Block Ciphers - Analysis, Design and Applications

Lars Ramkilde Knudsen

July 1, 1994

Contents

Acknowledgements

First of all, I would like to thank my supervisor, Ivan Bjerre Damgård for his support during my two and half years as a Ph.D. student. For always patiently listening to and commenting on my ideas and research topics and for not laughing whenever I "broke" the DES algorithm. Also, thank you Ivan for suggesting me to study differential cryptanalysis for my Masters thesis.

A very special thank you to the referee Bart Preneel for many comments that improved this thesis and for answering my many questions about hash functions. Also thank you Bart and Ria for your hospitality during my visit in Leuven.

My interest in cryptography started in a course given by Peter Landrock. We were given a sheet of ciphertext, some plaintext encrypted using the Vigenere cipher. I was very amazed that one week later, we implemented an attack, which on input this ciphertext output the plaintext a few seconds later, without knowledge of neither the key nor the plaintext in advance. So, thank you Peter for lighting my cryptographic candle and for your humour.

Also a big thank you to my colleagues, Torben Pedersen, Lidong Chen, and Jørgen Brandt of Aarhus University for many helpful comments and discussions and a big thanks to Torben for proof reading.

A special thank you to my dear co-authors, Kaisa Nyberg, Xuejia Lai, and Luke O'Connor for working with me on those specific projects and for many helpful comments and discussions in general and to Don Coppersmith for valuable comments on one of the papers.

I would like to thank the people at the ETH in Zürich, Kenny Paterson, Shirlei Serconek, Atsushi Fujioka, Gerhard Krämer, and last but not least James L. Massey. Thank you Jim for allowing me to stay at the ETH and

for your and your wife Lis' big hospitality during my stay in Switzerland. Thank you Lis for the "wild card" to the ATS seminar 1993.

Also thank you Tor Helleseth for arranging the Nordic crypto course in June 1992, and to Eli Biham, Kwangjo Kim, Willi Meier, Yuliang Zheng, and B. Schneier [105] for helpful comments and discussions.

Big thanks to D.Å.T. and the boys for *having me on tonight*, to the Rolling Stones, Chuck Berry, and Jack D. for general inspiration. *It's only cryptography, but I like it.*

Finally, thank you Heather for being; the most lovely and loving person, I have ever met.

Århus, July 1, 1994
Lars Ramkilde Knudsen

Abstract

In this thesis we study cryptanalysis, applications and design of secret key block ciphers. In particular, the important class of *Feistel ciphers* is studied, which has a number of rounds, where in each round one applies a cryptographically weak function.

Applications

The main application of block ciphers is that of encryption. We study the available modes of operation for encryption, introduce a new taxonomy for attacks on block ciphers and derive a new theoretical upper bound for attacks on block ciphers. Also another important application of block ciphers is studied; as building blocks for cryptographic hash functions. Finally we examine how to use block ciphers as building blocks in the design of digital signature schemes. In particular we analyse Merkle's proposed scheme and show that under suitable and reasonable conditions, Merkle's scheme is secure and practical.

Cryptanalysis

We study the most important known attacks on block ciphers, linear cryptanalysis and differential cryptanalysis and introduce a new attack based on simple relations. Differential cryptanalysis makes use of so-called *differentials* (A, B), i.e., a pair of plaintexts with difference A, which after a certain number of rounds result in a difference B with a non-negligible probability. This fact can be used to derive (parts of) the secret key. Ideas of how to

find the best such differentials are given. Also it is shown that higher order differentials, where more than two plaintexts are considered at a time, and partial differentials, where only a part of (A, B) can be predicted, both have useful applications. The above attacks and our new methods of attacks on block ciphers, are applied to the specific block ciphers, DES, LOKI'91, s^2-DES, xDES^1 and xDES^2.

Attacks on hash functions based on block ciphers are studied and new attacks on a large class of hash functions based on a block cipher, including three specific proposed schemes, are given. Also a fourth scheme, the AR Hash function, belonging to another class of hash functions based on block ciphers is studied. The scheme is faster than the known standard ones and was used in practice by German banks. It is shown that the scheme is completely insecure.

Design

We discuss principles for the design of secure block ciphers. For both linear and differential cryptanalysis we establish lower bounds on the complexities of success of attacks. It is furthermore shown that there exist functions, which can be used to construct block ciphers provable secure against both linear and differential attacks, the two most important attacks known to date. Furthermore we define so-called *strong key schedules*. A block cipher with a strong key schedule is shown to be secure against attacks based on simple relations and the improved immunity to other attacks is discussed. Also we give a simple design of a strong key schedule. A well-known and wide-spread way of improving the security of a block cipher is by means of multiple encryption, i.e., where a plaintext block is processed several times using the same (component) block cipher, but with different keys. We study the methods of multiple ecryption and give a new proposal of a scheme, which is provable as secure as the component block cipher using a minimum number of component keys.

Some of the work in this thesis has been written as separate articles. In cooperation with Ivan Damgård the papers [19, 20], with Kaisa Nyberg the papers [85, 86], with Xuejia Lai the papers [53, 57] and with Luke O'Connor the paper [54]. On my own the following papers [47, 48, 49, 50, 51, 52].

Chapter 1

Introduction

The thesis is organised as follows. In this chapter we give the outline of the thesis and explain the birthday paradox. In Chapter 2 an introduction to block ciphers is given. In Chapter 3 the applications of block ciphers, modes of operation for encryption, hash functions and digital signatures, are discussed. In Chapter 4 we describe the security, theoretical and practical, of block ciphers. In Chapter 5 methods of cryptanalysing block ciphers are given. The methods are applied to specific block ciphers in Chapter 6. Readers not interested in going into the details about cryptanalytic attacks may want to skip that part of this thesis. In Chapter 7 we discuss design principles of block ciphers, in particular we show how to build ciphers immune to the attacks described in previous chapters. In Chapter 8 hash functions based on block ciphers are cryptanalyzed. It is shown that a large class of these hash functions are not as secure as previously believed. In Chapter 9 we summarise our results. In the Appendix we first give a self-explanatory pictorial illustration of conventional cryptography. Furthermore we give some tedious proofs, which were left out of previous chapters and finally we give a description of the most well-known block cipher today, the Data Encryption Standard [90] and of one its successors LOKI'91 [14].

1.1 Birthday Paradox

One of the most used tricks in cryptanalysis is the use of the "birthday paradox". It is used throughout this thesis and stated explicitly here. The "paradox" has its name, because to most people it is a surprise, that in a collection of only 23 people, the probability that two persons have the same birthday is greater than one half. In general in a collection of n people the probability that at least two persons have the same birthday is

$$1 - (\frac{1}{365^n} \times \prod_{i=0}^{n-1}(365 - i))$$

where we have assumed that peoples birthdays are independent of each other and distributed uniformly over the year. For $n = 23$ this probability is about 0.51. The following more general result holds [82].

Theorem 1.1.1 *Let H be a function with image size m. Assume that on any input, H outputs one of the m values at random. If H is evaluated $k > (2cm)^{1/2}$ times where c is a constant, then the probability that two of the k outputs are equal, i.e., a collision occurs, is at least $1 - e^{-c}$, $e = 2.718\ldots$*

Corollary 1.1.1 *With $k \simeq m^{1/2}$ the probability of at least one collision is approximately one half.*

The obvious application of the birthday paradox in cryptography is in attacks on hash functions. Consider a hash function H with image size 2^m The standard collision attack goes as follows. Collect two sets of each $2^{m/2}$ hash values Then the probability that at least one element in one set equals one element in the other set, i.e., at least one collision is found, is

$$1 - (1 - 2^{-m/2})^{2m/2} \simeq 1 - e^{-1} \simeq 0.63.$$

It is well-known that given a function f on a finite domain and a randomly chosen starting point x, the sequence $f^0(x), f^1(x), \ldots, f^n(x), \ldots$, is ultimately periodic. That is, for some l and c, it holds that $f^{c+l}(x) = f^l(x)$ and that $f^{i+c}(x) = f^i(x)$ for all $i \geq l$ [106]. $f^0(x), \ldots, f^{l-1}(x)$ and $f^l(x), \ldots, f^{l+c-1}(x)$ are called the *leader* and *the cycle* of f on x respectively and similarly the integers l and c are called the *leader length* and the *cycle length* of f on x respectively.

In [30] it is shown that for a random mapping f, $l + c \simeq \sqrt{\pi n/2}$, where n is the size of the domain of f. It follows that if l and c are taken to be the minimum integers, s.t. l is the leader and c is the cycle of f on some x, we will obtain a collision for f, i.e., $f(f^{l-1}(x)) = f(f^{l+c-1}(x))$ and $f^{l-1}(x) \neq f^{l+c-1}(x)$. However, a naive approach would still require the storage of $\sqrt{n}$ points.

In [98, 99] Quisquater and Delescaille improved this method by introducing the method of *distinguished points*, where only points with a certain easy-to-calculate attribute are stored. As an example, for a function f with domain $GF(2)^{64}$ only points, where the leading 16 bits are zero are stored. When a cycle is detected one can go back and find the place where the leader ends and the cycle begins and find a collision for f. In this way only negligible storage is required for a collision.

Since good hash functions should "act like a random function", we will assume that a collision attack on a hash function with image size 2^n can be mounted in about $2^{n/2}$ steps without any memory requirements using the method of distinguished points.

Chapter 2

Block Ciphers - Introduction

The history of cryptography is long and goes back at least 4,000 years to the Egyptians, who used hieroglyphic codes for inscription on tombs [22]. Since then many cryptosystems, also called ciphers, have been developed and used. Many of these old ciphers are much too weak to be used in applications today, because of the tremendous progress in computer technology. There are essentially two types of encryption schemes, one-key and two-key ciphers. In one-key ciphers the encryption of a plaintext and the decryption of the corresponding ciphertext is performed using the same key. Until 1976 when Diffie and Hellman introduced *public-key* or two-key cryptography [26] all ciphers were one-key systems. Therefore one-key ciphers are also called conventional cryptosystems. Conventional cryptosystems are widely used throughout the world today, and new systems are published from time to time. There are two kinds of one-key ciphers, stream ciphers and block ciphers. In stream ciphers a long sequence of bits is generated from a short string of key bits, and is then added bitwise modulo 2 to the plaintext to produce the ciphertext. In block ciphers the plaintext is divided into blocks of a fixed length, which are then encrypted into blocks of ciphertexts using the same key. Block ciphers can be divided into three groups: Substitution ciphers, transposition ciphers and product ciphers. In the following a few examples of the different types of block ciphers are given.

Notation: Let $\mathcal{A}_\mathcal{M}$ and $\mathcal{A}_\mathcal{C}$ be the alphabets for plaintexts and ciphertexts, respectively. Let $M = m_0, m_1, \ldots, m_{n-1}$ be an n-character plaintext, s.t. for every i, $m_i \in \mathcal{A}_\mathcal{M}$ and let $C = c_0, c_1, \ldots, c_{n-1}$ be a ciphertext, s.t. for every

i, $c_i \in \mathcal{A}_\mathcal{C}$. We assume that an alphabet $\mathcal{A}_\mathcal{X}$ is isomorphic with $\mathbb{N}_{\mathcal{A}_\mathcal{X}}$

2.1 Substitution Ciphers

As indicated in the name every plaintext character is substituted by some ciphertext character. There are four kinds of substitution ciphers.

- Simple substitution
- Polyalphabetic substitution
- Homophonic substitution
- Polygram substitution

We restrict ourselves to consider substitution ciphers of the first two kinds.

2.2 Simple Substitution

In a cipher with a simple substitution each plaintext character is trans-formed into a ciphertext character via the same function **f**. More formally, $\forall i \ : \ 0 \leq i < n$

$$\begin{aligned} \mathbf{f} &: \mathcal{A}_\mathcal{M} \rightarrow \mathcal{A}_\mathcal{M} \\ c_i &= \mathbf{f}(m_i) \end{aligned}$$

As an example the following

2.2.1 Caesar substitution

It is believed that Julius Caesar encrypted messages by shifting every letter in the plaintext 3 positions to the right in the alphabet. This cipher is based on **shifted alphabets**, i.e., $\mathcal{A}_\mathcal{M} = \mathcal{A}_\mathcal{C}$ and is in general defined as follows

$$\mathbf{f}(m_i) = m_i + k \pmod{|\mathcal{A}_\mathcal{M}|}$$

For the Caesar cipher the secret key k is the number 3. In general, the cipher is easily broken in at most $|\mathcal{A}_\mathcal{M}|$ trials. Shift the ciphertexts one position until the plaintext arises.

2.3 Polyalphabetic Substitution

In a polyalphabetic substitution the plaintext characters are transformed into ciphertext characters using a j-character key $K = k_0, \ldots, k_{j-1}$, which defines j distinct functions $\mathbf{f}_{k_0}, \ldots, \mathbf{f}_{k_{j-1}}$. More formally $\forall i \ : \ 0 < i \leq n$

$$\begin{array}{rcll} \mathbf{f}_{k_l} & : & \mathcal{A}_\mathcal{M} \rightarrow \mathcal{A}_\mathcal{C} & \forall l \ : \ 0 \leq l < j \\ c_i & = & \mathbf{f}_{k_{i \bmod j}}(m_i) & \end{array}$$

As an example the following

2.3.1 The Vigenére cipher

The was first published in 1586 [23]. Let us assume again that $\mathcal{A}_\mathcal{M} = \mathcal{A}_\mathcal{C}$. Then the Vigenére cipher is defined as follows

$$c_i = \mathbf{f}_{k_i \bmod j}(m_i) = m_i + k_i \bmod j \pmod{|\mathcal{A}_\mathcal{M}|}$$

2.4 Transposition Systems

Dansposition systems are essentially permutations of the plaintext characters. Therefore $\mathcal{A}_\mathcal{M} = \mathcal{A}_\mathcal{C}$. A is defined as follows $\forall i \ : \ 0 \leq i < n$

$$\begin{array}{rcl} \mathbf{f} & : & \mathcal{A}_\mathcal{M} \rightarrow \mathcal{A}_\mathcal{M} \\ \eta & : & \{0, \ldots, (n-1)\} \rightarrow \{0, \ldots, (n-1)\}, \text{ a permutation} \\ c_i & = & \mathbf{f}(m_i) = m_{\eta(i)} \end{array}$$

Many transposition ciphers permute characters with a fixed period j. In that case

$$\begin{array}{rcl} \mathbf{f} & : & \mathcal{A}_\mathcal{M} \rightarrow \mathcal{A}_\mathcal{M} \\ \eta & : & \{0, \ldots, (j-1)\} \rightarrow \{0, \ldots, (j-1)\}, \text{ a permutation} \\ c_i & = & \mathbf{f}(m_i) = m_{(i \text{ div } j)+\eta(i \bmod j)} \end{array}$$

A convenient way to express the permutation $\eta(i)$ in easily memorable form is by a key word. The alphabetic order of the key characters then defines the permutation. For example the key K=LARS would represent the permutation $\eta(i) = \{1, 0, 2, 3\}$. Consider the following transposition cipher

2.4.1 Row transposition cipher

Let the key be $K = k_1, \ldots, k_d$. The plaintext is divided into blocks of d characters, and each block is permuted according to the alphabetic order of the characters in the key. Let us consider an example:
Example: Let $d = 4$, the key K=IVAN and the plaintext

$$M = \text{NOTASTRONGCIPHER}$$

I	V	A	N
1	3	0	2
O	A	N	T
T	O	S	R
G	I	N	C
H	R	P	E

The ciphertext is

$$C = \text{OANTTOSRGINCHRPE}$$

2.5 Product Systems

An obvious attempt to make stronger ciphers than the ones we've seen so far, is to combine substitution and transposition ciphers. These ciphers are called **product ciphers**. Many product ciphers have been developed, including Rotor machines [22]. Most of the block ciphers in use today are product ciphers. A product cipher is called an *iterated cipher* if the ciphertext is computed by iteratively applying a round function several times to the plaintext. In each round a round key is combined with the text input. More formally,

Definition 2.5.1 *In an* r**-round iterated block cipher** *the ciphertext is computed by iteratively applying a round function* g *to the plaintext, s.t.*

$$C_i = g(C_{i-1}, K_i),\ i = 1, \ldots, r \tag{2.1}$$

where C_0 *is the plaintext,* K_i *a round key and* C_r *is the ciphertext. Decryption is done by reversing (2.1) therefore, for a fixed key* K_i, g *must be invertible.*

In this thesis we consider mainly iterated block ciphers and assume that the plaintexts and ciphertexts are bit strings of equal length. The Data Encryption Standard (DES) [90] is by far the most widely used iterated block cipher today. Around the world, governments, banks, and standards organisations have made the DES the basis of secure and authentic communication [108]. The DES can be seen as a special implementation of a *Feistel* cipher, named after Horst Feistel [28].

Definition 2.5.2 *A* **Feistel cipher** , *with block size* $2n$ *and with* r *rounds is defined as follows. The round function is defined*

$$\begin{aligned} g \ : \ GF(2)^n \times GF(2)^n \times GF(2)^m &\rightarrow GF(2)^n \times GF(2)^n \\ g(X,Y,Z) &= (Y, F(Y,Z)+X) \end{aligned}$$

where F *can be any function taking two arguments of* n *bits and* m *bits re-spectively and producing* n *bits.* '$+$' *is a commutative group operation on the set of* n*-bit blocks. We will assume that* '$+$' *is the bitwise exclusive-or operation, if not explicitly stated otherwise.*

Given a plaintext $P = (P^L, P^R)$ *and* r *round keys* $K_1, K_2, \ldots, K_r$ *the ciphertext* $C = (C^L, C^R)$ *is computed in* r *rounds. Set* $C_0^L = P^L$ *and* $C_0^R = P^R$ *and compute for* $i = 1, 2, \ldots, r$

$$(C_i^L, C_i^R) = (C_{i-1}^R, F(C_{i-1}^R, K_i) + C_{i-1}^L)$$

Set $C_i = (C_i^L, C_i^R)$ *and* $C^L = C_r^R$ *and* $C^R = C_r^L$*. The round keys* $(K_1, K_2, \ldots, K_r)$*, where* $K_i \in GF(2)^m$*, are computed by a key schedule algorithm on input a master key* K*.*

A special class of Feistel ciphers is the so-called DES-like iterated ciphers.

Definition 2.5.3 *A* **DES-like iterated cipher** *as a Feistel cipher, where the* F *function is defined*

$$\begin{aligned} F(X, K_i) &= f(E(X) + K_i) \\ f &: GF(2)^m \rightarrow GF(2)^n, \ m \geq n \\ E &: GF(2)^n \rightarrow GF(2)^m, \text{ an affine expansion mapping} \end{aligned}$$

Because of the success of the DES, many of the block ciphers proposed in the last decade are Feistel ciphers. Recently, this tradition was broken by

X. Lai and J.L. Massey with their *Improved Proposed Encryption Standard* [58], later named IDEA, which does not have a Feistel structure.

In Appendix A we give a self explanatory pictorial illustration of the history of block ciphers. As can be seen, encrypted pictures are an excellent tool to illustrate old weak ciphers.

Chapter 3

Applications of Block Ciphers

In this chapter we give the applications of block ciphers. In Section 3.1 we give the modes of operations, which were published for the DES [91], when used for encryption. In section 3.2 cryptographic hash functions based on block ciphers are considered. In section 3.3 we show how a block cipher can be used to construct digital signature schemes, both private systems and public systems. The latter is illustrated by describing a proposal by Merkle [72, 73]. We show that under suitable assumptions Merkle's scheme is a secure digital signature scheme.

3.1 Modes of Operations

The most obvious and widespread use of a block cipher is for encryption. In 1980 a list of four modes of operation for the DES was published [91]. These four modes can be used with any block cipher and seem to cover most applications of block ciphers used for encryption [22]. In the following let $E_K(\cdot)$ be the permutation induced by using the block cipher E of block length n with the key K and let $P_1, P_2, \ldots, P_i, \ldots$ be the blocks of plaintexts to be encrypted. The four modes are

- **Electronic Code Book (ECB)** The native mode, where one block at a time is encrypted independently of the encryptions of other blocks. Encryption

$$C_i = E_K(P_i)$$

Decryption

$$P_i = E_K(C_i)$$

- **Cipher Block Chaining (CBC)** The chaining mode, where the encryption of a block depends on the encryptions of previous blocks. Encryption

$$C_i = E_K(P_i \oplus C_{i-1})$$

Decryption

$$P_i = D_K(C_i) \oplus C_{i-1}$$

where C_0 is a chosen initial value.

- **Cipher Feedback (CFB)** The first stream mode, where one m-bit character at a time is encrypted. Encryption

$$\begin{aligned} C_i &= P_i \oplus \mathrm{MSB}_m(E_K(X_i)) \\ X_{i+1} &= \mathrm{LSB}_{n-m}(X_i) \parallel C_i \end{aligned}$$

Decryption

$$\begin{aligned} P_i &= C_i \oplus \mathrm{MSB}_m(E_K(X_i)) \\ X_{i+1} &= \mathrm{LSB}_{n-m}(X_i) \parallel C_i \end{aligned}$$

where X_1 is a chosen initial value, $\parallel$ denotes concatenation of blocks, MSB_s and LSB_s denote the s most and least significant bits respectively or equivalently the leftmost and rightmost bits respectively. Here m can be any number between 1 and the block length of the cipher. If the plaintext consists of characters $m = 7$ or $m = 8$ is usually the well-chosen parameter.

- **Output Feedback (OFB)** The second stream mode, where the stream bits are not dependent on the previous plaintexts, i.e., only the stream bits are fed back, not the ciphertext as in CFB mode.

$$\begin{aligned} C_i &= P_i \oplus \mathrm{MSB}_m(E_K(X_i)) \\ X_{i+1} &= \mathrm{LSB}_{n-m}(X_i) \parallel \mathrm{MBS}_m(E_K(X_i)) \end{aligned}$$

Decryption

$$\begin{aligned} P_i &= C_i \oplus \mathrm{MSB}_m(E_K(X_i)) \\ X_{i+1} &= \mathrm{LSB}_{n-m}(X_i) \parallel \mathrm{MSB}_m(E_K(X_i)) \end{aligned}$$

where X_1 is a chosen initial value.

In fact, both the CFB and OFB modes have two parameters, the size of the plaintext block and the size of the feedback value. In the above definition we have chosen them equal and will do so also in the following.

The ECB is the native mode, well-suited for encryption of keys of fixed length. It is not suited for the encryption of larger plaintexts, since equal blocks are encrypted into equal blocks. To avoid this, the CBC mode is recommended. Not only does a current ciphertext block depend on the current plaintext but also on all previous ciphertext blocks. In some applications there is a need for encryptions of characters, instead of whole blocks, e.g. 8 bytes for the CBC mode of DES. For that purpose the CFB and OFB modes are suitable. The OFB should be used only with full feedback, i.e., with $m = n$, the block length, e.g. 64 for the DES. It comes from the fact, that for $m < n$ the feedback function is not one-to-one, and therefore has a relatively short cycle [22]. Furthermore the initial value X_1 in the OFB mode should be chosen uniformly at random. In the case where X_1 is the concatenation of n/m equal m-bit blocks, say $(a \parallel a \parallel \ldots \parallel a)$, for about 2^{k-m} keys $\mathrm{MSB}_m(E_K(X_1)) = a$. Therefore $X_2 = X_1$ and in general $X_i = X_1$. This is not dangerous for the CFB mode, where the X_i's are also dependent on the plaintext.

An important issue in the applications of the four modes is how an error in the transmission of ciphertexts is propagated. In the ECB mode an error in a ciphertext block of course affects only one plaintext block. An error in a ciphertext in the CBC mode affects two plaintexts blocks. As an example, assume that ciphertext C_3 has an error and that all other ciphertext blocks are error-free, then $P_4 = D_K(C_4) \oplus C_3$ inherits the error from C_3 and $P_3 = D_K(C_3) \oplus C_2$ will be completely garbled. Here we assume that even a small change in the plaintext to the block cipher will produce a very different ciphertext. All other plaintexts will be decrypted correctly. In the CFB mode an error in a ciphertext block C_i will be inherited by the corresponding plaintext block P_i, and moreover since X_{i+1} contains the garbled C_i the subsequent plaintexts blocks will be garbled until the X value is free of C_i,

i.e., when C_i has been shifted out. In other words in CFB mode with m-bit ciphertexts, at most $n/m + 1$ plaintext blocks will be garbled. In the OFB mode, since the feedback is independent of the plain- and ciphertexts, a transmission error in a ciphertext block garbles only the corresponding plaintext block and is not propagated to other plaintext blocks. In Section 4.4.1 we give an analysis of three other suggested modes of operation.

3.2 Cryptographic Hash Fhctions

A *hash function* takes as argument a bit string of arbitrary length and produces a hash-code of fixed length. Cryptographic on hash functions hash functions are used to provide data integrity and to produce short digital signatures [37, 55, 93]. When used for data integrity, the data blocks are hashed into a short length hash code, which is then stored securely. Any modifications in the data would be detected by applying the hash function to the modified data blocks. If the hash function is strong with a high probability the obtained hash code will be different from the secure stored hash code. Digital signature schemes are often based on expensive mathematical routines. Instead of signing a large document, it is first hashed into a short length hash code, which is then signed. If the hash function is strong it will be infeasible to find (meaningful) documents yielding equal hash codes.

In [93], Bart Preneel makes a distinction depending on whether a cryptographic hash function is used with a secret key, in which case the hash function is called a **MAC** (Message Authentication Code), or if the hash function is used without a secret key, in which case the hash function is called a **MDC** (Manipulation Detection Code). The non-keyed hash functions, the MDC's, are further categorised into one-way hash functions and collision-resistant hash functions.

Definition 3.2.1 *A* **collision resistant** *hash finction H satisfies the following conditions*

1. *The description of H must be publicly known and should not require any secret information for its operation.*
2. *The argument can be of arbitrary length and the hash code $H(\cdot)$ has a fixed length.*

3. *Given H and an argument X, it should be 'easy' to compute $H(X)$.*

4. *One-way-ness: Given a Y in the image of H, it is 'hard' to end a message X, s.t. $H(X) = Y$ and given X and $H(X)$ it is 'hard' to find a message $X' \neq X$, s.t. $H(X') = H(X)$.*

5. *Collision resitance: It is 'difficult' to find a pair X, X', s. t. $X \neq X'$ and $H(X) = H(X')$.*

The difference between a collision-resistant hash function and a one-way hash function is the lack of requirement (5.) for the latter. MAC's are used for message authentication and are standardised in the banking world, see for example [108]. The different applications for MAC's and MDC's are treated in a comprehensive manner in [93] and will not be treated any further here. From now on we will consider only collision resistant MDC's, if not stated otherwise.

Many of the proposed hash functions are so-called *iterated* hash functions, where one iterates a hash round function.

Definition 3.2.2 *In an iterated m-bit hash function, H, the hash code $H(M) = H_n$ of the message $M = M_1, \ldots, M_n$ is computed iteratively by the equation*

$$H_i = h(H_{i-1}, M_i)$$

where $h(\cdot, \cdot)$ is a function taking two arguments of m bits and l bits respectively and producing an m bit value and where H_0 is a chosen initial value.

For message data whose total length in bits is not a multiple of l, one can apply deterministic "padding" [38, 74] to the message to be hashed by h to increase the total length to a multiple of l. In the following set the initial value $H_0 = IV$. We distinguish between the following attacks on a hash function H, where IV' denotes an initial value, not necessarily equal to IV. We denote by $H(IV, X)$ explicitly the hash codes dependency on the initial value IV, see also [55].

Preimage attack. The attacker is given IV and $H(X)$ and finds X', s.t. $H(IV, X) = H(IV, X')$.

Second preimage attack. The attacker is given IV, X and $H(IV, X)$ and finds X', s.t. $X \neq X'$ and $H(IV, X) = H(IV, X')$.

Free-start preimage attack. The attacker is given IV and $H(X)$ and finds IV' and X', s.t. $IV \neq IV'$ and $H(IV, X) = H(IV', X')$.

Free-start second preimage attack. The attacker is given IV, X and $H(X)$ and finds IV' and X', s.t. $(IV, X) \neq (IV', X')$ and $H(IV, X) = H(IV', X')$.

Collision attack. The attacker is given IV and finds X and X', s.t. $X \neq X'$ and $H(IV, X) = H(IV, X')$.

Semi-free-start collision attack. The attacker finds IV', X and X', s.t. $X \neq X'$ and $H(IV', X) = H(IV', X')$.

Free-start collision attack. The attacker finds IV, IV', X and X', s.t. $(IV, X) \neq (IV', X')$ and $H(IV, X) = H(IV', X')$.

Preimage attacks are sometimes also called target attacks [55], where the intuition is that $H(X)$ is a given "target", that the attacker tries to "hit". It is clear that a free-start collision attack can never be harder than a free-start preimage attack and a collision attack is never harder than a preimage attack. For an m-bit hash function, brute force preimage attacks, in which one randomly chooses an M' until one hits a given $H_n = H(M)$, require about 2^m computations of hash values. It follows from the birthday paradox, section 1.1.1, that brute force collision attacks require about $2^{m/2}$ computations of hash values. In particular, for hash round functions with $l \geq m$ so that all 2^m hash values can be reached with one-block messages: brute-force preimage attacks require about 2^m computations of the round function h while brute force collision attacks require about $2^{m/2}$ computations of the round function h. These complexities also gives us upper bounds on the terms 'hard' and 'difficult' from Definition 3.2.1 for iterated hash functions, i.e., 'hard' is never harder than the computation of about 2^m hash values and 'difficult' is no more difficult than the computation of about $2^{m/2}$ hash values. There have been suggested many methods of how to construct 'secure' hash functions. A few of them have a security provably equivalent to a hard problem like factoring a large composite number or computing the logarithm in a finite field. Often hash functions are based on block ciphers and this is the

approach that we will take in this thesis. One obvious advantage of using block ciphers as building blocks in a hash function is to reduce the costs. If one already has a block cipher used for encryption, all one needs is a mode of operation of how to transform the cipher into a hash function. History shows that is not at all an easy task. To avoid some trivial collision attacks, see e.g. [55], where the messages found are not of the same length, one can do the following proposed independently by Damgård [18] and Merkle [74]

Definition 3.2.3 (The MD-strengthening) *Let $M = M_1, \ldots, M_n$ be the message to be hashed. Then one appends an extra last block, M_{n+1} to the message containing the length of the original message.*

With the MD-strengthening a secure hash round function implies a secure hash function [18, 74, 55] with roughly the same security level [18, 74, 55]. Since hash functions are used to produce short digital signatures they should be reasonably fast. When discussing hash functions based on block ciphers a natural measurement is

Definition 3.2.4 *The* **hash rate** *of an iterated hashfunction based on a block cipher is the number of message blocks processed by one encryption of the block cipher.*

$$\text{Hash rate} = \frac{\#\text{ message blocks}}{\#\text{ encryptions}}$$

We note, that in [93] Preneel defines the hash rate the opposite way, i.e., the hash rate is number of encryptions needed to process one message block. In our definition (also the one of [37]) the intuition is, the higher the hash rate, the faster the hash function.

If one has trust in a block cipher confidence can be obtained about the security of a hash function. The following hash function has a security level, which can be expressed in terms of the security of the block cipher, see also [74].

Theorem 3.2.1 *Let $E_K(\cdot)$ be an m-bit block cipher with a k bit key with $k > m$ and let the H be an iterated hash function with hash round function*

$$H_i = h(H_{i-1}, M_i) = E_{H_{i-1} \| M_i}(P_c)$$

where P_c is a constant m-bit block and the message blocks are of length $(k-m)$ bits. Assume that MD-strengthening is used. Then a free-start collision attack on H is at least as hard as finding a key collision of E in a known plaintext attack. And a free-start preimage attack on H is at least as hard as finding a key of E in a known plaintext attack.

Proof: Consider first the free-start collision attack. Assume that an attacker finds IV, IV' and messages M, M', s.t. $(IV, M) \neq (IV', M')$ and $H(IV, M) = H(IV', M')$, that is,

$$H(M) = E_{H_{n-1}\|M_n}(P_c) = E_{H'_{n-1}\|M'_n}(P_c) = H(M')$$

If M and M' are not of the same length, then $M_n \neq M'_{n'}$, and the attacker has found a key collision for E, i.e., $K \neq K'$ s.t. $E_K(P_c) = E_{K'}(P_c)$. Assume now that M and M' are of the same length, then it follows that either $H_{n-1} \neq H'_{n-1}$ in which case the attacker has found a key collision or $H_{n-1} = H'_{n-1}$. It follows by 'reverse' induction that for some i

$$H_i = E_{H_{i-1}\|M_i}(P_c) = E_{H'_{i-1}\|M'_i}(P_c) = H'_i \wedge (H_{i-1}, M_i) \neq /H'_{i-1}, M'_i)$$

Thus, a free-start collision for H implies a key collision for E.
Consider now the free-start preimage attack. The attacker is given IV and $H(M)$. By a similar argument as above, it follows that in case of a free-start preimage attack, the attacker finds a key K, s.t. $E_K(P_c) = C = H(M)$, i.e. the attacker has found the secret key in a known plaintext attack. If MD-strengthening is not used the hash function is trivially broken using a free-start attack. □

The hash functions of Theorem 3.2.1 require that the key size exceeds the block size, which is not the case for the DES, where the block size is 64 and the key size is 56. Since the DES is so widely in use as an encryption function many attempts have been made to build a hash mode suitable for DES.

In [74] Merkle proposed a hash function based on a block cipher (e.g. DES) based on the so-called "meta-method". The scheme is related to the idea of Theorem 3.2.1, but more than one encryption is needed in each round of the hash function to compensate for the small key and plaintexts. It is shown that the scheme is as secure as the underlying block cipher under the assumption that the block cipher is a random function. Since a permutation does not "act as a random function", Merkle uses a feedforward-(of the

plaintext) mode, that is believed to be one-way in some sense. Assume that an m-bit block cipher with a k-bit key is used, where $k < m - 1$. The hash code is of length $2k$ bits and the message blocks are of length $m + k - 1$. The drawback of this scheme is that the hash rate is low, only $\frac{m-k-1}{2m}$. In case of the DES this means that only 3.5 bits are hashed per encryption and the hash rate is 0.05. Merkle also suggests two improved schemes with the same kind of security connection to the block cipher. However, even the fastest one has a hash rate of only 0.27. To our knowledge this is the closest someone has come to "provable security" of a hash function based on the DES.

Many of the proposed hash round functions based on a block cipher are used in the feedforward-(of the plaintext) mode. A well-known example of such a hash function is the Davies-Meyer scheme (DM)[1] with hash rate 1, where the hash round function is given by

$$H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1} \tag{3.1}$$

For hash functions based on block ciphers we have the following definition.

Definition 3.2.5 *The* **complexity** *of an attack on a hash function based on a block cipher is the nunaber of encryptions (or decryptions) of the block cipher, that the attacker has to do.*

The DM-scheme with MD-strengthening is generally considered to be secure, if the underlying block cipher with block size m has no weaknesses [55], in the sense that the complexity of a free-start collision attack is about $2^{m/2}$ and the complexity of a free-start preimage attack is about 2^m. The DM-scheme is called a single block length hash function We have following definition.

