-round Feistel-network instantiated with independent URFs is a CPA-secure QRP. It then immediately follows that the construction is a CPA-secure PRP when the URFs are replaced by CPA-secure PRFs, since if it was not a CPA-secure PRP, we could use the distinguisher for it to build a distinguisher for the CPA-secure PRF (via a standard hybrid argument). Similarly, one can show that if the round functions are only nCPA- or KPA-secure PRFs, the construction is a secure PRP, but only against the class of attacks nCCA (hence also nCPA) and KPA, respectively.

RELAXATIONS. In order to achieve more efficient constructions of PRPs from PRFs, many researchers have investigated the security of weakened versions of the Luby-Rackoff ciphers. Several variations of the ciphers were proven to be pseudorandom where for example the round functions were not required to be independent [Pie90] or the distinguisher was given direct oracle access to some of the round functions [RR00]. It is also known that one can replace the first round of the three-round Luby-Rackoff cipher by a pairwise independent permutation [Luc96, NR99a].[16] These results further fortify the confidence in using Feistel-networks to design block ciphers. All these relaxed constructions need at least some of the round functions to be CPA-secure PRFs in order to get a CPA-secure PRP.

3.2 Contributions

All our results are summarized in Figure 3.2 on page 23.

FURTHER RELAXATIONS. We further relax the three-round Luby-Rackoff cipher (which uses a pairwise independent permutation as first round) by showing that the function in the last round only needs to be KPA-secure. This resolves an open question posed by Minematsu and Tsunoo in [MT05]. Furthermore, for achieving KPA-security of the cipher we show that the first round is not at all necessary and that it is sufficient to instantiate the round functions with a single instantiation of a KPA-secure function.

THE SECOND ROUND IS CRUCIAL. We prove that for constructing a CPA-secure permutation, i.e., PRP or QRP depending on the setting, one can not in general instantiate the second round with a function which is only nCPA-secure. This is shown by constructing a counterexample, i.e., a

[16] In fact, the permutation must only be such that on any two values, the collision probability on one half of the domain is small. For example one can use one normal Feistel round instantiated with an almost XOR-universal function.

Construction	Quasirandom	Pseudorandom	Reference
$\psi[RRR]$	CPA, nCCA, ¬ CCA		[LR86, Mau02]
$H \triangleright \psi[RR]$	CPA, ¬ nCCA		[Luc96, NR99a]
$H \triangleright \psi[RK]$	CPA , ¬ nCCA		§3.3
$\psi[RNR]$	¬ CPA		§3.3
$\psi[NNNN]$	CPA	¬ CPA (if IDDH groups exist)	§3.4 and §3.5
$\psi[RRRR]$	CCA		[LR86, Mau02]
$H \triangleright \psi[RR] \triangleright H^{-1}$	CCA		[Luc96, NR99a]
$\psi[NNNNN]$	CCA	?	§3.6
$\psi[RR]$	KPA, ¬ nCPA		[MT05]
$\psi[K^2]$	KPA, ¬ nCPA		§3.3
$\psi[NNN]$	nCCA, ¬ CPA		§3.3
$H \triangleright \psi[NK]$	nCPA, ¬ nCCA		§3.3
$\psi[RKR]$	¬ nCPA		§3.3

Security of the Feistel-network ψ with various security guarantees on the round functions. Here $\psi[f_1 \cdots f_r](\cdot)$ denotes the r-round Feistel-network with f_i in the i-th round, and $\psi[f^2] \stackrel{def}{=} \psi[ff]$, i.e., the same function f in both rounds. Each occurrence of R, N, and K stands for an independent CPA-, nCPA-, *and* KPA-*secure function (i.e., a PRF or a QRF depending on the setting) respectively. The same holds for H which is any "lightweight" permutation from which we only require that the collision probability be small on the left half of the output; e.g. an almost pairwise independent permutation or a Feistel round instantiated with an almost XOR-universal function is sufficient. The results in gray are implied by other results in the table and ? denotes an open question.*

Figure 3.1: Security of the Feistel-network ψ.

nCPA-secure function such that the three-round Feistel-network with this function in the second, and any random functions in the first and third round can easily be distinguished from a uniformly random permutation (URP) with only three adaptively chosen queries. Similarly, if one instantiates the second round with a KPA-secure function, then the construction will in general not even be nCPA-secure.

FOUR nCPA-SECURE ROUNDS. As a consequence, three rounds with nCPA-secure round functions are not enough to get CPA-security. On the positive side, we show that one extra nCPA secure round is sufficient (and necessary) in the quasirandom setting. Note that for the translation of a security proof from quasi- to pseudorandom systems it is crucial that we can construct a distinguisher for the components from a distinguisher for the whole system. But here the components have a weaker security guarantee (i.e., nCPA) than what we prove for the whole system (i.e., CPA). So even when we have a CPA distinguisher for the four-round Feistel-network, we cannot construct a nCPA distinguisher for any round function. This is not just a shortcoming of the used approach, but indeed, in the pseudorandom setting the situation is different: we show that here four rounds are not enough to get CPA-security. To show this we construct a nCPA-secure PRF (under standard assumptions), such that the four-round Feistel-network with such round functions can easily be distinguished from a URP with only three adaptive queries.

QUASIRANDOMNESS DOES NOT IMPLY PSEUDORANDOMNESS. This phenomenon that some construction implies adaptive security for quasirandom but not for pseudorandom systems has already been proven [MP04, MPR06, Pie05, Pie06] for two simple constructions: the sequential composition $f \triangleright g(.) \stackrel{\text{def}}{=} g(f(.))$ and the parallel composition $f \star g(.) \stackrel{\text{def}}{=} f(.) \star g(.)$ (where $\star$ stands for any group operation). The security proofs from [MP04] in the quasirandom setting crucially use the fact that the sequential composition of two permutations is a URP whenever at least one of the permutations is a URP, similarly the parallel composition of two functions is a URF whenever one of the components is a URF. The Feistel-network does not have this nice property of being ideal whenever one of the components is ideal, and we have to work harder here (using a more general approach from [MPR06]). Our counterexample for the pseudorandom setting, i.e., a four-round Feistel-network with nCPA-secure PRFs as round functions that is not a CPA-secure PRP, is similar to the counterexamples in [Pie05, Ple05] for sequential and parallel composition. In [Ple05], it is shown that the sequential composition of arbitrarily many nCPA-secure PRFs will not be a CPA-secure PRF in general, whereas for the parallel composition only a counterexample with two components is known [Pie05]. For the Feistel-network we also could only find a counterexample for four rounds. So we cannot rule out the possibility that five or more rounds imply adaptive security. However, if this was the case, then it seems likely that – like for sequential composition [Mye04] – there is no black-box proof for this fact.[17]

WHAT ABOUT CCA-SECURITY? While it seems unlikely in the pseudorandom setting to achieve CPA-security (and hence also CCA-security) of the Feistel-network with nCPA-secure round functions, we show that (even) CCA-security can be achieved in the quasirandom setting. In particular, we show that the five-round Feistel-network with nCPA-secure QRFs is a CCA-secure QRP.

UNCONDITIONAL VS. CONDITIONAL COUNTEREXAMPLES. The counterexample showing that the three-round Feistel-network with a nCPA-secure PRF F in the second round is not adaptively secure is unconditional[18] and black-box; with this we mean that we can construct F starting from any (nCPA-secure) PRF via a reduction which uses this PRF only as a black-box.[19] As four rounds are enough to get adaptive security for quasirandom systems, there cannot be a black-box counterexample (like for three rounds) for the four (or more) round case. Thus it is not surprising that our counterexample for four rounds is not unconditional. It relies on any IDDH group.

3.3 The Three-Round Luby-Rackoff Cipher and Relaxations

Let us first state some results for the three-round Feistel-network.

Proposition 1. *For any*

$$\left(\mathsf{ATK}, \mathsf{ATK}'\right) \in \left\{(\mathsf{CPA}, \mathsf{CPA}), (\mathsf{nCCA}, \mathsf{nCPA}), (\mathsf{KPA}, \mathsf{KPA})\right\}$$

[17] Myers [Mye04] constructs an oracle relative to which there exist PRPs that are nCPA-secure, but for which their sequential composition is not a CPA-secure PRP. The idea behind this oracle is quite general, and we see no reason (besides being technically challenging) why one should not be able to construct a similar oracle for the Feistel-network, and thus also rule out a black-box proof for showing that the Feistel-network with nCPA-secure PRFs as round functions is a CPA secure PRP.

[18] I.e., we make no other assumption besides the trivially necessary one that pseudorandom functions, which are equivalent to one-way functions [HILL99, GGM86], exist at all.

[19] We build F from a pseudorandom involution (PRI), how to construct a PRI from a PRP (via a black-box reduction) has been shown in [NR02].

and random function $\mathbf{F}$

$$\Delta_q^{\mathsf{ATK}}(\psi_{2n}[\mathbf{FFF}], \mathbf{P}) \leq 3 \cdot \Delta_q^{\mathsf{ATK}'}(\mathbf{F}, \mathbf{R}) + 2 \cdot \frac{q^2}{2^{n+1}}. \tag{3.1}$$

The analogous statement also holds in the computational case, i.e., for any efficient random function F

$$\mathbf{Adv}_{t,q}^{\mathsf{ATK}}(\psi_{2n}[\mathsf{FFF}], \mathbf{P}) \leq 3 \cdot \mathbf{Adv}_{t',q}^{\mathsf{ATK}'}(\mathsf{F}, \mathbf{R}) + 2 \cdot \frac{q^2}{2^{n+1}}, \tag{3.2}$$

where $t' = t + \mathsf{poly}(q, n)$ *for some polynomial* poly *which accounts for the overhead implied by the reduction we make.*

The classical result of Luby and Rackoff [LR86], states that the Feistel-network with three independent PRF rounds is a CPA-secure PRP, i.e., (3.2) for $(\mathsf{ATK}, \mathsf{ATK}') = (\mathsf{CPA}, \mathsf{CPA})$.

Luby and Rackoff proved this result directly. One gets a simpler proof by first showing that the three-round Feistel-network with URFs $\mathbf{R}$ is a CPA-secure QRP as this is a purely information-theoretic statement. In particular it was shown in [Mau02] that[20]

$$\Delta_q^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RRR}], \mathbf{P}) \leq 2 \cdot \frac{q^2}{2^{n+1}}. \tag{3.3}$$

This bound also holds for nCCA distinguishers (see Appendix A.2.1). These results directly imply Proposition 1 by a standard hybrid argument. [21]

THE FIRST ROUND. Lucks showed [Luc96] (see also [NR99a]) that the first round in the three-round Luby-Rackoff cipher can be replaced with a much weaker primitive which only must provide some guarantee on

[20] This bound has been improved – using larger number of rounds – in a series of papers. The latest [Pat04] by Patarin claims (optimal) security up to $q \ll 2^n$ (and not just $q \ll 2^{n/2}$) queries, using five rounds (five rounds are also necessary to get such optimal security).

[21] The hybrid argument goes as follows for pseudorandom systems: let $(\mathsf{ATK}, \mathsf{ATK}') \in \{(\mathsf{CPA}, \mathsf{CPA}), (\mathsf{nCCA}, \mathsf{nCPA}), (\mathsf{KPA}, \mathsf{KPA})\}$ and assume that there is an efficient ATK-distinguisher A for $\psi_{2n}[\mathsf{FFF}]$ and $\mathbf{P}$. Then by (3.3), A will also distinguish $\psi_{2n}[\mathsf{FFF}]$ from $\psi_{2n}[\mathbf{RRR}]$. Consider the hybrids $H_0 = \psi_{2n}[\mathsf{FFF}], H_1 = \psi_{2n}[\mathbf{R}\mathsf{FF}], \ldots, H_3 = \psi_{2n}[\mathbf{RRR}]$. By the triangle inequality there is an $0 \leq i \leq 2$ (say $i = 1$) such that A can distinguish H_i from H_{i+1}. Now, the distinguisher which – with access to an oracle G (implementing either F or $\mathbf{R}$) – simulates $\mathsf{A}\diamond\psi_{2n}[\mathbf{R}\mathsf{GF}]$ and outputs the output of A is an efficient ATK'-distinguisher for F with the same advantage as A's advantage for H_1 and H_2. The corresponding argument also holds in the quasirandom setting.

the collision probability on the left half of the output (for any two fixed inputs). In particular, an almost pairwise independent permutation or a Feistel-round with an almost XOR-universal function will do.

THE THIRD ROUND. We show that (in addition) the third round function can be replaced by a KPA-secure function.

Proposition 2. *For any*

$$\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{nCPA}, \mathsf{KPA}\},$$

any random functions $\mathbf{F}$, $\mathbf{G}$, *and any permutation* $\mathbf{H}$

$$\begin{aligned} & \Delta_q^{\mathsf{ATK}}(\mathbf{H} \triangleright \psi_{2n}[\mathbf{FG}], \mathbf{P}) \\ \leq \quad & \Delta_q^{\mathsf{ATK}}(\mathbf{F}, \mathbf{R}) + 2 \cdot \Delta_q^{\mathsf{KPA}}(\mathbf{G}, \mathbf{R}) + \mathsf{coll}_q({}_L\mathbf{H}) + \frac{q^2}{2^n}. \end{aligned} \tag{3.4}$$

The analogous statement also holds in the computational case: for any $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{nCPA}, \mathsf{KPA}\}$, *any efficient random functions* F, G, *and any efficient permutation* H

$$\begin{aligned} & \mathbf{Adv}_{t,q}^{\mathsf{ATK}}(\mathsf{H} \triangleright \psi_{2n}[\mathsf{FG}], \mathbf{P}) \\ \leq \quad & \mathbf{Adv}_{t',q}^{\mathsf{ATK}}(\mathsf{F}, \mathbf{R}) + \mathbf{Adv}_{t',q}^{\mathsf{KPA}}(\mathsf{G}, \mathbf{R}) + \mathsf{coll}_q({}_L\mathsf{H}) + \frac{q^2}{2^{n+1}}, \end{aligned} \tag{3.5}$$

where $t' = t + \mathsf{poly}(q, n)$ *for some polynomial* poly *which accounts for the overhead implied by the reduction we make.*

Let us stress that (3.5) does *not* directly follow from (3.4).[22] The proof of Proposition 2 is given in Appendix A.2.1.

THE KPA CASE. We relax the construction further for $\mathsf{ATK} = \mathsf{KPA}$ by showing that the first round can be removed completely (as opposed to when $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{nCPA}\}$[23]). Moreover, the round functions can be replaced by a *single* instantiation of a KPA-secure function. Note that if one in addition interchange the left and the right part of the output, the resulting construction is an involution, i.e., has the structural property of being self inverse. This result also generalizes Lemma 2.2 of [MT05] which states that the two round Feistel-network with CPA-secure PRFs is a KPA-secure PRP.

[22] The reason why a reduction – like the simple argument to show that Proposition 1 follows from (3.3) – fails here, is that the KPA-security guarantee for one of the components is weaker than the CPA-security for the whole construction. But fortunately the *proof* of (3.4) is such that it easily translates to the pseudorandom setting.

[23] $\psi_{2n}[\mathbf{RR}]$ can be distinguish from $\mathbf{P}$ with two non-adaptively chosen queries: query $0^n \| 0^n \mapsto {}_Ly \| {}_Ry$ and $0^n \| 1^n \mapsto {}_Ly' \| {}_Ry'$, and output 1 if ${}_Ry \oplus {}_Ry' = 1^n$ and 0 otherwise.

Proposition 3. *For any random function* $\mathbf{F}$

$$\Delta_q^{\mathsf{KPA}}(\psi_{2n}[\mathbf{F}^2], \mathbf{P}) \leq \Delta_{2q}^{\mathsf{KPA}}(\mathbf{F}, \mathbf{R}) + 4 \cdot \frac{q^2}{2^{n+1}}. \tag{3.6}$$

The analogous statement also holds in the computational case: for any (in particular efficient) random function $\mathbf{F}$

$$\mathbf{Adv}_{t,q}^{\mathsf{KPA}}(\psi_{2n}[\mathbf{F}^2], \mathbf{P}) \leq \mathbf{Adv}_{t',2q}^{\mathsf{KPA}}(\mathbf{F}, \mathbf{R}) + 4 \cdot \frac{q^2}{2^{n+1}}, \tag{3.7}$$

where $t' = t + \mathsf{poly}(q, n)$ *for some polynomial* poly *which accounts for the overhead implied by the reduction we make.*

The proof is given in Appendix A.2.1. Note that unlike in the previous propositions, here we do not require the round function $\mathbf{F}$ to be efficient in the computational case (the reason is that in the proof we do not need the distinguisher to simulate any round function).

THE SECOND ROUND IS CRUCIAL. The following proposition states that to achieve CPA-security in general with the three-round Luby-Rackoff cipher, it is not sufficient that the second round function is nCPA-secure.

There exists a nCPA-secure function, such that the three-round Feistel-network with this function in the second, and any random functions in the first and third round, is not CPA-secure.

Proposition 4. *There exists a random function* $\mathbf{F}$ *such that for any random functions* $\mathbf{G}$ *and* $\mathbf{G}'$ *(in particular for* $\mathbf{G} = \mathbf{R}$ *and* $\mathbf{G}' = \mathbf{R}$*)*

$$\Delta_q^{\mathsf{nCPA}}(\mathbf{F}, \mathbf{R}) \leq \frac{q^2}{2^{n-1}} \quad \textit{and} \quad \Delta_2^{\mathsf{CPA}}(\psi_{2n}[\mathbf{G}\mathbf{F}\mathbf{G}'], \mathbf{P}) \geq 1 - 2^{-n+1}.$$

The analogous statement also holds in the computational case: (informal) there is a nCPA*-secure PRF* F *such that* $\psi_{2n}[\mathbf{G}\mathsf{F}\mathbf{G}']$ *is not a* CPA*-secure PRP for any (not necessarily efficient) functions* $\mathbf{G}$ *and* $\mathbf{G}'$*.*

Proof. Let us first consider the quasirandom statement. Let $\mathbf{I}$ be a uniform random involution, i.e., $\mathbf{I}(\mathbf{I}(x)) = x$ for all x. Now, $\mathbf{F}$ is simply defined as $\mathbf{F}(x) = x \oplus \mathbf{I}(x)$, note that this $\mathbf{F}$ satisfies $\mathbf{F}(x) = \mathbf{F}(x \oplus \mathbf{F}(x))$ for all x.

The nCPA-security of $\mathbf{F}$ (which is simply the nCPA-security of $\mathbf{I}$) can be bounded as stated in the proposition by standard techniques (see Appendix A.2.1). Furthermore, $\psi_{2n}[\mathbf{G}\mathbf{F}\mathbf{G}']$ can easily be distinguished from

$\mathbf{P}$ with two adaptively chosen queries as follows. After a first query $0^n \| 0^n$, the output ${}_L Y \| Z$ contains the output Z of the internal function $\mathbf{F}$. Now make a second query $0^n \| Z$. If the (unknown) input to $\mathbf{F}$ in the first query was some value V, then in this query it will be $V \oplus Z$, and as $\mathbf{F}$ satisfies $\mathbf{F}(V) = \mathbf{F}(V \oplus \mathbf{F}(V)) = \mathbf{F}(V \oplus Z)$, the output of $\mathbf{F}$ will again be Z, and the overall output will be $({}_L Y \oplus Z) \| Z$. The proposition follows as the output of $\mathbf{P}$ will satisfy such a relation with probability at most $\frac{1}{2^n} + \frac{2^n - 1}{2^{2n} - 1} \leq 2^{-n+1}$.

The corresponding statement for the pseudorandom setting is proven almost identically. The only difference is that we need to use a CPA-secure pseudorandom involution I instead of the uniform random involution $\mathbf{I}$. It is shown in [NR02] how to construct a pseudorandom involution from any CPA-secure PRF. □

The next proposition states that the network will in general not (even) be nCPA-secure when the second round function is only secure against KPAs.

Proposition 5. *There exists a random function* $\mathbf{F}$ *such that for any random functions* $\mathbf{G}$ *and* $\mathbf{G}'$

$$\Delta_q^{\mathsf{KPA}}(\mathbf{F}, \mathbf{R}) \leq \frac{q^2}{2^{n+1}}, \quad \textit{and} \quad \Delta_2^{\mathsf{nCPA}}(\psi_{2n}[\mathbf{GFG}'], \mathbf{P}) \geq 1 - 2^{-n}.$$

The analogous statement also holds in the computational case: (informal) there is a KPA*-secure PRF* F *such that* $\psi_{2n}[\mathbf{G}\mathsf{F}\mathbf{G}']$ *is not a* nCPA*-secure PRP for any (not necessarily efficient) functions* $\mathbf{G}$ *and* $\mathbf{G}'$.

Proof. Let us first consider the statement in the quasirandom setting. Let $\mathbf{F}$ be a URF which ignores the first input bit, i.e., for all $x \in \{0,1\}^{n-1}$ we have $\mathbf{F}(0\|x) = \mathbf{F}(1\|x)$. The KPA-security of $\mathbf{F}$ follows from the fact that $\mathbf{F}$ looks completely random unless we happen to query two queries of the form $0\|x$ and $1\|x$. By the birthday bound the probability that this happens after q queries is at most $\frac{q^2}{2^{n+1}}$ (see Appendix A.2.1). Furthermore, $\psi_{2n}[\mathbf{GFG}']$ can be distinguished from $\mathbf{P}$ with two non-adaptively chosen queries. For instance on input $0^n \| 0^n$ and $0^n \| (1 \| 0^{n-1})$, the right half of the output will be identical. However, for $\mathbf{P}$ this only happens with probability at most $\frac{2^n - 1}{2^{2n} - 1} \leq 2^{-n}$.

The corresponding statement in the pseudorandom setting is proven exactly as above, except that we have to use a PRF F instead of $\mathbf{F}$. □

3.4 Four nCPA-Secure Feistel-Rounds (Quasirandom Case)

The following theorem shows that the four-round Feistel-network with nCPA-secure QRFs is a CPA-secure QRP. This is also the best possible as in Section 3.3 we showed that four rounds are also necessary. The theorem is even stronger as the third and fourth round function must only be KPA-secure QRFs.

Theorem 1. *For any random functions* $\mathbf{F}$ *and* $\mathbf{G}$

$$\Delta_q^{\mathsf{CPA}}(\psi_{2n}[\mathbf{FFGG}], \mathbf{P}) \leq 2 \cdot \left(\Delta_q^{\mathsf{nCPA}}(\mathbf{F}, \mathbf{R}) + \Delta_q^{\mathsf{KPA}}(\mathbf{G}, \mathbf{R})\right) + \frac{q^2}{2^{n-2}}.$$

The bound is more tight than the original one given in [MOPS06b]. The proof uses techniques from [MPR06] and is given in Appendix A.2.2.

3.5 Four nCPA-Secure Feistel-Rounds (Pseudorandom Case)

In this section, we again investigate the CPA-security of the four-round Feistel-network with nCPA-secure round functions, but this time for *pseudorandom* systems. We show that here the situation is dramatically different from the quasirandom setting by constructing a nCPA-secure PRF where the four-round Feistel-network with this PRF as round function is not CPA-secure.

Theorem 2. *(Informal) There exists a* nCPA*-secure PRF* F *based on any IDDH group such that the four-round Feistel-network where each round is instantiated with* F *(with independent keys) is* not *a* CPA *secure pseudorandom permutation.*

This theorem follows from Lemma 1 below which states that there exist nCPA-secure PRFs $\mathsf{F}_1, \mathsf{F}_2, \mathsf{F}_3$ such that the left half of the *three* round Feistel-network ${}_L\psi_{2n}[\mathsf{F}_1\mathsf{F}_2\mathsf{F}_3]$ is not a CPA-secure PRF. This implies that also $\psi_{2n}[\mathsf{F}_1\mathsf{F}_2\mathsf{F}_3\mathsf{G}]$ is not a CPA-secure PRP for any G (and thus proves Theorem 2) as follows. By the so-called PRF/PRP Switching Lemma (for instance see [Sho04]) any CPA-secure PRP P is also a CPA-secure PRF. Clearly, then also ${}_L\mathsf{P}$ must be a CPA-secure PRF. Now, by Lemma 1

${}_L\psi_{2n}[\mathsf{F}_1\mathsf{F}_2\mathsf{F}_3] = {}_L\psi_{2n}[\mathsf{F}_1\mathsf{F}_2\mathsf{F}_3\mathsf{G}]$ is not a CPA-secure PRF, so $\psi_{2n}[\mathsf{F}_1\mathsf{F}_2\mathsf{F}_3\mathsf{G}]$ cannot be a CPA-secure PRP.[24]

Lemma 1. *There exist* nCPA*-secure PRFs* F_1, F_2, F_3 *based on any IDDH group such that* ${}_L\psi_{2n}[\mathsf{F}_1\mathsf{F}_2\mathsf{F}_3]$ *is not a* CPA*-secure PRF: it can be distinguished efficiently from a URF with only three (adaptive) queries with high probability.*[25]

OUTLINE FOR THIS SECTION. In Section 3.5.1 we first show the construction from [Ple05] of a nCPA-secure PRF whose sequential composition will not be CPA-secure. This extremely simple and intuitive construction is the basis for the (more involved) counterexample for the Feistel-network (i.e., Lemma 1) given in Section 3.5.2.

3.5.1 Counterexample for Sequential Composition

In this section, we construct a simple nCPA-secure PRF F, but where the sequential composition of (arbitrary many) such F (with independent keys) is not CPA-secure. This function was stated in [Ple05]. Here we give the security proof.

F is based on an IDDH group $(G, g) = \mathcal{G}(n)$ where the elements of the group can be efficiently and densely encoded into $\{0,1\}^n$ (with dense we mean that all but a negligible fraction of the strings should correspond to an element of the group).[26] For example we can let G be a subgroup of prime order φ of $\mathbb{Z}_\rho^*$, where ρ is a safe prime (i.e., $2\varphi + 1$) and φ is close to 2^n ([Dam04] describes how to embed such a G into $\{0,1\}^n$).

Let $[.] : \mathcal{G}(n) \rightarrow \{0,1\}^n$ denote an (efficient) embedding of $\mathcal{G}$ into bitstrings (to save on notation we let $[a,b]$ denote the concatenation of $[a]$ and $[b]$). Let $\mathsf{R} : \mathcal{K} \times \{0,1\}^{4n} \rightarrow \mathbb{Z}_\varphi^4$ be any nCPA-secure PRF. Now consider

---

[24] The lemma talks about three different F_i's (and in the proof we really construct a different F_i for every round), but the theorem is stated for a single F. This does not really make a difference. For example this single F can be defined as behaving like F_i with probability $1/3$ for $i \in \{1,2,3\}$. Then with constant probability 3^{-3} the $\psi_{2n}[\mathsf{FFF}]$ behaves like $\psi_{2n}[\mathsf{F}_1\mathsf{F}_2\mathsf{F}_3]$.

[25] This is the only result in the thesis which we do not know how to prove a uniform version of. But one can do so from a somewhat stronger primitive than an IDDH group. Informally, this primitive is an IDDH group but where the attacker can choose the generator (in the challenge).

[26] For this construction we actually do not need this embedding, we could define F directly over the group. But we will need it (or more precisely, the fact that if X is in the range of F, also $X \oplus R$ for a random bitstring R is in the range with overwhelming probability) when we extend this construction to get the counterexample for the Feistel-network in the next section.

the following definition of a nCPA-secure PRF $\mathsf{F} : \{0,1\}^{4n} \to \{0,1\}^{4n}$ with secret key $(k \in \mathcal{K}, x \in \mathbb{Z}_\varphi^*)$.

The first thing F does on input $(\alpha, \beta, \gamma, \delta) \in \{0,1\}^{4n}$ is to generate some pseudorandom values using R, i.e.,

$$(r_1, r_2, r_3, r_4) \leftarrow \mathsf{R}(k, \alpha, \beta, \gamma, \delta). \tag{3.8}$$

Further, if there exists $(a, b, c, d) \in G^4$ s.t. $\alpha = [a], \beta = [b], \gamma = [c], \delta = [d]$ then F outputs (here x^{-1} is the inverse of x in $\mathbb{Z}_\varphi^*$)

$$\mathsf{F}([a, b, c, d]) \to ([a^{xr_1}, b^{r_1}, c^{x^{-1}r_2}, d^{r_2}]), \tag{3.9}$$

with r_1, r_2 generated as in (3.8). On the remaining inputs (which are a negligible fraction of $\{0,1\}^{4n}$) F outputs just the pseudorandom values $[g^{r_1}, g^{r_2}, g^{r_3}, g^{r_4}]$.

Now consider the cascade $\mathsf{F}' \triangleright \mathsf{F}'' \triangleright \mathsf{F}'''$ of three independent F's (with corresponding keys (x_1, k_1), (x_2, k_2), and (x_3, k_3)). Make a first query $[g, g, g, g]$

$$\mathsf{F}' \triangleright \mathsf{F}'' \triangleright \mathsf{F}'''([g, g, g, g]) \to [g^{x_1 x_2 x_3 r}, g^r, g^{x_1^{-1} x_2^{-1} x_3^{-1} r'}, g^{r'}].$$

Then the output will have the form $g^{x_1 x_2 x_3 r}, g^r, g^{x_1^{-1} x_2^{-1} x_3^{-1} r'}, g^{r'}$ for some r, r'. Now exchange the right and the left half of this output and use it as the second query

$$\mathsf{F}' \triangleright \mathsf{F}'' \triangleright \mathsf{F}'''([g^{x_1^{-1} x_2^{-1} x_3^{-1} r'}, g^{r'}, g^{x_1 x_2 x_3 r}, g^r]) \to [g^{r''}, g^{r''}, g^{r'''}, g^{r'''}]$$

so the output is of the form $[u, u, v, v]$ for some u, v and thus can be distinguished from random. Therefore $\mathsf{F}' \triangleright \mathsf{F}'' \triangleright \mathsf{F}'''$ is not a CPA-secure PRF. This proves that the sequential composition of nCPA-secure PRFs does not yield a CPA-secure function in general. Note that this distinguishing attack works for any number of rounds, not just three. The following lemma states that F is a nCPA-secure PRF if $\mathcal{G}$ is an IDDH group, R is a nCPA-secure PRF, and the encoding $[.]$ is dense (as then $(2^n - |G|)/2^n$ is negligible).

Lemma 2. *For* F *over* $(G, g) = \mathcal{G}(n)$ *we have*

$$\begin{aligned} & \mathbf{Adv}_{q,t}^{\mathsf{nCPA}}(\mathsf{F}, \mathbf{R}) \\ \leq \quad & 6 \cdot q \cdot \mathbf{Adv}_{t'}^{IDDH}(G, g) + \mathbf{Adv}_{q,t'}^{\mathsf{nCPA}}(\mathsf{R}, \mathbf{R}) + 4 \cdot q \cdot \frac{2^n - |G|}{2^n}, \end{aligned} \tag{3.10}$$

where $t' = t + \mathsf{poly}(q, n)$ *for some polynomial* poly *which accounts for the overhead implied by the reduction we make.*

Proof. The lemma follows from the Lemmata 3 and 4 below. $\square$

Instead of proving Lemma 2 directly, we consider a function $\widetilde{\mathsf{F}}^{\mathbf{R}'} : \mathbb{Z}_{|G|} \times G^4 \to G^4$ (defined below) which will be easier to analyze. $\widetilde{\mathsf{F}}^{\mathbf{R}'}$ is defined almost like F but with two differences. First, the PRF R used by F is replaced by a uniformly random function $\mathbf{R}'$, and second we do not embed the output of $\widetilde{\mathsf{F}}^{\mathbf{R}'}$ into $\{0,1\}^n$ as in F (using the embedding [.]).

We define $\widetilde{\mathsf{F}}^{\mathbf{R}'}$, with key $x \in Z_{|G|}$ and oracle access to $\mathbf{R}' : G^4 \to Z^2_{|G|}$ as

$$\widetilde{\mathsf{F}}^{\mathbf{R}'}(x, a, b, c, d) \to (a^{xr}, b^r, c^{x^{-1}r'}, d^{r'}) \quad \text{where} \quad \mathbf{R}'(a, b, c, d) \to (r, r').$$

By the following lemma, distinguishing $\widetilde{\mathsf{F}}^{\mathbf{R}'}$ from a URF is basically as hard as distinguishing F.

Lemma 3. *For URFs $\mathbf{R} : G^4 \to G^4$, $\mathbf{R}' : G^4 \to Z^2_{|G|}$, $\mathbf{R}'' : \{0,1\}^{4n} \to \{0,1\}^{4n}$ and R from the definition of F,*

$$\mathbf{Adv}^{\mathsf{nCPA}}_{q,t}(\mathsf{F}, \mathbf{R}'') \le \mathbf{Adv}^{\mathsf{nCPA}}_{q,t'}(\widetilde{\mathsf{F}}^{\mathbf{R}'}, \mathbf{R}) + \mathbf{Adv}^{\mathsf{nCPA}}_{q,t'}(\mathsf{R}, \mathbf{R}') + 4 \cdot q \cdot \frac{2^n - |G|}{2^n},$$

where $t' = t + \mathsf{poly}(q, n)$ for some polynomial poly which accounts for the overhead implied by the reduction we make.

