Diss. ETH No. 13106

On the Statistical Testing of Block Ciphers

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
Doctor of Technical Sciences

presented by
RICHARD J. DE MOLINER
Dipl. El.-Ing. ETH
Dipl. Informatik-Ing. ETH
born February 5, 1962
citizen of Zug ZG

accepted on the recommendation of
Prof. Dr. James L. Massey, referee
Dr. Markus Dichtl, co-referee
Prof. Dr. Ueli Maurer, co-referee

1999

Acknowledgments

I thank Professor James L. Massey for giving me the opportunity to do a doctoral dissertation with him. I further thank Jim for the care and effort he took in forming my approach to research. I am especially grateful for his patience and perseverance in doing this, although it took me some time to understand and appreciate his method. Working with Jim was a pleasure and an extraordinary experience. Thank you Jim!

I thank Professor George Moschytz for his exemplary guidance of the Signal and Information Processing Laboratory. I thank Dr. Markus Dichtl and Professor Ueli Maurer for acting as the co-referees.

The atmosphere at the Signal and Information Processing Laboratory was great. Among the many people who contributed to this great atmosphere I mention here only a few. I greatly appreciated the friendship within the "last" members of the Information Theory group: Beat Keusch, Gerhard Kramer, Zsolt Kukorelly, Urs Loher and Jossy Sayir. I greatly appreciated the relaxed atmosphere fostered by our last postdocs: Anne Canteaut, Alex Grant and Xuduan Lin.

The final word of gratitude must go to my closest loved ones. My parents, Rosa and Richard De Moliner, made my education possible and my companion, Cornelia Thalmann, shared my ups and downs during my doctoral dissertation and supported me with her patience, understanding and love. I am deeply grateful for that.

Abstract

Tests that are capable of analyzing any practical block cipher, no matter what the internal structure of the block cipher may be, are the subject of this work. It is argued that such tests must be statistical.

A discrete memoryless source producing a fixed-length sequence of output digits from a finite alphabet is considered. The problem of deciding whether the single letter probability distribution of the discrete memoryless source is equal to a given probability distribution or not is analyzed in detail. For this problem of statistical hypothesis testing the Pearson statistic is used. What can validly be concluded from statistical hypothesis testing is carefully considered.

We show that if a cryptanalyst cannot solve at least one of two basic problems for a given block cipher, then he cannot "break" this block cipher. These two basic problems are (1) to find an algorithm that is *distinguishing* for the given block cipher and (2) to find an algorithm that is *key-subset distinguishing* for the given block cipher and for a given decomposition of the key space.

An approach to finding an algorithm that is distinguishing for a given block cipher as well as an approach to finding an algorithm that is key-subset distinguishing for a given block cipher and for a given decomposition of the key space are described. These two approaches form the framework for the statistical testing of block ciphers.

A family of tests called bit-dependency tests is presented. The aim of a bit-dependency test is to say as much as possible about the quality of a block cipher when only a given subset of bits of the plaintext blocks and a given subset of bits of the corresponding ciphertext blocks

are observed.

Keywords: cryptography, cryptanalysis, block ciphers, bit-dependency tests, statistical hypothesis testing, statistical tests, Pearson statistic.

Kurzfassung

Der Gegenstand dieser Arbeit sind Tests, welche praktische Blockverschlüssler analysieren, ohne deren interne Struktur zu berücksichtigen. Es wird begründet, weshalb solche Tests statistisch sind.

Wir betrachten eine diskrete, gedächtnislose Quelle, welche eine Folge fester Länge von Ausgangssymbolen aus einem endlichen Alphabet generiert. Das Problem, zu entscheiden, ob die Wahrscheinlichkeitsverteilung der Ausgangssymbole gleich einer vorgegebenen Wahrscheinlichkeitsverteilung ist oder nicht, wird im Detail analysiert. Für dieses Problem des Testens statistischer Hypothesen verwenden wir die Pearsonstatistik. Dabei wird gründlich überlegt, was vom Testen statistischer Hypothesen wirklich gefolgert werden kann.

Wir zeigen, dass so lange ein Kryptoanalyst nicht zumindest eines von zwei grundlegenden Problemen für einen vorgegebenen Blockverschlüssler lösen kann, so lange wird dieser Kryptoanalyst diesen Blockverschlüssler auch nicht "brechen" können. Diese beiden grundlegenden Probleme sind (1) einen Algorithmus zu finden, der für den vorgegebenen Blockverschlüssler *unterscheidend* ist, und (2) einen Algorithmus zu finden, der für den vorgegebenen Blockverschlüssler und eine vorgegebene Unterteilung des Schlüsselraumes *schlüsselteilmengeunterscheidend* ist.

Im Folgenden beschreiben wir ein Verfahren, um einen Algorithmus zu finden, der für einen vorgegebenen Blockverschlüssler unterscheidend ist; ebenfalls wird ein Verfahren angegeben, um einen Algorithmus zu finden, der für einen vorgegebenen Blockverschlüssler und für eine vorgegebene Unterteilung des Schlüsselraumes schlüsselteilmengeunterschei-

dend ist. Diese beiden Verfahren bilden den Rahmen für das statistische Testen von Blockverschlüsslern.

Schliesslich wird eine Familie von Tests, sogenannte Bitabhängigkeitstests, vorgestellt. Das Ziel von Bitabhängigkeitstests ist, so viel wie möglich über die Qualität eines Blockverschlüsslers auszusagen, wenn nur eine vorgegebene Untermenge von Bits der Klartextblöcke und nur eine vorgegebene Untermenge von Bits der Kryptogrammblöcke beobachtet werden.

Stichworte: Kryptographie, Kryptoanalyse, Blockverschlüssler, Bitabhängigkeitstests, Testen statistischer Hypothesen, statistische Tests, Pearsonstatistik.

Contents

Chapter 1

Introduction

There is a paucity of literature on the statistical testing of block ciphers. From some of the few publications on this subject, one learns more about how not to do such statistical testing rather than about how to do it properly. For instance, in [2] a block cipher is tested using 13 tests. The first and second tests have no dependence whatsoever on the tested block cipher—the results of these tests say something about the quality of the pseudo-random number generator used but say nothing about the tested block cipher itself. The third and fourth tests test the same behavior of a block cipher—running either test twice is equivalent to running both tests once.

These examples give an idea of the pitfalls that one can encounter in the statistical testing of block ciphers. Designing good tests is precision work and requires careful analysis.

This dissertation is organized as follows. In Chapter 2 the mathematical theory of statistical hypothesis testing is presented in the manner that we will use it in Chapter 4 for the statistical testing of block ciphers. What can validly be concluded from statistical hypothesis testing is carefully considered.

In Chapter 3 we show how a cryptanalyst can use algorithms of a certain kind to attack a block cipher and we establish when a cryptanalyst cannot break a given block cipher. From these considerations,

we formulate two basic problems that a cryptanalyst can attempt to solve. We show that if a cryptanalyst cannot solve at least one of these two basic problems for a given block cipher, then he cannot break this block cipher. These two basic problems are (1) to find an algorithm that is *distinguishing* for a given block cipher and (2) to find an algorithm that is *key-subset distinguishing* for a given block cipher and for a given decomposition of the key space.

In Chapter 4 we describe an approach to finding an algorithm that is distinguishing for a given block cipher as well as an approach to finding an algorithm that is key-subset distinguishing for a given block cipher and for a given decomposition of the key space. We show that the core of any such algorithm is an algorithm for *extracting a feature* from an invertible function.

In Chapter 5 we present a family of algorithms for extracting a feature from an invertible function for what we call bit-dependency tests. The aim of a bit-dependency test is to say as much as possible about the quality of a block cipher when only a given subset of bits of the plaintext blocks and a given subset of bits of the corresponding ciphertext blocks are observed.

Chapter 2

Statistical Hypothesis Testing

In this chapter we present the mathematical theory of statistical hypothesis testing in the manner that we will use it for the statistical testing of block ciphers.

This chapter is organized as follows. In Section 2.1 we introduce the general model of statistical hypothesis testing and consider the two kinds of error that one can make in a decision. Section 2.2 treats the Neyman-Pearson solution for choosing a hypothesis. The Neyman-Pearson solution is applicable to one of the cases we will encounter later in the statistical testing of block ciphers. In Section 2.3 we formulate solutions for choosing a hypothesis for all other cases that we will encounter later in the statistical testing of block ciphers. In Section 2.4 we consider carefully what can validly be concluded from statistical hypothesis testing.

2.1 Model

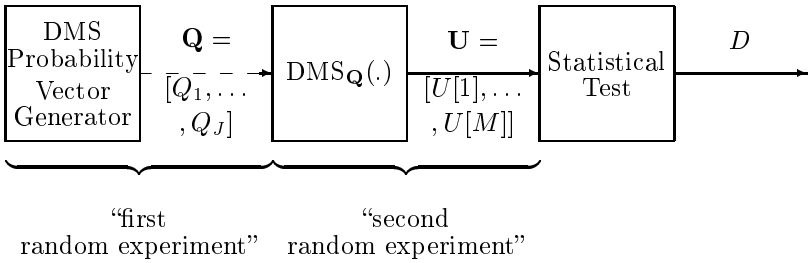

Figure 2.1: *Model for statistical hypothesis testing.*

A *probability vector* $\mathbf{p}$ in $\Re^n$ is a vector $\mathbf{p} = [p_1, p_2, \ldots, p_n]$ such that its components are nonnegative and sum to 1.

In the "first random experiment" shown in Figure 2.1, the probability vector $\mathbf{q}$ is chosen from a finite set $\mathcal{Q}$ according to some, usually unknown, probability distribution $P_{\mathbf{Q}}(\mathbf{q})$. Then, in the "second random experiment", the observation $\mathbf{u}$ is produced by generating M output digits from a discrete memoryless source whose (single-letter) probability distribution is $\mathbf{q}$, i.e., $u[m] \in \{\mu_1, \mu_2, \ldots, \mu_J\}$ and $P_{U[m]}(\mu_j) = q_j$ for $m = 1, 2, \ldots, M$. We assume that the parameters M and J, and therefore the set of possible observations $\mathcal{U}$, are finite and fixed in advance.

Based only on the observation $\mathbf{u}$, the statistical test has to decide whether the chosen probability vector $\mathbf{q}$ in the first random experiment was equal to a known probability vector $\mathbf{p}$ or not. This leads to the two hypotheses:

$$H_0: \quad \mathbf{q} = \mathbf{p} \tag{2.1}$$

$$H_1: \quad \mathbf{q} \neq \mathbf{p}. \tag{2.2}$$

In our treatment of hypothesis testing, we consider H_0 and H_1 to be events. The probabilities of these events are

$$\mathrm{P}[H_0] = P_{\mathbf{Q}}(\mathbf{p}) \tag{2.3}$$

$$\mathrm{P}[H_1] = \sum_{\mathbf{q} \in \mathcal{Q} \setminus \{\mathbf{p}\}} P_{\mathbf{Q}}(\mathbf{q}) = 1 - P_{\mathbf{Q}}(\mathbf{p}). \tag{2.4}$$

In most cases, these probabilities will be unknown, which is the almost universal assumption in hypothesis testing. For this reason, most treatments of hypothesis testing refrain from treating H_0 and H_1 as events. However, not to do so clouds the meaning of such function as $P_{\mathbf{Q}|H_1}(.)$, which is the standard notation for the probability distribution for $\mathbf{Q}$ conditioned on the occurrence of the event H_1. It seams preferable to us to avoid such notational questions by considering H_0 and H_1 to be events, albeit events with unknown probabilities. For a good discussion of this point the reader is referred to [6, pages 320–321].

We will always assume that the probability vector of the ideal source $\mathbf{p}$ is in $\mathcal{Q}$ and that $\mathbf{p}$ has no zero components. Hence every observation $\mathbf{u}$ in the set of possible observations $\mathcal{U}$ could have been generated by the ideal source. We will further always assume that, in addition to the probability vector of the ideal source $\mathbf{p}$, there is at least one other probability vector in $\mathcal{Q}$, i.e., $|\mathcal{Q}| \geq 2$, but we will *not* assume that probability vectors in $\mathcal{Q}$ (other than $\mathbf{p}$) have all non-zero components.

Every statistical test is a decomposition of the set of possible observations $\mathcal{U}$ into disjoint subsets $\mathcal{U}_0$ and $\mathcal{U}_1$ with the meaning that one accepts H_0 as true (i.e., $D = 0$) if $\mathbf{u} \in \mathcal{U}_0$ and one accepts H_1 as true (i.e., $D = 1$) if $\mathbf{u} \in \mathcal{U}_1$. We will call $\mathcal{U}_0$ and $\mathcal{U}_1$ the decision regions for the hypotheses H_0 and H_1, respectively.

There are two kinds of errors one can make:

$$\text{type I error}: \quad \alpha = P_{D|H_0}(1) = \sum_{\mathbf{u} \in \mathcal{U}_1} P_{\mathbf{U}|H_0}(\mathbf{u}) \tag{2.5}$$

$$\text{type II error}: \quad \beta = P_{D|H_1}(0) = \sum_{\mathbf{u} \in \mathcal{U}_0} P_{\mathbf{U}|H_1}(\mathbf{u}). \tag{2.6}$$

2.2 The Neyman-Pearson Statistical Test

One reasonable way to choose the decision regions $\mathcal{U}_0$ and $\mathcal{U}_1$ is implied by the following well-known theorem [27].

Theorem 2.1 (Neyman-Pearson) *For any positive real number T, let*

$$\mathcal{U}_0 = \left\{ \mathbf{u} : \mathbf{u} \in \mathcal{U} \text{ and } \frac{P_{\mathbf{U}|H_1}(\mathbf{u})}{P_{\mathbf{U}|H_0}(\mathbf{u})} \leq T \right\} \tag{2.7}$$

$$\mathcal{U}_1 = \left\{ \mathbf{u} : \mathbf{u} \in \mathcal{U} \text{ and } \frac{P_{\mathbf{U}|H_1}(\mathbf{u})}{P_{\mathbf{U}|H_0}(\mathbf{u})} > T \right\} \tag{2.8}$$

and let α and β be the probabilities of type I and type II error corresponding to this choice of decision regions. Suppose $\bar{\alpha}$ and $\bar{\beta}$ are the probabilities of type I and type II errors corresponding to some other choice of decision regions. Then $\bar{\alpha} < \alpha$ implies that $\bar{\beta} > \beta$ and, conversely, $\bar{\beta} < \beta$ implies that $\bar{\alpha} > \alpha$.

The proof of Theorem 2.1 is given on page 91.

This theorem does two things. First, it implies one reasonable way to choose the decision regions $\mathcal{U}_0$ and $\mathcal{U}_1$. Second, it shows that there exist no other choice of decision regions for which both types of error have smaller probabilities than those for this choice. In this sense, the theorem gives us a best solution to the problem.

To apply Theorem 2.1, one needs to know $P_{\mathbf{U}|H_0}(\mathbf{u})$ and $P_{\mathbf{U}|H_1}(\mathbf{u})$. Because H_0 and $\mathbf{Q} = \mathbf{p}$ are the same event, it follows that

$$P_{\mathbf{U}|H_0}(\mathbf{u}) = P_{\mathbf{U}|\mathbf{Q}}(\mathbf{u}|\mathbf{p}). \tag{2.9}$$

Hence, $P_{\mathbf{U}|H_0}(\mathbf{u})$ may always be considered to be known. To see what is required for $P_{\mathbf{U}|H_1}(\mathbf{u})$ to be known, we write

$$P_{\mathbf{U}|H_1}(\mathbf{u}) = \sum_{\mathbf{q}\in\mathcal{Q}} P_{\mathbf{U}\mathbf{Q}|H_1}(\mathbf{u},\mathbf{q}) \quad = \sum_{\mathbf{q}\in\mathcal{Q}} P_{\mathbf{U}|\mathbf{Q}H_1}(\mathbf{u}|\mathbf{q}) P_{\mathbf{Q}|H_1}(\mathbf{q}). \tag{2.10}$$

But given $\mathbf{Q} = \mathbf{q}$, $\mathbf{U}$ has no further dependence on H_1 so that

$$P_{\mathbf{U}|H_1}(\mathbf{u}) = \sum_{\mathbf{q}\in\mathcal{Q}} P_{\mathbf{U}|\mathbf{Q}}(\mathbf{u}|\mathbf{q}) P_{\mathbf{Q}|H_1}(\mathbf{q}) \tag{2.11}$$

$$= \sum_{\mathbf{q}\in\mathcal{Q}\setminus\{\mathbf{p}\}} P_{\mathbf{U}|\mathbf{Q}}(\mathbf{u}|\mathbf{q}) P_{\mathbf{Q}|H_1}(\mathbf{q}) \tag{2.12}$$

where we have used the fact that $P_{\mathbf{Q}|H_1}(\mathbf{p}) = 0$. It follows that $P_{\mathbf{U}|H_1}(\mathbf{u})$ is known when $P_{\mathbf{Q}|H_1}(\mathbf{q})$ is known for every $\mathbf{q} \in \mathcal{Q} \setminus \{\mathbf{p}\}$, which always is the case when $\mathcal{Q}$ contains only two probability vectors, $\mathbf{p}$ and $\bar{\mathbf{p}}$, since then $P_{\mathbf{Q}|H_1}(\bar{\mathbf{p}}) = 1$.

2.3 Components of a Statistical Test

We now show that, with virtually no loss of optimality, the statistical test box in Figure 2.1 can be realized as a statistical test on the "composition" of $\mathbf{U}$, rather than $\mathbf{U}$ itself, as shown in the following figure:

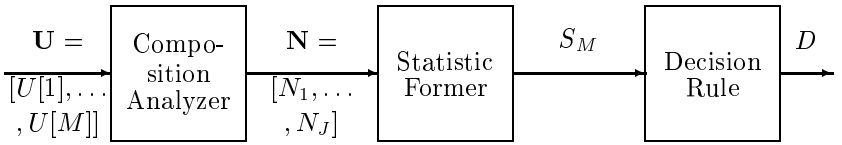

Figure 2.2: *A model for a statistical test.*

The composition of the sequence $\mathbf{u} = [u[1], u[2], \ldots, u[M]]$ with $u[m] \in \{\mu_1, \mu_2, \ldots, \mu_J\}$ for $m = 1, 2, \ldots, M$ is the vector $\mathbf{n} = [n_1, n_2, \ldots, n_J]$ where n_j is the number of occurrences of the symbol μ_j in $\mathbf{u}$ for $j = 1, 2, \ldots, J$. Note that $\sum_{j=1}^{J} n_j = M$.

In terms of the composition $\mathbf{N} = \mathbf{n} = [n_1, n_2, \ldots, n_J]$ of the observation $\mathbf{U} = \mathbf{u} = [u[1], u[2], \ldots, u[M]]$ in Figure 2.1, we have

$$P_{\mathbf{U}|\mathbf{Q}}(\mathbf{u}|\mathbf{q}) = q_1^{n_1} q_2^{n_2} \cdots q_J^{n_J} \tag{2.13}$$

as follows from the definition of a discrete memoryless source. The probability conditioned on $\mathbf{Q} = \mathbf{q}$ of a particular composition $\mathbf{N} = \mathbf{n}$ is obtained by summing $P_{\mathbf{U}|\mathbf{Q}}(\mathbf{u}|\mathbf{q})$ over all $\mathbf{u}$ with this composition, which gives

$$P_{\mathbf{N}|\mathbf{Q}}(\mathbf{n}|\mathbf{q}) = \frac{M!}{n_1! n_2! \cdots n_J!} q_1^{n_1} q_2^{n_2} \cdots q_J^{n_J} \tag{2.14}$$

$$= \frac{M!}{n_1! n_2! \cdots n_J!} P_{\mathbf{U}|\mathbf{Q}}(\mathbf{u}|\mathbf{q}) \tag{2.15}$$

where the fraction on the right is the multinomial coefficient that gives the count of the number of sequences $\mathbf{u}$ with the composition $\mathbf{n}$. Because H_0 and $\mathbf{Q} = \mathbf{p}$ are the same event, it follows that

$$P_{\mathbf{N}|H_0}(\mathbf{n}) = P_{\mathbf{N}|\mathbf{Q}}(\mathbf{n}|\mathbf{p}). \tag{2.16}$$

To determine $P_{\mathbf{N}|H_1}(\mathbf{n})$, we follow a derivation entirely similar to that which led to (2.12) to obtain

$$P_{\mathbf{N}|H_1}(\mathbf{n}) = \sum_{\mathbf{q}\in\mathcal{Q}\setminus\{\mathbf{p}\}} P_{\mathbf{N}|\mathbf{Q}}(\mathbf{n}|\mathbf{q}) P_{\mathbf{Q}|H_1}(\mathbf{q}). \tag{2.17}$$

Substituting (2.15) into (2.17) and (2.16) gives

$$\frac{P_{\mathbf{N}|H_1}(\mathbf{n})}{P_{\mathbf{N}|H_0}(\mathbf{n})} = \frac{P_{\mathbf{U}|H_1}(\mathbf{u})}{P_{\mathbf{U}|H_0}(\mathbf{u})} \tag{2.18}$$

where $\mathbf{n}$ is the composition of $\mathbf{u}$.

It follows from (2.18) that the Neyman-Pearson statistical test of Theorem 2.1 gives exactly the same result whether applied to the observation $\mathbf{U} = \mathbf{u}$ or to the "observation" $\mathbf{N} = \mathbf{n}$ of only the composition of $\mathbf{u}$. More generally, one sees from (2.15) that $P_{\mathbf{N}|\mathbf{Q}}(\mathbf{n}|\mathbf{q})$ and $P_{\mathbf{U}|\mathbf{Q}}(\mathbf{u}|\mathbf{q})$ are always proportional with the constant of proportionality independent of $\mathbf{q}$. This suggests that virtually any optimality criterion for a statistical hypothesis test will lead to the optimum statistical test giving the same result whether applied to the observation $\mathbf{U} = \mathbf{u}$ or to the observation $\mathbf{N} = \mathbf{n}$ of only the composition of $\mathbf{u}$. The only significant exception to this equivalence is the "minimax" criterion as we now explain.

The minimax criterion for a statistical test is the smallness of the maximum of the type I and type II error probability, i.e., the minimax statistical test minimizes $\max(\alpha, \beta)$.

Example: Assume the set $\mathcal{Q}$ of probability vectors contains only the two probability vectors $\mathbf{p} = [0.7, 0.3]$ and $\bar{\mathbf{p}} = [0.3, 0.7]$ and assume the produced sequence $\mathbf{u}$ has length $M = 2$. Then the probability distribution of the sequence $\mathbf{u}$ conditioned on the event H_0 and the event H_1, respectively, is as shown in Table 2.1. Also shown in Table 2.1 is the "likelihood ratio" used for the Neyman-Pearson statistical test of Theorem 2.1. Table 2.2 shows six of the sixteen statistical tests that can be defined for this case and shows for each of these six statistical tests the probability of type I error, α, and the probability of type II error, β. The statistical tests $D_1(.)$ and $D_2(.)$ in Table 2.2 are the minimax statistical tests and the statistical tests $D_3(.)$, $D_4(.)$, $D_5(.)$ and $D_6(.)$ in Table 2.2 are the Neyman-Pearson statistical tests. Note that this example does not contradict Theorem 2.1, e.g., $0.30 < 0.51$ does imply

$\mathbf{u}$	$P_{\mathbf{U}\|H_0}(\mathbf{u})$	$P_{\mathbf{U}\|H_1}(\mathbf{u})$	$\frac{P_{\mathbf{U}\|H_1}(\mathbf{u})}{P_{\mathbf{U}\|H_0}(\mathbf{u})}$
$[\mu_1, \mu_1]$	0.49	0.09	0.18
$[\mu_1, \mu_2]$	0.21	0.21	1.00
$[\mu_2, \mu_1]$	0.21	0.21	1.00
$[\mu_2, \mu_2]$	0.09	0.49	5.44

Table 2.1: *Probability distribution of the sequence* $\mathbf{u}$ *conditioned on the event* H_0 *and on the event* H_1*, respectively, and the "likelihood ratio".*

$\mathbf{u}$	$D_1(\mathbf{u})$	$D_2(\mathbf{u})$	$D_3(\mathbf{u})$	$D_4(\mathbf{u})$	$D_5(\mathbf{u})$	$D_6(\mathbf{u})$
$[\mu_1, \mu_1]$	0	0	0	0	0	1
$[\mu_1, \mu_2]$	0	1	0	0	1	1
$[\mu_2, \mu_1]$	1	0	0	0	1	1
$[\mu_2, \mu_2]$	1	1	0	1	1	1
α	0.30	0.30	0.00	0.09	0.51	1.00
β	0.30	0.30	1.00	0.51	0.09	0.00

Table 2.2: *Six statistical tests and there probability* α *of type I error and probability* β *of type II error.*

$0.30 > 0.09$. One could define a "randomized" Neyman-Pearson statistical test by choosing H_0 with probability $1/2$ whenever the sequence $\mathbf{u}$ is either equal to $[\mu_1, \mu_2]$ or equal to $[\mu_2, \mu_1]$. This "randomized" Neyman-Pearson statistical test would in fact be minimax and could be performed on the sequence $\mathbf{u}$ or on the composition $\mathbf{n}$.

It is essentially only for criteria such as the minimax criterion where a "randomized statistical test" does better than a (deterministic) statistical test that one loses some optimality (and usually then only a slight amount) when treating the composition $\mathbf{n}$ as the observation rather then the sequence $\mathbf{u}$ itself. The reason is that the fact that there are many sequences $\mathbf{u}$ with the same composition $\mathbf{n}$ allows one when observing $\mathbf{u}$ to place some of these sequences in the decision region $\mathcal{U}_0$ and the rest in $\mathcal{U}_1$; but this is equivalent to observing $\mathbf{n}$ and then choosing H_0 or H_1 according to the fraction of these conditionally equiprobable sequences that are placed in $\mathcal{U}_0$ and $\mathcal{U}_1$, respectively. For these reasons, we will hereafter consider only those statistical tests for use in Figure 2.1 that can be decomposed as shown in Figure 2.2 to operate on the composition $\mathbf{n}$ of the sequence $\mathbf{u}$ as their effective observation.

We now discuss further details of the three components of the statistical test model of Figure 2.2.

2.3.1 Composition Analyzer

When we base our decision on the composition of a sequence instead of on the sequence itself, then we will denote our decision regions by $\mathcal{N}_0$ and $\mathcal{N}_1$. This will correspond to the decision regions $\mathcal{U}_0$ and $\mathcal{U}_1$ which contains all sequences whose composition is in $\mathcal{N}_0$ and $\mathcal{N}_1$, respectively. Since $\mathcal{N}_1 = \mathcal{N} \setminus \mathcal{N}_0$, where $\mathcal{N}$ is the set of possible compositions, it is enough to specify $\mathcal{N}_0$.

2.3.2 Statistic Former

The statistic former in Figure 2.2 is a device that maps the composition $\mathbf{N}$ into a real random variable S_M, which we call the statistic for the observed sequence $\mathbf{U}$, in that manner that the decision rule is simply a threshold test on S_M, i.e., for some specified real number T, we decide on H_0 if $S_M \leq T$ and we decide on H_1 if $S_M > T$.

Likelihood Statistic

Assume that the DMS probability vector generator can choose only between two different probability vectors $\mathbf{p}$ and $\bar{\mathbf{p}}$, i.e., $\mathcal{Q} = \{\mathbf{p}, \bar{\mathbf{p}}\}$. The Neyman-Pearson statistical test of Theorem 2.1 compares the "likelihood ratio" $P_{\mathbf{U}|H_1}(\mathbf{u})/P_{\mathbf{U}|H_0}(\mathbf{u})$ to a threshold T. But this likelihood ratio can be written equivalently, according to (2.18), as $P_{\mathbf{N}|H_1}(\mathbf{n})/P_{\mathbf{N}|H_0}(\mathbf{n})$. Thus, we can realize the Neyman-Pearson statistical test by choosing the statistic S_M to be the logarithm of the likelihood ratio (where log(0) is taken to be minus infinity), i.e.,

$$s_M = \log \frac{P_{\mathbf{N}|H_1}(\mathbf{n})}{P_{\mathbf{N}|H_0}(\mathbf{n})} = \log \frac{\bar{p}_1^{n_1} \bar{p}_2^{n_2} \cdots \bar{p}_J^{n_J}}{p_1^{n_1} p_2^{n_2} \cdots p_J^{n_J}}. \tag{2.19}$$

Equivalently,

$$s_M = \sum_{\substack{j=1 \\ n_j \neq 0}}^{J} n_j \log \frac{\bar{p}_j}{p_j} \tag{2.20}$$

for the statistic of an observed length M sequence with composition $\mathbf{n}$ when $\mathbf{p}$ and $\bar{\mathbf{p}}$ are the ideal source probability vector $\mathbf{p}$ and the alternative probability vector $\bar{\mathbf{p}}$, respectively. Written as a random variable, this likelihood statistic is

$$S_M = \sum_{\substack{j=1 \\ N_j \neq 0}}^{J} N_j \log \frac{\bar{p}_j}{p_j}. \tag{2.21}$$

Kullback-Leibler Statistic

If X and $\bar{X}$ are two finite random variables taking values in the same set $\mathcal{X}$, then the *Kullback-Leibler distance* $D(P_{\bar{X}}||P_X)$ [4], from the probability distribution $P_{\bar{X}}$ to the probability distribution P_X is defined to be

$$D(P_{\bar{X}}||P_X) = \sum_{\substack{x \in \mathcal{X} \\ P_{\bar{X}}(x) \neq 0}} P_{\bar{X}}(x) \log \frac{P_{\bar{X}}(x)}{P_X(x)}. \tag{2.22}$$

We define the *Kullback-Leibler statistic* S_M for the observed length M sequence with composition $\mathbf{N}$ and ideal source probability vector $\mathbf{p}$ to be the quantity

$$S_M = \sum_{\substack{j=1 \\ N_j \neq 0}}^{J} N_j \log \frac{N_j}{M p_j}. \tag{2.23}$$

The empirical probability distribution for $\mathbf{U}$, given the observation $\mathbf{u}$ with composition $\mathbf{n}$, corresponds to the probability vector $\frac{1}{M}\mathbf{n}$, i.e., the empirical probability of μ_j is n_j/M for $j = 1, 2, \ldots, J$. The Kullback-Leibler statistic is identical to the Kullback-Leibler distance from the empirical probability distribution $P_{\mathbf{U}|\mathbf{Q}}(.|\frac{1}{M}\mathbf{n})$ for $\mathbf{U}$ to the probability distribution $P_{\mathbf{U}|\mathbf{Q}}(.|\mathbf{p})$ for $\mathbf{U}$ given the hypothesis H_0, which in turn is

identical to the Kullback-Leibler distance from the empirical probability distribution $P_{\mathbf{N}|\mathbf{Q}}(.|\frac{1}{M}\mathbf{n})$ for the composition $\mathbf{N}$ to the probability distribution $P_{\mathbf{N}|\mathbf{Q}}(.|\mathbf{p})$ for the composition $\mathbf{N}$ given the hypothesis H_0.