Definition 3.2.6 *A* **single block length** *iterated hash function, H, based on an m-bit block cipher E with a k-bit key, is an iterated hash function, where the hash round function is defined*

$$H_i = h(H_{i-1}, M_i) = E_{g_1(H_{i-1}, M_i)}(g_2(H_{i-1}, M_i)) \oplus (g_3(H_{i-1}, M_i))$$

where the g_i's are linear finctions of H_{i-1} and M_i and where the M_i's are of length k or m depending on the g_i's.

[1]The scheme has in fact never been proposed by D. Davies, as explained in a letter from Davies to Bart Preneel [92]. Since the hash function is widely known as the Davies-Meyer scheme, we will refer to it as such, often only by the shorter name, DM.

As can be seen it is possible to obtain 64 single block length hash functions for a block cipher. In [95] it was shown that only 12 of these are secure one-way hash functions. This subject is treated further in Chapter 8.

Since most block ciphers have a block length of only 64 bits, the hash code of a single block length hash function is only 64 bits and the complexity of a collision attack is small, see Section 1.1.1. Therefore much research has been done to construct hash functions with double block length. The message M is now split into subblocks as follows $M = M_1^1, M_1^2, \ldots, M_n^1, M_n^2$. First we give the parallel version of double block length hash functions.

Definition 3.2.7 *A* **parallel double block length** *iterated hash function, H, based on a block cipher E, is an iterated hash function, where two hash round finctions h_1, h_2 are defined*

$$\begin{aligned} H_i^1 = h^1(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2) &= E_{f_1}(f_2) \oplus (f_3) \\ H_i^2 = h^2(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2) &= E_{g_1}(g_2) \oplus (g_3) \end{aligned}$$

where both the f_i's and g_i's are linear functions of $H_{i-1}^1, H_{i-1}^2, M_i^1$ and M_i^2. H_0^1 and H_0^2 are the initial values and the haah code is (H_n^1, H_n^2).

In a serial version of a double block length hash function the hash value of one hash round function, say H_i^1, can be used in the computation of the hash value of the other hash round function.

Definition 3.2.8 *A* **serial double block length** *iterated hash function, H, based on a block cipher E, is an iterated hash function, where two hash round functions h^1, h^2 is defined*

$$\begin{aligned} H_i^1 &= h^1(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2) &&= E_{f_1}(f_2) \oplus (f_3) \\ H_i^2 &= h^2(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2, H_i^1) &&= E_{g_1}(g_2) \oplus (g_3) \end{aligned}$$

where the f_i's are linear functions of H_{i-1}^1, H_{i-1}^2, M_i^1 and M_i^2, and where the g_i's are linear functions of $H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2$ and H_i^1. H_0^1 and H_0^2 are the initial values and the haah code is (H_n^1, H_n^2).

It is possible to obtain $16^3 \times 32^3 = 2^{27}$ serial double block length "hash functions" for a block cipher. They are not all "real" hash functions e.g. the

hash functions were neither the f_i's nor the g_i's contain message blocks, and many of them are hopelessly weak. In Chapter 8 we will show attacks on a large class of these hash functions. The difference contain message between the parallel and serial hash functions is important in hardware, where a parallel hash function in general will be faster than a serial hash function. In (conventional) software everything is "serial", and there is no difference in efficiency of the two hash function classes.

Since the DM-scheme is generally considered secure with the only disadvantage being a small block length, many attempts have been made double block length based on the concatenation of two variants of the DM-scheme. One such scheme, the MDC-2 by Meyer and Schilling [10, 77] is submitted for publication as an IS0 standard [38].

3.3 Digital Signatures

A digital signature is the electronic version of a hand-written signature. The main difference is that the digital signature is an encryption of a cleartext and must be used only once. Therefore a digital signature must include the names of the participants and a time stamp or serial number etc. A digital signature scheme provides **sender authenticity** and **data integrity**. Digital signature systems are divided into two parts, the public and private systems. A public digital signature system identifies the sender to anyone from publicly available information, whereas a private digital signature system identifies the sender only to someone sharing a secret with the sender.

3.3.1 Private digital signature systems

A private digital signature system has the following properties. Imagine that party **A** is signing message M to party **B**. Then

1. **B** must be able to validate **A**'s signature on M.
2. It should be infeasible for anyone, including **B**, to forge **A**'s signature.
3. If **A** later denies to have signed M, it should be possible for a third party to resolve a dispute arising between **A** and **B**.

A conventional cryptosystem itself in its basic mode cannot be used to produce digital signatures. The key for encryption and decryption is the same in a conventional cryptosystem and is known by both parties, therefore **B** may be able to forge **A**'s signatures and a third party is not able to solve a dispute about whether **A** signed a message M or **B** made the signature himself. Using a trusted third party it is possible to obtain a digital signature satisfying all desired properties. The protocol in Figure 3.1 is useful, where **TP** denotes the trusted third party, X is a key known only to **TP** and K_{A} and $K_{\mathbf{B}}$ are **A**'s and **B**'s keys distributed by **TP**. I_A is a string identifying **A**. Let $E_{K_C}(\cdot)$ denote encryption using the key K_C. The requirements for a digital signature scheme are all met, but the scheme involves an active third party for every signature produced and is therefore very inefficient.

1. **A** sends $S_1 = E_{K_{\mathbf{A}}}(M)$ to **TP**
2. **TP** decrypts S_1, finds M and sends $S_2 = E_{K_X}(M \mid I_A)$ to **A**
3. **A** sends S_2 to **B**
4. **B** sends S_2 to **TP**
5. **TP** decrypts S_2, checks the identity of **A** and sends $S_3 = E_{K_{\mathbf{B}}}(M \mid I_A)$ to **B**
6. **B** finds $D_{K_{\mathbf{B}}}(S_3) = (M \mid I_A)$ and checks the identity of **A**.

Figure 3.1: A digital signature scheme based on a conventional cryptosystem.

3.3.2 Public digital signature systems

A public digital signature system has the following properties:

1. Anyone must be able to validate one party's, say, **A**'s signature on a message M without an active third party.
2. It should be infeasible for anyone to forge **A**'s signature.
3. If **A** later denies to have signed M, it should be possible for a third party to resolve a dispute arising between **A** and **B**.

As the name indicates public key cryptosystems are well-suited for public digital signature systems, see for example [78]. These signature systems can also be based on conventional cryptosystems. Rompel [101] has shown that the existence of one-way functions is a necessary and sufficient conditions for making secure signatures. Consider a block cipher $E_K(\cdot)$. If the block cipher is secure, it is necessary that given pairs (P_i, C_i), s.t. $C_i = E_K(P_i)$ it is difficult to find K. If this is the case, $E_K(\cdot)$ can be used to construct a one-way function, $F(K) = E_K(P)$ for a fixed plaintext P. Let F be a publicly known one-way function. To sign a one-bit message, a sender **A** selects two secret values, (x_0, x_1), computes $y_0 = F(x_0)$ and $y_1 = F(x_1)$ and authenticates (y_0, y_1) by placing them in a public file. To sign the one-bit message $b \in \{0, 1\}$ to **B**, **A** sends x_b, and y_b to **B**. **B** checks whether $F(x_b) = y_b$. This is the approach taken in the Lamport-Diffie signature scheme [26]. To sign an m-bit message, $2m$ F-values must be authenticated. To prevent forgery these values can be used only once, and the signatures produced are therefore called *one-time signatures*, Merkle has shown how to reduce the $2m$ values to about $m + log_2(m)$ values. The signer computes $m + log_2(m)$ values x_i and places the corresponding y_i values in the public file. He signs a message of m bits in the following way. First he counts the number of zeroes in the binary string m and appends a bit string representing this number to m yielding the string m'. Then he reveals x_i if the i'th bit of m' is a '1'-bit otherwise he does nothing. Winternitz-Merkle [72] proposed a way of signing an n-bit message by computing $2n$ values but revealing only 2 values. To sign an n-bit message, a sender **A** selects two secret values, (x_0, x_1), computes $y_0 = F^n(x_0)$ and $y_1 = F^n(x_1)$ and authenticates (y_0, y_1) by placing them in a public file. To sign the n-bit message $m_n \in \{0, \ldots, (n-1)\}$ to **B**, **A** sends m_n, $F^{m_n}(x_0)$ and $F^{n-m_n}(x_1)$ to **B**. **B** can verify which power of $F(x)$ **A** sent him by checking how many evaluations of F are needed to reach the y-values. **B** cannot use the values he got from **A** to forge the system, without having to invert F.

Merkle's digital signature tree

In [72, 73] Merkle proposed a digital signature scheme based on a conventional cryptosystem, e.g. DES, producing an infinite number of one-time signatures using a tree structure. The basic idea in Merkle's scheme is to authenticate the root of a tree by placing it in a public file. The root signs one message

and authenticates its sub-nodes, the sons of the node, in the tree. In general a node in the tree signs one message and authenticates its sub-nodes. The tree can be any K-ary tree, for simplicity let us consider a binary tree. We number the nodes in the following standard way. The root has number 1 and the subnodes of node j have the numbers $2j$ and $2j+1$ respectively. Assume that F is a publicly known one-way function and H a publicly known one-way hash function, as defined in Section 3.2, hashing a string of arbitrary length into a string of length n bits. The i'th node in the tree has three arrays,

$$x[i, \text{left}, *],\ x[i, \text{right}, *],\ x[i, \text{message}, *]$$

where $*$ denotes n values. For every node the signer computes three other arrays

$$y[i, \text{left}, *],\ y[i, \text{right}, *],\ y[i, \text{message}, *]$$

where $y[i, X, j] = F(x[i, X, j])$ for $X = \{\text{left, right, message}\}$ and $j = 1, \ldots, n$. Let the hash value of the concatenation of all n values in $y[i, X, *]$ be denoted by $H(y[i, X, *])$ and let

$$\text{Hash}(i) = H(H(y[i, \text{left}, *]) \parallel H(y[i, \text{right}, *]) \parallel H(y[i, \text{message}, *]))$$

i.e., the hash value of the concatenation of the hash values of the three y-arrays. To start the digital signature scheme the signer computes the $3 \times n$ values of the three x arrays and of the three y-arrays at node 1. The node is authenticated by applying the hash function H to all values of the three y-arrays, i.e., computing Hash(1), and putting it in a public file. Let again **A** and **B** be the parties in the scheme. **A** signs the i'th message M_i to **B** using the following protocol [72], where '/' means integer division:

1. **A** sets $j = i$, sends j and $y[j, \text{message}, *]$ to **B**.
2. **A** signs the message M_j by sending the appropriate values of the $x[j, \text{message}, *]$-array to **B**, in the same way as in the original Lamport-Diffie scheme.
3. **B** checks if the y-values and the x-values sign the message M_j using F.
4. **A** sends the hash values of the three $y[j, *, *]$ arrays, i.e., $H(y[j, \text{left}, *])$, $H(y[j, \text{right}, *])$ and $H(y[j, \text{message}, *])$ to **B**.

5. If $j = i$, **B** checks if the $y[j, \text{message}, *]$ received in step 1 yields the value $H(y[j, \text{message}, *])$, received in step 4.

   If $j \neq i$, **B** checks if the $y[j/2, \text{left}, *]$ (or $y[j/2, \text{right}, *]$) received previous round step 8 (or step 9) yields $H(y[j, \text{left}, *])$ (or $H(y[j, \text{right}, *])$ received in this round in step 4.

6. If $j = 1$, **B** uses H to check if the $H(y[1, *, *])$ values received in step 4 yield Hash(1) and the protocol terminates.

7. **A** computes Hash(j).

8. If j is even, **A** sends $y[j/2, \text{left}, *]$ to **B**. **A** signs Hash(j) by sending an appropriate subset of $x[j/2, \text{left}, *]$ to **B**. **B** computes Hash(j) and checks that it is signed correctly.

9. If j is odd, **A** sends $y[j/2, \text{right}, *]$ to **B**. **A** signs Hash(j) by sending an appropriate subset of $x[j/2, \text{right}, *]$ to **B**. **B** computes Hash(j) and checks that it is signed correctly.

10. **A** and **B** set j to $j/2$ and go to step 4.

Upon termination **B** has received $log_2(i) + 1$ one-time signatures. The first signature authenticates the message, the others authenticates the next signature and the last signature is authenticated by the entry in the public file. In Merkle's description [72] of the scheme step 5 is missing. We will show that this step is crucial for the security of the scheme. Also, since the signatures are one-time signatures only one message can be signed per signature number. We can prove the following result.

Theorem 3.3.1 *If signer* **A** *signs only one message per signature number, the above protocol implements a secure digital signature scheme in the sense that forging a signature implies either 1) an inversion of F or 2) an inversion for H, i.e., a preimage attack or a second preimage attack.*

Proof: Assume that an enemy **C** can forge a signature on a message, which **A** did not sign. Let $I \in \mathbb{N}$ be the signature numbers, $M^I = \{M_i \mid i \in I\}$ the messages and $S^I = \{S_i \mid i \in I\}$ the signatures, which **A** has signed in legitimate communications. In the following all variables in a forged signature are primed. Now assume that **C** produces a signature S'_j of the message

$M'_j \notin M^I$. It is clear that **C** cannot sign anything else than **A** has already signed using the appropriate subsets of $x[i, X, *] \in S^I$. So either

1. **C** found $x[i, X, a] \notin S^I$ for at least one a, s.t. $F(x[i, X, a]) = y[i, X, a] \in S^I$. i.e., **C** inverted F or
2. **C** found for at least one value b either
   (a) $y'[i, X, b] \neq y[i, X, b] \in S^I$, s.t. $\text{Hash}'(i) = \text{Hash}(i)$ for $i \in I$ or
   (b) $y'[i, X, b]$, s.t. $\text{Hash}'(k) = \text{Hash}(k)$ for $k \in I$ and $k < i$.

where we note that in step (2b) **C** has to hit either a Hash(k)-value produced by **A** or at least hit Hash(1) from the public file. Also note that that the attacks **C** has to perform in steps (2a) and (2b) corresponds to a second preimage attack and a preimage attack on H according to the classification of attacks in Section 3.2. □

The necessity of step 5 in Merkle's scheme

In the following we will show how an enemy **C** can forge **A**'s signatures, if step 5 is not included in Merkle's scheme. **C** gets a signature S_i from **A** on a message M_i. Now **C** can pretend to be **A** and sign $M'_i \neq M_i$ to a third party **B** by the following method

1. Choose at random $x'[i, X, *]$ for $X = \{\text{left,right,message}\}$
2. Compute $y'[i, X, j]) = F(x'[i, X, j])$ and send $y[i,\text{message},*]$ to **B**.
3. Sign M'_i by sending an appropriate subset of $x'[i,\text{message},*]$ to **B**.
4. Send $H(y[i, X, *])$ for $X = \{\text{left,right,message}\}$ from S_i obtained in communication with **A**.
5. Follow the protocol (without step 5) using the values from S_i obtained in communication with **A**.

Merkle's signature scheme can be based on any one-time signature, e.g. Winternitz's or Merkle's method and can be used with any K-ary tree structure. Winternitz method yields shorter signatures, but requires more evaluations of the one-way function F. Using a K-ary tree yields shorter signatures

Merkle's one-time signatures

Tree-structure	Size of signatures (bytes)	No. of DES operations
4-ary	6700	6300
8-ary	4600	7500

Winternitz's one-time signatures with $n = 8$

Tree-structure	Size of signatures (bytes)	No. of DES operations
4-ary	4300	10000
8-ary	3000	15000

Table 3.1: Trade-offs in Merkle's signature scheme with a maximum of 500 signatures implemented with the DES.

but requires $K + 1$ arrays at every node and more evaluations of the one-way hash function H. Using the DES as the one-way function and MDC-2 [10, 77] based on the DES, we get a trade-off between types of trees and one-time signature types measured in the number of DES operations. In a signature system with a maximum of 500 signatures the trade-off is shown in Table 3.1 where the size of signatures and no. of DES operations are worst-case considerations.

Chapter 4

Security of Secret Key Block Ciphers

In this chapter we consider the security of block ciphers. In Section 4.1 we describe the model of reality we are dealing with. In Section 4.2 we classify the possible attacks on a block cipher. In Section 4.3 we describe briefly Shannons theory of theoretical secrecy. In Section 4.4 we introduce a new classification of the kinds of success an attacker has in attacks on block ciphers. We give attacks on block ciphers used in three of the four well known modes of operation [91] and establish a new upper bound on the complexity of attacks on block ciphers used in these modes. Finally we examine two other non-standard modes of operation.

4.1 The Model of Reality

When discussing the security of cryptographic systems one needs to define a model of the reality. We will use the model of Shannon [107] which is depicted in Figure 4.1.

The sender and the receiver share a common key K, which has been transmitted over a secure channel. The sender can encrypt a plaintext P using the secret key K, send C over an insecure channel to the receiver, who can restore C into P using K. The attacker has access to the insecure channel and can intercept the ciphertexts (cryptograms) sent from the sender to the

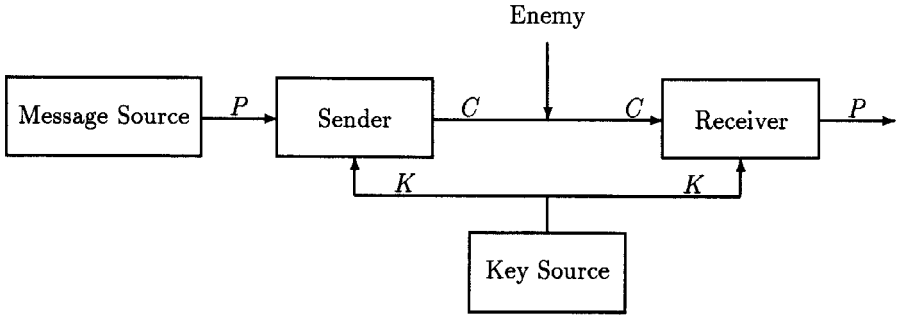


Figure 4.1: Shannon's model of a general secrecy system.

receiver. In this section we assume that the legitimate sender and receiver use a secret key cipher $E_K(\cdot)$ of block size n (bits), where the key K is of size k bits. To avoid an attacker to speculate in how the legitimate parties have constructed their common key, we will assume

Assumption 4.1.1 *All keys are equally likely and a key K is always chosen uniformly random.*

Also we will assume that all details about the cryptographic algorithm used by the sender and receiver are known to the attacker, except for the secret key. This assumption is known as

Assumption 4.1.2 (Kerckhoffs's Assumption [40]) *The enemy cryptanalyst knows* **all details of the enciphering process and deciphering process** *except for the value of the secret key.*

4.2 Classification of Attacks

Using these assumptions we classify the possible attacks an attacker can do.

- **Ciphertext only attack.**
  The attacker possesses a set of intercepted ciphertexts.
- **Known plaintext attack.**
  The attacker obtains a set of s plaintexts $P_1, P_2, \ldots, P_s$ and the cor-

responding ciphertexts $C_1, C_2, \ldots, C_s$. That is, the attacker has no control over the pairs of plain- and ciphertexts available to him.

- **Chosen plaintext attack.**
  The attacker chooses *a priori* a set of s plaintexts $P_1, P_2, \ldots, P_s$ and obtains in some way the corresponding ciphertexts $C_1, C_2, \ldots, C_s$.

- **Adaptively chosen plaintext attack.**
  The attacker chooses a set of plaintexts $P_1, P_2, \ldots, P_s$ interactively as he obtains the corresponding ciphertexts $C_1, C_2, \ldots, C_s$. That is, the attacker chooses P_1, obtains C_1, **then** chooses P_2 etc.

- **Chosen ciphertext attacks.**
  For symmetric ciphers these are similar to those of chosen plaintext attack and adaptively chosen plaintext attack, where the roles of plain- and ciphertexts are interchanged.

The chosen text attacks are of course the most powerful attacks an attacker can do. In many applications they are however also unrealistic attacks. If the plaintext space contains redundancy, it will be hard for an attacker to 'trick' a legitimate sender into encrypting non-meaningful plaintexts and similarly hard to get ciphertexts decrypted, which do not yield meaningful plaintexts. But if a system is secure against an adaptively chosen plaintext/ciphertext attack then it is also secure against all other attacks. An ideal situation for a designer would be to prove that her system is secure against an adaptively chosen plaintext attack, although an attacker may never be able to mount more than a ciphertext only attack.

4.3 Theoretical Secrecy

In his milestone paper from 1949 [107] Shannon defines perfect secrecy for secret key systems and shows that they exist. We will now give a brief description of Shannons theory and the most important results. Let $\mathbf{P}$, $\mathbf{C}$ and $\mathbf{K}$ be the random variables representing the plaintexts, ciphertexts and the keys respectively. Let $P_{\mathbf{X}}(x)$ be the probability that the random variable $\mathbf{X}$ takes on the value x.

Definition 4.3.1 (Shannon [107]) *The uncertainty (entropy) $H(\mathbf{X})$ of a random variable $\mathbf{X}$ is defined as the expectation of the negative logarithm of the corresponding probability distribution.*

Using the logarithm base 2, we get

$$H(\mathrm{X}) \stackrel{\text{def}}{=} E[-log_2 P_{\mathrm{X}}(x)] = - \sum_{x \in supp(P_{\mathbf{X}})} P_{\mathbf{X}}(x) \times log_2 P_{\mathbf{X}}(x)$$

where $supp(P_{\mathbf{X}}) \stackrel{\text{def}}{=} \{x : P_{\mathbf{X}}(x) \neq 0\}$. When using this logarithm the entropy of $\mathbf{X}$ can be seen as the number of bits needed to represent (the possible values of) $\mathbf{X}$ in an optimal binary coded form. Further we define the **conditional entropy of X given Y** as

$$\begin{aligned} H(\mathbf{X} \mid \mathbf{Y}) &\stackrel{\text{def}}{=} E[-log_2 P_{X|Y}(X \mid Y)] \\ &= - \sum_{x,y \in supp(P_{\mathbf{X},\mathbf{Y}})} P_{\mathbf{X},\mathbf{Y}}(x, y) \times log_2 P_{\mathbf{X}|\mathbf{Y}}(x \mid y) \end{aligned}$$

in other words the uncertainty about $\mathbf{X}$ given that $\mathbf{Y}$ is known. The quantity $I(X;Y) = H(\mathbf{X}) - H(\mathbf{X} \mid \mathbf{Y})$ is called **the information** that $\mathbf{Y}$ gives about $\mathbf{X}$.

Definition 4.3.2 (Shannon [107]) *A secret key cipher is* **perfect** *if and only if $H(\mathbf{P}) = H(\mathbf{P} \mid \mathbf{C})$, i.e., when the ciphertext $\mathbf{C}$ gives no information about the plaintext $\mathbf{P}$.*

This definition leads to the following obvious result.

Corollary 4.3.1 *A perfect cipher ia unconditionally secure against a ciphertext only attack.*

As noted by Shannon the Vernam cipher, also called the *one-time pad*, is a perfect secret key cipher. In the one-time pad the plaintext characters are exclusive-or'ed with independent key characters to produce the ciphertexts. However, the practical applications of perfect secret key ciphers are limited, since, as also noted by Shannon, it requires as many digits of secret key as there are digits to be enciphered [63]. A less stringent form of theoretical secrecy is possible, defined by Shannon in terms of

Definition 4.3.3 (Shannon [107]) *The* **unicity distance**, n_{ud}, *of a cipher is the smallest number s such that there is essentially only one value of the secret key K that is consistent with the ciphertexts $C_1, \ldots, C_s$.*

In other words, the unicity distance is the smallest s, s.t.

$$H(K \mid C_1, \ldots, C_s) \simeq 0$$

The unicity distance depends on both the key size and on the redundancy in the plaintext space. Redundancy is an effect of the fact that certain plaintexts appear more frequently than others. For a block cipher of size n, **the redundancy** ρ is defined as

$$\rho = 1 - H(\mathbf{P})/n$$

where $\mathbf{P}$ is the random variable representing the plaintexts. $H(\mathbf{P})/n$ estimates the average number of bits of information per bit in a plaintext.

Theorem 4.3.1 *(Shannons formula) The unicity distance of a cipher is*

$$n_{ud} = \frac{H(\mathbf{K})}{\rho}$$

where ρ is the redundancy of the plaintext space.

The smallest number N_{ud}, such that N_{ud} is a multiple of n, the block size, and $N_{ud} \geq n_{ud}$, is the least number of ciphertext bits an attacker needs from a legitimate sender in order to at least in principle be able to determine a unique key in a ciphertext only attack.

Example 4.3.1 ([63]) *The redundancy of English language messages is about* 0.8. *So for the DES, $k = 56$, $n = 64$ and*

$$n_{ud} = \frac{56}{0.8} \simeq 70$$

Therefore N_{ud} is 128 bits, the same is two ciphertext blocks.

Although the unicity distance is small as in the example, it does not necessarily mean that the DES can be broken using only 2 known ciphertexts.

First of all, Shannons measures are made using a random cipher model, but more important, the unicity distance gives no indication of the computational difficulty in breaking a cipher, merely a lower bound on the amount of ciphertext needed in a ciphertext only attack. However, if the plaintext space contains (close to) no redundancy, the unicity distance will tend to infinity, i.e., $n_{ud} \rightsquigarrow \infty$ as $\rho \rightsquigarrow 0$. In this case a ciphertext only attack will never succeed. A cipher, for which it can be shown that $H(K \mid C_1, \ldots, C_s)$ never approaches zero, even for large s, is called a **strongly ideal** cipher. The redundancy can be reduced heavily by inserting random bits in the plaintext.

Example 4.3.2 ([63]) *By adding 63 random bits to every bit of the plaintext, the unicity distance becomes*

$$n_{ud} = \frac{56 \times 64}{0.8} \simeq 4480$$

However, this slows down the perfoimance of the block cipher. The legitimate sender must encrypt and transmit 64 times as much plaintext as when no random bits are inserted. Using an insecure channel for transmission this allows an enemy to get 64 times as much ciphertext as before. A better way to remove redundancy in a plaintext space is by data compression, but no known methods achieve perfect data compression [55]. Since perfect and strongly ideal ciphers are both impractical, Shannon also considered computationally secrecy, or practical secrecy.

4.4 Practical Secrecy

Traditionally, cryptanalysis has been very focused on finding the key K of a secret key cipher. We classify the types of breaking a cipher as follows, inspired by the classification of forgeries on digital signature systems given by Goldwasser, Micali and Rivest in [31, 32].

- **Total break.**
  An attacker finds the secret key K.
- **Global deduction.**
  An attacker finds an algorithm A, functionally equivalent to $E_K(\cdot)$ (or $D_K(\cdot)$) without knowing the key K.

- **Instance (local) deduction.**
  An attacker finds the plaintext (ciphertext) of an intercepted ciphertext (plaintext), which he did not obtain from the legitimate sender.

- **Information deduction.**
  An attacker gains some (Shannon) information about key, plaintexts or ciphertexts, which he did not get directly from the sender and which he did not have before the attack.

We assume that all the above attacks are independent of how the keys used by the legitimate parties are chosen, i.e., we use Assumption 4.1.1. A global deduction is possible when a block cipher contains a "block structure". If certain subsets of the ciphertext are independent of certain subsets of the plaintext, then no matter how long the key is, the block cipher is vulnerable to a global deduction in a known plaintext attack. An instance deduction can be as dangerous as a total break, if the number of likely plaintexts is small. Consider the situation where the block cipher is used for encrypting a key in a key-exchange protocol. Here only one plaintext is encrypted and a total break is equal to an instance deduction. Information deduction is the least serious attack, however the legitimate parties are often interested in that no information at all about the plaintexts are obtained by any enemies.

From the above definitions we might derive the rule, that a block cipher is secure, if an enemy cannot do an information deduction in an adaptively chosen plaintext attack. But there are trivial attacks, which we have to consider first.

Brute-force (trivial)

- **Total break.** All block ciphers are totally breakable in a ciphertext only attack, simply by trying all keys one by one and check whether the computed plaintext is meaningful, using only about N_{ud} ciphertexts. This attack requires the computation of about 2^k encryptions. This number is not accurate and will increase for lower redundancy in the plaintexts.

  To the other extent, the table look-up attack, where the attacker, in a pre-computation phase, encrypts a fixed plaintext P under all possible keys and sorts and stores the ciphertexts, he obtains. Thereafter the

cipher is total breakable in a chosen plaintext attack requiring one chosen plaintext. There might be some keys encrypting P into the same ciphertext. Repeating the attack a few times with $P' \neq P$ will give a unique key.

- **Global deduction.** All block ciphers are globally deducible in a known/chosen plaintext attack. Simply get and store all possible plaintext/ciphertext pairs. The running time of a deduction is the time of one table lookup, i.e., negligible.

- **Instance deduction.** All block ciphers are instance deducible in a known plaintext attack using $2^n - 1$ known plaintexts, since the exclusive-or of all the intercepted ciphertexts will be the ciphertext of the remaining plaintext, for which the attacker did not get the ciphertext.

- **Information deduction.** All block ciphers are information deducible in a ciphertext only attack. Consider a block cipher used in the ECB mode. Denote two plaintexts by P_i and P_j and assume that an attacker intercepted the two corresponding ciphertext blocks, C_i and C_j. All entropy quantities after the interception are primed. It follows that $C_i \neq C_j \Rightarrow P_i \neq P_j$, which means that $H'(P_i \mid P_j) < H(P_i \mid P_j)$, i.e., the uncertain about P_i given P_j decreases, since we know that the plaintexts are different. Since $I(P_i; P_j) = H(P_i) - H(P_i \mid P_j)$, it follows that $I'(P_i; P_j) > I(P_i; P_j)$, i.e., the attacker has gained information. Obviously, if $C_i = C_j \Rightarrow P_i = P_j$, information is also gained. A similar result holds for block ciphers used in CBC mode.

Also, Hellman [33] has presented a time-memory trade-off attack on any block cipher, which finds the secret key after $2^{2k/3}$ encryptions using $2^{2k/3}$ words of memory. The $2^{2k/3}$ words of memory are computed in a preprocessing phase, which takes the time of 2^k encryptions.

The above illustrates that we have to consider both the time and the amount of data needed in an attack. Also of great importance are the storage requirements.

- **Data complexity.** The amount of data needed as input to an attack. Units are measured in blocks of length n. We denote this complexity C_d.

- **Processing complexity.** The time needed to perform an attack. Time units are measured as the number of encryptions an attacker has to do himself. We denote this complexity C_p.
- **Storage requirements.** The words of memory needed to do the attack. Units are measured in blocks of length n. We denote this complexity C_s.

As a rule of thumb, the complexity of an attack is taken to be the maximum of the three complexities, i.e., $C_a = max(C_d, C_p, C_s)$. In general, there are some deviations from this rule and furthermore the three types of complexity of an attack are relative to the attacker. As an example, we may say that the above attack by Hellman [33] on the DES has complexity $2^{2\times 56/3} \simeq 2^{38}$. Although the time of the pre-computation phase is 2^{56}, first of all, it is done only once after which any DES-key can be derived with complexity 2^{38}, secondly 2^{56} DES encryptions can be done reasonable fast in hardware on specially designed machines [112]. On the other hand, the storage requirements may be unrealistic for most attackers, e.g. the attack on the DES will require about 2^{20} Mbytes of memory.

Definition 4.4.1 (Weak definition of practical security) *A block cipher with block size n and key size k is practically secure, if an enemy cannot do an information deduction in an adaptively chosen plaintext attack with a complexity significantly lower than a brute force attack, i.e., with complexity* $C_a << min(2^k, 2^n)$

As indicated this definition of practical security is weak. What is "significantly lower"?? We will show that for almost all applications of block ciphers used for encryption, we can establish an upper bound of the complexity, C_{up}, of an information deduction in an adaptively chosen plaintext attack. For the remainder of this section we will always assume that the plaintext space contains some kind of redundancy.

Theorem 4.4.1 *Eveyy block cipher used in the Electronic Code Book (ECB) mode is information deducible with a non-trivial information gain in a ciphertext only attack with complexity about* $2^{H(\mathbf{P})/2}$.

Proof: Recall that $H(\mathbf{P})$ is the entropy of the plaintext space, that is, there are approximately $2^{H(\mathbf{P})}$ meaningful messages. Assume that $H(P_i \mid P_j) \geq 1$

for $i \neq j$, i.e., there is at least one bit uncertainty about one plaintixt P_i given another plaintext P_j before the attack. By the "birthday paradox" (1.1.1) in a collection of $t = \sqrt{2^{H(\mathbf{P})}} = 2^{H(\mathbf{P})/2}$ ciphertexts $C_1, \ldots, C_t$ with a high probability there will exist a pair (i, j), s.t. $C_i = C_j$ and $P_i = P_p$ Clearly $H'(P_i \mid P_j) = 0$ and $I'(P_i; P_j) - I(P_i; P_j) \geq 1$, i.e., the information gained is non-trivial.

Because of the redundancy in the plaintexts, the attacker can obtain (valuable) information about the plaintext. If the plaintexts are in natural English represented in ASCII characters, the single letter frequency of the English language can be exploited character by character. Of course also digram, trigram and N-gram frequencies can be used. By using a larger collection of ciphertexts the probability of finding $P_i = P_j = P_k$ increases, which greatly improves the attackers knowledge of the plaintexts. □

Remark, that the upper bound $2^{H(\mathbf{P})/2}$ is met only when all plaintexts have equal probabilities. The more redundancy in the plaintext space the less the complexity of the above attack. Theorem 4.4.1 is a trivial result and is why it is often recommended to use the cipher block chaining mode (CBC), when encrypting large plaintexts. See Appendix A for an illustration. However, for the CBC we have the following result.

Theorem 4.4.2 *Every block cipher used in the Cipher Block Chaining (CBC) mode is information deducible in a ciphertext only attack with complexity about* $2^{n/2}$.

Proof: By the birthday paradox in a collection of $t = \sqrt{2^n} = 2^{n/2}$ n-bit ciphertexts $C_1, \ldots, C_t$ there will with a high probability exist a pair (i, j), s.t. $C_i = C_j$ and thereby

$$\begin{aligned} E_K(P_i \oplus C_{i-1}) &= E_K(P_j \oplus C_{j-1}) \Rightarrow \\ P_i \oplus C_{i-1} &= P_j \oplus C_{j-1} \end{aligned}$$

Since by assumption we know the ciphertexts C_{i-1} and C_{j-1}, we can compute $P_i \oplus P_j = C_{i-1} \oplus C_{j-1} = \alpha$. It follows from the proof of Theorem 4.4.1, that the information gained is non-trivial. If the plaintexts are redundant then so is the exclusive-or of pairs of plaintexts. □

Here we have assumed that for a block cipher with a fixed key, when restricted to a subset of plaintexts, the corresponding ciphertexts are dis-

tributed uniformly on the set of all possible ciphertexts. Sometimes one makes the more informal assumption that the block cipher "behaves almost like a random function". There is a similar result for the CFB mode, also mentioned by Maurer [70],

Theorem 4.4.3 *Every block cipher used in the Cipher Feedback (CFB) mode is information deducible in a ciphertext only attack with complexity about* $2^{n/2}$.