Proof. Let $\mathsf{F}^{\mathbf{R}'}$ be F, but where one uses the URF $\mathbf{R}'$ instead of R. Then

$$\mathbf{Adv}^{\mathsf{nCPA}}_{q,t}(\mathsf{F}, \mathbf{R}'') \le \mathbf{Adv}^{\mathsf{nCPA}}_{q,t}(\mathsf{F}^{\mathbf{R}'}, \mathbf{R}'') + \mathbf{Adv}^{\mathsf{nCPA}}_{q,t'}(\mathsf{R}, \mathbf{R}').$$

$\mathsf{F}^{\mathbf{R}'}$ only differs from $\widetilde{\mathsf{F}}^{\mathbf{R}'}$ by the use of the embedding [.], as for a random $x \in G$, $[x]$ is $|G|/2^n$ close to uniform we further get

$$\mathbf{Adv}^{\mathsf{nCPA}}_{q,t}(\mathsf{F}^{\mathbf{R}'}, \mathbf{R}'') \le \mathbf{Adv}^{\mathsf{nCPA}}_{q,t'}(\widetilde{\mathsf{F}}^{\mathbf{R}'}, \mathbf{R}) + 4 \cdot q \cdot \frac{2^n - |G|}{2^n}.$$

$\square$

We now bound the indistinguishability of $\widetilde{\mathsf{F}}^{\mathbf{R}'}$ from random in terms of the maximal IDDH-advantage in (G, g).

Lemma 4.

$$\mathbf{Adv}^{\mathsf{nCPA}}_{q,t}(\widetilde{\mathsf{F}}^{\mathbf{R}'}, \mathbf{R}) \le 6 \cdot q \cdot \mathbf{Adv}^{IDDH}_{t'}(G, g),$$

where $t' = t + \mathsf{poly}(q, n)$ for some polynomial poly which accounts for the overhead implied by the reduction we make.

Proof. First we observe that for any non-uniform nCPA-distinguisher A for $\widetilde{\mathsf{F}}^{\mathbf{R}'}$, there is a deterministic nCPA-distinguisher A′ for $\widetilde{\mathsf{F}}^{\mathbf{R}'}$ with[27]

$$|\mathsf{A}'| \leq |\mathsf{A}| + O(q \cdot \log(|G|)) = |\mathsf{A}| + \mathsf{poly}(q, n)$$

that issues the same number (i.e., q) of queries, has at least the same nCPA-distinguishing advantage, and additionally "knows" all the discrete logarithms to (the publicly known) basis g of its q inputs, i.e., when A′ makes a query (a_1, a_2, a_3, a_4) where $a_i = g^{z_i}$ then the $z_1, \ldots, z_4$ are somehow hardwired into A′.[28]

The task of our distinguisher A′ is to distinguish q quadruples with uniform distribution over G^4 from q quadruples of the form

$$(a_1^{xr}, a_2^{r}, a_3^{x^{-1}r'}, a_4^{r'}) = (g^{z_1 xr}, g^{z_2 r}, g^{z_3 x^{-1} r'}, g^{z_4 r'}), \tag{3.11}$$

where $(a_1, \ldots, a_4)$ is a query chosen (non-adaptively) by A′ and x, r, r' are uniformly random (note that x, which is part of the key of $\widetilde{\mathsf{F}}^{\mathbf{R}'}$, is the same for all q quadruples, but the r, r' are independently generated by $\mathbf{R}'$ for each of the q quadruples). As we do assume that A′ knows the $z_1, \ldots, z_4$, this is equivalent[29] to distinguish

$$(g^{xr}, g^{r}, g^{x^{-1}r'}, g^{r'}) \quad \text{from} \quad (g^{r}, g^{r'}, g^{r''}, g^{r'''}), \tag{3.12}$$

where x and r, r', r'', r''' are uniformly random.

We make the task for A′ even simpler and additionally provide g^x and $g^{x^{-1}}$, i.e., A′ must distinguish

$$(g^{x}, g^{x^{-1}}, g^{xr}, g^{r}, g^{x^{-1}r'}, g^{r'}) \quad \text{from} \quad (g^{x}, g^{x^{-1}}, g^{r}, g^{r'}, g^{r''}, g^{r'''}). \tag{3.13}$$

Clearly the task given by (3.13) is at most as difficult as (3.12) as one can always ignore the first two elements. We call the corresponding problem

[27] As we require that group operations can be done in time polynomial in n, the representation of elements of $|G|$ — which is at least $\log(|G|)$ bits long — must also be polynomial (as otherwise one could not even read an element in polynomial time).

[28] This observation may seem silly, but this "knowledge" seems necessary in the following reduction. This is also the reason why we can only prove this lemma in the non-uniform setting.

[29] Here and below with problem A being "equivalent" or "easier" than problem B, we mean that if there is a distinguisher A with advantage ϵ for B, then there's a distinguisher $\tilde{\mathsf{A}}$ with the same advantage ϵ for A, where $|\tilde{\mathsf{A}}| \leq |\mathsf{A}| + \mathsf{poly}(q, n)$.

the *extended* DDH (EDDH) and let the maximal advantage for EDDH in (G, g) be denoted as

$$\mathbf{Adv}_t^{EDDH}(G,g) \stackrel{\text{def}}{=} \max_{\mathsf{A},|\mathsf{A}|\le t} \Bigg| \Pr_{x,r,r'}\left[\mathsf{A}(g^x, g^{x^{-1}}, g^{xr}, g^r, g^{x^{-1}r'}, g^{r'}) \to 1\right] - \Pr_{x,r,r',r'',r'''}\left[\mathsf{A}(g^x, g^{x^{-1}}, g^r, g^{r'}, g^{r''}, g^{r'''}) \to 1\right] \Bigg|,$$

where the maximum is taken over all distinguishers (for EDDH) of size at most t. Thus distinguishing $\widetilde{\mathsf{F}}^{\mathbf{R}'}$ from $\mathbf{R}$ is at most as hard as distinguishing

$$(g^x, g^{x^{-1}}, g^{xr_1}, g^{r_1}, g^{x^{-1}r'_1}, g^{r'_1}), \ldots, (g^x, g^{x^{-1}}, g^{xr_q}, g^{r_q}, g^{x^{-1}r'_q}, g^{r'_q}) \tag{3.14}$$

from

$$(g^x, g^{x^{-1}}, g^{r_1}, g^{r'_1}, g^{r''_1}, g^{r'''_1}), \ldots, (g^x, g^{x^{-1}}, g^{r_q}, g^{r'_q}, g^{r''_q}, g^{r'''_q}), \tag{3.15}$$

where x and all the $r_i, \ldots, r'''_i$ are uniformly random. We can use a hybrid argument to bound this distinguishing advantage in terms of the maximal advantage for EDDH in (G, g). Let H_i denote the i-th hybrid given by

$$\begin{gathered}
(g^x, g^{x^{-1}}, g^{xr_1}, g^{r_1}, g^{x^{-1}r'_1}, g^{r'_1}) \\
\vdots \\
(g^x, g^{x^{-1}}, g^{xr_i}, g^{r_i}, g^{x^{-1}r'_i}, g^{r'_i}) \\
(g^x, g^{x^{-1}}, g^{r_{i+1}}, g^{r'_{i+1}}, g^{r''_{i+1}}, g^{r'''_{i+1}}) \\
\vdots \\
(g^x, g^{x^{-1}}, g^{r_q}, g^{r'_q}, g^{r''_q}, g^{r'''_q}).
\end{gathered}$$

Note that the distribution (3.15) is just H_0 and the distribution (3.14) is H_q. Thus there is a j such that A' can distinguish H_{j-1} from H_j with advantage at least ϵ/q. Now consider the following distinguisher A'' for EDDH: on input $(a_1, \ldots, a_6)$ (which always satisfies $a_1 = g^x$ and $a_2 =$

$g^{x^{-1}}$ for a random x) A'' generates the distribution

$$\begin{gathered}(g^x, g^{x^{-1}}, g^{xr_1}, g^{r_1}, g^{x^{-1}r'_1}, g^{r'_1}) \\ \vdots \\ (g^x, g^{x^{-1}}, g^{xr_{j-1}}, g^{r_{j-1}}, g^{x^{-1}r'_{j-1}}, g^{r'_{j-1}}) \\ (g^x, g^{x^{-1}}, a_3, a_4, a_5, a_6) \\ (g^x, g^{x^{-1}}, g^{r_{j+1}}, g^{r'_{j+1}}, g^{r''_{j+1}}, g^{r'''_{j+1}}) \\ \vdots \\ (g^x, g^{x^{-1}}, g^{r_q}, g^{r'_q}, g^{r''_q}, g^{r'''_q})\end{gathered}$$

and runs A' on this input.[30] As the above distribution is equivalent to H_j if $(a_1, \ldots, a_6)$ is of the form as shown by the left side of (3.13), and H_{j-1} if it's of the form on the right side of (3.13), we conclude that A'' has the same advantage ϵ/q for EDDH as A' had in distinguishing H_{j-1} from H_j, so

$$\mathbf{Adv}^{\mathsf{nCPA}}_{q,t}(\widetilde{\mathsf{F}}^{\mathbf{R}'}, \mathbf{R}) \leq q \cdot \mathbf{Adv}^{EDDH}_{t'}(G, g).$$

To conclude the proof of the lemma, we must now reduce IDDH to EDDH:

Claim 1.

$$\mathbf{Adv}^{EDDH}_{t}(G, g) \leq 6 \cdot \mathbf{Adv}^{IDDH}_{t'}(G, g),$$

where $t' = t + \mathsf{poly}(q, n)$ for some polynomial poly *which accounts for the overhead implied by the reduction we make.*

Proof. First we show that EDDH is equivalent to deciding whether $z = xy$ or $z = r$ in the tuple $(g, g^{x^{-1}}, g^x, g^y, g^z)$, referred to as DDH$^-$ (up to a factor of 2). Reducing EDDH to DDH$^-$ is trivial, as we can ignore the unnecessary components from an EDDH tuple. For the reverse direction, we examine the following distributions:

$$\begin{aligned} H_0 &= (g, g^{x^{-1}}, g^x, g^y, g^{xy}, g^{y'}, g^{x^{-1}y'}) \\ H_1 &= (g, g^{x^{-1}}, g^x, g^y, g^c, g^{y'}, g^{c'}) \\ H_2 &= (g, g^{x^{-1}}, g^x, g^y, g^r, g^{y'}, g^{r'}), \end{aligned}$$

[30]Note that A'' really can efficiently sample this distribution as it knows g^x and $g^{x^{-1}}$ (which are given by a_1 and a_2 respectively).

where x, y, y', r, r' are chosen uniformly at random and with probability $1/2$ $(c = xy \wedge c' = r')$. In the other half of the cases $(c = r \wedge c' = x^{-1}y')$.

$$\begin{aligned}
\mathbf{Adv}_t^{EDDH}(G,g) &\leq \mathbf{Adv}_t(H_0,H_2) &(3.16)\\
&\leq \mathbf{Adv}_t(H_0,H_1)+\mathbf{Adv}_t(H_1,H_2) &(3.17)\\
&\leq 2\cdot\mathbf{Adv}_{t'}^{DDH^-}(G,g). &(3.18)
\end{aligned}$$

Step (3.17) follows by applying the triangle inequality. Given a distinguisher $\mathsf{A}_{0,1}$, that is able to distinguish between H_0 and H_1, we can build a distinguisher for DDH. To decide for a tuple $(g, g^{a^{-1}}, g^a, g^b, g^c)$ if $c = ab$ or $c = r$, it first chooses r' uniformly at random and generates $g^{r'}, g^{a^{-1}r'}$. Then with probability $1/2$ it returns the answer $\mathsf{A}_{0,1}$ gives to the input $(g, g^{a^{-1}}, g^a, g^b, g^c, g^{r'}, g^{a^{-1}r'})$ and otherwise $\mathsf{A}_{0,1}$'s response to the input $(g, g^a, g^{a^{-1}}, g^{r'}, g^{a^{-1}r'}, g^b, g^c,)$. Hence $\mathbf{Adv}_t(H_0, H_1) \leq \mathbf{Adv}_{t'}^{DDH^-}(G, g)$. An analogous argument can be used to tell H_1 and H_2 apart and therefore (3.18) follows.

In our next step we bound the distinguishing advantage of DDH$^-$ by demonstrating that

$$\mathbf{Adv}_t^{DDH^-}(G,g) \leq \mathbf{Adv}_{t'}^{DDH}(G,g) + 2\cdot\mathbf{Adv}_{t'}^{IDDH}(G,g). \tag{3.19}$$

Consider the following distributions

$$\begin{aligned}
D_0 &= (g, g^{a^{-1}}, g^a, g^b, g^{ab})\\
H_0 &= (g, g^r, g^a, g^b, g^{ab})\\
H_1 &= (g, g^r, g^a, g^b, g^c)\\
D_1 &= (g, g^{a^{-1}}, g^a, g^b, g^c).
\end{aligned}$$

We want to bound the distinguishing advantage of D_0 and D_1. To this purpose we use the triangle inequality

$$\begin{aligned}
\mathbf{Adv}_t^{DDH^-}(G,g) &= \mathbf{Adv}_t(D_0,D_1) &(3.20)\\
&\leq \mathbf{Adv}_t(D_0,H_0)+\mathbf{Adv}_t(H_0,H_1)+\mathbf{Adv}_t(H_1,D_1).
\end{aligned}$$

When distinguishing H_0 from H_1, we have to solve a plain DDH problem, as g^r carries no information on a and b. Hence

$$\mathbf{Adv}_t(H_0,H_1) \leq \mathbf{Adv}_{t'}^{DDH}(G,g). \tag{3.21}$$

Moreover g^b, g^c do not help distinguishing H_1 from D_1, and thus

$$\mathbf{Adv}_t(H_1, D_1) \leq \mathbf{Adv}_{t'}^{IDDH}(G, g). \tag{3.22}$$

We encounter a similar situation comparing the first two distributions. Since g^b, g^{ab} can be generated easily when knowing g^a, it follows that

$$\mathbf{Adv}_t(D_0, H_0) \leq \mathbf{Adv}_{t'}^{IDDH}(G, g). \tag{3.23}$$

Combining equations (3.21) – (3.23) proves (3.19). Equations (3.16) – (3.19) conclude the proof of the claim. △

□

3.5.2 Counterexample for the Four-Round Feistel

The Feistel-network can be seen as a sequential composition of the round functions, but where one additionally XORs the input to the i-th round function to the output of the $(i+1)$-th round function. So it is not surprising that we can use F_i's similar to the F from the previous section to prove Lemma 1. But the $\mathsf{F}_1, \mathsf{F}_2$, and F_3 (from the statement of the lemma) are a bit more complicated as we have to "work around" these additional XORs. Like F, each F_i has a $k_i \in \mathcal{K}$ as part of its secret key. Moreover F_1 has a $x \in \mathbb{Z}_\varphi^*$ and $s, t \in \{0,1\}^n$, F_2 has a $y \in \mathbb{Z}_\varphi^*$, and F_3 a $z \in \mathbb{Z}_\varphi^*$ as keys. On input $(\alpha, \beta, \gamma, \delta) = [a, b, c, d]$ the F_i's are defined as (with the r_i's generated as in (3.8))

$$\mathsf{F}_1([a,b,c,d]) = \begin{cases} [g^{xr_1}, g^{r_1}], s, t & \text{if } [a,b,c,d] = [0,0,0,0]; \\ [0,0,0,0] & \text{elseif } c = d^x; \\ [g^{xr_1}, g^{r_1}, ([\gamma \oplus s]^{-1})^{x^{-1}r_2}, ([\delta \oplus t]^{-1})^{r_2}] & \text{elseif } [a,b] = [0,0]; \\ [g^{r_1}, g^{r_2}, g^{r_3}, g^{r_4}] & \text{otherwise.} \end{cases}$$

$$\mathsf{F}_2([a,b,c,d]) = [c^{y^{-1}r_1}, d^{r_1}, a^{yr_2}, b^{r_2}]$$

$$\mathsf{F}_3([a,b,c,d]) = \begin{cases} [0,0,0,0] & \text{if } b^z = a; \\ [a^{z^{-1}r_1}, b^{r_1}, c^{zr_2}, d^{r_2}] & \text{otherwise.} \end{cases}$$

Proof of Lemma 1. The lemma follows from Claim 2 and 3 below. □

Claim 2. *One can distinguish* ${}_L\psi_{2n}[\mathsf{F}_1\mathsf{F}_2\mathsf{F}_3]$ *from a URF with three adaptively chosen queries with advantage almost* 1.

Proof (sketch). In Figure 3.2 we demonstrate an adaptive three query distinguishing attack on ${}_L\psi_{2n}[\mathsf{F}_1\mathsf{F}_2\mathsf{F}_3]$. In the figure, values which are not relevant for the attack are denoted by $*$. All r_i' values are random, but not necessarily equal to a random value generated by a round function (i.e., as in (3.8)).[31] To see that this is a legal attack note that every query Q_i can be computed from the previous output O_{i-1}. That the values will really have the form as described in the attack can be verified from the definition of the F_i's.[32] Since the third output starts with $[0,0]$ it can be distinguished from a random output with high probability. □

Claim 3. $\mathsf{F}_1, \mathsf{F}_2$, *and* F_3 *are* nCPA-*secure PRFs if* $\mathcal{G}$ *is an IDDH group.*

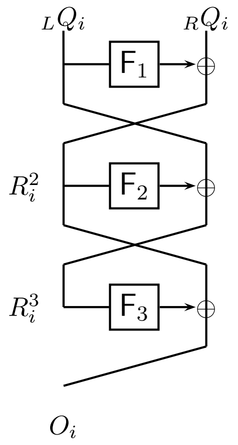
Proof (sketch). The nCPA-security of F_1, F_2, and F_3 follows from the nCPA-security of F from the previous section as stated in Lemma 2: F_2 is exactly F, so there is nothing else to prove here. The function F_3 behaves exactly as F unless it is queried on an input $[a, b, c, d]$ which satisfies $b^z = a$ for a random z. The probability that this happens on any (non-adaptive) query is just $|G|^{-1}$ (and thus exponentially small even after taking the union bound over all polynomially many queries).

To prove that F_1 is non-adaptively secure, we show how to turn any distinguisher D for F_1 into one for F_3 whose distinguishing advantage differs only by a negligible amount. First, below we completely ignore the cases where $c = d^x$ for F_1 and $a = b^z$ for F_3 as they only happen with exponentially small probability. Further, as whenever $[a, b] \neq [0, 0]$ the output of F_1 is pseudorandom, we can assume that the non-adaptive distinguisher D for F_1 only makes queries where $[a, b] = [0, 0]$.

Now consider the following distinguisher D' for F_3. First D' picks some uniformly random $s, t \in \{0,1\}^n$. D' basically simulates D, but when the query was $[0, 0, 0, 0]$ then the right half of the output is replaced with s, t. On all other queries (chosen by D) of the form $[0, 0], \gamma, \delta$, D'

[31] For instance, r_1' is the first random value generated by F_1 and r_2' is the product of r_1' and the second random value generated by F_2.

[32] Actually, there is an exponentially small probability that the values will not have that form, namely when the input to some round function "by chance" satisfies a condition that is checked. E.g. when R_1^3 is of the form $[b^z, b, c, d]$, then the "$b^z = a$" case of F_3 applies, which is only supposed to happen in the second and third query.



$$
\begin{array}{rcl@{\qquad}rcl}
{}_LQ_1 & : & [0,0,0,0] & {}_RQ_1 & : & [0,0,0,0] \\
R_1^2 & : & [g^{xr'_1}, g^{r'_1}], s, t & & & \\
R_1^3 & : & *, *, [g^{xyr'_2}, g^{r'_2}] & & & \\
O_1 & : & *, *, [g^{xyzr'_3}] \oplus s, [g^{r'_3}] \oplus t & & & \\
\\
{}_LQ_2 & : & [0,0], [g^{xyzr'_3}] \oplus s, [g^{r'_3}] \oplus t & {}_RQ_2 & : & [0,0,0,0] \\
R_2^2 & : & [g^{xr'_4}, g^{r'_4}, g^{yzr'_5}, g^{r'_5}] & & & \\
R_2^3 & : & [g^{zr'_6}, g^{r'_6}], *, * & & & \\
O_2 & : & [g^{xr'_4}, g^{r'_4}, g^{yzr'_5}, g^{r'_5}] & & & \\
\\
{}_LQ_3 & : & [0,0,g^{xr'_4}, g^{r'_4}] & {}_RQ_3 & : & [0,0,g^{yzr'_5}, g^{r'_5}] \\
R_3^2 & : & [0,0,g^{yzr'_5}, g^{r'_5}] & & & \\
R_3^3 & : & [g^{zr'_7}, g^{r'_7}], *, * & & & \\
O_3 & : & [0,0,g^{yzr'_5}, g^{r'_5}] & & &
\end{array}
$$

Figure 3.2: An adaptive three query distinguishing attack for $_L\psi_{2n}[\mathsf{F}_1\mathsf{F}_2\mathsf{F}_3]$.

invokes the system at hand by $[0,0], \gamma \oplus s, \delta \oplus t$. Finally D' outputs the decision bit of the simulated D.

If the system queried by D' is F_3 (with secret key x) then the output distribution that the simulated D gets to see is exactly as if it was generated by F_1 (with secret key x, s, t). Also note that when the system queried by D' is a URF, then also the output that D sees is uniformly random. Thus the distinguishing advantage of D' for F_1 (from a URF) is the same as the advantage of D for F_3. □

3.6 Five nCPA-Secure Feistel-Rounds

We have shown that the four-round Feistel-network with nCPA-secure round functions is CPA-secure in the information-theoretic, but in general not in the computational setting. A natural question to ask is how many rounds are necessary/not sufficient to achieve CCA-security.

In order to get a CCA-secure QRP, it is enough – by the following statement (taken from [MPR06]) – to cascade two nCPA secure QRPs (the second in inverse direction)

$$\Delta_q^{\mathsf{CCA}}(\mathbf{F} \triangleright \mathbf{G}^{-1}, \mathbf{P}) \leq \Delta_q^{\mathsf{nCPA}}(\mathbf{F}, \mathbf{P}) + \Delta_q^{\mathsf{nCPA}}(\mathbf{G}, \mathbf{P}).$$

With this and Proposition 1 we directly get that six rounds with nCPA-secure QRFs give a CCA-secure QRP, i.e.,

$$\Delta_q^{\mathsf{CCA}}(\psi_{2n}[\mathbf{FFFFFF}], \mathbf{P}) \leq 6 \cdot \Delta_q^{\mathsf{nCPA}}(\mathbf{F}, \mathbf{R}) + \frac{q^2}{2^{n-1}}.$$

So six nCPA-secure round functions are sufficient to get CCA security, and by Proposition 4 we know that at least four rounds are necessary.

Next we show that the five-round Feistel-network with nCPA-secure QRFs is a CCA-secure QRP. The following theorem is stated even stronger as the third round function must only be a KPA-secure QRF.

Theorem 3. *For any random functions* $\mathbf{F}$ *and* $\mathbf{G}$

$$\Delta_q^{\mathsf{CCA}}(\psi_{2n}[\mathbf{FFGFF}], \mathbf{P}) \leq 4 \cdot \Delta_q^{\mathsf{nCPA}}(\mathbf{F}, \mathbf{R}) + \Delta_q^{\mathsf{KPA}}(\mathbf{G}, \mathbf{R}) + \frac{q^2}{2^{n-3}}.$$

The bound is more tight than the one given in [MOPS06b]. The proof can be found in Appendix A.2.2.

It remains an open question whether four rounds are sufficient. As for the (in)security of the Feistel-network with nCPA-secure round-functions in the computational setting, we do not know anything beyond what is already implied by CPA-security alone, i.e., four rounds are not enough to get CCA-security (as it is not enough to get CPA-security by Theorem 2).

Chapter 4

Encryption based on Weak Pseudorandom Functions

The notion of a pseudorandom function (PRF) is very strong and, indeed, it is unclear whether functions such as block ciphers proposed in the literature have this very strong security property.[33] When designing cryptographic schemes, it is prudent to postulate weaker properties as this makes it more likely that a certain function has such properties and there are potentially more efficient implementations for the weaker requirement compared to the stronger. In this chapter, we investigate how to construct provably secure symmetric encryption schemes based on any weak PRF (WPRF). By now, there has been a fairly long line of research on WPRFs [NR98, ARV99, NPR99, NR99b, DN02, NR04, MOPS06a, MOPS06b, PS06, PS07]). Of course the security could be based on even weaker primitives, like any one-way function (OWF) [HILL99, GGM86]. However, such schemes are not of practical interest due to their inefficiency. The results of this chapter can also be found in [MS07].

4.1 Motivation

BACKGROUND. The main motivation of this work is Damgård and Nielsen's elegant work on WPRFs. In their paper [DN02], the Pseudoran-

[33] For example, the design criteria of AES did not include a requirement that a candidate proposal be a PRF, only that it be secure as a block cipher in certain modes of operation, against certain types of attacks.

dom Tree (PRT) construction is proposed for transforming any WPRF

$$\mathsf{F}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$$

(where the first argument is the key input) into a VOL-WPRF

$$\mathrm{PRT}^{\mathsf{F}} : \{0,1\}^{3n} \times \{0,1\}^n \times \mathbb{N} \to \{0,1\}^*.$$

It is also shown how to construct an efficient CPA-secure symmetric encryption scheme from $\mathrm{PRT}^{\mathsf{F}}$. This is achieved by simply encrypting a message $m \in \{0,1\}^*$ under a key $k \in \{0,1\}^{3n}$ and some auxiliary uniform randomness $r \in \{0,1\}^n$ as

$$(k, r, m) \mapsto \Big(r, \mathrm{PRT}^{\mathsf{F}}_k(r, |m|) \oplus m\Big). \tag{4.1}$$

To point out the efficiency of this encryption scheme (and also as a reference for the schemes presented in this work), let us compare it with standard modes of operation such as CBC and CTR. Whereas CBC and CTR invoke the underlying block cipher once per message block to encrypt/decrypt, this scheme invokes the underlying function F once per message block to encrypt/decrypt and roughly $2 \cdot \log_2(b)$ times (where b is the number of message blocks) for generating more key material from the initial key (see below). The key generation can be done offline, such that the throughput is exactly the same as for CBC and CTR. However, whereas CBC and CTR are CPA-secure if the underlying block cipher is a PRF, the Damgård-Nielsen scheme (4.1) is CPA-secure even when the underlying function is a WPRF. And as WPRFs can be much more efficiently implementable than PRFs, this scheme can also be the overall most efficient one. Unfortunately, these modes of operations are not secure against the stronger CCA. In [NR98, p. 279], Naor and Reingold posed an open problem of how to construct an efficient CCA-secure encryption scheme based on any WPRF. Damgård and Nielsen showed (using well-known techniques) how their CPA-secure scheme can be transformed to a CCA-secure one. Their open question [DN02, p. 464] whether this can be done more efficiently has been the main motivation for this work.

Before we present our results, let us briefly describe the underlying idea of the PRT-construction (illustrated in Figure 4.1(a) on page 50). In a first step, some key material $k_1, \ldots, k_d$ is generated from the initial key k by invoking F in an iterative manner, and then the output blocks are derived by applying F_{k_i}, for some $i \in \{1, \ldots, d\}$, iteratively to the input or a previously derived output block. For constructions of this type it is crucial for the security and the efficiency (in terms of the number of applications of F relative to the output length) that this is scheduled in the right way. Recently, two more efficient constructions of this type, the Expanded PRT (ERT) (see Figure 4.1(a)) and the Factorial Tree (FCT), were proposed in [MT05]. However, as we point out in Section 4.2.2, the latter and more efficient construction of the two turns out to be flawed. A natural problem that arises is to find the most efficient VOL-WPRF construction (of this type).

CONTRIBUTIONS. The contributions of this chapter are four-fold:

1. The Increasing Chain Tree (ICT); A VOL-WPRF from any WPRF.

   Our ICT-construction (see Figure 4.1(b)) is more efficient than PRT and ERT (with d generated keys our construction expands the input by a factor of $2^d - 1$, whereas PRT expands the input by roughly $1.44^d - 1$ and ERT by $1.73^d - 1$), and ICT also uses a shorter initial key (by a factor of 3). Interestingly, the generated key sequence $k_1, \ldots, k_d$ is not pseudorandom as opposed to the case for PRT and ERT. Indeed, we give strong arguments that ICT is optimal within the large and natural class of constructions described above, and hence also that it is optimal to use ICT instead of PRT in (4.1).

2. The Increasing Chain (IC) construction; A PRF from any WPRF.

   Our IC-construction is similar in nature to Goldreich, Goldwasser, and Micali's (GGM) [GGM86] construction of a PRF from any pseudorandom generator (PRG), but it is more than twice as efficient as first transforming the WPRF into a PRG and then applying the GGM-construction. It is also more efficient than Naor and Reingold's construction of a PRF based on any WPRF [NR99b][34]. This solves their open problem [NR98, p. 278] whether a more efficient construction exist positively.

   In particular, we prove that $\mathrm{IC}^{\langle\cdot\rangle}$ transform the WPRF

   $$\mathsf{exp} : \mathbb{Z}_{|G|} \times G \to G, \text{ defined by } \mathsf{exp}(k, x) = x^k, \qquad (4.2)$$

   whose security is based on a DDH group G (see [NPR99]), to Naor and Reingold's highly efficient DDH-based PRF [NR04] but with

[34] In that work the PRF is reduced to a pseudorandom synthesizer, which in turn is reduced to a WPRF.

a non-trivial[35] reduction of the key-material by a factor of roughly the input length of the PRF.

3. A CCA-secure encryption scheme from any WPRF.

   The above results, combined with a Wegman-Carter [WC81] based message authentication code (MAC) and the well-known encrypt-then-MAC method [KA98, BN00], yield an encryption scheme from any WPRF that is secure under a CCA and substantially more efficient than the one proposed by Damgård and Nielsen in [DN02] (their number of overhead applications to the WPRF is linear in the message length whereas ours is constant). We observe that for our purposes any WMAC[36] is sufficient for the MACing, i.e., encrypt-then-WMAC actually does the job. This raises the question of constructing possibly efficient WMACs from any WPRF.

4. A nCCA-secure encryption scheme from any WPRF and WMAC.

   Even though this type of security may (as CPA-security) be unsatisfactory in practice, the exact requirements for achieving standard security notions are interesting in their own right. This might also motivate further research on constructing stronger primitives from weaker ones. nCCA-secure encryption has been based on stronger primitives in [NR98].[37]

4.2 The Increasing Chain and Chain Tree Constructions

In this section, we introduce the *increasing chain* (IC) construction, for transforming a WPRF into a PRF, and the *Increasing Chain Tree* (ICT) construction, for transforming a WPRF into a VOL-WPRF. Throughout, let

$$\mathsf{F} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$$

[35]By non-trivial we mean that the key is not replaced by a pseudorandom sequence based on F, but by something more efficiently computable from F.

[36]Recall that a WMAC is an unforgeable function under a KPA (see also [NR98]).

[37]In [NR98], Naor and Reingold showed that if F is a WPRF then the encryption scheme, defined by encrypting an n-bit message m as $c = (r, \mathsf{F}_k(r) \oplus m)$ (where r is some auxiliary randomness), is CPA-secure but not nCCA-secure. Further, under the assumption that F is something stronger than a WPRF but weaker than a PRF (namely indistinguishable under adaptive samples and a random challenge) the nCCA-security of the scheme can be proven.

denote a keyed function where $\mathsf{F}_k(x) \stackrel{\text{def}}{=} \mathsf{F}(k, x)$ for all keys k and inputs x. Furthermore, let $|\mathsf{F}|$ denote the size of a circuit for computing F.

4.2.1 A Regular PRF from any Weak PRF

The IC-construction transforms F into

$$\mathrm{IC}^{\mathsf{F}} \quad : \quad (\{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n) \times \{0,1\}^N \to \{0,1\}^n,$$

for some fixed N, and is defined by the following algorithm for computing $\mathrm{IC}^{\mathsf{F}}_{k_1,r,\tau_1}(x)$:

for $i = 2$ to $|x|$ **do**
  $k_i = \mathsf{F}_{k_{i-1}}(r)$
for $i = 1$ to $|x|$ **do**
  if $x[i] = 1$ **then**
    $\tau_{i+1} = \mathsf{F}_{k_i}(\tau_i)$
  else
    $\tau_{i+1} = \tau_i$
return $\tau_{|x|+1}$

The following theorem states that IC^{F} is a PRF if F is a WPRF. This holds even if the r-value of the initial key is not kept secret. The proof of the theorem is given in Appendix A.3.1.[38]

Theorem 4. *For any t, q, and input length N of* IC^{F}

$$\mathbf{Adv}^{\mathsf{CPA}}_{t,q}(\mathrm{IC}^{\mathsf{F}}, \mathbf{R}_{N,n}) \leq N \cdot \left(\mathbf{Adv}^{\mathsf{KPA}}_{t',q}(\mathsf{F}, \mathbf{R}_{n,n}) + \frac{q(q+1)}{2^{n+1}} \right),$$

where $t' = t + \mathsf{poly}(q, |\mathsf{F}|)$ for some polynomial poly which accounts for the overhead implied by the reduction we make.