From the properties of the Kullback-Leibler distance it follows that $S_M \geq 0$ with equality if and only if $\frac{1}{M}\mathbf{N} = \mathbf{p}$, i.e., if and only if the observed empirical distribution coincides with the ideal source distribution.

In the sequel, we will make of no use of the Kullback-Leibler statistic. We mention it here only as an example of another statistic which could be used for the setting in Figure 2.2. The statistic we will most often use is the Pearson statistic that we next discuss.

Pearson Statistic

The *Pearson statistic* S_M [30] for the observed length M sequence with composition $\mathbf{N}$ and ideal source probability vector $\mathbf{p}$ is the quantity

$$S_M = \sum_{j=1}^{J} \frac{(N_j - Mp_j)^2}{Mp_j}. \tag{2.24}$$

Note that when $\mathbf{p} = [\frac{1}{J}, \ldots, \frac{1}{J}]$ is the uniform probability vector, then S_M is just MJ times the square of the Euclidean distance between $\frac{1}{M}\mathbf{N}$ and $\mathbf{p}$. The Pearson statistic is sometimes called the "chi-squared statistic" for reasons that Theorem 2.6 below will make evident. We now develop some of the properties of the Pearson statistic. The following theorem is well-known [3, 13, 29].

Theorem 2.2 *Let $\mathbf{p} = [p_1, p_2, \ldots, p_J]$ be a probability vector with no zero components, let $\mathbf{q} = [q_1, q_2, \ldots, q_J]$ be any probability vector, let M be a positive integer and let $\mathbf{N} = [N_1, N_2, \ldots, N_J]$ be a random vector whose probability distribution is multinomial with parameters M and $\mathbf{q}$, i.e., $P_{\mathbf{N}}(\mathbf{n}) = \frac{M!}{n_1!n_2!\cdots n_J!} q_1^{n_1} q_2^{n_2} \cdots q_J^{n_J}$ where the components of $\mathbf{n}$ are nonnegative integers that sum to M. Then the random variable*

$$S_M = \sum_{j=1}^{J} \frac{(N_j - Mp_j)^2}{Mp_j} \tag{2.25}$$

has mean

$$\mathrm{E}\left[S_M\right] = J - 1 + \delta_{11} + (M-1)\delta_{21} \tag{2.26}$$

and variance

$$\begin{aligned}\mathrm{Var}\left[S_M\right] = {} & 2(J-1) - \frac{J^2+2J-2}{M} + \frac{1}{M}\sum_{j=1}^{J}\frac{1}{p_j} \\ & + \left(8 - \frac{2J+8}{M}\right)\delta_{11} - \frac{1}{M}\delta_{11}^2 - \left(4 - \frac{4}{M}\right)\delta_{11}\delta_{21} \\ & + \frac{1}{M}\delta_{12} + \left(4M - 4J - 16 + \frac{4J+12}{M}\right)\delta_{21} \\ & - \left(4M - 10 + \frac{6}{M}\right)\delta_{21}^2 + \left(6 - \frac{6}{M}\right)\delta_{22} \\ & + \left(4M - 12 + \frac{8}{M}\right)\delta_{32},\end{aligned} \tag{2.27}$$

where

$$\delta_{11} = \sum_{j=1}^{J}\frac{q_j - p_j}{p_j}, \qquad \delta_{12} = \sum_{j=1}^{J}\frac{q_j - p_j}{p_j^2}, \tag{2.28}$$

$$\delta_{21} = \sum_{j=1}^{J}\frac{(q_j - p_j)^2}{p_j}, \qquad \delta_{22} = \sum_{j=1}^{J}\frac{(q_j - p_j)^2}{p_j^2}, \tag{2.29}$$

$$\delta_{32} = \sum_{j=1}^{J}\frac{(q_j - p_j)^3}{p_j^2}. \tag{2.30}$$

Theorem 2.2 is proven in [29, page 217].

Remark 2.3 *Theorem 2.2 implies immediately that*

$$\lim_{M\to\infty}\frac{\mathrm{E}\left[S_M\right]}{M} = \delta_{21}, \tag{2.31}$$

$$\lim_{M\to\infty}\frac{\mathrm{Var}\left[S_M\right]}{M} = 4(\delta_{21}(1-\delta_{21}) + \delta_{32}). \tag{2.32}$$

Corollary 2.4 *For the assumptions of Theorem 2.2,*

$$0 \leq S_M \leq M \left(\frac{1}{\min_{j=1}^{J} p_j} - 1 \right). \tag{2.33}$$

Corollary 2.5 *For the assumptions of Theorem 2.2, when* $\mathbf{q} = \mathbf{p}$*, then*

$$\mathrm{E}\left[S_M\right] = J - 1, \tag{2.34}$$

$$\mathrm{Var}\left[S_M\right] = 2(J-1) - \frac{J^2 + 2J - 2}{M} + \frac{1}{M} \sum_{j=1}^{J} \frac{1}{p_j}. \tag{2.35}$$

Theorem 2.2 gives the exact values for the mean and the variance of the Pearson statistic S_M, i.e., the exact values for the central moments of order one and two of the Pearson statistic S_M. What is still missing is the probability distribution function for the Pearson statistic S_M, which we now consider.

Let $X_1, X_2, \ldots, X_n$ be independent and identically distributed (i.i.d.) random variables having a normal probability distribution with zero mean and unit variance and let $\chi_n^2 = X_1^2 + X_2^2 + \ldots + X_n^2$. Then the probability distribution of the random variable χ_n^2 is called the *chi-squared probability distribution with n degrees of freedom.* It is well-known [8] that the random variable χ_n^2 has mean n, variance $2n$ and probability distribution function

$$\mathrm{P}\left[\chi_n^2 \leq \tau\right] = \begin{cases} 0 & \text{if } \tau < 0 \\ \frac{1}{2^{\frac{n}{2}}\Gamma(\frac{n}{2})} \int_0^\tau t^{\frac{n}{2}-1} e^{-\frac{t}{2}} dt & \text{if } \tau \geq 0. \end{cases} \tag{2.36}$$

The following theorem has been known for at least 100 years [30].

Theorem 2.6 *For the assumptions of Theorem 2.2, when* $\mathbf{q} = \mathbf{p}$ *and when* $J > 1$*, then for every real* τ

$$\lim_{M \to \infty} \mathrm{P}\left[S_M \leq \tau\right] = \mathrm{P}\left[\chi_{J-1}^2 \leq \tau\right] \tag{2.37}$$

where the random variable χ_n^2 *has the chi-squared probability distribution with n degrees of freedom, i.e.,* S_M *is asymptotically chi-squared distributed with* $J-1$ *degrees of freedom.*

n	$\mathrm{P}\left[\chi_n^2 \leq \tau\right]$				
	$1-10^{-6}$	$1-10^{-8}$	$1-10^{-10}$	$1-10^{-12}$	$1-10^{-14}$
1	23.93	32.84	41.82	50.84	59.90
2	27.63	36.84	46.05	55.26	64.47
3	30.66	40.13	49.54	58.92	68.27
4	33.38	43.07	52.67	62.20	71.68
5	35.89	45.79	55.56	65.24	74.85
6	38.26	48.36	58.29	68.10	77.83
7	40.52	50.81	60.90	70.84	80.68
8	42.70	53.17	63.40	73.47	83.42
9	44.81	55.45	65.82	76.01	86.06
10	46.86	57.66	68.17	78.47	88.63
11	48.87	59.82	70.46	80.87	91.13
12	50.83	61.93	72.69	83.22	93.57
13	52.75	64.00	74.89	85.52	95.96
14	54.64	66.03	77.03	87.77	98.31
15	56.49	68.03	79.15	89.98	100.61
16	58.32	69.99	81.23	92.16	102.87
17	60.13	71.93	83.27	94.30	105.10
18	61.91	73.84	85.29	96.41	107.30
19	63.68	75.73	87.29	98.50	109.46
20	65.42	77.60	89.26	100.56	111.61

Table 2.3: *Selected values of the chi-squared probability distribution function with n degrees of freedom. Example:* $\mathrm{P}\left[\chi_1^2 \leq 23.93\right] = 1 - 10^{-6}$.

Because a direct proof of this important result does not appear in the literature, we include a proof on page 92. This proof establishes the following "rule-of-thumb".

Rule-of-Thumb 2.7 *The probability distribution function of the Pearson statistic S_M is well approximated by the chi-squared probability distribution function with $J-1$ degrees of freedom if $J>1$, $\mathbf{q}=\mathbf{p}$ and if*

$$M > \frac{50}{\min_{j=1}^{J} p_j}. \tag{2.38}$$

Satisfaction of (2.38) guarantees that sufficiently many output digits from the discrete memoryless source of Figure 2.1 have been taken so that the probability distribution function for the number of occurrences of the least likely output symbol, which is a binomial distribution function with parameters M and $\min_{j=1}^{J} p_j$, is well approximated by the normal probability distribution function with same mean and variance. This approximation of the binomial distribution function by a normal probability distribution function is the only approximation invoked in the proof of Theorem 2.6.

Let $X_1, X_2, \ldots, X_n$ be independent random variables having normal probability distributions with respective means $\mathrm{E}[X_1]$, $\mathrm{E}[X_2]$, $\ldots$, $\mathrm{E}[X_n]$ and unit variance, let $\lambda = \mathrm{E}[X_1]^2 + \mathrm{E}[X_2]^2 + \ldots + \mathrm{E}[X_n]^2$ and let $\chi^2_{n,\lambda} = X_1^2 + X_2^2 + \ldots + X_n^2$. Then the probability distribution of the random variable $\chi^2_{n,\lambda}$ is called the *non-central chi-squared probability distribution with n degrees of freedom and with non-centrality parameter λ*. It is well-known [8] that the random variable $\chi^2_{n,\lambda}$ has mean $n+\lambda$, variance $2(n+2\lambda)$ and probability distribution function

$$\mathrm{P}\left[\chi^2_{n,\lambda} \le \tau\right] = \begin{cases} 0 & \text{if } \tau < 0 \\ \int_0^\tau \frac{e^{-\frac{t+\lambda}{2}} t^{\frac{n-2}{2}}}{2^{\frac{n}{2}}\Gamma(\frac{1}{2})} \sum_{r=0}^{\infty} \frac{\lambda^r t^r}{(2r)!} \frac{\Gamma(r+\frac{1}{2})}{\Gamma(r+\frac{n}{2})} dt & \text{if } \tau \ge 0. \end{cases} \tag{2.39}$$

The following theorem is well-known [7, 14, 29].

Theorem 2.8 *For the assumptions of Theorem 2.2, when $\mathbf{q} = \mathbf{p} + \frac{1}{\sqrt{M}}\mathbf{c}$, where $\mathbf{c} = [c_1, c_2, \ldots, c_J]$ is any vector in $\Re^n$ whose components sum to 0, and when $J>1$ then, for $\mathbf{p}$ and $\mathbf{c}$ fixed and M sufficiently large so that $\mathbf{q}$ is a probability vector and for every real τ,*

$$\lim_{M\to\infty} \mathrm{P}\left[S_M \le \tau\right] = \mathrm{P}\left[\chi^2_{J-1,\sum_{j=1}^{J} c_j^2/p_j} \le \tau\right] \tag{2.40}$$

where the random variable $\chi^2_{n,\lambda}$ has the non-central chi-squared probability distribution with n degrees of freedom and with non-centrality parameter λ, i.e., S_M is asymptotically non-central chi-squared distributed with $J-1$ degrees of freedom and with non-centrality parameter $\sum_{j=1}^{J} c_j^2/p_j$.

Theorem 2.8 is proven in [29, page 216].

Note that $\sum_{j=1}^{J} c_j^2/p_j = M\delta_{21}$, where δ_{21} is defined in (2.29).

We will write $\Phi(\tau)$ to denote the probability distribution function for a normal random variable with zero mean and unit variance, i.e.,

$$\Phi(\tau) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\tau} e^{-t^2/2} dt. \tag{2.41}$$

	$\Phi(\tau)$				
	10^{-6}	10^{-8}	10^{-10}	10^{-12}	10^{-14}
τ	-4.753	-5.612	-6.361	-7.034	-7.651

Table 2.4: *Selected values of the normal probability distribution function with zero mean and unit variance. Example:* $\Phi(-4.753) = 10^{-6}$.

Theorem 2.9 *For the assumptions of Theorem 2.2, when $\mathbf{q}$ has no zero components and $\mathbf{q} \ne \mathbf{p}$, then for every real τ*

$$\lim_{M\to\infty} \mathrm{P}\left[S_M \le \mathrm{E}[S_M] + \tau\sqrt{\mathrm{Var}[S_M]}\right] = \Phi(\tau), \tag{2.42}$$

i.e., $\frac{S_M - \mathrm{E}[S_M]}{\sqrt{\mathrm{Var}[S_M]}}$ is asymptotically normally distributed with zero mean and unit variance.

Theorem 2.9 is proven in [3].

Corollary 2.10 *For the assumptions of Theorem 2.2, when* $\mathbf{q}$ *is a probability vector whose only non-zero components are the components* q_1, q_2, ..., $q_{J'}$ *and when* $2 \leq J' < J$, *then for every real* τ

$$\mathrm{P}\left[S_M \leq \tau\right] = \mathrm{P}\left[S'_M \leq \tau\rho - M(1-\rho)\right] \tag{2.43}$$

where

$$S'_M = \sum_{j=1}^{J'} \frac{(N_j - Mp'_j)^2}{Mp'_j}, \qquad \rho = \sum_{j=1}^{J'} p_j, \qquad p'_j = \frac{p_j}{\rho}. \tag{2.44}$$

The proof of Corollary 2.10 is given on page 97.

Note that $\mathbf{p}' = [p'_1, p'_2, \ldots, p'_{J'}]$ and $\mathbf{q}' = [q_1, q_2, \ldots, q_{J'}]$ are probability vectors with no zero components and that $\mathbf{N}' = [N_1, N_2, \ldots, N_{J'}]$ is a random vector whose probability distribution is multinomial with parameters M and $\mathbf{q}'$. Therefore Theorem 2.2 applies to the random variable S'_M.

Corollary 2.11 *For the assumptions of Theorem 2.2, when* $\mathbf{q}$ *is a probability vector with a single non-zero component, say* $q_k = 1$, *then*

$$\mathrm{P}\left[S_M \leq \tau\right] = \begin{cases} 0 & \text{if } \tau < M\frac{1-p_k}{p_k} \\ 1 & \text{if } \tau \geq M\frac{1-p_k}{p_k}. \end{cases} \tag{2.45}$$

Proof: For this case, the random variable S_M of (2.25) reduces to the constant $M\frac{1-p_k}{p_k}$ because the discrete memoryless source with parameter $\mathbf{q}$ will emit only the k-th letter of its output alphabet, i.e., $N_k = M$ and $N_j = 0$ for $j \neq k$. □

The case where the probability vector $\mathbf{q}$ has zero components can result in a "strange" probability distribution for the Pearson statistic S_M. For instance, suppose that we have chosen M large enough so that (2.38) is satisfied. Then for a probability vector $\mathbf{q}$ with some components equal to zero but no component equal to one and with $\mathbf{q}' = \mathbf{p}'$, where $\mathbf{q}'$ and $\mathbf{p}'$ are as defined after Corollary 2.10, Rule-of-Thumb 2.7 implies that the probability distribution function of the Pearson statistic S_M is well approximated by a shifted version of the chi-squared probability distribution function with $J'-1$ degrees of freedom. The approximation

is good since satisfaction of (2.38) implies that $M > 50/\min_{j=1}^{J} p'_j$. This probability distribution function for the Pearson statistic S_M is much different from a normal probability distribution function, no matter how large we choose M, even if $\mathbf{q}$ is considerably different from $\mathbf{p}$.

Theorem 2.12 *For the assumptions of Theorem 2.2, when* $\mathbf{q}$ *is a probability vector with at least one zero component, then*

$$\mathrm{P}\left[S_M \leq \tau\right] = 0 \quad \textit{for} \quad \tau < M\frac{\min_{j=1}^{J} p_j}{1 - \min_{j=1}^{J} p_j}. \tag{2.46}$$

Proof: If $\mathbf{q}$ has a single non-zero component, then (2.46) follows from Corollary 2.11 because $(1-p_k)/p_k \geq (1 - \max_{j=1}^{J} p_j)/\max_{j=1}^{J} p_j \geq \min_{j=1}^{J} p_j/(1-\min_{j=1}^{J} p_j)$. Otherwise, (2.46) follows from Corollary 2.10 as we now show. Since $S'_M \geq 0$ we have $\mathrm{P}\left[S_M \leq \tau\right] = 0$ for $\tau < M(1-\rho)/\rho$ where ρ is defined in (2.44). But ρ is bounded by $\min_{j=1}^{J} p_j < \rho \leq 1 - \min_{j=1}^{J} p_j$. Therefore $(1-\rho)/\rho \geq \min_{j=1}^{J} p_j/(1 - \min_{j=1}^{J} p_j)$. □

By choosing M large enough for a given threshold T so that

$$M > T\left(\frac{1}{\min_{j=1}^{J} p_j} - 1\right), \tag{2.47}$$

we get $\mathrm{P}\left[S_M \leq T\right] = 0$ for all probability vectors $\mathbf{q}$ with zero components, i.e., if in the first random experiment of Figure 2.1 a probability vector $\mathbf{q}$ with zero components has been chosen, then the probability that we make a wrong decision will be zero. It follows that we can restrict our attention to the non-pathological case where all components of the probability vector $\mathbf{q}$ are positive. For this case we can well approximate the probability distribution function of the Pearson statistic S_M for large $M\delta_{21}$ by a normal probability distribution function with mean $\mathrm{E}\left[S_M\right]$ and variance $\mathrm{Var}\left[S_M\right]$.

An alternative to approximating the probability distribution function of the Pearson statistic S_M by a standard probability distribution function is to make use of Chebychev's Inequality to obtain bounds on this probability distribution function.

Theorem 2.13 *Let X be a finite random variable with mean $\mathrm{E}[X]$. Then for any positive integer n and for any $\tau \neq \mathrm{E}[X]$*

$$\mathrm{P}[X \leq \tau] \geq 1 - \frac{\mathrm{E}\left[(X - \mathrm{E}[X])^{2n}\right]}{(\tau - \mathrm{E}[X])^{2n}} \quad \textit{for} \quad \tau > \mathrm{E}[X], \tag{2.48}$$

$$\mathrm{P}[X \leq \tau] \leq \frac{\mathrm{E}\left[(X - \mathrm{E}[X])^{2n}\right]}{(\mathrm{E}[X] - \tau)^{2n}} \quad \textit{for} \quad \tau < \mathrm{E}[X]. \tag{2.49}$$

The proof of Theorem 2.13 is given on page 99.

Theorem 2.2 gives $\mathrm{E}[S_M]$ and $\mathrm{E}\left[(S_M - \mathrm{E}[S_M])^2\right] = \mathrm{Var}[S_M]$, which suffice for computing the bounds on $\mathrm{P}[S_M \leq \tau]$ in Theorem 2.13 for the simplest case when $n = 1$. To compute the bounds for $n > 1$, one needs to know the central moment of S_M of even order greater than 2. In [13], a method is presented for determining all moments of the Pearson statistic. With the aid of this method, at least in principle, one can compute the bounds on $\mathrm{P}[S_M \leq \tau]$ in Theorem 2.13 for any n.

2.3.3 Decision Rule

The decision rule to be used in Figure 2.2 is simple: compare the statistic S_M with a threshold T, which is fixed in advance, and check whether S_M is greater than T or not, i.e.,

$$D = \begin{cases} 0 & \text{if } S_M \leq T \\ 1 & \text{if } S_M > T. \end{cases} \tag{2.50}$$

The threshold is chosen according to the probability of type I error α one is willing to accept. In the experiments that we report on later, we chose T to produce an α on the order of 10^{-10}.

2.4 Interpreting the Result of Statistical Hypothesis Testing

The result of statistical hypothesis testing as shown in Figure 2.1 is the binary decision D. The result $D = 1$ means that hypothesis H_1 is accepted, i.e., we decide that the actual probability vector $\mathbf{q}$ of the source is not the ideal source probability vector $\mathbf{p}$. The probability that we make an error in this case is the probability of type I error, α, a value that we can choose to be very small by choosing the threshold T large enough. We say that we are $(1 - \alpha)$-certain that the actual probability vector of the source is different from the ideal source probability vector. This part of the interpretation is easy.

Not so easy is the interpretation of the result $D = 0$. A straightforward interpretation of $D = 0$ is that we accept hypothesis H_0, i.e., we decide that the actual probability vector $\mathbf{q}$ of the source is the ideal source probability vector $\mathbf{p}$. The probability that we make an error in this case is the probability of type II error, β. Since we generally do not know the probability distribution $P_{\mathbf{Q}|H_1}(\mathbf{q})$, except in the unrealistic case when $|\mathcal{Q}| = 2$, we cannot compute the probability of type II error β as is clear from (2.6). Thus, we cannot attribute a "confidence" value to our decision. One alternative interpretation would be to to assume a "worst case" probability distribution $P_{\mathbf{Q}|H_1}(\mathbf{q})$ and to compute an upper bound on the probability of type II error β. But if there is probability vector in the set $\mathcal{Q}$ which is only slightly different from the ideal source probability vector, as will be the case in most realistic situations, then this upper bound on the probability of type II error will be virtually 1 and therefore of no help.

To obviate these difficulties in the interpretation of the result $D = 0$, we introduce the notation of the probability of type II error conditioned on the event that a particular probability vector $\mathbf{q}$, $\mathbf{q} \neq \mathbf{p}$, has been chosen in the first random experiment, which we denote by

$$\beta(\mathbf{q}) = P_{D|\mathbf{Q}}(0|\mathbf{q}). \tag{2.51}$$

Then we choose a β^* on the order of 10^{-10} and determine the set of probability vectors $\mathcal{Q}_1$, that contains all probability vectors $\mathbf{q}$ which our statistical test will detect with error probability at most β^* to be different from the ideal source probability vector $\mathbf{p}$, i.e.,

$$\mathcal{Q}_1 = \{\mathbf{q} : \mathbf{q} \in \mathcal{Q} \setminus \{\mathbf{p}\} \text{ and } \beta(\mathbf{q}) \leq \beta^*\}. \tag{2.52}$$

The greater we choose M, the larger will be the set $\mathcal{Q}_1$. Because we have demanded that $\mathcal{Q}$ be a finite set, $\mathcal{Q}_1 = \mathcal{Q} \setminus \{\mathbf{p}\}$ will hold for all sufficiently large M, i.e., we will, at least in principle, detect any $\mathbf{q}$ that

is different from $\mathbf{p}$. On the other hand if M is too small, then the set $\mathcal{Q}_1$ might be empty.

What we can really conclude from the result $D = 0$ is that we are $(1 - \beta^*)$-certain that the actual probability vector of the source is not in the set $\mathcal{Q}_1$, i.e., we are $(1 - \beta^*)$-certain that the actual probability vector of the source is in the set $\mathcal{Q}_0$, where

$$\mathcal{Q}_0 = \mathcal{Q} \setminus \mathcal{Q}_1. \tag{2.53}$$

We summarize the steps in our approach to statistical hypothesis testing. In our statistical testing of block ciphers, we will always proceed with statistical hypothesis testing in the following manner.

- We choose the probability of type I error α (on the order of 10^{-10}) that we are willing to accept.

- Assuming that S_M is chi-squared distributed with $J - 1$ degrees of freedom (where J is the output alphabet size of the discrete memoryless source), we compute the threshold T as the smallest τ such that

$$\mathrm{P}\left[S_M > \tau | \mathbf{Q} = \mathbf{p}\right] \leq \alpha, \tag{2.54}$$

  making use of the tabulation of the chi-squared distribution given in Table 2.3.

- We choose M, the number of digits from the discrete memoryless source of Figure 2.1, to satisfy (2.38). This ensures that the probability distribution function of the Pearson statistic S_M for $\mathbf{q} = \mathbf{p}$ is well approximated by the chi-squared probability distribution function with $J-1$ degrees of freedom. If necessary, we increase M further so that (2.47) is satisfied. This ensures that, for all $\mathbf{q}$ with zero components, we always make the correct decision and that, for all $\mathbf{q}$ with no zero components, the probability distribution function of the Pearson statistic S_M is for large $M\delta_{21}$ well approximated by the normal probability distribution function with mean $\mathrm{E}\left[S_M\right]$ and variance $\mathrm{Var}\left[S_M\right]$ given by (2.26) and (2.27), respectively. In order obtain more reliable results we can increase M still further.

- We choose a β^* (on the order of 10^{-10}) and construct the set of probability vectors $\mathcal{Q}_1$ according to (2.52), which we approximate as

$$\begin{aligned}\mathcal{Q}_1 \approx &\{\mathbf{q} : \mathbf{q} \in \mathcal{Q} \text{ and } \mathbf{q} \text{ has zero components}\} \cup \\ &\left\{\mathbf{q} : \mathbf{q} \in \mathcal{Q} \setminus \{\mathbf{p}\} \text{ and } \mathrm{E}\left[S_M\right] + \Phi^{-1}(\beta^*)\sqrt{\mathrm{Var}\left[S_M\right]} \geq T\right\}\end{aligned} \tag{2.55}$$

  where $\Phi^{-1}(\beta^*)$ can be taken from Table 2.4, which is a tabulation of the normal probability distribution function. We thus obtain the set of all probability vectors which our statistical test can with high probability detect to be different from the ideal source probability vector.

- We perform the statistical hypothesis test and interpret the result of the test in the following manner: If $D = 1$, we say that we are $(1 - \alpha)$-certain that $\mathbf{q} \neq \mathbf{p}$. If $D = 0$, we say that we are $(1 - \beta^*)$-certain that $\mathbf{q} \notin \mathcal{Q}_1$, i.e., we are $(1 - \beta^*)$-certain that $\mathbf{q} \in \mathcal{Q}_0$, where $\mathcal{Q}_0 = \mathcal{Q} \setminus \mathcal{Q}_1$.

Chapter 3

Algorithmic Attacks on Block Ciphers

In this chapter we give precise definitions of a block cipher and of a special block cipher that we will call the complete block cipher. We show how a cryptanalyst can use algorithms of a certain kind to attack a block cipher and we establish when a cryptanalyst cannot break a given block cipher. From these considerations, we formulate two basic problems that a cryptanalyst can attempt to solve. One of the two basic problems is to find an algorithm that is *distinguishing* for a given block cipher. The other basic problem is to find an algorithm that is *key-subset distinguishing* for a given block cipher and for a given decomposition of the key space. We show that if a cryptanalyst cannot solve at least one of these two basic problems for a given block cipher, then he cannot break this block cipher.

3.1 Block Ciphers

Definition 3.1 *A (binary, non-expanding)* block cipher with block length N *is a bivariate function*

$$e : \{0,1\}^N \times \mathcal{Z}_e \to \{0,1\}^N : (x,z) \mapsto e_z(x), \tag{3.1}$$

where e_z is invertible for every z. The quantities x, z and $e_z(x)$ are the plaintext block*, the* secret key *and the* ciphertext block*, respectively. The functions e_z and e_z^{-1} are called the* encryption function for the secret key z *and the* decryption function for the secret key z*, respectively. The set $\mathcal{Z}_e$ is called the* key space*. The* key length *is* $\log_2(|\mathcal{Z}_e|)$ *bits. To be a* practical block cipher*, e_z and e_z^{-1} must be easy to compute for every z. We will always assume that the secret key is chosen uniformly at random from the key space $\mathcal{Z}_e$.*

We will write $\mathcal{F}_N$ to denote the set of all invertible functions $\{0,1\}^N \to \{0,1\}^N$. There are $|\mathcal{F}_N| = 2^N!$ functions in $\mathcal{F}_N$.

Definition 3.2 *The* complete block cipher with block length N *is the block cipher*

$$\breve{e} : \{0,1\}^N \times \mathcal{Z}_{\breve{e}} \to \{0,1\}^N : (x,z) \mapsto \breve{e}_z(x), \tag{3.2}$$

where for each invertible function f in $\mathcal{F}_N$ there is exactly one secret key z in $\mathcal{Z}_{\breve{e}}$ such that $\breve{e}_z = f$.

Note that the key space of the complete block cipher with block length N has cardinality $2^N!$ and hence its key length is $\log_2(2^N!) \approx 2^N(N - 1.44)$ [8, Stirling's Formula], which is astronomically large for practical values of N, say $N = 64$ or $N = 128$. Because of our assumption that the secret key is chosen uniformly at random from the key space $\mathcal{Z}_{\breve{e}}$, the complete block cipher of length N is equivalent to what is often called a "random permutation of $\{0,1\}^N$".

3.2 Algorithmic Attacks on Block Ciphers

Following Kerckhoffs's principle [11], we assume that the cryptanalyst knows the entire mechanism of encipherment except for the value of the secret key. We further assume that the cryptanalyst has access to a black box containing an encryption-function/decryption-function pair for the actual secret key. The cryptanalyst is allowed to make a chosen-text attack on the block cipher, i.e., in the course of his attack he may ask the black box two kind of questions:

- for a plaintext block chosen by the cryptanalyst, what is the corresponding ciphertext block?
- for a ciphertext block chosen by the cryptanalyst, what is the corresponding plaintext block?

In both cases the black box's answer provides a plaintext-block/ciphertext-block pair, i.e., an entry in the function table of the encryption function for the actual secret key. We now distinguish between four problems a cryptanalyst might try to solve:

- decrypt a ciphertext block chosen uniformly at random, without asking the black box to decrypt it.
- encrypt a plaintext block chosen uniformly at random, without asking the black box to encrypt it.
- find an additional entry in the function table of the encryption function beyond those learned by queries to the black box.
- find the secret key.

Later we will consider more precisely what it means to say that one of these problems is solved.