Proof: Let D_i be a collection of n/m consecutive m-bit ciphertext blocks starting with C_i, i.e., $D_i = C_i, \ldots, C_{i+(n/m-1)}$. By the birthday paradox in a collection of $t = \sqrt{2^n} = 2^{n/2}$ n-bit ciphertexts $D_1, \ldots, D_t$ there will exist a pair (i, j), s.t. $D_i = D_j$, which means that $X_{i+n/m} = X_{j+n/m}$ and

$$\begin{aligned} C_{i+n/m} = P_{i+n/m} \oplus E_K(X_{i+n/m}) \quad \wedge \quad C_{j+n/m} = P_{j+n/m} \oplus E_K(X_{j+n/m}) \Rightarrow \\ P_{i+n/m} \oplus C_{i+n/m} \quad = \quad P_{j+n/m} \oplus C_{j+n/m} \end{aligned}$$

Since by assumption we know the ciphertexts $C_{i+n/m}$ and $C_{j+n/m}$ we can compute $P_{i+n/m} \oplus P_{j+n/m} = C_{i+n/m} \oplus C_{j+n/m} = \alpha$. □

It is easy to see that the CBC mode and the CFB mode are both information deducible in a chosen ciphertext attack using only two chosen ciphertexts. Instead of using the birthday paradox to find two equal ciphertexts in a large collection of ciphertexts one simply chooses two ciphertext blocks equal. However, this is a very non-realistic attack and if the plaintext space contains redundancy there is only little chance that a chosen ciphertext decrypts into a meaningful plaintext.

Motivated by Theorem 4.4.1, 4.4.2 and 4.4.3 and from the fact that $H(\mathbf{P})/2$ is at most $n/2$ we get an upper bound for practical security and define

Definition 4.4.2 *A block cipher used in the ECB, CBC end the CFB modes with block size n and key size k is* **practically secure**, *if an enemy cannot do at leaat an information deduction with a non-trivial information gain in an adaptively chosen text attack with a complexity lower than* $C_{up} \simeq 2^{n/2}$.

Finally we note that the attacks in Theorem 4.4.1, 4.4.2 and 4.4.3 are independent of the key size. In the following we analyse two other suggested modes.

4.4.1 Other modes of operation

Apart from the four standard modes of Section 3.1 other modes have been suggested to improve the performance of a block cipher. In this suction we examine two other modes of operation. We examine both modes with respect to unicity distance, error propagation and attacks like the ones of Theorems 4.4.1, 4.4.2 and 4.4.3.

Davies-Price mode

In [22] Davies and Price suggest the following mode, but claim "no special virtues for this mode". Encryption

$$C_i = E_{K_1}(P_i \oplus E_{K_2}(C_{i-1}))$$

where C_0 is an initial value. Decryption

$$P_i = D_{K_1}(C_i) \oplus E_{K_2}(C_{i-1})$$

This is a variant of the CBC mode and uses two keys. Therefore the unicity distance is increased by a factor of two. Theorem 4.4.2 is not directly applicable, since it requires a match in pairs of two blocks, in other words we need to collect 2^n blocks of ciphertexts for a match. It also clear, however, that this mode is as vulnerable to a meet-in-middle-attack as a conventional double encryption scheme in CBC mode, see Section 7.9, which requires that two consecutive ciphertext blocks (and one plaintext block) are available. Simply decrypt C_i by all possible values of K_1 and store the values. Then encrypt C_{i-1} by all possible values of K_2 and for every value check whether the exclusive-or of this results with any value in the table yields a possible (or an intercepted) plaintext. However, we can use the same methods as in the proof of Theorem 4.4.2 to do an exhaustive search for the key K_1, K_2 with less memory. Collect $2^{n/2}$ ciphertext blocks and find a match $C_i = C_j$. Then

$$\begin{aligned} C_i &= C_j \Rightarrow \\ P_i \oplus E_{K_2}(C_{i-1}) &= P_j \oplus E_{K_2}(C_{j-1}) \Rightarrow \\ P_i \oplus P_j &= E_{K_2}(C_{i-1}) \oplus E_{K_2}(C_{j-1}) \end{aligned}$$

If the plaintexts contain redundancy, so does the exclusive-or of pairs of plaintexts. Therefore by doing an exhaustive search over K_2, we can check

if $E_{K_2}(C_{i-1}) \oplus E_{K_2}(C_{j-1})$ yields a likely exclusive-or of two plaintexts. The probability of the attack can be improved by finding more matches $C'_i = C'_j$. Also note that a match $C_i = C_j$ enables us to do an exhaustive search over K_1 as

$$C_i = C_j \Rightarrow P_{i+1} \oplus P_{j+1} = D_{K_1}(C_{i+1}) \oplus D_{K_1}(C_{j+1})$$

An error in the transmission of one ciphertext block propagates to two plaintext blocks. This is also the case, when a ciphertext block is deleted or an extra ciphertext block is inserted. The difference from the CBC mode is that here the two affected plaintext blocks are completely garbled.

OFBNLF mode

In [39] the "OFB with a Non-linear Function (OFBNLF)" is suggested. In [93] this nonlinear function is taken to be the cipher itself. The secret key is K. Encryption

$$C_i = E_{K_i}(P_i),\ K_i = E_K(K_{i-1})$$

where K_0 is an initial value. Decryption

$$P_i = E_{K_i}(C_i),\ K_i = E_K(K_{i-1})$$

This is a variant of the OFB mode, as the name indicates, but also a variant of the ECB mode. Theorem 4.4.1 is not applicable, since different keys are used to encrypt different plaintexts. It does not mean that the unicity distance increases, it remains the same as in the ECB mode. An error in the transmission of one ciphertext block propagates to only one plaintext block. However, an infinite error extension arises, when one ciphertext block is lost, or if an extra ciphertext block is inserted by an enemy. Applied to the DES, this mode will be slow in software applications, where the key scheduling is often slow. If the key scheduling takes twice the time of an encryption, this mode takes three times one DES encryption (with a fixed key) for encryption of one plaintext block.

Chapter 5

Cryptanalysis of Block Ciphers

Cryptanalysis is fun, especially in the morning ... at breakfast.

In this chapter cryptanalysis of block ciphers is considered. After a short introduction we describe differential cryptanalysis in Section 5.2. We give a simple method to find an important class of characteristics for DES-like ciphers to be used in differential attacks. Next we consider differentials, higher order differentials and partial differentials and show their applications. In Section 5.3 the method of linear cryptanalysis is described. We give a similar method as the one above for differentials to find an important class of characteristics for DES-like ciphers to be used in linear attacks. In Section 5.4 cryptanalysis of the key schedules in block ciphers is considered and it is shown how simple relations in the key schedules can be exploited in cryptanalytic attacks. Finally we define a new class of keys, the weak hash keys, which can be exploited in attacks on hash functions based on block ciphers.

5.1 Introduction

The history of cryptanalysis is long and at least as fascinating as the history of cryptography. As an example, in 1917 in an article in "Scientific American" the Vigenére cipher was claimed to be "impossible of translation" [23]. The Vigenére and in general substitution ciphers can be broken

when enough ciphertext is available to the cryptanalyst by the index of coincidence, Kasiski's method, etc. [22, 23, 40]. Transposition ciphers can be broken using the frequency distributions for digrams, trigrams and N-grams [22, 23, 40]. The interested reader will find a comprehensive treatment of early cryptanalysis in [40].

The most well-known method of analysing conventional cryptosystems today is *differential cryptanalysis*, published by Eli Biham and Adi Shamir in 1990. The method has proved to be very efficient and cryptosystems, which have been conjectured strong, have been broken, for some systems (e.g. GDES) almost alarmingly easy [7]. Differential cryptanalysis is a chosen plaintext attack, in which the attacker chooses plaintexts of certain well-considered differences. Although a chosen plaintext attack may not be a realistic attack in most settings, Biham and Shamir's attack on the full 16-round DES [7, 8] is the first attack in the open literature capable of finding the secret key faster than an exhaustive search of the key space.

Another method of analysing conventional cryptosystems, is *linear cryptanalysis*, proposed by Matsuru Matsui in 1993 [64]. A preliminary version of the attack on FEAL was described in 1992 [68]. The attack on the DES [65, 66] has proved to be more efficient than the attack based on differential cryptanalysis. First of all, the attacks based on linear cryptanalysis are known plaintext attacks and secondly the attack on the DES is faster than the attack by Biham and Shamir.

A third method of analysing conventional cryptosystems, is by means of *related keys*. The author introduced the method by giving a chosen plaintext attack on LOKI'91 [47], reducing an exhaustive key search by almost a factor of four. Later Biham improved the attack [3] on LOKI'91, reducing an exhaustive key search by almost a factor of six.

5.2 Differential Cryptanalysis

Differential cryptanalysis has been applied to a wide range of iterated ciphers including the DES [90], GDES [102, 104], Lucifer [109], FEAL [79], LOKI'89 [15], REDOC [17], PES [58] and Khafre [75]. For this reason the differential attack must be considered one of the most general cryptanalytic attacks known to date. Furthermore, differential cryptanalysis has caused the re-

vision and redesign of several cryptosystems and was the first attack which could (theoretically) recover DES keys in time less than the expected cost of exhaustive search [7, 8]. Differential cryptanalysis is universal in that it can be used against any cryptographic mapping which is constructed from iterating a fixed round function. We will give a brief description of differential cryptanalysis with respect to a general $2n$-bit iterated cipher.

We define a **difference** between two bit strings, X and X' of equal length as

$$\Delta X = X \otimes (X')^{-1}$$

where $\otimes$ is the group operation on the group of bit strings used to combine the key with the text input in the round function and where $(X)^{-1}$ is the inverse element of X w.r.t. $\otimes$. The idea behind this is, that the difference between the texts before and after the key is combined is equal, i.e., the difference is independent of the key. To see this, note that

$$\Delta X = (X \otimes K) \otimes (X' \otimes K)^{-1} = X \otimes K \otimes K^{-1} \otimes X'^{-1} = X \otimes (X')^{-1}$$

For most Feistel ciphers, including the above, it is possible to compute tables, so-called *dfference distribution tables*, containing the possible differences in the outputs for every difference in the inputs and the corresponding probabilities for one round of the cipher.

For a plaintext $P = C_0$ recall that C_i is the ciphertext after i rounds of encryption. An r-round *characteristic* is a series of differences defined as an $(r+1)$-tuple $(\alpha_0, \ldots, \alpha_r)$, where α_i is the expected value of ΔC_i and where α_0 is the chosen value of $\Delta P = \Delta C_0$. Here ΔP is said to be a plaintext difference and, ΔC_i is the ciphertext difference after i rounds of encryption. The probability of a characteristic is the conditional probability that $\Delta C_i = \alpha_i$ is the difference after i rounds given that $\Delta C_{i-1} = \alpha_{i-1}$ is the difference after $i-1$ rounds. More formally, for a random, uniformly selected round keys K_i, the probability of a characteristic is

$$\Pr(\Delta C_i = \alpha_i, \Delta C_{i-1} = \alpha_{i-1}, \ldots, \Delta C_1 = \alpha_1 \mid \Delta P = \alpha_0) \tag{5.1}$$

This probability can be hard to calculate. However, for certain ciphers the probability can be calculated from the probabilities of one-round characteristics, as we will show now. A sequence of stochastic variables $v_0, v_1, \ldots, v_r$

is a Markov chain, if for $0 \leq i < r$

$$\Pr(v_{i+1} = \beta_{i+1} \mid v_i = \beta_i, v_{i-1} = \beta_{i-1}, \ldots, v_0 = \beta_0 = \Pr(v_{i+1} = \beta_{i+1} \mid v_i = \beta_i)$$

A Markov chain is called *homogeneous*, if

$$\Pr(v_{i+1} = \beta \mid v_i = \alpha)$$

is independent of i for all α and β.

Definition 5.2.1 (Lai [55]) *An iterated cipher is called a* **Markov cipher**, *if there is a group operation $\otimes$, such that*

$$\Pr(\Delta C_1 = \beta \mid \Delta C_0 = \alpha,\ C_0 = \gamma) \tag{5.2}$$

is independent of γ for all α and β (both $\neq e$, the neutral elenaent of the group), when the round key K is uniformly random.

Theorem 5.2.1 (Lai [55]) *If an r-round iterated cipher is a Mavkov cipher and the r round keys are independent and uniformly random, then the sequence of differences*

$$\Delta P = \Delta C_0, \Delta C_1, \ldots, \Delta C_r$$

is a homogeneous Markov chain.

This means that for a Markov cipher the probability of an s-round characteristic (5.1) can be computed as follows

$$\Pr(\Delta C_s = \alpha_s, \Delta C_{s-1} = \alpha_{s-1}, \ldots, \Delta C_1 = \alpha_1 \mid \Delta P_0 = \alpha_0) = \prod_{i=1}^{s} \Pr(\Delta C_1 = \alpha_i \mid \Delta P = \alpha_{i-1}) \tag{5.3}$$

Theorem 5.2.2 *A DES-like iterated cipher, cf. Definition 2.5.3, is a Markov cipher with the difference induced by the '+' operation, if the found keys are independent and uniformly random.*

Proof: We will show that the expression in Definition 5.2.1 is independent of $\gamma = \gamma^L \parallel \gamma^R$. In a DES-like cipher $\Delta C_1^L = \Delta C_0^R$ independent of γ, so it suffices to prove the case of ΔC_1^R.

$$
\begin{aligned}
\Pr(\quad & \Delta C_1^R = \beta^R \mid \Delta C_0 = \alpha, C_0 = \gamma) = \\
\Pr(\quad & f(E(C_0^R) + K) - \\
& f(E(C_0^R + \alpha^R) + K) + \alpha^L = \beta^R \mid C_0^R = \gamma^R, \Delta C_0 = \alpha) = \\
\Pr(\quad & f(E(C_0^R) + K) - \\
& f(E(C_0^R) + K + E(\alpha^R)) + \alpha^L = \beta^R \mid C_0^R = \gamma^R, \Delta C_0 = \alpha) = \\
\Pr(\quad & f(X) - f(X + E(\alpha^R)) + \alpha^L = \beta^R \mid C_0^R = \gamma^R, \Delta C_0 = \alpha)
\end{aligned}
$$

where $X = E(C_0^R + K)$ is uniformly distributed, since K is. The probability does not depend on γ and the proof is complete. □

Experimental results on DES, LOKI'89 and FEAL [7, 6, 48] have shown that in these ciphers (5.3) also holds, when the round keys are derived from a key schedule algorithm. To simplify statistical arguments we assume in the following that the round keys are independent and uniformly random.

A plaintext pair P, P' of difference ΔP is called a *right pair* with respect to a key K and an r-round characteristic if when the pair P, P' is encrypted, the difference in the intermediate ciphertexts follow the characteristic. About $p \cdot 2^{2n}$ pairs are right pairs, where p is the probability of the characteristic and $2n$ is the block size of the cipher. On the other hand, if P, P' is not a right pair, then it is said to be a *wrong pair* (with respect to the characteristic and the key).

Differential cryptanalysis attempts to determine the round key K_r used in the final round of the cipher. Consider an iterated block cipher as defined in Definition 2.5.1. Let C_r and C'_r be the ciphertexts for some plaint ext pair. In a chosen plaintext attack the cryptanalyst does not know the inputs C_{r-1} and C'_{r-1} to the final round. However, a characteristic can be chosen so that the difference of the ciphertexts after $r-1$ rounds of encryptions, ΔC_{r-1}, is known either completely or partially with probability p. Then for two plaintexts P, P' of difference ΔP, the cryptanalyst can solve the following equation for K_r

$$g^{-1}(C_r, K_r) \otimes g^{-1}(C'_r, K_r)^{-1} = \Delta C_{r-1} \tag{5.4}$$

Let the solutions be $k_1, k_2, \ldots, k_j$, which we will call *candidate round keys*. If P, P' is a right pair then $K_r \in \{k_1, k_2, \ldots, k_j\}$. On the other hand, if P, P' is a wrong pair then we assume that the k_i are independent of K_r. Then if many pairs P, P' are examined, and the frequency of the candidate keys is recorded, we expect the correct round key K_r to be counted more often than

other keys. The method of differential cryptanalysis can be summarised as follows:

Step 1 Find an $r-1$-round characteristic $(\Delta P, \Delta C_1, \Delta C_2, \ldots, \Delta C_{r-1})$ which (partially) determines ΔC_{r-1} with a high probability.

Step 2 Uniformly select a plaintext pair P, P' with difference ΔP and get the encryptions of this pair, assuming that P, P' is a right pair. Determine candidate round keys $k_1, k_2, \ldots, k_j$ such that each k_i could have caused the observed output difference. Increment a counter for each candidate round key k_i.

Step 3 Repeat Step 2 until one round key k_i is distinguished as being counted significantly more often than other round keys. Take k_i to be the actual round key K_r.

It is then natural to define the *complexity* of a differential cryptanalysis to be the number of encrypted plaintext pairs of a specified difference required to determine the key or round key. from experiments on restricted versions of DES, Biham and Shamir [7] found that the complexity of the attack was approximately c/p, where p is the probability of the characteristic being used, and c is a constant bound as $2 < c < 8$.

To measure the efficiency of a differential attack Biham and Shamir use the so-called signal to noise ratio [7]. Assume that m pairs of chosen plaintexts are used in a differential attack and that p is the probability of the characteristic used. Then about $m \times p$ pairs are right pairs, each of which suggest the right key value among other values. In some attacks [7] the attacker can determine pairs of plaintexts as wrong pairs from the intercepted ciphertexts, in which case the pair is discarded and not used in the analysis. Let k be the number of possible values of the key, we are looking for, γ is the number of keys suggested by each non-discarded pair of plaintexts and λ is the ratio of non-discarded pairs to all pairs. The average number of times a random (wrong) key is suggested is now $\frac{m \times \gamma \times \lambda}{k}$. The signal to noise ratio, S/N, is the number of times the right key is counted over the number of times a random key is counted, i.e.,

$$S/N = \frac{m \times p}{\frac{m \times \gamma \times \lambda}{k}} = \frac{k \times p}{\gamma \times \lambda}$$

A necessary condition for the success of a differential attack is that the signal to noise ratio is greater than one, and the expected success of the attack increases with the ratio. The quantity λ is quite important for differential attacks on DES-like iterated ciphers.

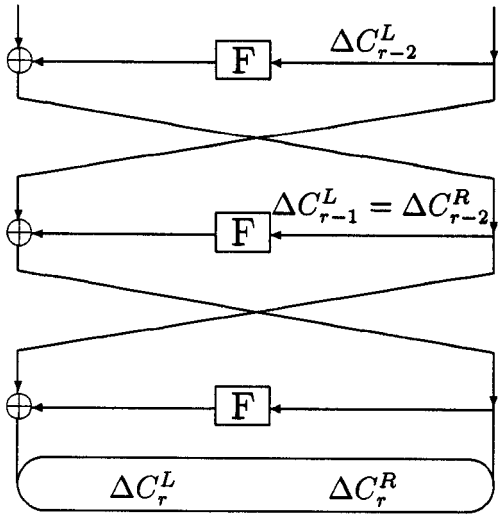


Figure 5.1: An r-round DES-like cipher.

Differential attacks on DES-like ciphers

For DES-like iterated ciphers the inputs to the F-function in the last round can be read as the right halves of the ciphertexts. In a differential attack on an r-round cipher, cf. above, knowledge about the difference of pairs of ciphertext after r - 1 rounds is necessary. In a DES-like iterated cipher the important difference to know for an attacker is the difference of the outputs of the F-function of the last round. But since $\Delta C^L_i = \Delta C^R_{i-1}$, in some cases it suffices to obtain knowledge about the difference of pairs of ciphertext after r - 2 rounds. The difference of the outputs of the F-function in the r'th round can then be read as the exclusive-or of ΔC^R_{r-2} and ΔC^L_r (the left halves of the difference in the ciphertexts), see Figure 5.1. Assume we are considering an r-round DES-like cipher with block size $2n$ bits. Knowledge about ΔC^R_{i-2} is used to determine the difference of the outputs of F and ΔC^L_{i-2} and to discard wrong pairs in the following way. If $\Delta C^L_{r-1} = \Delta C^R_{r-2}$ can lead to

only a fraction of all differences, say $\mathcal{M}$, in the outputs if the F-function in the second last round, an attacker can check whether $\Delta C_{r-2}^L \oplus \Delta C_r^R$ yields one of the possible differences in $\mathcal{M}$. If it does not, the pair is discarded and not used any further in the analysis. This prevents some suggested wrong values of the key. In that way, even though knowledge about only ΔC_{r-2}^R is used to obtain suggested key values, the knowledge about ΔC_{r-2}^L is in some cases crucial for a successful differential attack. To illustrate that, assume that $\Pr(\Delta C_{r-2} = \beta \mid \Delta P = \alpha) < \sqrt{2^{-2n}} = 2^{-n}$, and that the discarding of wrong pairs is avoided. Then the signal to noise ratio is less than one, since any other (random) value of ΔC_{r-2}^R has a probability of about 2^{-n} and the differential attack would fail.

The first round trick

In their attack on the full 16-round DES Biham and Shamir introduced, what we will call, the *first round trick*. Assume an attacker found a characteristic whose first two rounds have the values of Figure 5.2, where a difference Φ in the inputs to the F-function can lead to a difference Ψ in the outputs of the F-function with probability $p < 1$. This will be denoted $\Psi \leftarrow \Phi$.

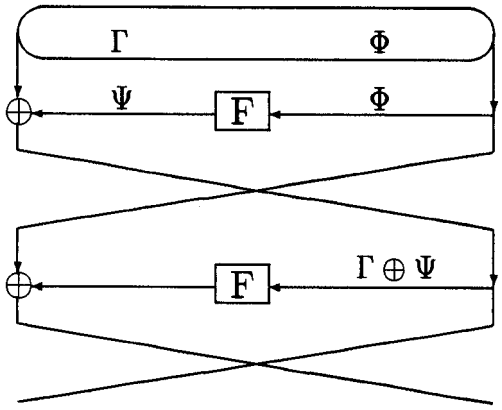


Figure 5.2: Two rounds in a characteristic.

In a conventional differential attack using $2N$ plaintexts to form N pairs of plaintexts, the attacker would get about $N \times p$ pairs of ciphertexts, whose difference after one round is $(\Phi, \Gamma \oplus \Psi)$, where p is the probability of the characteristic in the first round. This can be improved by choosing the

chosen plaintexts more carefully in the following way. Assume that

$$|\{v \mid v \leftarrow \Phi\}| = n$$

i.e., the number of possible output differences of the F-function is n, when Φ is the difference in the inputs. We number these values $v_1, \ldots, v_n$. Assume that the set $\{v_i\}$ is closed under the exclusive-or operation. If it is not, we extend it to be closed. Pick a random plaintext $P = (P^L \mid P^R)$. Choose n plaintexts of the form

$$(P^L \oplus v_j \oplus \Gamma \mid P^R \oplus \Phi)$$

and n plaintexts of the form

$$(P^L \oplus v_i \oplus \Psi \mid P^R)$$

By pairing each plaintext from the first set with each plaintext of the second set we obtain n^2 pairs of plaintexts, whose characteristic will have the form of Figure 5.3. But since there are only n possible values of v_l and of v_k by definition in n pairs out of the n^2 pairs $v_l = v_k$. That means that from $2n$ plaintexts the attacker gets n pairs with the desired difference after one round of encryption. The efficiency of this first round trick depends on the value of n. If n is too large the pairing of the two sets of plaintexts may increase the overall complexity of the differential attack too much.

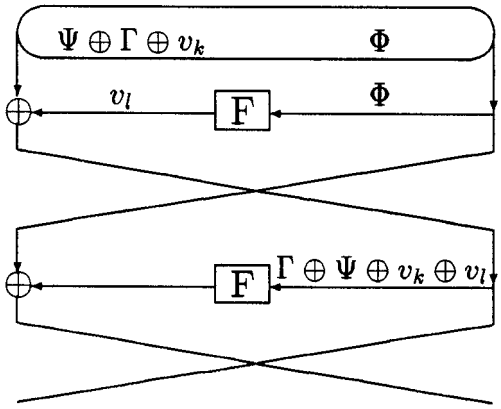


Figure 5.3: The first round trick.

The first two rounds trick

The above trick can be extended to two rounds in the following way. Assume an attacker has found a characteristic, whose first three rounds have the values in Figure 5.4. Assume now that

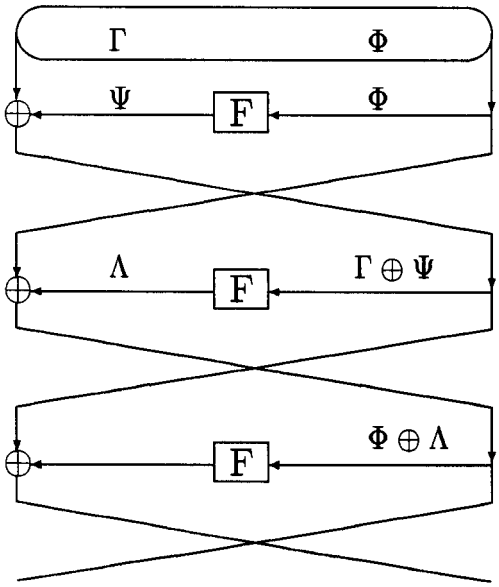


Figure 5.4: Three rounds in a characteristic.

$$|\{w \mid w \leftarrow \Gamma \oplus \Psi\}| = n_1,$$

and number these values $w_1, \ldots, w_{n_1}$. For the values w_i assume that

$$\sum_{i=1}^{n_1} |\{v \mid v \leftarrow \Phi \oplus w_i\}| = n_2,$$

and number these values $v_1, \ldots, v_{n_2}$. Assume that both sets are closed under the exclusive-or operation. Pick a random plaintext $P = (P^L \mid P^R)$. Choose $n_1 \times n_2$ plaintexts of the form

$$(P^L \oplus v_i \oplus \Gamma \mid P^R \oplus \Phi \oplus w_j)$$

and $n_1 \times n_2$ plaintexts of the form

$$(P^L \oplus v_l \oplus \Psi \mid P^R \oplus w_m)$$

By pairing each plaintext from the first set with each plaintext of the second

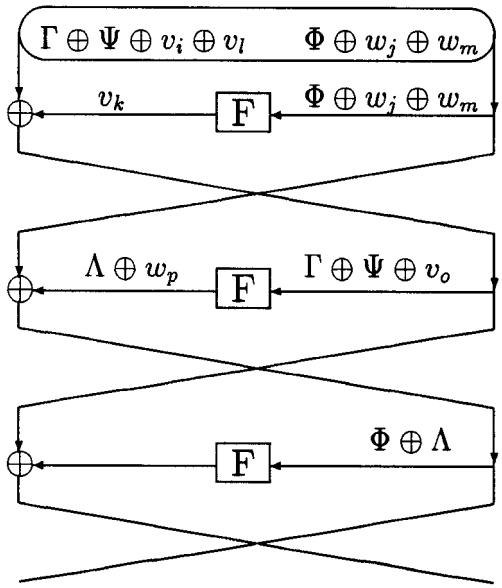


Figure 5.5: The first two rounds trick.

set we obtain $(n_1 \times n_2)^2$ pairs of plaintexts, whose characteristic will have the form of Figure 5.5. Since there are only n_2 possible values of v_k for about $(n_1 \times n_2)^2/n_2 = n_1^2 \times n_2$ pairs of plaintexts, the difference in the ciphertexts after one round will be $(\Phi \oplus w_j \oplus w_m \mid \Gamma \oplus \Psi)$ for some j and m, i.e., $v_0 = 0$. For one out of every n_1 of the remaining pairs $w_p = w_j \oplus w_m$, note that Λ is one of the n_1 possible w-values. It is seen that from $2 \times n_1 \times n_2$ chosen plaintexts an attacker gets about $(n_1 \times n_2)$ pairs with the desired difference after two rounds of encryption.

The first two rounds trick applied to the DES does not seem to improve a differential attack. By using the same characteristics as in the attack by Biham and Shamir [7], one gets structures with $n_1 \simeq 2^{12}$ and $n_2 \simeq 2^{32}$, i.e. structures of 2^{44} plaintexts. Although one can get one round further in the characteristic, the complexity of filtering out wrong pairs increases and slows down the attack.

The first two rounds trick seems to be applicable to the LOKI ciphers [15, 14], but more work has to be done in that direction.

5.2.1 Iterative characteristics

Most of the reported differential attacks make use of what is called iterative characteristics.

Definition 5.2.2 *For an iterated block cipher (see Definition 2.5.1) an s-round iterative characteritic is an s-tuple* $(\Delta C_i, \ldots, \Delta C_{i+s})$ *s.t.* $\Delta C_i = \Delta C_{i+s}$.

It is easily seen that an s-round iterative characteristic can be extended to an n-round characteristic, for any integer $n \geq s$.

5.2.2 Iterative characteristics for DES-like ciphers

In DES-like iterated ciphers equal inputs (to the F-function) always lead to equal outputs. This fact can be used to construct a 1-round characteristic with probability one, called a trivial one-round characteristic. An obvious attempt in construction of characteristics with a larger number of rounds is to include as many trivial one-round characteristics as possible. Building long characteristics from scratch has turned out to be a difficult task. An easier way is to construct iterative characteristics with a relatively small number of rounds, and concatenating these into longer characteristics. In the following we will show different models of iterative characteristics for DES-like iterated ciphers. In Sections 6.1, 6.2, 6.3 and 6.4 we will justify the usability of the models by showing concrete examples of these in DES, LOKI'91 and s^2-DES.

Recall that for a DES-like iterated cipher $\Delta C_i^L = \Delta C_{i-1}^R$. In this section we shall write $\Delta C_i = (\Delta C_i^L, \Delta C_i^R)$. The combinations $\beta \leftarrow \alpha$ of input and output differences for one round of most DES-like ciphers are easily calculated and saved in a difference distribution table. In the following an s-round characteristic is described as follows. On the first line we have the difference of the inputs to the characteristic, (ciphertexts after i round). Then for each round we state the difference of the inputs to the F-function and the resulting difference of the outputs of the F-function. On the last line the

difference of the ciphertexts after $(i+s)$-rounds is listed. We will also assume that the only one-round characteristic with probability one is obtained when the inputs to the F-function are equal. We call this a *zero-round*.

1-round iterative characteristics

For DES-like iterated ciphers a 1-round iterative characteristics must have the following form

$$0 \overset{(\Phi,\Phi)}{\underset{(\Phi,\Phi)}{\leftarrow}} \Phi \quad \text{prob. } p > 0$$

It comes from the fact that $\Delta C_i^L = \Delta C_{i-1}^R$. This characteristic is never used in attacks on DES-like iterated ciphers, since the combination $0 \leftarrow \Phi$ can be used to build a 2-round iterative characteristic with a better probability per round.

2-round iterative characterist its

Two consecutive zero-rounds in a characteristic of DES-like cryptosystems lead to equal inputs and outputs of all rounds. We get equal plaintexts resulting in equal ciphertexts, a trivial fact. The maximum occurrences of zero-rounds therefore is every second round. This situation evolves by using and iterating the following 2-round characteristic, also used in [7].

$$\begin{array}{lcll} & \overset{(\Phi,0)}{} & & \\ 0 & \leftarrow & 0 & \text{always} \\ 0 & \underset{(\Phi,\Phi)}{\leftarrow} & \Phi & \text{prob. } p > 0 \end{array}$$

3-round characteristics

We proceed to the situation where every third round in a characteristic is a zero-round.

$$
\begin{array}{rcll}
 & (\Gamma, 0) & & \\
0 & \leftarrow & 0 & \text{always} \\
\Phi & \leftarrow & \Gamma & \text{prob. } p_1 > 0 \\
\Gamma & \leftarrow & \Phi & \text{prob. } p_2 > 0 \\
 & (\Phi, 0) & &
\end{array}
$$

The differences of the inputs and outputs are not equal and is not iterative in the sense of Definition 5.2.2. However, the characteristic is one half of an iterative characteristic. Concatenated with the characteristic with rounds no. 2 and 3 interchanged we obtain:

$$
\begin{array}{rcll}
 & (\Gamma, 0) & & \\
0 & \leftarrow & 0 & \text{always} \\
\Phi & \leftarrow & \Gamma & \text{prob. } p_1 > 0 \\
\Gamma & \leftarrow & \Phi & \text{prob. } p_2 > 0 \\
0 & \leftarrow & 0 & \text{always} \\
\Gamma & \leftarrow & \Phi & \text{prob. } p_2 > 0 \\
\Phi & \leftarrow & \Gamma & \text{prob. } p_1 > 0 \\
 & (\Gamma, 0) & &
\end{array}
$$

In that way we get a 6-round iterative characteristic. Still we choose to call the 3-round characteristic an iterative characteristic.

4-round characteristic

When every fourth round is a zero-round we need a 4-round characteristic, which extended to 8 rounds becomes an iterative characteristic. It must have the following form:

$$
\begin{array}{rcll}
 & (\Gamma, 0) & & \\
0 & \leftarrow & 0 & \text{always} \\
\Phi & \leftarrow & \Gamma & \text{prob. } p_1 > 0 \\
\Gamma \oplus \Psi & \leftarrow & \Phi & \text{prob. } p_2 > 0 \\
\Phi & \leftarrow & \Psi & \text{prob. } p_3 > 0 \\
 & (\Psi, 0) & &
\end{array}
$$

It means that we have to find two input differences Ψ and Γ both resulting in Φ and Φ resulting in the difference between Ψ and Γ.

Longer characteristics

We can of course continue the search for n-round characteristics, $n > 4$. However, the complexity of finding the best combinations in an n-round iterative characteristic rapidly increases for larger n. For some ciphers it is not necessary to go much further than $n = 4$ as we will demonstrate for the ciphers, LOKI'89, LOKI'91 and s^2-DES. For the DES we also give strong evidence, that for $n > 4$ there are no 'good' iterative n-round iterative characteristics.

5.2.3 Differentials

A closer look at differential attacks shows that for an s-round characteristic $(\Delta P, \Delta C_1, \Delta C_2, \ldots, \Delta C_s)$ only the plaintext difference ΔP and the last ciphertext difference ΔC_s need to be fixed. That is, the intermediate differences $\Delta C_1, \Delta C_2, \ldots, \Delta C_{s-1}$ can be arbitrarily selected. The notion of *differentials* $(\Delta P, \Delta C_s)$ was introduced by Lai and Massey [58, 55] to account for this observation. The probability of an s-round differential $(\Delta P, \Delta C_s)$ is the conditional probability that given an input difference ΔP at the first round, the output difference at the s'th round will be ΔC_s. More formally, the probability of an s-round differential is given as

$$\Pr(\Delta C_s = \beta_s \mid \Delta P = \beta_0) = \sum_{\beta_1} \sum_{\beta_2} \cdots \sum_{\beta_{s-1}} \prod_{i=1}^{s} \Pr(\Delta C_i = \beta_i \mid \Delta C_{i-1} = \beta_{i-1}) \tag{5.5}$$

when $\Delta C_0 = \Delta P$. Here we are assuming that a sequence of differences can be modeled as a homogeneous Markov chain $\mathbf{P} = [P_{ij}], 1 \leq i, j < 2^n$. This subject is treated further in Section 7.4. Whereas in a Markov cipher the probability of an s-round characteristics can be easily calculated as the product of the probabilities of s one-round characteristics, the probabilities of s-round differentials for large s, $s > 2$, seem hard to calculate. Note that in a Feistel cipher the concepts of characteristics and differentials coincide for $s \leq 2$. LOKI'89 [15] was attacked using an iterative 3-round characteristic

[48]. This characteristic iterated to a 5-round characteristic has a probability of about 2^{-21}. The probability of a corresponding 5-round differential, i.e., where input and output differences are the same as in the characteristic, was approximated to $2^{-21} + 2^{-45}$, that is, not significantly higher than for the characteristic. Similar tests for characteristics and differentials with more than 5 rounds have a much higher complexity.