Note that F is invoked at most $2N-1$ times. However, the first $N-1$ invocations can be pre-processed and cached, and hence at most N invocations are necessary or, to be precise, as many invocations as there are ones in the input.

KEY-REDUCTION OF NAOR-REINGOLD'S DDH-BASED PRF. In [NR04], Naor and Reingold presented an efficient construction of a PRF based on

[38] We refer to [DN02] for constructing an n-bit block WPRF F from any WPRF.

any DDH group. It is easy to verify, that IC^{F} with F as defined in (4.2) is the same construction but with a significantly shorter key by a factor of roughly N (recall that N is the input length of IC^{F}). To be more precise, the first for-loop (in the IC-algorithm) generates a sequence $k_1, \ldots, k_N$ of keys from the initial key (k_1, r, τ_1) and the second for-loop exactly corresponds to the Naor-Reingold construction with $k_1, \ldots, k_N$ as its key. The reduction is non-trivial in the sense that $k_1, \ldots, k_N$ is not pseudorandom. For instance $\mathsf{F}_{k_1}^{-1}(k_2) = \mathsf{F}_{k_2}^{-1}(k_3)$ holds which can easily be verified given k_1, k_2, k_3.

THE GGM-APPROACH. An alternative approach to obtain a PRF from any WPRF F, is to first transform F into a pseudorandom generator (PRG) and then apply the so-called GGM-construction (which transforms a PRG into a PRF [GGM86]). Informally, a PRG is a deterministic function mapping a short truly random string (or seed) to a longer string which is computationally indistinguishable from truly random.[39] To illustrate that IC is the more efficient construction, let us describe the GGM-construction. It transforms a length-doubling PRG G into a PRF (say with N-bits input) as

$$\mathsf{GGM}_k(x_1 \| \ldots \| x_N) \stackrel{\text{def}}{=} \mathsf{G}_{x_1} \triangleright \ldots \triangleright \mathsf{G}_{x_N}(k),$$

where the x_i's are bits, and $\mathsf{G}_0(k)$ and $\mathsf{G}_1(k)$ denote the left and right half of $\mathsf{G}(k)$, respectively. The most efficient construction of a length-doubling PRG G from F – that we know of – is defined as

$$\mathsf{G}(k_1 \| r \| x) \stackrel{\text{def}}{=} x \| \mathsf{F}_{k_1}(x) \| \mathsf{F}_{k_2}(x) \| \mathsf{F}_{k_1} \triangleright \mathsf{F}_{k_2}(x) \| \mathsf{F}_{k_3}(x) \| r,$$

where $k_2 = \mathsf{F}_{k_1}(r)$ and $k_3 = \mathsf{F}_{k_2}(r)$.[40] Clearly, one needs 6 invocations of F per call to G, and for computing G_0 and G_1 separately one needs 3 and 4 invocations to F, respectively. Hence, to get a PRF with N-bits input and n-bits output, we hence need roughly $4N$ invocations of F per call in the worst case (cf. the efficiency of IC^{F}).

Remark 2. *Let us briefly point out a method for improving the computation time of* IC^{F} *at the cost of generating and storing more keys (say* N' *keys in-*

[39] More formally, a PRG is a family of functions $\mathsf{G} : \{0,1\}^{\ell(\gamma)} \to \{0,1\}^{L(\gamma)}$ – indexed by a security parameter γ – where $\ell(\gamma) < L(\gamma)$, G can be computed in polynomial (in γ) time by a UTM, and for any polynomial $p(.)$ it holds that $\mathbf{Adv}_{p(\gamma)}(\mathsf{G}(\mathcal{U}_{\ell(\gamma)}), \mathcal{U}_{L(\gamma)})$ is negligible in γ (here $\mathcal{U}_i$ denotes a uniform random string of length i).

[40] The proof follows from a simple hybrids argument. Alternatively, one notices that $\mathsf{G}(k_1 \| r \| x) = x \| \mathrm{ICT}^{\mathsf{F}}_{k_1,r}(x, 4n) \| r$, where ICT is defined as in the next section. Then it follows directly from the definition and security proof of ICT that G is a PRG based on F. In particular, $\mathbf{Adv}_t(\mathsf{G}(\mathcal{U}_{3n}, \mathcal{U}_{6n}) = \mathbf{Adv}^{\mathsf{KPA}}_{t,1,4n}(\mathrm{ICT}^{\mathsf{F}}, \mathbf{R}_{n,*})$ for all t, where $\mathcal{U}_i$ denotes a uniform random string of length i.

stead of N). On input x (of length N), x is first injectively mapped to a N'-bit string x' of Hamming weight at most some (fixed) c, satisfying

$$\sum_{i=0}^{c} \binom{N'}{i} \geq 2^N.$$

Then IC^{F} *is invoked on x' and the result is output. Here,* F *is invoked at most c (as opposed to N) times, as there are at most c ones in the input.*

4.2.2 Optimal Range Extension for Weak PRFs

The ICT-construction is illustrated in Figure 4.1(b) and is defined as

$$\begin{aligned}\mathrm{ICT}^{\mathsf{F}} \;\; : \;\; & \{0,1\}^{2n} \times \{0,1\}^n \times \mathbb{N} \to \{0,1\}^* \\ & ((k,r),x,l) \mapsto \left(\mathrm{IC}^{\mathsf{F}}_{k,r,x}(\langle 1\rangle)\| \cdots \| \, \mathrm{IC}^{\mathsf{F}}_{k,r,x}(\langle \lceil l/n \rceil \rangle)\right)[1,l],\end{aligned}$$

where $\langle i \rangle$ denotes the reversed standard bit encoding of the integer i (e.g. $\langle 0\rangle = 0, \langle 1\rangle = 1, \langle 2\rangle = 01, \langle 3\rangle = 11, \langle 4\rangle = 001$). Let us stress that $\mathrm{IC}^{\mathsf{F}}_{k,r,x}(\langle 0\rangle)$ can not be part of the output, as it equals the input x. It is easy to verify, see Figure 4.1(b), that $\mathrm{ICT}^{\mathsf{F}}_{k,r}(x,l)$ needs $d-1 = \lfloor \log_2(\lceil l/n \rceil)\rfloor$ calls to F for computing (or pre-computing) the needed keys $k_1,\ldots,k_d$ and further $\lceil l/n\rceil$ calls for computing the output (i.e., one call per output block). The next theorem states that $\mathrm{ICT}^{\mathsf{F}}$ is a VOL-WPRF if F is a WPRF. As for IC, the r-value of the key need not be kept secret. The proof is provided in Appendix A.3.1.

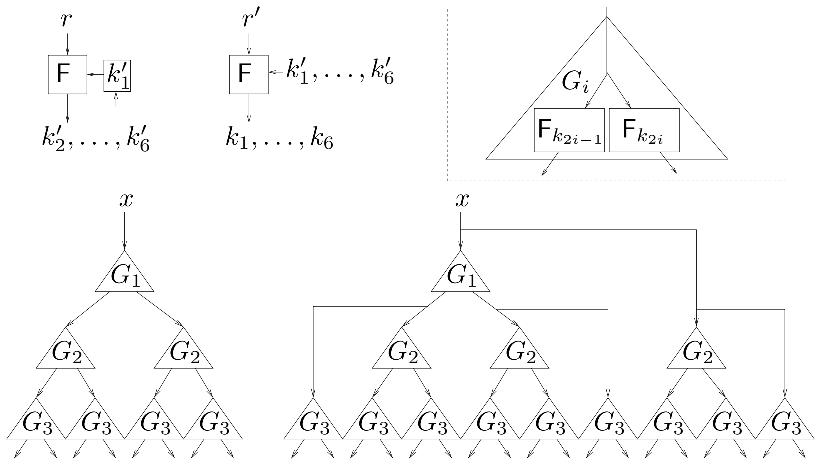
Theorem 5. *For any t, q, and μ*

$$\mathbf{Adv}^{\mathsf{VOL\text{-}KPA}}_{t,q,\mu}(\mathrm{ICT}^{\mathsf{F}}, \mathbf{R}_{n,*}) \leq d_{\max} \cdot \mathbf{Adv}^{\mathsf{KPA}}_{t',q\cdot(2^{d_{\max}-1}+1)}(\mathsf{F}, \mathbf{R}_{n,n}) + \frac{4^{d_{\max}} \cdot q^2}{2^n},$$

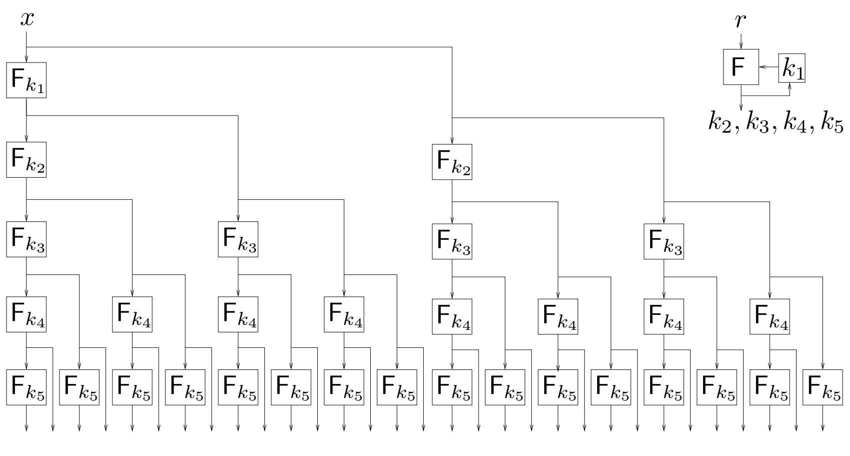
where $t' = t + \mathsf{poly}(q,\mu,|\mathsf{F}|)$ for some polynomial poly *which accounts for the overhead implied by the reduction we make, $d_{\max} = \lfloor \log_2(\lceil l_{\max}/n \rceil)\rfloor + 1$, and $l_{\max} \leq \mu$ is the maximum allowed output length of* $\mathrm{ICT}^{\mathsf{F}}$.

THE FCT-CONSTRUCTION IS FLAWED. Let us point out that the security proof of the FCT-construction (in [MT05]) is flawed. The maximal sized output of $\mathrm{FCT}^{\mathsf{F}}$ for two generated keys k_1 and k_2 is defined as

$$x \mapsto \mathsf{F}_{k_1}(x)\|\mathsf{F}_{k_2}(x)\|\mathsf{F}_{k_1} \triangleright \mathsf{F}_{k_2}(x)\|\mathsf{F}_{k_2} \triangleright \mathsf{F}_{k_1}(x). \tag{4.3}$$



(a) Computation of $\mathrm{PRT}^{\mathsf{F}}_{k'_1,r,r'}(x, 14n)$ (bottom left) and $\mathrm{ERT}^{\mathsf{F}}_{k'_1,r,r'}(x, 26n)$ (bottom right), i.e., the maximal sized output using 6 generated keys $k_1, \ldots, k_6$ (upper left). Here every output of G_i (defined upper right) for $i = 1, 2, 3$ is part of the global output.



(b) Computation of $\mathrm{ICT}^{\mathsf{F}}_{k_1,r}(x, 31n)$, i.e., the output of maximal size using 5 generated keys $k_1, \ldots, k_5$ (upper right). Here every output of F – except for the generated keys (upper right) – is part of the global output. We stress that the order of the output blocks are not the same as presented in the text.

Figure 4.1: Illustration of (a) PRT, ERT, and (b) ICT.

Clearly, the construction is insecure for WPRFs F that commute (i.e., for which $\mathsf{F}_k \triangleright \mathsf{F}_{k'}(x) = \mathsf{F}_{k'} \triangleright \mathsf{F}_k(x)$ for all k, k', x). Since such WPRFs exist based on any DDH group (see (4.2)), a fix of the security proof would contradict the existence of DDH groups and hence be a major breakthrough in number theory.[41]

ICT VS. OTHER CONSTRUCTIONS. The idea behind PRT of [DN02], ERT of [MT05], and ICT is to first generate keys $k_1, \ldots, k_d$ from the initial key (and F) and then to derive the output blocks sequentially by invoking F_{k_i} (with $i \in \{1, \ldots, d\}$) to the input or a previously computed output block (see Figure 4.1). ICT is superior to PRT and ERT for three reasons. First, the initial key of ICT is n bits (plus n bits that may be publicly known) versus $3n$ bits for PRT and ERT. Second, whereas ICT needs $d-1$ invocations of F to generate the d keys $k_1, \ldots, k_d$, PRT and ERT needs $2d-1$ invocations. Third, the maximal output size using the generated keys $k_1, \ldots, k_d$ is $(2^d-1)n$ for ICT, roughly $(3^{\frac{d}{2}}-1)n$ for ERT, and roughly $(2^{\frac{d}{2}+1}-2)n$ for PRT.[42] For all constructions, the keys needed for computing outputs of length bounded by some fixed value (say $l_{\max}$) can be pre-processed, such that one call of F is needed per output block. But whereas ICT needs to store say $s \stackrel{\text{def}}{=} \lfloor \log_2(\lceil l_{\max}/n \rceil) \rfloor + 1$ keys, ERT and PRT store about $\lceil 1.26s \rceil$ and $2s$ keys, respectively. The factor in front of the WPRF-advantage in the security reduction reduces correspondingly, i.e., for s as defined above we roughly have (for some $t' = t + \mathsf{poly}(q, \mu, |\mathsf{F}|)$):

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{VOL\text{-}KPA}}_{t,q}(\mathrm{ICT}^{\mathsf{F}}, \mathbf{R}_{n,*}) &\le \quad s \cdot \mathbf{Adv}^{\mathsf{KPA}}_{t',2^{s-1}q}(\mathsf{F}, \mathbf{R}) + 4^s q^2/2^n \\
\mathbf{Adv}^{\mathsf{VOL\text{-}KPA}}_{t,q}(\mathrm{ERT}^{\mathsf{F}}, \mathbf{R}_{n,*}) &\le 1.26s \cdot \mathbf{Adv}^{\mathsf{KPA}}_{t',2^{s-1}q/3}(\mathsf{F}, \mathbf{R}) + 4^s q^2/(2^n \cdot 9) \\
\mathbf{Adv}^{\mathsf{VOL\text{-}KPA}}_{t,q}(\mathrm{PRT}^{\mathsf{F}}, \mathbf{R}_{n,*}) &\le \quad 2s \cdot \mathbf{Adv}^{\mathsf{KPA}}_{t',2^{s-1}q/4}(\mathsf{F}, \mathbf{R}) + 4^s q^2/(2^n \cdot 16).
\end{aligned}$$

OPTIMALITY OF ICT. In [PS07], it is shown that there is no black-box proof of the security for constructions that expands more than ICT (for any fixed number of generated keys). Here, we show something stronger for the constructions with log-time random access to output blocks, i.e., for the rather balanced constructions where the maximal length of the composition chains are in $O(\log(l))$ for output length l, namely that ICT is

[41] However, information theoreticly (and even in Minicrypt, i.e., under the assumption that one-way functions exist but public-key cryptography does not) it turns out that (4.3) is secure [PS06, PS07].

[42] The latter two values are exact if d is even. Otherwise $(2 \cdot 3^{\frac{d-1}{2}} - 1)n$ and $(3 \cdot 2^{\frac{d-1}{2}} - 2)n$ are exact, respectively.

optimal within that class of constructions under the *inverse* DDH (IDDH) assumption [BDZ03].

To be more precise, note that – for $l = 3n$ – the value $\mathrm{ICT}^{\mathsf{F}}_{k_1,r}(x,l)$ is derived by first computing $k_2 = \mathsf{F}_{k_1}(r)$ and then returning

$$y := \mathsf{F}_{k_1}(x) \| \mathsf{F}_{k_2}(x) \| \mathsf{F}_{k_1} \triangleright \mathsf{F}_{k_2}(x).$$

For $l = 7n$, an extra key $k_3 = \mathsf{F}_{k_2}(r)$ is derived and

$$y \| \mathsf{F}_{k_3}(x) \| \mathsf{F}_{k_1} \triangleright \mathsf{F}_{k_3}(x) \| \mathsf{F}_{k_2} \triangleright \mathsf{F}_{k_3}(x) \| \mathsf{F}_{k_1} \triangleright \mathsf{F}_{k_2} \triangleright \mathsf{F}_{k_3}(x)$$

is returned. A natural question is whether more can be output before a new key needs to be generated, i.e., for a fixed number of generated keys (say k_1, k_2, and k_3), can we output more than $\mathrm{ICT}^{\mathsf{F}}$ maximally can (i.e., more than $7n$ bits) by invoking the instantiations (i.e., $\mathsf{F}_{k_1}, \mathsf{F}_{k_2}, \mathsf{F}_{k_3}$) *one* more time than $\mathrm{ICT}^{\mathsf{F}}$ does (i.e., 8 times instead of 7). The answer turns out to be "no" unless the IDDH assumption is false, since otherwise there is a WPRF F, described in (4.4), which with high probability both commutes and is self inverse, i.e., for all $k \neq k'$

$$\Pr_x[\mathsf{F}_k \triangleright \mathsf{F}_{k'}(x) = \mathsf{F}_{k'} \triangleright \mathsf{F}_k(x)] \approx 1/4 \quad \text{and} \quad \Pr_x[\mathsf{F}_k \triangleright \mathsf{F}_k(x) = x] \approx 1/2.$$

If F is used and more is output at least two output blocks will (by the pigeonhole principle) have the same value with high probability (which is unlikely for a uniform random VOL-function). F is defined for a group G of prime order ρ as

$$\mathsf{F} : \mathbb{Z}_\rho \times G \to G \ \text{ and } \ \mathsf{F}_k(x) \stackrel{\text{def}}{=} \begin{cases} x^k & \text{if } x \in P_1 \\ x^{k^{-1}} & \text{if } x \in P_2 \end{cases}, \tag{4.4}$$

where $k \cdot k^{-1} = 1 \pmod{\rho}$ and $\{P_1,\ P_2\}$ is a partition of G in roughly equal sized sets (where we assume that it is efficient to decide whether $x \in P_1$ or not). A proof that F is a WPRF if G is an IDDH group is given in [Kel06].

4.3 Encryption Schemes from Weak PRFs (and Weak MACs)

In this section, we optimize Damgård and Nielsen's CPA-secure encryption scheme (4.1) by using ICT instead of PRT. Then, we first make the scheme CCA-secure by applying IC and the well-known encrypt-then-MAC technique (actually we prove that what we call encrypt-then-WMAC does the job here), and, second, we make it non-adaptive CCA-secure by using a (fixed-input-length) WMAC for authenticating the auxiliary uniform randomness.

4.3.1 CPA-Secure Encryption

In [DN02], Damgård and Nielsen introduced an IND-P2-C0-secure encryption scheme based on any VOL-WPRF $\mathsf{V} : \{0,1\}^\kappa \times \{0,1\}^n \times \mathbb{N} \to \{0,1\}^*$. To be precise, their encryption scheme $\mathcal{SE}_1$ is defined by encrypting a message $m \in \{0,1\}^*$, under the key $k \in \{0,1\}^\kappa$ and some auxiliary uniform randomness $r \in \{0,1\}^n$ as

$$(k, r, m) \mapsto \Big(r, \mathsf{V}_k(r, |m|) \oplus m\Big). \qquad (\mathcal{SE}_1) \quad (4.5)$$

The following proposition originates from [DN02]. For completeness, the proof is provided in Appendix A.3.2.

Proposition 6. *For any t, q, and μ*

$$\mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C0}}_{t,q,\mu}(\mathcal{SE}_1) \leq 2 \cdot \mathbf{Adv}^{\mathsf{VOL\text{-}KPA}}_{t',q,\mu}(\mathsf{V}, \mathbf{R}) + \frac{q-1}{2^{n-1}},$$

where $t' = t + \mathsf{poly}(q, \mu, n)$ for some polynomial poly *which accounts for the overhead implied by the reduction we make.*

Remark 3. *Given the strong optimality arguments for* ICT*, it is clear that (4.1) is optimal when* ICT *is used (in place of* PRT*) unless a significantly different approach for range extension of WPRFs is invented.*

4.3.2 CCA-Secure Encryption

The well-known encrypt-then-MAC method is a general technique for constructing an INT-CTXT- and IND-P2-C2-secure encryption scheme from any IND-P2-C0-secure encryption scheme $\mathcal{SE} = (Enc, Dec)$ and any VIL-MAC W. The idea is to simply encrypt with Enc and then authenticate the ciphertext using W [KA98, BN00]. Here, we note that for the IND-P2-C0-secure scheme $\mathcal{SE}_1$ which is based on any VOL-WPRF $\mathsf{V} : \{0,1\}^{\kappa_1} \times \{0,1\}^n \times \mathbb{N} \to \{0,1\}^*$, it is sufficient if $\mathsf{W} : \{0,1\}^{\kappa_2} \times \{0,1\}^* \to \{0,1\}^\ell$ is a VIL-WMAC (as the ciphertexts of $\mathcal{SE}_1$ are pseudorandom).

To be precise, the scheme $\mathcal{SE}_2$, defined by encrypting $m \in \{0,1\}^*$ under a key $(k_1, k_2) \in \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_2}$ and auxiliary uniform randomness $r \in \{0,1\}^n$ as

$$\big((k_1,k_2), r, m\big) \mapsto \Big(r, \underbrace{\mathsf{V}_{k_1}(r,|m|) \oplus m}_{c}, \mathsf{W}_{k_2}(r\|c)\Big), \qquad (\mathcal{SE}_2) \quad (4.6)$$

is IND-P2-C2 secure if V is a VIL-WPRF and W is a VIL-WMAC. The proof (of the following theorem) is given in Appendix A.3.2.

Theorem 6. *For any t, q, μ, q', μ' (and efficient* V *and* W*)*

$$\begin{aligned}\mathbf{InSec}^{\mathsf{INT\text{-}CTXT}}_{t,q,\mu,q',\mu'}(\mathcal{SE}_2) \quad &\leq \quad \min\Big\{q' \cdot \mathbf{InSec}^{\mathsf{UF\text{-}CPA}}_{t',q,\mu+qn+\mu'}(\mathsf{W}),\\ &\qquad \mathbf{Adv}^{\mathsf{VOL\text{-}KPA}}_{t',q,\mu}(\mathsf{V},\mathbf{R}) + \frac{q^2}{2^{n+1}} + q' \cdot \mathbf{InSec}^{\mathsf{UF\text{-}KPA}}_{t',q,\mu+qn+\mu'}(\mathsf{W})\Big\}\\ \mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C2}}_{t,q,\mu,q',\mu'}(\mathcal{SE}_2) \quad &\leq \quad 2 \cdot \mathbf{InSec}^{\mathsf{INT\text{-}CTXT}}_{t',q,\mu,q',\mu'}(\mathcal{SE}_2) + \mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C0}}_{t',q,\mu}(\mathcal{SE}_1),\end{aligned}$$

where $t' = t + \mathsf{poly}(q, \mu, q', \mu', n)$ for some polynomial poly *which accounts for the overhead implied by the reduction we make.*

Remark 4. *The above result leads to an interesting open question for further research, namely, how efficient constructions there are of a VIL-WMAC* W *based on any WPRF* F*. One approach – for constructing* W *– would be to first transform* F *into the PRF $\mathrm{IC}^{\mathsf{F}} : \{0,1\}^{3n} \times \{0,1\}^N \to \{0,1\}^n$ (see Section 4.2.1) and then apply the following rather standard method [WC81, Sho96, BHK+99] for constructing a VIL-MAC (and thus also a VIL-WMAC) from any PRF. Simply hash the message using an ε-almost universal (AU) hash function $\mathsf{H} : \mathcal{K} \times \{0,1\}^* \to \{0,1\}^N$ (i.e., for all distinct $m, m' \in \{0,1\}^*$ we have that $\Pr\Big[k' \xleftarrow{\$} \mathcal{K} : \mathsf{H}_{k'}(m) = \mathsf{H}_{k'}(m')\Big] \leq \varepsilon$ [Sti92]) and then apply IC^{F} to the result: $\mathsf{W}_{k,k'}(x) \stackrel{\text{def}}{=} \mathsf{H}_{k'} \triangleright \mathrm{IC}^{\mathsf{F}}_k(x)$.[43] This method is appealing since* H *exists unconditionally and IC^{F} is invoked on "short" inputs (of size N). There are 2^{1-N}-AU hash functions, with $5N$-bit key size and maximal input length 2^N, that should do for most practical applications (see [WC81]).*

Remark 5. *By combining (4.6) with $\mathsf{V} = \mathrm{ICT}^{\mathsf{F}}$ and* W *(as defined above), we get a* CCA*-secure encryption scheme from any WPRF* F*. In [DN02], Damgård and Nielsen also proposed to use the encrypt-then-MAC method for achieving* CCA*-security of $\mathcal{SE}_1$. However, their approach for constructing the VIL-MAC from any WPRF introduces a too large overhead for the solution to be*

[43] For any $\mathsf{Q} : \mathcal{K}' \times \{0,1\}^N \to \{0,1\}^n$ and ϵ-AU hash function $\mathsf{H} : \mathcal{K} \times \{0,1\}^* \to \{0,1\}^N$, $\mathbf{InSec}^{\mathsf{UF\text{-}CPA}}_{t,q,\mu}(\mathsf{H} \triangleright \mathsf{Q}) \leq \mathbf{Adv}^{\mathsf{CPA}}_{t,q}(\mathsf{Q}, \mathbf{R}) + q(q-1)\varepsilon/2 + 1/2^n$ (see [BHK+99]).

practical. The number of applications of the WPRF per evaluation is in the order of the message length. The approach we give in Remark 4 is more efficient using at most N applications of the WPRF independently of the message length, where typically $N \ll n$ (recall that n is the block length of F). Whereas this additive overhead is of little concern for "long" messages, it is an open problem whether it can be improved for "short" messages.

4.3.3 nCCA-Secure Encryption

To achieve IND-P2-C1-security of $\mathcal{SE}_1$, we note that it is sufficient to WMAC the auxiliary randomness r. This has the advantage (over $\mathcal{SE}_2$) that the WMAC does not need to have VIL. To be precise, for $\mathsf{V} : \{0,1\}^{\kappa_1} \times \{0,1\}^n \times \mathbb{N} \to \{0,1\}^*$ and $\mathsf{W} : \{0,1\}^{\kappa_2} \times \{0,1\}^n \to \{0,1\}^\ell$, let $\mathcal{SE}_3$ denote the encryption scheme defined by encrypting a message $m \in \{0,1\}^*$ under the key $(k_1, k_2) \in \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_2}$ and some auxiliary uniform random string $r \in \{0,1\}^n$ as

$$((k_1, k_2), r, m) \mapsto \Big(r, \mathsf{V}_{k_1}(r, |m|) \oplus m, \mathsf{W}_{k_2}(r)\Big). \quad (\mathcal{SE}_3) \tag{4.7}$$

The proof of the following theorem is provided in Appendix A.3.2.

Theorem 7. *For any t, q, μ, q', μ' (and efficient V and W)*

$$\mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C1}}_{t,q,\mu,q',\mu'}(\mathcal{SE}_3) \leq 2 \cdot q' \cdot \mathbf{InSec}^{\mathsf{UF\text{-}KPA}}_{t',q}(\mathsf{W}) + \mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C0}}_{t',q,\mu+q\mu'}(\mathcal{SE}_1),$$

where $t' = t + \mathsf{poly}(q, \mu, q', \mu', n)$ for some polynomial poly which accounts for the overhead implied by the reduction we make.

Remark 6. *Combining (4.7) with $\mathsf{V} = \mathrm{ICT}^{\mathsf{F}}$ and $\mathsf{W} = \mathsf{H} \triangleright \mathrm{IC}^{\mathsf{F}}$ results in an IND-P2-C1-secure scheme based on any WPRF F, but with the advantage that the ε-AU hash function H only is applied on fixed-sized strings (of length n). Alternatively, using $\mathsf{W} = \mathrm{IC}^{\mathsf{F}}$ saves the call to H and results in $n/2$ overhead applications on average (as IC^{F} is then invoked on random inputs).*

4.4 Open Problems

Although several highly efficient candidates for weak PRFs exist, none were targeted at this particular security notion explicitly. It is an interesting question for further research how much block-cipher design can benefit from this weakening of the desired security goal. An other open question is whether more efficient constructions of weak MACs based on weak PRFs exist than the ones presented in this chapter.

Chapter 5

Domain Extension of Message Authentication Codes

In this chapter, we consider a construction paradigm for domain extension of MACs, i.e., for constructing VIL- or AIL-MACs from FIL-MACs. We propose a new construction, which is superior (in several aspects) to all previous known constructions given in the literature. The results of this chapter appeared in [MS05b, MS05a].

5.1 Motivation

BACKGROUND. In 1997, Naor and Reingold [NR98] constructed a FIL-PRF from any FIL-MAC. While this FIL-PRF could in principle be used in some well-known construction of an AIL-PRF (i.e., also a VIL-MAC) from any FIL-PRF (e.g. the CBC-MAC [BKR00]), it would be impractical due to efficiency reasons.[44] In 1999, the problem of constructing VIL-MACs from FIL-MACs, was proposed An and Bellare [AB99]. They showed that the CBC-MAC is insecure under this weaker assumption for the FIL-primitive and presented the first practical construction of a VIL-MAC

[44] The open question (given in [NR98]) whether a VIL-PRF can be obtained from any FIL-MAC at low cost is open to date.

based on any FIL-MAC, the so-called *nested iterated* (NI) construction illustrated in Figure 5.7. For completeness, we give a security proof of NI in Section 5.5.

CONTRIBUTIONS. In Section 5.2, we propose a natural and general paradigm for constructing AIL-MACs from FIL-MACs. In Section 5.3, we introduce an essentially optimal AIL-MAC construction, the PDI-construction, for practical use and prove its security. It uses a single key, is optimal in terms of number of invocations to the FIL-MAC, allow for on-line and parallel processing of the messages, and has an essentially tight security reduction. The only previously known construction, the NI-construction, uses two keys, has an upper bound of 2^b on the message length, and is not optimal in terms of the number of applications to the FIL-MAC (see Section 5.5). In Section 5.4, the paradigm is generalized to comprise a greater class of constructions and, in particular, we elaborate on an efficiency/security tradeoff for AIL-MAC constructions.

5.2 The Construction Paradigm

5.2.1 Constructions and Important Design Criteria

Throughout this chapter, let

$$\mathsf{G} \stackrel{\text{def}}{=} \{g_k : \{0,1\}^L \to \{0,1\}^\ell\}_{k\in\{0,1\}^\kappa}$$

denote a function family, with *compression* $b \stackrel{\text{def}}{=} L - \ell > 0$. We consider a general type of construction $\mathrm{C}^{\langle\cdot\rangle}$, which uses G to construct

$$\mathrm{C}^{\mathsf{G}} \stackrel{\text{def}}{=} \{\mathrm{C}^{g_k} : \mathcal{M} \to \{0,1\}^\ell\}_{k\in\{0,1\}^\kappa},$$

where $\mathcal{M}$ is either AIL (i.e., $\{0,1\}^*$) or VIL (i.e., $\{0,1\}^{\leq N}$). The instantiation C^{g_k} is constructed by invoking g_k several times in a black-box manner. To be more precise, let us describe the computation of the tag $\tau = \mathrm{C}^{g_k}(m)$ for an n-bit message m (see Figure 5.1). In a pre-processing step m is encoded into a bit string m' of length (denoted by) $\lambda(n)$, for instance by padding m and appending information about its length. The processing step is best described with a buffer initialized with m', where each call to g_k fetches (and deletes) some L bits and writes back the ℓ-bit result to the buffer. This reduces the number of bits in the buffer (by b bits) with each call to g_k. As soon as the number of bits is less than L, the content

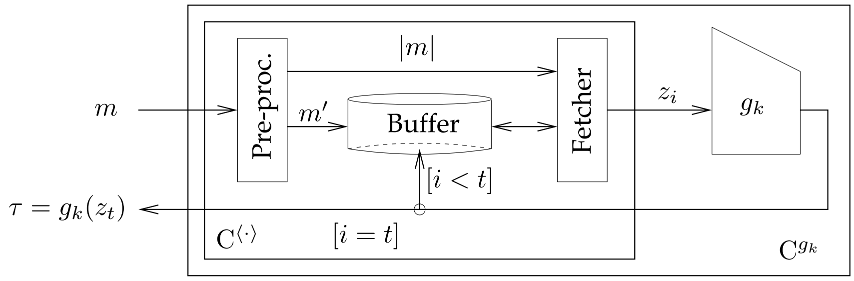


Figure 5.1: The construction paradigm

of the buffer is returned as the tag τ. To obtain an ℓ-bit output, an appropriate encoding is used such that $\lambda(n) = \phi(n) \cdot b + \ell$ for some $\phi(n)$. Note that $\phi(n)$ is exactly the number of calls to g_k required to compute τ, and that τ is the last output of g_k. The function $\phi(\cdot)$ is referred to as the *application* function of $C^{\langle\cdot\rangle}$. A particular construction can thus be described by the encoding function mapping m to m' and by the scheme by which the L-bit blocks are fetched.