Figure 3.1 illustrates a probabilistic algorithm for analyzing an invertible function f. We will soon see how the cryptanalyst could use such an algorithm to solve any one of the above four problems. The black box is some device (or oracle) that can compute an invertible function f and its inverse f^{-1}. The deterministic algorithm is allowed to query either of these two functions by submitting an argument, thus obtaining an entry in the function table of the invertible function f. In addition, the deterministic algorithm has access to a random table that provides all the "randomness" in the probabilistic algorithm. The random table is loaded initially with a random string R chosen according

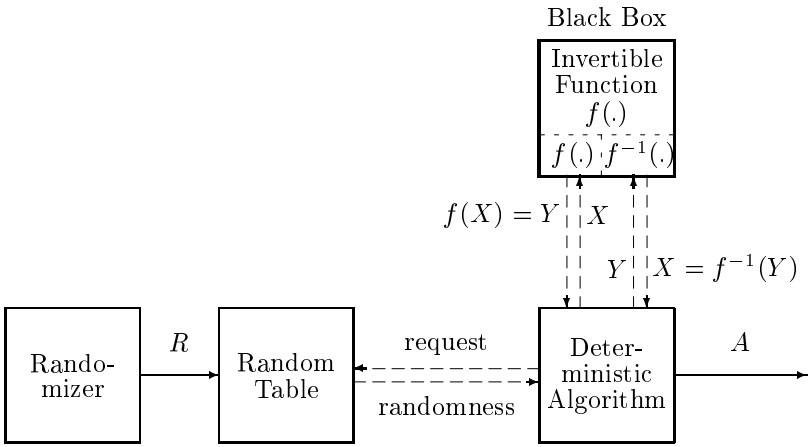

Figure 3.1: *Model for a probabilistic algorithm for analyzing an invertible function f.*

to a specified probability distribution P_R. The deterministic algorithm may query the random table to obtain values from this table. At the end of the execution, the deterministic algorithm outputs its analysis A of the invertible function f.

Definition 3.3 *A* probabilistic algorithm $\mathbb{A}$ for analyzing an invertible function f in $\mathcal{F}_N$ *is an algorithm with the structure shown in Figure 3.1. The input to the algorithm is a "black box" for the invertible function f which, when queried with the input X, returns $f(X)$, and, when queried with the input Y, returns $f^{-1}(Y)$. The randomizer chooses a random string R from the finite set $\mathcal{R}$ according to a specified probability distribution P_R and loads the random table with this string. We will always assume that every r in $\mathcal{R}$ has non-zero probability $P_R(r)$. The deterministic algorithm makes queries to the black box and to the random table, receives the results of these queries, and determines whether to stop or to continue with queries. When this algorithm stops, it outputs its analysis A of the invertible function f.*

Figure 3.2 shows how a probabilistic algorithm for analyzing an invertible function can be applied to a randomly chosen encryption func-

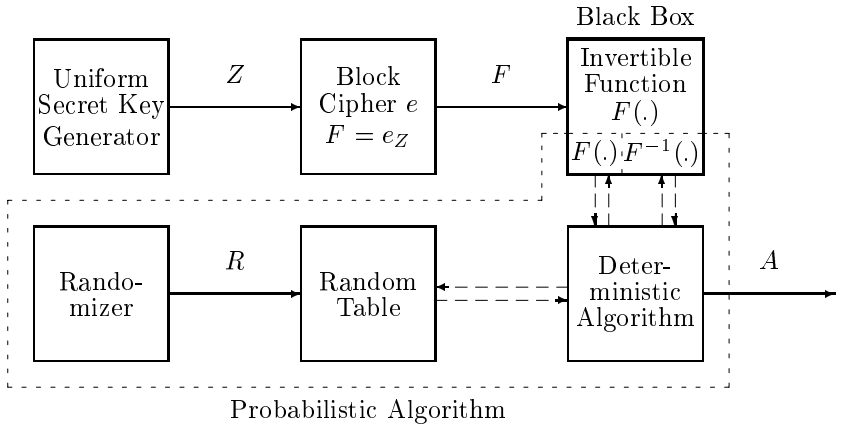

Figure 3.2: *Model for analyzing a randomly chosen encryption function of a block cipher e by a probabilistic algorithm for analyzing an invertible function.*

tion of a block cipher e. The random variable F and the random string R are the inputs to the deterministic algorithm and the random variable A is the output. The probability distribution of the random string R is specified by the probabilistic algorithm for analyzing an invertible function. The probability distribution of the random variable F is specified by the block cipher e whose encryption function for the chosen secret key Z is supplied to the probabilistic algorithm for analyzing an invertible function.

Definition 3.4 A probabilistic algorithm $\mathbb{A}$ for analyzing an invertible function is computationally feasible *if its deterministic algorithm is computationally feasible for every invertible function f in $\mathcal{F}_N$ and for every random string r in $\mathcal{R}$, under the assumptions that learning an entry in the function table of the invertible function f costs a fixed small amount of time (say, the time required to send a request to the black box and to receive its answer) and that accessing the random table is instantaneous.*

Note that the encryption and decryption time is not charged against the probabilistic algorithm for analyzing an invertible function, i.e., the

black box is really being treated as an oracle. We did so for two reasons. First, a cryptanalyst might be able to get access to several black boxes with the same invertible function in it such that for this cryptanalyst the encryption and decryption time might be neglected. Second, a slow block cipher should have no advantage over a fast block cipher just because it is harder to get the information out of him.

We now return to the four problems a cryptanalyst might try to solve.

To treat the first three of the four problems mentioned above, we need to specify an underlying random experiment. For a specified block cipher e with block length N and for the complete block cipher $\breve{e}$ with block length N, we take this underlying random experiment to be that of making a (not necessarily equally likely) random choice of either e or $\breve{e}$ and then choosing a secret key uniformly at random from the key space of the chosen block cipher. Let E_e and $E_{\breve{e}}$ denote the events that e or $\breve{e}$, respectively, is drawn in this experiment. We assume that the probabilities $\mathrm{P}[E_e]$ and $\mathrm{P}[E_{\breve{e}}] = 1 - \mathrm{P}[E_e]$ are unknown. Letting the random variable F be the invertible function realized by the chosen block cipher and the chosen secret key, we have

$$P_{F|E_e}(f) = \frac{|\{z : z \in \mathcal{Z}_e \text{ and } e_z = f\}|}{|\mathcal{Z}_e|} \tag{3.3}$$

and

$$P_{F|E_{\breve{e}}}(f) = \frac{1}{2^N!}. \tag{3.4}$$

Let $C_{\max}(\mathbb{A})$ denote the maximum over f in $\mathcal{F}_N$ of the maximum number of distinct pairs $[x, y]$ in the function table of f obtained by calls to f or f^{-1} when the probabilistic algorithm $\mathbb{A}$ is applied to f.

Let $\mathbb{A}_1$ be a probabilistic algorithm for analyzing an invertible function that chooses a ciphertext block Y_1 uniformly at random, performs a randomized analysis in which it computes a prediction $\hat{X}_1$ of $X_1 = F^{-1}(Y_1)$ without asking the black box to compute F^{-1} for the argument Y_1, and then outputs its analysis $A = [\hat{X}_1, Y_1]$. We will say that $\mathbb{A}_1$ *solves for the block cipher e the problem of decrypting a ciphertext block chosen uniformly at random without asking the black box to decrypt it* if

$$\mathrm{P}\left[\hat{X}_1 = F^{-1}(Y_1)|E_e\right] \gg \mathrm{P}\left[\hat{X}_1 = F^{-1}(Y_1)|E_{\breve{e}}\right], \tag{3.5}$$

i.e., if the probability that the plaintext prediction is correct is substantially greater when given that the block cipher e was chosen than when given that the complete block cipher $\breve{e}$ was chosen. The probability that the plaintext prediction is correct when given that the complete block cipher $\breve{e}$ was chosen is upper bounded by

$$\mathrm{P}\left[\hat{X}_1 = F^{-1}(Y_1)|E_{\breve{e}}\right] \leq \frac{C_{\max}(\mathbb{A}_1) + 1}{2^N}. \tag{3.6}$$

Proof of (3.6): During its execution the probabilistic algorithm for analyzing an invertible function f learned some entries $[X, Y]$ in the function table of the invertible function f. Let $\mathcal{X}$ denote the set of arguments X of the learned entries and let $\mathcal{Y}$ denote the set of dependent variables Y of the learned entries. Because the function f is invertible, $|\mathcal{X}| = |\mathcal{Y}|$. Expanding the probability that the plaintext prediction is correct when given that the complete block cipher $\breve{e}$ was chosen gives

$$\begin{aligned}\mathrm{P}\left[\hat{X}_1 = F^{-1}(Y_1)|E_{\breve{e}}\right] = {} & \overbrace{\mathrm{P}\left[\hat{X}_1 = F^{-1}(Y_1)|Y_1 \in \mathcal{Y}, E_{\breve{e}}\right]}^{\leq 1} \overbrace{\mathrm{P}\left[Y_1 \in \mathcal{Y}|E_{\breve{e}}\right]}^{\rho\cdot|\mathcal{Y}|/2^N} \\ & + \underbrace{\mathrm{P}\left[\hat{X}_1 = F^{-1}(Y_1)|Y_1 \notin \mathcal{Y}, E_{\breve{e}}\right]}_{\leq 1/(2^N - |\mathcal{Y}|)} \underbrace{\mathrm{P}\left[Y_1 \notin \mathcal{Y}|E_{\breve{e}}\right]}_{1-\rho\cdot|\mathcal{Y}|/2^N}.\end{aligned} \tag{3.7}$$

$\mathrm{P}[Y_1 \in \mathcal{Y}|E_{\breve{e}}]$ is zero if all queries to the black box are queries to the function f^{-1}, it is $|\mathcal{Y}|/2^N$ if all queries to the black box are queries to the function f and it is $\rho\cdot|\mathcal{Y}|/2^N$ for some ρ, $0 \leq \rho \leq 1$, in general. If $Y_1 \notin \mathcal{Y}$ and if the complete block cipher $\breve{e}$ was chosen, then any $\hat{X}_1$ in $\{0,1\}^N \setminus \mathcal{X}$ has probability $1/(2^N - |\mathcal{X}|)$ to be correct and any $\hat{X}_1$ in $\mathcal{X}$ has probability 0 to be correct. Therefore $\mathrm{P}\left[\hat{X}_1 = F^{-1}(Y_1)|Y_1 \notin \mathcal{Y}, E_{\breve{e}}\right] \leq 1/(2^N - |\mathcal{X}|)$. Hence we get

$$\mathrm{P}\left[\hat{X}_1 = F^{-1}(Y_1)|E_{\breve{e}}\right] \leq \rho\frac{|\mathcal{Y}|}{2^N} + \frac{1 - \rho\frac{|\mathcal{Y}|}{2^N}}{2^N - |\mathcal{Y}|}. \tag{3.8}$$

The maximum on the right side is obtained when ρ is 1 such that we get

$$\mathrm{P}\left[\hat{X}_1 = F^{-1}(Y_1)|E_{\breve{e}}\right] \leq \frac{|\mathcal{Y}| + 1}{2^N}. \tag{3.9}$$

And finally with $|\mathcal{Y}| \leq C_{\max}(\mathbb{A}_1)$, which hold according to the definition of $C_{\max}(.)$, we obtain (3.6). □

Let $\mathbb{A}_2$ be a probabilistic algorithm for analyzing an invertible function that chooses a plaintext block X_2 uniformly at random, performs a randomized analysis in which it computes a prediction $\hat{Y}_2$ of $Y_2 = F(X_2)$ without asking the black box to compute F for the argument X_2, and then outputs its analysis $A = [X_2, \hat{Y}_2]$. We will say that $\mathbb{A}_2$ *solves for the block cipher e the problem of encrypting a plaintext block chosen uniformly at random without asking the black box to encrypt it* if

$$\mathrm{P}\left[\hat{Y}_2 = F(X_2)|E_e\right] \gg \mathrm{P}\left[\hat{Y}_2 = F(X_2)|E_{\check{e}}\right], \tag{3.10}$$

i.e., if the probability that the ciphertext prediction is correct is substantially greater when given that the block cipher e was chosen than when given that the complete block cipher $\check{e}$ was chosen. The probability that the ciphertext prediction is correct when given that the complete block cipher $\check{e}$ was chosen is upper bounded by

$$\mathrm{P}\left[\hat{Y}_2 = F(X_2)|E_{\check{e}}\right] \leq \frac{C_{\max}(\mathbb{A}_2) + 1}{2^N}. \tag{3.11}$$

The proof of (3.11) is entirely similar to the proof of (3.6).

Let $\mathbb{A}_3$ be a probabilistic algorithm for analyzing an invertible function that outputs a prediction $A = [\hat{X}_3, \hat{Y}_3]$ of an entry in the function table of F that is different from any entry in the function table of F obtained by queries to the black box. We will say that $\mathbb{A}_3$ *solves for the block cipher e the problem of finding an additional entry in the function table of the encryption function* if

$$\mathrm{P}\left[\hat{Y}_3 = F(\hat{X}_3)|E_e\right] \gg \mathrm{P}\left[\hat{Y}_3 = F(\hat{X}_3)|E_{\check{e}}\right], \tag{3.12}$$

i.e., if the probability that the plaintext/ciphertext pair prediction is correct is substantially greater when given that the block cipher e was chosen than when given that the complete block cipher $\check{e}$ was chosen. The probability that the plaintext/ciphertext pair prediction is correct when given that the complete block cipher $\check{e}$ was chosen is upper bounded by

$$\mathrm{P}\left[\hat{Y}_3 = F(\hat{X}_3)|E_{\check{e}}\right] \leq \frac{1}{2^N - C_{\max}(\mathbb{A}_3)}. \tag{3.13}$$

Proof of (3.13): During its execution the probabilistic algorithm for analyzing an invertible function f learned some entries $[X, Y]$ in the function table of the invertible function f. Let $\mathcal{X}$ denote the set of arguments X of the learned entries and let $\mathcal{Y}$ denote the set of dependent variables Y of the learned entries. Because the function f is invertible, $|\mathcal{X}| = |\mathcal{Y}|$. Since the complete block cipher was chosen any pair $[\hat{X}_3, \hat{Y}_3]$ with $\hat{X}_3 \notin \mathcal{X}$ and $\hat{Y}_3 \notin \mathcal{Y}$ has probability $1/(2^N - |\mathcal{X}|)$ to be an entry in the function table of f and any other pair $[\hat{X}_3, \hat{Y}_3]$ has probability 0 to be an entry in the function table of f. Therefore $\mathrm{P}\left[\hat{Y}_3 = F(\hat{X}_3)|E_{\check{e}}\right] \leq 1/(2^N - |\mathcal{X}|)$. With $|\mathcal{X}| \leq C_{\max}(\mathbb{A}_1)$, which hold according to the definition of $C_{\max}(.)$, we obtain (3.13). □

To treat the fourth and last of the four problems mentioned above, we change to a different underlying random experiment. For a specified block cipher e with block length N, we take this underlying random experiment to be that of choosing a secret key Z for e uniformly at random from the key space $\mathcal{Z}_e$. For this random experiment, let $\mathbb{A}_4$ be a probabilistic algorithm for analyzing an invertible function that outputs a prediction $A = \hat{Z}$ of the actual secret key Z. We will say that $\mathbb{A}_4$ *solves for the block cipher e the problem of finding the secret key* if

$$\mathrm{P}\left[\hat{Z} = Z\right] \gg \frac{1}{|\mathcal{Z}_e|}, \tag{3.14}$$

i.e., if the probability that the secret key prediction is correct is substantially greater than for a random prediction of the secret key.

Definition 3.5 *A cryptanalyst* cannot break *the block cipher e if this cryptanalyst knows no computationally feasible probabilistic algorithm for analyzing an invertible function that (1) solves for the block cipher e the problem of decrypting a ciphertext block chosen uniformly at random without asking the black box to decrypt it, or that (2) solves for the block cipher e the problem of encrypting a plaintext block chosen uniformly at random without asking the black box to encrypt it, or that (3) solves for the block cipher e the problem of finding an additional entry in the function table of the encryption function, or that (4) solves for the block cipher e the problem of finding the secret key. The cryptanalyst* can break *the block cipher e if he knows a computationally feasible probabilistic algorithm for analyzing an invertible function that solves for the block cipher e at least one of these four problems.*

Example: The complete block cipher with block length $N = 2$ has a key space with cardinality $2^2! = 24$. A cryptanalyst could design a probabilistic algorithm for analyzing an invertible function that asks the black box to see 3 of the 4 entries in the function table of the encryption function for the actual secret key and then efficiently computes the actual secret key. This cryptanalyst would know a computationally feasible probabilistic algorithm for analyzing an invertible function that solves for the complete block cipher with block length 2 the problem of finding the secret key and could therefore break the complete block cipher with block length 2. Note that it is impossible to solve any of the first three of the four problems mentioned above for any complete block cipher.

Based on this notion of breaking, we start now to build a bridge to the next chapter, which is about statistical tests for block ciphers. We now formulate a sufficient, but not necessary, condition for a cryptanalyst to be unable to break a given block cipher e. This sufficient condition can be tested directly by statistical tests as will be shown in the next chapter.

Definition 3.6 *A probabilistic algorithm for analyzing an invertible function is* distinguishing for the block cipher e *if (in the random experiment described before (3.3)) it outputs a binary decision $A = D$, $D \in \{0, 1\}$ such that*

$$\mathrm{P}\left[D = 1 | E_e\right] \gg \mathrm{P}\left[D = 1 | E_{\check{e}}\right], \tag{3.15}$$

i.e., if the probability that the probabilistic algorithm for analyzing an invertible function outputs a 1 is substantially greater when given that the block cipher e was chosen than when given that the complete block cipher $\check{e}$ was chosen.

Lemma 3.7 *If a computationally feasible probabilistic algorithm $\mathbb{A}$ for analyzing an invertible function is known that (1) solves for the block cipher e the problem of decrypting a ciphertext block chosen uniformly at random without asking the black box to decrypt it, or that (2) solves for the block cipher e the problem of encrypting a plaintext block chosen uniformly at random without asking the black box to encrypt it, or that (3) solves for the block cipher e the problem of finding an additional entry in the function table of the encryption function, then a computationally feasible probabilistic algorithm $\mathbb{A}'$ for analyzing an invertible function can be efficiently constructed that is distinguishing for the block cipher e.*

The proof of Lemma 3.7 is given on page 101.

$\{\mathcal{Z}_1, \mathcal{Z}_2, \ldots, \mathcal{Z}_L\}$ is called a decomposition (or "partition") of the non-empty set $\mathcal{Z}$ if the sets $\mathcal{Z}_1$, $\mathcal{Z}_2$, ..., $\mathcal{Z}_L$ are non-empty, pairwise disjoint subsets of the set $\mathcal{Z}$ and if the union of the sets $\mathcal{Z}_1$, $\mathcal{Z}_2$, ..., $\mathcal{Z}_L$ is equal to the set $\mathcal{Z}$.

Definition 3.8 *A probabilistic algorithm for analyzing an invertible function is* key-subset distinguishing for the block cipher e and for the decomposition $\{\mathcal{Z}_{e^1}, \mathcal{Z}_{e^2}, \ldots, \mathcal{Z}_{e^L}\}$ of the key space $\mathcal{Z}_e$ *if it outputs an L-ary decision $A = W$, $W \in \{1, 2, \ldots, L\}$ such that*

$$\mathrm{P}\left[Z \in \mathcal{Z}_{e^W}\right] \gg \max_{l=1}^{L} \frac{|\mathcal{Z}_{e^l}|}{|\mathcal{Z}_e|}, \tag{3.16}$$

i.e., if the probability that the secret key lies in the predicted subset of the decomposition is substantially greater than for a prediction that the secret key lies in the largest subset.

Example: For $L = 1$ the only decomposition of a key space $\mathcal{Z}_e$ is $\{\mathcal{Z}_{e^1}\}$ with $\mathcal{Z}_{e^1} = \mathcal{Z}_e$. For this decomposition of the key space the right part of (3.16) is equal to 1. Since no probability can be greater than 1 there exists also no probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for a block cipher e and for the decomposition $\{\mathcal{Z}_{e^1}\}$ of the key space $\mathcal{Z}_e$.

Example: Assume a block cipher e has a key space $\mathcal{Z}_e = \{0, 1\}^K$ and assume a probabilistic algorithm $\mathbb{A}$ for analyzing an invertible function outputs for the block cipher e the value of the first secret key bit with probability substantially greater than 1/2. Then let $\{\mathcal{Z}_{e^1}, \mathcal{Z}_{e^2}\}$ be the decomposition of the key space $\mathcal{Z}_e$ where $\mathcal{Z}_{e^1}$ contains all secret keys with the first secret key bit equal to 0 and $\mathcal{Z}_{e^2}$ contains all secret keys with the first secret key bit equal to 1. For this decomposition of the key space the right part of (3.16) is equal to 1/2. Simply by adding 1 to the output of the algorithm $\mathbb{A}$ one obtains a probabilistic algorithm $\mathbb{A}'$ for analyzing an invertible function that is key-subset distinguishing for the block cipher e and for the decomposition $\{\mathcal{Z}_{e^1}, \mathcal{Z}_{e^2}\}$ of the key space $\mathcal{Z}_e$.

Lemma 3.9 *If a computationally feasible probabilistic algorithm $\mathbb{A}$ for analyzing an invertible function is known that solves the problem of finding the secret key for the block cipher e, then (1) a decomposition $\{\mathcal{Z}_{e^1}, \mathcal{Z}_{e^2}, \ldots, \mathcal{Z}_{e^L}\}$ of the key space $\mathcal{Z}_e$ can efficiently be constructed and (2) a computationally feasible probabilistic algorithm $\mathbb{A}'$ for analyzing an invertible function that is key-subset distinguishing for the block cipher e and for this decomposition of the key space can be efficiently constructed.*

The proof of Lemma 3.9 is given on page 103.

Theorem 3.10 *If a cryptanalyst knows neither a computationally feasible probabilistic algorithm for analyzing an invertible function that is distinguishing for the block cipher e nor a decomposition $\{\mathcal{Z}_{e^1}, \mathcal{Z}_{e^2}, \ldots, \mathcal{Z}_{e^L}\}$ of the key space $\mathcal{Z}_e$ and a computationally feasible probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the block cipher e and for this decomposition of the key space, then this cryptanalyst cannot break the block cipher e.*

Proof: We prove this by showing that the contrapositive is true. The contrapositive is: if a cryptanalyst can break the block cipher e, then this cryptanalyst knows either a computationally feasible probabilistic algorithm for analyzing an invertible function that is distinguishing for the block cipher e or a decomposition $\{\mathcal{Z}_{e^1}, \mathcal{Z}_{e^2}, \ldots, \mathcal{Z}_{e^L}\}$ of the key space $\mathcal{Z}_e$ and a computationally feasible probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the block cipher e and for this decomposition of the key space. But this follows directly from combining Definition 3.5 with Lemma 3.7 and Lemma 3.9. □

Theorem 3.10 states that a cryptanalyst cannot break a block cipher e as long as he is not able to solve at least one of two basic problems for the block cipher e. One may ask whether it is possible to reduce one of these two basic problems to the other so that one could focus on trying to solve only one of the basic problems. To demonstrate that such a reduction does not exist, we give two examples. In each example we present a block cipher that can easily be broken and for which it is easy to solve one of the two basic problems while it is impossible to solve the other basic problem.

Example: Let e be a practical block cipher whose encryption functions are all identical. Since the block cipher e is practical, the single encryption function e_z is easy to compute for every secret key z. Let $\mathbb{A}_1$ be a probabilistic algorithm for analyzing an invertible function that chooses a ciphertext block Y_1 uniformly at random, computes $\hat{X}_1 = e_z(Y_1)$ for any z and outputs its analysis $A = [\hat{X}_1, Y_1]$. Clearly $\mathbb{A}_1$ is a computationally feasible probabilistic algorithm for analyzing an invertible function that solves for the block cipher e the problem of decrypting a ciphertext block chosen uniformly at random without asking the black box to decrypt it. Lemma 3.7 then implies that one can also construct a computationally feasible probabilistic algorithm $\mathbb{A}'_1$ for analyzing an invertible function that is distinguishing for the block cipher e. Similar arguments show that one can find a computationally feasible probabilistic algorithm $\mathbb{A}_2$ for analyzing an invertible function that solves for the block cipher e the problem of encrypting a plaintext block chosen uniformly at random without asking the black box to encrypt it and that one can construct a computationally feasible probabilistic algorithm $\mathbb{A}_3$ for analyzing an invertible function that solves for the block cipher e the problem of finding an additional entry in the function table of the encryption function. However, since analyzing the single encryption function e_z reveals no information about the chosen secret key z, there does not exist a probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the block cipher e and for any decomposition of the key space $\mathcal{Z}_e$ nor does there exist a probabilistic algorithm for analyzing an invertible function that solves for the block cipher e the problem of finding the secret key.

Example: Let e be the complete block cipher with block length $N = 2$. In the example given on page 34, we showed how a computationally feasible probabilistic algorithm for analyzing an invertible function could be constructed that solves the problem of finding the secret key. Lemma 3.9 then implies that one can find a decomposition of the key space and can construct a computationally feasible probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the block cipher e and for this decomposition of the key space. However, since the block cipher e is the complete block cipher $\check{e}$, the events E_e and $E_{\check{e}}$ are the same event. Therefore there exists no probabilistic algorithm for analyzing an invertible function that (1) is distinguishing for the block cipher, or that (2) solves for the block cipher e the problem of decrypting a ciphertext block chosen uniformly at

random without asking the black box to decrypt it, or that (3) solves for the block cipher e the problem of encrypting a plaintext block chosen uniformly at random without asking the black box to encrypt it, or that (4) solves for the block cipher e the problem of finding an additional entry in the function table of the encryption function.

Theorem 3.10 provides a sufficient condition for a cryptanalyst to be unable to break a given block cipher e. To demonstrate that this sufficient condition is not also a necessary one, we give two examples. In each example, we show a situation where not fulfilling the sufficient condition does not imply that the cryptanalyst can break the block cipher e.

Example: Assume that the only things a cryptanalyst knows about a block cipher e are the entire mechanism of enciphering (except for the value of the secret key) and a computationally feasible probabilistic algorithm $\mathbb{A}$ that is distinguishing for the block cipher e. Since the cryptanalyst already knows that he is analyzing a randomly chosen encryption function of the block cipher e, the probabilistic algorithm $\mathbb{A}$ reveals no additional information to him.

Example: Let e be a block cipher with key space $\mathcal{Z}_e = \{0,1\}^K$ and let $\{\mathcal{Z}_{e_1}, \mathcal{Z}_{e_2}\}$ be a decomposition of the key space $\mathcal{Z}_e$ where $\mathcal{Z}_{e_1}$ contains all secret keys with first key bit equal to 0 and where $\mathcal{Z}_{e_2}$ contains all secret keys with first key bit equal to 1. Assume that the only things a cryptanalyst knows about the block cipher e are the entire mechanism of enciphering (except for the value of the secret key) and a computationally feasible probabilistic algorithm $\mathbb{A}$ for analyzing an encryption function e_z that outputs a binary decision $A = W$, $W \in \{1,2\}$, such that $z \in \mathcal{Z}_W$ for all secret keys z in the key space $\mathcal{Z}_e$. Note that the probabilistic algorithm $\mathbb{A}$ uniquely determines the value of the first key bit of the secret key z. According to Definition 3.8 and since probability 1 is significantly greater than probability $1/2$, the probabilistic algorithm $\mathbb{A}$ is a computationally feasible probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the block cipher e and for the decomposition $\{\mathcal{Z}_{e_1}, \mathcal{Z}_{e_2}\}$ of the key space $\mathcal{Z}_e$. However, the cryptanalyst obtains no information about the value of the remaining $K-1$ bits of the secret key and therefore cannot do better than probability 2^{1-K} of guessing the correct value of the secret key. If the key length is large enough, e.g. $K = 256$, the cryptanalyst does not know a probabilistic algorithm for analyzing an invertible function that solves for the block cipher e the problem of finding the secret key.

Shannon [33] focused on probabilistic algorithms for analyzing an invertible function that are key-subset distinguishing for a block cipher and for a decomposition of the key space. Differential and linear cryptanalysis make use of a feature of a block cipher that can be used to construct a probabilistic algorithm for analyzing an invertible function that is distinguishing for this block cipher.

Chapter 4

Statistical Testing of Block Ciphers

In this chapter we describe an approach to finding an algorithm that is distinguishing for a given block cipher and an approach to finding an algorithm that is key-subset distinguishing for a given block cipher and a given decomposition of the key space.

This chapter is organized as follows. Section 4.1 gives the motivation for performing statistical testing of block ciphers. In Section 4.2 we show that the core of any algorithm that is distinguishing for some block cipher is an algorithm for *extracting a feature* from an invertible function. We then formulate an approach to finding an algorithm that is distinguishing for a given block cipher by first designing many different algorithms for extracting a feature from an invertible function and then testing which of these algorithms behave differently when applied to randomly chosen encryption functions of a given block cipher as compared to when they are applied to randomly chosen encryption functions of the complete block cipher. In Section 4.3 we consider certain block ciphers that can be derived from a given block cipher. We show under what conditions it makes sense to analyze the *dual* of a block cipher by the approach described in Section 4.2. We then formulate an approach to finding an algorithm that is key-subset distinguishing for a given block cipher and a given decomposition of the key space by first considering

many different subsets of the key space of a given block cipher and then analyzing *reduced-key-space* versions of the block cipher by the approach described in Section 4.2.

4.1 Why Statistical Testing

Statistical testing of block ciphers is intended to provide tests that are capable of analyzing any practical block cipher, no matter what the internal structure of the block cipher may be. Therefore such tests should analyze a block cipher based only on the input-output-behavior of its bivariate function e, where the bivariate function e is as defined in (3.1). To demonstrate how little of the function table of the bivariate function e can be considered by a test, we give an example. Suppose a test analyzes a block cipher e with block length $N = 64$ and key length $K = 128$. Suppose further that the test has access to 1 billion processors, each of which can compute entries in the function table of the bivariate function e at the speed of 1 billion entries per second. Finally suppose that the test has 1000 years time to present the result of its analysis. How many entries in the function table of the bivariate function e can the test consider for its analysis? The answer is 10^9 devices $\cdot 10^9$ entries/second/device $\cdot 1000$ years $\cdot 32 \cdot 10^6$ seconds/year $\approx 2^{95}$ entries. But the function table of the bivariate function e has a total of $2^N \cdot 2^K = 2^{196}$ entries. Therefore the test is forced to present the result of its analysis of the block cipher e after having seen only $1/2^{101}$ of the entries in the function table of the bivariate function e. This example makes it obvious why the nature of such a test can only be statistical.