In order to make a successful attack on a DES-like iterated cipher by differential cryptanalysis the existence of good characteristics is sufficient. On the other hand to prove security against differential attacks for DES-like iterated ciphers one has to ensure that there is no differential with a probability high enough to enable successful attacks. Whereas it is difficult to approximate the probability of a specific differential, it is possible to determine lower bounds on the probabilities for all differentials. A bound on the probability of all differentials can be obtained in terms of p_{max}, the probability of the most likely 1-round difference. We will return to this topic in Chapter 7.

Hypothesis of stochastic equivalence

In a differential attack the attacker does not know the key. Therefore in finding a good differential, the attacker computes the probabilities of differentials assuming that all the round keys are uniformly random and independent. However, the pairs of encryption an attacker gets are encrypted using the same key, where the round keys are fixed and (can be) dependent. Put informally "there is a difference between what an attacker can expect to see and what he actually sees". In [55] this problem is dealt with by introducing the

Definition 5.2.3 (Hypothesis of stochastic equivalence.) *For virtually all high probability $(r-1)$-round diflerentials (α, β)*

$$Pr_P(\Delta C_1 = \beta \mid \Delta P = \alpha, K = k) \approx Pr_{P,K}(\Delta C_1 = \beta \mid \Delta P = \alpha,)$$

holds for a substantial fraction of the key values k.

In Section 6.1 we will show that for the DES the probability of the best known differential varies for different subspaces of the key space. The hypothesis of stochastic equivalence is further discussed in Section 7.4.

5.2.4 Higher order differentials

In [56] the definition of derivatives of cryptographic functions was given.

Definition 5.2.4 (Lai [56]) *Let $(S,+)$ and $(T,+)$ be Abelian groups. For a function $f : S \mapsto T$, the derivative of f at the point $a \in S$ is defined as*

$$\Delta_a f(x) = f(x+a) - f(x)$$

Definition 5.2.5 (Lai [56]) *Let f be as in Definition 5.2.4. The i'th derivative of f at the point $a_1, \ldots, a_i$ is defined as*

$$\Delta^{(i)}_{a_1,\ldots,a_i} f(x) = \Delta_{a_i}(\Delta^{(i-1)}_{a_1,\ldots,a_{i-1}} f(x))$$

Note that the characteristics and differentials used by Biham and Shamir in their attacks correspond to the first order derivative described by Lai. Therefore it seems natural to extend the notion of differential into **higher order differentials.**

Definition 5.2.6 *A one round differential of order i is an i+1-tuple $(\alpha_1, \ldots, \alpha_i, \beta)$, s.t.*

$$\Delta^{(i)}_{\alpha_1,\ldots,\alpha_i} f(x) = \beta$$

When considering functions over $GF(2)$ the points $a_1, \ldots, a_i$ must be linearly independent for the i'th derivative not to be trivial zero.

Proposition 5.2.1 (Lai [56]) *Let $L[a_1, a_2, \ldots, a_i]$ be the list of all 2^i possible linear combinations of $a_1, a_2, \ldots, a_i$. Then*

$$\Delta^{(i)}_{a_1,\ldots,a_i} f(x) = \sum_{\gamma \in L(\alpha_1,\ldots,\alpha_i)} f(x \oplus \gamma)$$

If a_i is linearly dependent of $a_1, \ldots, a_{i-1}$, then

$$\Delta^{(i)}_{a_1,\ldots,a_i} f(x) = 0$$

We use also the following proposition in this paper.

Proposition 5.2.2 (Lai [56]) *Let $ord(f)$ denote the nonlinear order*[1] *of a multi-variable polynomial function $f(x)$. Then*

$$ord(\Delta_a f(x)) \leq ord(f(x)) - 1$$

This leads to the following proposition.

Proposition 5.2.3 *If $\Delta_{a_1,\ldots,a_i} f(x)$ is not a constant, then the nonlinear order of f is greater than i.*

Proof: From Proposition 5.2.2 it follows that

$$ord(f) \geq ord(\Delta_{a_1} f(x)) + 1 \geq \ldots\ldots\ldots\ldots\ldots \geq ord(\delta_{a_1,\ldots,a_i} f(x)) + i$$

□

5.2.5 Attacks using higher order differentials

We consider in the following DES-like iterated block ciphers with block size of $\log_2 p^2$, where p is a prime. The plaintext block is divided into two halves L and R each of a size $\log_2 p$. Each round takes a text input of size $\log_2 p^2$ and a round key of size $\log_2 p$. We assume that there is no expansion of the text input to the F-function. One also calls the function F, the round function. In this section we adopt this convention for convenience, since it should cause no confusion. In the attacks we are going to present the complexity is measured as the number of encryptions of the analysed cipher, that the attacker has to perform for success.

Theorem 5.2.3 *Let $f(x,k) = (x+k)^2 \bmod p$, p prime, be the round function in a DES-like iterated cipher of block size $\log_2 p^2$, where '+' is addition module p. Then every non-trivial one round differential of f has a probability of $1/p$. Secondly, the second order derivative of f is a constant.*

Proof: Since a differential in general is independent of the key we will write $f(x)$ instead of $f(x,k)$ in the following. To prove the first statement, consider

[1] In [56] called the nonlinear degree.

a fixed $a \neq 0 \bmod p$. Then

$$\begin{aligned} f(x) - f(x+a) &=_p f(y) - f(y+a) \Leftrightarrow \\ x^2 - (x^2 + a^2 + 2ax) &=_p y^2 - (y^2 + a^2 + 2ay) \Leftrightarrow \\ -a^2 - 2ax &=_p -a^2 - 2ay \Leftrightarrow \\ 2a(x-y) &=_p 0 \Leftrightarrow \\ x &=_p y \end{aligned}$$

since p is prime. To prove the second statement, let a_1, a_2 be constants, then

$$\begin{aligned} &\Delta_{a_1,a_2} f(x) \\ &= \quad f(x + a_1 + a_2) - f(x + a_1) - f(x + a_2) + f(x) \\ &= \quad a_2^2 + 2a_2(x + a_1) - (a_2^2 + 2a_2 x) \\ &= \quad 2a_1 a_2 \end{aligned}$$

□

Theorem 5.2.4 *Let $f(x,k) = (x+k)^2 \bmod p$, p prime, be the round function in a 5 round DES-like iterated cipher of block size $\log_2 p^2$ with independent round keys, i.e., a key size of $5\times \log_2 p$. Then a differential attack using first order diflerentials needs about $2p$ chosen plaintexts and has a running time of about p^3.*

Proof: When doing a differential attack counting on the round key in the fifth round of the above cipher we need a 3 (or 4) round differential. It is easy to see that every 3 round differential has a probability of at most p^{-1} and we obtain

$$S/N = \frac{p \times \frac{1}{p}}{1 \times 1} = 1$$

where S/N is the signal to noise ratio defined on page 58 and $\lambda = 1$, since we use all pairs in the analysis and $\gamma = 1$, since in average one key value will be suggested by a pair. This attack is not possible, since the right key cannot be distinguished from other random keys. When doing a differential attack counting on the round keys in both the fourth and fifth rounds we need only a 2 round differential. And since the concepts of characteristics and differentials coincide for 2 rounds in a DES-like cipher, the probability

of a 2 round differential is at least $1/p$ for the above cipher. In this case we obtain

$$S/N = \frac{p^2 \times 1/p}{1 \times 1} = p$$

This attack is possible. We need about $2p$ chosen plaintexts and for every pair of plaintexts we do two rounds of encryption for every p^2 possible keys of the fourth and fifth rounds. Therefore we obtain a complexity of about p^3. □

Theorem 5.2.5 *Let $f(x,k) = (x+k)^2 \bmod p$, p prime, be the round function in a 5 round DES-like iterated cipher of block size $\log_2 p^2$ with independent round keys, i.e., a key size of $5\times \log_2 p$. Then a differential attack using second order differentials needa about 8 chosen plaintext with a running time of about p^2.*

Proof: In the following addition is modulo p. Consider $\Delta_{\alpha,\beta} f(x)$ where $\alpha = a \parallel 0$ and $\beta = b \parallel 0$ for some fixed a, b, i.e the right halves of α and β are zero. See also Figure 5.6, where $(0,0)$ denotes the trivial second order derivative of f and where in the second round the second order derivative is $(a, b, 2 \times a \times b)$. Consider the following attack

1. Choose plaintext P_1 at random.
2. Set $P_2 = P_1 + \alpha, P_3 = P_1 + \beta$ and $P_4 = P_1 + \alpha + \beta$.
3. Get the encryptions $C_1, \ldots, C_4$ of $P_1, \ldots, P_4$
4. For every value k_5 of the round key RK_5 do
   (a) Decrypt all ciphertexts $C_1, \ldots, C_4$ one round using k_5. Denote these 4 ciphertexts $D_1, \ldots, D_4$.
   (b) For every value k_4 of the round key RK_4 do
      i. Calculate $t_i = f(D_i^R + k_4)$ for $i = 1, \ldots, 4$.
      ii. If $(t_1 + t_4 - (t_2 + t_3)) - (D_1^L + D_4^L - (D_2^L + D_3^L)) = 2 \times a \times b$ then output k_5 and k_4.

Here X^L and X^R denote the left and right halves of X respectively. In the first round all inputs to the f-function are equal. In the second round

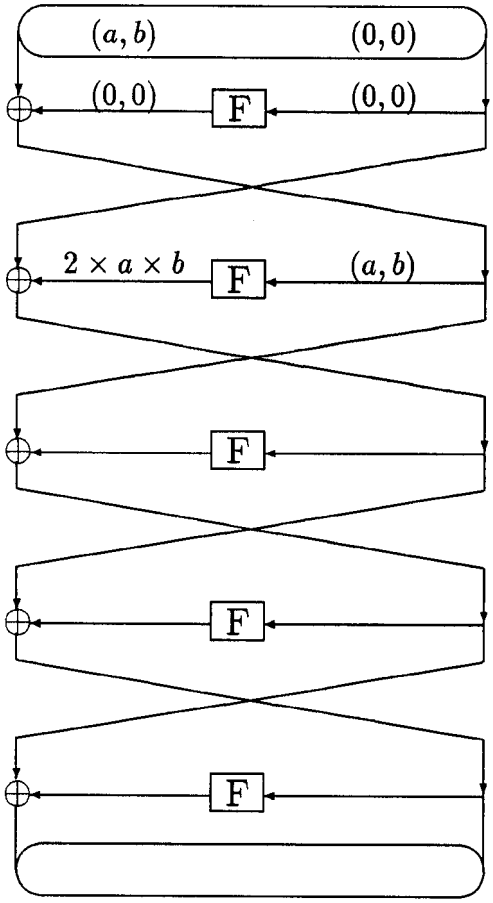


Figure 5.6: A second order differential of a five round DES-like iterated cipher.

the inputs form a second order differential with $(a, b, 2 \times a \times b)$. Since this differential has probability one according to Theorem 5.2.3, the difference in the four inputs to the third round is $\Gamma = 2 \times a \times b$. Therefore the difference in the outputs of the fourth round can be computed as the exclusive-or sum of Γ and of the left halves of the ciphertexts after four rounds. Upon termination a few keys will have been suggested, among which the right keys appear, since the two round second order differential has probability one. Therefore by repeating this attack a few times only one value of (RK_4, RK_5) is suggested every time. This value is guaranteed to be the secret fourth and fifth round

keys. The signal to noise ratio of the attack is

$$S/N = \frac{p^2 \times 1}{1 \times 1} = p^2$$

where we have assumed that one key in average is suggested by each pair of plaintexts. Now it is trivial to find the remaining three round keys by similar attacks on cryptosystems with less than five rounds. As in [7] we can pack the chosen plaintexts in economical structures, thus as an example obtain four second order differentials from 8 chosen plaintexts. □

If the prime p above is of size, say about 2^{25}, according to Theorem 5.2.4 a differential attack using first order differential has a complexity of about 2^{75} using about 2^{26} chosen plaintexts, i.e., not at all a practical attack. According to Theorem 5.2.5 a differential attack using second order differentials has a complexity of about 2^{50} using only about 8 chosen plaintexts, a practical attack or at least not far from being one.

The attack in the proof of Theorem 5.2.5 can be applied to any 5 round DES-like iterated cipher, where the round function contains no expansion and where the output coordinates are quadratic, i.e., the nonlinear order of f is 2. Furthermore the attack can be converted into an attack on any 5 round DES-like iterated cipher, as we will show now. For convenience let us consider functions over $GF(2)$. We state explicitly the definition of higher order differentials for this important case.

Definition 5.2.7 *Consider a Feistel cipher. A one round differential of order i is an $(i+1)$-tuple $(\alpha_1, \ldots, \alpha_i, \beta)$, s.t. all α_j's are linearly independent and*

$$\oplus_{\gamma \in L(\alpha_1, \ldots, \alpha_i)} g(P \oplus \gamma) = \beta$$

where g is the round function.

It is seen there are 2^i plaintexts in a differential of order i.

Theorem 5.2.6 *Let $f(x, k)$ be the round function in a 5 round DES-like iterated cipher of block size $2n$ with independent round keys, i.e., a key size of $5 \times n$ bits. Assume that the nonlinear order of f is r. Then a differential attack using differentials of order r needs about 2^{r+1} chosen plaintexts with*

a running time of about 2^n.

Proof: According to Proposition 5.2.3 the r'th-order derivative of a function of nonlinear order r is a constant. Therefore we can obtain a 2 round r'th-order differential with probability one and do a similar attack as in the proof of Theorem 5.2.5. □

To illustrate that the above attack works, we consider now the mappings $f(x) = x^{2^k+1}$ in $GF(2^n)$ described in [85]. According to Theorem 7.3.3 every 3 round differential has a probability of at most 2^{3-2n}, when n is odd and $gcd(k, n) = 1$.

Lemma 5.2.1 *Consider* $f(x) = x^{2^k+1}$ *in* $GF(2^n)$ *for* n *odd and* $gcd(k, n) = 1$. *Then every non-trivial one round differential of* f *has a probability of at most* $\frac{2}{2^n} = 2^{1-n}$ *and the second order derivative of* f, $\Delta_{\alpha,\beta} f(x)$ *is a constant with the value* $\Gamma = \alpha \times \beta \times (\alpha^{2^k-1} \oplus \beta^{2^k-1})$.

Proof: The first statement is proved in Theorem 7.3.4 and that the second derivative is a constant follows from Proposition 5.2.2. The constant is computed as follows.

$$\begin{aligned}
\Delta_{\alpha,\beta} f(x) &= f(x \oplus \alpha \oplus \beta) \oplus f(x \oplus \alpha) \oplus f(x \oplus \beta) \oplus f(x) \\
&= (x \oplus \alpha \oplus \beta)^{2^k+1} \oplus (x \oplus \alpha)^{2^k+1} \oplus (x \oplus \beta)^{2^k+1} \oplus (x)^{2^k+1} \\
&= (\alpha \oplus \beta)^{2^k+1} \oplus \alpha^{2^k+1} \oplus \beta^{2^k+1} \\
&= \alpha \times \beta \times (\alpha^{2^k-1} \oplus \beta^{2^k-1})
\end{aligned}$$

where we note that $(x \oplus \alpha)^{2^k+1} = (\alpha \times x^{2^k}) \oplus (x \times \alpha^{2^k}) \oplus x^{2^k+1} \oplus \alpha^{2^k+1}$ □

We implemented the attack of Theorem 5.2.5 counting on both the fourth and fifth round key using second order differentials in a five round DES-like iterated cipher with $f(x)$ of Lemma 5.2.1 as round function and with $n = 9$ and $k = 1$, i.e., a 18-bit cipher with a 45 bit key. In 100 tests using 12 chosen plaintexts only one pair of keys was suggested and every time this pair was the right values of the fourth and fifth secret round keys. We could have used quartets as defined in [7], thereby reducing the number of chosen plaintexts to about 8.

Note, that for this cipher the probability of any 3 round differential is at most 2^{3-2n} [85] where $2n$ is the block size. Therefore in a differential

attack using first order differentials counting only on the round key in the fifth round, the last round, would yield a signal to noise ratio of

$$S/N = \frac{2^n \times 2^{3-2n}}{1 \times 1} = 2^{3-n}$$

and would not be possible for $n > 3$. A differential attack counting on the round keys in both the fourth and fifth rounds using, what we will call, *partial differentials* is possible as is demonstrated in the next section.

Conclusion of the attacks. We showed there exist ciphers secure against a differential attack using first order differentials, but which can be broken using second order differentials. We used quadratic functions as round functions and second order differential attacks. We exploit the fact that for quadratic functions the second derivative is a constant. The attack can also be applied to ciphers using higher order functions as round functions. In general, a cipher with five rounds (or less) using round functions of nonlinear order r can be attacked using r'th-order differentials. However, attacks on a cipher with round functions of nonlinear order r involve encryptions of 2^r chosen plaintexts and the practicality of the attack decreases as r increases. Our attacks are limited to ciphers with 5 rounds or less and cannot be extended to 6 or more rounds. In the following we will show that even in the case where the round functions are of high order, differential attacks can be mounted.

5.2.6 Partial differentials

The attacks we are about to demonstrate use so-called partial differentials of first order and can be used in attacks on any four or five round cipher, where for at least one non-trivial difference in two inputs to the f-function not all differences in the outputs are possible. These attacks are therefore **not** applicable to the cipher example of Theorem 5.2.5.

In [83] it is shown that the functions $f(x) = x^{-1}$ in $GF(2^n)$, where $f(x) = 0$ for $x = 0$, are differentially 2-uniform for odd n and differentially 4-uniform for even n. In both cases the nonlinear order of the outputs is $n-1$. As an example consider a 5 round cipher using as round function the inverse function above for n odd. This cipher is highly resistant against differential

attacks using full differentials, since any 3-round differential has a probability of at most 2^{3-2n} according to Theorem 7.3.3. That is using differentials, where full n-bit differences are used. Note that in a $2n$-bit DES-like iterated cipher differentials are constructed from the concatenation of two n-bit values in each round. However, for every non-trivial input difference to one round there are only 2^{n-1} possible differences in the outputs, each one with a probability of $2/2^n$, since the round function is differentially 2-uniform. That is, for a non-trivial input difference we get one bit of information about the output differences. From this fact we can construct a 2 round differential of probability one, where only one bit of the differences after 2 rounds of encryption is predicted. We call that a **partial differential**. The following result holds.

Theorem 5.2.7 *Let $f(x)$ in $GF(2^n)$ be the round finction in a 5 round DES-like iterated cipher with block size $2n$ bits using 5 round keys, each of size n bits. Let α be an input difference for which only a fraction W of all output differences are possible. Then a differential attack using partial differentials has a complexity of 2^{2n} using about $2L$ plaintexts, where L is the smallest integer s.t. $(W)^L < 2^{-2n}$.*

Proof: Consider the following attack.

1. Let α be the non-trivial difference of two inputs to f, for which only a fraction W of the output differences can occur.
2. Compute a table T (initialised to zero in all entries), s.t. for $i = 0, \ldots, 2^n - 1$
   $T[f(i) \oplus f(i \oplus \alpha)] = 1$.
3. Choose plaintext P_1 at random and set $P_2 = P_1 \oplus (\alpha \parallel 0)$.
4. Get the encryptions C_1 and C_2 of P_1 and P_2
5. For every value k_5 of the round key RK_5 do
   (a) Decrypt the ciphertexts C_1, C_2 one round using k_5. Denote these ciphertexts D_1, D_2.
   (b) For every value k_4 of the round key RK_4 do
      i. Calculate $t_i = f(D_i^R \oplus k_4)$ for $i = 1, 2$.
      ii. If $T[t_1 \oplus t_2 \oplus D_1^L \oplus D_2^L] > 0$ then output k_5 and k_4.

Since the nonlinear order of $f(x)$ can be as high as $n-1$, the one bit information about the output differences we get from a given input difference is not necessarily easily determined. Therefore we first compute a table T, s.t. for a fixed input difference α, if $T[\beta] > 0$ then an output difference β is possible. The inputs to the first round are equal and the inputs to the second have difference α. That is, we can compute a fraction W of all possible values of the output difference of the fourth round from the right halves of the ciphertexts and from the values in table T. Upon termination about $W \times 2^{2n}$ of the possible values of (RK_4, RK_5) have been suggested, one of which is the right pair of keys. By repeating the attack sufficiently many times only one unique pair of keys, the right pair of keys, will be left suggested. Any other keys will be suggested with probability $W \times 2^{-2n}$ for each run of the above attack. Therefore after trying L pairs of plaintexts any key but the right key, is suggested L times with a probability of $(W)^L$ and if $(W)^L < 2^{-2n}$ with a high probability the right keys are uniquely determined. □

The attack can be extended to work on ciphers with any number of rounds by counting on all but the first three round keys. We implemented the attack on a 5 round 18-bit cipher with a key of 45 bits using as round function $f(x) = x^{-1}$ in $GF(2^9)$. In this case W is one half. Using 24 pairs of chosen plaintexts in 100 tests only one pair of keys was found, the right keys in the fourth and fifth rounds. The attack can be applied to a 5 round cipher with the cubing function of the previous section as the round function with the same probability of success.

Conclusion of the attacks. We showed an attack exploiting the fact that a cipher with five rounds (or less), where for one difference in the inputs to the round function only a fraction of the differences in the outputs are possible. The success of the attack depends on the size of this fraction in the way that the smaller the fraction the faster the attack, and on the size of the round keys. Imagine a five round 64-bit cipher constructed as above where the text input to the f-function is expanded to 48 bits, whereafter a 48-bit key is exclusive-or'ed. The above attack would then have a running time of at least about 2^{96} encryptions, which is hardly possible to do in the next few decades. As for the attacks using higher order differentials it is not possible to extend the above attacks to ciphers with 6 or more rounds.

5.2.7 Differential cryptanalysis in different modes of operation

The attacks by differential cryptanalysis are chosen plaintext attacks. However, the efficiency of a differential attack depends on the mode of operation, which is used for the attacked block cipher. The complexity of the attacks by Biham and Shamir [7] is the number of encryptions of chosen plaintexts, which the attacker needs in the attack, assuming that the block cipher is used in the native ECB mode. A chosen plaintext attack is not a realistic attack in many settings. lt is possible to Convert a differential attack on a block cipher used in ECB mode into a known plaintext attack, which is a more realistic attack. Assume that we need m pairs of plaintexts with a certain difference. By collecting about $2^{n/2} \times \sqrt{2m}$ known plaintexts, we can form, at least theoretically,

$$\frac{(2^{n/2} \times \sqrt{2m}) \times (2^{n/2} \times \sqrt{2m} - 1)}{2} \simeq 2^n \times m$$

pairs of plaintext pairs. If m and n are big, this forming of pairs may be computationally infeasible. Anyway, since there are exactly 2^n pairs of plaintexts with any certain difference, we can expect to get about m pairs with the needed difference [7]. This is not the whole story, though. If the plaintexts contain redundancy the differences of plaintext pairs are not necessarily uniformly distributed. As an example, take the exclusive-or as the difference operation. Then if the plaintexts consist of ASCII characters, every parity bit in a byte is zero, therefore in an exclusive-or of any two plaintexts all parity bits are zero.

For most of the differential attacks many pairs of plaintexts are needed [7]. It is not advisable to use the ECB mode when many plaintext blocks are to be encrypted, therefore an attacker can expect that the attacked block cipher is **not** used in the ECB mode. When a block cipher is used in the CBC mode, the attacker has no control over the inputs to the block cipher. Assume in the following that a differential attack needs m pairs of chosen plaintexts, when the block cipher is used in the ECB mode. Then there exists a differential attack on the same block cipher used in the CBC mode, that needs $2^{n/2} \times \sqrt{2m}$ known or chosen plaintexts by an argument similar as above. A complexity of m pairs can be obtained for the block cipher used in the CBC mode in an adaptively chosen plaintext attack, if the initial

value is not secret. The attacker chooses one random plaintext P and gets the encryption $C = E_K(P \oplus IV)$, where IV is an initial value. He then repeats the with plaintext P', s.t. $P \oplus P'$ have the desired difference and gets $C' = E_K(P' \oplus IV)$. Assuming that the initial value is unchanged the inputs to the block cipher for the two encryptions have the desired difference. If the initial value changes for every CBC encryption an attacker cannot do an adaptively chosen plaintext attack.

An adaptively chosen plaintext attack is the least practical of all attacks. It is important to note, that even though Biham and Shamir's results on the DES [7] are very impressive, the attacks are by far no practical attacks.

Assume again that a differential attack needs m pairs of chosen plaintexts, when the block cipher is used in the ECB mode. Then there exists a differential attack on the same block cipher used in the OFB mode with full feedback, that needs $2^{n/2} \times \sqrt{2m}$ known or chosen plaintexts by an argument similar as for the CBC mode. A similar argument holds for the CFB mode. But whereas the OFB mode should be used only with full feedback, as noted in Section 3.1, the CFB can be used with any feedback. If $n' < n$ bits are fed back, the attacker does not know the full output of the block cipher and the success of a differential attack decreases. Differential attacks on the DES used in the CFB mode have been considered in [97]. A modified differential attack is presented, which works for $m \geq 3$, where m is the size of plaintext and ciphertext blocks and where m bits are used in the feedback. The attacks on the DES are faster than exhaustive search only for a restricted number of rounds, i.e., up to 10 rounds. The work is motivated by the fact that for $m < n$, encryption in the CFB mode is slow and for small m it may be tempting to reduce the number of rounds in the DES to achieve better performance.

5.3 Linear Cryptanalysis

In 1993 M. Matsui introduced linear cryptanalysis of the DES [64]. A similar attack on FEAL appeared already in 1992 [68]. Linear cryptanalysis [64] is a known plaintext attack in which the attacker exploits linear approximations of some bits of the plaintext, ciphertext and key. In the attack on the DES (or on DES-like iterated ciphers) the linear approximations are obtained by combining approximations for each round under the assumption of indepen-

dent round keys. The attacker hopes in this way to find an expression (5.6), which holds with probability $p_L \neq \frac{1}{2}$ over all keys [64] T such that $|p_L - \frac{1}{2}|$ is maximal.

$$(P \cdot \alpha) \oplus (C \cdot \beta) = (K \cdot \gamma) \tag{5.6}$$

where P, C, α, β, γ are m-bit strings and where '$\cdot$' denotes the dot product.

Since an expression (5.6) in the ideal case will have a probability one half, and since it contains only linear expressions, we call the expression (5.6) a *linear approximation*. Given an approximation (5.6) a linear attack using N plaintexts and the N corresponding ciphertexts goes as follows.

Linear attack [64]

1. For all plaintexts, P, and ciphertexts, C, let T be the number of times the lefthand side of (5.6) is zero.

2. If $T > N/2$ guess that $K \cdot \gamma = 0$, otherwise guess that $K \cdot \gamma = 1$.

The attack finds one bit of information about the key, $K \cdot \gamma$, and the complexity of a successful attack, i.e., the number of known plaintexts needed, using the above algorithm can be approximated in the following way. Let $\mathbf{T}$ be a binomial random variable taking on the value 0 with probability p. Assume that $|p - 1/2|$ is small and w.l.o.g. that $p > 1/2$. Then

$$\begin{aligned}
\Pr(T > N/2) &= 1 - \Pr(T \leq N/2) \\
&\simeq 1 - \Phi(\frac{N/2 + 1/2 - Np}{\sqrt{p(1-p)} \times \sqrt{N}}) \\
&\simeq 1 - \Phi(-2\sqrt{N}|p - 1/2|) \\
&= \Phi(2\sqrt{N}|p - 1/2|)
\end{aligned}$$

where Φ is the normal distribution function. With $N = |p - 1/2|^{-2}$ the success rate is about 97.72%. Since the number of plaintexts needed is the dominating factor in a linear attack, the complexity, N_P, of the above linear attack is [64]

$$N_P \simeq |p_L - 1/2|^{-2}$$

where p_L is the probability of a linear approximation of the form (5.6). This estimate shows that the quantity of interest in a linear attack is $|p_L - 1/2|^{-2}$. For DES-like iterated ciphers linear approximations of the form (5.6) can be found by combining linear approximations of each round in the cipher. As in differential cryptanalysis we can define characteristics to be used in linear cryptanalysis.

Definition 5.3.1 *A one-round linear characteristic is a list of input, key and output bits of one round of the block cipher and a probability* p *over all keys and plaintexts, s.t. the boolean value obtained by adding (modulo 2) the input and key bita equals the boolean value obtained by adding (modulo 2) the output bits with probability* p. *An* r*-round* linear characteristic *is the concatenation of* r *one-round linear characteristics.*

In some rounds of a linear characteristic linear approximations are not needed. We call these rounds **trivial** one-round linear characteristics.

As in differential cryptanalysis by assuming that the r one round approximations are independent we can calculate the probability of an r-round linear approximation from the probabilities of the r one round approximations, for example by assuming that the round keys in the cipher are independent. The probability of an r-round linear characteristic is calculated using the **Piling Up-Lemma** [64].

Lemma 5.3.1 *Let* $Z_i, 1 \leq 2 \leq n$, *be independent random variables, whose boolean values are* 0 *with probability* p_i. *Then*

$$\Pr(Z_1 \oplus Z_2 \oplus \cdots \oplus Z_n = 0) = 1/2 + 2^{n-1} \prod_{i=1}^{n} (p_i - 1/2) \tag{5.7}$$

The above linear attack is not very efficient, since it finds only one bit of information about the key. However, there exists an extended linear attack, which finds more key bits. Instead of approximating the first and last round in an r-round iterated cipher, since we know both the plaintext and the ciphertext, we can count on all keys which affects the bits in the linear approximation (5.6) in the first and last round, yielding the following approximation

$$(P \cdot \alpha) \oplus (C \cdot \beta) \oplus (F(P_R, K_1) \cdot \alpha_1) \oplus (F(C_R, K_r) \cdot \alpha_r) = (K \cdot \gamma) \tag{5.8}$$

where P_R, C_R are the right halves of the plain- and ciphertexts respectively. K_1 and K_r are the key bits affecting the linear approximation in the first and r'th rounds. For all choices of the keys K_1 and K_r the approximation (5.8) can be seen as an approximation of a cipher of r-2 rounds, i.e., two rounds shorter than the original cipher. The attack goes as follows with N available plaintexts.

Extended linear attack [64]

1. For all, say n, values of the two keys, K_1 and K_r do:

   For all plaintexts, P, and ciphertexts, C, let T_i, $i = 1, \ldots, n$, be the number of times the lefthand side of (5.6) is zero.

2. Let T_{max} and T_{min} be the maximum and minimum values of the T_i's for $i = 1, \ldots, n$. If $|T_{max} - N/2| > |T_{min} - N/2|$ guess that K_1 and K_r are the key values from the computation of T_{max}.

   If $|T_{max} - N/2| < |T_{min} - N/2|$ guess that K_1 and K_r are the key values from the computation of T_{min}.

In case of the DES it is conjectured and confirmed by computer experiments [64, 65, 66] that the efficiency of (5.8) decreases, when the values of K_1 or K_r are incorrect values. The complexity of success of this extended attack is somewhat larger than the complexity using the first attack. In [64, 65, 66] it is estimated that the complexity of an extended linear attack on the DES with up to 16 rounds is about

$$N_P \simeq c \times |p_L - 1/2|^{-2}$$

where $c \leq 8$ [65, 66]. Note that the success of the extended attack is independint of the parity of the key bits from the intermediate rounds, $K \cdot \gamma$. And opposite to Matsui's attack we will not use the right side of (5.8) in the attack. The reason for this follows in the coming section. Note that the practicality of this extended attack depends also on how many key bits are needed to count on in the first and last rounds. In his attack on the DES, because only one S-box is active in every round of the linear approximation, Matsui counts on and finds 12 bits of the key. By using other linear approximations other bits of the key can be found.

5.3.1 The probabilities of linear characteristics

Let $X \in GF(2)^m$ and $K \in GF(2)^\ell$ be random variables and $Y = Y(X, K)$, $Y \in GF(2)^n$, be a random variable which is a function of X and K. Then we have the following generalisation of Parseval's Theorem, see Theorem 7.5.1, an important result found recently by Kaisa Nyberg [84].

Theorem 5.3.1 (The Fundamental Theorem) *If X and K are independent and K is uniformly distributed, then for all $a \in GF(2)^m$, $b \in GF(2)^n$ and $\gamma \in GF(2)^\ell$*

$$2^{-\ell} \sum_{k \in GF(2)^\ell} |P_X(X \cdot a + Y(X,k) \cdot b = 0) - 1/2|^2 =$$
$$2^{-\ell} \sum_{k \in GF(2)^\ell} |P_X(X \cdot a + Y(X,k) \cdot b + k \cdot \gamma = 0) - 1/2|^2 =$$
$$\sum_{c \in GF(2)^\ell} |P_{X,K}(X \cdot a + Y(X,K) \cdot b + K \cdot c = 0) - 1/2|^2 =$$

For DES-like ciphers this can interpreted in the following manner [84].

Theorem 5.3.2 *If the round keys of r rounds of a DES-like cipher are independent and unifomaly random then $\ell = mr$ and for all a and b*

$$\sum_{c \in GF(2)^\ell} |P_{X,K}(X_0 \cdot a + Y(X_0,K) \cdot b + K \cdot c = 0) - 1/2|^2 =$$
$$4^r \sum_{c \in GF(2)^\ell} |P_X(X_0 \cdot (a + b^0) = 0) - 1/2|^2 \times$$
$$\prod_{i=1}^{r} |P_Z(f(Z) \cdot b_R^i = Z \cdot c_i) - 1/2|^2 \tag{5.9}$$

where X_0 is the plaintext, $Y = X_r$ the corresponding ciphertext and $Z = E((X^R) + K)$. Furthermore,

$$b^r = (b_L, b_R),\ b^{i-1} = (b_R^i, b_L^i + E^t(c_i)),\ \textit{for } i = 1, 2, \ldots, r,\ \textit{and}$$
$$c = (c_1, \ldots, c_r)$$

and E^t is the transpose of the expansion E.

Corollary 5.3.1 ([84]) *If the plaintexts are uniformly distributed*

$$\sum_{c \in GF(2)^\ell} |P_{X,K}(X \cdot a + Y(X;K) \cdot b + K \cdot c = 0) - 1/2|^2 =$$

$$4^{r-1} \sum_{ck \in GF(2)^\ell} \prod_{i=1}^{r} |P_Z(f(Z) \cdot b_R^i = Z \cdot c_i) - 1/2|^2$$

where $a_L + b_L + \sum_{i=1}^{r/2} E^t(c_{2i}) = 0$ *and* $a_R + b_R + \sum_{i=1}^{(r-1)/2} E^t(c_{2i-1}) = 0$, *assuming that* r *is even.*

These theorems say that the probability of an approximation (5.6) does not depend on the value of γ. Moreover for the probability p of a linear approximation it holds that $|p - 1/2|^2$ is the sum of $|p_\gamma - 1/2|^2$ for all values of γ.