In a more general variant of such a construction, several (say 2) instantiations g_{k_1} and g_{k_2} of G can be used to build an instantiation $C^{g_{k_1}, g_{k_2}}$ of $C^{\mathsf{G},\mathsf{G}} \stackrel{\text{def}}{=} \{C^{g_{k_1}, g_{k_2}} : \mathcal{M} \to \{0,1\}^\ell\}_{k_1, k_2 \in \{0,1\}^\kappa}$ (with key space $(\{0,1\}^\kappa)^2$). The only difference in the computation of the tag, described above, is that for each L-bit block that is fetched, the instantiation to be invoked needs to be specified. For such schemes $\phi^i(n)$ (with $i \in \{1,2\}$) denotes the number of calls needed to g_{k_i} in order to compute the tag of an n-bit message, and $\phi(n) \stackrel{\text{def}}{=} \phi^1(n) + \phi^2(n)$.

Note that the key space of $C^{\mathsf{G},\mathsf{G}}$ is twice the size of the key space of C^{G}. We refer to $C^{\langle\cdot\rangle}$ as a single-key construction and to $C^{\langle\cdot,\cdot\rangle}$ as a 2-key construction. We now discuss the main design criteria for the constructions:

Number of Keys: We will propose single-key constructions (like $C^{\langle\cdot\rangle}$) for practical use and see that there is essentially no reason for considering multiple-key constructions (like $C^{\langle\cdot,\cdot\rangle}$).

Efficiency: The efficiency can be measured in the number of *applications* $\phi(n)$ of G, or equivalently in terms of the *waste*

$$w(n) \stackrel{\text{def}}{=} \lambda(n) - n = \phi(n) \cdot b + \ell - n,$$

i.e., the amount by which pre-processing expands the message.

Type of Processing: It is desirable that a message can be processed *on-line*, i.e., as the message bits arrive, without knowing the message length in advance. Moreover, it is desirable that the computation of the tag τ can be *parallelized*, i.e., sped up by a factor of roughly c (over the construction using one processor) when c processors are available.

Message Space: As we will see, it turns out that no bound on the message length is necessary, and therefore our focus is on AIL constructions.

5.2.2 Security Reduction (Single Key)

To prove the security of a MAC based on a FIL-MAC one shows that the existence of a (t, q, μ, ε)-forger A for the MAC implies the existence of a (t', q', ε')-forger A' for the FIL-MAC, where t', q', and ε' are polynomials in t, q, μ, and ε.

In all our security proofs A is called only once by A'. Therefore, the size of A' is essentially that of A, i.e., $t' \approx t$, with some small overhead that is obvious from the construction of A'. We will therefore not bother to explicitly compute the size t' as this complicates the analysis unnecessarily without providing more insight.

A (t, q, μ, ε)-forger A for a MAC C^{G} is allowed at most q oracle queries to its oracle C^{g_k} (where k is chosen uniformly at random) of total length at most μ (including the length of the forgery message) and then returns a valid forgery (m, τ) with probability at least ε. We refer to $\mathsf{A}\blacklozenge\mathrm{C}^{g_k}$ as the process in which A's queries to C^{g_k} are computed and returned to A, and where A's forgery (m, τ) is verified by computing $\mathrm{C}^{g_k}(m)$. Let us consider the random variables occurring at the interface to g_k (in the process $\mathsf{A}\blacklozenge\mathrm{C}^{g_k}$). Let z_i denote the i-th input to g_k and let $y_i \stackrel{\text{def}}{=} g_k(z_i)$. The sequences $\mathbf{Z} \stackrel{\text{def}}{=} (z_1, z_2, \ldots)$ and $\mathbf{Y} \stackrel{\text{def}}{=} (y_1, y_2, \ldots)$ are thus naturally defined. Note that as soon as the key k and the random coins of A are fixed, all values in $\mathbf{Z}$ and $\mathbf{Y}$ are determined, and also whether A is successful or not. Let $\mathcal{E}$ denote the event that A is successful. Without loss of generality we assume that A's forgery message m is distinct from A's oracle queries. Thus $\mathcal{E}$ occurs if and only if $\mathrm{C}^{g_k}(m) = \tau$.

We consider a FIL-MAC forger A' for G that simulates $\mathsf{A}\blacklozenge\mathrm{C}^{g_k}$ with the help of A and its oracle access to g_k. At some query z_i to g_k it stops the simulation and returns a forgery (z', τ') for g_k (without making any other oracle queries to g_k). Such a forger is characterized by the moment it stops (i.e., i) and the way it produces its forgery. We refer to this as the *strategy* s of A' and let A'_s denote the corresponding forger.

The most simple strategy is the *naïve* strategy s_{na}. $\mathsf{A}'_{s_{\mathrm{na}}}$ stops the simulation of $\mathsf{A}\blacklozenge\mathrm{C}^{g_k}$ at the very last query $\mathbf{z}$ to g_k (i.e., $\mathbf{z}$ is the last entry in $\mathbf{Z}$). Then it returns $(\mathbf{z}, \tau)$ as a forgery, where τ is the forgery tag of A's forgery (m, τ) for C^{g_k}. $\mathsf{A}'_{s_{\mathrm{na}}}$ is successful if the following two conditions hold. First, $\mathcal{E}$ occurs, i.e., $\mathrm{C}^{g_k}(m) = \tau$ (and thus $g_k(\mathbf{z}) = \tau$ by the definition of $\mathrm{C}^{\langle\cdot\rangle}$), and second $\mathbf{z}$ is new, i.e., $\mathbf{z}$ only occurs at the last entry in $\mathbf{Z}$. Let $\mathcal{E}_{\text{new}}$ denote the event that $\mathbf{z}$ is new. Thus $\mathsf{A}'_{s_{\mathrm{na}}}$ is successful whenever $\mathcal{E} \wedge \mathcal{E}_{\text{new}}$ occurs.

Imagine that there is a set $\mathcal{S}$ of strategies such that whenever $\bar{\mathcal{E}}_{\text{new}}$ occurs there exists at least one strategy $s \in \mathcal{S}$ for which A'_s is successful. We refer to such a set $\mathcal{S}$ as *complete* for the construction. Obviously, the set $\mathcal{S} \cup \{s_{\mathrm{na}}\}$ has the property that whenever $\mathcal{E}$ occurs, there is at least one strategy $s \in \mathcal{S} \cup \{s_{\mathrm{na}}\}$ for which A'_s is successful. Thus an overall strategy of A' is to pick its strategy uniformly at random from $\mathcal{S} \cup \{s_{\mathrm{na}}\}$. Its success probability is at least the probability that $\mathcal{E}$ occurs divided by $\#\mathcal{S} + 1$, since the choice of strategy is independent of $\mathcal{E}$. As A''s number of oracle queries is at most $|\mathbf{Z}|$, which is a random variable, it is convenient to introduce the following function.

Definition 22. *The* expansion *function Φ of a construction $\mathrm{C}^{\langle\cdot\rangle}$ is defined as*

$$\Phi(\tilde{q}, \tilde{\mu}) \stackrel{def}{=} \max \left\{ \sum_{i=1}^{\tilde{q}} \phi(n_i) : n_1, \ldots, n_{\tilde{q}} \in \mathbb{N}_0,\ n_1 + \cdots + n_{\tilde{q}} \le \tilde{\mu} \right\},$$

where $\phi(\cdot)$ is the application function of $\mathrm{C}^{\langle\cdot\rangle}$.

It follows that $|\mathbf{Z}| \le \Phi(q+1, \mu)$, since there are at most $q+1$ queries of total length at most μ to C^{g_k} in $\mathsf{A}\blacklozenge\mathrm{C}^{g_k}$.

Proposition 7. *The existence of a complete set $\mathcal{S}$ for a construction $\mathrm{C}^{\langle\cdot\rangle}$ and a (t, q, μ, ε)-forger A for C^{G} implies the existence of a (t', q', ε')-forger A' for G, where $q' = \Phi(q+1, \mu)$, $\varepsilon' = \frac{\varepsilon}{\#\mathcal{S}+1}$, and $t' = t + t''$ where t'' denotes the size of any circuit which picks a strategy at random from $\mathcal{S}$ and runs it using one black-box invocation to A and at most q' queries to its oracle G.*

Proof. A' picks its strategy s uniformly at random from $\mathcal{S} \cup \{s_{\mathrm{na}}\}$. Let $\mathcal{E}'$ denote the event that A' is successful, and let $\mathcal{E}$ and $\mathcal{E}_{\text{new}}$ be defined as above.

$$\begin{aligned}\underbrace{\Pr[\mathcal{E}']}_{=:\varepsilon'} &\geq \underbrace{\Pr[\mathcal{E}' \mid \mathcal{E} \wedge \mathcal{E}_{\text{new}}]}_{\geq 1/(\#\mathcal{S}+1)} \cdot \Pr[\mathcal{E} \wedge \mathcal{E}_{\text{new}}] + \underbrace{\Pr\left[\mathcal{E}' \mid \bar{\mathcal{E}}_{\text{new}}\right]}_{\geq 1/(\#\mathcal{S}+1)} \cdot \underbrace{\Pr\left[\bar{\mathcal{E}}_{\text{new}}\right]}_{\geq \Pr\left[\mathcal{E} \wedge \bar{\mathcal{E}}_{\text{new}}\right]} \\ &\geq \underbrace{\frac{\Pr[\mathcal{E}]}{\#\mathcal{S}+1}}_{=\varepsilon/(\#\mathcal{S}+1)}\end{aligned}$$

□

5.2.3 Deterministic Strategies

An important class of strategies for A' are the deterministic strategies. A deterministic strategy s is characterized by a pair (i, f), where $i \in \{1, \ldots, \Phi(q+1, \mu)\}$ is an index and f a function mapping $(\mathbf{Z}_i, \mathbf{Y}_{i-1})$ to some value $\hat{y}_i \in \{0,1\}^\ell$ (which can be seen as a prediction of y_i). More precisely, the corresponding forger A'_s stops (the simulation of $\mathsf{A}\blacklozenge\mathsf{C}^{g_k}$) at query z_i and returns $(z_i, \hat{y}_i)$ as a forgery.[45] The forger is successful if $\hat{y}_i = y_i$ and if z_i is new, i.e., does not occur in $\mathbf{Z}_{i-1}$. Next we present three particular sets of strategies, which will be used in the sequel:

- Let $s_{i,y}$ (with $y \in \{0,1\}^\ell$) denote the strategy of stopping at query z_i and returning (z_i, y) as a forgery. Note that whenever the event occurs that g_k outputs y, i.e., when y is an entry in $\mathbf{Y}$, then there is a strategy s in

  $$\mathcal{S}_y \stackrel{\text{def}}{=} \left\{ s_{i,y} | i \in \{1, \ldots, \Phi(q+1, \mu)\} \right\}$$

  for which A'_s is successful. We have

  $$\#\mathcal{S}_y = \Phi(q+1, \mu). \tag{5.1}$$

- Let $s_{\text{coll},i,j}$ (with $i > j$) denote the strategy of stopping at query z_i and returning (z_i, y_j) as a forgery. Note that whenever a non-trivial collision for g_k occurs, i.e., $\alpha, \beta \in \{1, \ldots, |\mathbf{Z}|\}$ satisfying $z_\alpha \neq z_\beta$ and $y_\alpha = y_\beta$, then there is a strategy s in

  $$\mathcal{S}_{\text{coll}} \stackrel{\text{def}}{=} \left\{ s_{\text{coll},i,j} | i, j \in \{1, \ldots, \Phi(q+1, \mu)\}, i > j) \right\}$$

  for which A'_s is successful. The cardinality of $\mathcal{S}_{\text{coll}}$ is

  $$\#\mathcal{S}_{\text{coll}} = \frac{\Phi(q+1, \mu)^2}{2} - \frac{\Phi(q+1, \mu)}{2}. \tag{5.2}$$

[45] If $i > |\mathbf{Z}|$ the forger aborts.

- Let $s_{\text{coll2},i,j,a,\text{left}}$ (with $a \in \{0,1\}$ and $i > j$) denote the strategy of stopping at input z_i and returning $(z_i, a\|y_j[1,\ell-1])$ as a forgery, and let $s_{\text{coll2},i,j,a,\text{right}}$ denote the strategy of stopping at input z_i and returning $(z_i, y_j[2,\ell]\|a)$ as a forgery. Note that whenever the event occurs that there are $\alpha, \beta \in \{1,\ldots,|\mathbf{Z}|\}$ satisfying $z_\alpha \neq z_\beta$ and $g_k(z_\alpha)[2,\ell] = g_k(z_\beta)[1,\ell-1]$, then there is a strategy s in

$$\mathcal{S}_{\text{coll2}} \stackrel{\text{def}}{=} \left\{ s_{\text{coll2},i,j,a,d} \,\middle|\, \begin{array}{l} i,j \in \{1,\ldots,\Phi(q+1,\mu)\}, \\ i>j, a \in \{1,2\}, d \in \{\text{left}, \text{right}\} \end{array} \right\}$$

for which A'_s is successful. The cardinality of $\mathcal{S}_{\text{coll2}}$ is

$$\#\mathcal{S}_{\text{coll2}} = 2 \cdot \Phi(q+1,\mu)^2 - 2 \cdot \Phi(q+1,\mu). \tag{5.3}$$

5.3 Concrete Constructions

In this section, we present new on-line AIL-MAC constructions. First, we introduce the Prefix-Free Iterated (PI) construction which has linear waste (i.e., $w(n) \in \theta(n)$) but is efficient for short messages. Then, we present the Double-Iterated (DI) construction which has constant waste (i.e., $w(n) \in \theta(1)$) and therefore is efficient for long messages. Finally, we propose the Prefix-Free Double-Iterated (PDI) construction, which is a hybrid constructions between the DI- and the PI-construction. The construction depends on a design parameter $r \in \mathbb{N}_0$. For $r = 0$ the construction is equivalent to the DI-construction and for $r \to \infty$ to the PI-construction. For values of r between this range the advantages of both the DI- and the PI-construction are exploited. The idea is to simply apply the PI-construction for short messages and the DI-construction for long messages. What short and long means depends on the design parameter r.

5.3.1 The Iteration Method

Before the AIL-MAC constructions are presented, we analyze the iteration (I) method of a function $h : \{0,1\}^{b+\ell} \to \{0,1\}^\ell$ as illustrated in Figure 5.2. $\mathrm{I}^h_{\mathrm{IV}}(\cdot)$ where IV denotes a fixed ℓ-bit initialization value is defined by the following recursion (see also Section 9.3.1 of [MvOV97]).

The value $\tau = \mathrm{I}^h_{\mathrm{IV}}(m)$ for a string $m \in (\{0,1\}^b)^*$, i.e., $m_1\|\cdots\|m_t = m$ for some $t \geq 1$ and $|m_i| = b$ for $i \in \{1,\ldots,t\}$, is computed as

$$y_0 = \mathrm{IV}; \quad y_i = h(y_{i-1}\|m_i)\ ,\ 1 \leq i \leq t; \quad \tau = y_t.$$

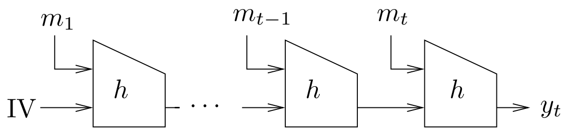


Figure 5.2: The Iteration (I) method

Lemma 5. *A non-trivial collision in* $\mathrm{I}^h_{\mathrm{IV}}(\cdot)$ *implies a non-trivial collision in* h *or that an output of* h *equals* IV*.*

Proof. Let $m \neq m'$ and $\mathrm{I}^h_{\mathrm{IV}}(m) = \mathrm{I}^h_{\mathrm{IV}}(m')$ denote a non-trivial collision in $\mathrm{I}^h_{\mathrm{IV}}(\cdot)$. Furthermore, let $(z_1, \ldots, z_t)$ and $(z'_1, \ldots, z'_{t'})$ be the sequence of inputs to h in the computation of $\mathrm{I}^h_{\mathrm{IV}}(m)$ and $\mathrm{I}^h_{\mathrm{IV}}(m')$, respectively. Note that $h(z_t) = \mathrm{I}^h_{\mathrm{IV}}(m) = \mathrm{I}^h_{\mathrm{IV}}(m') = h(z'_{t'})$.

Let i denote the smallest index (if any) such that $z_{t-i} \neq z'_{t'-i}$ and $h(z_{t-i}) = h(z'_{t'-i})$. The existence of i directly implies a non-trivial collision in $h(\cdot)$. The non-existence of such an index i implies that one of the sequences $(z_1, \ldots, z_t)$ and $(z'_1, \ldots, z'_{t'})$ is a suffix of the other with $t \neq t'$ since $m \neq m'$. Assume without loss of generality that $t < t'$. In this case we have $\mathrm{IV}\|v = z_1 = z'_{t'-t+1} = h(z_{t'-t})\|v$ for some $v \in \{0,1\}^b$, which means that an output of h equals IV. $\square$

Lemma 6. $\mathrm{I}^h_{\mathrm{IV}}(m) = \mathrm{I}^h_{\mathrm{IV}'}(m')$ *with* $m, m' \in (\{0,1\}^b)^*$ *and* $\mathrm{IV} \neq \mathrm{IV}'$ *imply a non-trivial collision in* h*, or that an output of* h *equals* IV *or* IV$'$*.*

Proof. Let $(z_1, \ldots, z_t)$ and $(z'_1, \ldots, z'_{t'})$ denote the sequence of inputs to h in the computation of $\mathrm{I}^h_{\mathrm{IV}}(m)$ and $\mathrm{I}^h_{\mathrm{IV}'}(m')$, respectively. Note that $h(z_t) = \mathrm{I}^h_{\mathrm{IV}}(m) = \mathrm{I}^h_{\mathrm{IV}'}(m') = h(z'_{t'})$.

Let i denote the smallest index (if any) for which $z_{t-i} \neq z'_{t'-i}$ and $h(z_{t-i}) = h(z'_{t'-i})$. The existence of i directly implies a non-trivial collision in $h(\cdot)$. The non-existence of such an index i implies that one of the sequences $(z_1, \ldots, z_t)$ and $(z'_1, \ldots, z'_{t'})$ is a suffix of the other with $t \neq t'$ since $\mathrm{IV} \neq \mathrm{IV}'$. If $t < t'$ we have $\mathrm{IV}\|v = z_1 = z'_{t'-t+1} = h(z_{t'-t})\|v$ for some $v \in \{0,1\}^b$, which means that an output of h equals IV. Analogously, one shows that if $t > t'$ an output of h equals IV$'$. $\square$

Remark 7. *The Merkle-Damgård (MD) iteration method [Dam89, Mer90] for collision-resistant hashing is a result of similar nature. The computation of the hash value* $\mathrm{MD}^h_{\mathrm{IV}}(m)$*, where* $m \in \{0,1\}^{\leq 2^b}$ *(and* $\mathrm{IV} \in \{0,1\}^\ell$*), is defined by first breaking* m *into a sequence of b-bit blocks* $m_1, \ldots, m_t$ *(where* m_t *is padded with zeroes if necessary) and then returning the value* $\mathrm{I}^h_{\mathrm{IV}}(m_1\| \cdots \|m_t\|\langle|m|\rangle_b)$*. A non-trivial collision in* $\mathrm{MD}^h_{\mathrm{IV}}(\cdot)$ *implies a non-trivial collision in* $h(\cdot)$*.*

5.3.2 The Prefix-Free Iterated Construction

As a first example of an AIL-MAC construction, we present the prefix-free iterated (PI) construction. Let us first introduce the concept of a prefix-free encoding:

Definition 23. *An encoding* $\sigma : \{0,1\}^* \to \{0,1\}^*$ *is called* prefix-free *if there are no three strings* x*,* x'*,* $y \in \{0,1\}^*$ *such that* $x \neq x'$ *and* $\sigma(x)\|y = \sigma(x')$*.*

The construction $\mathrm{PI}^{\langle\cdot\rangle}$ uses a prefix-free encoding

$$\sigma : \{0,1\}^* \to (\{0,1\}^b)^*,$$

to be defined later, for transforming G into the AIL-MAC

$$\mathrm{PI}^{\mathsf{G}} \stackrel{\text{def}}{=} \{\mathrm{PI}^{g_k} : \{0,1\}^* \to \{0,1\}^\ell\}_{k \in \{0,1\}^\kappa},$$

defined by

$$\mathrm{PI}^{g_k}(m) \stackrel{\text{def}}{=} \mathrm{I}^{g_k}_{0^\ell}(\sigma(m)).$$

The on-line property and the efficiency of the construction (hence also the expansion function Φ) depend on which prefix-free encoding σ that is used.

Lemma 7. *For any* t*,* q*,* $\mu \geq 1$*, and prefix-free encoding* σ

$$\mathbf{InSec}^{\mathsf{UF\text{-}CPA}}_{t,q,\mu}(\mathrm{PI}^{\mathsf{G}}) \leq \left(\frac{1}{2}q'^2 + \frac{1}{2}q' + 1\right) \cdot \mathbf{InSec}^{\mathsf{UF\text{-}CPA}}_{t',q'}(\mathsf{G}),$$

where $q' = \Phi(q+1, \mu)$ *and* $t' = t + c$ *for some* c *(depending on* σ*,* q*,* μ*,* L*) that accounts for the overhead implied by the reduction we make.*

Proof. We prove that

$$\mathcal{S} = \mathcal{S}_{\text{coll}} \cup \mathcal{S}_{0^\ell}$$

is complete for $\mathrm{PI}^{\langle\cdot\rangle}$, the rest follows from Proposition 7. Assume $\mathbf{z}$ is not new, then by Lemma 5 and the fact that an old $\mathbf{z}$ implies a non-trivial collision in $\mathrm{I}^{g_k}_{0^\ell}(\cdot)$ (due to the prefix-free encoding), either there is a non-trivial collision in g_k or a 0^ℓ-output of g_k. □

It is an open problem whether there is a prefix-free encoding for which the construction is on-line and has waste $w(n) \in O(\log(n))$.[46] However, allowing linear waste, i.e., $w(n) \in \theta(n)$, there are prefix-free encodings for which the construction has the on-line property. Throughout, we let σ be defined as follows:

Definition 24. *For $m \in \{0,1\}^*$, let*

$$\sigma(m) \stackrel{def}{=} 0\|m_1\|0\|m_2\| \cdots \|0\|m_{t-1}\|1\|m_t,$$

where$m_1, \ldots, m_t$ are $(b-1)$-bit blocks such that $m_1\| \cdots \|m_t = m\|10^\nu$ for a $\nu \in \{0, \ldots, b-2\}$.

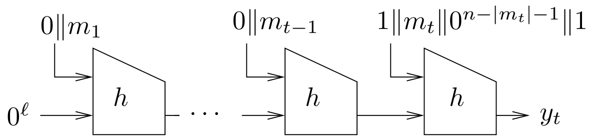


Figure 5.3: The PI-construction (with encoding σ)

The PI-construction, with prefix-free encoding σ (as just defined), is illustrated in figure Figure 5.3. The application function is $\phi(n) = \lceil (n+1)/(b-1) \rceil$ which results in waste $w(n) \in \theta(n)$. We get the following theorem.

Theorem 8. *For any t, q, $\mu \geq 1$ and with σ defined as in Definition 24*

$$\mathbf{InSec}^{\mathsf{UF\text{-}CPA}}_{t,q,\mu}(\mathrm{PI}^{\mathsf{G}}) \leq \left(\frac{1}{2}q'^2 + \frac{1}{2}q' + 1\right) \cdot \mathbf{InSec}^{\mathsf{UF\text{-}CPA}}_{t',q'}(\mathsf{G}),$$

where $q' = \lfloor \frac{\mu}{b-1} \rfloor + (q+1)$ and $t' = t + \mathsf{poly}(q, \mu, L)$ for some polynomial poly *that accounts for the overhead implied by the reduction we make.*

Proof. The proof follows directly from Lemma 7 and the fact that there exist $n_1, \ldots, n_{q+1} \in \mathbb{N}_0$ such that

$$\begin{aligned} \Phi(q+1, \mu) &= \sum_{i=1}^{q+1} \phi(n_i) = \sum_{i=1}^{q+1} \left\lceil \frac{n_i + 1}{b-1} \right\rceil \leq \left\lfloor \sum_{i=1}^{q+1} \frac{n_i + b - 1}{b-1} \right\rfloor \\ &\leq \left\lfloor \frac{\mu}{b-1} \right\rfloor + (q+1) =: q'. \end{aligned}$$

[46]The prefix-free encoding, described next, has logarithmic waste but is not on-line. Let $\sigma' : \{0,1\}^* \to (\{0,1\}^b)^*$ be defined by $r = |\langle |m| \rangle| - 1$ and $\sigma'(m) \stackrel{\text{def}}{=} 0^r 1 \| \langle |m| \rangle \| m \| 0^\nu$, where $\nu \in \{0, \ldots, b-1\}$ is chosen such that the length is a multiple of b.

As a consequence, $\#\mathcal{S} + 1 \leq q'^2/2 + q'/2 + 1$ by (5.1) and (5.2). $\square$

5.3.3 The Double-Iterated Construction

The Double-Iterated (DI) construction transforms any FIL-MAC to an AIL-MAC with constant waste. To be precise, $\mathrm{DI}^{\langle\cdot\rangle}$ uses any FIL-MAC G to construct an AIL-MAC

$$\mathrm{DI}^{\mathsf{G}} \stackrel{\text{def}}{=} \{\mathrm{DI}^{g_k} : \{0,1\}^* \to \{0,1\}^\ell\}_{k\in\{0,1\}^\kappa},$$

defined by

$$\mathrm{DI}^{g_k}(m) \stackrel{\text{def}}{=} \begin{cases} \mathrm{I}_{1^\ell}^{g_k}\left(\mathrm{I}_{0^\ell}^{g_k}(m_1\|\cdots\|m_{t-1})\|m_t\right) & \text{if } t > 1 \\ \mathrm{I}_{1^\ell}^{g_k}(0^\ell\|m_1) & \text{otherwise} \end{cases},$$

where the message $m \in \{0,1\}^*$ (of length n) is broken into a sequence of b-bit blocks $m_1, \ldots, m_{t-1}$ (if $t > 1$) and a $(\lceil \ell/b \rceil b - \ell)$-bit block m_t, where a 1 followed by 0's is used as padding, i.e., $m_1\|\cdots\|m_t = m\|10^\nu$ for some $\nu \in \{0, \ldots, b-1\}$. The computation is illustrated in Figure 5.4 (for the case when $b > \ell$).

The application function is $\phi(n) = \left\lceil \frac{n+1+\ell}{b} \right\rceil$ (resulting in the waste $w(n) \in \Theta(1)$).

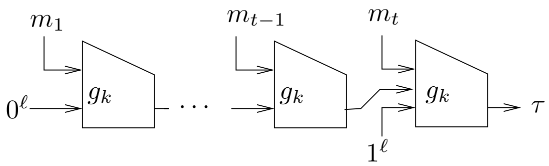


Figure 5.4: The Double-Iterated (DI) construction for the case when $b > \ell$

Note that DI is more efficient than PI if (and only if) the message length is at least $\ell(b-1)$. The next theorem states that the DI-construction preserves unforgeability.[47]

[47] Recently, Bellare and Ristenpart [BR06] have shown that DI preserves several other properties (such as collision resistance, pseudorandomness, etc.) when the last block m_t encodes the message length.

Theorem 9. *For any* t, q, $\mu \geq 1$

$$\mathbf{InSec}^{\mathsf{UF\text{-}CPA}}_{t,q,\mu}(\mathrm{DI}^{\mathsf{G}}) \leq \left(\frac{1}{2}q'^2 + \frac{3}{2}q' + 1\right) \cdot \mathbf{InSec}^{\mathsf{UF\text{-}CPA}}_{t',q'}(\mathsf{G}),$$

where $q' = \lfloor \frac{\mu}{b} + \frac{b+\ell}{b} \cdot (q+1) \rfloor$ *and* $t' = t + \mathsf{poly}(q, \mu, L)$ *for some polynomial* poly *that accounts for the overhead implied by the reduction we make.*

Proof of Theorem 9. We prove that $\mathcal{S} = \mathcal{S}_{\mathrm{coll}} \cup \mathcal{S}_{0^\ell} \cup \mathcal{S}_{1^\ell}$ is complete for $\mathrm{DI}^{\langle\cdot\rangle}$, the rest follows from Proposition 7. Let us assume that $\mathbf{z}$ is not new, that no non-trivial collision in g_k occurs, and that no output of g_k equals 0^ℓ or 1^ℓ. Then by Lemma 5, there can not be a non-trivial collision in $\mathrm{I}^{g_k}_{0^\ell}(\cdot)$. Furthermore, no output of $\mathrm{I}^{g_k}_{0^\ell}(\cdot)$ equals 0^ℓ, since this would directly imply a 0^ℓ-output of g_k. As a consequence, the last input $\tilde{m}$ to $\mathrm{I}^{g_k}_{1^\ell}(\cdot)$ is distinct from the other inputs to $\mathrm{I}^{g_k}_{1^\ell}(\cdot)$.[48] Since $\mathbf{z}$ is not new, $\mathbf{z}$ must have been an earlier query to g_k, resulting from some query $m' = m'_1 \| \cdots \| m'_{t'}$ to $\mathrm{I}^{g_k}_{\mathrm{IV}}(\cdot)$ with $\mathrm{IV} \in \{0^\ell, 1^\ell\}$. Let $z'_1, \ldots, z'_{t'}$ denote the sequence of queries to g_k in the computation of $\mathrm{I}^{g_k}_{\mathrm{IV}}(m')$ and let s be the index for which $z'_s = \mathbf{z}$. Thus, we have $\mathrm{I}^{g_k}_{\mathrm{IV}}(m'_1 \| \cdots \| m'_s) = \mathrm{I}^{g_k}_{1^\ell}(\tilde{m})$. There are two cases to distinguish:

- If $\mathrm{IV} = 0^\ell$, we arrive at a contradiction by Lemma 6.
- If $\mathrm{IV} = 1^\ell$, it follows from the construction that $|m'| = |\tilde{m}|$. Thus, we have $m'_1 \| \cdots \| m'_s \neq \tilde{m}$, since $\tilde{m}$ is distinct (from the other queries to $\mathrm{I}^{g_k}_{1^\ell}(\cdot)$). As a consequence, we arrive at a contradiction by Lemma 5.

By definition of $\Phi(q+1, \mu)$, there exist $n_1, \ldots, n_{q+1} \in \mathbb{N}_0$ such that:

$$\Phi(q+1, \mu) = \sum_{i=1}^{q+1} \phi(n_i) = \sum_{i=1}^{q+1} \left\lceil \frac{n_i + 1 + \ell}{b} \right\rceil \leq \left\lfloor \frac{\mu + (b+\ell)(q+1)}{b} \right\rfloor =: q'.$$

Thus $\#\mathcal{S} + 1 \leq q'^2/2 + 3q'/2 + 1$ by (5.1) and (5.2). □

PARALLELIZING THE DI-CONSTRUCTION. Let us modify DI to allow $c \geq 1$ processors to compute the tag in parallel, achieving a speed up by a factor of roughly c for long messages. The tag τ of an n-bit message m is computed according to the following recursion:

1. If $c \leq \lceil (n+1)/b \rceil$ then set $c' := c$, and else set $c' := \lceil (n+1)/b \rceil$.

[48] Recall that, without loss of generality, we assume that the forgery message m of A is distinct from its oracle queries.

2. Parse m into $m_1\|\cdots\|m_{c't} = m\|10^\nu$, where $m_1, \ldots, m_{c't}$ are b-bit blocks and $\nu \in \{0, \ldots, c'b-1\}$. Set $m_{i,j} := m_{i+(j-1)c'}$ for $i \in \{1, \ldots, c'\}$ and $j \in \{1, \ldots, t\}$.

3. Set $y_{i,0} := 0^\ell$, and let $y_{i,j} := g_k(y_{i,j-1}\|m_{i,j})$ for $i \in \{1, \ldots, c'\}, j \in \{1, \ldots, t\}$.