A cryptanalyst can use statistical testing of a block cipher as a first step towards breaking a block cipher. The cryptanalyst can run several tests on the block cipher and, if some of these tests show a non-ideal behavior of the block cipher, then he can analyze the internal structure of the block cipher to see what caused the non-ideal behavior. This might give him ideas about how he could break the block cipher.

A cryptographer can use statistical testing of a block cipher to convince himself that a block cipher he designed at least does not have the weaknesses that he tested.

And finally, someone who has to evaluate block ciphers can use statistical testing of block ciphers to compare the behavior of different block ciphers without the need to analyze their internal structure.

4.2 Testing Model

In this section we simplify the task of finding a computationally feasible probabilistic algorithm for analyzing an invertible function that is distinguishing for the block cipher e. The core of any probabilistic algorithm for analyzing an invertible function that is distinguishing for the block cipher e is a probabilistic algorithm for analyzing an invertible function whose output has some dependency on the analyzed invertible function. Because of this dependency on the analyzed invertible function, we call such a probabilistic algorithm for analyzing an invertible function a probabilistic algorithm for *extracting a feature* from an invertible function. Note that a probabilistic algorithm for extracting a feature from an invertible function is independent of the block cipher e.

We will show how a probabilistic algorithm for extracting a feature from an invertible function can be embedded in a testing model that contains the block cipher e and that can be used to test whether the probabilistic algorithm for extracting a feature from an invertible function can be used to build a probabilistic algorithm for analyzing an invertible function that is distinguishing for the block cipher e.

The essential task is to design probabilistic algorithms for extracting a feature from an invertible function and then to test which of them can be used to build a probabilistic algorithm for analyzing an invertible function that is distinguishing for the block cipher e. The testing is done so that, if one concludes that a probabilistic algorithms for extracting a feature from an invertible function can be used to build a probabilistic algorithm for analyzing an invertible function that is distinguishing for the block cipher e, then one also knows how to build a probabilistic algorithm for analyzing an invertible function that is distinguishing for the block cipher e.

In order to have an additional degree of freedom, we allow our probabilistic algorithms to extract a feature from a *sequence* of G invertible functions, instead of from only a single invertible function.

Embedding a probabilistic algorithm for extracting a feature from a sequence of G invertible functions in a testing model that contains the block cipher e will, for $G > 1$, not lead directly to a probabilistic algorithm for analyzing an invertible function that is distinguishing for the block cipher e. But it might suggest how to design a probabilistic algorithm for extracting a feature from an invertible function that then leads to a probabilistic algorithm for analyzing an invertible function that is distinguishing for the block cipher e.

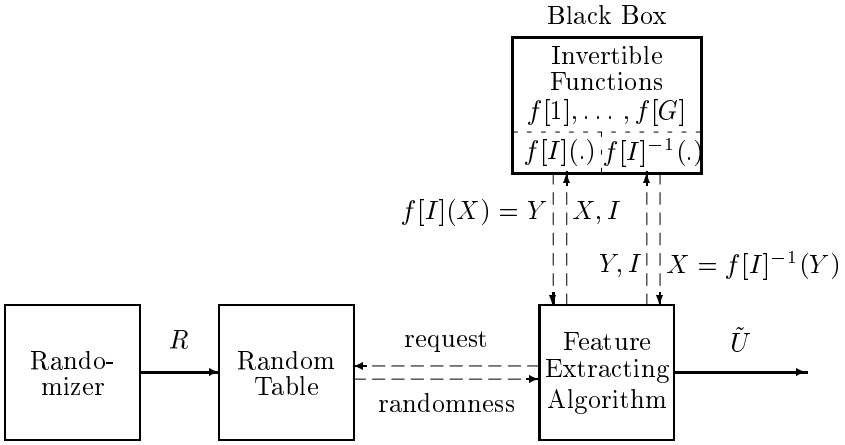

Figure 4.1: *Model for a probabilistic algorithm for extracting a feature from a sequence of G invertible functions $[f[1], f[2], \ldots, f[G]]$.*

Definition 4.1 *A* probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of G invertible functions $[f[1], f[2], \ldots, f[G]]$ in $\mathcal{F}_N^G$ *is an algorithm with the structure shown in Figure 4.1. The input to the algorithm is a "black box" for the sequence of G invertible functions $[f[1], f[2], \ldots, f[G]]$ which, when queried with the inputs X and I, returns $f[I](X)$, and, when queried with the inputs Y and I, returns $f[I]^{-1}(Y)$. The randomizer chooses a random string R from the finite set $\mathcal{R}$ according to a specified probability distribution P_R and loads the random table with this string. We will always assume that every r in $\mathcal{R}$ has non-zero probability $P_R(r)$. The feature extracting algorithm is a deterministic algorithm that makes queries to the black box and to the random table, receives the results of these queries, and determines*

whether to stop or to continue with queries. When this algorithm stops, it outputs its extracted feature $\tilde{U}$ *from the sequence of G invertible functions $[f[1], f[2], \ldots, f[G]]$. We will always assume that $\tilde{U}$ takes on a value in the finite set $\{\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_{\tilde{J}}\}$. The function computed by the feature extracting algorithm will be called the* feature extracting function $\tilde{U}(\mathbf{f}, r)$. *We require that the probability distribution of the extracted feature $\tilde{U}$ has some dependency on the sequence of G invertible functions $[f[1], f[2], \ldots, f[G]]$, i.e., the probability distribution of the extracted feature $\tilde{U}$ is not identical for all sequences of G invertible functions $[f[1], f[2], \ldots, f[G]]$ in $\mathcal{F}_N^G$.*

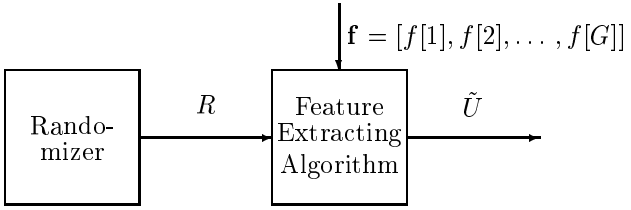

Figure 4.2: *Simplified diagram for a probabilistic algorithm for extracting a feature from a sequence of G invertible functions $[f[1], f[2], \ldots, f[G]]$.*

Figure 4.2 shows a simplified diagram for a probabilistic algorithm for extracting a feature from a sequence of invertible functions $\mathbf{f}$. The feature extracting algorithm computes the feature extracting function $\tilde{U}(\mathbf{f}, r)$, i.e., for a given random string r and a given sequence of invertible functions $\mathbf{f}$, the feature extracting algorithm deterministically computes the extracted feature $\tilde{u} = \tilde{U}(\mathbf{f}, r)$. The random string R is chosen from the finite set $\mathcal{R}$ according to the probability distribution P_R as specified by the probabilistic algorithm for extracting a feature from a sequence of invertible functions.

To describe a probabilistic algorithm for extracting a feature from a sequence of invertible functions, it is enough to define the feature extracting function $\tilde{U}(\mathbf{f}, r)$, the finite set $\mathcal{R}$, and the probability distribution P_R.

Example: Assume the feature extracting function has the structure $\tilde{U}(\mathbf{f}, r) = h(f[1](r))$, where the function h is not constant. This is a valid

feature extracting function since its value depends on the sequence of invertible functions $\mathbf{f}$. If this feature extracting function would be combined with a randomizer with $\mathcal{R} = \{0,1\}^N$ and $P_R(r) = 2^{-N}$, then this combination would not give a probabilistic algorithm for extracting a feature from a sequence of invertible functions since the probability distribution of the extracted feature would be independent of the sequence of invertible functions $\mathbf{f}$. The reason for this is that a uniform distribution at the input of an invertible function gives a uniform distribution at the output. On the other hand, combining the same feature extracting function as above with a randomizer with $\mathcal{R} = \{0\}^N$ and $P_R(r) = 1$ gives a probabilistic algorithm for extracting a feature from a sequence of invertible functions.

We now embed a probabilistic algorithm for extracting a feature from a sequence of invertible functions into the testing model that contains the block cipher e. A probabilistic algorithm for extracting a feature from a sequence of invertible functions has only one input, namely the sequence of invertible functions. In the testing model that contains the block cipher e, we choose M sequences of invertible functions, where each invertible function is an encryption function of the block cipher e chosen independently at random. For each of these M sequences of invertible functions, the probabilistic algorithm for extracting a feature from a sequence of invertible functions, with such a sequence of invertible functions as its input, is independently executed $\tilde{M}$ times. This produces a total of $M \cdot \tilde{M}$ extracted features which will then be analyzed. Figure 4.3 shows what is done for each of these M sequences of invertible functions and Figure 4.4 shows the complete testing model where all M sequences of invertible functions are used.

Figure 4.3 shows in its first "row" how one sequence of invertible functions is generated. For $g = 1, 2, \ldots, G$, the memoryless uniform secret key generator box chooses independently and uniformly at random a secret key $Z[g]$ from the key space $\mathcal{Z}_e$ of the block cipher e. The block cipher box takes a secret key $Z[g]$ and outputs the invertible function $F[g]$ realized by the the encryption function $e_{Z[g]}$. The framer for G invertible functions box collects G such invertible functions and frames them into one sequence of invertible functions $\mathbf{F}$. This sequence of invertible functions $\mathbf{F}$ is one input to the feature extracting algorithm box.

For $\tilde{m} = 1, 2, \ldots, \tilde{M}$, the memoryless randomizer box chooses a

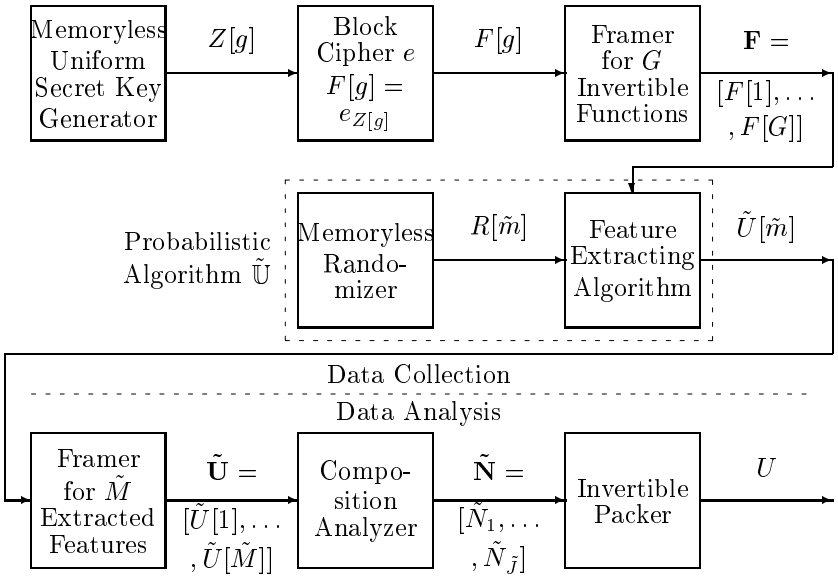

Figure 4.3: *Model for $\tilde{M}$ independent executions of a probabilistic algorithm for extracting a feature from a sequence of G invertible functions $[F[1], F[2], \ldots, F[G]]$ and for the analysis of the extracted features, where $F[1]$, $F[2]$, ..., $F[G]$ are randomly chosen encryption functions of a block cipher e.*

random string $R[\tilde{m}]$ from the set $\mathcal{R}$ independently and according to the probability distribution P_R . This random string $R[\tilde{m}]$ is the other input to the feature extracting algorithm box. The feature extracting algorithm box then computes the feature extracting function $\tilde{U}(\mathbf{F}, R[\tilde{m}])$ and outputs the extracted feature $\tilde{U}[\tilde{m}]$. The framer for $\tilde{M}$ extracted features box collects $\tilde{M}$ such extracted features and frames them into one sequence of extracted features $\tilde{\mathbf{U}}$.

Since the random strings $R[1]$, $R[2]$, ..., $R[\tilde{M}]$ are generated independently, we do not distinguish between sequences of extracted features which differ only by a rearrangement of their extracted features. For this reason, the sequence of extracted features $\tilde{\mathbf{U}}$ is reduced by the composition analyzer box to its composition $\tilde{\mathbf{N}}$ in the following manner:

The composition of the sequence $\tilde{\mathbf{u}} = [\tilde{u}[1], \tilde{u}[2], \ldots, \tilde{u}[\tilde{M}]]$ with $\tilde{u}[\tilde{m}] \in \{\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_{\tilde{J}}\}$ for $\tilde{m} = 1, 2, \ldots, \tilde{M}$ is the vector $\tilde{\mathbf{n}} = [\tilde{n}_1, \tilde{n}_2, \ldots, \tilde{n}_{\tilde{J}}]$ where $\tilde{n}_{\tilde{j}}$ is the number of occurrences of the symbol $\tilde{\mu}_{\tilde{j}}$ in $\tilde{\mathbf{u}}$ for $\tilde{j} = 1, 2, \ldots, \tilde{J}$. Note that $\sum_{\tilde{j}=1}^{\tilde{J}} \tilde{n}_{\tilde{j}} = \tilde{M}$. Note further that two sequences have the same composition if and only if they differ only by a rearrangement of their symbols.

$\tilde{\mathbf{n}}$	u
$[\tilde{M}, 0, 0, \ldots, 0]$	μ_1
$[\tilde{M}-1, 1, 0, \ldots, 0]$	μ_2
$\vdots$	$\vdots$
$[0, \ldots, 0, 1, \tilde{M}-1]$	μ_{J-1}
$[0, \ldots, 0, 0, \tilde{M}]$	μ_J

Table 4.1: *Invertible packer: mapping from compositions* $\tilde{\mathbf{n}}$ *to symbols* u *and vice versa.*

The composition $\tilde{\mathbf{N}}$ is packed by the invertible packer box into the symbol U. This packing is done so that it is invertible. Table 4.1 shows how compositions $\tilde{\mathbf{n}}$ are mapped to symbols u and vice versa. The number J of possible compositions is the same as the number of combinations with repetitions of $\tilde{M}$ elements from $\tilde{J}$ elements, i.e.,

$$J = \binom{\tilde{J} + \tilde{M} - 1}{\tilde{M}}. \tag{4.1}$$

Because of our assumption that the secret key is chosen uniformly at random from the key space we do not distinguish between block ciphers with block length N which differ only by an invertible renaming of the secret keys. Therefore there are $\binom{|\mathcal{F}_N|+1-1}{1}$ different block ciphers with a key space with cardinality 1 and there are $\binom{|\mathcal{F}_N|+2-1}{2}$ different block ciphers with a key space with cardinality 2 and so on. Hence there are countable infinitely many block ciphers with block length N. We will write $\mathcal{E}_N$ to denote the countable infinite set of block ciphers with block length N.

To compare the behavior of different block ciphers we need to specify an underlying random experiment. Let $\mathcal{E}_N^*$ be a given finite subset of the set $\mathcal{E}_N$ of block ciphers with block length N that contains at least the complete block cipher $\breve{e}$. We take this underlying random experiment to be that of making a (not necessarily uniform) random choice of a block cipher from the finite set of block ciphers $\mathcal{E}_N^*$. The probability distribution according to which block ciphers are chosen from $\mathcal{E}_N^*$ is assumed to be unknown. Let E_e denote the event that the block cipher e is drawn in this random experiment. Letting the random variable F be the invertible function realized by the chosen block cipher e and a secret key chosen uniformly at random from the key space of the block cipher e, we have

$$P_{F|E_e}(f) = \frac{|\{z : z \in \mathcal{Z}_e \text{ and } e_z = f\}|}{|\mathcal{Z}_e|}. \tag{4.2}$$

Since the sequence of invertible functions $\mathbf{F}$ in Figure 4.3 is a sequence of G independently chosen invertible functions F, we have

$$P_{\mathbf{F}|E_e}([f[1], f[2], \ldots, f[G]]) = P_{F|E_e}(f[1]) \cdot P_{F|E_e}(f[2]) \cdots P_{F|E_e}(f[G]). \tag{4.3}$$

For the complete block cipher $\breve{e}$, (4.3) reduces to

$$P_{\mathbf{F}|E_{\breve{e}}}([f[1], f[2], \ldots, f[G]]) = \frac{1}{2^{N!^G}}. \tag{4.4}$$

The probability that, for a given sequence of invertible functions $\mathbf{f}$, the output of the feature extracting algorithm takes on a particular value $\tilde{u}$ is

$$P_{\tilde{U}|\mathbf{F}}(\tilde{u}|\mathbf{f}) = \sum_{r \in \mathcal{R}} P_{\tilde{U}|\mathbf{F}R}(\tilde{u}|\mathbf{f}r) P_R(r) = \sum_{\substack{r \in \mathcal{R} \\ \tilde{U}(\mathbf{f}, r) = \tilde{u}}} P_R(r), \tag{4.5}$$

where we have used the fact that $P_{\tilde{U}|\mathbf{F}R}(\tilde{u}|\mathbf{f}r)$ is 1 for $\tilde{u} = \tilde{U}(\mathbf{f}, r)$ and 0 otherwise. In terms of the composition $\tilde{\mathbf{N}} = \tilde{\mathbf{n}} = [\tilde{n}_1, \tilde{n}_2, \ldots, \tilde{n}_{\tilde{J}}]$ of the observation $\tilde{\mathbf{U}} = \tilde{\mathbf{u}} = [\tilde{u}[1], \tilde{u}[2], \ldots, \tilde{u}[\tilde{M}]]$ in Figure 4.3, we have

$$P_{\tilde{\mathbf{U}}|\mathbf{F}}(\tilde{\mathbf{u}}|\mathbf{f}) = P_{\tilde{U}|\mathbf{F}}(\tilde{\mu}_1|\mathbf{f})^{\tilde{n}_1} \cdot P_{\tilde{U}|\mathbf{F}}(\tilde{\mu}_2|\mathbf{f})^{\tilde{n}_2} \cdots P_{\tilde{U}|\mathbf{F}}(\tilde{\mu}_{\tilde{J}}|\mathbf{f})^{\tilde{n}_{\tilde{J}}} \tag{4.6}$$

since the sequence $\tilde{\mathbf{U}}$ can be viewed as being generated by a discrete memoryless source with the (single-letter) probability distribution (4.5). The probability of a particular composition $\tilde{\mathbf{N}} = \tilde{\mathbf{n}}$ conditioned on $\mathbf{F} = \mathbf{f}$ is obtained by summing $P_{\tilde{\mathbf{U}}|\mathbf{F}}(\tilde{\mathbf{u}})$ over all $\tilde{\mathbf{u}}$ with this composition, which gives

$$P_{\tilde{\mathbf{N}}|\mathbf{F}}(\tilde{\mathbf{n}}|\mathbf{f}) = \frac{\tilde{M}!}{\tilde{n}_1!\tilde{n}_2!\cdots\tilde{n}_{\tilde{J}}!} P_{\tilde{\mathbf{U}}|\mathbf{F}}(\tilde{\mathbf{u}}|\mathbf{f}) \tag{4.7}$$

where the fraction on the right is the multinomial coefficient that gives the count of the number of sequences $\tilde{\mathbf{u}}$ with the composition $\tilde{\mathbf{n}}$. Summing over all invertible functions gives

$$P_{\tilde{\mathbf{N}}}(\tilde{\mathbf{n}}) = \sum_{\mathbf{f}\in\mathcal{F}_N^G} P_{\tilde{\mathbf{N}}|\mathbf{F}}(\tilde{\mathbf{n}}|\mathbf{f}) P_{\mathbf{F}}(\mathbf{f}) \tag{4.8}$$

and conditioning on the event that the block cipher e has been chosen gives

$$P_{\tilde{\mathbf{N}}|E_e}(\tilde{\mathbf{n}}) = \sum_{\mathbf{f}\in\mathcal{F}_N^G} P_{\tilde{\mathbf{N}}|\mathbf{F}E_e}(\tilde{\mathbf{n}}|\mathbf{f}) P_{\mathbf{F}|E_e}(\mathbf{f}). \tag{4.9}$$

But given $\mathbf{F} = \mathbf{f}$, $\tilde{\mathbf{N}}$ has no further dependence on E_e so that

$$P_{\tilde{\mathbf{N}}|E_e}(\tilde{\mathbf{n}}) = \sum_{\mathbf{f}\in\mathcal{F}_N^G} P_{\tilde{\mathbf{N}}|\mathbf{F}}(\tilde{\mathbf{n}}|\mathbf{f}) P_{\mathbf{F}|E_e}(\mathbf{f}). \tag{4.10}$$

Sometimes it is easier to compute $P_{\tilde{\mathbf{N}}|E_e}(\tilde{\mathbf{n}})$ if an intermediate random variable $\tilde{\mathbf{Q}}$ is introduced. For a given probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of invertible functions, the probability distribution of the random variable $\tilde{U}$ depends only on the chosen sequence of invertible functions $\mathbf{f}$. We write this probability distribution as the probability vector

$$\tilde{\mathbf{q}}_{\mathbf{f}} = \left[P_{\tilde{U}|\mathbf{F}}(\tilde{\mu}_1|\mathbf{f}), P_{\tilde{U}|\mathbf{F}}(\tilde{\mu}_2|\mathbf{f}), \ldots, P_{\tilde{U}|\mathbf{F}}(\tilde{\mu}_{\tilde{J}}|\mathbf{f})\right]. \tag{4.11}$$

Since the sequence of invertible functions is chosen from a finite set $\mathcal{F}_N^G$, there are finitely many probability vectors in

$$\tilde{\mathcal{Q}} = \left\{\tilde{\mathbf{q}}_{\mathbf{f}} : \mathbf{f} \in \mathcal{F}_N^G\right\}. \tag{4.12}$$

The intermediate random variable $\tilde{\mathbf{Q}}$ takes on a value in the finite set $\tilde{\mathcal{Q}}$ according to the probability distribution

$$P_{\tilde{\mathbf{Q}}|E_e}(\tilde{\mathbf{q}}) = \sum_{\substack{\mathbf{f}\in\mathcal{F}_N^G \\ \tilde{\mathbf{q}}_{\mathbf{f}}=\tilde{\mathbf{q}}}} P_{\mathbf{F}|E_e}(\mathbf{f}). \tag{4.13}$$

To write $P_{\tilde{\mathbf{N}}|E_e}(\tilde{\mathbf{n}})$ in terms of the probability distribution of the intermediate random variable $\tilde{\mathbf{Q}}$ instead in terms of the probability distribution of the random variable $\mathbf{F}$, we follow a derivation entirely similar to that which led to (4.10) to obtain

$$P_{\tilde{\mathbf{N}}|E_e}(\tilde{\mathbf{n}}) = \sum_{\tilde{\mathbf{q}}\in\tilde{\mathcal{Q}}} \frac{\tilde{M}!}{\tilde{n}_1!\tilde{n}_2!\cdots\tilde{n}_{\tilde{J}}!} \tilde{q}_1^{\tilde{n}_1}\tilde{q}_2^{\tilde{n}_2}\cdots\tilde{q}_{\tilde{J}}^{\tilde{n}_{\tilde{J}}} \cdot P_{\tilde{\mathbf{Q}}|E_e}(\tilde{\mathbf{q}}). \tag{4.14}$$

Because $U = u$ and $\tilde{\mathbf{N}} = \tilde{\mathbf{n}}$ are the same event, it follows that

$$P_{U|E_e}(u) = P_{\tilde{\mathbf{N}}|E_e}(\tilde{\mathbf{n}}). \tag{4.15}$$

For a given probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of invertible functions and for a given value of the parameter $\tilde{M}$, the probability distribution of the random variable U depends only on the chosen block cipher e. We write this probability distribution as the probability vector

$$\mathbf{q}_e = \left[P_{U|E_e}(\mu_1), P_{U|E_e}(\mu_2), \ldots, P_{U|E_e}(\mu_J)\right]. \tag{4.16}$$

What remains to be done is to execute the random experiment shown in Figure 4.3 M times for the same chosen block cipher e. This is shown in Figure 4.4. The upper part of Figure 4.4 is the same as that of Figure 4.3 except that all random variables have the additional index m which takes on the values $1, 2, \ldots, M$. Instead of one symbol U, there are now M symbols generated, viz., $U[1]$, $U[2]$, ..., $U[M]$. The framer for M symbols box collects M such symbols and frames them into a single sequence of symbols $\mathbf{U}$. The memoryless uniform secret key generator and the memoryless randomizer are assumed to be independent

random sources. Therefore the sequence $\mathbf{U}$ can be viewed as being generated by a discrete memoryless source with (single-letter) probability distribution $\mathbf{q}_e$.

The ideal source probability vector is obtained when the complete block cipher $\breve{e}$ is chosen, i.e.,

$$\mathbf{p} = \mathbf{q}_{\breve{e}}. \tag{4.17}$$

Since the block cipher is chosen from a finite set of block ciphers $\mathcal{E}_N^*$, there are finitely many probability vectors in

$$\mathcal{Q} = \{\mathbf{q}_e : e \in \mathcal{E}_N^*\}. \tag{4.18}$$

Note that we are now where we started in Chapter 2. In a first random experiment the probability vector $\mathbf{q}$ is chosen from the finite set $\mathcal{Q}$ according to some unknown probability distribution $P_{\mathbf{Q}}(\mathbf{q})$. Then, in a second random experiment, the sequence $\mathbf{u}$ is generated by a discrete memoryless source with (single-letter) probability distribution $\mathbf{q}$. The question is whether the generated sequence $\mathbf{u}$ looks different than a sequence that a discrete memoryless source with (single-letter) probability distribution $\mathbf{p}$ would have generated or not. The answer to this question is given in Chapter 2. Therefore the last three boxes in Figure 4.4 are the three boxes of the statistical test shown in Figure 2.2.

We now discuss the set $\mathcal{Q}$. What is always required to be known is the ideal source probability vector $\mathbf{p}$ which is equal to $\mathbf{q}_{\breve{e}}$. This we need to able to computed for the given probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of invertible functions and for the values of the parameter $\tilde{M}$ we are using. In Section 5.2 we show how this is done for a given example. Suppose we are analyzing the block cipher $\bar{e}$. Then we can try to compute $\mathbf{q}_{\bar{e}}$. In Section 5.2 we also give an example for which this is possible. If we can compute $\mathbf{q}_{\bar{e}}$, then we define

$$\mathcal{E}_N^* = \{\breve{e}, \bar{e}\}$$

and compute the set $\mathcal{Q}$ according to (4.18). If the set $\mathcal{Q}$ has cardinality 1, i.e., if $\mathbf{q}_{\breve{e}} = \mathbf{q}_{\bar{e}}$, then the given probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of invertible functions with the given parameter $\tilde{M}$ cannot be used to construct a probabilistic algorithm that is

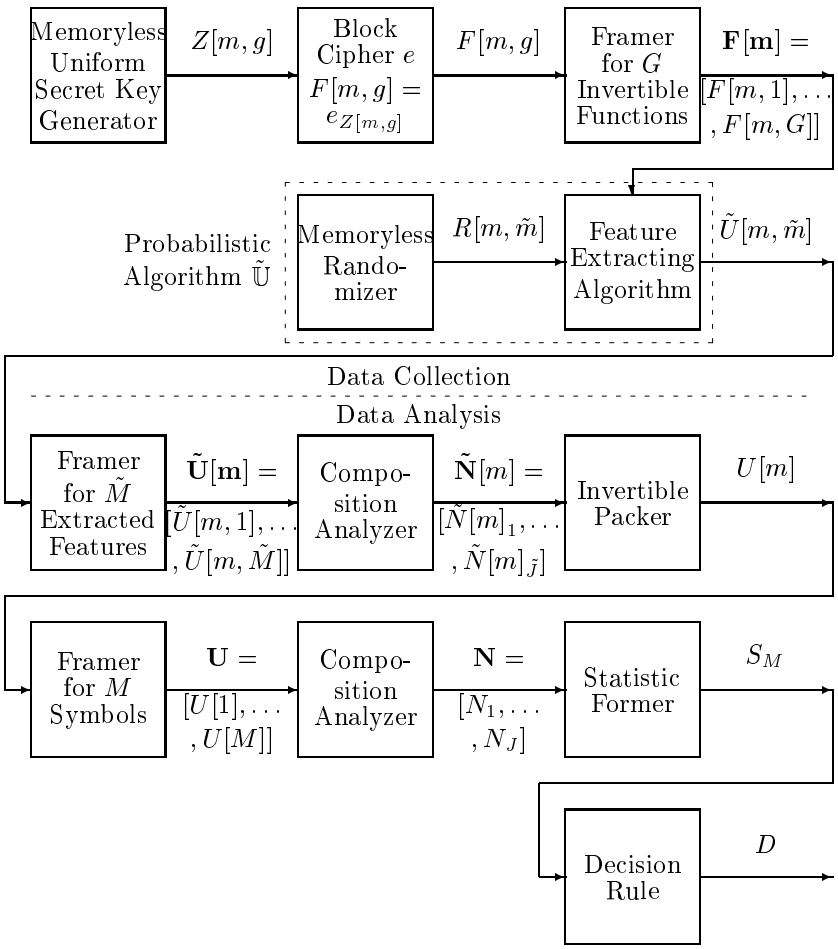

Figure 4.4: *A model for statistical testing of a block cipher e.*

distinguishing for the block cipher $\bar{e}$. If the set $\mathcal{Q}$ has cardinality 2, i.e., if $\mathbf{q}_{\check{e}} \neq \mathbf{q}_{\bar{e}}$, then we do not need to run a statistical test at all since we can compute the threshold T and the sequence length M, according to the results of Chapter 2, for which the statistical testing of the block cipher $\bar{e}$ shown in Figure 4.4 shows the difference between $\mathbf{q}_{\check{e}}$ and $\mathbf{q}_{\bar{e}}$. If we cannot compute $\mathbf{q}_{\bar{e}}$, then we define $\mathcal{E}_N^*$ such that it contains the complete block cipher $\check{e}$, the actual block cipher $\bar{e}$ and finitely many other block ciphers with the goal that for this choice of $\mathcal{E}_N^*$ we can compute the set $\mathcal{Q}$. For example one could define

$$\mathcal{E}_N^* = \{\check{e}\} \cup \{e : e \in \mathcal{E}_N \text{ and } e \text{ has same key length as } \bar{e}\}.$$

Again, if the set $\mathcal{Q}$ has cardinality 1, i.e., if $\mathbf{q}_{\check{e}} = \mathbf{q}_{\bar{e}}$, then the given probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of invertible functions with the given parameter $\tilde{M}$ cannot be used to construct a probabilistic algorithm that is distinguishing for the block cipher $\bar{e}$. If the set $\mathcal{Q}$ has cardinality 2 then we compute the threshold T and the sequence length M, according to the results of Chapter 2, for which the statistical testing of the block cipher $\bar{e}$ shown in Figure 4.4 shows either that $\mathbf{q}_{\bar{e}} = \mathbf{q}_{\check{e}}$ or that $\mathbf{q}_{\bar{e}} \neq \mathbf{q}_{\check{e}}$. To show this we have to run the test. If the set $\mathcal{Q}$ has cardinality larger than 2 or if the set $\mathcal{Q}$ is unknown then we run the test shown in Figure 4.4 for the actual block cipher $\bar{e}$.