For the first linear attack this may have the effect that the probability of success decreases. As also noted by Biham [4], if there exists more than one expression of (5.6) for different values of γ, they may cancel the effect of each other.

It is seen that the above way of calculating the probabilities of a linear approximation is reminiscent of the way of calculating the probabilities of differentials in differential cryptanalysis, see 5.5 on page 67, which at the same time indicates that in practice for longer characteristics/approximations it is hard to calculate the exact probability. In Section 6.1.5 we show the effect of the above results on Matsui's attack on the DES [64, 65, 66].

5.3.2 Iterative linear characteristics for DES-like ciphers

As noted by Matsui [64, 65, 66] we can obtain iterative linear approximations for DES-like ciphers, if approximations exist where only bits of the right halves of (intermediate) ciphertexts are known. As in differential cryptanalysis our goal is to maximise the number of trivial one round characteristics. For these rounds in linear cryptanalysis no linear approximations are needed. In the following let X_i be the right half of the ciphertext after i rounds of encryption, i.e., X_i is the input to the F-function in the $(i+1)$'th round. X_0 denotes the right half of the plaintext input to the linear characteristic.

Also for every round let us fix a key k and for convenience let $F(X)$ denote $F(X, k)$.

2-round iterative characteristics

In this type of characteristic every second round contains no linear approximation, see Figure 5.7. This type of characteristic is not possible if the coordinate functions of the F-function are all balanced, e.g. if F is a permutation. We assume that we have knowledge about the bits $X_0 \cdot \alpha$ and that $F(X_1) \cdot \alpha = 0$ with probability $p \neq 1/2$. Then

$$|\Pr_X(X_2 \cdot \alpha = X_0 \cdot \alpha) - 1/2|^2 = |p - 1/2|^2$$

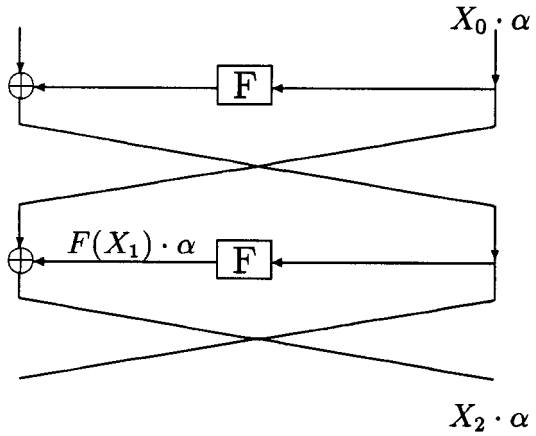


Figure 5.7: A 2 round iterative linear characteristic.

3-round iterative characteristks

In this type of characteristic every third round contains no linear approximation, see Figure 5.8. It follows that $X_3 \cdot \beta = (F(X_2) \cdot \beta) \oplus (X_1 \cdot \beta)$ and that $X_0 \cdot \alpha = (F(X_1) \cdot \alpha) \oplus (X_2 \cdot \alpha)$. Assuming that $F(X_1) \cdot \alpha = X_i \cdot \beta$ with probability p_1 and that $F(X_2) \cdot \beta = X_2 \cdot \alpha$ with probability p_2, it follows that for the 3 round characteristic

$$|\Pr_X(X_3 \cdot \beta = X_0 \cdot \alpha) - 1/2|^2 = 4 \times |p_1 - 1/2|^2 \times |p_2 - 1/2|^2$$

When concatenating two three round iterative characteristic, the rounds two and three in the second characteristic are interchanged.

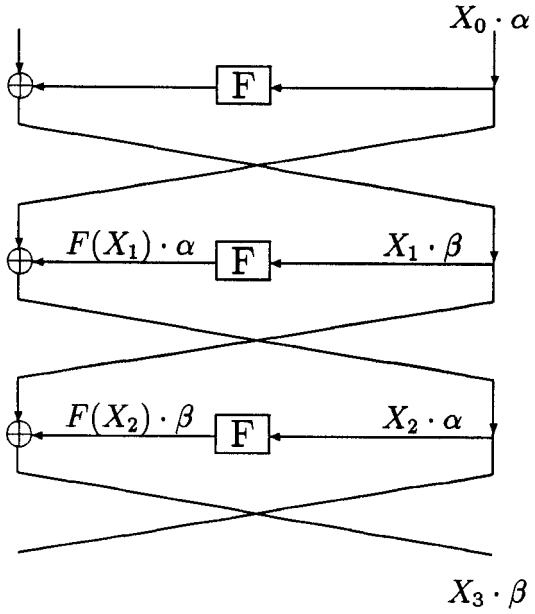


Figure 5.8: A 3 round iterative linear characteristic.

4-round iterative characteristics

In this type of characteristic every fourth round contains no linear approximation, see Figure 5.9. Let us fix a value of A from Figure 5.9 and let the probabilities of the 3 last rounds be $p_1(A), p_2(A), p_3(A)$ respectively. It follows that $X_4 \cdot \beta = (F(X_3) \cdot \beta) \oplus (X_2 \cdot \beta)$ and that $X_0 \cdot \alpha = (F(X_1) \cdot \alpha) \oplus (X_2 \cdot \alpha)$.

$$\begin{aligned} X_2 \cdot (\beta \oplus \alpha) &= (F(X_3) \cdot \beta) \oplus (F(X_1) \cdot \alpha) \oplus (X_4 \cdot \beta) \oplus (X_0 \cdot \alpha) \Rightarrow \\ X_2 \cdot (\beta \oplus \alpha) &= (x_3 \cdot A) \oplus (X_1 \cdot A) \oplus (X_4 \cdot \beta) \oplus (X_0 \cdot \alpha) \\ &= (F(X_2) \cdot A) \oplus (X_4 \cdot \beta) \oplus (X_0 \cdot \alpha) \end{aligned}$$

with probability $1/2 + 2(p_1(A) - 1/2)(p_3(A) - 1/2)$. And since $X_2 \cdot (\beta \oplus \alpha) = F(X_2) \cdot A$ with probability $p_2(A)$ it follows from Theorem 5.3.2 that

$$\begin{aligned} &|\Pr_X(X_4 \cdot \beta = X_0 \cdot \alpha) - l/2|^2 \\ &= 4^2 \times \sum_A |p_1(A) - 1/2|^2 \times |p_2(A) - 1/2|^2 \times |p_3(A) - 1/2|^2 \end{aligned}$$

When concatenating two four round iterative characteristic, the rounds two and four in the second characteristic are interchanged. This is the type of characteristic Matsui uses in his attack on the full 16-round DES. As we will see, the probability of Matsui's approximation is (somewhat) better than his estimate in [64, 65, 66].

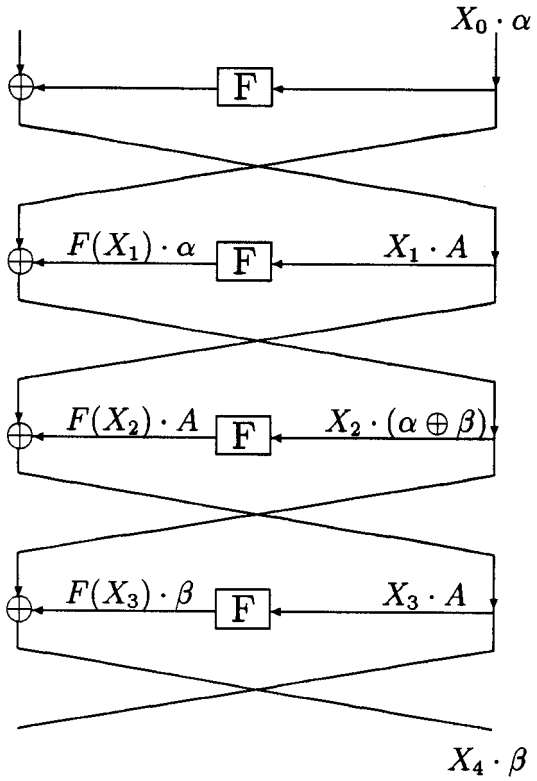


Figure 5.9: A 4-round iterative linear characteristic.

Longer characteristics

As for differential cryptanalysis it is possible, and quite trivial, to go further in the above process to n-round iterative linear characteristics. For a 5-round iterative characteristic there will be two 'free' variables (like A for 4-round characteristics) in the equations to solve. In practice, to calculate the exact probability gets much more complex for larger values of n. Note here the

resemblance with characteristics and differentials in differential cryptanalysis. In [67] Matsui devised a method to find the best linear characteristics in a Feistel cipher, but although that in itself is impressive, note that his characteristics are for one value of the variables like A of Figure 5.9.

5.4 Analysis of the Key Schedules

In this section we consider the key schedules of block ciphers. Much research on the DES has been focused on the S-boxes, but a weak key schedule can be exploited in cryptanalytic attacks.

5.4.1 Weak and pairs of semi-weak keys

We consider an n-bit block cipher, where $E_K(\cdot)$ denotes encryption with the key K and $D_K(\cdot)$ denotes decryption.

Definition 5.4.1 *A weak key K, is a key for which encryption equals decryption, i.e., $E_K(X) = D_K(X)$ for all n-bit texts X.*

Definition 5.4.2 *A pair of semi-weak keys K, K^*, are keys for which encryption with one keys equals decryption with the other key, i.e., $E_K(X) = D_{K^*}(X)$ for all n-bit texts X or equivalently, $D_K(X) = E_{K^*}(X)$ for all n-bit texts X.*

A well-known example is

Example 5.4.1 *There are at least four weak keys and six pairs of semi-weak keys for the DES.*

In [16] D. Coppersmith showed that there are exactly 2^{32} fixpoints for the DES used with a weak key.

Theorem 5.4.1 *Consider an r-round DES like iterated cipher for r even. A key K, for which all r round keys $K_i, i = 1, \ldots, r$ are equal, is a weak key. Furthermore, there are exactly $2^{n/2}$ fixpoint for the cipher used with a weak key.*

Proof: Since the only difference between the encryption and decryption function of a DES-like iterated cipher is the values of the round keys, the first part of the proof follows. To prove the second part, let the value of the input to the $(r/2)$'th and $(r/2+1)$'th rounds be equal, say a. Because the round keys are equal the outputs of the rounds are also equal and therefore the inputs to the $(r/2-1)$'th and $(r/2+2)$'th rounds are equal. It follows by induction that the inputs to the i'th and the $(r-i+1)$'th rounds are equal for all i and that the plaintext equals the ciphertext. Since there are 2^{32} possible values of a there are at least as many fixpoints. Now let P be a fixpoint for a weak key. The inputs to the F-function in the first and last rounds are equal, therefore the outputs are equal. But then the inputs to the F-function in the second and second last rounds are equal etc. Finally the inputs to the $(r/2)$'th and $(r/2+1)$'th rounds axe equal, proving that there are exactly 2^{32} fixpoints for a weak key. □

If a weak key is known, the $2^{n/2}$ fixpoints can be computed using only half an encryption. If the number of weak and pairs of semi-weak keys are small they are of no importance for the security of a block cipher used for encryption in practice, if the keys are chosen at random, i.e., under Assumption 4.1.1. However, when block ciphers are used in hash modes where e.g. the key input can be chosen by the attacker in attempts to find collisions, they play an important role as demonstrated in Section 8.2.4.

5.4.2 Simple relations

First we define

Definition 5.4.3 *Let E be a block cipher, s.t. $E_K(\cdot)$ denotes the encryption finction using the key K and let f, g_1, g_2 be 'simple' functions, such that the total complexity of one evaluation of each of f, g_1, g_2 is smaller than one evaluation of E (one encryption). Then if*

$$E_K(P) = C \Rightarrow E_{f(K)}(g_1(P, K)) = g_2(C, K) \tag{5.10}$$

E is said to contain a **simple relation** *between the encryption functions $E_K(\cdot)$ and $E_{f(K)}(\cdot)$.*

This definition is different from that of *linear structures* given in [27]. Simple relations for which (5.10) holds for all plaintexts and all keys can be exploited in a chosen plaintext attack as follows

1. Denote by PK the set of all potential keys.
2. Choose a random plaintext P.
3. Get the encryption $C = E_K(P)$ where K is the secret key.
4. Choose a key $K' \in PK$
   (a) Calculate $C' = E_{K'}(P)$. If $C' = C$ output K' and stop
   (b) Get the encryption $C^* = E_K(g_1(P, K'))$.
   If $g_2(C', K') = C^*$ output $f(K')$ and stop
5. Remove K' and $f(K')$ from PK and go to 4

Note that in step 4b we get $E_{f(K')}(g_1(P, K')) = g_2(C', K') = C^*$. That is, in general one can check two keys using one chosen plaintext and doing one encryption and one evaluation of f and the g_i's. The restriction to 'easy' evaluations of f and the g_i's is now obvious and the efficiency of this attack depends on the complexity of the evaluations of the simple functions. Also of great importance is the complexity of the enumeration of the keys. The operations in steps (1) and (5) have to be of low complexity. If the g_i's are independent of the keys a further improvement of the attack is possible as we will illustrate now.

For the DES and LOKI'91 there is a well-known simple relation known as the complementation property, where $f(K) = \overline{K}$ (the complemented value of K) and $g_i(X, K) = \overline{X}$. In this case we need only ask for the chosen plaintext once in step 4b of the above attacks.

5.4.3 Weak hash keys

We consider as before iterated block ciphers with block size m and for convenience we assume that the group operation is the exclusive-or.

Definition 5.4.4 *A* **weak hash key K** *is a key for which*

$$P \oplus E_K(P) = \delta \tag{5.11}$$

with probability $p >> 2^{-m}$ *for fixed* δ *over all plaintexts* P.

It is clear that weak hash keys should be avoided in hash modes where the input to the block cipher is added modulo 2 to the output to obtain some kind of one-wayness.

As stated earlier, for each weak key in DES, LOKI'89 and LOKI'91 there are 2^{32} fixpoints, therefore a weak key in DES-like iterated ciphers, e.g. the DES and the LOKI's, is also a weak hash key. In [80] Moore and Simmons generalised the idea of Coppersmith for the DES to the following.

Theorem 5.4.2 (DES) *Suppose for some key* K, *that* $\forall i : K(i) = K(17 - i) = E(\sigma)$ *where* E *is the 48 bit expansion of some 32 bit string* σ. *Then there are exactly* 2^{32} *plaintexts* P, *s.t.* $\mathrm{DES}_K(P) \oplus P = \sigma \parallel \sigma$.

Proof: Assume that for some key K, $K(i) = K(17 - i) = E(\sigma)$. Then choose the inputs to the two middle rounds $(r/2)$ and $(r/2 + 1)$, s.t. the difference is σ. Now the inputs after addition of the keys in the two middle rounds are equal and the difference between the inputs to the $(r/2 - 1)$'th and $(r/2 + 2)$'th rounds before addition of the keys will be σ, in general the difference between the inputs to the i'th and $((r + 1) - i)$'th round will be σ. Finally the difference between the plaintext and the ciphertext will be $(\sigma \parallel \sigma)$. To complete the proof we note that there are exactly 2^{32} ways to choose a pair of 32 bit strings with difference σ. □

Also, Moore and Simmons stated the following result [80]

Corollary 5.4.1 *For the DES there are only eight keys satisfying the condition in Theorem 5.4.2. Four are weak keys and the other four are semi-weak keys.*

The method can be extended to the case where the inputs after addition of the key in the i'th and $((r + 1) - i)$'th round are not equal, but where equal outputs of the rounds are obtained with some probability. We leave it as an open problem if there exist other keys for the DES than the above eight

for which (5.11) holds with a probability greater than 2^{-64}. We proceed and define

Definition 5.4.5 *A pair of* **semi-weak hash keys** *are keys for which*

$$E_{K_1}(P) \oplus \alpha = D_{K_2}(P \oplus \alpha) \tag{5.12}$$

with probability $p >> 2^{-m}$ *for fixed* α.

In this case we have a parallel to the semi-weak keys. With $\alpha = 0$ equation (5.12) is always true for semi-weak key pairs. Semi-weak hash keys may be found using differential cryptanalysis and can be used in cryptanalytic attacks. Assume we have a pair of semi-weak hash keys K_1 and K_2 for which equation (5.12) holds with probability p. With $E_{K_1}(P_1) = C_1, C_2 = P_1 \oplus \alpha$ and $P_2 = D_{K_2}(C_2)$ one obtains:

$$E_{K_1}(P_1) \oplus P_1 \oplus C_2 = D_{K_2}(C_2) \Rightarrow C_1 \oplus P_1 = P_2 \oplus C_2$$

with probability p. In that way, we would find a (free-start) collision for a hash mode, where the plaintext is added to the ciphertext.

Chapter 6

Analysis of Specific Block Ciphers

In this chapter we analyse specific block ciphers. In Section 6.1 we analyse the most well known block cipher, the Data Encryption Standard (DES) [90]. First we do differential cryptanalysis on the algorithm and show that the characteristics used by Biham and Shamir [7] are the best choices. Furthermore we show how to improve Biham and Shamir's attack on the full 16-round DES by using more characteristics. Next we analyse the key schedule and show several new pairs of weak keys for the DES. Interesting higher order and partial differentials for the DES are given and it is shown that partial differentials are useful in attacks on the DES with a small number of rounds. Finally we consider linear cryptanalysis of the DES and show several new linear characteristics. It is shown that there exists linear characteristics, which improve the probability of the best known linear approximations of the DES. Although this improvement is hardly measurable for the DES, it illustrates that the method of calculating the probability of linear approximations given in Section 5.3 is important. In Section 6.2 we analyse the LOKI ciphers, proposed by Brown, Pieprzyk and Seberry in 1990 [15] and by Brown, Kwan, Pieprzyk and Seberry in 1991 [14]. We concentrate our analysis on the latest proposal, LOKI'91 and do differential cryptanalysis, show a weakness in the F-function, and give a chosen plaintext attack reducing an exhaustive key search by a factor of four exploiting a weakness in the key schedule. Finally we show that there are several weak hash keys for LOKI'89 and a few for LOKI'91. In Section 6.3 we analyse the s^2-DES

S-boxes proposed by K. Kim in [43] and show there are several characteristics much better than the known characteristics for the DES, thus showing that a conjectured improvement of the DES S-boxes was not obtained. In Section 6.4 we analyse the s^3-DES S-boxes proposed by K. Kim in [44] as a consequence to our attacks on s^2-DES. We show that the estimates given by Kim for the best possible characteristics are too optimistic and give one characteristic with a probability 2^{30} times better than Kim's estimates, though still not enabling a successful differential attack. In Section 6.5 we analyse the $x\text{DES}^i$ block ciphers proposed by Zheng in [115]. We give attacks on both $x\text{DES}^1$ and $x\text{DES}^2$, which show that Zheng's constructions are not optimal.

6.1 DES

The Data Encryption Standard (DES) [90] is the most popular encryption system in use today. Around the world, governments, banks, and standard organisations have made the DES the basis of secure and authentic communication [108]. Since its publication in January 1977, a huge volume of research on the DES has been published; we refer to [13] and to the references in this thesis. This research probably makes the DES the most analysed cipher ever and to its credit, no serious weaknesses have been found in the algorithm. The best an attacker can do is to search exhaustively for a key. On the other hand this has become reasonably feasible as demonstrated by Wiener [112]. Although no practical shortcut attacks on the DES have been found, the analysis on the algorithm has been very fruitful and in some cases attacks has been generalised and applied to other ciphers with big success [7]. Already in 1976, before the official publication of the DES as a Federal Information Processing Standard in the U.S.A., a group of scientists at Stanford University, U.S.A.Y cryptanalyzed the algorithm and found the only real weakness known to date [35].

> **The complementation property** Let $C = \text{DES}_K(P)$, i.e., C is the encrypted value of plaintext P using key K. Then $\text{DES}_{\overline{K}}(\overline{P}) = \overline{C}$.

The complementation property can be used in a chosen plaintext attack using one known and one chosen plaintext to reduce an exhaustive search for the key by a factor 2. See the simple relation attack in Section 5.4.2. The full

description of the DES algorithm is given in Appendix C. The cryptographic strength in the DES lies in the substitution boxes (S-boxes) of which the following 5 properties are well known [12].

1. No S-box is a linear or affine function.
2. Changing one bit in the input to an S-box results in changing at least two output bits.
3. The S-boxes were chosen to minimise the difference between the number of 1's and 0's when any single bit is held constant.
4. $S(\mathbf{x})$ and $S(\mathbf{x} \oplus (001100))$ differ in at least two bits.
5. $S(\mathbf{x}) \neq S(\mathbf{x} \oplus (11ef00))$ for any e and f.

In Section 6.1.1 we give our analysis of the search for the best characteristics to be used in a differential attack on the DES. Also, we give a slightly improved differential attack on the DES evolved by a closer study of the key schedule. In Section 6.1.2 we give an analysis of the key schedule and show new pairs of weak keys. In Sections 6.1.3 and 6.1.4 we give new higher order differentials and partial differentials of the DES.

6.1.1 Iterative characteristics

In differential cryptanalysis of the DES the difference of two bit strings is defined as the bitwise exclusive-or of the strings. A DES S-box consists of 4 rows of 4-bit bijective functions. The input to an S-box is 6 bits. The left outermost bit and the right outermost bit (the row bits) determine through which function the four remaining bits (the column bits) are to be evaluated. This fact gives us a sixth property of the DES S-boxes important for differential cryptanalysis.

6. $S(\mathbf{x}) \neq S(\mathbf{x} \oplus (0abcd0))$ for any a, b, c and d, $abcd \neq 0000$.

The inner input bits for an S-box are input bits that do not affect the inputs of other S-boxes. There are two inner input bits for every S-box. Because of the P-permutation the following property is also important for differential cryptanalysis.

The inner input bits property. The inner input bits for an S-box, S_i, come from S-boxes, whose inner input bits cannot come from S_i.

Example: The inner input bits for $S1$ come from $S2$ and $S5$, whose inner input bits come from $S3$ and $S7$ respectively $S2$ and $S6$.

In the following sections we will refer to the skeletons of iterative characteristics from Section 5.2.2 without further notice.

2-round iterative characteristics

As shown in [67] the best characteristics for a differential attack on the 16-round DES is based on a 2-round iterative characteristic. The following theorem was already proven in [24]. We give the proof in a different manner.

Theorem 6.1.1 *If two inputs to the F-unction result in equal outputs, the inputs must differ in at least 3 neighbouring S-boxes.*

Proof: If the inputs differ only in the input to one S-box the expanded input xor must have the following form: $00ab00$ (binary), where $ab \neq 00$. Because of properties 2 and 4 above, these inputs cannot give equal outputs. This also tells us that the inputs must differ in neighbouring S-boxes. If the inputs differ in only two neighbouring S-boxes, S_i and $S(i+1)$, the two input xors must have the following forms: $Si : 00abcd$ and $S(i+1) : cdef00$. Now

$cd \neq 00$, because of properties 2 and 4.
$cd \neq 01$, because of property 6 for $S(i+1)$.
$cd \neq 10$, because of property 6 for $S(i)$.
$cd \neq 11$, because of property 5 for $S(i+1)$.

□

There are several 2-round iterative characteristics for DES, where the inputs differ in three neighbouring S-boxes. By consulting the *difference distribution table* for the 8 S-boxes it is easy to find the best possibilities. The two best of these are used in [[7] to break the full 16-round DES using 2^{47} chosen plaintexts. The probability of the two characteristics is $\frac{1}{234}$ for the two rounds.

3-round iterative characteristics

We proceed trying to find a 3-round iterative characteristic for the DES with a higher probability per round than the 2-round iterative characteristic. The highest probability for a non trivial input/output xor combination in the DES is $\frac{1}{4}$. Because $(\frac{1}{4})^x \geq (\frac{1}{234})^{1.5} \Rightarrow x < 6$, there can be different inputs to at most 5 S-boxes for the two nonzero rounds together. Because of the inner input bits property of the P-permutation in DES, Φ and Γ must differ in the inputs to at least two S-boxes each. Property 2 of the S-boxes implies that the inputs differ in at least one additional S-box, making Φ and Γ together differ in the inputs to at least 5 S-boxes. The full proof is given in the Appendix B.1, Theorem B.1.1. In a 3-round iterative characteristic the input/output xors, where the inputs together differ in the inputs to 5 S-boxes and yielding the highest probability are $\Phi = 31200000_x$ and $\Gamma = 00004200_x$. The probability for the 3-round iterative characteristic is $2^{-18.42}$. This probability is very low and there are in fact input xors, which together differ in the inputs to 6 S-boxes with a higher probability, $\Phi = 03140000_x$ and $\Gamma = 00004014_x$. The probability for the 3-round iterative characteristic is $2^{-18.1}$ Both characteristics have a probability too low to be used in a successful differential attack.

4-round iterative characteristics

For a 4-round iterative characteristic with a higher probability per round than the 2-round iterative characteristic, there can be different inputs to at most 7 S-boxes, because $(\frac{1}{4})^x \geq (\frac{1}{234})^2 \Rightarrow x < 8$, however there is no 4-round iterative characteristics for the DES with a probability higher than for the best 2-round iterative characteristic concatenated with itself. The proof is tedious and is given in Appendix B.1.

Longer characteristics

We believe that it can be proved that one cannot find n-round iterative characteristics, $n > 4$, with probabilities higher than for the best 2-round iterative characteristic concatenated with itself $\frac{n}{2}$ times. To obtain this for a 5-round iterative characteristic there can be different inputs to at most 9 S-boxes, as $(\frac{1}{4}) \geq (\frac{1}{234})^{2.5} \Rightarrow x < 10$. It seems impossible that one can find

such a characteyistic different in the inputs to 9 S-boxes and all combinations with a probability close to the highest possible of $\frac{1}{4}$ If one goes one round further to a 6-round iterative characteristic the doubt will be even bigger. Before making any conclusions for the best differential attack on the DES using characteristics, one must also check that no non-iterative characteristics exist, as stated in Section 5.2.2. Recently, M. Matsui [67] published the result of an exhaustive search for the best characteristics of the DES confirming our scepticism.

Probabilities of iterative characteristics

As stated earlier the best characteristics for a differential attack on the DES are based on 2-round iterative characteristics. The two best of these have the following input xors in the second round: $\Phi = 19600000_x$ and $\Gamma = 1b600000_x$. Both xors lead to equal outputs with probability $\frac{1}{234}$, when calculated assuming independent inputs to neighbouring S-boxes. However, this probability is only an "average" probability. As stated in [7, section 4.4.5], if the sixth key bit used in S2 is different from the second key bit used in S3 the probability for Φ increases to $\frac{1}{146}$ and the probability for Γ decreases to $\frac{1}{585}$. If the two key bits are equal the probabilities will be interchanged. We call these key bits, **critical** key bits for Φ and Γ. In their attack on the DES [7] Biham and Shamir use these two characteristics to build 13-round characteristics, where six rounds have input xor Φ or Γ. The probability is calculated to be $(\frac{1}{234})^6 \simeq 2^{-47.22}$. But depending on the values of the six pairs of critical key bits the probability for Φ will vary from $(\frac{1}{146})^6 \simeq 2^{-43.16}$ to $(\frac{1}{585})^6 \simeq 2^{-55.16}$ and the other way around for Γ. Using both characteristics as in [7] we can expect to get one characteristic with a probability of at least $(\frac{1}{146\times585})^3 \simeq 2^{-49.16}$. Table 6.1 shows the probabilities and for how many keys they will occur. As noted earlier in Section 5.2 calculating the probabilities of a characteristic as the product of the probabilities of one-round characteristics is only a valid method if the round keys are independent, which they are not for the DES. However, we made tests confirming that in an actual attack with fixed keys the probabilities for the above characteristics used by Biham and Shamir are close to $\frac{1}{146}$ and $\frac{1}{585}$ per round depending on the values of the critical key bits. It is seen that for one out of 32 keys, we will get a 13-round characteristic with the highest probability and for about one out of three keys we will get the lowest probability. We found that for other 2-round

iterative characteristics the probability splits into more than one depending on equality/inequality of certain critical key bits. It turns out that we can find 2-round iterative characteristics for which the best of these probabilities is better than the lowest for Φ and Γ. For the 2-round characteristic (with input xor) 00196000_x there is only one probability. It means that regardless of the key values this characteristic will have a probability of $\frac{1}{256}$.

#Keys ($\log_2$)	Probability ($\log_2$)
51.00	-43.16
53.58	-45.16
54.88	-47.16
54.30	-49.16

Table 6.1: The probabilities for the best 13-round characteristic obtained by using the 2 characteristics Φ and Γ.

Table 6.2 shows the probabilities for Φ and Γ and for the 2-round iterative characteristics, whose best probability is higher than $\frac{1}{256}$. It seems unlikely that we can find n-round characteristic, $n > 2$, for which the exact probabilities will be higher than for the above mentioned 2-round iterative characteristics. The round keys in the DES are dependent, therefore some key bits might be critical for one characteristic in one round and for another characteristic in another round. For example by using characteristic 19400000_x we have the two probabilities $\frac{1}{195}$ and 0. But this division of the probability depends on the values of the same critical key bits as Φ and Γ and we would get a probability of $\frac{1}{146}$ for either Φ or Γ. The characteristics marked with (+) in Table 6.2 depend on the values of the same critical key bits as for Φ and Γ. Two of these characteristics also show that is important to consider this splitting of the probabilities. Assume that the characteristic 19400000_x or $1b400000_x$ is (by far) the best characteristic for an attack on the DES A 13-round characteristic built from this characteristic will have probability zero if the value of the critical key bits are 'wrong' in just one round. It means that the characteristic only holds if the value of the critical key bits are 'right' in all 6 rounds, where they matter, i.e., 13-round characteristics built from either the characteristic 19400000_x or $1b400000_x$ are only possible for one out of 64 keys. Doing an attack on the DES similar to the one given in [7], this time using the first 5 of the above characteristics will

Characteristic	Probabilities (1/n)	Average Prob.(1/n)
19600000_x	146, 585	234
$1b600000_x$	585, 146	234
00196000_x	256	256
$000003d4_x$	210, 390	273
$4000001d_x$	205, 1024	341
19400000_x (+)	0, 195	390
$1b400000_x$ (+)	195, 0	390
40000019_x ($)	248, 390, 744, 1170	455
$4OOOOO1f_x$ ($)	248, 390, 744, 1170	455
$1d600000_x$ (+)	205, 512, 819, 2048	468
$1f600000_x$ (+)	205, 512, 819, 2048	468

Table 6.2: Exact probabilities for 11 characteristics.

give us better probabilities for a 13-round characteristic. Table 6.3 shows

#Keys ($\log_2$)	Probability ($\log_2$)
51.00	-43.16
53.58	-45.16
49.64	-46.07
49.64	-46.29
54.88	-47.16
50.90	-47.18
54.10	-48.00

Table 6.3: The probabilities for the best 13-round characteristic obtained by using 5 characteristics.

the best probabilities and for how many keys these will occur. The above probabilities are calculated by carefully examining the critical key bits for the 5 characteristics in the rounds no. 3, 5, 7, 9, 11 and 13, i.e., the rounds where we will expect the above input xors to be. By using the two characteristics in Table 6.2 marked with ($) in addition would yield slightly better probabilities. However, the best probability we would get by using these characteristics is $(\frac{1}{248})^6 \simeq 2^{-47.7}$ and it would occur only for a small number

of keys. As indicated in Table 6.3 we can expect to get a characteristic with a probability of at least 2^{-48} . However, the attacks will become more complex.

6.1.2 Analysis of the key schedule

In this section we consider the key schedule of the DES, as described in Appendix C. Theorem 6.1.1 shows that to have equal outputs of the F-function with two different inputs using the same key, the inputs must be different in the inputs to at least 3 neighbouring S-boxes. We state here a converse result, i.e.,

Lemma 6.1.1 (DES) *There exist pairs of round keys different in the inputs to only one S-box, such that using the same (text)-input, equal outputs of the F-function are obtained.*

Proof: Because the keys are added to the input after the expansion, they do not (automatically) affect neighbouring S-boxes. □

Furthermore there exist many pairs of 48 bit keys K_i and K_i' different in the inputs to only one S-box, such that equal inputs lead to equal outputs.

Example 6.1.1 *From the difference distribution table of the DES (see [7]) it follows that for S-box 1 an input xor* 03_x *leads to the output xor* 00_x *with probability* $\frac{14}{64}$. *This means that for two round keys* K_i *and* K_i' *different only in the inputs to S-box 1 with xor* 03_x, *equal inputa will lead to equal outputa with probability* $\frac{14}{64}$.

Note that although Lemma 6.1.1 tells us that we can get equal outputs of one round in the DES with keys different in the inputs to only one S-box, it does not mean, that it is easy to find iterative characteristics in this case. Once we have chosen a certain difference in the keys in one round, because of the dependencies of the round keys in DES, we have at the same time chosen the difference in all other rounds. And they might **not** lead to equal outputs. In fact we believe that it is impossible to find pairs of keys for the DES, such that in each round equal inputs and different round keys lead to equal outputs. However, we can use Lemma 6.1.1 to find what we will call **quasi weak keys** for DES.

Quasi weak keys for DES

It is clear that there should be no simple relation between the two functions $DES_K(\cdot)$ and $DES_{K^*}(\cdot)$ for any two keys K and K^*. The wellknown exceptions are the weak and semi-weak keys, a total of 16 for DES. We show that for several other pairs of keys for the DES there exists a simple relation between the encryption functions, at least for a fraction of all plaintexts.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LS_i	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1
$a[i]$	1	2	4	6	8	10	12	14	15	17	19	21	23	25	27	28

Table 6.4: The circular shifts in the key schedule of DES.

Next we consider the key schedule of the DES. The input is a 64 bit key. First the key is permuted and the parity bits are removed. This permutation has no importance for what we are about to show and we assume in the following that the input is a 56 bit (permuted) key. The 56 bits are divided into two blocks C_0 and D_0 of 28 bits each. The round keys K_i for $i = 1, \ldots, 16$ are defined

$$K_i = PC2(C_i \parallel D_i)$$

where $C_i = LS_i(C_{i-1})$, $D_i = LS_i(D_{i-1})$, $PC2$ is a permutation and where LS_i is a left circular shift by the no. of positions given in Table 6.4. Alternatively, we could define

$$L_i(C_0 \parallel D_0) = (LS_{a[i]}(C_0) \parallel LS_{a[i]}(D_0))$$

where $a[i]$ is the accumulated number of shifts given in Table 6.4 and then define

$$K_i = PC2(L_i(K))$$

where $K = (C_0 \parallel D_0)$, the 56 bit key. In the following we will use the alternative definition of the key schedule of DES.

Theorem 6.1.2 (DES) *For every key K, there exists a key K^*, such that*

$$K_{i+1} = K_i^*, \text{ for } i \in \{2, \ldots, 7\} \cup \{9, \ldots, 14\}$$

i.e., K and K^* have 12 common round keys.