4. Return $\tau := \mathrm{DI}^{g_k}(y_{1,t}\|\cdots\|y_{c',t})$.[49]

The waste remains constant and the on-line property is preserved. We omit the proof that $\mathcal{S} = \mathcal{S}_{\text{coll}} \cup \mathcal{S}_{0^\ell} \cup \mathcal{S}_{1^\ell}$ is complete for the construction, as it is similar to the proof that $\mathcal{S}$ is complete for the DI-construction.

BETTER EFFICIENCY FOR SHORT MESSAGES. In various applications, messages of short length are more frequent than long messages. For such applications it is obviously crucial that (also) short messages are processed efficiently. Next, we improve the efficiency of the DI-construction for $n \stackrel{\text{def}}{=} |m| < rb$, where $r \in \mathbb{N}_0$ is a design parameter. The computation of the tag τ is redefined for messages m of length shorter than rb as follows. Parse m into a sequence of b-bit blocks $m_1, \ldots, m_t$ such that $m_1\|\ldots\|m_t = m\|10^\nu$ where $\nu \in \{0, \ldots, b-1\}$:

$$y_0 := \langle t \rangle_\ell,\ y_i := g_k(y_{i-1}\|m_i) \text{ for } i \in \{1, \ldots, t\}, \text{ and } \tau := y_t.$$

Now, $\phi(n) = \lceil (n+1)/b \rceil$ if $n < rb$ (and $\phi(n) = \lceil (n+1+\ell)/b \rceil$ if $n \geq rb$). The proof that $\mathcal{S}_{\text{coll}} \cup \mathcal{S}_{0^\ell} \cup \mathcal{S}_{1^\ell} \cup (\cup_{i=1}^r \mathcal{S}_{\langle i \rangle_\ell})$ is complete for the construction is omitted. The only modification of Theorem 9 is thus that

$$\mathbf{InSec}^{\mathsf{UF\text{-}CPA}}_{t,q,\mu}(\mathrm{DI}^{\mathsf{G}}) \leq \left(\frac{1}{2}q'^2 + \left(\frac{3}{2} + r\right)q' + 1\right) \cdot \mathbf{InSec}^{\mathsf{UF\text{-}CPA}}_{t',q'}(\mathsf{G}),$$

i.e., the reduction is essentially as tight (as for $r = 0$) for reasonable r's. The disadvantage, however, is that the construction is not completely on-line (one must know if the message is short or long).

In the next section, we propose a new construction that is on-line and which is more efficient than the DI-construction for short messages, at the expense of being slightly less efficient for long messages.

[49] The construction can be further parallelized by replacing step 4 as follows. For simplicity assume $b = \ell$ (the generalization to $b \geq \ell$ is straight forward). Apply g_k to every pair of adjacent blocks in $(y_{1,t}, \cdots, y_{c',t})$, resulting in a new sequence of $\lceil c'/2 \rceil$ blocks, and repeat this until a single block y is obtained. Then set $\tau := I_{1^\ell}^{g_k}(y)$.

By setting $c := \infty$ this construction is *fully* parallelized (FP) (here meaning that the computation time is in $\Theta(\log(n))$ when arbitrary many processors are available) with $w(n) \in \Theta(n)$.

5.3.4 The Prefix-Free Double-Iterated Construction

The Prefix-Free Double-Iterated Construction (PDI) is an AIL-MAC construction, which is a hybrid construction between PI and DI. It exploits the advantage of PI for being efficient for short messages and the advantage of DI for being efficient for long messages. It is defined as follows.

Let $r \in \mathbb{N}_0$ be a design parameter. The construction $\mathrm{PDI}_r^{\langle\cdot\rangle}$ transforms any FIL-MAC G into the AIL-MAC

$$\mathrm{PDI}_r^{\mathsf{G}} \overset{\text{def}}{=} \{\mathrm{PDI}_r^{g_k} : \{0,1\}^* \to \{0,1\}^\ell\}_{k\in\{0,1\}^\kappa},$$

where $\mathrm{PDI}_r^{g_k}(m)$ for a message $m \in \{0,1\}^*$ (of length n) is defined as

$$\mathrm{PDI}_r^{g_k}(m) \overset{\text{def}}{=} \begin{cases} \mathrm{PI}^{g_k}(m) & \text{if } n < r(b-1) \\ \mathrm{DI}^{g_k}(0\|m_1\|0\|m_2\|\cdots\|0\|m_r\|m_{r+1}) & \text{otherwise} \end{cases},$$

where (for $n \geq r(b-1)$) the message m is parsed into $(b-1)$-bit blocks $m_1,\ldots,m_r$ and a bitstring m_{r+1} such that $m_1\|\cdots\|m_r\|m_{r+1} = m$. The application function is

$$\phi(n) = \begin{cases} \left\lceil \frac{n+1}{b-1} \right\rceil & \text{if } n < r(b-1) \\ \left\lceil \frac{n+1+\ell+r}{b} \right\rceil & \text{otherwise} \end{cases}.$$

Although not directly clear from the definition above, this construction is on-line (no matter whether $|m| < r(b-1)$ or not, the processing of m starts out in the same way).

We stress that PDI is equivalent to DI for $r = 0$ and to PI for $r \to \infty$. As is obvious from the definition of $\mathrm{PDI}_r^{\langle\cdot\rangle}$, the construction is as efficient as $\mathrm{PI}^{\langle\cdot\rangle}$ for messages of shorter length than $r(b-1)$ and slightly less efficient than $\mathrm{DI}^{\langle\cdot\rangle}$ otherwise.

Theorem 10. *For any t, q, $\mu \geq 1$*

$$\mathbf{InSec}_{t,q,\mu}^{\mathsf{UF\text{-}CPA}}(\mathrm{PDI}_r^{\mathsf{G}}) \leq \left(\frac{1}{2}q'^2 + \left(\frac{1}{2}+\xi\right)q' + 1\right)\cdot \mathbf{InSec}_{t',q'}^{\mathsf{UF\text{-}CPA}}(\mathsf{G}),$$

where $q' = \left\lfloor \frac{\mu}{b-1} + (q+1) + \frac{\ell+r}{b}\cdot\Lambda - \frac{1}{b\cdot(b-1)}\cdot\Pi \right\rfloor$,

$$(\Lambda,\Pi) = \begin{cases} (q+1,\mu) & \textit{if } r=0 \\ \left(\left\lfloor \frac{\mu}{r(b-1)} \right\rfloor, 0\right) & \textit{if } \frac{\mu}{q+1} \leq r(b-1)-1 \\ \left(\min\left(q+1, \left\lfloor \frac{\mu}{r(b-1)} \right\rfloor\right), \mu - q\,(r\,(b-1)-1)\right) & \textit{otherwise} \end{cases},$$

ξ takes the value 1 if $\mu \geq r \cdot (b-1)$ and 0 otherwise, and $t' = t + \mathsf{poly}(q, \mu, L)$ for some polynomial poly *that accounts for the overhead implied by the reduction we make.*

Proof (sketch). Let ξ be an indicator variable that takes the value 1 if $\mu \geq r \cdot (b-1)$ and 0 otherwise. We omit the proof that $\mathcal{S}_{\text{coll}} \cup \mathcal{S}_{0^\ell}$ is complete for the construction if $\xi = 0$ and that $\mathcal{S}_{\text{coll}} \cup \mathcal{S}_{0^\ell} \cup \mathcal{S}_{1^\ell}$ is complete for the PDI-construction otherwise, since it is similar to the proof of the DI- and PI-construction.[50] By definition of $\Phi(q+1, \mu)$, there exist $n_1, \ldots, n_{q+1} \in \mathbb{N}_0$ such that $\Phi(q+1, \mu) = \sum_{i=1}^{q+1} \phi(n_i)$. By letting ζ_i be an indicator variable that takes value 1 if $n_i \geq r(b-1)$ and 0 otherwise, we get

$$\begin{aligned}
\sum_{i=1}^{q+1} \phi(n_i) &\leq \sum_{i=1}^{q+1} \zeta_i \cdot \left\lceil \frac{n_i + 1 + \ell + r}{b} \right\rceil + (1 - \zeta_i) \cdot \left\lceil \frac{n_i + 1}{b - 1} \right\rceil \\
&\leq \sum_{i=1}^{q+1} \zeta_i \cdot \frac{n_i + b + \ell + r}{b} + (1 - \zeta_i) \cdot \frac{n_i + b - 1}{b - 1} \\
&\leq \left\lfloor \frac{\mu}{b-1} + (q+1) + \frac{\ell + r}{b} \cdot \sum_{i=1}^{q+1} \zeta_i - \frac{1}{b \cdot (b-1)} \sum_{i=1}^{q+1} \zeta_i \cdot n_i \right\rfloor .
\end{aligned}$$

Furthermore, it is easy to verify that the following two inequalities hold

$$\sum_{i=1}^{q+1} \zeta_i \leq \begin{cases} q+1 & \text{if } r = 0 \\ \min\left(q+1, \left\lfloor \frac{\mu}{r(b-1)} \right\rfloor\right) & \text{otherwise} \end{cases} =: \Lambda$$

$$\sum_{i=1}^{q+1} \zeta_i \cdot n_i \geq \begin{cases} \mu & \text{if } r = 0 \\ 0 & \text{if } \frac{\mu}{q+1} \leq r(b-1) - 1 \\ \mu - q(r(b-1) - 1) & \text{otherwise} \end{cases} =: \Pi.$$

As a consequence, $\#\mathcal{S} + 1 \leq q'^2/2 + (1/2 + \xi) \cdot q' + 1$ by (5.1) and (5.2). And by applying Proposition 7 the proof is concluded. $\square$

5.4 The Generalized Construction Paradigm

In this section, we generalize the construction paradigm to comprise a greater class of constructions. Furthermore, we investigate a tradeoff be-

[50] Note that if $\mu < r(b-1)$ all queries issued by the forger for $\mathrm{PDI}_r^{g_k}(\cdot)$ (including the forgery message) are shorter than $r(b-1)$ and hence $\mathrm{DI}^{g_k}(\cdot)$ is never invoked.

tween the efficiency of a construction and the tightness of the security reduction in detail.

5.4.1 An Efficiency/Security Tradeoff

A general design goal of AIL-MAC constructions is to minimize the number of applications $\phi(n)$ of the FIL-MAC (where n denotes the message length). A natural approach to decrease the number of applications, that is not implied by the type of construction $\mathrm{C}^{\langle\cdot\rangle}$, is to increase the compression parameter of the FIL-MAC before it is transformed by some construction $\mathrm{C}^{\langle\cdot\rangle}$. However, this is at the cost of a less tight security reduction.

To be more precise, let $f : \{0,1\}^{\ell} \to \{0,1\}^{\ell-\delta}$ be a compression function with compression parameter $\delta > 0$ and let $f^{-1}(y)$ denote the set of all preimages[51] of $y \in \{0,1\}^{\ell-\delta}$. Let $[\cdot]_f$ denote the construction, which transforms G into a FIL-MAC

$$[\mathsf{G}]_f \stackrel{\text{def}}{=} \{[g_k]_f : \{0,1\}^L \to \{0,1\}^{\ell-\delta}\}_{k\in\{0,1\}^{\kappa}},$$

defined by

$$[g_k]_f(x) \stackrel{\text{def}}{=} f(g_k(x)).$$

Lemma 8. *A (t, q, ε)-forger A for $[\mathsf{G}]_f$ implies a $(t', q, \varepsilon/s)$-forger A' for G, where $s = \max\{\#f^{-1}(y) : y \in \{0,1\}^{\ell-\delta}\}$ and $t' = t + t''$ where t'' is the size of a circuit for inverting f.*

Proof. The forger A' runs A, answering all its oracle queries with the help of its own oracle. When A returns a forgery (m, τ), A' chooses an element $\hat{\tau}$ uniformly at random from $f^{-1}(\tau)$ and outputs $(m, \hat{\tau})$ as its own forgery. If A' is successful it follows that $\tau = [g_k]_f(m) = f(g_k(m))$. Thus, there is an element $\tau' \in f^{-1}(\tau)$ for which $\tau' = g_k(m)$. The probability that $\hat{\tau} = \tau'$ is

$$1/\#f^{-1}(\tau) \geq 1/s, \quad \text{where} \quad s = \max\{\#f^{-1}(y) : y \in \{0,1\}^{\ell-\delta}\}.$$

Let $\mathcal{E}'$ denote the event that A' is successful and $\mathcal{E}$ the event that A is successful. Then

$$\Pr[\mathcal{E}'] \geq \Pr[\mathcal{E}' \mid \mathcal{E}] \cdot \Pr[\mathcal{E}] \geq \underbrace{\Pr[\hat{\tau} = \tau']}_{\geq 1/s} \cdot \underbrace{\Pr[\mathcal{E}]}_{=\varepsilon}.$$

□

[51] We assume that, for all $y \in \{0,1\}^{\ell-\delta}$, one can efficiently sample an element uniformly at random from $f^{-1}(y)$.

To get as tight a security reduction in Lemma 8 as possible the largest preimage set of the key-less compression function must be as small as possible. A function achieving this is

$$\Delta_\delta : \{0,1\}^\ell \to \{0,1\}^{\ell-\delta}, \quad \text{defined by} \quad x \mapsto x[1, \ell - \delta],$$

which simply cuts off the δ least significant bits of the input. As a consequence, Δ_δ can always be chosen as the compression function without loss of generality. To simplify the notation, we write $[\cdot]_\delta$ to denote the construction $[\cdot]_{\Delta_\delta}$.

Corollary 1. *A (t, q, ε)-forger for $[\mathsf{G}]_\delta$ implies a $(t, q, \varepsilon/2^\delta)$-forger for G.*

Proof. Since each image of $\Delta_\delta(\cdot)$ has equally many preimages, namely 2^δ, the largest preimage set is as small as possible. Apply Lemma 8. □

5.4.2 Generalized Constructions

The AIL-MAC $\mathrm{C}^{[\mathsf{G}]_\delta}$ is defined by simply letting the construction $\mathrm{C}^{\langle\cdot\rangle}$ transform the FIL-MAC $[\mathsf{G}]_\delta$, which has compression parameter $b' = b + \delta$ and output length $\ell' = \ell - \delta$. This is illustrated in Figure 5.5. Since $[\mathsf{G}]_\delta$

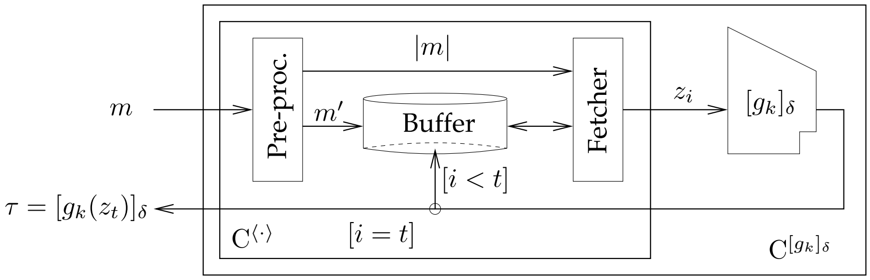


Figure 5.5: The generalized construction paradigm

compresses more than G, the number of applications of the FIL-MAC G is in general smaller for $\mathrm{C}^{[\cdot]_\delta}$ than for $\mathrm{C}^{\langle\cdot\rangle}$. However, this is at the cost of having a less tight security reduction for $\mathrm{C}^{[\mathsf{G}]_\delta}$ by a factor of roughly 2^δ.

Corollary 2. *Let b denote the compression parameter and ℓ the output length of a FIL-MAC G.*[52] *If $\phi_{b,\ell}(n)$ is the application function of $\mathrm{C}^{\langle\cdot\rangle}$, then $\phi_{b+\delta,\ell-\delta}(n)$ is the application function of $\mathrm{C}^{[\cdot]_\delta}$. Further, if a (t,q,μ,ε)-forger for C^{G} implies a (t',q',ε')-forger for G, where*

$$q' = q'_{b,\ell}(t,q,\mu,\varepsilon),\ \ \varepsilon' = \varepsilon'_{b,\ell}(t,q,\mu,\varepsilon),\ \ \textit{and}\ \ t' = t'_{b,\ell}(t,q,\mu,\varepsilon),$$

then a (t,q,μ,ε)-forger for $\mathrm{C}^{[\mathsf{G}]_\delta}$ implies a $(t'',q'',\varepsilon''/2^\delta)$-forger for G, where

$$q'' = q'_{b+\delta,\ell-\delta}(q,\mu,\varepsilon)\ \ \varepsilon'' = \varepsilon'_{b+\delta,\ell-\delta}(q,\mu,\varepsilon),\ \ \textit{and}\ \ t'' = t''_{b+\delta,\ell-\delta}(t,q,\mu,\varepsilon).$$

Proof. The FIL-MAC $[\mathsf{G}]_\delta$ has compression parameter $b' = b+\delta$ and output length $\ell' = \ell - \delta$. Apply Corollary 1. □

As we illustrate next, the tradeoff between the efficiency and the tightness should be taken into account when comparing AIL-MAC constructions with each other.

AN ILLUSTRATIVE EXAMPLE. To illustrate the generalization and the security/efficiency tradeoff, let us introduce a new AIL-MAC construction which we call Chain-Rotate (CR). The CR-construction transforms any FIL-MAC G into the AIL-MAC

$$\mathrm{CR}^{\mathsf{G}} \stackrel{\text{def}}{=} \{\mathrm{CR}^{g_k} : \{0,1\}^* \to \{0,1\}^\ell\}_{k\in\{0,1\}^\kappa},$$

as illustrated in Figure 5.6. The application function is $\phi(n) = \lceil \frac{n+1}{b} \rceil$ and as a consequence the waste $w(n) \in \Theta(1)$. To be more precise, let $\mathcal{RR}(\cdot)$ denote the right-rotate operator on bitstrings, i.e.,

$$\mathcal{RR}(x) \stackrel{\text{def}}{=} x[|m|]\|x[1,|m|-1].$$

The tag $\tau = \mathrm{CR}^{g_k}(m)$ for a message $m \in \{0,1\}^*$ is computed by first parsing m into a sequence $\{m_i\}_{i=1}^t$ of b-bit blocks such that $m_1\|\cdots\|m_t = m\|10^\nu$ for a $\nu \in \{0,\dots,b-1\}$, and then let

$$\mathrm{CR}^{g_k}(m) \stackrel{\text{def}}{=} g_k(\mathcal{RR}\left(y\|m_t\right)),\ \text{where } y := \begin{cases} \mathrm{I}^{g_k}_{0^\ell}\ (m_1\|\cdots\|m_{t-1}) & \text{if } t>1 \\ 0^\ell & \text{otherwise} \end{cases}.$$

[52] Here we make the parameters b and ℓ explicit.

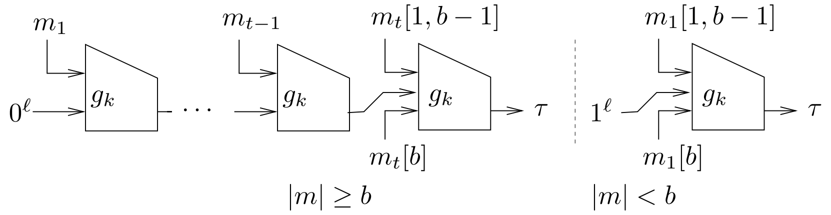


Figure 5.6: The Chain-Rotate (CR) construction for the case when $|m| \geq b$ (left) and when $|m| < b$ (right).

Theorem 11. *For any t, q, $\mu \geq 1$*

$$\mathbf{InSec}^{\textsf{UF-CPA}}_{t,q,\mu}(\mathrm{CR}^{\mathsf{G}}) \leq \left(\frac{5}{2} \cdot q'^2 + \left(\frac{3}{2} + I\right) q' + 1\right) \cdot \mathbf{InSec}^{\textsf{UF-CPA}}_{t',q'}(\mathsf{G}),$$

where $q' = \lfloor \frac{\mu}{b} \rfloor + (q+1)$ and $t' = t + \mathsf{poly}(q, \mu, L)$ for some polynomial poly *which accounts for the overhead implied by the reduction we make.*

Proof. We prove that

$$\mathcal{S} = \mathcal{S}_{\text{coll}} \cup \mathcal{S}_{\text{coll2}} \cup \mathcal{S}_{0^\ell} \cup \mathcal{S}_{1^\ell} \cup \mathcal{S}_{0^{\ell-1}1} \cup \mathcal{S}_{01^{\ell-1}}$$

is complete for $\mathrm{CR}^{\langle\cdot\rangle}$, the rest follows from Proposition 7. Let us assume that the last entry $\mathbf{z}$ of $\mathbf{Z}$ (in the experiment $\mathsf{A}\blacklozenge\mathrm{CR}^{g_k}$) is not new. We now show that this implies a non-trivial collision in g_k, a collision of type 2, or an output from g_k equals $0^\ell, 1^\ell, 0^{\ell-1}1$ or $01^{\ell-1}$. Let $\tilde{z}_1, \ldots, \tilde{z}_t$ denote the sequence of queries to g_k resulting from the last query m_β to CR^{g_k}. As m_β is the forgery message of A it is distinct from the previous queries to CR^{g_k} (per assumption). And since $\tilde{z}_t = \mathbf{z}$ is not new, $\tilde{z}_t$ must have been an earlier query to g_k, resulting from some query m_α (with $\alpha \leq \beta$) to CR^{g_k}. Let $\tilde{z}'_1, \ldots, \tilde{z}'_{t'}$ denote the sequence of queries to g_k in the computation of $\mathrm{CR}^{g_k}(m_\alpha)$. There are three cases to distinguish depending on the index $i \in \{1, \ldots, t'\}$ for which $\tilde{z}_t = \tilde{z}'_i$.

At the end of the chain ($\tilde{z}_t = \tilde{z}'_{t'}$): First, we note that this can not be the case if $\alpha = \beta$, since in this case $\tilde{z}'_{t'}$ is not an earlier occurring query. Thus a non-trivial collision must occur, i.e., $m_\alpha \neq m_\beta$ and $\mathrm{CR}^{g_k}(m_\alpha) = \mathrm{CR}^{g_k}(m_\beta)$. Without loss of generality, we assume that $t' \geq t$. Thus, either there exists an $i \in \{1, \ldots, t-1\}$ (if $t > 1$ that is) such that $\tilde{z}_{t-i} \neq \tilde{z}'_{t'-i}$ and $\tilde{z}_{t-i+1} = \tilde{z}'_{t'-i+1}$ (i.e., $g_k(\tilde{z}_{t-i}) =$

$g_k(\tilde{z}'_{t'-i})$), which implies a non-trivial collision for g_k, or else $t' > t$ (since $m_\alpha \neq m_\beta$) and $\tilde{z}_1 = \tilde{z}'_{t'-t+1} = g_k(\tilde{z}_{t'-t})\|x$ for some $x \in \{0,1\}^b$, which implies a 0^ℓ- or 1^ℓ-output of g_k (since depending on whether $t = 1$ or $t > 1$, $\tilde{z}_1[1,\ldots,\ell]$ equals 1^ℓ or 0^ℓ).

In the middle of the chain ($\tilde{z}_t = \tilde{z}'_i$ with $1 < i < t'$): We have that $\tilde{z}_t = \mathcal{RR}(y\|v) = g_k(\tilde{z}'_{i-1})\|v' = \tilde{z}'_i$ for some $v, v' \in \{0,1\}^b$ and $y \in \{0,1\}^\ell$. If $t = 1$ it follows that $y = 1^\ell$, which implies that $g_k(\tilde{z}'_{i-1}) = b\|1^{\ell-1}$ for some $b \in \{0,1\}$. If $t > 1$ we have that $y = g_k(\tilde{z}_{t-1})$ which implies that $g_k(\tilde{z}_{t-1})[1,\ell-1] = g_k(\tilde{z}'_{i-1})[2,\ell]$. Thus if $\tilde{z}_{t-1} \neq \tilde{z}'_{i-1}$ we have a collision of type 2 and else a 0^ℓ- or 1^ℓ-output from g_k.

At the beginning of the chain (of length > 1) ($\tilde{z}_t = \tilde{z}'_1$): This is not possible if $t = 1$ since $\tilde{z}_t = \mathcal{RR}(1^\ell\|v) \neq 0^\ell\|v' = \tilde{z}'_1$ (for some $v, v' \in \{0,1\}^b$). So let us assume $t > 1$. Then $\tilde{z}_t = \mathcal{RR}(g_k(\tilde{z}_{t-1})\|v) = 0^\ell\|v'$ for some $v, v' \in \{0,1\}^b$ which implies $g_k(\tilde{z}_{t-1}) = 0^{\ell-1}\|b$ for $b \in \{0,1\}$.

As there exist $n_1,\ldots,n_{q+1} \in \mathbb{N}_0$ such that

$$\Phi(q+1,\mu) = \sum_{i=1}^{q+1} \phi(n_i) \leq \sum_{i}^{q+1} \left\lceil \frac{n_i+1}{b} \right\rceil \leq \left\lfloor \sum_{i=1}^{q+1} \frac{n_i+b}{b} \right\rfloor \leq \left\lfloor \frac{\mu}{b} \right\rfloor + (q+1) =: q',$$

we get $\#\mathcal{S} + 1 \leq 5q'^2/2 + 3q'/2 + 1$ by (5.1), (5.2), and (5.3). □

The efficiency of $\mathrm{CR}^{\langle\cdot\rangle}$ is better than for $\mathrm{DI}^{\langle\cdot\rangle}$ and $\mathrm{PI}^{\langle\cdot\rangle}$ (just compare the application functions). However, note that the tightness of the security reduction is roughly a factor 5 worse. At first sight one might be tempted to neglect the factor 5 and consider the CR-construction as the better construction. However, by applying Corollary 2 it is straight forward to verify that (for any fixed δ) the application function is equivalent for $\mathrm{PI}^{[\cdot]_{\delta+1}}$ and $\mathrm{CR}^{[\cdot]_\delta}$ (and hence the efficiency is the same), but that the security reduction of $\mathrm{PI}^{[\cdot]_{\delta+1}}$ is tighter by a factor of roughly 2.5. This illustrates the importance of taking the security/efficiency tradeoff into account when comparing AIL-MAC constructions.

THE GENERALIZED PDI-CONSTRUCTION. As $\mathrm{PDI}_r^{[\mathsf{G}]_\delta}$ is equivalent to $\mathrm{DI}^{[\mathsf{G}]_\delta}$ for $r = 0$ and to $\mathrm{PI}^{[\mathsf{G}]_\delta}$ for $r \to \infty$, we conclude that for all AIL-MAC constructions – given in the literature – there is a choice for r and δ for which $\mathrm{PDI}_r^{[\cdot]_\delta}$ is as efficient and secure. The concrete choice for δ and the design parameter r is application dependent.

5.5 Domain Extensions with Multiple Keys

Although, we have argued that single-key constructions are essentially optimal (in all aspects), we nevertheless extend the proof technique from Section 5.2.2 to comprise 2-key constructions. The reason is that we want to revisit the Nested Iterated (NI) construction of An and Bellare [AB99] for completeness, and also exemplify the usefulness of our technique by deriving some improvements of NI.

5.5.1 Security Reduction (2 Keys)

Motivated by the NI-construction (see Figure 5.7), we consider constructions $\mathrm{C}^{\langle\cdot,\cdot\rangle}$ (using two instantiations g_{k_1} and g_{k_2} of G to construct an instantiation $\mathrm{C}^{g_{k_1},g_{k_2}}$ of $\mathrm{C}^{\mathsf{G},\mathsf{G}}$), where one of the instantiations (say g_{k_2} without loss of generality) is invoked at the end of the computation.

Let A denote a (t, q, μ, ε)-forger for $\mathrm{C}^{\mathsf{G},\mathsf{G}}$ and as before let

$$\mathsf{A}\blacklozenge\mathrm{C}^{g_{k_1},g_{k_2}}$$

denote the process in which for each query $\tilde{m}$ issued by A, the corresponding tag $\mathrm{C}^{g_{k_1},g_{k_2}}(\tilde{m})$ is computed and returned to A, and once A returns a forgery (m, τ), the forgery is verified by computing $\mathrm{C}^{g_{k_1},g_{k_2}}(m)$. For $i = 1, 2$, let $\mathbf{Z}^i := (z_1^i, z_2^i, \ldots)$ and $\mathbf{Y}^i := (y_1^i, y_2^i, \ldots)$ be the sequence of inputs respectively outputs occurring at the interface to instantiation g_{k_i}.

We now consider the a forger A' for G that simulates $\mathsf{A}\blacklozenge\mathrm{C}^{g_{k_1},g_{k_2}}$ by letting its own oracle simulate one of the instantiations g_{k_i} (say the instantiation *under attack*) and by choosing a random key for the other, but stops the simulation at some query z_j^i to its oracle and returns a forgery (without making any further query to any instantiation of G). This is equivalent to first instantiating g_{k_1} and g_{k_2} (by choosing the keys k_1, k_2 uniformly at random) and then letting A' specify which instantiation to attack, i.e., consider as its own oracle, after which the key to the other instantiation is revealed to A'. We adopt this view. Any such forger is characterized by its *strategy*, i.e., which instantiation it attacks (i.e., i), the moment it stops (i.e., j), and the way it produces its forgery.

Let s_{na} denote the naïve strategy described in Section 5.2.2, with the only modification that the second instantiation, g_{k_2} is put under attack (recall that the tag τ is an output of g_{k_2}). $\mathsf{A}'_{s_{\mathrm{na}}}$ stops at the very last query $\mathbf{z}$

to g_{k_2} and returns $(\mathbf{z}, \tau)$ as a forgery. Of course A' is successful if the following two conditions hold. First, $\mathcal{E}$ occurs, i.e., $\mathrm{C}^{g_{k_1}, g_{k_2}}(m) = \tau$ (and thus $g_{k_2}(\mathbf{z}) = \tau$),[53] and second $\mathcal{E}_{new}$ holds, i.e., $\mathbf{z}$ is new for g_{k_2} or equivalently $\mathbf{z}$ is only the last entry in $\mathbf{Z}^2$.

Imagine as before, that a *complete* set of strategies $\mathcal{S}$ exists, i.e., a set for which whenever $\bar{\mathcal{E}}_{\text{new}}$ occurs, there exists a strategy $s \in \mathcal{S}$ such that A'_s is successful. Then an overall strategy of A' is to pick its strategy uniformly at random from $\mathcal{S} \cup \{s_{\text{na}}\}$. Its success probability is at least the probability that $\mathcal{E}$ occurs (i.e., ε) divided by the number $\#\mathcal{S} + 1$ of strategies, since the choice of strategy is independent of the event $\mathcal{E}$. Since A''s number of queries to its oracle is upper bounded by $\max\{|\mathbf{Z}^1|, |\mathbf{Z}^2|\}$, which is a random variable, it is convenient to introduce the expansion function for each instantiation, i.e., for $i \in \{1, 2\}$ (where ϕ^i denotes the application function for the i-th instantiation of G)

$$\Phi^i(\tilde{q}, \tilde{\mu}) \stackrel{\text{def}}{=} \max\left\{ \sum_{j=1}^{\tilde{q}} \phi^i(n_j) : n_1, \ldots, n_{\tilde{q}} \in \mathbb{N}_0,\ n_1 + \cdots + n_{\tilde{q}} \leq \tilde{\mu} \right\}.$$

Thus $|\mathbf{Z}^i| \leq \Phi^i(q + 1, \mu)$. Proposition 7 generalizes as follows.

Proposition 8. *The existence of a complete set $\mathcal{S}$ for a construction $\mathrm{C}^{\langle \cdot, \cdot \rangle}$ and a (t, q, μ, ε)-forger A for $\mathrm{C}^{\mathsf{G},\mathsf{G}}$ implies a (t', q', ε')-forger A' for G, where $q' = \max(\Phi^1(q+1, \mu), \Phi^2(q+1, \mu))$, $\varepsilon' = \frac{\varepsilon}{\#\mathcal{S}+1}$, and $t' = t + t''$, where t'' denotes the size of any circuit which picks a strategy at random from $\mathcal{S}$ and runs it using one black-box invocation to A and at most q' to its oracle G.*

A *deterministic* strategy s is now characterized by a triple of values (i, j, f), where i denotes the instantiation to attack, z^i_j the moment to stop, and f a function mapping $(\mathbf{Z}^i_j, \mathbf{Y}^i_{j-1})$ to some value $\hat{y}^i_j \in \{0, 1\}^\ell$. The pair $(z^i_j, \hat{y}^i_j)$ is the forgery of A'_s. The sets of deterministic strategies introduced in Section 5.2.3 is naturally defined for each instantiation. Let $\mathcal{S}^i_y$, $\mathcal{S}^i_{\text{coll}}$, and $\mathcal{S}^i_{\text{coll2}}$ denote the corresponding sets for the i-th instantiation.