In (2.52) we determined the set of probability vectors $\mathcal{Q}_1$, that contains all probability vectors $\mathbf{q}$ which our statistical test will detect with error probability at most β^* to be different from the ideal source probability vector $\mathbf{p}$. Based on this we determine the set of block ciphers $\mathcal{E}_{N,1}^*$, that contains all block ciphers e which our statistical test will detect with error probability at most β^* to be different from the complete block cipher $\check{e}$, i.e.,

$$\mathcal{E}_{N,1}^* = \{e : e \in \mathcal{E}_N^* \text{ and } \mathbf{q}_e \in \mathcal{Q}_1\}. \tag{4.19}$$

4.3 Block Ciphers to be Tested

In the previous section we showed how a block cipher can be analyzed. In this section we show which block ciphers can be derived from a given block cipher. All the derived block ciphers can then be analyzed by the method described in the previous section.

Definition 4.2 *The block cipher e with block length N and key space $\mathcal{Z}_e$ and the block cipher $e^\perp$ with block length N and key space $\mathcal{Z}_{e^\perp} = \mathcal{Z}_e$ are* duals *if the encryption function $e_z^\perp$ of the block cipher $e^\perp$ is identical to the decryption function e_z^{-1} of the block cipher e for every secret key z in the key space $\mathcal{Z}_e$.*

Under what condition does it make sense to analyze both the given block cipher e and its dual cipher $e^\perp$ by the approach described in the previous section? As shown in Figure 4.4, the final result of a test is the binary decision D. For a given probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of invertible functions and for given parameters $\tilde{M}$, M and T, the probability distribution of the random variable D depends only on the analyzed block cipher. Therefore, if one is not able to show for the given probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of invertible functions and for the given parameters $\tilde{M}$, M and T that

$$\mathrm{P}\left[D = 0 | E_e\right] = \mathrm{P}\left[D = 0 | E_{e^\perp}\right], \tag{4.20}$$

i.e., if one cannot show that the probability distribution of the random variable D is the same whether the given block cipher e or its dual cipher $e^\perp$ is analyzed, a reasonable strategy is to analyze both block ciphers. In this case, it is wiser to give simulation results as plots of the statistics S_M instead of as plots of the binary decisions D because the former plots reveal more information. Therefore, if one is not able to show for the given probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of invertible functions and for the given parameters $\tilde{M}$ and M that

$$\mathrm{P}\left[S_M \leq \tau | E_e\right] = \mathrm{P}\left[S_M \leq \tau | E_{e^\perp}\right] \quad \text{for all } \tau, \tag{4.21}$$

i.e., if one cannot show that the probability distribution function of the random variable S_M is the same whether the given block cipher e or its dual cipher $e^\perp$ is analyzed, a reasonable strategy is to analyze both block ciphers. This is the strategy that we will follow in the next chapter.

One could try to verify (4.21) for the given block cipher e only, or one could try to prove that this block cipher belongs to a family of block ciphers for which one can prove that every member of this family of block ciphers satisfies (4.21). For example, every block cipher that

is its own dual satisfies (4.21). (Note that all self-dual block ciphers can be broken since a ciphertext block can be decrypted by encrypting it a second time.) But as long as this family of block ciphers does not contain the set of all block ciphers with the same block length as the given block cipher, both approaches require that one analyze the internal structure of the given block cipher, which is the main thing we try to avoid in statistical testing of block ciphers. Therefore we prefer to try to show, for a given probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of G invertible functions in $\mathcal{F}_N^G$ and for given parameters $\tilde{M}$ and M, that (4.21) holds not only for the given block cipher with block length N but also for all block ciphers with block length N.

Definition 4.3 *For a given probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of G invertible functions in $\mathcal{F}_N^G$ and for given parameters $\tilde{M}$ and M,* the test shown in Figure 4.4 is undirected *if*

$$\mathrm{P}\left[S_M \leq \tau | E_e\right] = \mathrm{P}\left[S_M \leq \tau | E_{e^\perp}\right] \text{ for all } \tau \text{ and for all } e \in \mathcal{E}_N, \quad (4.22)$$

i.e., if, for all block ciphers e with block length N, the probability distribution function of the statistic S_M is the same whether the block cipher e or its dual cipher $e^\perp$ is analyzed. Otherwise the test shown in Figure 4.4 is directed.

Definition 4.3 moves the burden of proving that a test is undirected from the user of a test to the designer of a test. The simplified recommendation to the user of a test is then to analyze both block ciphers, the given block cipher e and its dual cipher $e^\perp$, by a test that need not be undirected and to analyze only one of these two block ciphers by a test that is certified to be undirected.

In order to show that a test is undirected, the designer of the test may investigate the probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of invertible functions that he used inside his test.

Definition 4.4 *A* probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of G invertible functions $\mathbf{f} = [f[1], f[2], \ldots, f[G]]$ in $\mathcal{F}_N^G$ is undirected *if the probability distribution for the extracted feature $\tilde{U}$ is the same whether conditioned on $\mathbf{F} = \mathbf{f}$ or conditioned on $\mathbf{F} = \mathbf{f}^{-1}$,*

where $\mathbf{f}^{-1} = [f[1]^{-1}, f[2]^{-1}, \ldots, f[G]^{-1}]$, i.e., if

$$P_{\tilde{U}|\mathbf{F}}(.|\mathbf{f}) = P_{\tilde{U}|\mathbf{F}}(.|\mathbf{f}^{-1}) \quad \text{for all } \mathbf{f} \text{ in } \mathcal{F}_N^G. \quad (4.23)$$

Otherwise the probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of invertible functions is directed.

Any undirected probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of invertible functions embedded in the model shown in Figure 4.4 will result in an undirected test. However, a directed probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of invertible functions embedded in the model shown in Figure 4.4 may result in either a directed test or an undirected test. In the next chapter we give an example of a directed probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of invertible functions that results in an undirected test (Figure 5.31).

We have treated the basic problem of finding a computationally feasible probabilistic algorithm for analyzing an invertible function that is distinguishing for a given block cipher e. We now consider the other basic problem of finding both a decomposition $\{\mathcal{Z}_{e^1}, \mathcal{Z}_{e^2}, \ldots, \mathcal{Z}_{e^L}\}$ of the key space $\mathcal{Z}_e$ of a given block cipher e and a computationally feasible probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for this block cipher and for this decomposition of the key space. Assume first that we have a decomposition $\{\mathcal{Z}_{e^1}, \mathcal{Z}_{e^2}, \ldots, \mathcal{Z}_{e^L}\}$ of the key space $\mathcal{Z}_e$ and that we want to find a computationally feasible probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the given block cipher e and for this decomposition of the key space.

Definition 4.5 *A block cipher e^* with block length N and key space $\mathcal{Z}_{e^*}$ is a* reduced-key-space version of a block cipher e *with block length N and key space $\mathcal{Z}_e$ if the key space $\mathcal{Z}_{e^*}$ is a subset of the key space $\mathcal{Z}_e$ and if the encryption function e_z^* is identical to the encryption function e_z for every secret key z in the key space $\mathcal{Z}_{e^*}$.*

Since the key spaces $\mathcal{Z}_{e^1}$, $\mathcal{Z}_{e^2}$, ..., $\mathcal{Z}_{e^L}$ in the assumed decomposition are all subsets of the key space $\mathcal{Z}_e$, we can according to Definition 4.5 obtain L reduced-key-space versions of the given block cipher e, namely

the block cipher e^1 with key space $\mathcal{Z}_{e^1}$, the block cipher e^2 with key space $\mathcal{Z}_{e^2}$, ... and the block cipher e^L with key space $\mathcal{Z}_{e^L}$.

In order to find a probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the given block cipher e and for the given decomposition of the key space, we have to find a probabilistic algorithm for analyzing an invertible function that behaves differently for at least two of the L reduced-key-space versions of the given block cipher e, say for the block ciphers e^k and e^l, when applied to a randomly chosen encryption function of the block cipher e^k as opposed to when applied to a randomly chosen encryption function of the block cipher e^l.

In the previous section we solved a similar problem. There we had to find a probabilistic algorithm for analyzing an invertible function that behaves differently when applied to a randomly chosen encryption function of a given block cipher e as opposed to when applied to a randomly chosen encryption function of the complete block cipher $\breve{e}$. The solution there was to design probabilistic algorithms $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function and then to test which of them can be used to build a probabilistic algorithm that behaves differently when applied to a randomly chosen encryption function of the given block cipher e as opposed to when applied to a randomly chosen encryption function of the complete block cipher $\breve{e}$. But there we had the advantage that we could compute the exact behavior of the probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function when applied to a randomly chosen encryption function of the complete block cipher $\breve{e}$ in the sense that we could test the behavior of the probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function when applied to a randomly chosen encryption function of the given block cipher e against the known behavior of the probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function when applied to a randomly chosen encryption function of the complete block cipher $\breve{e}$. This advantage we do not have here. In principle we have to test the behaviors of the probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function when applied to a randomly chosen encryption function against each other for all L reduced-key-space versions of the given block cipher e. Fortunately, this is not needed. It is sufficient to test individually the behaviors of the probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function when applied to a randomly chosen encryption

function for each of the L reduced-key-space versions of the given block cipher e against the known behavior of the probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function when applied to a randomly chosen encryption function of the complete block cipher $\breve{e}$. To show that this individual testing is sufficient, we distinguish three cases:

- If the L behaviors of the probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function when applied to a randomly chosen encryption function of each of the L reduced-key-space versions of the given block cipher e are not all identical, then some of these L behaviors must be different from the known behavior of the probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function when applied to a randomly chosen encryption function of the complete block cipher $\breve{e}$. If this difference in behavior is significant, then we can use this probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function to build a computationally feasible probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the given block cipher e and for the given decomposition of the key space.
- If the L behaviors of the probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function when applied to a randomly chosen encryption function of each of the L reduced-key-space versions of the given block cipher e are all identical but are different from the known behavior of the probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function when applied to a randomly chosen encryption function of the complete block cipher $\breve{e}$, then this probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function cannot be used to build a computationally feasible probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the given block cipher e and for the given decomposition of the key space. However, if this difference in behavior is significant, then we can use this probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function to build a computationally feasible probabilistic algorithm for analyzing an invertible function that is distinguishing for the given block cipher e.
- If the $L+1$ behaviors of the probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function when applied to a

randomly chosen encryption function of each of the L reduced-key-space versions of the given block cipher e and to a randomly chosen encryption function of the complete block cipher $\check{e}$ are all identical, then this probabilistic algorithm $\tilde{\mathbb{U}}$ for extracting a feature from an invertible function cannot be used to build a computationally feasible probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the given block cipher e and for the given decomposition of the key space nor can it be used to build a computationally feasible probabilistic algorithm for analyzing an invertible function that is distinguishing for the given block cipher e.

Therefore, instead of looking directly for a probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the given block cipher e and for the given decomposition $\{\mathcal{Z}_{e^1}, \mathcal{Z}_{e^2}, \ldots, \mathcal{Z}_{e^L}\}$ of the key space $\mathcal{Z}_e$, it is preferable to look for a probabilistic algorithm for analyzing an invertible function that is distinguishing for some of the L reduced-key-space versions of the given block cipher e, i.e., for some of the block ciphers e^1, e^2, $\ldots$, e^L. The block ciphers e^1, e^2, $\ldots$, e^L are all derived from the given block cipher e and they can all be analyzed by the method described in the previous section.

Two independent tasks remain to be done, namely the designing of decompositions of key spaces and the designing of probabilistic algorithms $\tilde{\mathbb{U}}$ for extracting a feature from a sequence of G invertible functions in $\mathcal{F}_N^G$. In the next chapter, we give examples of how to do both tasks.

Chapter 5

Bit-Dependency Tests for Block Ciphers

What can we say about the quality of a block cipher when we look only at a given subset of bits of the plaintext blocks and a given subset of bits of the ciphertext blocks? In this chapter we seek to answer this question as completely as possible. We start with the simplest case, develop a test for it, analyze the properties of this test and generalize it. As a result we obtain a family of tests that we call the bit-dependency tests.

5.1 Definition

The simplest bit-dependency tests are those where we look at only one bit of the plaintext block and only one bit of the ciphertext block. For such tests, we first consider the behavior of the encryption function in the forward direction only. We now explain how such tests are performed and analyzed.

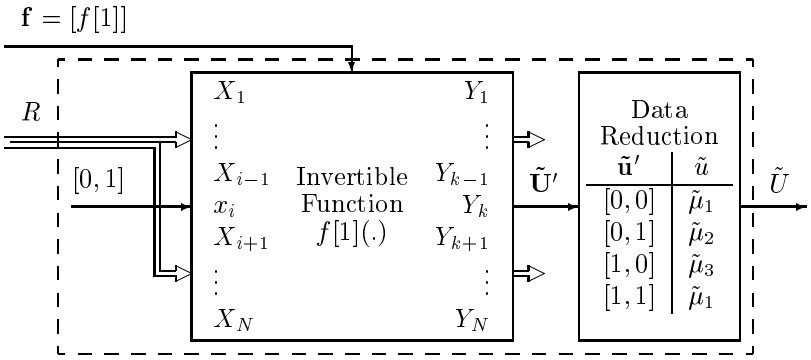

Figure 5.1: *Feature extracting algorithm for the directed bit-dependency test of i-th input bit and k-th output bit.*

First we select the input bit and the output bit to consider. Since there are N input bits and N output bits, there are N^2 possible combinations, and hence N^2 such single-bit tests. In the example of Figure 5.1, we have chosen the i-th input bit and the k-th output bit as the ones to consider in the test to be performed and analyzed.

The invertible function $\mathbf{f} = [f[1]]$ and the random string R are the inputs to the feature extracting algorithm in Figure 5.1 and the random variable $\tilde{U}$ is the output. The invertible function $\mathbf{f} = [f[1]]$ takes values in the set $\mathcal{F}_N^1$. The random string R takes values in the set $\mathcal{R}$ of vectors with $N-1$ binary components according to the uniform probability distribution $P_R(r) = 2^{1-N}$. The values of these two inputs determine the value of the random variable $\tilde{U}$ in the following way:

For given values of the invertible function $\mathbf{f} = [f[1]]$ and the random string R, we first set the value of the i-th input bit to a 0 and observe the value of the k-th output bit. Then we set the value of the i-th input bit to a 1 and again observe the value of the k-th output bit. There are four possible pairs that we can observe, namely $[0,0]$, $[0,1]$, $[1,0]$ and $[1,1]$. These four possible values are then reduced to the three values $\tilde{\mu}_1$, $\tilde{\mu}_2$ and $\tilde{\mu}_3$ by mapping both $[0,0]$ and $[1,1]$ to the same value, for reasons that will be explained in the next section. This mapping gives the random variable $\tilde{U}$.

Note that $\tilde{U}$ is a deterministic function of $\mathbf{f}$ and R. All the randomness involved in studying the behavior of an invertible function $\mathbf{f} = [f[1]]$ resides in the random string R. We assume that $f[1]$ is easy to compute (as will be the case when $f[1]$ is the encryption function of a practical block cipher for some choice of the secret key) and hence it is easy to compute $\tilde{U}$ from $\mathbf{f}$ and R. Nevertheless, for practical values of N, say $N = 64$ or $N = 128$, it is computationally infeasible for a given invertible function $\mathbf{f} = [f[1]]$ to compute $\tilde{U}$ for all 2^{N-1} possible values of R. We are forced to settle for computing $\tilde{U}$ for only a small fraction of all possible values of R.

5.2 Analysis

The random string R in Figure 5.1 can take 2^{N-1} different values. For a given invertible function $\mathbf{f} = [f[1]]$ and for a given random string R we observe a sequence $\tilde{\mathbf{u}}'$ that is either $[0,0]$, $[0,1]$, $[1,0]$ or $[1,1]$. If we go over all 2^{N-1} random strings R we will observe for example k_1 sequences $[0,0]$, k_2 sequences $[0,1]$, k_3 sequences $[1,0]$ and k_4 sequences $[1,1]$. Since the function $\mathbf{f} = [f[1]]$ is invertible the total number of 0's and the total number of 1's in all 2^{N-1} observed sequences $\tilde{\mathbf{u}}'$ will be the same, i.e., k_1 is equal to k_4 for all invertible functions $\mathbf{f} = [f[1]]$ in $\mathcal{F}_N^1$. Because the random string R is chosen uniformly at random we have

$$P_{\tilde{\mathbf{U}}'|\mathbf{F}}([0,0]|\mathbf{f}) = P_{\tilde{\mathbf{U}}'|\mathbf{F}}([1,1]|\mathbf{f}) \tag{5.1}$$

for all invertible functions $\mathbf{f}$ in $\mathcal{F}_N^1$. We make use of this *a priori* knowledge by not distinguishing between an observed sequence $[0,0]$ and an

observed sequence $[1,1]$.

$$\tilde{\mathcal{Q}} = \left\{ \frac{1}{2^{N-1}}[2k_1, k_2, k_3] : k_1, k_2, k_3 \in \aleph \text{ and } 2k_1 + k_2 + k_3 = 2^{N-1} \right\}, \tag{5.2}$$

$$P_{\tilde{\mathbf{Q}}|E_{\ddot{e}}}\left(\frac{1}{2^{N-1}}[2k_1, k_2, k_3]\right) = \frac{2^{N-1}!^3}{k_1!^2 k_2! k_3! 2^N!}. \tag{5.3}$$

The conditional probability distribution $P_{U|E_{\ddot{e}}}$ of the random variable U in the testing model in Figure 4.4 is obtained by inserting (5.3) in (4.14), directly computing the sum and applying (4.15). The results for $\tilde{M} = 1$, $\tilde{M} = 2$, and $\tilde{M} = 3$ are shown in Table 5.1, Table 5.2 and Table 5.3, respectively.

We will write $\mathcal{A}_N$ to denote the set of all affine invertible functions $\{0,1\}^N \to \{0,1\}^N$, i.e., $\mathcal{A}_N = \{f : f \in \mathcal{F}_N$ and there exists a matrix A and a vector $\mathbf{b}$ such that $f(\mathbf{x}) = \mathbf{x} \cdot \mathsf{A} + \mathbf{b}$ for all $\mathbf{x} \in \{0,1\}^N\}$. There are $|\mathcal{A}_N| = 2^N \prod_{i=0}^{N-1}(2^N - 2^i)$ functions in $\mathcal{A}_N$.

Definition 5.1 *The* affine block cipher with block length N *is the block cipher*

$$\ddot{e} : \{0,1\}^N \times \mathcal{Z}_{\ddot{e}} \to \{0,1\}^N : (x, z) \mapsto \ddot{e}_z(x), \tag{5.4}$$

where for each affine invertible function f in $\mathcal{A}_N$ there exists exactly one secret key z in $\mathcal{Z}_{\ddot{e}}$ such that $\ddot{e}_z = f$.

The affine block cipher $\ddot{e}$ with block length N can easily be broken. After encrypting N linearly independent plaintext blocks and one additional plaintext block, that is a linear combination of the previous N plaintext blocks, one can solve the obtained $N+1$ linear equations and easily compute the matrix A and the vector $\mathbf{b}$ for which $\ddot{e}_z(\mathbf{x}) = \mathbf{x} \cdot \mathsf{A} + \mathbf{b}$, where z is the actual secret key.

For the underlying random experiment described on page 49 and for the affine block cipher $\ddot{e}$, (4.2) reduces to

$$P_{F|E_{\ddot{e}}}(f) = \begin{cases} \frac{1}{|\mathcal{A}_N|} & \text{if } f \in \mathcal{A}_N \\ 0 & \text{if } f \notin \mathcal{A}_N. \end{cases} \tag{5.5}$$

$$P_{\tilde{\mathbf{Q}}|E_{\ddot{e}}}(\tilde{\mathbf{q}}) = \begin{cases} \frac{1}{2} - \frac{1}{2(2^N-1)} & \text{if } \tilde{\mathbf{q}} = [1,0,0] \text{ or } \tilde{\mathbf{q}} = [0, \frac{1}{2}, \frac{1}{2}] \\ \frac{1}{2(2^N-1)} & \text{if } \tilde{\mathbf{q}} = [0,1,0] \text{ or } \tilde{\mathbf{q}} = [0,0,1] \\ 0 & \text{else.} \end{cases} \tag{5.6}$$

The conditional probability distribution $P_{U|E_{\ddot{e}}}$ of the random variable U in the testing model in Figure 4.4 is obtained by inserting (5.6) in (4.14), directly computing the sum and applying (4.15). The results for $\tilde{M} = 1$, $\tilde{M} = 2$, and $\tilde{M} = 3$ are shown in Table 5.1, Table 5.2 and Table 5.3, respectively.

$\tilde{\mathbf{n}}$	u	$P_{U\|E_{\ddot{e}}}(u)$	$P_{U\|E_{\ddot{e}}}(u)$
$[1,0,0]$	μ_1	$\frac{1}{2} - \frac{2}{4(2^N-1)}$	$\frac{1}{2} - \frac{2}{4(2^N-1)}$
$[0,1,0]$	μ_2	$\frac{1}{4} + \frac{1}{4(2^N-1)}$	$\frac{1}{4} + \frac{1}{4(2^N-1)}$
$[0,0,1]$	μ_3	$\frac{1}{4} + \frac{1}{4(2^N-1)}$	$\frac{1}{4} + \frac{1}{4(2^N-1)}$

Table 5.1: *Probability distribution of the random variable U while analyzing the complete block cipher $\ddot{e}$ and the affine block cipher $\ddot{e}$, respectively, by the directed bit-dependency test for 1 input bit and 1 output bit by performing $\tilde{M} = 1$ executions of the body of the inner loop per trial.*

$\tilde{\mathbf{n}}$	u	$P_{U\|E_{\ddot{e}}}(u)$	$P_{U\|E_{\ddot{e}}}(u)$
$[2,0,0]$	μ_1	$\frac{1}{4} + \frac{4}{16(2^N-1)(2^N-3)}$	$\frac{1}{2} - \frac{4}{8(2^N-1)}$
$[1,1,0]$	μ_2	$\frac{1}{4} - \frac{8 \cdot 2^N - 20}{16(2^N-1)(2^N-3)}$	0
$[1,0,1]$	μ_3	$\frac{1}{4} - \frac{8 \cdot 2^N - 20}{16(2^N-1)(2^N-3)}$	0
$[0,2,0]$	μ_4	$\frac{1}{16} + \frac{8 \cdot 2^N - 23}{16(2^N-1)(2^N-3)}$	$\frac{1}{8} + \frac{3}{8(2^N-1)}$
$[0,1,1]$	μ_5	$\frac{1}{8} + \frac{2}{16(2^N-1)(2^N-3)}$	$\frac{1}{4} - \frac{2}{8(2^N-1)}$
$[0,0,2]$	μ_6	$\frac{1}{16} + \frac{8 \cdot 2^N - 23}{16(2^N-1)(2^N-3)}$	$\frac{1}{8} + \frac{3}{8(2^N-1)}$

Table 5.2: *Probability distribution of the random variable U while analyzing the complete block cipher $\ddot{e}$ and the affine block cipher $\ddot{e}$, respectively, by the directed bit-dependency test for 1 input bit and 1 output bit by performing $\tilde{M} = 2$ executions of the body of the inner loop per trial.*

Note that the probability distributions $P_{U|E_{\ddot{e}}}(.)$ and $P_{U|E_{\ddot{e}}}(.)$ shown in Table 5.1 are identical. Therefore, the directed bit-dependency test

$\tilde{\mathbf{n}}$	u	$P_{U\|E_{\breve{e}}}(u)$	$P_{U\|E_{\ddot{e}}}(u)$
$[3,0,0]$	μ_1	$\frac{1}{8}+\frac{24\cdot 2^{3N}-216\cdot 2^{2N}+568\cdot 2^{N}-512}{64\cdot 2^{N}(2^{N}-1)(2^{N}-3)(2^{N}-5)}$	$\frac{1}{2}-\frac{8}{16(2^{N}-1)}$
$[2,1,0]$	μ_2	$\frac{3}{16}-\frac{36\cdot 2^{3N}-348\cdot 2^{2N}+972\cdot 2^{N}-768}{64\cdot 2^{N}(2^{N}-1)(2^{N}-3)(2^{N}-5)}$	0
$[2,0,1]$	μ_3	$\frac{3}{16}-\frac{36\cdot 2^{3N}-348\cdot 2^{2N}+972\cdot 2^{N}-768}{64\cdot 2^{N}(2^{N}-1)(2^{N}-3)(2^{N}-5)}$	0
$[1,2,0]$	μ_4	$\frac{3}{32}+\frac{18\cdot 2^{3N}-258\cdot 2^{2N}+1098\cdot 2^{N}-1344}{64\cdot 2^{N}(2^{N}-1)(2^{N}-3)(2^{N}-5)}$	0
$[1,1,1]$	μ_5	$\frac{3}{16}-\frac{60\cdot 2^{3N}-540\cdot 2^{2N}+1452\cdot 2^{N}-1152}{64\cdot 2^{N}(2^{N}-1)(2^{N}-3)(2^{N}-5)}$	0
$[1,0,2]$	μ_6	$\frac{3}{32}+\frac{18\cdot 2^{3N}-258\cdot 2^{2N}+1098\cdot 2^{N}-1344}{64\cdot 2^{N}(2^{N}-1)(2^{N}-3)(2^{N}-5)}$	0
$[0,3,0]$	μ_7	$\frac{1}{64}+\frac{21\cdot 2^{3N}-123\cdot 2^{2N}-17\cdot 2^{N}+544}{64\cdot 2^{N}(2^{N}-1)(2^{N}-3)(2^{N}-5)}$	$\frac{1}{16}+\frac{7}{16(2^{N}-1)}$
$[0,2,1]$	μ_8	$\frac{3}{64}+\frac{15\cdot 2^{3N}-129\cdot 2^{2N}+333\cdot 2^{N}-288}{64\cdot 2^{N}(2^{N}-1)(2^{N}-3)(2^{N}-5)}$	$\frac{3}{16}-\frac{3}{16(2^{N}-1)}$
$[0,1,2]$	μ_9	$\frac{3}{64}+\frac{15\cdot 2^{3N}-129\cdot 2^{2N}+333\cdot 2^{N}-288}{64\cdot 2^{N}(2^{N}-1)(2^{N}-3)(2^{N}-5)}$	$\frac{3}{16}-\frac{3}{16(2^{N}-1)}$
$[0,0,3]$	μ_{10}	$\frac{1}{64}+\frac{21\cdot 2^{3N}-123\cdot 2^{2N}-17\cdot 2^{N}+544}{64\cdot 2^{N}(2^{N}-1)(2^{N}-3)(2^{N}-5)}$	$\frac{1}{16}+\frac{7}{16(2^{N}-1)}$

Table 5.3: *Probability distribution of the random variable U while analyzing the complete block cipher $\breve{e}$ and the affine block cipher $\ddot{e}$, respectively, by the directed bit-dependency test for 1 input bit and 1 output bit by performing $\tilde{M}=3$ executions of the body of the inner loop per trial.*

for one input bit and one output bit with parameter $\tilde{M}=1$ behaves the same when analyzing the affine block cipher $\ddot{e}$ as when analyzing the complete block cipher $\breve{e}$.

To demonstrate that there exist block ciphers e for which the directed bit-dependency test for one input bit and one output bit with parameter $\tilde{M}=1$ behaves differently when analyzing the block cipher e as opposed to when analyzing the complete block cipher $\breve{e}$, we now describe a family of block ciphers e for which $P_{U|E_e}(\mu_3)\neq P_{U|E_{\ddot{e}}}(\mu_3)$. Using (4.1) gives $J=\binom{3+1-1}{1}=3$, using Table 4.1 and (4.15) gives $P_{U|E_e}(\mu_3)=P_{\tilde{N}|E_e}([0,0,1])$, and using (4.14) gives $P_{U|E_e}(\mu_3)=\sum_{\tilde{\mathbf{q}}\in\tilde{\mathcal{Q}}}\tilde{q}_3 P_{\tilde{\mathbf{Q}}|E_e}(\tilde{\mathbf{q}})$. According to (5.2), every probability vector $\tilde{\mathbf{q}}$ in $\tilde{\mathcal{Q}}$ has the property that the value of its last component $\tilde{q}_3$ is a rational number $k_3/2^{N-1}$, where k_3 is some nonnegative integer smaller than or equal to 2^{N-1}. It follows that

$$P_{U|E_e}(\mu_3)=\sum_{k_3=0}^{2^{N-1}}\frac{k_3}{2^{N-1}}\sum_{\substack{\tilde{\mathbf{q}}\in\tilde{\mathcal{Q}}\\ \tilde{q}_3=\frac{k_3}{2^{N-1}}}}P_{\tilde{\mathbf{Q}}|E_e}(\tilde{\mathbf{q}}). \tag{5.7}$$

Combining (5.7) with (4.13) and (4.2) gives

$$P_{U|E_e}(\mu_3)=\sum_{k_3=0}^{2^{N-1}}\frac{k_3}{2^{N-1}}\frac{i_{k_3}}{|\mathcal{Z}_e|}, \tag{5.8}$$

where the constants i_0, i_1, $\ldots$, $i_{2^{N-1}}$ are nonnegative integers. Using Table 5.1 gives

$$P_{U|E_{\ddot{e}}}(\mu_3)=\frac{2^{N-2}}{2^N-1}. \tag{5.9}$$

Multiplying (5.8) and (5.9) by $2^{N-1}|\mathcal{Z}_e|(2^N-1)$ gives

$$P_{U|E_e}(\mu_3)2^{N-1}|\mathcal{Z}_e|(2^N-1)=(2^N-1)\sum_{k_3=0}^{2^{N-1}}k_3 i_{k_3} \tag{5.10}$$

and

$$P_{U|E_{\ddot{e}}}(\mu_3)2^{N-1}|\mathcal{Z}_e|(2^N-1)=|\mathcal{Z}_e|2^{2N-3}. \tag{5.11}$$

For $|\mathcal{Z}_e| = 2^K$ where K is a nonnegative integer and for $N \geq 2$, the prime factorization of the right side of (5.10) contains at least one odd prime while the prime factorization of the right side of (5.11) contains no odd prime. Hence $P_{U|E_e}(\mu_3) \neq P_{U|E_{\check{e}}}(\mu_3)$. Therefore, for any block cipher e with block length $N \geq 2$ and with a key space of cardinality $|\mathcal{Z}_e| = 2^K$ where K is a nonnegative integer, the directed bit-dependency test for one input bit and one output bit with parameter $\tilde{M} = 1$ behaves differently when analyzing the block cipher e as opposed to when analyzing the complete block cipher $\check{e}$. Note that virtually all contemporary block ciphers as well as all of the block ciphers we analyzed are of this type. For any block cipher of this type and for any combination of a single input bit and a single output bit, there exists a large enough number of trials M for which the directed bit-dependency test for one input bit and one output bit with parameter $\tilde{M} = 1$ can detect this difference in behavior. However, for some block ciphers of this type and some combinations of a single input bit and a single output bit, the required number of trials M to detect this difference in behavior can be so large that the resulting directed bit-dependency test for one input bit and one output bit with parameter $\tilde{M} = 1$ is computationally infeasible.