Proof: Suppose we are given the key K. Set $K^* = L_2(K)$, where L is defined as above. Now it follows easily that

$$K_3 = PC2(L_3(K)) = PC2(L_2(K^*)) = K_2^*.$$

And similarly, $K_{i+1} = K_i^*$ for $i = 2, \ldots, 7$. Further, $K_9 = PC2(L_9(K))$ and $K_8^* = PC2(L_8(K^*))$. After this the round keys get 're-synchronised', since

$$K_{10} = PC2(L_{10}(K)) = PC2(L_9(K^*)) = K_9^*.$$

And $K_{i+1} = K_i^*$ for $i = 9, \ldots, 14$. □

Theorem 6.1.3 (DES) *There exist 256 pairs of keys K and K^*, such that*

$$K_{i+1} = K_i^*, \text{ for } i \in \{2, \ldots, 14\}$$

i.e., K and K^* have 13 common round keys.

Proof: From Theorem 6.1.2 and by searching exhaustively for pairs of keys K and K^*, for which $K_9 = PC2(L_9(K)) = PC2(L_8(K^*)) = K_8^*$. □

For these pairs of keys we found that there is some connection between the two encryption functions defined by the pair. In the following δ_i and ϵ_j denote 32 bit values. For every pair $\{\delta_i, \epsilon_j\}$ a probability $p_{i,j}$ is connected.

Theorem 6.1.4 (DES) *Let K and K^* be a pair of keys from Theorem 6.13 Then there exist sequences $\{\delta_i, p_{\delta_i}\}$ and $\{\epsilon_j, p_{\epsilon_j}\}$, such that with $P = P_L \parallel P_R$ and $P^* = P_R \oplus \delta_i \parallel P_L \oplus F(K_1, P_R)$*

$$\begin{aligned} DES(K, P) &= C_L \parallel C_R \Rightarrow \\ DES(K^*, P^*) &= C_R \oplus F(K_{16}^*, C_L \oplus \epsilon_j) \parallel C_L \oplus \epsilon_j \end{aligned} \tag{6.1}$$

with probability $p_{\delta_i} \times p_{\epsilon_j} = p_{i,j}$. Furthermore for the pairs of keys of Theorem 6.1.3

$$\sum_{i,j} p_{i,j} = 1$$

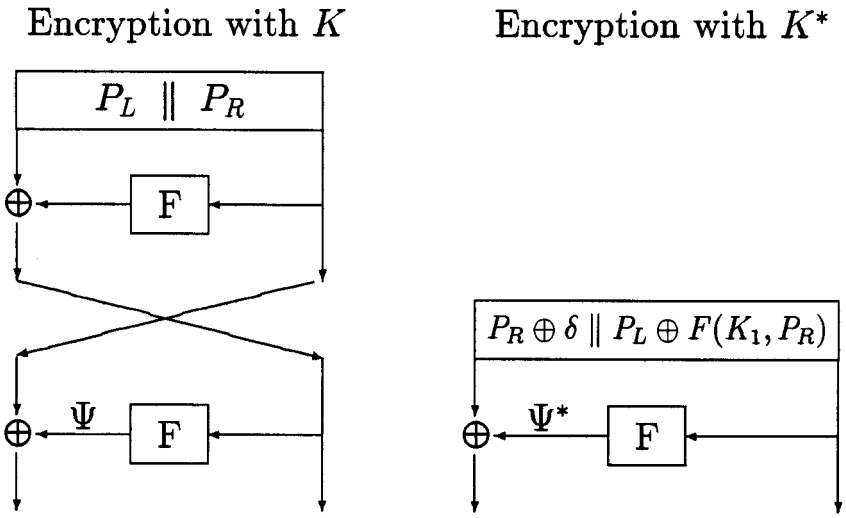


Figure 6.1: The two encryptions using quasi weak keys.

Proof: Let K and K^* be a pair of keys from Theorem 6.1.3. Choose a random plaintext $P = P_L \parallel P_R$. Encrypt P using K obtaining $C = C_L \parallel C_R = DES(K, P)$. Let the right half of P^* be $P_L \oplus F(K_1, P_R)$. The right half inputs (before addition of the keys) to the second round of $DES(K, P)$ and the first round of $DES(K^*, P^*)$ are equal, see also Figure 6.1. Let the difference in the round keys be $\Delta K_{2,1} = K_2 \oplus K_1^*$. That is, the difference in the inputs to the S-boxes of respectively the second and first round is $\Delta K_{2,1}$.

It is now easy from the difference distribution table of the DES to find a possible xor of the outputs of the respective rounds. Denote the outputs Ψ and Ψ^* and define $\delta = \Psi \oplus \Psi^*$; the corresponding probability from the xor table is denoted p_δ. Let the left half of P^* be $P_R \oplus \delta$. Now the right half input to the third round of the encryption with K is $P_R \oplus \Psi$ and the right half of the input to the second round of the encryption with K^* is $P_R \oplus \delta \oplus \Psi^*$, i.e.,the inputs are equal, since $P_R \oplus \Psi \oplus P_R \oplus \delta \oplus \Psi^* = 0$. The left halves of the inputs to the corresponding rounds are also equal and since the keys are equal from now on and until the 16'th and 15'th round respectively, according to Theorem 6.1.3, it follows that the two encryptions are the same until the last and second last round respectively. For these rounds the right halves of the inputs are equal and the xor of keys is $\Delta K_{16,15} = K_{16} \oplus K_{15}^*$. Let ϵ denote a possible xor of the outputs with input xor $\Delta K_{16,15}$ and the

corresponding probability p_ϵ.
Now equation (6.1) holds with probability $p_{\delta,\epsilon} = p_\delta \times p_\epsilon$. To complete the proof we notice that for a given plaintext there is only one value for δ and ϵ above and that for all plaintexts there are only a limited number of choices for δ and ϵ, which depend on the keys (K, K^*) and they can easily be identified using the difference distribution table. □

Example 6.1.2 *Let* $K^* = 4020\ 0000\ 1080\ 9080_x$ *and* $K = 0000\ 0080\ 9080\ 9080_x$ *in hexadecimal notation, this pair is one of the pairs from Theorem 6.1.3. The connection between the round keys of the pair is as follows.* $K_i^* = K_{i+1}$ *for* $i = 2, \ldots, 14$ *and*

$$\begin{aligned} K_1^* \oplus K_2 &= 00_x, 20_x, 00_x, 00_x, 00_x, 00_x, 00_x, 00_x \\ K_{15}^* \oplus K_{16} &= 05_x, 00_x, 00_x, 00_x, 00_x, 00_x, 00_x, 00_x \end{aligned}$$

where we have arranged the key bits into 8 groups of 6 bits each (hex).
From the difference distribution table of the DES we find that for S-box 2, there are 9 possible xors of the outputs with an input xor 20_x*. The most likely xor of the outputs is* C_x*, which has probability* $\frac{14}{64}$*. Let* $\delta_1 = P(0C000000_x)$*, where* P *is the 32-bit permutation at the end of the F-function, and denote the probability* p_{δ_1}*.*
Similarly, we find that there are 14 possible xors of the outputs with an input xor 05_x *for S-box 1. The most likely xor of the outputs is (again)* C_x*, which has probability* $\frac{12}{64}$*. Let* $\epsilon_1 = P(C0000000_x)$ *and denote the probability* p_{ϵ_1}*. With*

$$P = P_L \parallel P_R \text{ and } P^* = P_R \oplus \delta_1 \parallel P_L \oplus F(K_1, P_R)$$

we obtain $DES(K, P) = C_L \parallel C_R \Rightarrow$

$$C^* = DES(K^*, P^*) = C_R \oplus F(K_{16}^*, C_L \oplus \epsilon_1) \parallel C_L \oplus \epsilon_1$$

with probability $p_{1,1} = p_{\delta_1} \times p_{\epsilon_1} = \frac{14 \times 12}{64^2} \simeq \frac{1}{24}$*. For the two keys in this example there are* $9 \times 14 = 126$ *pairs* $\{\delta_i, \epsilon_j\}$ *in Theorem 6.1.4.*

Since this phenomenon is due to only the xor of some round keys of K and K^*, a similar result holds for the complemented pairs of keys $\overline{K}$ and $\overline{K^*}$.

For all pairs of keys, K and K^* from Theorem 6.1.2, $K_9 \neq K_8^*$ except for the 256 pairs of keys of Theorem 6.1.3. As shown above the input to the

ninth round for encryption with K^* and the input to the eighth round for encryption with $*$ will be equal with some probability δ. That means that the input xor for the two encryptions will be $(K_9 \oplus K_8^*)$, since the (text)-inputs are equal. Lemma 6.1.1 shows that it is possible for keys that differ in the inputs to only one S-box to lead to equal outputs. Since the key schedule of the DES operates on 24 bit halves it is possible to do an exhaustive search for this phenomenon for all keys. We implemented this test and found that for $2^{48.7}$ pairs of keys, K and K^*, the input xor $(K_9 \oplus K_8^*)$ will lead to equal outputs for some fraction of all plaintexts. For the $2^{48.7}$ pairs of keys this fraction varies from $\frac{1}{4}$ to 2^{-39}. Therefore for these keys we have a parallel to Theorem 6.1.4.

Theorem 6.1.5 (DES) *For $2^{48.7}$ pairs of keys K and K^*, it holds that for a fraction p_{KK^*} of all plaintexts there exist sequences $\{\delta_i, p_{\delta_i}\}$ and $\{\epsilon_j, p_{\epsilon_j}\}$, such that with pobability $p_{\delta_i} \times p_{\epsilon_j} = p_{i,j}$ and*

$$P = P_L \parallel P_R \text{ and } P^* = P_R \oplus \delta_i \parallel P_L \oplus F(K_1, P_R)$$

$$\begin{aligned} DES(K, P) = C = C_L \parallel C_R \Rightarrow \\ DES(K^*, P^*) = C_R \oplus F(K_{16}^*, C_L \oplus \epsilon_j) \parallel C_L \oplus \epsilon_j \end{aligned}$$

where $p_{i,j}$ is defined as in Theorem 6.1.4. Similarly it holds that

$$\sum_{i,j} p_{i,j} = 1$$

Corollary 6.1.1 *There are 2368 pairs of keys for which the fraction P_{KK^*} is $\frac{1}{4}$.*

We conclude that for many pairs of keys in the DES there is a simple relation between the encryption functions induced by these keys. This simple relation corresponds to one round of DES encryption and for 256 pairs of keys it holds for all plaintexts. For other $2^{48.7}$ pairs of keys it holds for a fraction of all plaintexts.

Applications. Since the phenomenon of Theorem 6.1.4 and Theorem 6.1.5 holds only for a small subset of keys and for most keys only for a fraction of all plaintexts, it is doubtful that the quasi weak keys can be

exploited in attacks on the DES itself. However, it is interesting to note that the phenomenon could easily have been avoided, e.g. by changing two '2' shifts, e.g. in rounds 6 and 7, see Table 6.4, in the key schedule by a '3' and a '1' shift respectively.

The DES is often used in hash functions where the keys are fixed or can be chosen as part of the (hash) message [93]. It seems possible that quasi weak keys can be exploited in attacks on these hash functions. In differential attacks on hash functions based on block ciphers one could find two plaintexts, such that the (δ, ϵ)'s of Theorem 6.1.4 are equal, thereby in a differential the δ's and the rightmost ϵ's in (6.1) would cancel out.

By trying sufficiently many pairs of plaintexts useful differentials (with fixed keys) might be found and used in attacks on hash functions.

The round key bits

The key schedule of the DES take a 64 bit input K and outputs 16 round keys of 48 bits each, a total of 768 bits. The parity bits of K are removed and only 56 bits of K are used. Since $768/56 \simeq 13.7$ is not an integer, some bits in key K are used more often than other key bits in the round keys. By a closer look at the key schedule it follows that the key bits are contained in either 12, 13, 14 or 15 round keys. Table 6.5 lists the exact number of round keys for all key bits in $K = k_1, \ldots, k_{64}$. Consider an attack where the

# Round keys	The bit numbers
12	3 , 42 , 52 , 58
13	7 , 10 , 12 , 14 , 19 , 23 , 26 , 28 29 , 33 , 36 , 38 , 39 , 43 , 45 , 49 54 , 55 , 59 , 61
14	1 , 4 , 5 , 6 , 9 , 11 , 13 , 15 17 , 20 , 21 , 22 , 25 , 27 , 30 , 35 46 , 51 , 62 , 63
15	2 , 18 , 31 , 34 , 37 , 41 , 44 , 47 50 , 53 , 57 , 60

Table 6.5: The number of times the key bits appear in round keys.

attacker guesses some of the key bits and tries to find the remaining key bits

faster than by exhaustive search, a "chosen key bit" attack. By guessing the 12 key bits, which appear in 15 round keys, the attacker would get 180 out of 768 round key bits, i.e., $\frac{180}{768} \simeq 23.4\%$ of the round key bits, by guessing $\frac{12}{56} \simeq 21.4\%$ of the bits in the key K. Whether this phenomenon can be used in a cryptanalytic attack is an open question.

6.1.3 Higher order differentials

As mentioned in Section 5.2.4 the use of higher order differentials is restricted to iterated block ciphers with a small number of rounds. Next we consider higher order differentials for the 8 S-boxes of the DES. As noted in Section 5.2.4 first order differentials correspond to the differentials used by Biham and Shamir [7]. Therefore the set of first order differentials for one S-box cor-

S-box no.	1. order	2. order	3. order	4. order
1	16	24	48	64
2	16	28	48	64
3	16	28	40	64
4	16	48	64	64
5	16	28	40	64
6	16	24	40	64
7	16	28	40	64
8	16	28	40	64

Table 6.6: The probabilities ($\times 64$) of the best higher order differentials for the 8 S-boxes of DES.

responds to the difference distribution tables for the DES S-boxes [7]. Table 6.6 lists the probabilities of the most likely n'th order differentials for the 8 S-boxes of the DES, for $n = 1, \ldots, 4$. Note that the probability of any fifth order differential is one, since the output coordinates of the DES S-boxes have order 5 and the fifth derivative is a constant, see Section 5.2.4. The numbers for S-box 4 in Table 6.6 are substantially different from the numbers of the other S-boxes and there exist 3. order differentials with probability one.

Example 6.1.3 *With* $\alpha_1 = 25_x, \alpha_2 = 24_x$ *and* $\alpha_1 = 30_x$ *the third order differential of S-boz 4,* $\Delta_{\alpha_1,\alpha_2,\alpha_3}(S4) = f_x$ *with prbability one, Note that*

$\Delta_{\alpha_1,\alpha_2,\alpha_3}(S4)$ *is the exclusive-or of eight six bit inputs.*

We have found no way of exploiting higher order differentials for the DES, other than by attacking a four round version of the DES. However, since the DES with four rounds is trivially broken using first order differentials, this application is not of much use.

6.1.4 Partial differentials

As mentioned in Section 5.2.6 the use of partial differentials is restricted to iterated block ciphers with a small number of rounds. For the DES there are partial differentials with probability one. When two inputs to the F-function are equal in the inputs to an S-box, the outputs from that S-box are always equal, independently of the values of the inputs to other S-boxes. These partial differentials are used to a wide extent in Biham and Shamir's attacks on the DES [7].

The outputs of S-box	Does not affect S-boxes
1	1, 7
2	2, 6
3	3, 1
4	4, 2
5	5, 8
6	6, 4
7	7, 5
8	8, 3

Table 6.7: Flow of the S-box output bits.

The output of an S-box affects the inputs of at most six S-boxes in the following round, because of the P-permutation, see Table 6.7. This fact can be used to construct a four round partial differential for the DES, which gives knowledge about the difference of eight bits in the ciphertext after four rounds. Consider a pair of plaintexts with a difference, such that the right halves are equal and the left halves differ, such that the inputs to only one S-box, say S-box 1, are different after the E-expansion. The first round in the differential holds always, and in the second the outputs of all S-boxes

except S-box 1 are equal. In the inputs to the third round the inputs of two S-boxes, S-boxes 1 and 7, are always equal, since S-box 1 does not affect these S-boxes according to Table 6.7. Therefore the outputs of these S-boxes are equal, and the xor of eight bits in the right halves of the ciphertexts after three rounds are known, since the xor in the inputs in the second round is known. Since the right halves after three rounds equal the left halves after four rounds, the xor of eight bits after four rounds of encryption are known with probability one. This differential can be used to attack the DES with 6 rounds in a differential at tack using only a few chosen plaintexts.

Theorem 6.1.6 *There exists a differential attack on the DES with 6 rounds, which finds the secret key using* 32 *chosen plaintexts in time about 20,000 encryptions, which can be done in a few minutes on a PC.*

Proof: We consider a differential chosen plaintext attack using the differential in Figure 6.2. Assume first that the outputs of the first round have difference α. The inputs to the third round differ in only two bits both affecting only S-box 1. According to the above discussion, the inputs with difference X to the fourth round are equal in the inputs to the S-boxes 1 and 7. Therefore eight bits of the difference Y are zero. Since the difference of the inputs to the third round is known, the attacker knows eight bits of the difference of the outputs of the F-function in the sixth round, since he knows the difference in the ciphertexts. These eight bits are the output bits of S-boxes 1 and 7. The attacker now tries for all 64 possible values of the key whether the inputs to S-box 1 yield the computed expected output difference, and does the same for S-box 7. For every pair of ciphertexts used in the analysis for both S-boxes the attacker will get an average of 4 suggested key values, among which the right key value appears, since the used differential has probability one. By trying a few pairs, e.g. four pairs, only one key value, the right key value, will be left suggested by all pairs.

In the following, let K_{ij} denote the six bit key in S-box no. j in the i'th round. We assumed above that the difference of the outputs of the first round is α, which it will not automatically be. However, we can apply a variation of the first round trick described in Section 5.3. First we note that since the inputs to the first round differ in the inputs to only one S-box, there are only 16 possible values of α. Choose a set of 16 plaintexts $P_i = (a_i \mid P_R)$, for $i = 0, \ldots, 15$, where P_R is a randomly chosen 32-bit string and the values a_i are different only in the four bits corresponding to the outputs of S-box 1

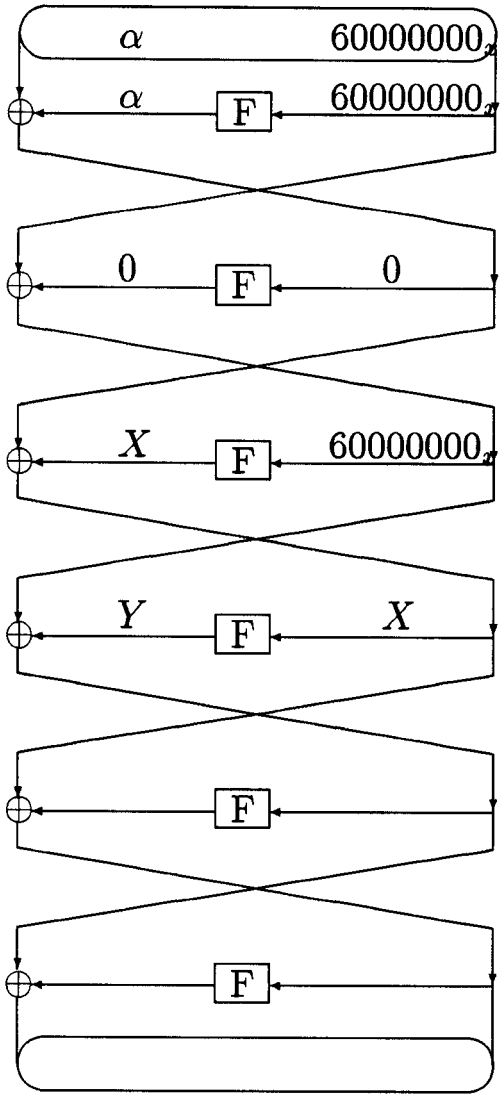


Figure 6.2: A 4 round differential af DES.

after the P-permutation, i.e., the exclusive-or $a_i \oplus a_j = P(z0000000_x)$ for all 16 values of i, j. Get the encryptions of those 16 plaintexts. Now choose a set of 4 plaintexts $P_{1,j} = (b_{1,j} \mid P_R \oplus \Phi_1)$, for $j = 0, \ldots, 3$, where $\Phi_1 = 60000000_x$ and where the $b_{1,j}$'s differ only in the same subset of bits as the a_i's and get the encryptions of these plaintexts. The attack proceeds as follows.

1. For every value $k_{1,1}$ of the key $K_{1,1}$ to S-box 1 in the first round do
   (a) Find $c_1 = F(k_{1,1}, P_R)$ and $c_2 = F(k_{1,1}, P_R \oplus \Phi_1)$
   For $j = 0$ to 3 find the plaintext P_i, such that $a_i = c_1 \oplus c_2 \oplus b_{1,j}$. The pair of plaintexts P_i and $P_{1,j}$ is a right pair with respect to the characteristic in Figure 6.2.
   (b) Use the four right pairs in the differential attack described above. First do the attack on S-box 1 in the last round. If one key value $k_{6,1}$ of $K_{6,1}$ is suggested by all four pairs, perform the differential attack on S-box 7 in the last round. If one key value $k_{6,7}$ of $K_{6,7}$ is suggested by all four pairs, take $k_{6,1}$ and $k_{6,7}$ as the key values of $K_{6,1}$ and $K_{6,7}$ and take $k_{1,1}$ as the values of $K_{1,1}$.

The above attack finds 18 key bits with a high probability. For every value of $K_{1,1}$ we do two rounds of encryption in the first round. Then for every value of $K_{6,1}$ we do one round of encryption for the 8 ciphertexts in the 4 pairs, totally the time used is about the time of 5000 encryptions of six round DES. Note that the differential used in the attack has probability one. The remaining key bits can be found in similar attacks by choosing further 3 sets of 4 plaintexts $P_{n,j} = (b_{n,j} \mid P_R \oplus \Phi_n)$, for $j = 0, \ldots, 3$, and $n = 2, \ldots, 4$, where $\Phi_2 = 06000000_x$, $\Phi_3 = 00600000_x$ and $\Phi_4 = 00060000_x$. The 16 plaintexts P_i in the above described attack can be reused. The attack needs a total of 32 plaintexts and runs in time about $20,000$ encryptions of six round DES, which can be done in a few minutes on a PC. Since the DES has dependent round keys many of the key bits tried in the first and in the sixth round will be the same, and many key values do not have to be tried. Finally we note that the performance of the attack can be improved by pre-computing tables to reduce the number of encryptions needed in step (1b) of the attack. Also our estimates are worst-case considerations; the expected time will be the time of about $10,000$ encryptions. □

It should be noted that the linear attack combined with differential 'techniques' by Hellman and Langford [34] exploits the same phenomenon as in

our attack, but the two attacks are different.

There are other interesting partial differentials for the DES. Consider a six bit input difference (xor) to one S-box, $x_1, x_2, x_3, x_4, x_5, x_6$ and the corresponding difference in the outputs y_1, y_2, y_3, y_4. Instead of considering all 4 output bits as in traditional differentials, we consider only one of the y_i's. The probabilities of these partial *1-bit output* differentials in the ideal case will be 1/2. In Table 6.8 for all 8 S-boxes the partial differentials for which $|p - 1/2| \geq 20/64$ are listed, where p is the probabilities of the differentials. Note That if p is the probability that an xor bit is one, $1-p$ is the probability that the bit is zero. As an example consider S-box 2, where an input xor of 20_x leads to an output xor, for which the xor of the second most significant bits of the outputs is one in 60 out of all 64 possible pairs of inputs. It is also interesting to note that for the S-box 4, the probabilities of partial 1-bit output differentials are all between 20/64 and 44/64, i.e., $|p-1/2| \leq 12/64$ for S-box 4. S-box 4 has been the subject of much debate since the publication of the DES and it has been conjectured the weakest S-box. It is 75% redundant, see [35] and it has a strange difference distribution table (see [7]). See also the previous section on higher order differentials. To our knowledge the above properties show for the first time a case, where S-box 4 is the strongest of the 8 S-boxes.

Another interesting property of the S-boxes is revealed by considering pairs of input where the only the two middle input bits differ, i.e., xors 04_x, 08_x and $0c_x$. These xors are of particular interest in differential cryptanalysis, since this allows neighbouring S-boxes to have equal inputs, i.e., xors 00_x. For these input xors, the probability that one particular bit in the output xor is zero is at most 36/64 for S-boxes no. 2, 3 and 7. For the S-boxes 1, 5, 6 and 8 the probability is at most 32/64 and for S-box 4 at most 24/64. We can use the above partial differentials to construct a four round differential, which gives knowledge about the difference of more than eight bits in the ciphertext after four rounds.

As an example, for S-box 7, an input xor of 04_x will yield an output xor, such that the xor of the second output bits (y_2) is zero with probability 36/64. Consider a pair of plaintexts with difference $00000020_x \mid 00000000_x$, that is where the right halves are equal and the left halves differ in only one bit. After one round of encryption the difference will always be $00000000_x \mid 00000020_x$. After two rounds of encryption the difference will be $00000020_x \mid Y_x$ with

S-box	Input xor (hex)	Bit i (y_i) of output xor i	Probability $(p - 1/2) \times 64$
1	24	3	-20
2	2	2	20
	c	4	20
	20	2	28
	22	2	-20
	2d	1	20
3	2	1	20
	4	1	20
	10	4	24
	20	2	24
5	1	2	20
6	4	3	20
	c	4	20
	24	1	-24
7	2	2	24
	c	2	20
	e	2	-20
	20	4	24
8	1	2	20
	10	3	20
	20	4	20

Table 6.8: The partial 1-bit output differentials with $|p - 1/2| \geq 20/64$ for the 8 S-boxes of DES.

probability 36/64, where $E(Y)$ is different in the inputs to only S-boxes 1, 2, 6, and 8. Therefore the outputs of S-boxes 3, 4, 5, and 7 will be equal after three rounds of encryption. In other words one gets knowledge of the xor of 16 bits in the right halves of the ciphertexts after three rounds and therefore in the left halves of the ciphertexts after four rounds of encryption with probability 36/64.

In a similar way, one can use two of the combinations of Table 6.8, namely the input xor 04_x for S-box 6 and the input xor 04_x for S-box 3, both with probability 52/64 to obtain a four round partial differential with probability $(52/64)^2 \simeq 42/64$ where the xor of nine ciphertext bits are known.

Note that although the above differentials can be used to deduce key bits of the DES with 6 or fewer rounds in a partial differential attack, it is also clear that when considering the DES with more than 6 rounds the method will only work locally in the first few and last few rounds of the cipher.

Finally we note that in [97] Preneel et al. exploited, what they called, *reduced exors*, in differential attacks on the DES in CFB mode. The reduced exors have some resemblance with partial differentials.

6.1.5 Linear cryptanalysis

In this section we examine the DES for the iterative linear characteristics described in Section 5.3. We will use the same notation as given in [64].

Definition 6.1.1 (DES) *For a given S-box* $S_i, i = 1, \ldots, 8, \alpha \in GF(2)^6$ *and* $\beta \in GF(2)^4$ *define*

$$NS_i(\alpha, \beta) = \#\{x \in GF(2)^6,\ x \cdot \alpha = S_i(x) \cdot \beta\} \tag{6.2}$$

where '$\cdot$' denotes the dot product.

In the following we refer to the figures in Section 5.3 and denote by X_i be the right half of the ciphertext after i rounds of encryption, i.e., X_i is the input to the F-function in the $(i+1)$'th round. X_0 denotes the right half of the plaintext input to the linear characteristic. Also for every round let us fix a key k and for convenience let $F(X)$ denote $F(X, k)$.

2-round iterative characteristics

For this type of characteristic we consider the following expression

$$max_{\alpha}|\Pr_X(F(X)\cdot\alpha = 0) - 1/2| \tag{6.3}$$

One S-box of the DES is a balanced function, therefore (6.3) is zero for all Q only affecting one S-box. In [65] equation (6.3) was examined for two neighbouring S-boxes. The best expression uses the following two approximations,

$$NS_7(3_x, f_x) \text{ and } NS_8(30_x, d_x)$$

which combined give a probability of 0.453. Since the input bits to the two S-boxes are shared combining the two expressions will cancel out all input bits and leave only output and key bits. It is the best linear approximation of the type (6.3). Also one can look for the following expressions with three neighbouring S-boxes involved.

$$\begin{array}{l} NS_i(2_x, *),\ NS_{i+1}(23_x, *) \text{ and } NS_{i+2}(30_x, *) \\ NS_i(3_x, *),\ NS_{i+1}(33_x, *) \text{ and } NS_{i+2}(30_x, *) \\ NS_i(3_x, *),\ NS_{i+1}(31_x, *) \text{ and } NS_{i+2}(10_x, *) \end{array}$$

where '$*$' denotes the any of the 16 possible values. When combined each of these three expressions will leave only output and key bits in the linear expression. For the DES the best one of these is of the third type,

$$NS_1(3_x, 9_x)NS_2(31_x, b_x) \text{ and } NS_3(10_x, b_x)$$

with a probability of 0.488.

3-round iterative characteristics

For this type of characteristic we consider the following expression

$$max_{\alpha,\beta}(|\Pr_{X_1}(F(X_1)\cdot\alpha = X_1\cdot\beta) - 1/2| \times |\Pr_{X_2}(F(X_2)\cdot\beta = X_2\cdot\alpha) - 1/2|) \tag{6.4}$$

This kind of characteristic has not been reported by Matsui. However, they exist for the DES with only one active S-box per round. The best one is $NS_7(4_x, 8_x)$ and $NS_8(2_x, 4)$, with a probability of $1/2 - 2^{-8}$. Extended to a 14 round characteristic, one obtains a probability of $1/2 - 2^{-32}$.

4-round iterative characteristics

As noted in Section 5.3 this is the type of characteristic used by Matsui in the attack on 16-round DES [64, 65, 66]. For this type of characteristic we consider the following expression

$$max_{\alpha,\beta}|\Pr_X(X_4 \cdot \beta = X_0 \cdot \alpha) - 1/2|^2 \tag{6.5}$$

where (see also Figure 5.9, page 88)

$$\begin{aligned}
&|\Pr_X(X_4 \cdot \beta = X - 0 \cdot \alpha) - 1/2|^2 \\
&\quad = |p_{4R} - 1/2|^2 \\
&\quad = 4^2 \times \sum_A |p_1(A) - 1/2|^2 \times |p_2(A) - 1/2|^2 \times |p_3(A) - 1/2|^2
\end{aligned}$$

and where

$$\begin{aligned}
p_1(A) &= \Pr_{X_1}(X_1 \cdot A = F(X_1) \cdot \alpha) \\
p_2(A) &= \Pr_{X_2}(X_2 \cdot (\alpha \oplus \beta) = F(X_2) \cdot A) \\
p_3(A) &= \Pr_{X_3}(X_3 \cdot A = F(X_3) \cdot \beta)
\end{aligned}$$

For one value of A one obtains the best probability using

$$NS_5(10_x, e_x),\ NS_1(4_x, 4_x) \text{ and } NS_5(10_x, f_x)$$

in the second, third and fourth rounds respectively, where the 32 bit quantities are $A = 00008000, \alpha = 01040080_x$ and $\beta = 21040080_x$. The probability is 0.506 [64]. However, there is another value of A for the same values of α and β, namely $A = 00008800$, where the combinations are $NS_5(11_x, e_x)$ in the second round, $NS_8(03_x, 1_x)$ and $NS_1(34_x, 4_x)$ in the third round and $NS_5(11_x, f_x)$ in the fourth round. The probability is $1/2 - 2^{-15}$. For the 4 round iterative characteristic with $\alpha = 01040080_x$ and $\beta = 21040080_x$

$$|\Pr_X(X_4 \cdot \beta = X_0 \cdot \alpha) - 1/2|^2 \geq 4^2 \times (|0.006|^2 + |2^{-15}|^2) \simeq 16 \times |0.006|^2 \tag{6.6}$$

It is seen that the improvement is hardly measurable. However, "many a little makes a mickle" and in the coming section we illustrate that for longer characteristics the improvement is bigger.

Name	2. round	3. round	4. round	$-log_2(\|p_L - 1/2\|)$
L_1	$NS_5(10_x, e_x)$	$NS_1(4_x, 4_x)$	$NS_5(10_x, f_x)$	07.36
L_2	$NS_5(11_x, e_x)$	$NS_8(03_x, 1_x)$ $NS_1(34_x, 4_x)$	$NS_5(11_x, f_x)$	15.00
L_3	$NS_5(22_x, e_x)$	$NS_8(03_x, 4_x)$ $NS_1(30_x, 4_x)$	$NS_5(22_x, e_x)$	10.00
L_4	$NS_5(22_x, f_x)$	$NS_8(03_x, 4_x)$ $NS_1(30_x, 4_x)$	$NS_5(22_x, f_x)$	10.83
L_5	$NS_5(22_x, e_x)$	$NS_8(03_x, 4_x)$ $NS_1(38_x, 4_x)$	$NS_5(22_x, a_x)$	11.42
L_6	$NS_5(22_x, f_x)$	$NS_8(03_x, 4_x)$ $NS_1(38_x, 4_x)$	$NS_5(22_x, b_x)$	11.83

Table 6.9: 4 round iterative linear characteristics.

Longer characteristics

As noted earlier calculating the probabilities of linear r-round characteristics for large r, $r > 4$, is a difficult task. We end this section by giving some other linear characteristics for the DES, for which the bits of the plaintext and ciphertext are the same as for Matsui's 14 round linear characteristic used in his attack on 16 round DES [65, 66]. Matsui counts on key bits in the first and last rounds, so we consider the characteristic starting in the second round, where we assume that we know the bit $X_1 \cdot \alpha$, where X_1 is the right half of the ciphertext after one round of encryption and $\alpha = 01040080_x$. The characteristic ends in the fifteenth round with knowledge about the bit $X_{15} \cdot \beta$, where $\beta = 21040080_x$. Let $\overline{L_i}$ denote the four round characteristic obtained from L_i by interchanging the rounds number two and four. Matsui's characteristic is the concatenation of L_1, $\overline{L_1}, L_1$, one round without any approximation and one round with the combination $NS_5(10_x, f_x)$. Totally this characteristic has a probability, p_L, s.t. $|p_L - 1/2|^2 \simeq 2^{-42.97}$. In Table 6.9 we have listed some other 4 round iterative linear characteristics, that can replace L_1 in a 14 round characteristic. As also explained above L_2 can replace L_1 directly, since for four rounds exactly the same bits of the plaintext and ciphertext are affected. The concatenation $L_3, \overline{L_3}$ can replace $L_1, \overline{L_1}$. This holds also for $L_5, \overline{L_5}$. The concatenation $L_4, \overline{L_4}$ can replace $L_1, \overline{L_1}$. This holds also for $L_6, \overline{L_6}$. Totally this gives us eight paths from $X_1 \cdot \alpha$ to $X_{15} \cdot \beta$,

yielding

$$|\Pr_X(X_1 \cdot \alpha = X_{15} \cdot \beta) - 1/2|^2 \geq 2^{-42.96}.$$

This is only slightly higher than for the best single characteristic.