5.5.2 Improvements of the Nested Iterated Construction

The NI-construction [AB99] transforms any FIL-MAC G into a VIL-MAC

$$\mathrm{NI}^{\mathsf{G},\mathsf{G}} \stackrel{\text{def}}{=} \{\mathrm{NI}^{g_{k_1}, g_{k_2}} : \{0,1\}^{\leq 2^b} \to \{0,1\}^\ell\}_{k_1, k_2 \in \{0,1\}^\kappa}$$

[53] We assume w.l.o.g. that A's forgery message m is distinct from its oracle queries.

as illustrated in Figure 5.7. More precisely, a message $m \in \{0,1\}^{\leq 2^b}$ of length $n \stackrel{\text{def}}{=} |m|$, $\mathrm{NI}^{g_{k_1},g_{k_2}}(m)$ is computed as follows. First break m into $t-1 = \lceil n/b \rceil$ blocks $\{m_i\}_{i=1}^{t-1}$ of length b, where m_{t-1} is padded with zeroes if necessary, and then set $m_t := \langle n \rangle_b$. Finally, let

$$\mathrm{NI}^{g_{k_1},g_{k_2}}(m) \stackrel{\text{def}}{=} g_{k_2}\left(\mathrm{I}_{0^\ell}^{g_{k_1}}(m_1\| \cdots \|m_{t-1})\|m_t\right). \tag{5.4}$$

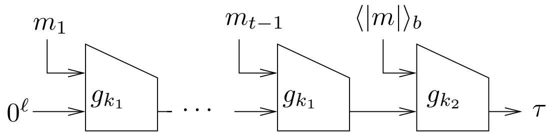


Figure 5.7: The nested iterated (NI) construction

The application function is $\phi(n) = \lceil \frac{n}{b} + 1 \rceil$ and on-line processing is possible. Note that the message space is VIL, due to the encoding of the length of the messages in the last block (recall that $m_t := \langle n \rangle_b$). The following theorem is from [AB99]. We give a proof for completeness.

Theorem 12. *For any t, q, $\mu \geq 1$*

$$\mathbf{InSec}_{t,q,\mu}^{\mathsf{UF\text{-}CPA}}(\mathrm{NI}^{\mathsf{G}}) \leq \left(\frac{1}{2}\cdot q'^2 - \frac{1}{2}\cdot q' + 1\right)\cdot \mathbf{InSec}_{t',q'}^{\mathsf{UF\text{-}CPA}}(\mathsf{G}),$$

where $q' = \left\lfloor \frac{\mu}{b} \right\rfloor + (q+1)$, $t' = t + \mathsf{poly}(q,\mu,|\mathsf{G}|)$ for some polynomial poly that accounts for the overhead implied by the reduction we make, and $|\mathsf{G}|$ denotes the size of a circuit for computing any instantiation of G.

Proof. We prove that $\mathcal{S}_{\text{coll}}^1$ is complete for $\mathrm{NI}^{\langle\cdot,\cdot\rangle}$, the rest follows from Proposition 8. We assume that the last entry $\mathbf{z}$ in $\mathbf{Z}^2$ is not new, and show that this implies a non-trivial collision in g_{k_1}. Let m denote the forgery message of A, i.e., the last query to $\mathrm{NI}^{g_{k_1},g_{k_2}}(\cdot)$ in $\mathsf{A}\blacklozenge\mathrm{NI}^{g_{k_1},g_{k_2}}$. Now, as the last entry $\mathbf{z}$ in $\mathbf{Z}^2$ is not new, there is a query m' (issued by A and different from m) which has the same input to g_{k_2} in the computation of $\mathrm{NI}_{0^\ell}^{g_{k_1},g_{k_2}}(m')$ as in the computation of $\mathrm{NI}_{0^\ell}^{g_{k_1},g_{k_2}}(m)$. As part of the input to g_{k_2} encodes the length of the input message, it follows that $|m| = |m'|$. Now break m and m' into b-bit blocks $m_1,\ldots,m_{t-1}$ and $m'_1,\ldots,m'_{t-1}$, where the last blocks are padded with zeroes if necessary, and let $m_t = m'_t = \langle |m| \rangle_b$. It follows that

$$\overbrace{g_{k_2}\Big(\underbrace{\mathrm{I}_{0^\ell}^{g_{k_1}}(m_1\|\cdots\|m_{t-1})\big\|\langle|m|\rangle_b}_{z}\Big)}^{\mathrm{NI}_{0^\ell}^{g_{k_1},g_{k_2}}(m)\,=} = \overbrace{g_{k_2}\Big(\underbrace{\mathrm{I}_{0^\ell}^{g_{k_1}}(m'_1\|\cdots\|m'_{t-1})\big\|\langle|m'|\rangle_b}_{z'}\Big)}^{\mathrm{NI}_{0^\ell}^{g_{k_1},g_{k_2}}(m')\,=},$$

for which $z = z'$ and $m_1\|\cdots\|m_{t-1} \neq m'_1\|\cdots\|m'_{t-1}$. And hence,

$$\mathrm{I}_{0^\ell}^{g_{k_1}}(m_1\|\cdots\|m_{t-1}) = \mathrm{I}_{0^\ell}^{g_{k_1}}(m'_1\|\cdots\|m'_{t-1}).$$

Let $z_1, \ldots, z_{t-1}$ and $z'_1, \ldots, z'_{t-1}$ denote the inputs to g_{k_1} as they occur in the computation of $\mathrm{I}_{0^\ell}^{g_{k_1}}(m_1\|\cdots\|m_{t-1})$ and $\mathrm{I}_{0^\ell}^{g_{k_1}}(m'_1\|\cdots\|m'_{t-1})$, respectively. As $m_1\|\cdots\|m_{t-1} \neq m'_1\|\cdots\|m'_{t-1}$, there is an index $i > 0$ such that $z_{t-i} \neq z'_{t-i}$ and $g_{k_1}(z_{t-i}) = g_{k_1}(z'_{t-i})$, i.e., a non-trivial collision in $g_{k_1}(\cdot)$.

Now, as $\Phi^2(q{+}1, \mu) \leq \Phi^1(q{+}1, \mu)$ and $n_1, \ldots, n_{q+1} \in \mathbb{N}_0$ exist such that

$$\Phi^1(q{+}1, \mu) = \sum_{i=1}^{q+1} \phi^1(n_i) \leq \left\lfloor \sum_{i=1}^{q+1} \frac{n_i{+}b{-}1}{b} \right\rfloor \leq \left\lfloor \frac{\mu}{b}{+}q{+}1 \right\rfloor =: q',$$

we get that $\#\mathcal{S}^1_{\mathrm{coll}} + 1 \leq q'^2/2 - q'/2 + 1$ by (5.2). □

Finally, let us briefly point out three natural improvements of NI:

1. By replacing $y_0 := 0^\ell$ with an ℓ-bit message block, the waste decreases by ℓ bits, the security reduction gets slightly tighter, and the on-line property is preserved. The security proof for this modified construction is identical to that of the NI-construction.

2. The block $m_t := \langle n \rangle_b$, encoding the message length, is superfluous. It can be replaced by a message block with appropriate padding. This decreases the waste of the construction, improves the tightness of the reduction, lifts the message space to AIL, and preserves the on-line property. To be precise, the tag τ of a message m is defined by first parsing m into a sequence of b-bit blocks $m_1, \ldots, m_t$ such that $m_1\|\ldots\|m_t = m\|10^\nu$ with $\nu \in \{0, \ldots, b-1\}$ and then computing
$$\tau = g_{k_2}\big(\mathrm{I}_{0^\ell}^{g_{k_1}}(m_1\|\cdots\|m_{t-1})\|m_t\big)\,.$$
It is easy to verify that $\mathcal{S}^1_{\mathrm{coll}} \cup \mathcal{S}^1_{0^\ell}$ is complete for this construction.

3. If the block encoding the message length is used as the first block instead of the last, the two keys can actually be replaced by a single key. As shown in Section 5.3.2, this works for any other prefix-free encoding (in particular the ones preserving the on-line property).

Bibliography

[**Note:** *the numbers after each item denote pages, on which the item was referenced.*]

[AB99] Jee Hea An and Mihir Bellare. Constructing VIL-MACs from FIL-MACs: Message authentication under weakened assumptions. In *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 252–269. Springer-Verlag, 199. 6, 57, 77, 78, 79

[ARV99] William Aiello, Sivaramakrishnan Rajagopalan, and Ramarathnam Venkatesan. High-speed pseudorandom number generation with small memory. In *Fast Software Encryption*, volume 1636 of *Lecture Notes in Computer Science*, pages 290–304. Springer-Verlag, 1999. 43

[BDJR97] Mihir Bellare, Anand Desai, Eron Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *Proceedings of the 38th Symposium on Foundations of Computer Science (FOCS)*, pages 394–403. IEEE, 1997. 18, 19

[BDZ03] Feng Bao, Robert H. Deng, and Huafei Zhu. Variations of Diffie-Hellman problem. In *Information and Communications Security – ICICS '03*, volume 2836 of *Lecture Notes in Computer Science*, pages 301–312. Springer-Verlag, 2003. 15, 52

[BGR95] Mihir Bellare, Roch Guérin, and Phillip Rogaway. XOR MACs: New methods for message authentication using finite pseudorandom functions. In *Advances in Cryptology – CRYPTO '95*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 1995. 6

[BHK+99] John Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway. Umac: Fast and secure message authentication. In *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 313–328. Springer-Verlag, 1999. 54

[BKR00] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences (JCSS)*, 61(3):362–399, 2000. 6, 57

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *Advances in Cryptology – ASIACRYPT '00*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer-Verlag, 2000. 18, 20, 46, 53, 105

[Bon98] Dan Boneh. The decision Diffie-Hellman problem. In *Third Algorithmic Number Theory Symposium*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer-Verlag, 1998. 15

[BR97] Mihir Bellare and Phillip Rogaway. Collision-resistant hashing: Towards making UOWHFs practical. In *Advances in Cryptology – CRYPTO '97*, volume 1294 of *LNCS*, pages 470–484. Springer-Verlag, 1997. 6

[BR06] Mihir Bellare and Thomas Ristenpart. Multi-property-preserving hash domain extension and the EMD transform. In *Advances in Cryptology – ASIACRYPT '06*, volume 4284 of *Lecture Notes in Computer Science*, pages 299–314. Springer-Verlag, 2006. 6, 67

[CDMP05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård revisited : How to construct a hash function. In *Advances in Cryptology – CRYPTO '05*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448. Springer-Verlag, 2005. 6

[Dam89] Ivan Damgård. A design principle for hash functions. In *Advances in Cryptology – CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 110–132. Springer-Verlag, 1989. 6, 65

[Dam04] Ivan Damgård. Discrete log based cryptosystems, 2004. Manuscript, `www.daimi.au.dk/ivan/DL.pdf`. 31

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976. 2, 14

[DN02] Ivan Damgård and Jesper B. Nielsen. Expanding pseudorandom functions; or: From known-plaintext security to chosen-plaintext security. In *Advances in Cryptology – CRYPTO '02*, volume 2442 of *Lecture Notes in Computer Science*, pages 449–464. Springer-Verlag, 2002. 5, 43, 44, 46, 47, 51, 53, 54

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986. 5, 25, 43, 45, 48

[Gol04] Oded Goldreich. *Foundations of Cryptography – Volume II – Basic Applications*. Cambridge University Press, 2004. 5

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *Siam Journal on Computation*, 28(4):1364–1396, 1999. 5, 25, 43

[KA98] Stephen Kent and Ran Atkinson. IP encapsulating security payload (ESP), November 1998. Request for Comments 2406. 46, 53

[Kel06] Marcel Keller. Constructing weak pseudorandom functions with prescribed structure. Semester Thesis, ETH Zurich, 2006. 52

[KY00] Jonathan Katz and Moti Yung. Complete characterization of security notions for probabilistic private-key encryption. In *Proceedings of the 32nd ACM Annual Symposium on Theory of Computing (STOC)*, pages 245–254, 2000. 18, 19

[LR86] Michael Luby and Charles Rackoff. Pseudo-random permutation generators and cryptographic composition. In *Proceedings of the 18th ACM Symposium on the Theory of Computing (STOC)*, pages 356–363, 1986. 3, 21, 23, 26

[Luc96] Stefan Lucks. Faster Luby-Rackoff ciphers. In *Fast Software Encryption*, volume 3557 of *Lecture Notes in Computer Science*, pages 189–203. Springer-Verlag, 1996. 22, 23, 26

[Mau93] Ueli Maurer. Secret key agreement by public discussion. *IEEE Transaction on Information Theory*, 39(3):733–742, 1993. 1, 2

[Mau02] Ueli Maurer. Indistinguishability of random systems. In *Advances in Cryptology – EUROCRYPT '02*, volume 2332 of *Lecture Notes in Computer Science*, pages 110–132. Springer-Verlag, 2002. 4, 6, 9, 23, 26, 89, 90, 99

[Mer78] Ralph Merkle. Secure communications over insecure channels. *Communications of the ACM*, 21(4):294–299, 1978. 2

[Mer90] Ralph Merkle. A certified digital signature. In *Advances in Cryptology – EUROCRYPT '89*, volume 435 of *Lecture Notes in Computer Science*, pages 218–232. Springer-Verlag, 1990. 6, 65

[MOPS06a] Ueli Maurer, Yvonne Anne Oswald, Krzysztof Pietrzak, and Johan Sjödin. Luby-Rackoff ciphers from weak round functions? In *Advances in Cryptology – EUROCRYPT '06*, volume 4004 of *Lecture Notes in Computer Science*, pages 391–408. Springer-Verlag, 2006. Proceedings version of [MOPS06b]. 4, 21, 43, 84

[MOPS06b] Ueli Maurer, Yvonne Anne Oswald, Krzysztof Pietrzak, and Johan Sjödin. Luby-Rackoff ciphers from weak round functions? Cryptology ePrint Archive, Report 2006/213, `http://eprint.iacr.org/2006`, 2006. This is the full version of [MOPS06a]. 4, 21, 30, 41, 43, 84

[MP04] Ueli Maurer and Krzysztof Pietrzak. Composition of random systems: When two weak make one strong. In *Theory of Cryptography – TCC '04*, volume 2951 of *Lecture Notes in Computer Science*, pages 410–427. Springer-Verlag, 2004. 4, 24, 90

[MPR06] Ueli Maurer, Krzysztof Pietrzak, and Renato Renner. Indistinguishability amplification, 2006. Manuscript. 4, 24, 30, 41, 89, 90

[MS05a] Ueli Maurer and Johan Sjödin. Domain expansion of MACs: Alternative uses of the FIL-MAC. In *Cryptography and Coding 2005*, volume 3796 of *Lecture Notes in Computer Science*, pages 168–185. Springer-Verlag, 2005. 6, 57

[MS05b] Ueli Maurer and Johan Sjödin. Single-key AIL-MACs from any FIL-MAC. In *Automata, Languages and Programming – ICALP '05*, volume 3580 of *Lecture Notes in Computer Science*, pages 472–484. Springer-Verlag, 2005. 6, 57

[MS07] Ueli Maurer and Johan Sjödin. A fast and key-efficient reduction of chosen-ciphertext to known-plaintext security. In *Advances in Cryptology – EUROCRYPT '07*, volume 4515 of *Lecture Notes in Computer Science*, pages 498–516. Springer-Verlag, 2007. 5, 43

[MT05] Kazuhiko Minematsu and Yukiyasu Tsunoo. Expanding weak PRF with small key size. In *Information Security and Cryptology - ICISC '05*, volume 3935 of *Lecture Notes in Computer Science*, pages 284–298. Springer-Verlag, 2005. 4, 22, 23, 27, 45, 49, 51

[MvOV97] Alfred Menezes, Paul van Oorschot, and Scott Vanstone. *Handbook of applied cryptography*. CRC Press, 1997. 3, 63

[Mye04] Steven Myers. Black-box composition does not imply adaptive security. In *Advances in Cryptology – EUROCRYPT '04*, volume 3027 of *Lecture Notes in Computer Science*, pages 189–206. Springer-Verlag, 2004. 25

[NPR99] Moni Naor, Benny Pinkas, and Omer Reingold. Distributed pseudo-random functions and KDCs. In *Advances in Cryptology – EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 327–346. Springer-Verlag, 1999. 43, 45

[NR98] Moni Naor and Omer Reingold. From unpredictability to indistinguishability: A simple construction of pseudorandom functions from MACs. In *Advances in Cryptology – CRYPTO '98*, Lecture Notes in Computer Science, pages 267–282. Springer-Verlag, 1998. 5, 43, 44, 45, 46, 57

[NR99a] Moni Naor and Omer Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. *Journal of Cryptology*, 12(1):29–66, 1999. 22, 23, 26

[NR99b] Moni Naor and Omer Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. *Journal of Computer and System Sciences (JCSS)*, 58(2):336–375, 1999. 5, 43, 45

[NR02] Moni Naor and Omer Reingold. Constructing pseudo-random permutations with a prescribed structure. *Journal of Cryptology*, 15(2):97–102, 2002. 25, 29

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *Journal of the ACM*, 51(2):231–262, 2004. 43, 45, 47

[Pat04] Jacques Patarin. Security of random feistel schemes with 5 or more rounds. In *Advances in Cryptology – CRYPTO '04*, volume 3152 of *Lecture Notes in Computer Science*, pages 106–122. Springer-Verlag, 2004. 26

[Pie90] Josef Pieprzyk. How to construct pseudorandom permutations from single pseudorandom functions. In *Advances in Cryptology – EUROCRYPT '90*, volume 537 of *Lecture Notes in Computer Science*, pages 140–150. Springer-Verlag, 1990. 22

[Pie05] Krzysztof Pietrzak. Composition does not imply adaptive security. In *Advances in Cryptology – CRYPTO '05*, volume 3621 of *Lecture Notes in Computer Science*, pages 55–65. Springer-Verlag, 2005. 4, 24, 25

[Pie06] Krzysztof Pietrzak. *Indistinguishability and Composition of Random Systems*. PhD thesis, ETH Zürich, 2006. ISBN 3-86628-063-7. 24, 89, 91

[Ple05] Patrick Pletscher. Adaptive security of composition. Semester Thesis, ETH Zurich, 2005. 24, 31

[PR00] Erez Petrank and Charles Rackoff. CBC MAC for real-time data sources. *Journal of Cryptology*, 13(3):315–338, 2000. 6

[PS06] Krzysztof Pietrzak and Johan Sjödin. Weak pseudorandom functions in minicrypt, November 2006. Manuscript. 13, 43, 51

[PS07] Krzysztof Pietrzak and Johan Sjödin. Domain extension for weak PRFs; The good, the bad, and the ugly. In *Advances in Cryptology – EUROCRYPT '07*, volume 4515 of *Lecture Notes in Computer Science*, pages 517–533. Springer-Verlag, 2007. 43, 51

[RR00] Zulfikar Ramzan and Leonid Reyzin. On the round security of symmetric-key cryptographic primitives. In *Advances in Cryptology – CRYPTO '00*, volume 1880 of *Lecture Notes in Computer Science*, pages 376–393. Springer-Verlag, 2000. 22

[RSA78] Ronald Rivest, Adi Shamir, and Leonard Adelman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. 2

[Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28:656–715, 1949. 1, 2

[Sho96] Victor Shoup. On fast and provably secure message authentication based on universal hashing. In *Advances in Cryptology – CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 313–328. Springer-Verlag, 1996. 54

[Sho00] Victor Shoup. A composition theorem for universal one-way hash functions. In *Advances in Cryptology – EUROCRYPT '00*, volume 1807 of *Lecture Notes in Computer Science*, pages 445–452. Springer-Verlag, 2000. 6

[Sho04] Victor Shoup. Sequences of games: A tool for taming complexity in security proofs. Cryptology ePrint Archive: Report 2004/332, `http://eprint.iacr.org/2006`, 2004. 30

[Sti92] Douglas R. Stinson. Universal hashing and authentication codes. In *Advances in Cryptology – CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 74–85. Springer-Verlag, 1992. 54

[WC81] Mark N. Wegman and J. Lawrence Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences (JCSS)*, 22:265–279, 1981. 46, 54

Appendix A

Deferred Proofs

In this appendix, we give proofs of the propositions, claims, and theorems of Chapter 3 and 4. First, however, we recall some tools from the indistinguishability framework introduced by Maurer in [Mau02] (see also [Pie06] and the yet unpublished [MPR06]).

A.1 Tools for Random Systems

Let us define the concept of *monotone conditions* for random systems and show how they can be used to prove bounds on the indistinguishability of random systems.

A monotone condition $\mathcal{A}$ for a $(\mathcal{X}, \mathcal{Y})$-random system $\mathbf{F}$ is an event sequence $A_1, A_2, \ldots$, where $A_i \in \{a_i, \overline{a}_i\}$. Here a_i ($\overline{a}_i$) denotes the event that the condition is satisfied (failed) after the i-th query to $\mathbf{F}$ has been processed. Monotone means that if the condition failed, it will never hold again (i.e., $\overline{a}_i \Rightarrow \overline{a}_{i+1}$). So the event $\overline{a}_i$ immediately implies $\overline{a}_j$ for all $j > i$.

We denote a $(\mathcal{X}, \mathcal{Y})$-*random system* $\mathbf{F}$ with a monotone condition $\mathcal{A}$ as $\mathbf{F}^{\mathcal{A}}$, and model the monotone condition by an extra binary output of the system A_i (where $A_i = 0$ indicates the event a_i and $A_i = 1$ the event $\overline{a}_i$). As this output A_i is never used as an input to a distinguisher or another system, it is convenient to think of $\mathbf{F}^{\mathcal{A}}$ as of $\mathbf{F}$ with a lamp. This lamp is initially off ($A_0 = 0$), but may turn on at some point to indicate that the condition failed, i.e., the lamp is on after the i-th query if and only if $A_i = 1$.

Definition 25. *A* $(\mathcal{X},\mathcal{Y})$-random system $\mathbf{F}$ *with a monotone condition* $\mathcal{A}$*, denoted* $\mathbf{F}^{\mathcal{A}}$*, is the same random system but with an additional monotone binary variable sequence* $A_1, A_2, \ldots$ *defined on it. The value of* $A_i \in \{0,1\}$ *is determined after the* i*-th query. Monotone means* $\mathbf{F}^{\mathcal{A}}$ *is given by the infinite sequence of conditional probability distributions* $\mathsf{P}^{\mathbf{F}}_{A_iY_i|X^iY^{i-1}A_{i-1}}$ *for* $i \geq 1$ *(or equivalently by* $\mathsf{P}^{\mathbf{F}}_{A_iY^i|X^i}$ *for* $i \geq 1$*).* $A_i = 0$ *means that* $\mathcal{A}$ *holds after the* i*-th query, this event is denoted by* a_i*, the event* $A_i = 1$ *is denoted with* $\overline{a}_i$*.*

Let $\mathbf{F}$ and $\mathbf{G}$ be random systems and $\mathcal{A}$ be a condition defined for $\mathbf{F}$. We define three relations for random systems with conditions

$$\mathbf{F}^{\mathcal{A}} \equiv \mathbf{G}^{\mathcal{B}} \iff \forall i \geq 1 : \mathsf{P}^{\mathbf{F}}_{a_iY^i|X^i} = \mathsf{P}^{\mathbf{G}}_{b_iY^i|X^i}$$

$$\mathbf{F}|\mathcal{A} \equiv \mathbf{G} \iff \forall i \geq 1 : \mathsf{P}^{\mathbf{F}}_{Y^i|X^ia_i} = \mathsf{P}^{\mathbf{G}}_{Y^i|X^i}$$

$$\mathbf{F}^{\mathcal{A}} \preceq \mathbf{G} \iff \forall i \geq 1 : \mathsf{P}^{\mathbf{F}}_{a_iY^i|X^i} \leq \mathsf{P}^{\mathbf{G}}_{Y^i|X^i}.$$

Note that $\mathbf{F}^{\mathcal{A}} \preceq \mathbf{G}$ is implied by $\mathbf{F}|\mathcal{A} \equiv \mathbf{G}$ and $\mathbf{F}^{\mathcal{A}} \equiv \mathbf{G}^{\mathcal{B}}$, respectively. The following proposition states that if $\mathbf{F}^{\mathcal{A}} \preceq \mathbf{G}$, then distinguishing $\mathbf{F}$ from $\mathbf{G}$ is at least as hard as making the condition fail. To be more precise:

Definition 26. *For a random system* $\mathbf{F}$ *with a condition* $\mathcal{A}$*, we define the probability of the event* $\overline{a}_q$ *in the random experiment where* $\mathbf{D}$ *is querying* $\mathbf{F}$ *as*

$$\nu^{\mathbf{D}}(\mathbf{F}^{\mathcal{A}}, \overline{a}_q) \stackrel{\text{def}}{=} \mathsf{P}^{\mathbf{D}\diamond\mathbf{F}}_{\overline{a}_q}. \tag{A.1}$$

The probability of the best ATK*-distinguisher to provoke* $\overline{a}_q$ *is*

$$\nu^{\mathsf{ATK}}(\mathbf{F}^{\mathcal{A}}, \overline{a}_q) \stackrel{\text{def}}{=} \max_{\mathsf{ATK}-distinguisher\ \mathbf{D}} \mathsf{P}^{\mathbf{D}\diamond\mathbf{F}}_{\overline{a}_q}. \tag{A.2}$$

Proposition 9. *If* $\mathbf{F}^{\mathcal{A}} \preceq \mathbf{G}$ *(which is implied by* $\mathbf{F}|\mathcal{A} \equiv \mathbf{G}$ *and* $\mathbf{F}^{\mathcal{A}} \equiv \mathbf{G}^{\mathcal{B}}$*, respectively) then for any distinguisher* $\mathbf{D}$ *we have*

$$\Delta^{\mathbf{D}}_q(\mathbf{F}, \mathbf{G}) \leq \nu^{\mathbf{D}}(\mathbf{F}^{\mathcal{A}}, \overline{a}_q).$$

If $\mathbf{F}^{\mathcal{A}} \equiv \mathbf{G}^{\mathcal{B}}$ *it holds that*

$$\nu^{\mathbf{D}}(\mathbf{F}^{\mathcal{A}}, \overline{a}_q) = \nu^{\mathbf{D}}(\mathbf{G}^{\mathcal{B}}, \overline{b}_q).$$

By this proposition we can bound $\Delta^{\mathsf{ATK}}_q(\mathbf{F}, \mathbf{G})$ by first finding a condition $\mathcal{A}$ for $\mathbf{F}$ which satisfies $\mathbf{F}^{\mathcal{A}} \preceq \mathbf{G}$ and then trying to prove an upper bound on $\nu^{\mathsf{ATK}}(\mathbf{F}^{\mathcal{A}}, \overline{a}_q)$. The proof of this proposition was given in the original paper [Mau02]. The following proposition is from (the yet unpublished) [MPR06], a weaker version can be found in [MP04].

Proposition 10. *For any random systems* $\mathbf{F}$ *and* $\mathbf{G}$*, there exist conditions* $\mathcal{A}$ *and* $\mathcal{B}$ *such that*

$$\mathbf{F}^{\mathcal{A}} \equiv \mathbf{G}^{\mathcal{B}}$$

and for any distinguisher $\mathbf{D}$ *and any* $q > 0$

$$\Delta_q^{\mathbf{D}}(\mathbf{F}, \mathbf{G}) = \nu^{\mathbf{D}}(\mathbf{F}^{\mathcal{A}}, \overline{a}_q) = \nu^{\mathbf{D}}(\mathbf{G}^{\mathcal{B}}, \overline{b}_q).$$

The next proposition states that if $\mathbf{F}|\mathcal{A}$ is itself a random system, then adaptivity is of no use when one want to make $\mathcal{A}$ fail. We will use this proposition many times as dealing with non-adaptive distinguishers is usually much easier than to handle adaptive ones. A proof of this proposition appeared in [Pie06].

Proposition 11. *For any* $i \in \mathbb{N}$*, if for a random system* $\mathbf{F}$ *with a condition* $\mathcal{A}$ *there exists a random system* $\mathbf{G}$ *such that* $\mathbf{F}|\mathcal{A} \equiv \mathbf{G}$*, i.e., for all* $i \geq 1$

$$\mathsf{P}^{\mathbf{F}}_{Y^i|X^i a_i} \equiv \mathsf{P}^{\mathbf{G}}_{Y^i|X^i}, \tag{A.3}$$

then adaptivity does not help in provoking $\overline{a}_i$*, i.e.*

$$\nu^{\mathsf{CPA}}(\mathbf{F}^{\mathcal{A}}, \overline{a}_i) = \nu^{\mathsf{nCPA}}(\mathbf{F}^{\mathcal{A}}, \overline{a}_i). \tag{A.4}$$

We will frequently make use of a random system called beacon, denoted by $\mathbf{B}$.

Definition 27 (Beacon)**.** *An* $(\mathcal{X} \to \mathcal{Y})$-beacon $\mathbf{B}$ *is a random system for which the outputs* $Y_1, Y_2, \ldots$ *are independent and uniformly distributed over the range* $\mathcal{Y}$ *(and in particular independent of the inputs).*

Note that $\mathbf{R}|\mathcal{A} \equiv \mathbf{B}$, if $\mathcal{A}$ denotes the condition that the inputs to the URF $\mathbf{R}$ are distinct. Hence, by Proposition 9 it follows that

$$\begin{aligned}
\Delta_q^{\mathsf{KPA}}(\mathbf{F}, \mathbf{B}) - \Delta_q^{\mathsf{KPA}}(\mathbf{F}, \mathbf{R}) &\overset{tri.\,ineq.}{\leq} \Delta_q^{\mathsf{KPA}}(\mathbf{R}, \mathbf{B}) \\
&\overset{Prop.\,9}{\leq} \nu^{\mathsf{KPA}}(\mathbf{R}, \overline{a}_q) \\
&\overset{b\text{-}bound}{\leq} \frac{q^2}{2^{n+1}}.
\end{aligned} \tag{A.5}$$

Consider the random system $\mathcal{E}(\mathbf{F})$, defined by some random system $\mathcal{E}(\cdot)$ having an interface for interaction with some compatible random system $\mathbf{F}$.[54] In the sequel we will frequently make use the following two arguments:

[54]Note that a distinguisher for $\mathbf{F}$ is an example of such a system $\mathcal{E}(\cdot)$.

(i) Consider a monotone condition $\mathcal{A}$, defined on $\mathcal{E}(\cdot)$. Then it follows (which we show next) that

$$\nu^{\mathsf{ATK}}(\mathcal{E}^{\mathcal{A}}(\mathbf{F}), \overline{a}_q) - \nu^{\mathsf{ATK}}(\mathcal{E}^{\mathcal{A}}(\mathbf{G}), \overline{a}_q) \leq \Delta_q^{\mathsf{ATK}}(\mathcal{E}^{\mathcal{A}}(\mathbf{G}), \mathcal{E}^{\mathcal{A}}(\mathbf{F})). \quad \text{(A.6)}$$

Consider the ATK-distinguisher $\mathbf{D}$ for which it holds that

$$\nu^{\mathbf{D}}(\mathcal{E}^{\mathcal{A}}(\mathbf{F}), \overline{a}_q) = \nu^{\mathsf{ATK}}(\mathcal{E}^{\mathcal{A}}(\mathbf{F}), \overline{a}_q),$$

and the distinguisher $\mathbf{D}'$ that simply runs $\mathbf{D}$ and outputs 1 if $\overline{a}_q$ is provoked and otherwise 0. Clearly, $\mathbf{D}'$ distinguishes $\mathcal{E}^{\mathcal{A}}(\mathbf{G})$ from $\mathcal{E}^{\mathcal{A}}(\mathbf{F})$ with advantage $\nu^{\mathbf{D}}(\mathcal{E}^{\mathcal{A}}(\mathbf{F}), \overline{a}_q) - \nu^{\mathbf{D}}(\mathcal{E}^{\mathcal{A}}(\mathbf{G}), \overline{a}_q)$, from which (A.6) follows.