5.3 Simulation Results, Part I

In this section we show the simulations results we obtained by analyzing different block ciphers by the directed bit-dependency test for 1 input bit and 1 output bit with parameter $\tilde{M} = 1$.

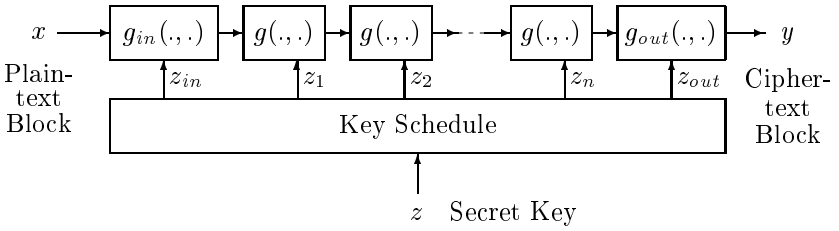

Figure 5.2: *Structure of an n-round iterated block cipher with bivariate round function g.*

All of the block ciphers we analyzed are iterated block ciphers which have a structure as shown in Figure 5.2. In an iterated block cipher a simple bivariate function g is iterated several times. Each iteration is called a *round*. The simple bivariate function g takes a vector with N binary components and a subkey as the inputs and outputs a vector with N binary components. The subkeys are derived from the secret key by a key schedule algorithm. Some of the analyzed block ciphers have an additional simple bivariate function g_{in} at the input and/or an additional simple bivariate function g_{out} at the output. The number of rounds is part of the definition of an iterated block cipher. What we analyzed are reduced versions of iterated block ciphers obtained by reducing the number of rounds.

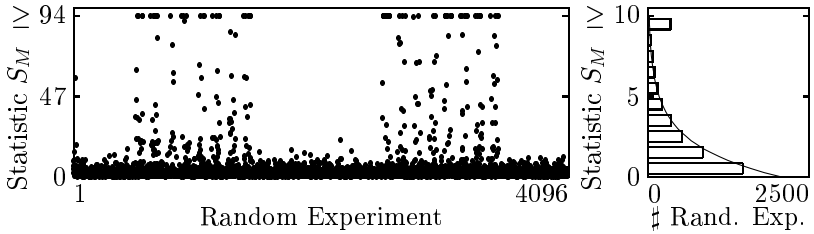

Figure 5.3: *The block cipher* **DES** *reduced to* **6 rounds** *analyzed by the directed bit-dependency test for 1 input bit and 1 output bit by performing $M = 2^{26}$ trials and $\tilde{M} = 1$ execution of the body of the inner loop per trial.*

The block cipher DES (Data Encryption Standard) is a 16-round iterated block cipher with block length $N = 64$ and key length $K = 56$ developed by IBM and taken in 1977 by the National Bureau of Standards [26].

Figure 5.3 shows the simulation results obtained by analyzing the block cipher DES reduced to 6 rounds by the directed bit-dependency test for one input bit and one output bit by performing $M = 2^{26}$ trials and $\tilde{M} = 1$ executions of the body of the inner loop per trial. Since there are N^2 possible combinations of a single input bit and a single output bit we did run $64^2 = 4096$ different random experiments. The left plot in Figure 5.3 shows on the abscissa these different random experiments numbered form 1 to 4096 and on the ordinate the obtained statistic S_M

for each of these different random experiments. Any statistic S_M that was larger than or equal to 94 is shown on the level labeled "$\geq$ 94". Note that we show the statistic S_M instead of the decision D. We do so in order to obtain more information from our simulations.

As can be seen from Figure 5.1 the feature extracting algorithm of the directed bit-dependency test for one input bit and one output bit has $\tilde{J} = 3$ different output values. Together with (4.1) and with $\tilde{M} = 1$ we get $J = \binom{3+1-1}{1} = 3$ as the length of the ideal source probability vector. The ideal source probability vector is shown in Table 5.1 in the column labeled with $P_{U|E_{\tilde{e}}}(u)$.

We accept a probability of type I error $\alpha = 10^{-10}$. From Table 2.3 we get $\mathrm{P}\left[\chi^2_{J-1} \leq 46.05\right] = 1 - 10^{-10}$ and obtain the threshold $T = 46.05$. In the left plot in Figure 5.3 we have chosen the scale of the ordinate such that the chosen threshold T is in the middle of the plot. Any point in the left plot that has an $S_M > 46.05$ identifies an input-bit/output-bit pair for which we are $(1 - \alpha)$-certain that it behaves non-ideally according to the directed bit-dependency test.

The right plot in Figure 5.3 shows the histogram for the 4096 statistics S_M. The histogram shows on its abscissa the number of statistics S_M in the range shown on its ordinate. The number of statistics $S_M > 10$ is added to the bar just below the label "$\geq$ 10". The solid line in the left plot shows the ideal distribution that we would observe if the complete block cipher would be tested.

The simulation results shown in Figure 5.3 were obtained by performing $M = 2^{26}$ trials and $\tilde{M} = 1$ executions of the body of the inner loop per trial. Since for each executions of the body of the inner loop one has to encrypt two plaintext blocks, we had for each point in the left plot of Figure 5.3 to encrypt 1 gigabyte of plaintext. For all 4096 points we had to encrypt a total of 4 terabyte of plaintext. This would have required 110 days of computation time if we had used a single one of the available computers. By using several computers in parallel, we obtained the simulation results shown in Figure 5.3 within 10 days.

All the simulation results shown in the remainder of this section were obtained by the directed bit-dependency test for one input bit and one output bit by performing $M = 2^{21}$ trials and $\tilde{M} = 1$ execution of the body of the inner loop per trial.

To show how many rounds of encryption are required before the directed bit-dependency test for one input bit and one output bit with parameters $\tilde{M} = 1$ and $M = 2^{21}$ fails to detect a difference between the tested block cipher and the complete block cipher, we show on each page the simulation results for two block ciphers. For example, Figure 5.4 shows the simulation results for the block cipher DES reduced to 5 rounds where some of the statistics S_M are larger than the threshold T. Increasing the number of rounds by one gives the block cipher DES reduced to 6 rounds. Figure 5.5 shows the simulation results for the block cipher DES reduced to 6 rounds where none of the statistics S_M are larger than the threshold T. Therefore, 6 rounds of encryption are required before the directed bit-dependency test for one input bit and one output bit with parameters $\tilde{M} = 1$ and $M = 2^{21}$ fails to detect a difference between a reduced version of the block cipher DES and the complete block cipher.

Since the directed bit-dependency test for one input bit and one output bit with parameter $\tilde{M} = 1$ is a directed test, we run this test not only for a block cipher e but also for its dual cipher $e^{\perp}$. For example, Figure 5.6 shows the simulation results for the dual of the block cipher DES reduced to 5 rounds and Figure 5.7 shows the simulation results for the dual of the block cipher DES reduced to 6 rounds.

Since, in the ideal case, the expected number of statistics S_M in the range $9 \leq S_M < 10$ is $4096 \cdot \mathrm{P}\left[9 \leq S_M \text{ and } S_M < 10 | E_{\tilde{e}}\right] = 17.9$ and the expected number of statistics S_M larger than or equal to 9 is $4096 \cdot \mathrm{P}\left[9 \leq S_M | E_{\tilde{e}}\right] = 45.5$, and since the top bar in the histogram indicates not the number of statistics S_M in the range $9 \leq S_M < 10$ but the number of statistics S_M larger than or equal to 9, in the ideal case it is expected that the top bar in the histogram reaches a little above the solid line of the ideal distribution. However, Figure 5.5 shows 86 statistics S_M larger than or equal to 9, which is nearly twice as many as expected in the ideal case. Also the left plot in Figure 5.5 shows some random experiments with rather large statistics S_M. This indicates that there may be non-ideal behaviors just on the verge of being detected. What one does in such a situation is to increase the number of trials M and check whether there are now non-ideal behaviors. Figure 5.3 shows that analyzing the same

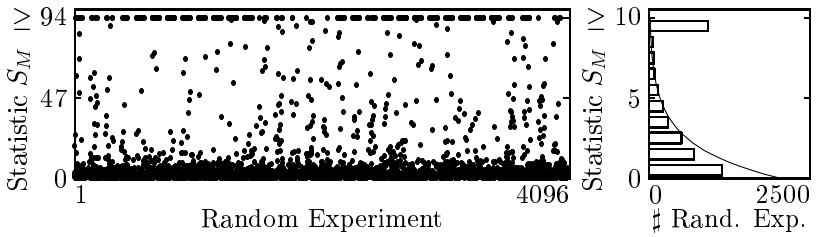

Figure 5.4: *The block cipher* **DES** *reduced to* **5 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

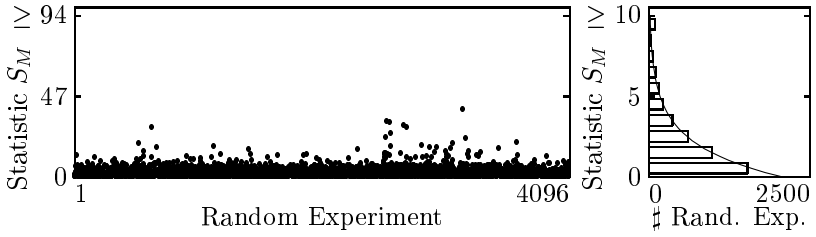

Figure 5.5: *The block cipher* **DES** *reduced to* **6 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

block cipher as for Figure 5.5 by the directed bit-dependency test for one input bit and one output bit with parameters $\tilde{M} = 1$ and $M = 2^{26}$ did indeed detect non-ideal behaviors.

Figures 5.6 and 5.7 show essentially the same behaviors as Figures 5.4 and 5.5. Therefore, analyzing the block cipher DES reduced to 5 rounds and to 6 rounds, respectively, by the directed bit-dependency test for one input bit and one output bit with parameters $\tilde{M} = 1$ and $M = 2^{21}$ and analyzing the dual ciphers by this same test gave no essential difference in the detection of non-ideal behaviors.

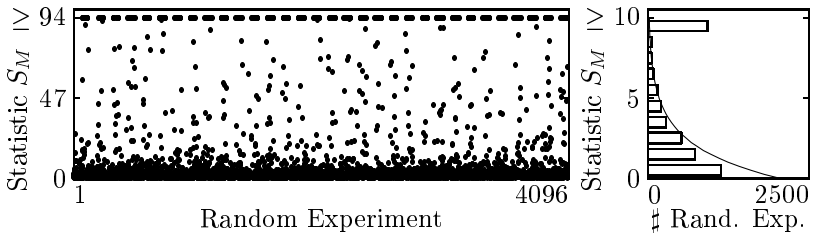

Figure 5.6: *The* **dual** *of the block cipher* **DES** *reduced to* **5 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

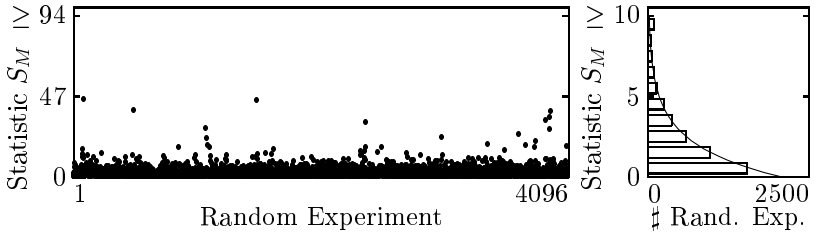

Figure 5.7: *The* **dual** *of the block cipher* **DES** *reduced to* **6 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

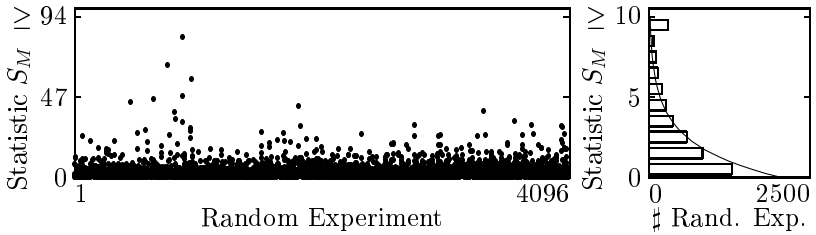

Figure 5.8: *The block cipher* **RC2** *reduced to* **5 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

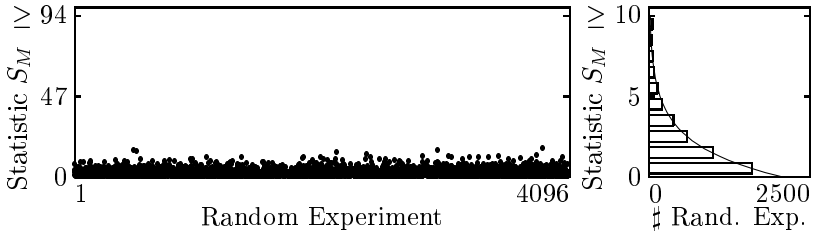

Figure 5.9: *The block cipher* **RC2** *reduced to* **6 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

The block cipher RC2 is a 16-round iterated block cipher with block length $N = 64$ and variable key length developed by Ronald L. Rivest [32] in 1989. We used RC2 with a key length $K = 128$. The structure of RC2 is not exactly as shown in Figure 5.2 since there is just before the 6-th round and just before the 12-th round and additional simple bivariate function g'.

The histogram in Figure 5.9 shows the remarkable accuracy of the chi-squared approximation to the distribution at the point where differences cannot be detected.

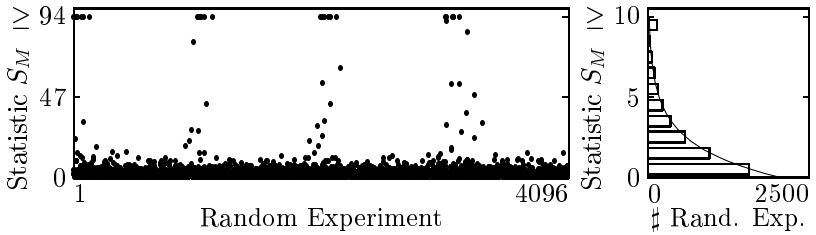

Figure 5.10: *The* **dual** *of the block cipher* **RC2** *reduced to* **6 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

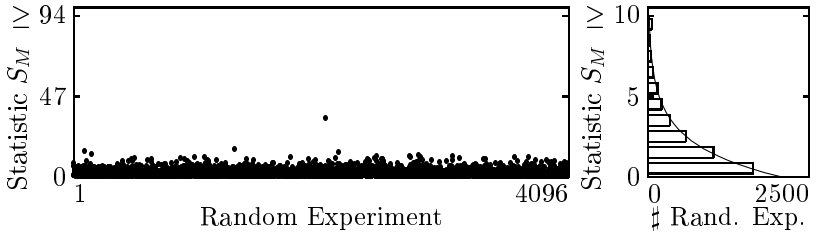

Figure 5.11: *The* **dual** *of the block cipher* **RC2** *reduced to* **7 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

The block cipher RC2 reduced to 6 rounds is an example of a block cipher that behaves according to the directed bit-dependency test for one input bit and one output bit with parameters $\tilde{M} = 1$ and $M = 2^{21}$ (Figure 5.9) significantly better than its dual cipher (Figure 5.10).

In Figure 5.11 one sees one random experiment with a surprisingly large statistic S_M. To check this random experiment, we repeated it with a much larger number of trials M. The result was an ideal behavior. It must be remembered that all these tests are statistical and anomalies like the single large statistic S_M in Figure 5.11 can be observed even if all random experiments should exhibit an ideal behavior.

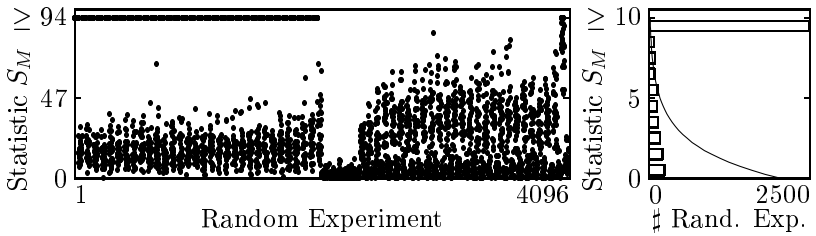

Figure 5.12: *The block cipher* **RC5-32/12/16** *reduced to* **4 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

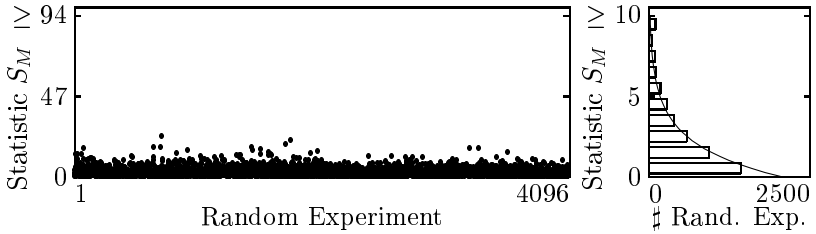

Figure 5.13: *The block cipher* **RC5-32/12/16** *reduced to* **5 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

The block cipher RC5-32/12/16 is a 12-round iterated block cipher with block length $N = 64$ and key length $K = 128$ developed by Ronald L. Rivest [31] in 1994.

Even if the left plot in Figure 5.13 does not look much different from plots showing an ideal behavior, the histogram in Figure 5.13 shows that statistics S_M larger than 3 occur a little more frequently than expected in the ideal case and statistics S_M smaller than 2 occur a little less frequently than expected in the ideal case. This indicates that there may be non-ideal behaviors just on the verge of being detected. By increasing the number of trials to $M = 2^{26}$ we could show that there are indeed non-ideal behaviors.

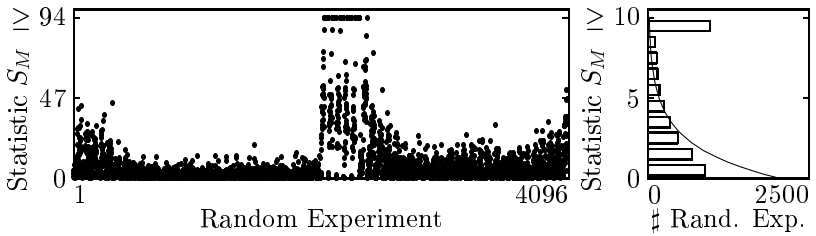

Figure 5.14: *The* **dual** *of the block cipher* **RC5-32/12/16** *reduced to* **5 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

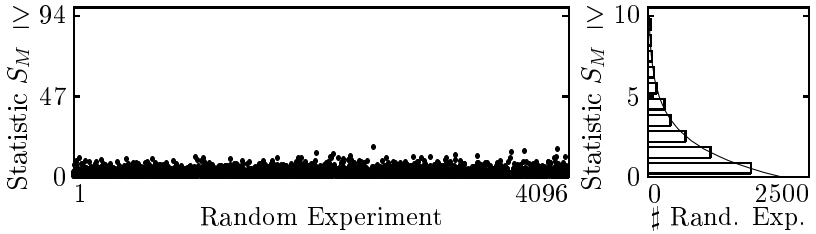

Figure 5.15: *The* **dual** *of the block cipher* **RC5-32/12/16** *reduced to* **6 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

The block cipher RC5-32/12/16 reduced to 5 rounds is another example of a block cipher that behaves according to the directed bit-dependency test for one input bit and one output bit with parameters $\tilde{M} = 1$ and $M = 2^{21}$ (Figure 5.13) significantly better than its dual cipher (Figure 5.14).

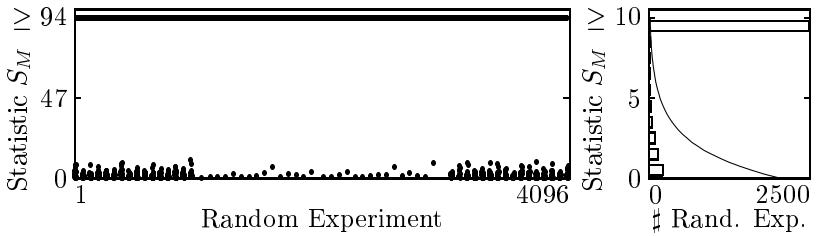

Figure 5.16: *The block cipher* **IDEA** *reduced to* **0 round** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

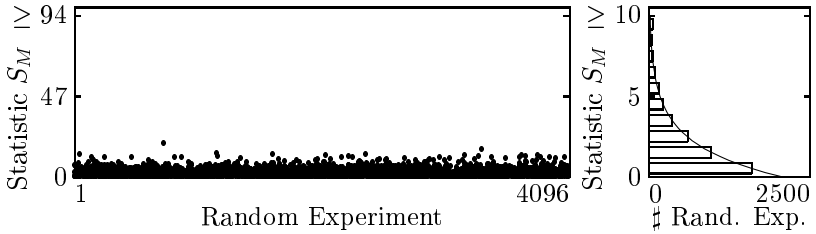

Figure 5.17: *The block cipher* **IDEA** *reduced to* **1 round** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

The block cipher IDEA (International Data Encryption Algorithm) is a 8-round iterated block cipher with block length $N = 64$ and key length $K = 128$ developed by Xuejia Lai and James L. Massey [16, 17, 15] in 1990.

The simulation results obtained by analyzing the reduced versions of the block cipher IDEA by the directed bit-dependency test for one input bit and one output bit with parameter $\tilde{M} = 1$ are quite surprising. The block cipher IDEA reduced to a single round exhibited ideal behavior when analyzed by the directed bit-dependency test for one input bit and one output bit with parameters $\tilde{M} = 1$ and $M = 2^{21}$ (Figure 5.17). Moreover, analyzing its dual cipher (Figure 5.19) and increasing the

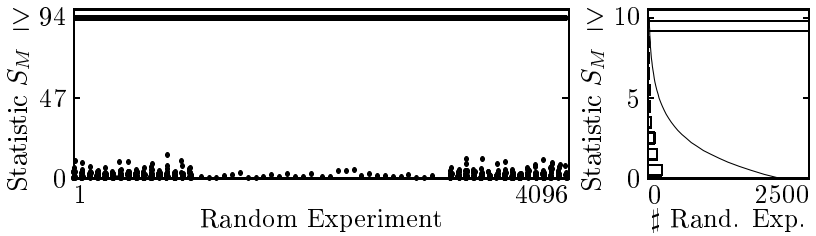

Figure 5.18: *The* **dual** *of the block cipher* **IDEA** *reduced to* **0 round** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

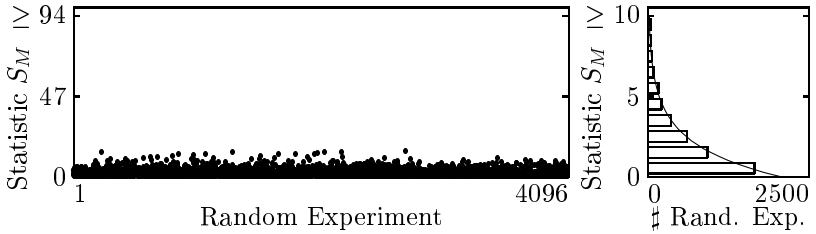

Figure 5.19: *The* **dual** *of the block cipher* **IDEA** *reduced to* **1 round** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

number of trials to $M = 2^{26}$ gave simulation results similar to those shown in Figure 5.17. However, since the block cipher IDEA reduced to 1 round has a key space of cardinality 2^{128} and a block length $N \geq 2$, we can use the result given on page 68 to conclude that the directed bit-dependency test for one input bit and one output bit with parameter $\tilde{M} = 1$ behaves differently when analyzing the block cipher IDEA reduced to 1 round as opposed to when analyzing the complete block cipher. Obviously, this difference in behavior is remarkably small for IDEA reduced to 1 round.

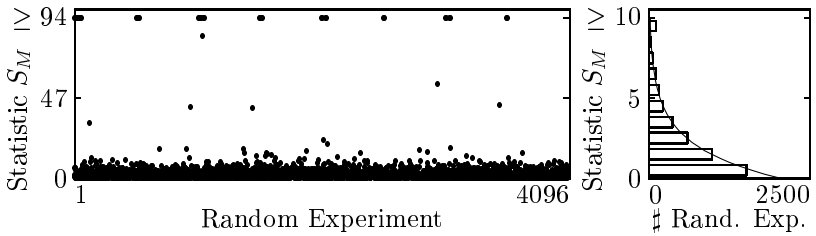

Figure 5.20: *The block cipher* **SAFER SK-128** *reduced to* **2 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

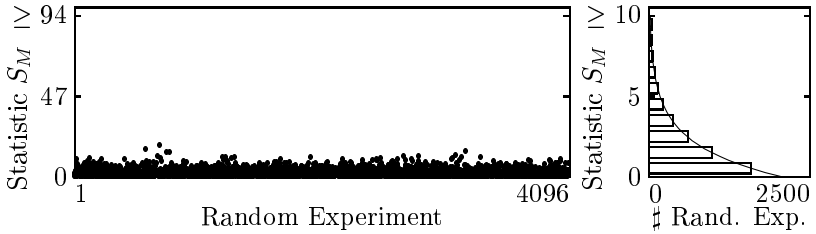

Figure 5.21: *The block cipher* **SAFER SK-128** *reduced to* **3 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

The block cipher SAFER SK-128 (Secure And Fast Encryption Routine) is a 10-round iterated block cipher with block length $N = 64$ and key length $K = 128$ developed by James L. Massey [20, 21, 12, 22] in 1993.

Figure 5.20 and Figure 5.22 show a slight difference in behavior depending on whether the block cipher SAFER SK-128 reduced to 2 rounds or its dual cipher is analyzed by the directed bit-dependency test for one input bit and one output bit with parameters $\tilde{M} = 1$ and $M = 2^{21}$. Figure 5.20 shows more statistics $S_M \geq 94$ than does Figure 5.22.

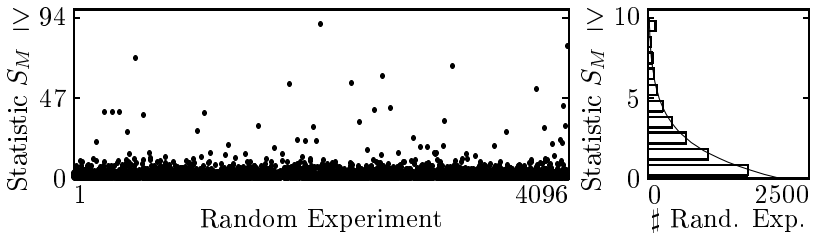

Figure 5.22: *The* **dual** *of the block cipher* **SAFER SK-128** *reduced to* **2 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

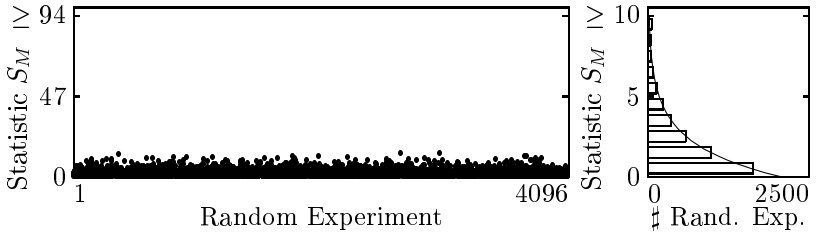

Figure 5.23: *The* **dual** *of the block cipher* **SAFER SK-128** *reduced to* **3 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

For both the block cipher SAFER SK-128 reduced to 2 rounds and its dual cipher, all the input-bit/output-bit combinations that resulted in a non-ideal behavior when analyzed by the directed bit-dependency test for one input bit and one output bit with parameters $\tilde{M} = 1$ and $M = 2^{21}$ are combinations of a least significant bit of a byte in the input with a least significant bit of a byte in the output.

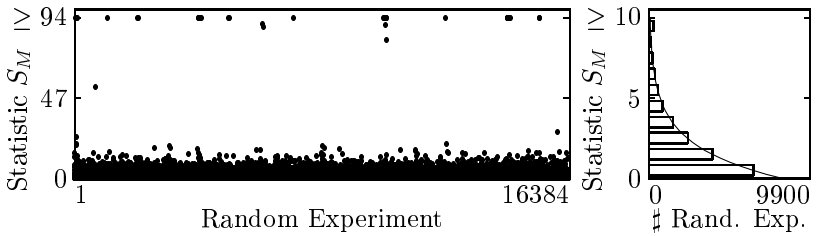

Figure 5.24: *The block cipher* **SAFER+256** *reduced to* **2 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

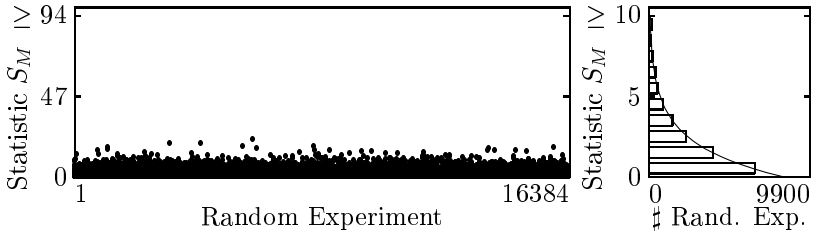

Figure 5.25: *The block cipher* **SAFER+256** *reduced to* **3 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

The block cipher SAFER+256 (Secure And Fast Encryption Routine Plus) is a 16-round iterated block cipher with block length $N = 128$ and key length $K = 256$ developed by James L. Massey, Gurgen H. Khachatrian and Melsik K. Kuregian [24] in 1998.