We found other characteristics like the ones in Table 6.9, however with even smaller probabilities. We looked only for other 4-round iterative characteristics. By examining also n-round characteristics, $n > 4$, one might get other interesting results. Since Matsui's expression involves at most one S-box for each round and optimises the use of the best one-round expression, with probability $\frac{52}{64}$ it is doubtful that the probability of his characteristic will be higher than the above estimate.

Probabilities of iterative characteristics

In Section 6.1.1 we showed that the probabilities of characteristics in differential attacks varies, when neighbouring S-boxes are considered. One S-box in the DES take a 6 bit text input and a 6 bit key input. Two neighbouring S-boxes in the DES share two input text bits and take a 10 bit text input and a 12 bit key input. Also in linear characteristics the probabilities of approximations involving neighbouring S-boxes varies depending on the actual values of the four key bits, that affect the shared text bits for two S-boxes.

Example 6.1.4 *Consider L_2 from Table 6.9. In the second round we use*

$$NS_8(03_x, 1_x), NS_1(34_x, 4_x)$$

When the two approximations are calculated separately the probability, $|p_L - 1/2|$ is $2 \times \frac{2\times4}{64\times64} \simeq 2^{-8}$ using the Piling-Up Lemma. However, if the sixth key bit of S_8 and the second key bit of S_1 are equal the exact probability, $|p_L - 1/2|$ is $\frac{6}{1024} \simeq 2^{7.42}$ and when they are different, $|p_L - 1/2|$ is $\frac{2}{1024} \simeq 2^{-9}$.

This splitting of the probability does not seem to have the same importance in linear cryptanalysis on the DES as in differential cryptanalysis on DES. Whether the phenomenon can be used to find good linear approximation, where neighbouring S-boxes in some rounds are considered, is left as an open problem.

6.1.6 Epilogue

Since its introduction in the late seventies, the DES has been the subject of intense debate and cryptanalysis. Like any other practical cryptosystem, the DES can be broken by searching exhaustively for the key.

One natural direction of research is therefore to find attacks that will be faster than exhaustive search, measured in the number of necessary encryption operations. The most successful attack known of this kind is the linear attack by Matsui [64, 65, 66]. This attack requires about 2^{43} known plaintext blocks. Although this is less than the expected 2^{55} encryptions required for exhaustive key search, the attack is by no means more practical than exhaustive search. There are two reasons for this: first, one cannot in practice neglect the time needed to obtain the information about the plaintext; second, when doing exhaustive key search the enemy is free to invest as much in technology as he is capable of to make the search more efficient, while in a known plaintext attack he is basically restricted to the technology of the legitimate owner of the key, and to the frequency with which the key is used. In virtually any practical application, a single DES key will be applied to much less than 2^{43} blocks, even in its entire life time. The difference between the two kinds of attacks is illustrated in a dramatic way by the results of Wiener [112] who shows by concrete design of a key search machine that if the enemy is willing to make a million dollar investment, exhaustive key search for the DES is certainly not infeasible.

As a result, we have a situation where the DES has proved very resistant over a long period to cryptanalysis and therefore seems to be as secure as it can be in the sense that by far the most practical attack is a simple brute force search for the key. The only problem is that the key is too short given today's technology, and that therefore, depending on the value of the data you are protecting, plain DES may not be considered secure enough anymore.

What can be done about this problem? One obvious solution is to try to design a completely new algorithm. This can only be a very long term solution: a new algorithm has to be analysed over a long period before it can be considered secure; also the vast number of people who have invested in DES technology will not like the idea of their investments becoming worthless overnight. An alternative is to devise a new system with a longer key using the DES as a building block. This way existing DES implementations can still be used. This subject is treated further in Section 7.9.

6.2 LOKI'91

In 1990 Brown, Seberry and Pieprzyk [15] proposed a new encryption primitive, called LOKI, later renamed LOKI'89, as an alternative to the Data Encryption Standard (DES), with which it is interface compatible. Cryptanalysis showed weaknesses in LOKI'89 [7, 14, 48] and a redesign, LOKI'91 was proposed in [14]. We give a full description of LOKI'91 in Appendix D. The ciphers from the LOKI family are variants of the DES-like iterated block ciphers of Definition 2.5.3, where the key is added to the text input before the expansion. The F-function uses 4 (identical) S-boxes, each substituting a 12 bit value by a 8 bit value. Four bits of the input are used to select one of sixteen functions, each of which are exponentiations in the finite field $GF(2^8)$. The F-function is defined as follows

$$F(K_i, R_{i-1}) = P(S(E(R_{i-1} \oplus K_i)))$$

where E is an expansion from 32 to 48 bits, S are the 4 identical S-boxes, P a 32-bit permutation and K_i is a 32-bit round key.

In the first section we do differential cryptanalysis of LOKI'91 and show that there is no characteristic with a probability high enough to do a successful differential attack. In the second section we examine the size of the image of the F-function, the round function in LOKI'91. Because the key is added to the input text before the expansion in the F-function, the inputs to the 4 S-boxes are dependent. We show that this has the effect that the size of the image of the F-function is $\frac{8}{13} \times 2^{32}$. In the third section we show a simple relation of LOKI'91 and exploit this in a chosen plaintext attack that reduces an exhaustive key search by almost a factor 4 using $2^{33} + 2$ chosen plaintexts.

6.2.1 Differential cryptanalysis of LOKI'91

In [14] it is indicated that LOKI'91 is resistant against differential cryptanalysis. As stated in Section 5.2 the existence of a r-3-round characteristic in an r-round DES-like cipher with a too high probability might enable a successful differential attack. The difference distribution table for LOKI'91 is a table with 2^{20} entries. Table 6.10 shows the most likely combinations for input/output xors for one S-box in isolation. Note that although input xor

Input	Output	Probability (× 4096)	Input	Output	Probability (× 4096)
004_x	01_x	132	$00c_x$	01_x	76
080_x	04_x	52	$0a0_x$	$e8_x$	46
173_x	$f7_x$	46	185_x	90_x	46
$37b_x$	cd_x	48	$3e0_x$	24_x	48
$42a_x$	41_x	46	498_x	cf_x	56
$49e_x$	97_x	46	790_x	46_x	50
$a20_x$	00_x	46	$a21_x$	$d7_x$	48
$c43_x$	76_x	46	$c76_x$	$f0_x$	48
deb_x	$c9_x$	46	$e7b_x$	$5f_x$	48
$ea6_x$	$5d_x$	46	eec_x	ab_x	46
$f33_x$	$e9_x$	46			

Table 6.10: The most likely combinations from the difference distribution table in hex notation.

004_x leads to output xor 01_x, written $004_x \rightarrow 01_x$, with probability $\frac{132}{4096}$ for one S-box it does not mean that there exists a one round characteristic with this probability. Because the key is added to the input text before the E-expansion in LOKI'91 the inputs to two neighbouring S-boxes are dependent. In the above case a neighbouring S-box will have input xor $4ij_x$ where i, j are some hex digits. The best one-round characteristic with a nonzero input difference has probability $\frac{52}{4096} \simeq 2^{-6.29}$, where for one S-box $080_x \rightarrow 04_x$. Therefore to find a 13-round characteristic with a probability high enough to enable a successful differential attack some of the 13 rounds must be zero rounds. In the following we will use the skeletons for iterative characteristics from Section 5.2.2 without further notice. The best characteristic for an attack on LOKI'89 is based on a 3-round iterative characteristic [14, 46, 48], where every third round is a zero round. The probability of the best 2-round iterative characteristic for LOKI'91 is $\frac{122}{2^{20}} \simeq 2^{-13}$ [14]. In the following we refer to the figures of Section 5.2.2.

Lemma 6.2.1 *The probability of the best 3-round iterative characteristic is* $(\frac{16}{2^{12}})^2 = 2^{16}$.

Proof: Consider the case where Φ and Γ differ in the input to only one

Input	Output	Probability (× 4096)
080_x	80_x	10
040_x	20_x	16
020_x	08_x	6
010_x	02_x	12

Table 6.11: XOR combinations with only inner input bits set.

S-box each, S_i and S_j respectively. Because of the P-permutation it follows easily that the input/output xor combinations for S_i and S_j must have one of the four forms in Table 6.11. The highest probability for a 3-round iterative characteristic is therefore when the combination in both Φ and Γ is $040_x \rightarrow 20_x$. From the difference distribution table it follows easily that if Φ and Γ differ in the inputs to more than 2 S-boxes totally, we obtain probabilities smaller than 2^{-16}. □

Lemma 6.2.2 *The probability for a 4-round iterative characteristic is at most* 2^{-22}.

Proof: By a similar argument as for the 2-round characteristics, we can assume that Φ, Γ and Ψ differ in the inputs to only one S-box each. Obviously Γ and Ψ must differ in the inputs to the same S-box. It means that the combination in round $(i+1)$ must have one of the forms from Table 6.11. The combination in both round (i) and $(i+2)$ must have the following form: $0Y0_x \rightarrow Z$, where $Z \in \{2_x, 8_x, 20_x, 80_x\}$ and $Y \in \{0_x, \ldots\ldots, f_x\}$. The two highest probabilities for $0Y0_x \rightarrow Z$ are $\frac{34}{4096}$ and $\frac{28}{4096}$ found by exhaustive search, therefore the probability of a 4-round iterative characteristic is at most

$$\frac{34 \times 16 \times 28}{2^{36}} < 2^{-22}$$

Note that $\Psi \neq \Gamma$ otherwise round $(i+1)$ must be $\Phi \rightarrow 0$ and γ would have to differ in the inputs to at least two neighbouring S-boxes [14]. □

Now we can prove the following theorem.

Theorem 6.2.1 *A 13-round characteristic for LOKI'91 has a probability*

of at most 2^{-63}.

Proof: The best one-round characteristic with a nonzero input difference has probability $\frac{52}{4096} \simeq 2^{-6.29}$. Because

$$(\frac{52}{4096})^n > 2^{-63} \Rightarrow n \leq 10$$

there must be at least 3 rounds with equal inputs in the 13-round characteristic (13R). Since two consecutive zero-rounds force all rounds to be zero-rounds there can be at most 7 zero-rounds.
7 zero-rounds: There are six 2-round iterative characteristics, therefore

$$P(13R) \leq (2^{-13})^6 = 2^{-78}$$

6 zero-rounds: There are at least three 2-round iterative characteristics, the remaining 6 non-zero rounds have a probability at most $2^{-6.29}$ each, therefore

$$P(13R) \leq (2^{-13})^3 \times (2^{-6.29})^4 = 2^{-64.2}$$

5 zero-rounds: There can be at most one 2-round iterative characteristic, since

$$(2^{-13})^n \times (2^{-6.29})^{8-n} > 2^{-63} \Rightarrow n \leq 1$$

There are two cases to consider

1. No 2-round characteristics, thereby four 3-round characteristics

$$P(13R) \leq (2^{-16})^4 = 2^{-64}$$

2. One 2-round characteristic, thereby at least two 3-round characteristics

$$P(13R) \leq 2^{13} \times (2^{-16})^2 \times (2^{-6.29})^3 = 2^{-63.9}$$

4 zero-rounds: There are no 2-round characteristics, since

$$(2^{-13})^n \times (2^{-6.29})^{9-n} > 2^{-63} \Rightarrow n < 1$$

There can be at most one 3-round characteristic, since

$$(2^{-16})^m \times (2^{-6.29})^{9-2m} > 2^{-63} \Rightarrow m \leq 1$$

There are two cases to consider

1. No 3-round characteristic, thereby three 4-round characteristics:

$$P(13R) \leq (2^{-22})^3 = 2^{-66}$$

2. One 3-round characteristic, thereby at least one 4-round characteristic

$$P(13R) \leq 2^{-16} \times 2^{-22} \times (2^{-6.29})^4 = 2^{-63.2}$$

3 zero-rounds: All 10 nonzero rounds must be based on the best combination $080_x \rightarrow 04_x$, since the second best combination has probability $2^{-6.47}$ and $(2^{-6.29})^9 \times 2^{-6.47} < 2^{-63}$. However it is easy to see that it is not possible to construct a 13-round characteristic based solely on the best combination. □

6.2.2 The F-function of LOKI'91

In the redesign of LOKI'89 [14] one of the guidelines was

- to ensure that there is no way to make all S-boxes give 0 outputs, to increase the ciphers security when used in hashing modes.

004	0049	08e	0d3	514	559	59e	5e3
a24	a69	aae	af3	c03	f34	f79	fbe

Table 6.12: Inputs yielding 0 output for one S-box (hex notation).

The 4 S-boxes in LOKI'91 are identical. Each S-box takes a 12 bit input and produces an 8 bit output. Each output value occurs exactly 16 times. The inputs to one S-box that result in a 0 output are listed in Table 6.12. Because the key is added to the input text before the E-expansion, the input to one

S-box is dependent on the inputs to neighbouring S-boxes. Let the input to one S-box be hij_x, then the input to one of the neighbouring S-boxes is jkl_x. From Table 6.12 we see that to get 0 output from both S-boxes it must hold that $h, j \in \{0, 5, a, c, f\}$ and $j, l \in \{3, 4, 9, e\}$ leaving no possible values for j. Therefore we cannot get 0 outputs from any two neighbouring S-boxes. Let the output from the F-function be $B = \{b_1, b_2, b_3, b_4\}$ where b_i represents the output byte from S-box i. Then $B = \{0, 0, *, *\}$, $B = \{*, 0, 0, *\}$, $B = \{*, *, 0, 0\}$ and $B = \{0, *, *, 0\}$, where '$*$' represents any byte value, are values not in the image of the F-function. We found many similar values and therefore made an exhaustive search for the size of the image of the F-function in LOKI'91.

Theorem 6.2.2 *The F-function is not surjective, indeed the size of the image of F is about* $2^{31.3}$.

Note that once we found that $B = \{b_1, b_2, b_3, b_4\}$, where b_i represents the output byte from S-box i, is not in the image of F, then because the 4 S-boxes in LOKI'91 are equal any rotation of the four bytes yields a value not in the image of F. The size of the image of the F-function is about $\frac{8}{13} \times 2^{32}$. It means that about 5 out of 13 values are never hit in the output of the F-function. In the DES the inputs to the S-boxes in one round are independent, because the key is added after the E-expansion. Therefore the size of the image of the F-function in the DES is 2^{32}. A consequence of the observation is that for LOKI'91 the left and right halves of a ciphertext reveals 0.7 bit of information about the inputs (before addition of the keys) to the second last round respectively the third last round of the encryption. It is an open question how to exploit this observation in an attack on LOKI'91.

6.2.3 A chosen plaintext attack reducing key search

We begin by giving the notation used in this section.
Notation:

- $Rol_n(X)$ is bitwise rotation of X n positions to the left.
- $E_{16}(P, K)$ is a full 16 round encryption of P using K.
- $E_2(P, K')$ is a 2 round encryption of P using the 32 bit key K' in the

first round and $Rol_{13}(K')$ in the second round.

- $Swap(X, Y)$ is the swapping of X and Y.
- $Swap(Z)$ is the swapping of the left and right halves of Z.

The attack we are to describe makes use of a property of the key schedule in LOKI'91. The key size is 64 bits. The key is divided into two 32 bit halves K_L, K_R and the 16 round keys $K(i)$, $i = 1, \ldots, 16$, are derived as follows:

1. $i = 1$
2. $K(i) = K_L;\ i = i + 1$
3. $K_L = Rol_{13}(K_L)$
4. $K(i) = K_L;\ i = i + 1$
5. $K_L = Rol_{12}(K_L)$
6. $Swap(K_L, K_R)$
7. go to 2.

The key schedule allows two different keys to have several round keys in common.

Theorem 6.2.3 *For every key K there exists a key K^*, such that*

$$K(2 + i) = K^*(i), i = 1, \ldots, 14$$

i.e., K and K^* have 14 mmd keys in common.

Proof: Let $K(1), \ldots, K(16)$ be the round keys for $K = K_L \parallel K_R$. Let $K^* = K_L \parallel Rol_{25}(K_L)$. Then $K(2 + i) = K^*(i)$, $i = 1, \ldots, 14$. □

For exactly two keys, the all zero and the all one key, the two keys K and K^* in the construction in the above proof are the same key. But these keys are the only ones for which that happens, because $K = K^* \Rightarrow (K_L = K_R) \wedge (K_R = Rol_{25}(K_L)) \Rightarrow K_R = K_L = Rol_{25}(K_L) \Rightarrow K = 00\ldots.00 \vee K = 11\ldots.11$, since $gcd(25, 32) = 1$.

Corollary 6.2.1 *There exists 2^{36} pairs of keys, K and K^*, such that K and K^* have 16 round keys in common.*

Proof: Let $K = K_L \parallel K_R$, $K_L = hh \dots hh_x$ for some hex digit h and let K_R be any 32 bits. From Theorem 6.2.3 it follows that there exists a key K^* such that K and K^* have 14 round keys in common and furthermore $K^*(15) = Rol_{100}(K_L) = K_L = K(1)$ and $K^*(16) = Rol_{113}(K_L) = Rol_{13}(K_L) = K(2)$, i.e. K and K^* have 16 round keys in common. □

Theorem 6.2.3 can be used in a chosen plaintext attack to reduce an exhaustive key search by almost a factor 2. The complementation property, see page 96, holds also for LOKI'91. This property and Theorem 6.2.3 can be used to reduce an exhaustive key search by almost a factor 4 in a chosen plaintext attack that needs $2^{33} + 2$ plaintexts. Note that the above phenomenon is what we call a simple relation, cf. Section 5.4.2. The algorithm is similar to the general one given in Section 5.4.2. We state here the exact algorithm for the attack on LOKI'91.

Algorithm:

1. Pick $P = P_L \parallel P_R$ a random. Get encryptions C, C^* for P, $\overline{P}$.
2. For all $a \in \{0, 1, \dots, (2^{32} - 1)\}$:
   Let $P(a)$ be $E_2(P, a)$. More precisely $P(a) = P_L(a) \parallel P_R(a)$, where
   $$P_L(a) = F(P_R, a) \oplus P_L$$
   $$P_R(a) = F(P_L(a), Rol_{13}(a)) \oplus P_R.$$
3. Get encryptions $C(a)$, $C^*(a)$ for $P(a)$, $\overline{P(a)}$ for all a.
4. Let all keys be non discarded.
5. Exhaustive search for key:
   For every non discarded key $K = K_L \parallel K_R$, do
   (a) Find $C' = E_{16}(P, K)$
   (b) Then

Estimates for	Time	Space	Chosen plaintexts
	1.07×2^{62}	$2^{33} + 2$	$2^{33} + 2$

Table 6.13: Complexity of the chosen plaintext attack on LOKI'91.

- if $C' = C$ return K and stop
- if $C' = \overline{C^*}$ return $\overline{K}$ and stop
- if $E_2(Swap(C'), Rol_{100}(K_L)) = C(K_L)$ return $(K_R \parallel Rol_{25}(K_L))$ and stop
- if $E_2(Swap(C'), Rol_{100}(K_L)) = \overline{C^*(K_L)}$ return $(\overline{K_R} \parallel Rol_{25}(\overline{K_L})$ and stop

(c) Discard the four keys in (b).

Upon termination we have found either the secret key or a collision for LOKI'91, i.e., $K \neq K^*$, such that $E_{16}(P, K) = E_{16}(P, K^*)$. Note that in step 5, once we have encrypted P using key $K = K_L \parallel K_R$ without success, we do not have to encrypt P using neither $\overline{K}$, $(K_R \parallel Rol_{25}(K_L))$ nor $(\overline{K_R} \parallel Rol_{25}(\overline{K_L}))$. If one of these three keys is the secret key, then the algorithm would have terminated earlier. At some points in the algorithm some of the four keys in 5(b) are equal, for example the all zero key will appear twice in the same iteration of step 5. Therefore we cannot find an enumeration of the keys in step 5, s.t. the total number of iterations of step 5 is exactly one quarter of the size of the key space, i.e., 2^{62}. There exists however an enumeration, s.t. the number of iterations of step 5 is about $2^{62} + 2^{48}$. This is proved in Appendix B.2 in every glory detail. Table 6.13 shows the estimates for space, time and number of chosen plaintexts for the attack, where one time unit is a full 16-round encryption and one space unit is 64 bits. The estimate for *Time* is the number of encryptions made in the analysis. In every iteration of step 5 we do one full 16.round encryption in 5(a). For the two last tests in step 5(b) we do at most 2 rounds of encryption. For most iterations however, we need only to do one round of encryption, because we can test for equality of the right halves of $E_2(Swap(C'), Rol_{100}(K_L))$ and $C(K_L)$ (resp. $\overline{C^*(K_L)}$) already after one round of encryption of $Swap(C')$. If the tests fail we need not do the second round of encryption. Therefore for only about one out of 2^{31} iterations we need to do two rounds of encryption

in 5(b). The total amount of time therefore is

$$(2^{62} + 2^{48}) \times \frac{17}{16} + (\frac{2^{62} + 2^{48}}{2^{31}}) \times \frac{1}{16} \simeq 1.07 \times 2^{62}.$$

Compared to this the time used in step 2 is negligible. The above attack is a weak attack. First of all, it is not very likely that we can get the encryptions of $2^{33} + 2$ chosen plaintexts, furthermore an improvement of a factor four of an exhaustive search for the key is not much. The LOKI cipher is meant as an alternative to DES, with which it is interface compatible. The steps 2, 3 and 5 can be carried out in parallel, for instance by letting $K_L = a$ in step 5, in that way we don't have to store the $2^{32} C(a), C^*(a)$'s in step 3. It seems impossible however to obtain an enumeration that at the same time makes the total number of iterations of step 5 be close to 2^{62} and enables a parallel run of the algorithm.

6.2.4 Weak hash keys for LOKI'89 and LOKI'91

For the LOKI ciphers the keys are added before the expansion and the following result holds for the LOKI ciphers.

Theorem 6.2.4 (LOKI) *If $K(i) \oplus K(17 - i) = \sigma$ for all $i \in \{1, \ldots, 16\}$ then K is a weak hash key and (5.11), page 92, holds with probability 2^{-32} over all plaintexts.*

Note that the inputs to the eighth and ninth round uniquely determine both the plaintext and ciphertext and that the difference will be σ for exactly 2^{32} plaintexts. Also note that although equation (5.11) holds with probability only 2^{-32} for the above keys the plain- and ciphertexts, for which (5.11) holds, can be found using only half an encryption, when the key is known. It is clear that once we have found a weak hash key for the LOKI's (or DES), the complemented key is also a weak hash key.

Corollary 6.2.2 *For LOKI'89 there are at least 2^{16} weak hash keys.*

Proof: It follows from the key schedule of LOKI'89, that the keys $K = K_L \parallel K_R$, where

$$K_L = vwyzvwyz_x \text{ and } K_R = VWYZVWYZ_x$$

where $v \oplus w \oplus y = z$ and $V \oplus W \oplus Y = Z$ and $v \oplus V = w \oplus W = y \oplus Y$, satisfy the condition in Theorem 6.2.4. The key $K = K_L \parallel K_R$ is added (modulo 2) to the plaintext and the 'swapped' key $(K_R \parallel K_L)$ is added to the ciphertext [15]. The xor of the plaintext and the ciphertext for LOKI'89 (δ in (5.11)) is $\sigma \oplus c \parallel \sigma \oplus c$, where $c = K_L \oplus K_R$. □

Corollary 6.2.3 *For LOKI'91 there are at lead* 16 *weak hash keys.*

Proof: Let h be a hex digit, $h \in \{0, 3, 5, 6, 9, A, C, F\}$. From the key schedule of LOKI'91 it follows that the keys $K = K_L \parallel K_R$, where $K_L = hhhhhhhh_x$ and $K_R = Rol_3(K_L)$ or $K_R = Rol_3(\overline{K_L})$ are weak hash keys. □

Eight of these keys are also either weak or semi-weak [14], but the other eight are neither weak nor semi-weak.

6.2.5 Conclusion and open problems

We have shown that we cannot find a characteristic for LOKI'91 good enough to do a successful differential attack on LOKI'91. Still it is not enough to conclude that LOKI'91 is secure against this kind of attack. To do that we need an efficient way of calculating the probabilities of differentials.

We have shown that the size of the image of the F-function in LOKI'91 is only $\frac{8}{13}$ of the size of the image of the F-function in DES. This is a weakness, since it means that the ciphertext reveals information about the inputs to the second and third last rounds. Whether it represents a serious weakness for the algorithm is left as an open question.

We introduced a chosen plaintext attack on LOKI'91 that reduces an exhaustive key search by almost a factor 4. The attack exploits a simple relation based on a weakness in the key schedule of LOKI'91. It might also be possible to use this simple relation to find collisions for LOKI'91 when used in a hash function. This is left as an open question.

Finally we showed that there are many weak hash keys for LOKI'89 and a few weak hash keys for LOKI'91.

6.3 s^2-DES

After Biham and Shamir's differential attack on the full 16-round DES research has been going on to try to reconstruct the DES to withstand this kind of attack. There has been a huge volume of research on DES, since its publication in the mid 70's. Some of this work has been concentrating on the design of secure S-boxes. In [43] Kwangjo Kim provides a way of constructing DES-like S-boxes based on boolean functions satisfying the SAC (Strict Avalanche Criterion). Kim lists 5 criteria for the constructions, including "Resistance against differential attacks". Furthermore 8 concrete examples of these S-boxes, the s^2-DES S-boxes, are listed. The cryptosystem s^2-DES is obtained by replacing all the 8 DES S-boxes by the 8 s^2-DES S-boxes, keeping everything else as in DES. It is suggested that s^2-DES withstands differential attacks better than DES. We show that this is indeed not the case. The conclusion is that Kim's 5 criteria for the construction of DES-like S-boxes are insufficient to assure resistance against differential attacks.

In differential cryptanalysis of the s^2-DES the difference of two bit strings are defined as the bitwise exclusive-or of the strings. Kim's s^2-DES S-boxes do not have the DES properties 2, 4 and 5. They do have a property though that is part of property 2 for the DES S-boxes.

4a. $\mathrm{S}(\mathbf{x}) \neq \mathrm{S}(\mathbf{x} \oplus (a0000b))$ for $ab \neq 00$.

Since the s^2-DES S-boxes are built as 4 rows of 40 bit bijective functions, they have property 6 like the DES S-boxes.

2-round characteristics

Because of property 6 there is no 2-round iterative characteristic for Kim's s^2-DES S-boxes where the inputs differ only in one S-box, however the lack of property 5 enables us to build a 2-round iterative characteristic where the inputs differ in two neighbouring S-boxes. We have

$$0_x \leftarrow 00000580_x \text{ with prob. } \frac{8\times 10}{64\times 64} \simeq \frac{1}{51}$$

Extending this characteristic to 15 rounds yields a probability of $2^{-39.7}$. Using the original attack by Biham and Shamir [5] we will need about 2^{42} chosen

plaintexts for a successful differential attack. To do a similar attack as by Biham and Shamir in [7] we construct a 13-round characteristic with probability 2^{-34}. The structures of plaintexts, see [7], used in the attack will consist of 2^9 plaintexts and we will need a total of about 2^{35} chosen plaintexts for the attack. The above characteristic is not the only 2-round iterative characteristic for s^2-DES that is better than the best 2-round iterative characteristics for DES. There are several others, the two second best characteristics both with probability $\frac{6\times10}{64\times64} \simeq \frac{1}{68}$ are based on the combinations: $0_x \leftarrow 07e00000_x$ and $0_x \leftarrow 5c000000_x$.

3-round characteristics

We proceed trying to find a better characteristic than the ones we have already found. The best non-trivial input/output xor combination in s^2-DES has probability $\frac{1}{4}$. Therefore there can be at most 4 S-boxes with different inputs in the 3 rounds all together, as $(\frac{1}{4})^x \geq (\frac{1}{50})^{1.5} \Rightarrow x < 5$. As with DES, because of the P-permutation, Φ and Γ must differ in the inputs to at least two S-boxes each. Unlike for the DES it is possible for two inputs different in only 1 bit to result in two outputs different in 1 bit. Therefore we can build a 3-round characteristic with $\Phi = 04040000_x$ and $\Gamma = 00404000_x$. The probability for the characteristic is $\frac{8\times6\times4\times10}{64^4} \simeq 2^{-13.5}$. This is the best 3-round characteristic we have found for s^2-DES. We can build a 13-round characteristic to be used as in the attack in [7]. The probability for the characteristic is $2^{-52.5}$. However, we can use the combinations from the 3-round characteristic to build 6-round iterative characteristics, which are better, as we will show later.

4-round characteristics

There can be at most 5 S-boxes with different inputs, because $(\frac{1}{4})^x \geq (\frac{1}{51})^2 \Rightarrow x < 6$, and again we exploit the fact that s^2-DES S-boxes do not have property 2. We construct a 4-round characteristic based on the following combinations:

$$\begin{aligned}
&00000002_x \leftarrow 0000006\mathrm{e}_x \text{ withprob. } \tfrac{8\times10}{64\times64}\\
&00080000_x \leftarrow 00020000_x \text{ withprob. } \tfrac{8}{64}\\
&00000002_x \leftarrow 0000002\mathrm{e}_x \text{ withprob. } \tfrac{6\times10}{64\times64}
\end{aligned}$$

We have $P(00000002_x) = 00020000^x$ and $P(00080000)_x = 00000040_x = 0000006e_x \oplus 0000002e_x$. The total probability for the 4round characteristic is $2^{-14.77}$. Extended to 13 rounds we obtain a probability of $2^{-44.3}$.

Longer characteristics

A 5-round iterative characteristic will have to differ in the inputs to at least 6 S-boxes. However, we can find 6-round iterative characteristics also different in the inputs to only 6 S-boxes as indicated above. The P-permutation makes it impossible to find $\Phi \rightarrow \Gamma$ and $\Gamma \rightarrow \Phi$, where both Φ and Γ differ only in the inputs to one S-box. However, it is possible to find Φ, Γ, Ψ and Ω, all four different only in the input to one S-box and such that $\Phi \rightarrow \Gamma, \Gamma \rightarrow \Psi, \Psi \rightarrow \Omega$ and $\Omega \rightarrow \Phi$. We use this observation to construct a 6-round characteristic:

$$
\begin{array}{ccccc}
 & & (\Phi, 0) & & \\
0 & \leftarrow & 0 & & \text{prob. 1} \\
\Gamma & \leftarrow & \Phi & & \text{some prob.} \\
\Psi & \leftarrow & \Gamma & & \text{some prob.} \\
\Gamma \oplus \Omega & \leftarrow & \Phi \oplus \Psi & & \text{some prob.} \\
\Phi & \leftarrow & \Omega & & \text{some prob.} \\
\Omega & \leftarrow & \Psi & & \text{some prob.} \\
 & & (\Psi, 0) & &
\end{array}
$$

With $\Phi = 04000000_x$, $\Gamma = 00004000_x$, $\Psi = 00040000_x$ and $\Omega = 00400000_x$ we get a total probability for the 6-round characteristic of $\frac{8\times10\times8\times6\times4\times6}{64^6} \simeq 2^{-19.5}$. Extended to 13 rounds the probability becomes 2^{-39}. Starting with $(\Gamma, 0)$ we get a similar 6-round characteristic with probability $2^{-19.5}$. Starting with $(\Psi, 0)$ or $(\Omega, 0)$ yield 6-round characteristics with probibility $2^{-19.8}$. These 6-round characteristics differ in the inputs to 6 S-boxes, that is, different inputs to one S-box per round on the average. In the construction of n-round iterative characteristics, $n > 6$, one will get more than one S-box difference per round on the average.

Conclusion on Kim's s^2-DES S-boxes.

The above illustrates that one has to ensure that DES-like S-boxes have the six properties of the DES S-boxes as listed in Section 6.1. The fact that

for s^2-DES two inputs different only in the inputs to two neighbouring S-boxes can result in equal outputs enabled us to build a 2-round iterative characteristic more than 4 times as good as the best 2-round characteristic for DES, thus enabling a differential attack on s^2-DES about 2^{13} times better than Biham and Shamir's best attack on DES. The fact that two S-box inputs different in only one bit can result in outputs different in one bit enabled us to construct 4-round and 6-round iterative characteristics both better for differential attacks on s^2-DES than the 2-round characteristic for DES.

6.4 s^3-DES

As a consequence of our differential attacks on s^2-DES, a new method of constructing DES-like S-boxes was proposed in [45]. Furthermore 8 concrete examples of these S-boxes, the s^3-DES S-boxes, are listed. The cryptosystem s^3-DES is obtained by replacing the 8 DES S-boxes by the 8 s^3-DES S-boxes, keeping everything else as in DES. These S-boxes have the property that they prevent the construction of good 2-round iterative characteristics, because to have equal outputs of one round the inputs to all 8 S-boxes in a pair must be different. In the difference distribution table for s^3-DES there are 15 entries with probability $\frac{20}{64}$ and 22 with probability $\frac{18}{64}$. For comparison, for the DES the highest probability for a non-trivial entry is $\frac{16}{64}$, which does not necessarily mean that we can find better characteristic for s^3-DES than for DES.

Iterative characteristics

It is easy to show that we cannot find a 2-round or 3-round characteristic for s^3-DES that is better than the 2-round iterative characteristic used by Biham and Shamir in their attack on DES. We did an ad-hoc search for a 4-round iterative characteristics and found one based on the following combinations.

$$\begin{aligned}
00000054_x &\leftarrow 070a0000_x \text{ with prob. } 2^{-7.09} \\
02100000_x &\leftarrow 00000054_x \text{ with prob. } 2^{-6.19} \\
00000054_x &\leftarrow 051a0000_x \text{ with prob. } 2^{-7.09}
\end{aligned}$$

That is, for 4 rounds we obtain a probability of $2^{-22.16}$. To do an attack similar to the one by Biham and Shamir in [7], we need a 13-round characteristic.

Using the 4-round iterative characteristic we obtain a probability of $2^{-66.50}$, which is too low for a differential attack to be successful. However, it is almost 2^{30} times better than the estimates made in [45]. This illustrates once again that we have to be careful in concluding resistance against differential cryptanalysis. For certain key bits the above 4-round characteristic will have a probability of $2^{-19.5}$ and for the 13-round characteristic $2^{-58.5}$. That is for certain keys the complexity of a differential attack is not far away from the exhaustive key search border of 2^{56}.

We conclude that the analysis in [45] is insufficient to conclude resistance against differential attacks. There exists a 4-round characteristic that extended to 13 rounds yields a much better probability than the estimates made in [45]. We stress that our analysis has not been an exhaustive search for the best characteristics and note that a new version of [45] contains our above analysis [44]. Although it seems like the s^3-DES S-boxes resist differential attacks better than the DES, we will show in Section 8.3 that in differential attacks on hash functions based on block cipher, there exist characteristics for s^3-DES much better than for the DES.

6.5 xDES^i

In [115, 113] Zheng, Matsumoto and Imai used the DES as a building block to build a series of block ciphers, called xDES^i, based on the work by Luby and Rackoff [61] and on a theory of the construction of secure block ciphers developed in [114, 113]. The block ciphers are mainly used as building blocks of hash functions, but it is noted that they can be used for encryption also [115, 113]. xDES^0 is just the DES and for $i > 0$,

$$xDES^i \ : \ GF(2)^{56i(2i+1)} \times GF(2)^{128i} \rightarrow GF(2)^{128i}$$

i.e., a $128 \times i$ bit block cipher with a $56 \times i \times (2i+1)$ bit key. We consider first xDES^1, a 128 bit block cipher using a 168 bit key. xDES^1 is defined as a three round Feistel-cipher, where in each round the DES is used as the f-function and where all three 56 bit round keys $K_i, i = 1, \ldots, 3$, are independent. It is noted in [115] that any secure block cipher may be used instead of the DES in the construction.