(ii) Suppose there is an ATK-distinguisher $\mathbf{D}$ for $\mathcal{E}(\mathbf{F})$ and $\mathcal{E}(\mathbf{G})$, from which we can construct a distinguisher $\mathbf{D}\diamond\mathcal{E}(\cdot)$ for $\mathbf{F}$ and $\mathbf{G}$. For $\mathsf{ATK}' = \{\mathbf{D}\diamond\mathcal{E}(\cdot) | \mathbf{D} \in \mathsf{ATK}\}$ and $k' = c \cdot q$, where c is the number of invocations that $\mathcal{E}(\mathbf{E})$ makes to its component $\mathbf{E}$ on every invocation, it follows that

$$\begin{aligned} \Delta_q^{\mathsf{ATK}}(\mathcal{E}(\mathbf{F}), \mathcal{E}(\mathbf{G})) &\leq \Delta_{q'}^{\mathsf{ATK}'}(\mathbf{F}, \mathbf{G}) && \text{(A.7)} \\ \nu^{\mathsf{ATK}}(\mathcal{E}(\mathbf{F}^{\mathcal{A}}), \overline{a}_q) &\leq \nu^{\mathsf{ATK}'}(\mathbf{F}^{\mathcal{A}}, \overline{a}_{q'}). && \text{(A.8)} \end{aligned}$$

A.2 Proofs for Chapter 3

Consider the random system $\psi_{2n}[\mathbf{FG}]$ for some random systems $\mathbf{F}$ and $\mathbf{G}$. Throughout this section, we let $\psi_{2n}[\mathbf{F}^{\mathcal{A}^1}\mathbf{G}^{\mathcal{A}^2}]$ (for some monotone conditions $\mathcal{A}^1$ and $\mathcal{A}^2$, defined on $\mathbf{F}$ and $\mathbf{G}$, respectively) denote the random system $\psi_{2n}[\mathbf{FG}]^{\mathcal{A}}$ with the condition $\mathcal{A}$ defined as $\mathcal{A} = \mathcal{A}^1 \wedge \mathcal{A}^2$. This naturally generalizes to Feistel-networks with more than two rounds.

A.2.1 The Two and Three Round Feistel-Network

In this section, we prove Propositions 2 - 5 and the following fact

$$\Delta_q^{\mathsf{nCCA}}(\psi_{2n}[\mathbf{RRR}], \mathbf{P}) \leq 2 \cdot \frac{q^2}{2^{n+1}}.$$

We start with the latter. After i queries by a distinguisher, let $Q_i^{\rightarrow}$ and $O_i^{\rightarrow}$ denote the set of queries and outputs (of the system) in the forward direction, respectively. Similarly, let $Q_i^{\leftarrow}$ and $O_i^{\leftarrow}$ denote the queries and outputs in the reverse direction, respectively. Furthermore, let $c_i^{\leftrightarrow}$ denote the event that the input to the second round function are all distinct after i queries, let $c_i^{\rightarrow}$ denote the event that there are distinct $x, x' \in O_i^{\rightarrow}$ or $(x, x') \in O_i^{\rightarrow} \times Q_i^{\leftarrow}$ such that ${}_Lx = {}_Lx'$, and let $c_i^{\leftarrow}$ denote the event that there are distinct $x, x' \in O_i^{\leftarrow}$ or $(x, x') \in O_i^{\leftarrow} \times Q_i^{\rightarrow}$ such that ${}_Lx = {}_Lx'$. As

$$\psi_{2n}[\mathbf{R}^{\mathcal{C}^{\leftarrow}}\mathbf{R}^{\mathcal{C}^{\leftrightarrow}}\mathbf{R}^{\mathcal{C}^{\rightarrow}}] \equiv \psi_{2n}[\mathbf{R}^{\mathcal{C}^{\leftarrow}}\mathbf{B}^{\mathcal{C}^{\leftrightarrow}}\mathbf{R}^{\mathcal{C}^{\rightarrow}}] \preceq \mathbf{P}$$

and $|Q^{\leftarrow}| + |Q^{\rightarrow}| \leq q$, we get

$$\begin{aligned}
&\Delta_q^{\mathsf{nCCA}}(\psi_{2n}[\mathbf{RRR}], \mathbf{P}) \\
\overset{Prop.\ 9}{\leq}\quad & \nu^{\mathsf{nCCA}}(\psi_{2n}[\mathbf{R}^{\mathcal{C}^{\leftarrow}}\mathbf{B}^{\mathcal{C}^{\leftrightarrow}}\mathbf{R}^{\mathcal{C}^{\rightarrow}}], \overline{c_q^{\leftarrow}} \vee \overline{c_q^{\leftrightarrow}} \vee \overline{c_q^{\rightarrow}}) \\
\leq\quad & \nu^{\mathsf{nCCA}}(\psi_{2n}[\mathbf{R}^{\mathcal{C}^{\leftarrow}}\mathbf{BR}], \overline{c_q^{\leftarrow}}) + \\
& \nu^{\mathsf{nCCA}}(\psi_{2n}[\mathbf{RBR}^{\mathcal{C}^{\rightarrow}}], \overline{c_q^{\rightarrow}}) + \\
& \nu^{\mathsf{nCCA}}(\psi_{2n}[\mathbf{RB}^{\mathcal{C}^{\leftrightarrow}}\mathbf{R}], \overline{c_q^{\leftrightarrow}}) \\
\overset{union\ bound}{\leq}\quad & \frac{|Q_q^{\leftarrow}|^2}{2^{n+1}} + |Q_q^{\leftarrow}| \cdot \frac{|O_q^{\rightarrow}|}{2^{n+1}} + \\
& \frac{|Q_q^{\rightarrow}|^2}{2^{n+1}} + |Q_q^{\rightarrow}| \cdot \frac{|O_q^{\leftarrow}|}{2^{n+1}} + \\
& \frac{(|Q_q^{\rightarrow}| + |Q_q^{\leftarrow}|)^2}{2^{n+1}} \\
\leq\quad & 2 \cdot \frac{q^2}{2^{n+1}}.
\end{aligned}$$

Proof (of Proposition 2). Without loss of generality (since we are dealing with stateless systems) we assume that the distinguisher only makes distinct queries. Let $\mathcal{C}$ denote a monotone condition for any function defined by letting c_i denote the event that the first i *inputs* of the function are distinct. Let $\mathcal{A}$ denote a monotone condition for any function defined by letting a_i denote the event that the first i *outputs* of the function are distinct. It follows that

$$\begin{aligned}
\mathbf{B}^{\mathcal{C}} &\equiv \mathbf{R}^{\mathcal{C}} \\
\mathbf{H} \triangleright \psi_{2n}[\mathbf{R}^{\mathcal{C}}\mathbf{G}] &\equiv \mathbf{H} \triangleright \psi_{2n}[\mathbf{B}^{\mathcal{C}}\mathbf{G}] \\
\mathbf{H} \triangleright \psi_{2n}[\mathbf{BB}] &\equiv \mathbf{B} \\
\mathbf{B}|(\mathcal{A} \wedge \mathcal{C}) &\equiv \mathbf{P}.
\end{aligned}$$

For $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{nCPA}, \mathsf{KPA}\}$, we get

$$
\begin{aligned}
& \Delta_q^{\mathsf{ATK}}(\mathbf{H} \triangleright \psi_{2n}[\mathbf{FG}], \mathbf{P}) \\
\overset{tri.\ ineq.}{\leq}\ & \Delta_q^{\mathsf{ATK}}(\mathbf{H} \triangleright \psi_{2n}[\mathbf{FG}], \mathbf{H} \triangleright \psi_{2n}[\mathbf{RG}]) + \\
& \Delta_q^{\mathsf{ATK}}(\mathbf{H} \triangleright \psi_{2n}[\mathbf{RG}], \mathbf{H} \triangleright \psi_{2n}[\mathbf{BG}]) + \\
& \Delta_q^{\mathsf{ATK}}(\mathbf{H} \triangleright \psi_{2n}[\mathbf{BG}], \mathbf{H} \triangleright \psi_{2n}[\mathbf{BB}]) + \\
& \Delta_q^{\mathsf{ATK}}(\mathbf{H} \triangleright \psi_{2n}[\mathbf{BB}], \mathbf{P}) \\
\overset{(ii),\ Prop.\ 9}{\leq}\ & \Delta_q^{\mathsf{ATK}}(\mathbf{F}, \mathbf{R}) + \nu^{\mathsf{ATK}}(\mathbf{H} \triangleright \psi_{2n}[\mathbf{B}^{\mathcal{C}}\mathbf{G}], \overline{c}_q) + \\
& \Delta_q^{\mathsf{KPA}}(\mathbf{G}, \mathbf{B}) + \Delta_q^{\mathsf{ATK}}(\mathbf{B}, \mathbf{P}) \\
\overset{(i),(ii)}{\leq}\ & \Delta_q^{\mathsf{ATK}}(\mathbf{F}, \mathbf{R}) + \nu^{\mathsf{ATK}}(\mathbf{H} \triangleright \psi_{2n}[\mathbf{B}^{\mathcal{C}}\mathbf{B}], \overline{c}_q) + \\
& \Delta_q^{\mathsf{KPA}}(\mathbf{G}, \mathbf{B}) + \Delta_q^{\mathsf{KPA}}(\mathbf{G}, \mathbf{B}) + \Delta_q^{\mathsf{ATK}}(\mathbf{B}, \mathbf{P}) \\
\overset{Prop.\ 11,\ tri.\ ineq.}{\leq}\ & \Delta_q^{\mathsf{ATK}}(\mathbf{F}, \mathbf{R}) + \nu^{\mathsf{nCPA}}(\mathbf{H} \triangleright \psi_{2n}[\mathbf{B}^{\mathcal{C}}\mathbf{B}], \overline{c}_q) + \\
& 2 \cdot \big(\Delta_q^{\mathsf{KPA}}(\mathbf{G}, \mathbf{R}) + \Delta_q^{\mathsf{KPA}}(\mathbf{R}, \mathbf{B})\big) + \Delta_q^{\mathsf{ATK}}(\mathbf{B}, \mathbf{P}) \\
\overset{Prop.\ 9}{\leq}\ & \Delta_q^{\mathsf{ATK}}(\mathbf{F}, \mathbf{R}) + \mathsf{coll}_q({}_L H) + 2 \cdot \Delta_q^{\mathsf{KPA}}(\mathbf{G}, \mathbf{R}) + \\
& 2 \cdot \nu^{\mathsf{KPA}}(\mathbf{R}^{\mathcal{C}}, \overline{c}_q) + \nu^{\mathsf{ATK}}(\mathbf{B}^{\mathcal{A} \wedge \mathcal{C}}, \overline{a}_q \vee \overline{c}_q) \\
\leq\ & \Delta_q^{\mathsf{ATK}}(\mathbf{F}, \mathbf{R}) + \mathsf{coll}_q({}_L H) + 2 \cdot \Delta_q^{\mathsf{KPA}}(\mathbf{G}, \mathbf{R}) + \\
& 2 \cdot \nu^{\mathsf{KPA}}(\mathbf{R}^{\mathcal{C}}, \overline{c}_q) + \nu^{\mathsf{ATK}}(\mathbf{B}^{\mathcal{A}}, \overline{a}_q) + \nu^{\mathsf{ATK}}(\mathbf{B}^{\mathcal{C}}, \overline{c}_q) \\
\overset{b\text{-}bound}{\leq}\ & \Delta_q^{\mathsf{ATK}}(\mathbf{F}, \mathbf{R}) + \mathsf{coll}_q({}_L H) + \\
& 2 \cdot \Delta_q^{\mathsf{KPA}}(\mathbf{G}, \mathbf{R}) + 2 \cdot \frac{q(q-1)}{2^{n+1}} + 2 \cdot \frac{q(q-1)}{2^{2n+1}}.
\end{aligned}
$$

We omit the proof of the analogous statement in the pseudorandom setting, since the corresponding arguments (in the above proof) easily translates. □

Proof (of Proposition 3). Let $\mathcal{B}$ denote the monotone condition defined by letting b_i denote the event that all values at the left half of the inputs and the right half of the outputs are all distinct (up to the i-th query). Using the fact that

$$\psi_{2n}[\mathbf{R}^2]^{\mathcal{B}} \equiv \mathbf{P}^{\mathcal{B}},$$

it follows that

$$
\begin{aligned}
& \Delta_q^{\mathsf{KPA}}(\psi_{2n}[\mathbf{F}^2], \mathbf{P}) \\
\overset{tri.\ ineq.}{\leq} \quad & \Delta_q^{\mathsf{KPA}}(\psi_{2n}[\mathbf{F}^2], \psi_{2n}[\mathbf{R}^2]) + \Delta_q^{\mathsf{KPA}}(\psi_{2n}[\mathbf{R}^2], \mathbf{P}) \\
\overset{Prop.\ 9}{\leq} \quad & \Delta_q^{\mathsf{KPA}}(\psi_{2n}[\mathbf{F}^2], \psi_{2n}[\mathbf{R}^2]) + \nu^{\mathsf{KPA}}(\mathbf{P}^{\mathcal{B}}, \overline{b}_q) \\
\overset{(ii)}{\leq} \quad & \Delta_{2q}^{\mathsf{KPA}}(\mathbf{F}, \mathbf{R}) + \nu^{\mathsf{KPA}}(\mathbf{P}^{\mathcal{B}}, \overline{b}_q) \\
\overset{b\text{-}bound}{\leq} \quad & \Delta_{2q}^{\mathsf{KPA}}(\mathbf{F}, \mathbf{R}) + \frac{(2q)^2}{2^{n+1}}.
\end{aligned}
$$

In the third inequality, we used the fact that a KPA-distinguisher A for $\psi_{2n}[\mathbf{F}^2]$ and $\psi_{2n}[\mathbf{R}^2]$ implies a KPA-distinguisher A′ for $\mathbf{F}$ and $\mathbf{R}$ with the same distinguishing advantage. A′ simply runs A and answers its oracle queries with help of its own oracle. Note that given two random input-output pairs of any function $\mathbf{F}$ one can easily construct a random input output pair of $\psi_{2n}[\mathbf{F}^2]$, and hence A′ needs twice as many oracle queries as A.

We omit the proof of the analogous statement in the pseudorandom setting, since the corresponding arguments (in the above proof) also hold in the pseudorandom setting. □

Proof (of Proposition 4, continued). Since $\mathbf{F}(x) := x \oplus \mathbf{I}(x)$ it follows that $\Delta_q^{\mathsf{nCPA}}(\mathbf{F}, \mathbf{R}) \overset{(ii)}{=} \Delta_q^{\mathsf{nCPA}}(\mathbf{I}, \mathbf{R})$, and hence it remains to show that

$$
\Delta_q^{\mathsf{nCPA}}(\mathbf{I}, \mathbf{R}) \leq \frac{q^2}{2^{n-1}}. \tag{A.9}
$$

Let $\mathcal{B}'$ denote the monotone condition that all outputs are distinct and no input equals a previous or subsequent output, i.e., formally

$$
\overline{b}'_q \iff \exists i, j \leq q,\ i \neq j : [x_i = y_j] \vee [y_i = y_j].
$$

Clearly, $\mathbf{R} \,|\, \mathcal{B}' \equiv \mathbf{I}$ and thus

$$
\Delta_q^{\mathsf{nCPA}}(\mathbf{I}, \mathbf{R}) \overset{Prop.\ 9}{\leq} \nu^{\mathsf{nCPA}}(\mathbf{R}, \overline{b}'_q).
$$

Since we assume (without loss of generality) that the distinguishers only issue distinct queries to $\mathbf{R}$, it follows that both $x_i = y_j$ and $y_i = y_j$ occurs with probability $\frac{1}{2^n}$, respectively. By applying the union, it follows that

$$
\nu^{\mathsf{nCPA}}(\mathbf{R}, \overline{b}'_q) \overset{union\ bound}{\leq} 2 \cdot q(q-1) \cdot \frac{1}{2^n},
$$

which concludes the proof. □

Proof (of Proposition 5, continued). Recall that $\mathbf{F}$ is a uniform random function which ignores the first bit (so the output does not change if one flips the first bit). Let $\mathcal{B}''$ denote the monotone condition, where $\overline{b}''_q$ is the event that there exist two inputs x_i and x_j (with $i < j \leq q$) for which the first bit differs and the latter $n-1$ bits are the same. As $\mathbf{F} \mid \mathcal{B}'' \equiv \mathbf{R}$ we get

$$\Delta_q^{\mathsf{KPA}}(\mathbf{F}, \mathbf{R}) \overset{Prop.\ 9}{\leq} \nu^{\mathsf{KPA}}(\mathbf{R}, \overline{b}''_q) \overset{b\text{-}bound}{\leq} \frac{q^2}{2^{n+1}},$$

which concludes the proof. □

A.2.2 The Four and Five Round Feistel-Network

In this section, we prove Theorem 1 and Theorem 3.

Proof (of Theorem 1). For every i, $1 \leq i \leq 4$, let $\mathcal{A}^i$ and $\mathcal{B}^i$ be conditions (which exist by Proposition 10) such that

$$\mathbf{F}^{\mathcal{A}^i} \equiv \mathbf{R}^{\mathcal{B}^i} \ \text{ and } \ \Delta_q^{\mathsf{ATK}}(\mathbf{F}, \mathbf{R}) = \nu^{\mathsf{ATK}}(\mathbf{F}^{\mathcal{A}^i}, \overline{a}^i_q) = \nu^{\mathsf{ATK}}(\mathbf{R}^{\mathcal{B}^i}, \overline{b}^i_q), \quad \text{(A.10)}$$

for any attack ATK. As a consequence it holds that

$$\psi_{2n}[\mathbf{F}^{\mathcal{A}^1}\mathbf{F}^{\mathcal{A}^2}\mathbf{F}^{\mathcal{A}^3}\mathbf{F}^{\mathcal{A}^4}] \equiv \psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{R}^{\mathcal{B}^2}\mathbf{R}^{\mathcal{B}^3}\mathbf{R}^{\mathcal{B}^4}]$$

(recall that $\psi_{2n}[\mathbf{F}^{\mathcal{A}^1}\mathbf{F}^{\mathcal{A}^2}\mathbf{F}^{\mathcal{A}^3}\mathbf{F}^{\mathcal{A}^4}]$ is defined as $\psi_{2n}[\mathbf{FFFF}]^{\mathcal{A}}$ with $\mathcal{A} \overset{\text{def}}{=} \mathcal{A}^1 \wedge \mathcal{A}^2 \wedge \mathcal{A}^3 \wedge \mathcal{A}^4$).

For two monotone conditions $\mathcal{D}$ and $\mathcal{E}$ (defined on some random system), let $d \Rightarrow_q e$ denote the event that for all j, $1 \leq j \leq q$, if d_j holds then also e_j. Now, let $\overline{b}'^i_q$ denote the event $\overline{b}^i_q \wedge [b^i \Rightarrow_q [b^1 \wedge b^2 \wedge b^3 \wedge b^4]]$. As $\overline{b}^1_q \vee \overline{b}^2_q \vee \overline{b}^3_q \vee \overline{b}^4_q$ holds iff $\overline{b}'^1_q \vee \overline{b}'^2_q \vee \overline{b}'^3_q \vee \overline{b}'^4_q$ holds, it follows that

$$\begin{aligned}
& \Delta_q^{\mathsf{CPA}}(\psi_{2n}[\mathbf{FFFF}], \psi_{2n}[\mathbf{RRRR}]) \\
\overset{Prop.\ 9}{\leq} \quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{R}^{\mathcal{B}^2}\mathbf{R}^{\mathcal{B}^3}\mathbf{R}^{\mathcal{B}^4}], \overline{b}^1_q \vee \overline{b}^2_q \vee \overline{b}^3_q \vee \overline{b}^4_q) \\
= \quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{R}^{\mathcal{B}^2}\mathbf{R}^{\mathcal{B}^3}\mathbf{R}^{\mathcal{B}^4}], \overline{b}'^1_q \vee \overline{b}'^2_q \vee \overline{b}'^3_q \vee \overline{b}'^4_q) \\
\overset{union\ bound}{\leq} \quad & \sum_{i=1}^{4} \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{R}^{\mathcal{B}^2}\mathbf{R}^{\mathcal{B}^3}\mathbf{R}^{\mathcal{B}^4}], \overline{b}^i_q \wedge [b^i \Rightarrow_q [b^1 \wedge b^2 \wedge b^3 \wedge b^4]]) \\
\leq \quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{RRR}], \overline{b}^1_q) + \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RR}^{\mathcal{B}^2}\mathbf{RR}], \overline{b}^2_q) + \\
& \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RRR}^{\mathcal{B}^3}\mathbf{R}], \overline{b}^3_q) + \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RRRR}^{\mathcal{B}^4}], \overline{b}^4_q).
\end{aligned}$$

To complete the proof, we bound the four terms of the last step. Without loss of generality (since we are dealing with stateless systems), we assume that the distinguishers only make distinct queries. Let $\mathcal{C}$ ($\mathcal{C}'$) denote a monotone condition for any function defined by letting c_i (c_i') denote the event that the first i inputs of the function are distinct. Furthermore, note that

$$\begin{aligned} \mathbf{R}^{\mathcal{C}} &\equiv \mathbf{B}^{\mathcal{C}} \\ \mathbf{R}^{\mathcal{C}'} &\equiv \mathbf{B}^{\mathcal{C}'}. \end{aligned}$$

Since $\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}] \triangleright \mathbf{P} \,|\, \mathcal{B}^1 \equiv \mathbf{P}$, it follows that

$$\begin{aligned} & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{RRR}], \overline{b}_q^1) \\ = \quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}] \triangleright \psi_{2n}[\mathbf{RRR}], \overline{b}_q^1) \\ \overset{(i)}{\leq} \quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}] \triangleright \mathbf{P}, \overline{b}_q^1) + \\ & \Delta_q^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}] \triangleright \psi_{2n}[\mathbf{RRR}], \psi_{2n}[\mathbf{R}^{\mathcal{B}^1}] \triangleright \mathbf{P}) \\ \overset{(ii)}{\leq} \quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}] \triangleright \mathbf{P}, \overline{b}_q^1) + \Delta_q^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RRR}], \mathbf{P}) \\ \overset{Prop.\ 11}{\leq} \quad & \nu^{\mathsf{nCPA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}] \triangleright \mathbf{P}, \overline{b}_q^1) + \Delta_q^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RRR}], \mathbf{P}) \\ \overset{(ii), (3.3)}{\leq} \quad & \nu^{\mathsf{nCPA}}(\mathbf{R}^{\mathcal{B}^1}, \overline{b}_q^1) + 2 \cdot \frac{q^2}{2^{n+1}}. \end{aligned}$$

Further, as $\psi_{2n}[\mathbf{RRB}^{\mathcal{C}}\mathbf{B}] \,|\, \mathcal{C} \equiv \mathbf{B}$ and $\psi_{2n}[\mathbf{RR}^{\mathcal{B}^2}\mathbf{BB}] \,|\, \mathcal{B}^2 \equiv \mathbf{B}$, we get

$$\begin{aligned} & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RR}^{\mathcal{B}^2}\mathbf{RR}], \overline{b}_q^2) \\ \leq \quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RR}^{\mathcal{B}^2}\mathbf{R}^{\mathcal{C}}\mathbf{R}^{\mathcal{C}'}], \overline{b}_q^2 \vee \overline{c}_q \vee \overline{c}_q') \\ = \quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RR}^{\mathcal{B}^2}\mathbf{B}^{\mathcal{C}}\mathbf{B}^{\mathcal{C}'}], \overline{b}_q^2 \vee \overline{c}_q \vee \overline{c}_q') \\ \leq \quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RR}^{\mathcal{B}^2}\mathbf{BB}], \overline{b}_q^2) + \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RRB}^{\mathcal{C}}\mathbf{B}], \overline{c}_q) + \\ & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RRBB}^{\mathcal{C}'}], \overline{c}_q') \\ \overset{Prop.\ 11}{\leq} \quad & \nu^{\mathsf{nCPA}}(\psi_{2n}[\mathbf{RR}^{\mathcal{B}^2}\mathbf{BB}], \overline{b}_q^2) + \nu^{\mathsf{nCPA}}(\psi_{2n}[\mathbf{RRB}^{\mathcal{C}}\mathbf{B}], \overline{c}_q) + \\ & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RRBB}^{\mathcal{C}'}], \overline{c}_q') \\ \overset{(ii), b\text{-}bound}{\leq} \quad & \nu^{\mathsf{nCPA}}(\mathbf{R}^{\mathcal{B}^2}, \overline{b}_q^2) + 2 \cdot \frac{q^2}{2^{n+1}}. \end{aligned}$$

We also have

$$\begin{aligned}
& \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}\mathbf{R}\mathbf{R}^{\mathcal{B}^3}\mathbf{R}], \overline{b}^3_q) \\
\leq\quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}\mathbf{R}^{\mathcal{C}}\mathbf{R}^{\mathcal{B}^3}\mathbf{R}], \overline{c}_q \vee \overline{b}^3_q) \\
=\quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{R}^{\mathcal{B}^3}\mathbf{R}], \overline{c}_q \vee \overline{b}^3_q) \\
\leq\quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}\mathbf{B}\mathbf{R}^{\mathcal{B}^3}\mathbf{R}], \overline{b}^3_q) + \\
& \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{R}\mathbf{R}], \overline{c}_q) \\
\overset{(ii),\, b\text{-}bound}{\leq}\quad & \nu^{\mathsf{KPA}}(\mathbf{R}^{\mathcal{B}^3}, \overline{b}^3_q) + \frac{q^2}{2^{n+1}},
\end{aligned}$$

and

$$\begin{aligned}
& \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}\mathbf{R}\mathbf{R}\mathbf{R}^{\mathcal{B}^4}], \overline{b}^4_q) \\
\leq\quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}\mathbf{R}^{\mathcal{C}}\mathbf{R}^{\mathcal{C}'}\mathbf{R}^{\mathcal{B}^4}], \overline{c}_q \vee \overline{c}'_q \vee \overline{b}^4_q) \\
=\quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{B}^{\mathcal{C}'}\mathbf{R}^{\mathcal{B}^4}], \overline{c}_q \vee \overline{c}'_q \vee \overline{b}^4_q) \\
\leq\quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}\mathbf{B}\mathbf{B}\mathbf{R}^{\mathcal{B}^4}], \overline{b}^4_q) + \\
& \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{B}\mathbf{R}], \overline{c}_q) + \\
& \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}\mathbf{B}\mathbf{B}^{\mathcal{C}'}\mathbf{R}], \overline{c}'_q) \\
\overset{(ii),\, b\text{-}bound}{\leq}\quad & \nu^{\mathsf{KPA}}(\mathbf{R}^{\mathcal{B}^4}, \overline{b}^4_q) + 2 \cdot \frac{q^2}{2^{n+1}}.
\end{aligned}$$

Using the fact from (A.10), that $\Delta^{\mathsf{ATK}}_q(\mathbf{F}, \mathbf{R}) = \nu^{\mathsf{ATK}}(\mathbf{R}^{\mathcal{B}^i}, \overline{b}^i_q)$ for any attack ATK, concludes the proof. □

Proof (of Theorem 3). For every i, $1 \leq i \leq 5$, let $\mathcal{A}^i$ and $\mathcal{B}^i$ be conditions (which exist by Proposition 10) such that

$$\mathbf{F}^{\mathcal{A}^i} \equiv \mathbf{R}^{\mathcal{B}^i} \;\text{ and }\; \Delta^{\mathsf{ATK}}_q(\mathbf{F}, \mathbf{R}) = \nu^{\mathsf{ATK}}(\mathbf{F}^{\mathcal{A}^i}, \overline{a}^i_q) = \nu^{\mathsf{ATK}}(\mathbf{R}^{\mathcal{B}^i}, \overline{b}^i_q), \quad \text{(A.11)}$$

for any attack ATK. As a consequence it holds that

$$\psi_{2n}[\mathbf{F}^{\mathcal{A}^1}\mathbf{F}^{\mathcal{A}^2}\mathbf{F}^{\mathcal{A}^3}\mathbf{F}^{\mathcal{A}^4}\mathbf{F}^{\mathcal{A}^5}] \equiv \psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{R}^{\mathcal{B}^2}\mathbf{R}^{\mathcal{B}^3}\mathbf{R}^{\mathcal{B}^4}\mathbf{R}^{\mathcal{B}^5}].$$

Analogously, to the proof of Theorem 1, we get

$$\begin{aligned}
& \Delta_q^{\mathsf{CPA}}(\psi_{2n}[\mathbf{FFFFF}], \psi_{2n}[\mathbf{RRRRR}]) \\
\overset{Prop.\,9}{\leq} \quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{R}^{\mathcal{B}^2}\mathbf{R}^{\mathcal{B}^3}\mathbf{R}^{\mathcal{B}^4}\mathbf{R}^{\mathcal{B}^5}], \overline{b}_q^1 \vee \overline{b}_q^2 \vee \overline{b}_q^3 \vee \overline{b}_q^4 \vee \overline{b}_q^5) \\
\leq \quad & \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{RRRR}], \overline{b}_q^1) + \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RR}^{\mathcal{B}^2}\mathbf{RRR}], \overline{b}_q^2) + \\
& \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RRR}^{\mathcal{B}^3}\mathbf{RR}], \overline{b}_q^3) + \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RRRR}^{\mathcal{B}^4}\mathbf{R}], \overline{b}_q^4) + \\
& \nu^{\mathsf{CPA}}(\psi_{2n}[\mathbf{RRRRR}^{\mathcal{B}^5}], \overline{b}_q^5).
\end{aligned}$$

It remains to bound the five terms of the last step. Without loss of generality (since we are dealing with stateless systems) we assume that the distinguisher only makes distinct queries. Let $\mathcal{C}$ (respectively $\mathcal{C}'$ and $\mathcal{C}''$) denote a monotone condition for any function defined by letting c_i (respectively c_i' and c_i'') denote the event that the first i inputs of the function are distinct. Furthermore, note that $\mathbf{R}^{\mathcal{C}} \equiv \mathbf{B}^{\mathcal{C}}$ (respectively $\mathbf{R}^{\mathcal{C}'} \equiv \mathbf{B}^{\mathcal{C}'}$ and $\mathbf{R}^{\mathcal{C}''} \equiv \mathbf{B}^{\mathcal{C}''}$).

For a random permutation $\mathbf{Q}$ over $\mathcal{X}$, let us define $\langle \mathbf{Q} \rangle$ to be the $(\mathcal{X} \times \{0,1\}, \mathcal{X})$-random system defined as follows

$$\langle \mathbf{Q}(x_i, z_i) \rangle = \begin{cases} \mathbf{Q}(x_i) & \text{if } z_i = 0 \\ \mathbf{Q}^{-1}(x_i) & \text{if } z_i = 1. \end{cases}$$

Note that a CCA (nCCA) on Q is now the same as a CPA (nCPA) on $\langle Q \rangle$.