Since the block cipher SAFER+256 has a block length $N = 128$, there are $128 \cdot 128 = 16384$ possible combinations of a single input bit and a single output bit and hence there are 16384 random experiments shown in Figures 5.24, 5.25, 5.26 and 5.27.

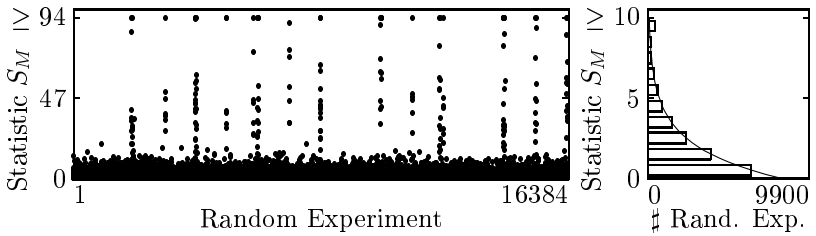

Figure 5.26: *The* **dual** *of the block cipher* **SAFER+256** *reduced to* **2 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

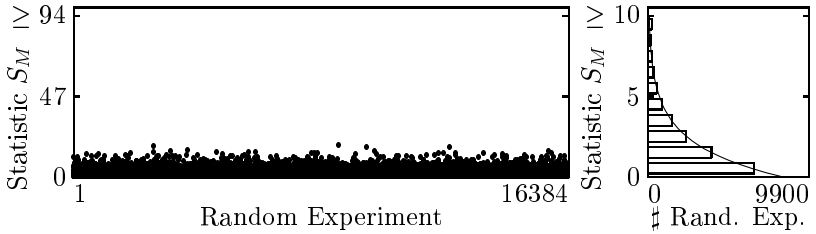

Figure 5.27: *The* **dual** *of the block cipher* **SAFER+256** *reduced to* **3 rounds** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

Figure 5.24 and Figure 5.26 show a slight difference in behavior depending on whether the block cipher SAFER+256 reduced to 2 rounds or its dual cipher is analyzed by the directed bit-dependency test for one input bit and output bit with parameters $\tilde{M} = 1$ and $M = 2^{21}$.

For both the block cipher SAFER+256 reduced to 2 rounds and its dual cipher, all the input-bit/output-bit combinations that resulted in a non-ideal behavior when analyzed by the directed bit-dependency test for one input bit and one output bit with parameters $\tilde{M} = 1$ and $M = 2^{21}$ are combinations of a least significant bit of a byte in the input with a least significant bit of a byte in the output.

5.4 Simulation Results, Part II

In the last section we showed the simulations results we obtained by analyzing different block ciphers by the directed bit-dependency test for one input bit and one output bit with parameter $\tilde{M} = 1$. What happens when we choose the parameter $\tilde{M}$ to be 2 and to be 3 is shown in Figure 5.28 and Figure 5.29, respectively. For both tests the block cipher IDEA reduced to 1 round did show non-ideal behaviors.

In the last section we showed both, the behavior of a block cipher and the behavior of its dual cipher. The next derived block ciphers we can analyze are the reduced-key-space versions of a block cipher. Figure 5.30 shows the simulation result we got by analyzing the reduced-key-space versions of the block cipher IDEA reduced to 1 round obtained by keeping one bit of the secret key fixed by the directed bit-dependency test for one input bit and one output bit with parameter $\tilde{M} = 1$. Since the block cipher IDEA has a key length $K = 128$ we can choose 128 different key bits to be kept fix. The value we assign to this fixed key bit is either 0 or 1. Therefore there are $2 \cdot 128 = 256$ reduced-key-space versions of the block cipher IDEA reduced to 1 round that we analyzed. For each of these reduced-key-space versions of the block cipher we did run the 4096 random experiments of the directed bit-dependency test for one input bit and one output bit with parameter $\tilde{M} = 1$. This gives the total of $1'048'576$ random experiments shown in Figure 5.30. Also for this test the block cipher IDEA reduced to 1 round did show non-ideal behaviors.

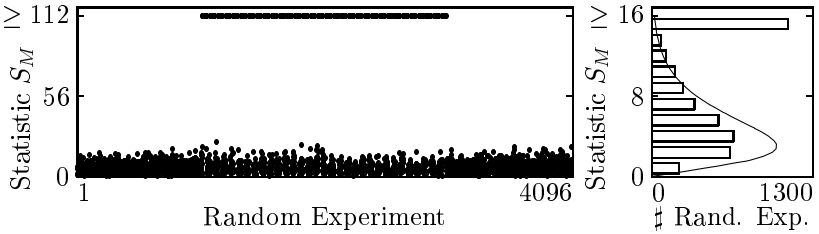

Figure 5.28: *The block cipher* **IDEA** *reduced to* **1 round** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{20}$ *trials and* $\tilde{M} = \mathbf{2}$ *executions of the body of the inner loop per trial.*

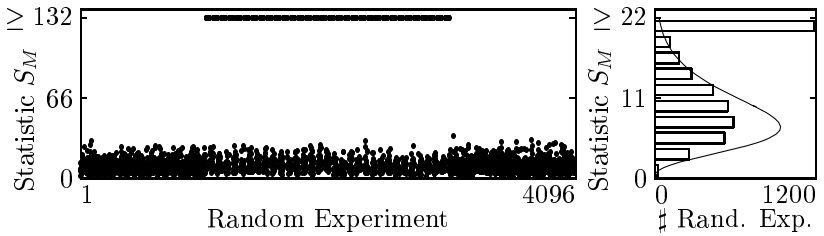

Figure 5.29: *The block cipher* **IDEA** *reduced to* **1 round** *analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = \lceil \frac{1}{3} 2^{21} \rceil$ *trials and* $\tilde{M} = \mathbf{3}$ *executions of the body of the inner loop per trial.*

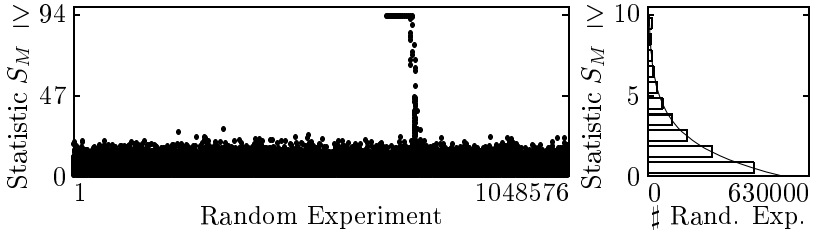

Figure 5.30: *The* **reduced-key-space** *versions of the block cipher* **IDEA** *reduced to* **1 round** *obtained by keeping* 1 *bit of the secret key fixed analyzed by the directed bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{19}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

5.5 Generalizations

The first generalization we make is instead to look at only 1 bit of the plaintext block and only 1 bit of the ciphertext block to look at s bits of the plaintext block and t bits of the ciphertext block. Figure 5.32 shows the simulation result obtained by looking at $s = 1$ bit of the plaintext block and $t = 2$ bits of the ciphertext block. Figure 5.33 shows the

simulation result obtained by looking at $s = 2$ bits of the plaintext block and $t = 1$ bit of the ciphertext block. For both cases the block cipher IDEA reduced to 1 round did show non-ideal behaviors.

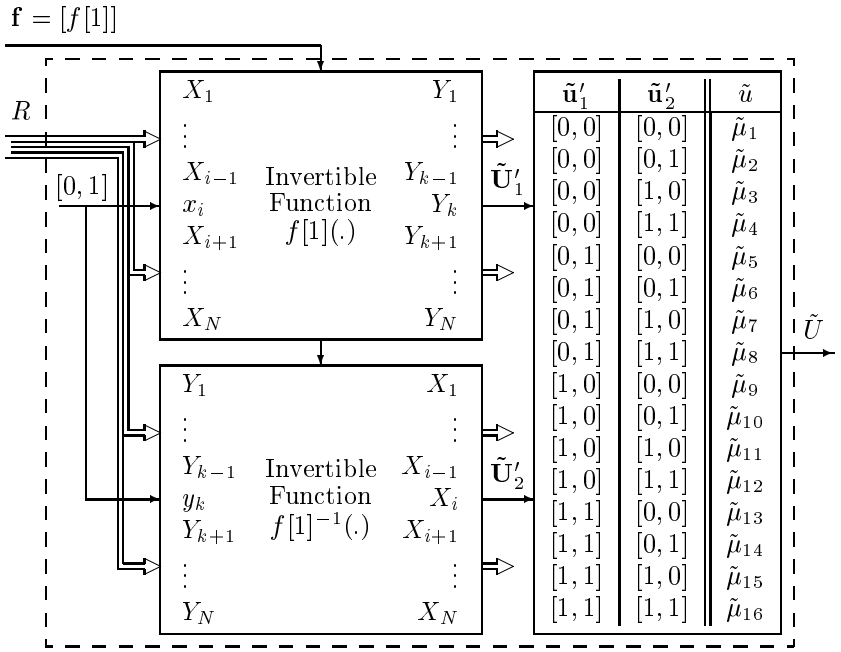

Figure 5.31: *Feature extracting algorithm for the undirected bit-dependency test of i-th input bit and k-th output bit.*

The second generalization we make is instead to analyze a given block cipher by the directed bit-dependency test to analyze the block cipher by the *undirected* bit-dependency test. Figure 5.31 shows the feature extracting algorithm for the undirected bit-dependency test for one input bit and one output bit. The random string R takes on values in the set $\mathcal{R}$ of vectors with $2N-2$ binary components according to the uniform probability distribution $P_R(r) = 2^{2-2N}$. The simulation result obtained by analyzing the block cipher IDEA reduced to 1 round by the undirected bit-dependency test for one input bit and one output bit is shown in Figure 5.34. Also for this test the block cipher IDEA reduced to 1 round did show non-ideal behaviors.

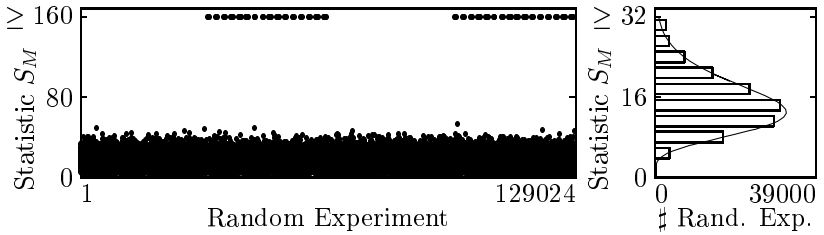

Figure 5.32: *The block cipher* **IDEA** *reduced to* **1 round** *analyzed by the directed bit-dependency test for* 1 *input bit and* **2 output bits** *by performing* $M = 2^{21}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

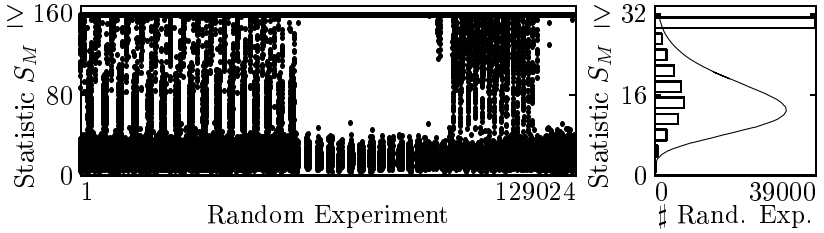

Figure 5.33: *The block cipher* **IDEA** *reduced to* **1 round** *analyzed by the directed bit-dependency test for* **2 input bits** *and* 1 *output bit by performing* $M = 2^{20}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

The third generalization we make is instead to analyze a given block cipher by a directed bit-dependency test that analyzes single invertible functions to analyze the block cipher by the directed bit-dependency test that analyzes pairs of invertible functions. Figure 5.35 shows the feature extracting algorithm for the directed bit-dependency test for one input bit and one output bit that analyzes pairs of invertible functions. The random string R takes on values in the set $\mathcal{R}$ of vectors with $N-1$ binary components according to the uniform probability distribution $P_R(r) = 2^{1-N}$. The simulation result obtained for this test did not show any new behavior of the analyzed block ciphers.

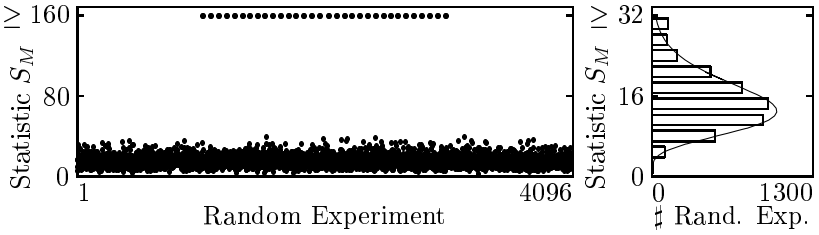

Figure 5.34: *The block cipher* **IDEA** *reduced to* **1 round** *analyzed by the* **undirected** *bit-dependency test for* 1 *input bit and* 1 *output bit by performing* $M = 2^{20}$ *trials and* $\tilde{M} = 1$ *execution of the body of the inner loop per trial.*

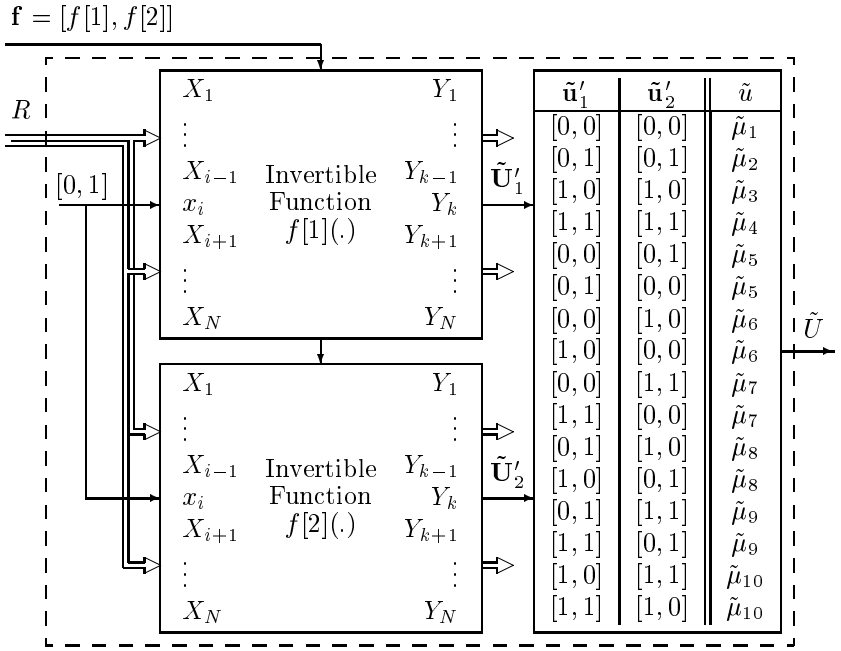

Figure 5.35: *Feature extracting algorithm for the directed bit-dependency test of i-th input bit and k-th output bit that analyzes pairs of invertible functions.*

Chapter 6

Concluding Remarks

In Chapter 2 we chose the Pearson statistic for use in the statistic former for statistical hypothesis testing. We analyzed in detail those properties of the Pearson statistic that are needed to interpret the results of statistical hypothesis testing properly. Of course, one could use a different statistic in place of the Pearson statistic. If one does so, then one must go through a similar analysis to determine how to interpret the results of the corresponding statistical hypothesis tests properly.

The optimal statistic for use in the statistic former would be the one that for

- a given set of probability vectors $\mathcal{Q}$,
- a given ideal source probability vector $\mathbf{p}$,
- a given observation sequence length M,
- a given probability of type I error α,
- and a given β^*

yields the largest set of probability vectors $\mathcal{Q}_1$ that the statistical test can detect with high probability to be different from the ideal source probability vector, where $\mathcal{Q}_1$ is as defined in (2.52).

Most results in the literature on the optimality of various statistics hold for an observation sequence length M going to infinity. These results do not add much to our work—we have already shown for the Pearson statistic that, for M going to infinity, one can detect any probability vector different from the ideal source probability vector. What would be desirable are techniques for showing the optimality of a given statistic for the given finite M that one wishes to use, but the literature is of no help here.

In Chapter 3 we stated the problem of finding an algorithm that is distinguishing for a given block cipher as the first of the two basic problems that a cryptanalyst can try to solve. Hiltgen [9] has shown that, for $N > 5$, virtually all invertible functions in $\mathcal{F}_N$ have a gate complexity larger than $2^N/5$. Therefore, for $N \geq 64$, virtually all invertible functions in $\mathcal{F}_N$ are certainly hard to compute. Suppose now that a cryptanalyst knows a computationally feasible probabilistic algorithm for analyzing an invertible function that can distinguish with high probability between an invertible function that is easy to compute and an invertible function that is hard to compute. Since all encryption functions and all decryption functions of a practical block cipher must be easy to compute, this probabilistic algorithm would be a computationally feasible probabilistic algorithm for analyzing an invertible function that is distinguishing for all practical block ciphers. This cryptanalyst would thus have solved for all practical block ciphers the first of the two basic problems stated in Theorem 3.10. There would be no practical block cipher for which we could conclude that this cryptanalyst could not break this cipher. Whether such an algorithm even exists is an open question, but if it does, it would open up the possibility of finding a probabilistic algorithm for analyzing an invertible function that breaks all practical block ciphers.

In Chapter 4 we presented the framework for the statistical testing of block ciphers and in Chapter 5 we described and analyzed tests that can be performed inside this framework. Also in Chapter 5 we presented simulation results for various block ciphers to show the usefulness of the different tests. In developing new tests, the tests in Chapter 5 can serve as examples of how to design such tests carefully and scientifically.

Appendix A

Proofs

A.1 Proof of Theorem 2.1

Let $\bar{\mathcal{U}}_0$ and $\bar{\mathcal{U}}_1$ denote decision regions corresponding to the probabilities of type I and type II error $\bar{\alpha}$ and $\bar{\beta}$, respectively, and let $\mathcal{U}_0$ and $\mathcal{U}_1$ denote be the "Neyman-Pearson" regions defined in (2.7) and (2.8) with type I and type II error α and β, respectively. With (2.5) and with (2.6) we get

$$\alpha - \bar{\alpha} = \sum_{\mathbf{u} \in \mathcal{U}_1} P_{\mathbf{U}|H_0}(\mathbf{u}) - \sum_{\mathbf{u} \in \bar{\mathcal{U}}_1} P_{\mathbf{U}|H_0}(\mathbf{u}), \tag{A.1}$$

$$\bar{\beta} - \beta = \sum_{\mathbf{u} \in \bar{\mathcal{U}}_0} P_{\mathbf{U}|H_1}(\mathbf{u}) - \sum_{\mathbf{u} \in \mathcal{U}_0} P_{\mathbf{U}|H_1}(\mathbf{u}). \tag{A.2}$$

We now define two regions

$$\mathcal{B}_1 = \mathcal{U}_1 \setminus \bar{\mathcal{U}}_1, \tag{A.3}$$

$$\mathcal{B}_0 = \bar{\mathcal{U}}_1 \setminus \mathcal{U}_1. \tag{A.4}$$

Since $\mathcal{U}_1 = \mathcal{U} \setminus \mathcal{U}_0$ and since $\bar{\mathcal{U}}_1 = \mathcal{U} \setminus \bar{\mathcal{U}}_0$ it follows that

$$\mathcal{B}_1 = \bar{\mathcal{U}}_0 \setminus \mathcal{U}_0, \tag{A.5}$$

$$\mathcal{B}_0 = \mathcal{U}_0 \setminus \bar{\mathcal{U}}_0. \tag{A.6}$$

With these definitions we rewrite (A.1) and (A.2) to obtain

$$\alpha - \bar{\alpha} = \sum_{\mathbf{u}\in\mathcal{B}_1} P_{\mathbf{U}|H_0}(\mathbf{u}) - \sum_{\mathbf{u}\in\mathcal{B}_0} P_{\mathbf{U}|H_0}(\mathbf{u}), \tag{A.7}$$

$$\bar{\beta} - \beta = \sum_{\mathbf{u}\in\mathcal{B}_1} P_{\mathbf{U}|H_1}(\mathbf{u}) - \sum_{\mathbf{u}\in\mathcal{B}_0} P_{\mathbf{U}|H_1}(\mathbf{u}). \tag{A.8}$$

From (2.8) and (2.7) it follows that

$$\mathbf{u} \in \mathcal{B}_1 \Rightarrow \mathbf{u} \in \mathcal{U}_1 \Rightarrow P_{\mathbf{U}|H_1}(\mathbf{u}) > T P_{\mathbf{U}|H_0}(\mathbf{u}), \tag{A.9}$$

$$\mathbf{u} \in \mathcal{B}_0 \Rightarrow \mathbf{u} \in \mathcal{U}_0 \Rightarrow P_{\mathbf{U}|H_1}(\mathbf{u}) \leq T P_{\mathbf{U}|H_0}(\mathbf{u}). \tag{A.10}$$

Applying these two inequalities to (A.8) gives

$$\bar{\beta} - \beta \geq T \sum_{\mathbf{u}\in\mathcal{B}_1} P_{\mathbf{U}|H_0}(\mathbf{u}) - T \sum_{\mathbf{u}\in\mathcal{B}_0} P_{\mathbf{U}|H_0}(\mathbf{u}) \tag{A.11}$$

and making use of (A.7) results in

$$\bar{\beta} - \beta \geq T(\alpha - \bar{\alpha}). \tag{A.12}$$

For $T > 0$, we obtain the two implications in Theorem 2.1

$$\bar{\alpha} < \alpha \Rightarrow T(\alpha - \bar{\alpha}) > 0 \Rightarrow \bar{\beta} - \beta > 0 \Rightarrow \bar{\beta} > \beta, \tag{A.13}$$

$$\bar{\beta} < \beta \Rightarrow \bar{\beta} - \beta < 0 \Rightarrow T(\alpha - \bar{\alpha}) < 0 \Rightarrow \bar{\alpha} > \alpha. \tag{A.14}$$

□

A.2 Proof of Theorem 2.6

We define the random vector $\mathbf{Y}$ to be

$$\mathbf{Y} = \left[\frac{N_1 - Mp_1}{\sqrt{Mp_1}}, \frac{N_2 - Mp_2}{\sqrt{Mp_2}}, \ldots, \frac{N_J - Mp_J}{\sqrt{Mp_J}}\right] \tag{A.15}$$

and write (2.25) as the inner product

$$S_M = \mathbf{Y}\mathbf{Y}^\top. \tag{A.16}$$

Let the vectors $\mathbf{e}_1$, $\mathbf{e}_2$, ..., $\mathbf{e}_J$ be an orthonormal basis for the vector space $\Re^J$, i.e.,

$$\mathbf{e}_i \mathbf{e}_k^\top = \begin{cases} 0 & \text{if } i \neq k \\ 1 & \text{if } i = k, \end{cases} \tag{A.17}$$

with the vector $\mathbf{e}_J$ having the special form

$$\mathbf{e}_J = [\sqrt{p_1}, \sqrt{p_2}, \ldots, \sqrt{p_J}\,]. \tag{A.18}$$

Let A be the invertible matrix

$$\mathsf{A} = \left[\mathbf{e}_1^\top, \mathbf{e}_2^\top, \ldots, \mathbf{e}_J^\top\right] \tag{A.19}$$

and let $\mathbf{X}$ be the random vector

$$\mathbf{X} = \mathbf{Y}\mathsf{A}. \tag{A.20}$$

Using (A.20), the fact that $\mathsf{A}^\top = \mathsf{A}^{-1}$ (i.e., that A is an orthogonal matrix), and (A.16) gives

$$\mathbf{X}\mathbf{X}^\top = \mathbf{Y}\mathsf{A} \cdot \mathsf{A}^\top \mathbf{Y}^\top = \mathbf{Y}\mathbf{Y}^\top = S_M. \tag{A.21}$$

The last component of $\mathbf{X} = [X_1, X_2, \ldots, X_J]$ is

$$X_J = \mathbf{Y}\mathbf{e}_J^\top = \sum_{j=1}^{J} \frac{N_j - Mp_j}{\sqrt{Mp_j}} \sqrt{p_j} = \frac{M}{\sqrt{M}} - \sqrt{M} = 0 \tag{A.22}$$

and therefore, according to (A.21),

$$S_M = X_1^2 + X_2^2 + \ldots + X_{J-1}^2. \tag{A.23}$$

We now show that, for $M \to \infty$, the random variables X_1, X_2, ..., X_{J-1} are independent and identically distributed random variables having a normal probability distribution with zero mean and unit variance and therefore, according to (2.36), S_M has a chi-squared probability distribution function with $J-1$ degrees of freedom. We show this in two steps. In the first step, we show that the random variables X_1, X_2, ..., X_{J-1} have zero mean and unit variance and that they are uncorrelated. This holds for any positive integer M. Then in the second step, we show that, for $M \to \infty$, the random variables X_1, X_2, ..., X_{J-1} have a normal probability distribution and thus are independent.

Since $\mathbf{q} = \mathbf{p}$, the random vector $\mathbf{N}$ has the probability distribution $P_{\mathbf{N}}(\mathbf{n}) = \frac{M!}{n_1!n_2!\cdots n_J!} p_1^{n_1} p_2^{n_2} \cdots p_J^{n_J}$. The probability distribution of the j-th component of the random vector $\mathbf{N}$ is the binomial probability distribution with parameters M and p_j, i.e.,

$$P_{N_j}(n_j) = \frac{M!}{n_j!(M-n_j)!} p_j^{n_j} (1-p_j)^{M-n_j}, \tag{A.24}$$

which has mean

$$\mathrm{E}[N_j] = Mp_j \tag{A.25}$$

and variance $\mathrm{Var}[N_j] = Mp_j(1-p_j)$. Using $\mathrm{Var}[N_j] = \mathrm{E}\left[N_j^2\right] - \mathrm{E}[N_j]^2$ gives

$$\mathrm{E}\left[N_j^2\right] = Mp_j + M(M-1)p_j^2. \tag{A.26}$$

In the following, we assume that $i \neq k$. The joint probability distribution of a pair N_i and N_k of components of the random vector $\mathbf{N}$ is the multinomial probability distribution

$$P_{N_i,N_k}(n_i,n_k) = \frac{M!}{n_i!n_k!(M-n_i-n_k)!} p_i^{n_i} p_k^{n_k} (1-p_i-p_k)^{M-n_i-n_k} \tag{A.27}$$

which has mean

$$\mathrm{E}[N_i N_k] = \sum_{n_i=0}^{M} \sum_{n_k=0}^{M-n_i} n_i n_k P_{N_i,N_k}(n_i,n_k). \tag{A.28}$$

Since $n_i = M$ implies that $n_k = 0$ and hence that the product $n_i n_k$ is zero, we can take $M-1$ as the upper limit of the outer sum. Inserting (A.27) in (A.28) gives

$$\mathrm{E}[N_i N_k] = \sum_{n_i=0}^{M-1} n_i \frac{M!}{n_i!(M-n_i)!} p_i^{n_i} (1-p_i)^{M-n_i} \cdot \underbrace{\sum_{n_k=0}^{M-n_i} n_k \frac{(M-n_i)!}{n_k!(M-n_i-n_k)!} \left(\frac{p_k}{1-p_i}\right)^{n_k} \left(1-\frac{p_k}{1-p_i}\right)^{M-n_i-n_k}}_{(M-n_i)\frac{p_k}{1-p_i}}. \tag{A.29}$$

The expression over the brace in (A.29) is the formula for the mean of a random variable having a binomial distribution with parameters $M-n_i$ and $p_k/(1-p_i)$, which is $(M-n_i)p_k/(1-p_i)$ as indicated above. Thus,

$$\mathrm{E}[N_i N_k] = \underbrace{\sum_{n_i=0}^{M-1} n_i \frac{(M-1)!}{n_i!(M-1-n_i)!} p_i^{n_i} (1-p_i)^{M-1-n_i}}_{(M-1)p_i} \frac{M!}{(M-1)!} p_k. \tag{A.30}$$

The expression over the brace in (A.30) is the formula for the mean of a random variable having a binomial distribution with parameters $M-1$ and p_i, which is $(M-1)p_i$. We thus have

$$\mathrm{E}[N_i N_k] = M(M-1)p_i p_k. \tag{A.31}$$

Let B be the diagonal matrix

$$\mathsf{B} = \begin{bmatrix} \sqrt{p_1} & 0 & \cdots & 0 \\ 0 & \sqrt{p_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{p_J} \end{bmatrix} \tag{A.32}$$

whose inverse is the matrix

$$\mathsf{B}^{-1} = \begin{bmatrix} 1/\sqrt{p_1} & 0 & \cdots & 0 \\ 0 & 1/\sqrt{p_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/\sqrt{p_J} \end{bmatrix}. \tag{A.33}$$

Combining (A.25) with (A.18) and (A.32) gives

$$\mathrm{E}[\mathbf{N}] = M\mathbf{e_J}\mathsf{B}, \tag{A.34}$$

combining (A.26) and (A.31) with (A.18) and (A.32) gives

$$\mathrm{E}\left[\mathbf{N}^\top \mathbf{N}\right] = M\mathsf{B}\mathsf{B} + M(M-1)\mathsf{B}\mathbf{e}_J^\top \mathbf{e}_J \mathsf{B}, \tag{A.35}$$

and combining (A.15) with (A.18) and (A.33) gives

$$\mathbf{Y} = \frac{1}{\sqrt{M}} \mathbf{N}\mathsf{B}^{-1} - \sqrt{M}\mathbf{e_J}. \tag{A.36}$$

The mean of the random vector $\mathbf{Y}$ is

$$\mathrm{E}\left[\mathbf{Y}\right] = \frac{1}{\sqrt{M}} M \mathbf{e_J} \mathsf{B}\mathsf{B}^{-1} - \sqrt{M}\mathbf{e_J} = \mathbf{0}, \tag{A.37}$$

the all zero vector $\mathbf{0}$. To compute the mean of the random matrix $\mathbf{Y}^\top\mathbf{Y}$, we make use of (A.36), of the fact that ${\mathsf{B}^{-1}}^\top = \mathsf{B}^{-1}$, of (A.35), of (A.34) and of the fact that $\mathsf{B}^\top = \mathsf{B}$, to obtain

$$\mathrm{E}\left[\mathbf{Y}^\top\mathbf{Y}\right] = \mathrm{E}\left[\left(\frac{1}{M}\mathsf{B}^{-1}\mathbf{N}^\top - \sqrt{M}{\mathbf{e_J}}^\top\right)\left(\frac{1}{\sqrt{M}}\mathbf{N}\mathsf{B}^{-1} - \sqrt{M}\mathbf{e_J}\right)\right], \tag{A.38}$$

$$\begin{aligned}\mathrm{E}\left[\mathbf{Y}^\top\mathbf{Y}\right] &= \frac{1}{\sqrt{M}}\mathsf{B}^{-1}\,\mathrm{E}\left[\mathbf{N}^\top\mathbf{N}\right]\mathsf{B}^{-1} - \mathsf{B}^{-1}\,\mathrm{E}\left[\mathbf{N}\right]^\top\mathbf{e}_J \\ &\quad - \mathbf{e}_J^\top\,\mathrm{E}\left[\mathbf{N}\right]\mathsf{B}^{-1} + M\mathbf{e}_J^\top\mathbf{e}_J,\end{aligned} \tag{A.39}$$

$$\mathrm{E}\left[\mathbf{Y}^\top\mathbf{Y}\right] = I + (M-1)\mathbf{e}_J^\top\mathbf{e}_J - M\mathbf{e}_J^\top\mathbf{e}_J - M\mathbf{e}_J^\top\mathbf{e}_J + M\mathbf{e}_J^\top\mathbf{e}_J, \tag{A.40}$$

$$\mathrm{E}\left[\mathbf{Y}^\top\mathbf{Y}\right] = I - \mathbf{e}_J^\top\mathbf{e}_J, \tag{A.41}$$

where I denotes the identity matrix. From (A.20) and (A.37), it follows that

$$\mathrm{E}\left[\mathbf{X}\right] = \mathrm{E}\left[\mathbf{Y}\right]\mathsf{A} = \mathbf{0}\mathsf{A} = \mathbf{0} \tag{A.42}$$

and hence the random variables X_1, X_2, ..., X_{J-1} have zero mean. From (A.20) and (A.41) and from the fact that $\mathsf{A}^\top = \mathsf{A}^{-1}$, it follows that

$$\mathrm{E}\left[\mathbf{X}^\top\mathbf{X}\right] = \mathsf{A}^\top\,\mathrm{E}\left[\mathbf{Y}^\top\mathbf{Y}\right]\mathsf{A} = I - (\mathbf{e}_J\mathsf{A})^\top(\mathbf{e}_J\mathsf{A}) \tag{A.43}$$

or, equivalently,

$$\mathrm{E}\left[\mathbf{X}^\top\mathbf{X}\right] = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{A.44}$$

Since $\mathrm{Var}\left[X_j\right] = \mathrm{E}\left[X_j^2\right] - \mathrm{E}\left[X_j\right]^2 = 1$ for $j = 1, 2, \ldots, J-1$ and since $\mathrm{Cov}\left[X_i, X_k\right] = \mathrm{E}\left[X_i X_k\right] - \mathrm{E}\left[X_i\right]\mathrm{E}\left[X_k\right] = 0$ for $i \neq k$, it follows that

the random variables X_1, X_2, ..., X_{J-1} have unit variances and are uncorrelated.