In the sequel, the following fact will be used. For any condition $\mathcal{B}$:

$$\begin{aligned}
& \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}}\mathbf{BB}], \overline{b}_q) \\
= \quad & \nu^{\mathsf{nCCA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}}\mathbf{BB}], \overline{b}_q) \\
\overset{(ii)}{\leq} \quad & \nu^{\mathsf{nCPA}}(\mathbf{R}^{\mathcal{B}}, \overline{b}_q),
\end{aligned} \tag{A.12}$$

where the equality follows from Theorem 2 in [Mau02] and the fact that

$$\mathsf{P}^{\langle \psi_{2n}[\mathbf{RBB}] \rangle}_{b_i | X^i Y^{i-1} b_{i-1}} = \mathsf{P}^{\langle \psi_{2n}[\mathbf{RBB}] \rangle}_{b_i | X^i b_{i-1}} \quad \text{for } i \geq 1.$$

Furthermore, note that there is a random system $\mathbf{G}$ satisfying[55]

$$\langle \psi_{2n}[\mathbf{RB}^{\mathcal{C}}\mathbf{B}] \rangle \,|\, \mathcal{C} \equiv \mathbf{G} \tag{A.13}$$

[55] $\mathbf{G}({}_L x_i \| {}_R x_i, z_i) := ({}_L y_i \| {}_R y_i)$, where ${}_L y_i \| {}_R y_i$ is chosen uniformly at random from $\{0,1\}^{2n}$ if $z_i = 0$ (or $i > 2^n$), and otherwise ${}_L y$ is chosen uniformly at random from $\{0,1\}^n$ and ${}_R y_i = \mathbf{R}({}_L y) \oplus \mathbf{P}(\langle i \rangle_n)$, where $\langle i \rangle_n$ is the n-bit standard binary encoding of i.

and hence by Proposition 11 it holds that

$$\begin{aligned}
& \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{B}], \overline{c}_q) \\
\stackrel{Prop.\ 11, (A.13)}{=} \quad & \nu^{\mathsf{nCCA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{B}], \overline{c}_q) \leq \frac{q^2}{2^{n+1}}.
\end{aligned} \tag{A.14}$$

Since $\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{R}^{\mathcal{C}}\mathbf{R}^{\mathcal{C}'}\mathbf{R}^{\mathcal{C}''}\mathbf{R}] \equiv \psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{B}^{\mathcal{C}}\mathbf{B}^{\mathcal{C}'}\mathbf{B}^{\mathcal{C}''}\mathbf{R}]$, we get

$$\begin{aligned}
& \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{R}\mathbf{R}\mathbf{R}\mathbf{R}^{\mathcal{B}^5}], \overline{b}_q^5) \stackrel{sym.}{=} \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{R}\mathbf{R}\mathbf{R}\mathbf{R}], \overline{b}_q^1) \\
\leq \quad & \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{R}^{\mathcal{C}}\mathbf{R}^{\mathcal{C}'}\mathbf{R}^{\mathcal{C}''}\mathbf{R}], \overline{b}_q^1 \vee \overline{c}_q \vee \overline{c}'_q \vee \overline{c}''_q) \\
\stackrel{Prop.\ 9}{=} \quad & \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{B}^{\mathcal{C}}\mathbf{B}^{\mathcal{C}'}\mathbf{B}^{\mathcal{C}''}\mathbf{R}], \overline{b}_q^1 \vee \overline{c}_q \vee \overline{c}'_q \vee \overline{c}''_q) \\
\leq \quad & \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{B}\mathbf{B}\mathbf{B}\mathbf{R}], \overline{b}_q^1) + \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{B}\mathbf{B}\mathbf{R}], \overline{c}_q) + \\
& \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{B}\mathbf{B}^{\mathcal{C}'}\mathbf{B}\mathbf{R}], \overline{c}'_q) + \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{B}\mathbf{B}\mathbf{B}^{\mathcal{C}''}\mathbf{R}], \overline{c}''_q) \\
\stackrel{(ii)}{\leq} \quad & \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^1}\mathbf{B}\mathbf{B}], \overline{b}_q^1) + \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{B}], \overline{c}_q) + \\
& \frac{q^2}{2^{n+1}} + \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{B}\mathbf{B}^{\mathcal{C}''}\mathbf{R}], \overline{c}''_q) \\
\stackrel{(A.12),(A.14)}{\leq} \quad & \nu^{\mathsf{nCPA}}(\mathbf{R}^{\mathcal{B}^1}, \overline{b}_q^1) + 3 \cdot \frac{q^2}{2^{n+1}}.
\end{aligned}$$

Clearly, it holds that $\psi_{2n}[\mathbf{R}\mathbf{R}^{\mathcal{B}^2}\mathbf{R}^{\mathcal{C}}\mathbf{R}^{\mathcal{C}'}\mathbf{R}] \equiv \psi_{2n}[\mathbf{R}\mathbf{R}^{\mathcal{B}^2}\mathbf{B}^{\mathcal{C}}\mathbf{B}^{\mathcal{C}'}\mathbf{R}]$ and thus

$$\begin{aligned}
& \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{R}\mathbf{R}\mathbf{R}^{\mathcal{B}^4}\mathbf{R}], \overline{b}_q^4) \stackrel{sym.}{=} \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{R}^{\mathcal{B}^2}\mathbf{R}\mathbf{R}\mathbf{R}], \overline{b}_q^2) \\
\leq \quad & \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{R}^{\mathcal{B}^2}\mathbf{R}^{\mathcal{C}}\mathbf{R}^{\mathcal{C}'}\mathbf{R}], \overline{b}_q^2 \vee \overline{c}_q \vee \overline{c}'_q) \\
\stackrel{Prop.\ 9}{=} \quad & \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{R}^{\mathcal{B}^2}\mathbf{B}^{\mathcal{C}}\mathbf{B}^{\mathcal{C}'}\mathbf{R}], \overline{b}_q^2 \vee \overline{c}_q \vee \overline{c}'_q) \\
\leq \quad & \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{R}^{\mathcal{B}^2}\mathbf{B}\mathbf{B}\mathbf{R}], \overline{b}_q^2) + \\
& \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{B}\mathbf{R}], \overline{c}_q) + \\
& \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{R}\mathbf{B}\mathbf{B}^{\mathcal{C}'}\mathbf{R}], \overline{c}'_q) \\
\stackrel{(ii)}{\leq} \quad & \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}^{\mathcal{B}^2}\mathbf{B}\mathbf{B}], \overline{b}_q^2) + \\
& \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{B}], \overline{c}_q) + \\
& \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{B}\mathbf{B}^{\mathcal{C}'}\mathbf{R}], \overline{c}'_q) \\
\stackrel{(A.12),(A.14)}{\leq} \quad & \nu^{\mathsf{nCPA}}(\mathbf{R}^{\mathcal{B}^2}, \overline{b}_q^2) + 2 \cdot \frac{q^2}{2^{n+1}}.
\end{aligned}$$

Finally, as $\psi_{2n}[\mathbf{R}\mathbf{R}^{\mathcal{C}}\mathbf{R}^{\mathcal{B}^3}\mathbf{R}^{\mathcal{C}'}\mathbf{R}] \equiv \psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{R}^{\mathcal{B}^3}\mathbf{B}^{\mathcal{C}'}\mathbf{R}]$ it follows that

$$
\begin{aligned}
& \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{R}\mathbf{R}^{\mathcal{B}^3}\mathbf{R}\mathbf{R}], \overline{b}_q^3) \\
\leq\; & \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{R}^{\mathcal{C}}\mathbf{R}^{\mathcal{B}^3}\mathbf{R}^{\mathcal{C}'}\mathbf{R}], \overline{c}_q \vee \overline{b}_q^3 \vee \overline{c}'_q) \\
\overset{Prop.\ 9}{=}\; & \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{R}^{\mathcal{B}^3}\mathbf{B}^{\mathcal{C}'}\mathbf{R}], \overline{c}_q \vee \overline{b}_q^3 \vee \overline{c}'_q) \\
\leq\; & \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{B}\mathbf{R}^{\mathcal{B}^3}\mathbf{B}\mathbf{R}], \overline{b}_q^3) + \\
& \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{R}\mathbf{B}^{\mathcal{C}'}\mathbf{R}], \overline{c}_q \vee \overline{c}'_q) \\
\overset{(i),(ii)}{\leq}\; & \nu^{\mathsf{KPA}}(\mathbf{R}^{\mathcal{B}^3}, \overline{b}_q^3) + \Delta_q^{\mathsf{KPA}}(\mathbf{R}, \mathbf{B}) + \\
& \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{B}\mathbf{B}^{\mathcal{C}'}\mathbf{R}], \overline{c}_q \vee \overline{c}'_q) \\
\leq\; & \nu^{\mathsf{KPA}}(\mathbf{R}^{\mathcal{B}^3}, \overline{b}_q^3) + \Delta_q^{\mathsf{KPA}}(\mathbf{R}, \mathbf{B}) + 2 \cdot \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{B}\mathbf{B}\mathbf{R}], \overline{c}_q) \\
\overset{(ii)}{\leq}\; & \nu^{\mathsf{KPA}}(\mathbf{R}^{\mathcal{B}^3}, \overline{b}_q^3) + \Delta_q^{\mathsf{KPA}}(\mathbf{R}, \mathbf{B}) + 2 \cdot \nu^{\mathsf{CCA}}(\psi_{2n}[\mathbf{R}\mathbf{B}^{\mathcal{C}}\mathbf{B}], \overline{c}_q) \\
\overset{(A.5),(A.14)}{\leq}\; & \nu^{\mathsf{KPA}}(\mathbf{R}^{\mathcal{B}^3}, \overline{b}_q^3) + 3 \cdot \frac{q^2}{2^{n+1}}.
\end{aligned}
$$

Using the fact from (A.11), that $\Delta_q^{\mathsf{ATK}}(\mathbf{F}, \mathbf{R}) = \nu^{\mathsf{ATK}}(\mathbf{R}^{\mathcal{B}^i}, \overline{b}_q^i)$ for any attack ATK, concludes the proof. □

A.3 Proofs for Chapter 4

A.3.1 The Increasing Chain and Increasing Chain Tree

In this section, we prove Theorems 4 and 5.

Proof (of Theorem 4). Let Π_0 denote the following random experiment for a CPA-distinguisher A of size t making at most q queries to its oracle:

$$
\begin{aligned}
&(k_1, r, \tau_1) \xleftarrow{\$} \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n, \\
&\mathsf{A} \diamond \mathrm{IC}^{\mathsf{F}}_{k_1, r, \tau_1}, \\
&b \leftarrow \mathsf{A}.
\end{aligned}
$$

Recall the algorithm describing IC^{F} on page 47. Note that for any query x issued by A and any $s \in \{1, \ldots, N\}$, the sequence $(\tau_1, \ldots, \tau_s)$ (resulting from the second for-loop) does not depend on $x[s,N]$. Hence, $(\tau_1,\dots,\tau_s)$ can be reused for any other query x' for which $x'[1,s-1]=x[1,s-1]$. We assume that $\mathrm{IC}^{\mathsf{F}}_{k_1,r,\tau_1}$ reuses previously computed τ-values (for saving calls to F) whenever possible, by maintaining a look-up table with all the entries $(x[1,i],\tau_{i+1})$ for which x is a query to $\mathrm{IC}^{\mathsf{F}}_{k_1,r,\tau_1}$, $i\in\{1,\dots,N\}$, and $x[i]=1$. We also assume that the calls to F in the first for-loop are pre-processed and cached. For $j=1,\dots,N$, let Π_{2j-1} be the same experiment as Π_{2j-2} except that F_{k_j} is replaced by a random function $\mathbf{R}_j$, and let Π_{2j} be the same experiment as Π_{2j-1} except that for each query x issued by A, for which $x[j]=1$ and $x[1,j]$ is not in the look-up table, the output of $\mathbf{R}_j$ is replaced by a uniform random $R\in\{0,1\}^n$ and $(x[1,j],R)$ is inserted into the table. Let S_i be the event that $b=1$ in Π_i, for $i=0,\dots,2N$. Now, as Π_{2N} is equivalent to

$$\mathsf{A}\diamond\mathbf{R}_{N,n},\ b\leftarrow\mathsf{A},$$

we get

$$\begin{aligned}
&\mathbf{Adv}^{\mathsf{CPA}}_{\mathsf{A}}(\mathrm{IC}^{\mathsf{F}},\mathbf{R}_{N,n})\stackrel{\text{def}}{=}\left|\Pr[S_0]-\Pr[S_{2N}]\right|\\
\leq\ &\sum_{j=1}^{N}\left|\Pr[S_{2j-2}]-\Pr[S_{2j-1}]\right|+\sum_{j=1}^{N}\left|\Pr[S_{2j-1}]-\Pr[S_{2j}]\right|\\
\leq\ &\sum_{j=1}^{N}\mathbf{Adv}^{\mathsf{KPA}}_{t',\min\{q+1,2^{j-1}+1\}}(\mathsf{F},\mathbf{R})+\sum_{j=1}^{N}\min\left\{\frac{(q+1)q}{2^{n+1}},\frac{(2^{j-1}+1)2^{j-1}}{2^{n+1}}\right\}\\
\leq\ &N\cdot\left(\mathbf{Adv}^{\mathsf{KPA}}_{t',q+1}(\mathsf{F},\mathbf{R})+\frac{(q+1)q}{2^{n+1}}\right),
\end{aligned}$$

due to the triangle inequality and the following two facts. First, for $j=1,\dots,N$, A can be transformed to the following WPRF distinguisher A′ for F, which makes at most $\min(q+1,2^{j-1}+1)$ oracle invocations and has advantage at least $|\Pr[S_{2j-2}]-\Pr[S_{2j-1}]|$. A′ with oracle T, simulates the experiment Π_{2j-2} if T is an instance of F and Π_{2j-1} if T is a random function $\mathbf{R}$ (which is possible as all queries to F_{k_j} in Π_{2j-2} and to $\mathbf{R}_j$ in Π_{2j-1} are distributed uniformly at random). Finally, A′ decides as A does. Second, Π_{2j-1} and Π_{2j} are equivalent experiments as long as no collision among the inputs on which $\mathbf{R}_j$ is invoked occurs. As these inputs are at most $\min\{q+1,2^{j-1}+1\}$ and random, the probability of this event is upper bounded by $\min\left\{(q+1)q/2^{n+1},(2^{j-1}+1)2^{j-1}/2^{n+1}\right\}$. □

Proof (of Theorem 5). Let Π_0 denote the random experiment

$$(k, r) \xleftarrow{\$} \{0,1\}^n \times \{0,1\}^n,$$
$$\mathsf{A} \diamond \mathrm{ICT}^{\mathsf{F}}_{k,r},$$
$$b \leftarrow \mathsf{A},$$

where A is a KPA-distinguisher A of size t that makes at most q queries of total output length at most μ and has advantage $\mathbf{Adv}^{\mathsf{A}}(\mathrm{ICT}^{\mathsf{F}}, \mathbf{R}_{n,*}) = \mathbf{Adv}^{\mathsf{KPA}}_{t,q,\mu}(\mathrm{ICT}^{\mathsf{F}}, \mathbf{R}_{n,*})$.

Let d denote the maximal number of generated keys (for F), needed for answering the queries to $\mathrm{ICT}^{\mathsf{F}}$ issued by A, and note that for $i \in \{1, \ldots, d\}$, the i-th instantiation F_{k_i} is queried at most $q_i = q \cdot (2^{i-1}+1)$ times. Let Π_i denote the same random experiment as Π_{i-1} except that the i-th instance of F is replaced by a beacon $\mathbf{B}$. Furthermore, let Π_{d+1} denote the following random experiment;

$$\mathsf{A} \diamond \mathbf{R}_{n,*}, b \leftarrow \mathsf{A}.$$

Finally, for $i \in \{0, \ldots, d+1\}$, let S_i denote the event that $b = 1$ in Π_i. Now, as Π_d is equivalent to[56]

$$\mathsf{A} \diamond \mathbf{B}_{n,*}, b \leftarrow \mathsf{A},$$

we get

$$\Big|\Pr[S_d] - \Pr[S_{d+1}]\Big| = \mathbf{Adv}^{\mathsf{A}}(\mathbf{R}_{n,*}, \mathbf{B}_{n,*}) \le \underbrace{\mathbf{Adv}^{\mathsf{VOL\text{-}KPA}}_{t,q,\mu}(\mathbf{R}_{n,*}, \mathbf{B}_{n,*})}_{\le \Delta^{\mathsf{KPA}}_q(\mathbf{R},\mathbf{B}) \overset{(A.5)}{\le} \frac{q^2}{2^{n+1}}}. \quad (A.15)$$

Further, for $i \in \{0, \ldots, d-1\}$, it follows that

$$\begin{aligned} \Big|\Pr[S_i] - \Pr[S_{i+1}]\Big| &\le \mathbf{Adv}^{\mathsf{KPA}}_{t',q_i}(\mathsf{F}, \mathbf{B}) \quad (A.16)\\ &\overset{tri.ineq.}{\le} \mathbf{Adv}^{\mathsf{KPA}}_{t',q_i}(\mathsf{F}, \mathbf{R}) + \underbrace{\mathbf{Adv}^{\mathsf{KPA}}_{t',q_i}(\mathbf{R}, \mathbf{B})}_{\le \Delta^{\mathsf{KPA}}_{q_i}(\mathbf{R},\mathbf{B}) \overset{(A.5)}{\le} \frac{q_i^2}{2^{n+1}}}, \end{aligned}$$

where the first inequality follows from the fact that A can be transformed into the following KPA-distinguisher A' for F and $\mathbf{B}$ that makes q_i oracle

[56] Here $\mathbf{B}_{n,*}$ is a $(\{0,1\}^n \times \mathbb{N}, \{0,1\}^*)$-random system that on any input (x, l) outputs a random l-bit string.

queries and has advantage $|\Pr[S_i] - \Pr[S_{i+1}]|$. A' with oracle T simply simulates the random experiment that is equivalent to Π_i if T is an instance of F and to Π_{i+1} if T is a beacon $\mathbf{B}$ (this is possible as the inputs to $\mathsf{F}_{k_{i+1}}$ in Π_i and to $\mathbf{B}_{i+1}$ in Π_{i+1} are distributed uniformly at random). Finally, A' decides as A does.

It follows that

$$\begin{aligned}
& \mathbf{Adv}^{\mathsf{KPA}}_{t,q,\mu}(\mathrm{ICT}^{\mathsf{F}}, \mathbf{R}_{n,*}) = \mathbf{Adv}^{\mathsf{A}}(\mathrm{ICT}^{\mathsf{F}}, \mathbf{R}_{n,*}) \\
= \quad & \Big|\Pr[S_0] - \Pr[S_{d+1}]\Big| \overset{tri.\ ineq.}{\leq} \sum_{i=0}^{d} \Big|\Pr[S_i] - \Pr[S_{i+1}]\Big| \\
\overset{(A.15),(A.16)}{\leq} \quad & \frac{q^2}{2^{n+1}} + \sum_{i=0}^{d-1} \left(\mathbf{Adv}^{\mathsf{KPA}}_{t',q_i}(\mathsf{F}, \mathbf{R}) + \frac{q_i^2}{2^{n+1}} \right) \\
\leq \quad & \frac{q^2}{2^{n+1}} + \frac{(\sum_{i=0}^{d-1} q_i)^2}{2^{n+1}} + d \cdot \mathbf{Adv}^{\mathsf{KPA}}_{t',q\cdot(2^{d-1}+1)}(\mathsf{F}, \mathbf{R}) \\
\leq \quad & \frac{q^2}{2^{n+1}} + \frac{q^2 \cdot (2^d - 1)^2 + q^2 \cdot d^2}{2^{n+1}} + d \cdot \mathbf{Adv}^{\mathsf{KPA}}_{t',q\cdot(2^{d-1}+1)}(\mathsf{F}, \mathbf{R}) \\
\leq \quad & \frac{4^d \cdot q^2}{2^n} + d \cdot \mathbf{Adv}^{\mathsf{KPA}}_{t',q\cdot(2^{d-1}+1)}(\mathsf{F}, \mathbf{R}).
\end{aligned}$$

□

A.3.2 Encryption Schemes from WPRFs and WMACs

In this section, we prove Proposition 6, Theorem 6, and Theorem 7.

Proof (of Proposition 6). Recall that $\mathcal{SE}_1 = (\mathsf{Enc}, \mathsf{Dec})$ is defined as

$$\mathsf{Enc}_k(m) \overset{\text{def}}{=} (r, \mathsf{V}_k(r, |m|) \oplus m)\,. \tag{A.17}$$

For an IND-P2-C0-adversary A for $\mathcal{SE}_1$ with resources (t, q, μ) and advantage $\mathbf{Adv}^{\mathsf{A}}(\mathcal{SE}_1) = \mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C0}}_{t,q,\mu}(\mathcal{SE}_1)$, let Π_0 denote the experiment:

$$\begin{aligned}
& k \overset{\$}{\leftarrow} \{0,1\}^{\kappa}, \\
& b \overset{\$}{\leftarrow} \{0,1\}, \\
& \mathsf{A} \diamond [\mathsf{Enc}_k, \mathsf{Enc}_k(\mathcal{LR}(\cdot,\cdot,b))], \\
& \hat{b} \leftarrow \mathsf{A}.
\end{aligned}$$

Furthermore, let Π_1 be the same random experiment as Π_0, except that V_k is replaced by $\mathbf{R}_{n,*}$ in (A.17). Let Π_2 be the same random experiment as Π_1, except that the output from A's query to $\mathsf{Enc}_k(\mathcal{LR}(\cdot,\cdot,b))$ is replaced by a uniform random string. For $i \in \{0,1,2\}$, let S_i denote the event that $\hat{b} = b$ in random experiment Π_i. Then

$$\begin{aligned} &\mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C0}}_{t,q,\mu}(\mathcal{SE}_1) = \mathbf{Adv}^{\mathsf{A}}(\mathcal{SE}_1) = 2 \cdot \Pr[S_0] - 1 \\ = \;& 2\left(\Pr[S_2] + \sum_{i=0}^{1} \Pr[S_i] - \Pr[S_{i+1}]\right) - 1 \\ \leq \;& 2 \cdot \left(\frac{1}{2} + \mathbf{Adv}^{\mathsf{VOL\text{-}KPA}}_{t',q,\mu}(\mathsf{V}) + \frac{q-1}{2^n}\right) - 1, \end{aligned}$$

where the inequality follows from the following three facts. First, A can be transformed into the following VOL-WPRF distinguisher A′ for V, that makes at most q oracle queries totaling at most μ bits and has advantage $\Pr[S_0] - \Pr[S_1]$. A′ with oracle T simply simulates the experiment Π_0 if T is an instance of V and Π_1 if T is a uniform random VOL-function $\mathbf{R}$ (this is possible as the inputs to V_k in Π_0 and to $\mathbf{R}_{n,*}$ in Π_1 are distributed uniformly at random), and then A′ returns whatever A does. Second, Π_1 and Π_2 are equivalent experiments as long as the auxiliary random string r in A's call to $\mathsf{Enc}_k(\mathcal{LR}(\cdot,\cdot,b))$ is distinct from the auxiliary random strings used in A's calls to $\mathsf{Enc}_k(\cdot)$, an event upper bounded by $(q-1)/2^n$. Third, $\Pr[S_2] = 1/2$ since $\hat{b}$ is independent of b. □

Proof (of Theorem 6). Recall that $\mathcal{SE}_2 = (\mathsf{Enc}, \mathsf{Dec})$ is defined as

$$\mathsf{Enc}_{k_1,k_2}(m) \overset{\text{def}}{=} \big(r, \underbrace{\mathsf{V}_{k_1}(r,|m|) \oplus m}_{c}, \mathsf{W}_{k_2}(r\|c)\big). \tag{A.18}$$

The proof of the first inequality consists of two parts. For the first part, i.e., $\mathbf{InSec}^{\mathsf{INT\text{-}CTXT}}_{t,q,\mu,q',\mu'}(\mathcal{SE}_2) \leq q' \cdot \mathbf{InSec}^{\mathsf{UF\text{-}KPA}}_{t,q,\mu+qn+\mu'}(\mathsf{W})$, we refer to [BN00]. For the second part, let Π_0 denote the following random experiment for an INT-CTXT-adversary A for $\mathcal{SE}_2$ with resources (t,q,μ,q',μ') and success probability $\mathbf{Succ}^{\mathsf{A}}(\mathcal{SE}_2) = \mathbf{InSec}^{\mathsf{INT\text{-}CTXT}}_{t,q,\mu,q',\mu'}(\mathcal{SE}_2)$:

$$\begin{aligned} &(k_1,k_2) \overset{\$}{\leftarrow} \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_2}, \\ &\mathsf{A}\Diamond[\mathsf{Enc}_{k_1,k_2}, \mathsf{Dec}^*_{k_1,k_2}]. \end{aligned}$$

Let Π_1 denote the same random experiment as Π_0 except that V_{k_1} has been replaced by $\mathbf{R}_{n,*}$ in (A.18). Furthermore, let Π_2 be the same random experiment as Π_1 except that $\mathbf{R}_{n,*}$ is replaced by a VOL-beacon $\mathbf{B}_{n,*}$ (see

Footnote 56 on page 103). For $i \in \{0,1,2\}$, let $\mathcal{E}_i$ denote the event that D^* outputs 1 in Π_i. Then

$$\begin{aligned}
&\mathbf{InSec}^{\mathsf{INT\text{-}CTXT}}_{t,q,\mu,q',\mu'}(\mathcal{SE}_2) = \mathbf{Succ}^{\mathsf{A}}(\mathcal{SE}_2) = \Pr[\mathcal{E}_0] \\
&= \Big(\Pr[\mathcal{E}_0] - \Pr[\mathcal{E}_1]\Big) + \Big(\Pr[\mathcal{E}_1] - \Pr[\mathcal{E}_2]\Big) + \Pr[\mathcal{E}_2] \\
&\le \mathbf{Adv}^{\mathsf{VOL\text{-}KPA}}_{t',q,\mu}(\mathsf{V}) + \frac{(q-1)q}{2^{n+1}} + q' \cdot \mathbf{InSec}^{\mathsf{UF\text{-}KPA}}_{t',q,\mu+qn+\mu'}(\mathsf{W}),
\end{aligned}$$

where the inequality follows from the following three facts. First, A can be transformed into the following VOL-KPA distinguisher A' of for V that issues at most q queries of total output length μ and has advantage $|\Pr[\mathcal{E}_0] - \Pr[\mathcal{E}_1]|$. A' with oracle T simply simulates Π_0 if T is an instance of V and Π_1 if T is a uniform random VOL-function $\mathbf{R}$ (this is possible as the inputs to V_{k_1} in Π_0 and to $\mathbf{R}_{n,*}$ in Π_1 are distributed uniformly at random), and then A' outputs 1 if and only if A is successful. Second, Π_1 and Π_2 are equivalent random experiments unless the auxiliary random r-values (in A's queries to $\mathsf{Enc}_{k_1,k_2}(\cdot)$) are not all distinct, an event upper bounded by $q(q-1)/2^{n+1}$. Third, from A we can construct a $(t', q, \mu + qn + \mu', \Pr[\mathcal{E}_2]/q')_{\mathsf{KPA}}$-forger A'' for W. A'' picks a random element $i \in \mathbb{Z}_{q'}$ and simply simulates Π_2 but with its own oracle in place of W_{k_2} (which is possible as all inputs are distributed uniformly at random due to the beacon) until A makes its $(i+1)$-th query to $\mathsf{Dec}^*_{k_1,k_2}$. Then A'' returns this query as its forgery.

For the proof of the second inequality of the theorem, let A denote an IND-P2-C2-adversary for $\mathcal{SE}_2$ with resources (t, q, μ, q', μ') that has advantage $\mathbf{Adv}^{\mathsf{A}}(\mathcal{SE}_2) = \mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C2}}_{t,q,\mu,q',\mu'}(\mathcal{SE}_2)$. Furthermore, let Π'_0 denote the IND-P2-C2 random experiment for A, i.e.,

$$\begin{aligned}
&(k_1,k_2) \xleftarrow{\$} \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_2}, \\
&b \xleftarrow{\$} \{0,1\}, \\
&\mathsf{A}\Diamond[\mathsf{Enc}_{k_1,k_2}, \mathsf{Dec}_{k_1,k_2}, \mathsf{Enc}_{k_1,k_2}(\mathcal{LR}(\cdot,\cdot,b))], \\
&\hat{b} \leftarrow \mathsf{A}.
\end{aligned}$$

Without loss of generality we assume that A does not query Dec_{k_1,k_2} with an output from Enc_{k_1,k_2}. Let Π'_1 be the same random experiment as Π'_0, except that all queries to Dec_{k_1,k_2} are rejected. Moreover, let S_i for $i \in$

$\{0,1\}$ denote the event that $\hat{b} = b$ in Π'_i. Then

$$\begin{aligned}
&\mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C2}}_{t,q,\mu,q',\mu'}(\mathcal{SE}_2) = \mathbf{Adv}^{\mathsf{A}}(\mathcal{SE}_2) = 2 \cdot \Pr[S_0] - 1 \\
= \;& 2 \cdot \Big(\Pr[S_0] - \Pr[S_1]\Big) + 2 \cdot \Pr[S_1] - 1 \\
\leq \;& 2 \cdot \Pr[\mathcal{E}] + \mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C0}}_{t,q,\mu}(\mathcal{SE}_2) \leq 2 \cdot \Pr[\mathcal{E}] + \mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C0}}_{t',q,\mu}(\mathcal{SE}_1),
\end{aligned}$$

where $\mathcal{E}$ denotes the event that a query to Dec_{k_1,k_2} in Π'_1 (or Π'_0) is a valid ciphertext (and is hence not rejected). The first inequality follows from the fact that Π'_0 and Π'_1 are equivalent random experiments unless $\mathcal{E}$ occurs, and since Π'_1 is the IND-P2-C0 random experiment for A and $\mathcal{SE}_2$ (with the slight modification that A does not bother to query any decryption queries, as they are all rejected). The second inequality is trivially satisfied as the WMAC is superfluous by Proposition 6.

As A can be transformed into the following INT-CTXT-adversary A‴ for $\mathcal{SE}_2$ with resources (t', q, μ, q', μ') and success probability $\Pr[\mathcal{E}]$, we get

$$\Pr[\mathcal{E}] \leq \mathbf{InSec}^{\mathsf{INT\text{-}CTXT}}_{t',q,\mu,q',\mu'}(\mathcal{SE}_2).$$

A‴ simply runs A, by answering A's encryption queries with its own encryption oracle and rejecting all of A's decryption queries. In addition, A‴ forwards A's decryption queries to its own verification oracle Dec*. If A presents its challenge input (m_0, m_1), A‴ queries its encryption oracle with m_b for a random chosen bit b and returns the result to A. ☐

Proof (of Theorem 7). Recall that $\mathcal{SE}_3 = (\mathsf{Enc}, \mathsf{Dec})$ is defined as

$$\mathsf{Enc}_{k_1,k_2}(m) \stackrel{\text{def}}{=} \big(r, \mathsf{V}_{k_1}(r, |m|) \oplus m, \mathsf{W}_{k_2}(r)\big). \tag{A.19}$$

Consider an IND-P2-C1-adversary A with resources (t, q, μ, q', μ') and advantage $\mathbf{Adv}^{\mathsf{A}}(\mathcal{SE}_3) = \mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C1}}_{t,q,\mu,q',\mu'}(\mathcal{SE}_3)$, and let Π_0 denote the corresponding IND-P2-C1 random experiment for A, i.e.,

$$\begin{aligned}
&(k_1, k_2) \stackrel{\$}{\leftarrow} \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_2}, \\
&b \stackrel{\$}{\leftarrow} \{0,1\}, \\
&\mathsf{A}\diamond[\mathsf{Enc}_{k_1,k_2}, \mathsf{Dec}_{k_1,k_2},\ \mathsf{Enc}_{k_1,k_2}(\mathcal{LR}(\cdot,\cdot,b))], \\
&\hat{b} \leftarrow \mathsf{A}.
\end{aligned}$$

Furthermore, let Π_1 be the same random experiment as Π_0 except for replacing A with the following adversary B that has the same advantage as

A and does not issue any query to Dec_{k_1,k_2} for which the auxiliary random part is the same as for a ciphertext returned previously by Enc_{k_1,k_2}. To be precise, let $\ell_{\max}$ denote the maximal length of the second input part of the decryption queries issued by A (clearly $\ell_{\max} < \mu'$). The adversary B simply runs A and for each encryption query m issued by A, B appends zeroes such that it is of length $\ell_{\max}$, i.e., $m' = m \| 0^{\ell_{\max}-|m|}$, and then queries the encryption oracle with m'. On output (r, c', w) from the encryption oracle, B returns $(r, c'[1, |m|], w)$ to A and stores $(m', (r, c', w))$ in a look-up table. If A queries some decryption query, say (r, c, w'), for which r occurs in the look-up table, say as $(m', (r, c', w))$, B returns $c \oplus c'[1, |c|] \oplus m'[1, |c|]$ if $w = w'$ and otherwise rejects. When A presents its challenge input (m_0, m_1), B queries its encryption oracle with m_b for a random bit b and returns the result to A. Finally, B decides as A does. Further, let Π_2 be defined as Π_1 except that all queries to Dec_{k_1,k_2} are rejected.

Now, for $i \in \{0, 1, 2\}$, let S_i denote the event that $\hat{b} = b$ in Π_i. Then

$$
\begin{aligned}
&\mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C1}}_{t,q,\mu,q',\mu'}(\mathcal{SE}_3) = \mathbf{Adv}^{\mathsf{A}}(\mathcal{SE}_3) = 2 \cdot \Pr[S_0] - 1 \\
= \quad & 2 \cdot \Big(\Pr[S_0] - \Pr[S_1]\Big) + 2 \cdot (\Pr[S_1] - \Pr[S_2]) + 2 \cdot \Pr[S_2] - 1 \\
\leq \quad & 2 \cdot \Pr[\mathcal{E}] + \mathbf{Adv}^{\mathsf{IND\text{-}P2\text{-}C0}}_{t',q,\mu+q\mu'}(\mathcal{SE}_1),
\end{aligned}
$$

where $\mathcal{E}$ denotes the event that B queries a valid ciphertext to its decryption oracle. The inequality follows from the following facts. First, $\Pr[\Pi_0] = \Pr[\Pi_1]$ as B decides as A does. Second, Π_1 and Π_2 are equivalent experiments unless the event $\mathcal{E}$ occurs. Third, Π_2 corresponds to the IND-P2-C0 experiment for B and $\mathcal{SE}_3$ (with the slight modification that B does not bother to query any decryption queries, as they are all rejected), and it is trivial to transform B into an $\mathsf{IND\text{-}P2\text{-}C0}$-adversary B' for $\mathcal{SE}_1$ with resources $(t', q, \mu + q\mu')$ and advantage at least $2 \cdot \Pr[S_2] - 1$ (B' simply runs B, answering its oracle queries with help of its own oracles and an instantiation of W, and finally decides as B does).

It remains to show

$$\Pr[\mathcal{E}] \leq q' \cdot \mathbf{InSec}^{\mathsf{UF\text{-}KPA}}_{t',q}(\mathsf{W}),$$

which follows as B can be transformed to a $(t', q, \Pr[\mathcal{E}]/q')_{\mathsf{KPA}}$-forger B'' for W. B'' picks a random $i \in \mathbb{Z}_{q'}$ and runs B by answering its encryption-oracle queries with help of its own oracle (and an instantiation of V) and by rejecting its first i queries to the decryption oracle. When B (if at all) issues its i-th decryption query (r_i, c_i, w_i), B'' returns (r_i, w_i) as its forgery (without making any extra calls to its oracle). $\square$