The probability distribution of the random variable N_j is the binomial probability distribution (A.24) which has mean Mp_j and variance $Mp_j(1-p_j)$. Since $0 < p_j < 1$, the probability distribution function of the random variable N_j is, for large M, well approximated by a normal probability distribution function with mean Mp_j and variance $Mp_j(1-p_j)$, i.e.,

$$\lim_{M\to\infty} \mathrm{P}\left[\frac{N_j - Mp_j}{\sqrt{Mp_j(1-p_j)}} \leq \tau\right] = \Phi(\tau), \tag{A.45}$$

where $\Phi(\tau)$ is the probability distribution function (2.41) for a normal random variable with zero mean and unit variance. Since the random variable Y_j is according to (A.15) equal to $(N_j - Mp_j)/\sqrt{Mp_j}$, it follows that

$$\lim_{M\to\infty} \mathrm{P}\left[\frac{Y_j}{\sqrt{1-p_j}} \leq \tau\right] = \Phi(\tau), \tag{A.46}$$

i.e., for $M \to \infty$, the random variables Y_1, Y_2, ..., Y_J all have a normal probability distribution function. In [5] it is proven, that the random vector $\mathbf{Y}$ has a normal probability distribution function along the hyperplane $Y_1\sqrt{p_1} + Y_2\sqrt{p_2} + \ldots + Y_J\sqrt{p_J} = 0$. Using (A.22) and the fact that $\mathbf{X} = \mathbf{Y}\mathsf{A}$ is an orthonormal transformation, it follows that the random vector $\mathbf{X}$ has a normal probability distribution function along the hyperplane $X_J = 0$, i.e., the random vector $[X_1, X_2, \ldots, X_{J-1}]$ has a normal probability distribution function. But the random variables X_1, X_2, ..., X_{J-1} are uncorrelated and with a joint normal probability distribution function, and hence are also independent. □

A.3 Proof of Corollary 2.10

In (2.25) S_M is defined to be

$$S_M = \sum_{j=1}^{J} \frac{(N_j - Mp_j)^2}{Mp_j}. \tag{A.47}$$

From $q_{J'+1} = q_{J'+2} = \ldots = q_J = 0$ it follows that $N_{J'+1} = N_{J'+2} = \ldots = N_J = 0$ and hence

$$S_M = \sum_{j=J'+1}^{J} Mp_j + \sum_{j=1}^{J'} \frac{(N_j - Mp_j)^2}{Mp_j}. \tag{A.48}$$

The ρ defined in (2.44) is always positive so we can write

$$S_M = M(1-\rho) + \sum_{j=1}^{J'} \frac{\left[(N_j - Mp_j/\rho) + (Mp_j/\rho - Mp_j)\right]^2}{Mp_j}. \tag{A.49}$$

Expanding the squared term gives

$$S_M = M(1-\rho) + \frac{1}{\rho}\sum_{j=1}^{J'} \frac{(N_j - Mp_j/\rho)^2}{Mp_j/\rho} + \sum_{j=1}^{J'} \frac{\left[2(N_j - Mp_j/\rho) + (Mp_j/\rho - Mp_j)\right](Mp_j/\rho - Mp_j)}{Mp_j}. \tag{A.50}$$

With S'_M as defined in (2.44) we get

$$S_M = M(1-\rho) + \frac{1}{\rho}S'_M + \sum_{j=1}^{J'} (2N_j - Mp_j/\rho - Mp_j)\frac{1-\rho}{\rho}, \tag{A.51}$$

$$S_M = \frac{1}{\rho}S'_M + M(1-\rho)\left[1 + \frac{1}{\rho}\sum_{j=1}^{J'} (2N_j/M - p_j/\rho - p_j)\right]. \tag{A.52}$$

With $\sum_{j=1}^{J'} N_j = M$ and with $\sum_{j=1}^{J'} p_j = \rho$, the above equation simplifies to

$$S_M = \frac{1}{\rho}S'_M + M(1-\rho)\left[1 + \frac{1}{\rho}(2 - 1 - \rho)\right], \tag{A.53}$$

$$S_M = \frac{1}{\rho}S'_M + M\frac{1-\rho}{\rho}. \tag{A.54}$$

Equation (2.43) now follows immediately, i.e.,

$$\mathrm{P}\left[S_M \leq \tau\right] = \mathrm{P}\left[\frac{1}{\rho}S'_M + M\frac{1-\rho}{\rho} \leq \tau\right], \tag{A.55}$$

$$\mathrm{P}\left[S_M \leq \tau\right] = \mathrm{P}\left[S'_M \leq \tau\rho - M(1-\rho)\right]. \tag{A.56}$$

□

A.4 Chebychev's Inequality

Theorem A.1 (Chebychev) *Let Y be a nonnegative finite random variable. Then for any $a > 0$*

$$\mathrm{P}\left[Y \geq a\right] \leq \frac{\mathrm{E}\left[Y\right]}{a}. \tag{A.57}$$

Proof: Cf. [18]. Let $\mathcal{Y}$ denote the set of possible values of the random variable Y and let $P_Y(y)$ denote the probability distribution of Y. Because $y \geq 0$ for all $y \in \mathcal{Y}$, we have

$$\mathrm{E}\left[Y\right] = \sum_{y\in\mathcal{Y}} yP_Y(y) \geq \sum_{\substack{y\in\mathcal{Y}\\ y\geq a}} yP_Y(y) \geq \sum_{\substack{y\in\mathcal{Y}\\ y\geq a}} aP_Y(y) = a\,\mathrm{P}\left[Y \geq a\right]. \tag{A.58}$$

□

A.5 Proof of Theorem 2.13

First we consider the case where $\tau > \mathrm{E}\left[X\right]$ and use Theorem A.1 with $a = (\tau - \mathrm{E}\left[X\right])^{2n}$ and with $Y = (X - \mathrm{E}\left[X\right])^{2n}$ to obtain

$$\mathrm{P}\left[(X - \mathrm{E}\left[X\right])^{2n} \geq (\tau - \mathrm{E}\left[X\right])^{2n}\right] \leq \frac{\mathrm{E}\left[(X - \mathrm{E}\left[X\right])^{2n}\right]}{(\tau - \mathrm{E}\left[X\right])^{2n}} \tag{A.59}$$

and from this

$$\mathrm{P}\left[|X - \mathrm{E}\left[X\right]| \geq \tau - \mathrm{E}\left[X\right]\right] \leq \frac{\mathrm{E}\left[(X - \mathrm{E}\left[X\right])^{2n}\right]}{(\tau - \mathrm{E}\left[X\right])^{2n}}. \tag{A.60}$$

The part on the left can be decomposed

$$\mathrm{P}\left[|X - \mathrm{E}\left[X\right]| \geq \tau - \mathrm{E}\left[X\right]\right] = \overbrace{\mathrm{P}\left[X - \mathrm{E}\left[X\right] \geq \tau - \mathrm{E}\left[X\right]\right]}^{\mathrm{P}[X\geq\tau]} + \underbrace{\mathrm{P}\left[\mathrm{E}\left[X\right] - X \geq \tau - \mathrm{E}\left[X\right]\right]}_{\geq 0} \tag{A.61}$$

to get a lower bound on it

$$\mathrm{P}\left[X \geq \tau\right] \leq \mathrm{P}\left[\left|X - \mathrm{E}\left[X\right]\right| \geq \tau - \mathrm{E}\left[X\right]\right]. \tag{A.62}$$

Combining (A.60) and (A.62) gives

$$\mathrm{P}\left[X \geq \tau\right] \leq \frac{\mathrm{E}\left[(X - \mathrm{E}\left[X\right])^{2n}\right]}{(\tau - \mathrm{E}\left[X\right])^{2n}} \tag{A.63}$$

and with $\mathrm{P}\left[X > \tau\right] \leq \mathrm{P}\left[X \geq \tau\right]$ we get

$$\mathrm{P}\left[X > \tau\right] \leq \frac{\mathrm{E}\left[(X - \mathrm{E}\left[X\right])^{2n}\right]}{(\tau - \mathrm{E}\left[X\right])^{2n}}. \tag{A.64}$$

And finally changing the sign and adding one to both sides gives

$$1 - \mathrm{P}\left[X > \tau\right] \geq 1 - \frac{\mathrm{E}\left[(X - \mathrm{E}\left[X\right])^{2n}\right]}{(\tau - \mathrm{E}\left[X\right])^{2n}} \tag{A.65}$$

which corresponds to (2.48).

Second we consider the case where $\tau < \mathrm{E}\left[X\right]$ and use Theorem A.1 with $a = (\mathrm{E}\left[X\right] - \tau)^{2n}$ and with $Y = (X - \mathrm{E}\left[X\right])^{2n}$ to obtain

$$\mathrm{P}\left[(X - \mathrm{E}\left[X\right])^{2n} \geq (\mathrm{E}\left[X\right] - \tau)^{2n}\right] \leq \frac{\mathrm{E}\left[(X - \mathrm{E}\left[X\right])^{2n}\right]}{(\mathrm{E}\left[X\right] - \tau)^{2n}} \tag{A.66}$$

and from this

$$\mathrm{P}\left[\left|X - \mathrm{E}\left[X\right]\right| \geq \mathrm{E}\left[X\right] - \tau\right] \leq \frac{\mathrm{E}\left[(X - \mathrm{E}\left[X\right])^{2n}\right]}{(\mathrm{E}\left[X\right] - \tau)^{2n}}. \tag{A.67}$$

The part on the left can be decomposed

$$\begin{aligned}\mathrm{P}\left[\left|X - \mathrm{E}\left[X\right]\right| \geq \mathrm{E}\left[X\right] - \tau\right] = {} & \overbrace{\mathrm{P}\left[X - \mathrm{E}\left[X\right] \geq \mathrm{E}\left[X\right] - \tau\right]}^{\geq 0} \\ & + \underbrace{\mathrm{P}\left[\mathrm{E}\left[X\right] - X \geq \mathrm{E}\left[X\right] - \tau\right]}_{\mathrm{P}[X \leq \tau]}\end{aligned} \tag{A.68}$$

to get a lower bound on it

$$\mathrm{P}\left[X \leq \tau\right] \leq \mathrm{P}\left[\left|X - \mathrm{E}\left[X\right]\right| \geq \mathrm{E}\left[X\right] - \tau\right]. \tag{A.69}$$

Combining (A.67) and (A.69) gives

$$\mathrm{P}\left[X \leq \tau\right] \leq \frac{\mathrm{E}\left[(X - \mathrm{E}\left[X\right])^{2n}\right]}{(\mathrm{E}\left[X\right] - \tau)^{2n}} \tag{A.70}$$

which is equivalent to (2.49).

A.6 Proof of Lemma 3.7

Assume a computationally feasible probabilistic algorithm $\mathbb{A}_1$ for analyzing an invertible function is known that solves for the block cipher e the problem of decrypting a ciphertext block Y_1 chosen uniformly at random without asking the black box to decrypt it. The algorithm $\mathbb{A}_1$ outputs its analysis $A = [\hat{X}_1, Y_1]$ and the probability that its plaintext prediction is correct is substantially greater when given that the block cipher e was chosen than when given that the complete block cipher $\breve{e}$ was chosen, i.e.,

$$\mathrm{P}\left[\hat{X}_1 = F^{-1}(Y_1) | E_e\right] \gg \mathrm{P}\left[\hat{X}_1 = F^{-1}(Y_1) | E_{\breve{e}}\right]. \tag{A.71}$$

We modify the algorithm $\mathbb{A}_1$ just before it outputs its analysis $A = [\hat{X}_1, Y_1]$ to obtain a new algorithm $\mathbb{A}'_1$ by accessing the black box one additional time to compute $F(\hat{X}_1)$. The new algorithm $\mathbb{A}'_1$ outputs its analysis $A = D'_1$, where

$$D'_1 = \begin{cases} 0 & \text{if } F(\hat{X}_1) \neq Y_1 \\ 1 & \text{if } F(\hat{X}_1) = Y_1. \end{cases} \tag{A.72}$$

Since $D'_1 = 1$ and $\hat{X}_1 = F^{-1}(Y_1)$ are the same events, it follows from (A.71) that

$$\mathrm{P}\left[D'_1 = 1 | E_e\right] \gg \mathrm{P}\left[D'_1 = 1 | E_{\breve{e}}\right] \tag{A.73}$$

and therefore that the new probabilistic algorithm $\mathbb{A}'_1$ for analyzing an invertible function is distinguishing for the block cipher e. To obtain the new algorithm $\mathbb{A}'_1$, we had to access the black box only one additional time, which costs only a small amount of time. Since the algorithm $\mathbb{A}_1$ is computationally feasible, it follows that the new algorithm $\mathbb{A}'_1$ is also computationally feasible.

Assume a computationally feasible probabilistic algorithm $\mathbb{A}_2$ for analyzing an invertible function is known that solves for the block cipher e the problem of encrypting a plaintext block X_2 chosen uniformly at random without asking the black box to encrypt it. The algorithm $\mathbb{A}_2$ outputs its analysis $A = [X_2, \hat{Y}_2]$ and the probability that its ciphertext prediction is correct is substantially greater when given that the block cipher e was chosen than when given that the complete block cipher $\breve{e}$

was chosen, i.e.,

$$\mathrm{P}\left[\hat{Y}_2 = F(X_2)|E_e\right] \gg \mathrm{P}\left[\hat{Y}_2 = F(X_2)|E_{\check{e}}\right]. \tag{A.74}$$

We modify the algorithm $\mathbb{A}_2$ just before it outputs its analysis $A = [X_2, \hat{Y}_2]$ to obtain a new algorithm $\mathbb{A}'_2$ by accessing the black box one additional time to compute $F^{-1}(\hat{Y}_2)$. The new algorithm $\mathbb{A}'_2$ outputs its analysis $A = D'_2$, where

$$D'_2 = \begin{cases} 0 & \text{if } F^{-1}(\hat{Y}_2) \neq X_2 \\ 1 & \text{if } F^{-1}(\hat{Y}_2) = X_2. \end{cases} \tag{A.75}$$

By an argument entirely similar to the one just given, it follows that the new algorithm $\mathbb{A}'_2$ is a computationally feasible probabilistic algorithm for analyzing an invertible function that is distinguishing for the block cipher e.

Assume a computationally feasible probabilistic algorithm $\mathbb{A}_3$ for analyzing an invertible function is known that solves for the block cipher e the problem of finding an additional entry $[\hat{X}_3, \hat{Y}_3]$ in the function table of the encryption function. The algorithm $\mathbb{A}_3$ outputs its analysis $A = [\hat{X}_3, \hat{Y}_3]$ and the probability that its plaintext/ciphertext pair prediction is correct is substantially greater when given that the block cipher e was chosen than when given that the complete block cipher $\check{e}$ was chosen, i.e.,

$$\mathrm{P}\left[\hat{Y}_3 = F(\hat{X}_3)|E_e\right] \gg \mathrm{P}\left[\hat{Y}_3 = F(\hat{X}_3)|E_{\check{e}}\right]. \tag{A.76}$$

We modify the algorithm $\mathbb{A}_3$ just before it outputs its analysis $A = [\hat{X}_3, \hat{Y}_3]$ to obtain a new algorithm $\mathbb{A}'_3$ by accessing the black box one additional time to compute $F(\hat{X}_3)$. The new algorithm $\mathbb{A}'_3$ outputs its analysis $A = D'_3$, where

$$D'_3 = \begin{cases} 0 & \text{if } F(\hat{X}_3) \neq \hat{Y}_3 \\ 1 & \text{if } F(\hat{X}_3) = \hat{Y}_3. \end{cases} \tag{A.77}$$

By an argument entirely similar to those just given, it follows that the new algorithm $\mathbb{A}'_3$ is a computationally feasible probabilistic algorithm for analyzing an invertible function that is distinguishing for the block cipher e. □

A.7 Proof of Lemma 3.9

Assume a computationally feasible probabilistic algorithm $\mathbb{A}$ for analyzing an invertible function is known that solves the problem of finding the secret key Z for the block cipher e. The algorithm $\mathbb{A}$ outputs its analysis $A = \hat{Z}$ and the probability that its secret key prediction is correct is substantially greater than for a random prediction of the secret key, i.e.,

$$\mathrm{P}\left[\hat{Z} = Z\right] \gg \frac{1}{|\mathcal{Z}_e|}, \tag{A.78}$$

where $\mathcal{Z}_e$ is the key space of the block cipher e. Let g be an invertible mapping from the key space $\mathcal{Z}_e$ to the set $\{1, 2, \ldots, |\mathcal{Z}_e|\}$ of positive integers less than or equal to $|\mathcal{Z}_e|$ that is easy to compute.

Example: For the key space $\mathcal{Z}_e = \{0,1\}^K$, g could be the invertible mapping that, for a secret key z, interprets it as the binary representation of an integer, increments this integer by one, and takes this new integer as the mapped value of the secret key z. Then the all-zero secret key $[0, 0, \ldots, 0]$ would be mapped to 1 and the all-one secret key $[1, 1, \ldots, 1]$ would be mapped to 2^K.

Let $\{\mathcal{Z}_{e^1}, \mathcal{Z}_{e^2}, \ldots, \mathcal{Z}_{e^{|\mathcal{Z}_e|}}\}$ be the decomposition of the key space $\mathcal{Z}_e$ where $\mathcal{Z}_{e^1} = \{g^{-1}(1)\}$, $\mathcal{Z}_{e^2} = \{g^{-1}(2)\}$, ..., $\mathcal{Z}_{e^{|\mathcal{Z}_e|}} = \{g^{-1}(|\mathcal{Z}_e|)\}$. We now modify the algorithm $\mathbb{A}$ to obtain a new algorithm $\mathbb{A}'$ that, instead of outputting the analysis $A = \hat{Z}$, computes $W = g(\hat{Z})$ and outputs the analysis $A = W$. For this new algorithm $\mathbb{A}'$, we can compute the probability that the secret key lies in the predicted subset of the decomposition as

$$\begin{aligned} \mathrm{P}\left[Z \in \mathcal{Z}_{e^W}\right] &= \mathrm{P}\left[Z \in \{g^{-1}(W)\}\right] \\ \mathrm{P}\left[Z \in \mathcal{Z}_{e^W}\right] &= \mathrm{P}\left[Z = g^{-1}(W)\right] \\ \mathrm{P}\left[Z \in \mathcal{Z}_{e^W}\right] &= \mathrm{P}\left[Z = \hat{Z}\right]. \end{aligned} \tag{A.79}$$

Combining (A.78) and (A.79) gives

$$\mathrm{P}\left[Z \in \mathcal{Z}_{e^W}\right] \gg \frac{1}{|\mathcal{Z}_e|}. \tag{A.80}$$

But since every subset of the decomposition contains exactly one secret key in the given decomposition of the key space, we have

$$\max_{l=1}^{|\mathcal{Z}_e|} \frac{|\mathcal{Z}_{e^l}|}{|\mathcal{Z}_e|} = \frac{1}{|\mathcal{Z}_e|} \tag{A.81}$$

and therefore

$$\mathrm{P}\left[Z \in \mathcal{Z}_{e^W}\right] \gg \max_{l=1}^{|\mathcal{Z}_e|} \frac{|\mathcal{Z}_{e_l}|}{|\mathcal{Z}_e|}. \tag{A.82}$$

This makes the new algorithm $\mathbb{A}'$ a probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the block cipher e and for the decomposition $\{\mathcal{Z}_{e^1}, \mathcal{Z}_{e^2}, \ldots, \mathcal{Z}_{e^{|\mathcal{Z}_e|}}\}$ of the key space $\mathcal{Z}_e$. Since the algorithm $\mathbb{A}$ is computationally feasible and since the invertible mapping g is easy to compute, it follows that the new algorithm $\mathbb{A}'$ is also computationally feasible. □

Bibliography

[1] Eli Biham and Adi Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer-Verlag, New York, 1993. ISBN 0-387-97930-1.

[2] Ernest F. Brickell, Dorothy E. Denning, Stephen T. Kent, David P. Maher, and Walter Tuchman. SKIPJACK review, interim report, the SKIPJACK algorithm. Technical report, Sandia National Laboratories, Georgetown University, BBN Communications Corporation, AT&T, Amperif Corporation, July 1993.

[3] James D. Broffitt and Ronald H. Randles. A power approximation for the chi-square goodness-of-fit test: Simple hypothesis case. *Journal of the American Statistical Association*, 72(359):604–607, September 1977.

[4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley series in telecommunication. John Wiley & Sons, Inc., New York, 1991. ISBN 0-471-06259-6.

[5] Harald Cramér, editor. *Mathematical Methods of Statistics*. Princeton Landmarks in Mathematics. Princeton University Press, nineteenth printing edition, 1999. First printed 1946, ISBN 0-691-00547-8.

[6] Wilbur B. Davenport and William L. Root. *An Introduction to the Theory of Random Signals and Noise*. Mc Graw-Hill Book Company, Inc., Lincoln Laboratory, Massachusetts Institute of Technology, 1958.

[7] Churchill Eisenhart. The power function of the χ^2*test*. *Bulletin of the American Mathematical Society*, 44:32, 1938.

[8] Michiel Hazewinkel, editor. *Encyclopaedia of Mathematics*, volume 1–10. Kluwer Academic Publishers, Dordrecht, Holland, 1987. ISBN 1-55608-010-7.

[9] Alain P.L. Hiltgen. *Cryptographically Relevant Contributions to Combinational Complexity Theory*, volume 3 of *ETH Series in Information Processing (Ed. J.L.Massey)*. Hartung-Gorre Verlag Konstanz, 1994. ISBN 3-89191-745-7.

[10] Maurice Kendall and Alan Stuart. *The Advanced Theory of Statistics, Interference and Relationship*, volume 2. Charles Griffin & Company Limited, fourth edition, 1979. ISBN 0-85264-255-5.

[11] Auguste Kerckhoffs. La cryptographie militaire. *Journal des Sciences Militaires*, 9:5–38,161–191, January and February 1883.

[12] Lars R. Knudsen. A key-schedule weakness in SAFER K-64. In Don Coppersmith, editor, *Advances in Cryptology—CRYPTO'95*, pages 274–286, Berlin, August 1995. Springer-Verlag. Lecture Notes in Computer Science 963, ISBN 3-540-60221-6.

[13] Kenneth Koehler. A general formula for moments of the pearson goodness-of-fit statistic for alternatives. *Biometrika: a journal for the statistical study of biological problems*, 66(2):397–399, 1979. London.

[14] Samuel Kotz and Norman L. Johnson, editors. *Encyclopedia of Statistical Sciences*, volume 1–9, Suppl. John Wiley & Sons, Inc., New York, 1982–1989. ISBN 0-471-05546-8.

[15] Xuejia Lai. *On the Design and Security of Block Ciphers*, volume 1 of *ETH Series in Information Processing (Ed. J.L.Massey)*. Hartung-Gorre Verlag Konstanz, 1992. ISBN 3-89191-573-X.

[16] Xuejia Lai and James L. Massey. A proposal for a new block encryption standard. In G. Goos and J. Hartmanis, editors, *Advances in Cryptology—EUROCRYPT'90*, pages 389–404. Springer-Verlag, 1990. Lecture Notes in Computer Science 473, ISBN 3-540-53587-X.

[17] Xuejia Lai, James L. Massey, and Sean Murphy. Markov ciphers and differential cryptanalysis. In D.W. Davies, editor, *Advances in Cryptology—EUROCRYPT'91*, pages 17–38. Springer-Verlag, 1991. Lecture Notes in Computer Science 547, ISBN 3-540-54620-0.

[18] John W. Lamperti. *Probability: A Survey of the Mathematical Theory*. Wiley series in probability and statistics. John Wiley & Sons, Inc., New York, second edition, 1996. ISBN 0-471-15407-5.

[19] Erich L. Lehmann. *Testing Statistical Hypotheses*. Springer texts in statistics. Springer-Verlag, New York, second edition, 1997. ISBN 0-387-94919-4.

[20] James L. Massey. SAFER K-64: A byte-oriented block ciphering algorithm. In Ross Anderson, editor, *Fast Software Encryption*, pages 1–17, Heidelberg and New York, December 1993. Cambridge Security Workshop, Cambridge, U.K., Springer-Verlag. Lecture Notes in Computer Science 809, ISBN 3-540-58108-1.

[21] James L. Massey. SAFER K-64: One year later. In Bart Preneel, editor, *Fast Software Encryption*, pages 212–241, Heidelberg and New York, December 1994. K. U. Leuven Workshop on Cryptographic Algorithms, Springer-Verlag. Lecture Notes in Computer Science 1008, ISBN 3-540-60590-8.

[22] James L. Massey. Announcement of a strengthened key schedule for the cipher SAFER. e-mail, September 1995.

[23] James L. Massey. Cryptography: Fundamentals and applications. copies of transparencies, Zürich, 1995. advanced technology seminars.

[24] James L. Massey, Gurgen H. Khachatrian, and Melsik K. Kuregian. Nomination of SAFER+ as candidate algorithm for the advanced encryption standard (AES). Internet: http://www.cylink.com/SAFER, June 1998.

[25] Alfred J. Menezes, Paul van Oorschot, and Scott Vanstone. *Handbook of Applied Cryptography*. Discrete mathematics and its applications. CRC Press, Inc., New York, 1996. ISBN 0-8493-8523-7.

[26] NBS. Guidelines for implementing and using the NBS data encryption standard. Federal Information Processing Standards Publication FIPS PUB 74, U.S. Department of Commerce, National Bureau of Standards, April 1981.

[27] J. Neyman and E.S. Pearson. On the use and interpretation of certain test criteria for purposes of statistical inference. *Bio-*

metrika: a journal for the statistical study of biological problems, 20A:175–240,263–294, 1928. Art. III.

[28] NIST. Announcing the standard for security requirements for cryptographic modules. Federal Information Processing Standards Publication FIPS PUB 140-1, National Institute of Standards and Technology, January 1994.

[29] P. B. Patnaik. The non-central χ^2- and F-distributions and their applications. *Biometrika: a journal for the statistical study of biological problems*, 36:202–232, 1949. London.

[30] Karl Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, 50(5):157–175, July 1900.

[31] Ronald L. Rivest. The RC5 encryption algorithm. In Bart Preneel, editor, *Fast Software Encryption*, pages 86–96, Heidelberg and New York, December 1994. K. U. Leuven Workshop on Cryptographic Algorithms, Springer-Verlag. Lecture Notes in Computer Science 1008, ISBN 3-540-60590-8.

[32] Ronald L. Rivest. A description of the RC2 encryption algorithm. Internet: http://www.rsa.com, March 1998. Network Working Group, Request for Comments: 2268.

[33] Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656–715, Oct. 1949.

[34] Gustavus J. Simmons, editor. *Contemporary Cryptology: the Science of Information Integrity*. IEEE Press, New York, 1992. ISBN 0-87942-277-7.