It is clear that xDES^1 is faster than conventional triple encryption, cf. Section 7.9, since the plaintext blocks are twice as large. However, it follows

that there is a simple meet in the middle attack on $x\text{DES}^1$, which given the encryptions of a few known plaintexts finds the secret key using a table of size about 2^k, where k is the key size of the underlying block cipher. Simply encrypt the right half of a plaintext with all possible values of K_1 exclusive-or the left half of the plaintext and store these 2^k values and exclusive-or the left half of the ciphertext to all entries in the table. Then encrypt the right half of the ciphertext with all possible values of K_3 and look for a match in the table. If a match is found, the values of the pair K_1, K_3 are possible candidates for the right key. By repeating the attack a few times, only one pair of values will remain. This attack is independent of the underlying block cipher and of how the three round keys are computed. Although the memory requirements are quite large, i.e., 2^{56} when using the DES as the basic block cipher, it is clear that $x\text{DES}^1$ is not an optimal solution. Furthermore we show in the following that in a chosen plaintext attack using only negligible memory, $x\text{DES}^1$ is not much stronger that the underlying block cipher.

6.5.1 A chosen plaintext attack on $x\text{DES}^1$

We introduce some notation. I_j and O_j are defined to be the input to and output from the f-function in the j'th round. Now we can prove the following result:

Theorem 6.5.1 *Let $\mathcal{Y}$ be a 3-round $2m$-bit Feistel cipher using the m-bit cipher $\mathcal{X}$ as the round function with 3 independent round keys, RK_i $i = 1, 2, 3$. Assume that exhaustive key search of $\mathcal{X}$ takes time 2^t using one known plaintext. Then there exists an attack that finds the key of $\mathcal{Y}$ in time 2^{t+2} using one chosen plaintext and one known plaintext.*

Proof: Let $P = P_L \mid P_R$ be a known plaintext and $C = C_L \mid C_R$ the corresponding ciphertext. Define $P' = P'_L \mid P'_R$, where P'_L is a random m-bit value and $P'_R = P_R$ and denote by $C' = C'_L \mid C'_R$ the corresponding ciphertext. The intermediate values, the I's and O's, are also primed for this encryption. Now calculate

$$\begin{aligned} O_3 \oplus O'_3 &= P_L \oplus P'_L \oplus O_1 \oplus O'_1 \oplus C_L \oplus C'_L \\ &= P_L \oplus P'_L \oplus C_L \oplus C'_L \end{aligned}$$

since $O_1 = O'_1$.

Since I_3 and I_3' can be read as the right halves of the ciphertexts, the third round key can be found by exhaustive search in time (about) 2×2^t. With the knowledge of RK_3, the ciphertext C (and/or C') can be decrypted one round obtaining the encrypted values after two rounds of encryption, which means that I_2 is now known. Since $O_2 = P_R \oplus C_R$, an exhaustive search finds the second round key, RK_2 in time 2^t. Similarly, $I_1 = P_R$ and $O_1 = P_L \oplus I_2$, and an exhaustive search finds RK_1 also in time 2^t. Totally the attack takes time 4×2^t. □

Corollary 6.5.1 *There exists an attack on* $x\text{DES}^1$ *which on input one known plaintext and one chosen plaintext attack finds the secret key in time about* 2^{58}.

We conclude that $x\text{DES}^1$ is not a candidate for an extended version of DES. The security gained is too small compared to the large increase in the key. Note that the chosen plaintext attack also work for dependent round keys. It is clear from our attacks that the major weakness in $x\text{DES}^1$ is the use of only three rounds.

It should be noted that our attacks are not contradictory with the results by Luby and Rackoff [61], which say that if $\mathcal{X}$ (from Theorem 6.5.1) is a pseudorandom function then $\mathcal{Y}$ is a pseudorandom permutation. Their result says nothing about the security connection between $\mathcal{X}$ and $\mathcal{Y}$, merely that $\mathcal{Y}$ is secure against attacks by polynomial time running algorithms, if $\mathcal{X}$ is.

6.5.2 A differential attack on $x\text{DES}^2$

In this section we consider $x\text{DES}^2$ which has a 560 bit key working on 256 bit blocks. $x\text{DES}^2$ can be seen as two five round Feistel ciphers, where in each of the ten rounds the DES is used as the f-function and where the outputs of each round are mixed, see Figure 6.3. In [115] it is assumed that all ten 56 bit round keys are independent and noted that any secure block cipher may be used instead of DES. We will assume in the following that the DES is used. The 256 bit plaintext is divided into four block of 64 bits each, $c_{0,1}, c_{0,2}, c_{0,3},$

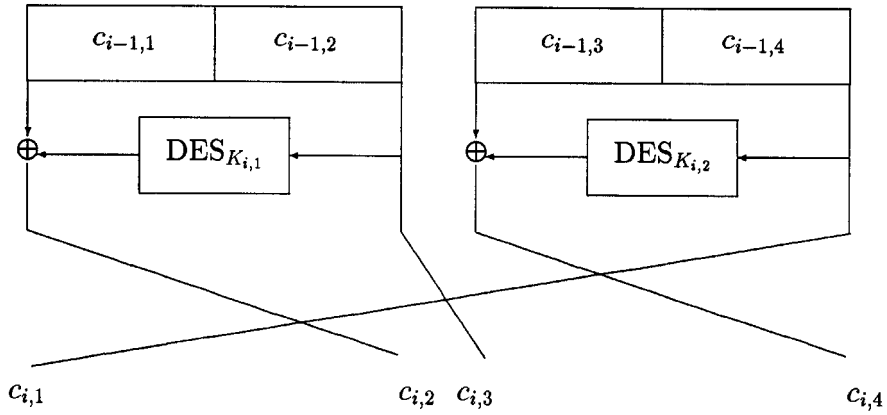


Figure 6.3: One round of $x\mathrm{DES}^2$

and $c_{0,4}$. In general the computation of the ciphertext is as follows.

$$\begin{aligned} c_{i,1} &= c_{i-1,4} \\ c_{i,2} &= c_{i-1,1} \oplus DES_{K_{i,1}}(c_{i-1,2}) \\ c_{i,3} &= c_{i-1,2} \\ c_{i,4} &= c_{i-1,3} \oplus DES_{K_{i,2}}(c_{i-1,4}) \end{aligned}$$

where $K_{i,1}$ and $K_{i,2}$ for $i = 1, \ldots, 5$ are ten 56 bit round keys. After five rounds of encryption the ciphertext is defined as the concatenation of $c_{5,4}, c_{5,3}, c_{5,2}$, and $c_{5,1}$. Note that the final permutation of these ciphertext blocks is not the same as the permutation of the four intermediate ciphertext blocks in the round function. The final permutation of the ciphertext block has no influence on our attacks, so in the following we will assume that permutation of blocks are the same in all rounds. We can prove the following result.

Theorem 6.5.2 *There exists an attack on* $x\mathrm{DES}^2$, *which on input about* 2^{33} *known plaintext finds the secret 560 bit key in time* $O(2^{64})$.

Proof: First we describe a chosen plaintext attack. We define the following characteristic $\Delta P = (0 \mid 0 \mid \Gamma \mid \Phi)$ for some values of Γ and Φ. If a difference Φ in the inputs to the rightmost DES-encryption of Figure 6.3 lead to outputs with difference Γ in the first round, the difference $\Delta C_i = c_i \oplus c'_i$ in the values $c_i = (c_{i,1} \mid c_{i,2} \mid c_{i,3} \mid c_{i,4})$ and $c'_i = (c'_{i,1} \mid c'_{i,2} \mid c'_{i,3} \mid c'_{i,4})$ for $i = 0, \ldots, 5$

of the two encryptions are depicted in Table 6.14 where for $i = 1, \ldots, 4$, X_i are values we cannot predict. The attack now proceeds as follows

$\Delta C_0 = 0$	$\mid$	0	$\mid$	Γ	$\mid$	Φ
$\Delta C_1 = \Phi$	$\mid$	0	$\mid$	0	$\mid$	0
$\Delta C_2 = 0$	$\mid$	Φ	$\mid$	0	$\mid$	0
$\Delta C_3 = 0$	$\mid$	X_1	$\mid$	Φ	$\mid$	0
$\Delta C_4 = 0$	$\mid$	X_2	$\mid$	X_1	$\mid$	Φ
$\Delta C_5 = \Phi$	$\mid$	X_3	$\mid$	X_2	$\mid$	X_4

Table 6.14: A five round characteristic for $x\text{DES}^2$.

1. Choose a plaintext pair with the desired difference $\Delta P = (0 \mid 0 \mid \Gamma \mid \Phi)$.
2. Get the corresponding encryptions in a chosen plaintext attack.
3. If the difference of the leftmost 64 bit ciphertext blocks $c_{5,1}$ is Φ, try for all possible values of the key $K_{1,2}$ if the encryptions of the two corresponding 64 bit plaintext blocks $c_{0,4}$ yield a difference Γ and store the keys for which that holds.
4. If the difference of the leftmost 64 bit ciphertext blocks $c_{5,1}$ is not Φ, go to step 1.

In all pairs of plaintexts we choose the same values in the fourth plaintext blocks with difference Φ. Since there are 2^{64} possible values of the exclusive-or of the outputs of the rightmost DES encryption, we let the third plaintext blocks run through all possibilities, i.e., the input differences we use are $\Delta P_i = (0 \mid 0 \mid \Gamma_i \mid \Phi)$ for $i = 0, \ldots, 2^{64} - 1$. After 2^{64} trials with different pairs of plaintext blocks, we are guaranteed that for at least one pair the condition for an exhaustive search in step 3 will be true and we will find the right value of $K_{1,2}$. Also we will get wrong pairs that by accident hit the difference Φ in leftmost ciphertext blocks, but by repeating the attack a few times, only the right key value of $K_{1,2}$ will be left suggested. Also we can use the knowledge in ΔC_4 above to search for the key $K_{5,1}$. For a pair satisfying the condition in step 3 it holds that encryptions of the ciphertext blocks with difference X_2 yield a difference of X_3 after encryption with $K_{5,1}$, since the xor'ed difference from the fourth round is zero. Also we can use a

similar characteristic to attack the leftmost DES encryption in the first round and find $K_{1,1}$ and $K_{5,2}$ in a similar way as above. Then we can decrypt all ciphertexts one round and repeat the attack on a 4-round version of $x\text{DES}^2$, where our probability of success will be higher.

As shown by Matsui [67] the best characteristic for an attack on the DES is the concatenation of the 2-round iterative characteristic. For the full 16-round DES this characteristic will have an average probability of $(\frac{1}{234})^8 \simeq 2^{-63}$. Using this characteristic we fix the values for Φ and Γ above iId we need to try about 2^{63} pairs of plaintexts to get one right pair. However, we can do better than that. We assume that for a random pair of plaintexts, the output difference is distributed uniformly for the DES. That seems a reasonable assumption in general for a good block cipher, and especially for the DES, where only two characteristics have a probability below the trivial one of 2^{-64}. Therefore for a randomly chosen pair of plaintexts the probability that the condition in step 3 is satisfied is about 2^{-64}. In a collection of 2^{32} plaintexts we can form about 2^{64} pairs of plaintexts and with a high probability we get at least one right pair. We note that, first of all, these characteristics are dependent in some way and the success of some characteristic may depend on the success of other characteristics and secondly, since the DES with a fixed key is a permutation, pairs with $\Gamma = 0$ will never be a right pair. However, by using more plaintexts, i.e., 2^{33} or 2^{34}, we can expect to get a right pair. □

Finally we note that our attacks can also be applied when $x\text{DES}^2$ is used with dependent round keys.

For $i \geq 3$, $x\text{DES}^i$ is probably too big for the cipher to be a serious candidate-for a block cipher. As an example, for $i = 3$ the block size is 384 bits and the key size is 1176 bits and 21 encryptions of the DES are needed to encrypt one block of size six times a DES block.

Chapter 7

Design of Block Ciphers

In this chapter we discuss some of the problems involved in the design of a block cipher. In Section 7.1 the danger of focusing solely on a few design criteria is discussed and a set of necessary design principles is listed. In Section 7.2 we discuss the block and key sizes in block ciphers. In Section 7.3 it is shown how to obtain provable security against a differential attack. In Section 7.4 the Markov theory for block ciphers is considered and it is shown that with a high probability all iterated block ciphers will be resistant against differential attacks after sufficiently many rounds. In Section 7.5 it is shown how to obtain provable security against a linear attack. In Section 7.7 we define and show how to build *strong* key schedules. In Section 7.8 we give a new test for checking the nonlinear order of a block cipher. Finally in Section 7.9 multiple encryption of a block cipher is considered. We give a new proposal for a triple encryption scheme, which under reasonable assumptions is as secure as the underlying scheme though requiring only a minimum number of component keys.

7.1 Design Principles

Two generally accepted design principles for practical ciphers are the principles of confusion and diffusion that were suggested by Shannon. In his own words: *"The method of confusion is to make the relation between the simple statistics of the ciphertext and the simple description of the key a very com-*

1. Confusion
2. Diffusion
3. Sufficiently large block size
4. Sufficiently large key size
5. Resistance against known attacks
   (a) - differential attacks
   (b) - linear attacks
6. All keys are equally good
7. No simple relations
8. High nonlinear order

Table 7.1: Special design principles for block ciphers.

plex and involved one" [107].
"In the method of diffusion the statistical structure of the plaintext which leads to its redundancy is dissipated into long range statistics" [107]. Massey [63] interprets Shannon's concepts of confusion and diffusion as follows

> **Confusion**
> *The ciphertext statistics should depend on the plaintext statistics in a manner too complicated to be exploited by the cryptanalyst.*
>
> **Diffusion**
> *Each digit of the plaintext and each digit of the secret key should influence many digits of the ciphertext.*

Shannon's design principles are very general and informal. There have been many suggestions in the past of how to obtain good properties (diffusion/ confusion) for a block cipher, but so far there is no known exact recipe of how to construct a secure block cipher. A necessary and obvious requirement is that the cipher is resistant against all known attacks. In Table 7.1 we list

more specific design principles for the design of block ciphers. We stress that a cryptographic design principle should not be over-interpreted. Design principles should be seen as "guidelines" in the construction of ciphers, evolved from years of experience, and as necessary, but **not** sufficient requirements. There are many examples of this in the history of cryptography. We give a few of the most recent examples.

Consider SP (substitution-permutation)-networks, product ciphers, where the ciphertext is computed from the plaintext by applying in turn (key-dependent) substitutions and permutations to the plaintexts. The DES can be seen as a special implementation of a SP-network. In [41] a method of constructing SP-networks is given, where for every key all ciphertext bits depend on all plaintext bits. However, this fact is information that an attacker can exploit. In [88] O'Connor describes a differential attack on the SP-networks of [41] using a number of chosen plaintexts linear in the number of S-boxes in the network.

In [62] the group properties of a cryptosystem based on permutation groups were studied. It was claimed that the ability of a system to generate the symmetric group on the message space is "one of the strongest security conditions that can be offered". In [81] an example of a cipher was given, that generates the symmetric group, but still is a weak cipher vulnerable to a known plaintext at tack.

7.2 Sufficiently Large Block and Key Size

It is clear from the discussion in Section 4.2 that if either the block or key size is too small or both, a block cipher is vulnerable to a brute force attack. These attacks are independent of the internal structure and intrinsic properties of an algorithm. Most block ciphers, e.g. DES, IDEA, FEAL, LOKI, have a block size of 64 bits. For these ciphers the birthday attacks of Theorems 4.4.1, 4.4.2 and 4.4.3 require storage/collection of 2^{32} ciphertext blocks for a success of about one half. These are not realistic attacks. First of all, no single key is likely to be used to process that many ciphertexts, second storage of 2^{32} ciphertext blocks of each 64 bits will require about 2^{15} Mbytes of memory. Still, if an attacker can predict approximately how frequently a key is changed, he can repeat his attack several times with fewer ciphertext blocks and get a non-negligible probability of success. This should be taken

into consideration, when designing new block ciphers.

The key size of the DES is only 56 bits, which is too short as discussed in Section 6.1.6. Some of the latest block cipher proposals have a larger key, e.g. IDEA [55] has a key size of 128 bits. In [11] Denning et al. estimated that a block cipher with a key size of 80 bits is not vulnerable to an exhaustive search within the next 30-40 years. The fastest exhaustive search machine on the DES is the one by Wiener [112], which at the cost of 1 million US$ finds the secret key of the DES in average time 3.5 hours. Using this estimate it would take about 6,700 years to break a block cipher with a 80 bit key.

7.3 Resistance Against Differential Attacks

We consider an r-round iterated block cipher with round function g. Denote by p_g the highest probability of a non-trivial one-round differential achievable by the cryptanalyst.

$$p_g = \max_{\beta} \max_{\alpha \neq 0} \Pr_K(\Delta C_1 = \beta \mid \Delta P = \alpha) \tag{7.1}$$

where the probabilities are taken over all possible keys. In the following we will omit the subscript of the probabilities. In Section 5.2.3 the probability of a differential is given (5.5). It is easy to obtain a lower bound of any differential in an r-round iterated cipher expressed in terms of p_g.

Theorem 7.3.1 *Consider an r-round iterated cipher with independent round keys. Any s-round differential, $s \geq 1$, has a probability of at most p_g, where p_g has the probability of the most likely one-round differential.*

Proof: The case $s = 1$ is trivial. For any $s > 1$,

$$\begin{aligned} &\Pr(\Delta C_s = \beta_s \mid \Delta P = \beta_0) = \\ &\Sigma_{\beta_{s-1}} \ \Pr(\Delta C_s = \beta_s \mid \Delta C_{s-1} = \beta_{s-1}, \Delta P = \beta_0) \\ &\times \quad \Pr(\Delta C_{s-1} = \beta_{s-1} \mid \Delta P = \beta_0) \end{aligned}$$

Since the cipher is a Markov cipher, see Definition 5.2.1 (page 56),

$$\begin{aligned} &\Pr(\Delta C_s = \beta_s \mid \Delta C_{s-1} = \beta_{s-1}, \Delta P = \beta_0 = \\ &\Pr(\Delta C_s = \beta_s \mid \Delta C_{s-1} = \beta_{s-1}) \end{aligned}$$

Now

$$\Pr(\Delta C_s = \beta_s \mid \Delta C_{s-1} = \beta_{s-1}) \leq p_g$$

and

$$\sum_{\beta_{s-1}} \Pr(\Delta C_{s-1} = \beta_{s-1} \mid \Delta C_0 = \beta_0) \leq 1$$

Therefore $\Pr(\Delta C_s = \beta_s \mid \Delta P = \beta_0) \leq p_g$. □

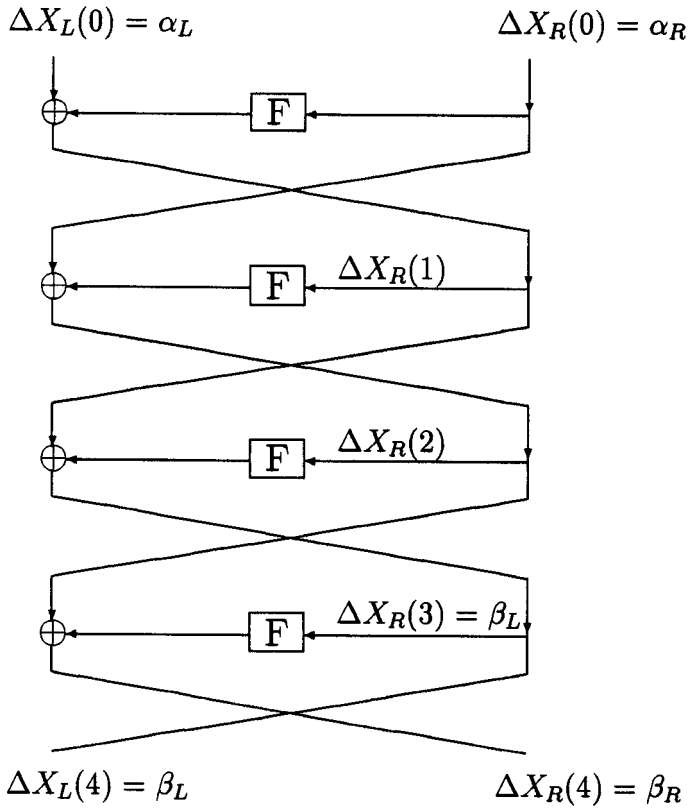


Figure 7.1: A four round differential of a DES-like cipher.

For DES-like iterated ciphers, Theorem 7.3.1 is trivial, since $p_g = 1$, when the right halves of a pair of inputs are equal. For DES-like iterated ciphers, these differentials are called trivial one-round differentials. It is possible to get a lower bound on all differentials in a DES-like iterated cipher expressed

in terms of the most likely non-trivial one-round differential. Let now p_{max} denote

$$p_{max} = \max_{\beta} \max_{\alpha_R \neq 0} \Pr(\Delta C_1 = \beta \mid \Delta P = \alpha) \tag{7.2}$$

where α_R is the right half of α. We assume in the following that $p_{max} < 1$.

Theorem 7.3.2 *Consider an r-round iterated DES-like cipher with independent round keys. Any s-round diferential, $s \geq 4$, has a probability of at most $2p_{max}^2$.*

Proof: It follows from Theorem 5.2.2 that we are considering a Markov cipher. We shall first give the proof for $s = 4$, i.e.,

$$\Pr(\Delta X(4) = \beta \mid \Delta X(0) = \alpha) \leq 2p_{max}^2$$

for any $\beta, \alpha(\neq 0)$. Let α_L, α_R and β_L, β_R be the left and right halves of α and β. We denote by $\Delta X_R(i)$ the right input differences at the i'th round, see Figure 7.1. Let $\delta \rightarrow \epsilon$ denote that, in order for the s-round differential (α, β) to occur, it is *necessary* that inputs to F with difference δ lead to outputs with difference ϵ. We split the proof into cases where $\beta_L = 0$ and $\beta_L \neq 0$. Note that when $\beta_L = 0$ then $\beta_R \neq 0$, otherwise $\alpha_L = \alpha_R = \beta_L = \beta_R = 0$, which is of no use in differential cryptanalysis. Similarly if $\alpha_L = 0$ then $\alpha_R \neq 0$.

1. $\beta_L = 0$. Then clearly $\Delta X_R(2) = \beta_R \neq 0$. If $\Delta X_R(1) = 0$ then $\Delta X_R(2) = \alpha_R = \beta_R \neq 0$. It then follows that $\alpha_R = \beta_R \rightarrow \alpha_L$ in the first round and $\Delta X_R(2) = \beta_R \rightarrow 0$ in the third round, both combinations with probability at most p_{max}. If $\Delta X_R(1) \neq 0$ then it follows that for any given $\Delta X_R(1)$ the second round must be $\Delta X_R(1) \rightarrow \alpha_R + \beta_R$ and the third round must be $\Delta X_R(2) = \beta_R \rightarrow \Delta X_R(1)$, both combinations with probability at most p_{max}. We obtain

$$
\begin{aligned}
&\Pr(\Delta X(4) = \beta \mid \Delta X(0) = \alpha) \\
&= \sum_{\Delta X_R(2)} \Pr(\Delta X_R(1) \mid \Delta X(0) = \alpha) \times \\
&\qquad \Pr(\Delta X(4) = \beta \mid \Delta X(0) = \alpha, \Delta X_R(1)) \\
&= \Pr(\Delta X_R(2) = 0 \mid \Delta X(0) = \alpha) \times \\
&\quad \Pr(\Delta X(4) = \beta \mid \Delta X(0) = \alpha, \Delta X_R(1) = 0) \\
&+ \sum_{\Delta X_R(2) \neq 0} \Pr(\Delta X_R(1) \mid \Delta X(0) = \alpha) \times \\
&\qquad \Pr(\Delta X(4) = \beta \mid \Delta X(0) = \alpha, \Delta X_R(1)) \\
&\leq p_{max}^2 + \sum_{\Delta X_R(1) \neq 0} \Pr(\Delta X_R(1) \mid \Delta X(0) = \alpha) \times p_{max}^2 \\
&\leq 2p_{max}^2
\end{aligned}
$$

since $\sum_{\Delta X_R(1) \neq 0} \Pr(\Delta X_R(1) \mid \Delta X(0) = \alpha) \leq 1$.

2. $\beta_L \neq 0$.

We consider first the 3-round differential obtained by fixing $\Delta X_R(1)$. In the first inequality we use that if $\Delta X_R(2) = 0$ then $\Delta X_R(1) = \beta_L \neq 0$, and it follows that in the second and fourth rounds each of the combinations are upper bounded by p_{max}. For any given non-zero value of $\Delta X_R(2)$ each of the

combinations in the last two rounds are upper bounded by p_{max}. We obtain

$$\begin{aligned}
&\Pr(\Delta X(4) = \beta \mid \Delta X(0) = \alpha, \Delta X_R(1)) \\
&= \sum_{\Delta X_R(2)} \Pr(\Delta X_R(2) \mid \Delta X(0) = \alpha, \Delta X_R(1)) \times \\
&\qquad \Pr(\Delta X(4) = \beta \mid \Delta X(0) = \alpha, \Delta X_R(1), \Delta X_R(2)) \\
&= \Pr(\Delta X_R(2) = 0 \mid \Delta X(0) = \alpha, \Delta X_R(1)) \times \\
&\quad \Pr(\Delta X(4) = \beta \mid \Delta X(0) = \alpha, \Delta X_R(1), \Delta X_R(2) = 0) \\
&+ \sum_{\Delta X_R(2) \neq 0} \Pr(\Delta X_R(2) \mid \Delta X(0) = \alpha, \Delta X_R(1)) \times \\
&\qquad \Pr(\Delta X(4) = \beta \mid \Delta X(0) = \alpha, \Delta X_R(1), \Delta X_R(2)) \\
&\leq p_{max} \times p_{max} + \sum_{\Delta X_R(2) \neq 0} \Pr(\Delta X_R(2) \mid \Delta X(0) = \alpha, \Delta X_R(1)) \times p_{max}^2 \\
&\leq 2p_{max}^2
\end{aligned}$$

The above shows that Theorem 7.3.2 holds for s-round differentials for $s \geq 3$, if $\beta_L \neq 0$. Now

$$\begin{aligned}
&\Pr(\Delta X(4) = \beta \mid \Delta X(0) = \alpha) \\
&= \sum_{\Delta X_R(1)} \Pr(\Delta X_R(1) \mid \Delta X(0) = \alpha) \times \\
&\qquad \Pr(\Delta X(4) = \beta \mid \Delta X(0) = \alpha, \Delta X_R(1)) \\
&\leq \sum_{\Delta X_R(1)} \Pr(\Delta X_R(1) \mid \Delta X(0) = \alpha) \times 2p_{max}^2 \\
&\leq 2p_{max}^2
\end{aligned}$$

Let now $s > 4$. Then

$$\begin{aligned}
&\Pr(\Delta X(s) = \beta \mid \Delta X(0) = \alpha) \\
&= \sum_{\Delta X(s-4)} \Pr(\Delta X(s-4) \mid \Delta X(0) = \alpha) \times \\
&\qquad \Pr(\Delta X(s) = \beta \mid \Delta X(0) = \alpha, \Delta X(s-4))
\end{aligned}$$

Since we assumed that the round keys are independent and uniformly random it follows from the proof for $s = 4$ that

$$\Pr(\Delta X(s) = \beta \mid \Delta X(0) = \alpha)\Delta X(s-4)) = \Pr(\Delta X(s) = \beta \mid \Delta X(s-4) \leq 2p_{max}^2$$

Thus $\Pr(\Delta X(s) = \beta \mid \Delta X(0) = \alpha) \leq 2p_{max}^2$. □

We say that the F-function in a DES-like cipher is a permutation, if F is a permutation, when each one of the two arguments are held constant. In this case Theorem 7.3.2 can be proved for $s \geq 3$. It comes from the fact that to have equal outputs of one round we must have equal inputs.

Theorem 7.3.3 *Consider an r-round iterated DES-like cipher with independent round keys where the F-function in a permutation. Any s-round differential, $s \geq 3$, has a probability of at most $2p_{max}^2$.*

Proof: We give the proof for $s = 3$. The general case can then be proved like in the preceding theorem. Again we separate between two cases and use the same notation as before.
1. $\beta_L = 0$. Then $\Delta X_R(0) = \alpha_R \neq 0$, otherwise different inputs would have to yield equal outputs in the second round, but that is not possible, since f is a permutation. The difference in the inputs at the first round is $\alpha_R \neq 0$ and the difference in the inputs at the second round is $\beta_R \neq 0$, thus $\Pr(\Delta X(3) = \beta \mid \Delta X(0) = \alpha) \leq p_{max}^2$.
2. $\beta_L \neq 0$. Like in the proof of Theorem 7.3.2 we split into cases where $\Delta X_R(1)$ is zero or not. Note that $\Delta X_R(1) = 0 \Rightarrow \alpha_L \neq 0$ otherwise $\alpha_R \Rightarrow \alpha_L = 0 \Rightarrow \alpha_R = 0$. We obtain

$$\begin{aligned}
&\Pr(\Delta X(3) = \beta \mid \Delta X(0) = \alpha) \\
&= \sum_{\Delta X_R(1)} \Pr(\Delta X_R(1) \mid \Delta X(0) = \alpha) \times \\
&\qquad \Pr(\Delta X(3) = \beta \mid \Delta X(0) = \alpha, \Delta X_R(1)) \\
&= \Pr(\Delta X_R(1) = 0 \mid \Delta X(0) = \alpha) \times \\
&\quad \Pr(\Delta X(3) = \beta \mid \Delta X(0) = \alpha, \Delta X_R(1) = 0)
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{\Delta X_R(1)\neq 0} \Pr(\Delta X_R(1) \mid \Delta X(0) = \alpha) \times \\
&\qquad \Pr(\Delta X(3) = \beta \mid \Delta X(0) = \alpha, \Delta X_R(1)) \\
&\leq p_{max}^2 + \sum_{\Delta X_R(1)\neq 0} \Pr(\Delta X_R(1) \mid \Delta X(0) = \alpha) \times p_{max}^2 \\
&\leq 2p_{max}^2
\end{aligned}
$$

□

We will show in the following that the round function in an iterated cipher can be chosen in such a way that the probability of any non-trivial one-round differential, p_{max}, is small.

7.3.1 Differentially uniform mappings

By using the functions studied in [85, 2, 83, 25] one can obtain round functions in a DES-like cipher such that p_{max} is small. The functions in a binary field can be used to construct mappings with a difference distribution table, whose entries are either 0 or 2. This is the 'most uniform' distribution of differences one can obtain in a field where the difference operation used is commutative.

Definition 7.3.1 (Nyberg [83]) *A mapping $F : G_1 \rightarrow G_2$, where G_1 and G_2 are Abelian groups, is* differentially δ-uniform, *if for all $\alpha \in G_1 \setminus \{0\}$ and $\beta \in G_2$*

$$|\{z \in G_1 \mid F(z + a) - F(z) = \beta\}| \leq \delta$$

As an example, consider

Theorem 7.3.4 *The mapping $f(x) = x^{2^k+1}$ in $GF(2^n)$ over $GF(2)$, where $gcd(2^k - 1, 2^n - 1) = t$ is differentially $(t + 1)$-uniform for the difference induced by addition module 2. Furthermore, if $gcd(2^k + 1, 2^n - 1) = 1$, f is a pemautation.*

Proof: First note that for $\alpha \neq 0$ the equation

$$f(x \oplus \alpha) \oplus f(x) = \beta \tag{7.3}$$

$f(x)$	δ_f	$N(f)$	$ord(f)$	Conditions
x^{2^k+1}	2^s		2	$s = gcd(k,n)$ [83, 2]
x^{2^k+1}		$2^{n-1} - 2^{(n+s)/2-1}$		$s = gcd(k,n)$, $\frac{n}{s}$ odd[83]
$(x^{2^k+1})^{-1}$	2	$2^{n-1} - 2^{(n+s)/2}$	$(n+1)/2$	$gcd(k,n) = 1$, n odd [83]
x^{-1}	2	$2^{n-1} - 2^{n/2}$	$n-1$	n odd [83, 2]
x^{-1}	4	$2^{n-1} - 2^{n/2}$	$n-1$	n even [83, 2]
x^7	6		3	n odd [2]

Table 7.2: Differentially uniform mappings in $GF(2^n)$ over $GF(2)$.

has at least two or no solutions for x, i.e., if $x = \gamma$ is a solution, then so is $x = \gamma \oplus \alpha$. And

$$\begin{aligned}(x \oplus \alpha)^{2^k+1} \oplus x^{2^k+1} &= (x \oplus \alpha) \times (x \oplus \alpha)^{2^k} \oplus x^{2^k+1} = \\ (x \oplus a) \times (x^{2^k} \oplus \alpha^{2^k}) \oplus x^{2^k+1} &= (\alpha \times x^{2^k}) \oplus (x \times \alpha^{2^k}) \oplus \alpha^{2^k+1}\end{aligned}$$

Thus, by letting x and y be two different solutions to equation (7.3) it follows that

$$\begin{aligned}(\alpha \times x^{2^k}) \oplus (x \times \alpha^{2^k}) \oplus \alpha^{2^k+1} &= (\alpha \times y^{2^k}) \oplus (y \times \alpha^{2^k}) \oplus \alpha^{2^k+1} \Leftrightarrow \\ \alpha \times (x^{2^k} \oplus y^{2^k}) &= \alpha^{2^k} \times (x \oplus y) \Leftrightarrow \\ (x \oplus y)^{2^k-1} &= \alpha^{2^k-1} \qquad (7.4)\end{aligned}$$

since $x \neq y$. It is well known (see e.g. [59, Th. 1.151]) that for any positive divisor d of the order of a cyclic group, the group contains precisely one subgroup of order d. Here the group order is $N = 2^n - 1$ and since t divides N by definition, there is a subgroup $Z = \{z_1, z_2, \ldots, z_t\}$. It follows that there are exactly t solutions to equation (7.4), namely $x \oplus y = \alpha \times z_i$ and therefore there are either $t+1$ or no solutions to equation (7.3). The last part of the proof is trivial. □

We note that, $\delta_f = t+1$, where $t = gcd(2^n-1, 2^k-1)$ is equivalent to $\delta_f = 2^s$, where $s = gcd(k,n)$. We summarise the results of [85, 2, 83] in Table 7.2, where δ_f is the highest non-trivial entry in the difference distribution table for f. $N(f)$ is the nonlinearity of f, i.e., the smallest of the Hamming distances of any non-zero linear combination of the output coordinate functions to the set of all affine functions. $ord(f)$ is the order of the coordinate functions of f, e.g. when $ord(f) = 2$, every output coordinate of f is a function of quadratic

and linear terms of the input coordinates. Note that squaring in $GF(2^n)$ over $GF(2)$ is a linear function, which means that for any of functions $f(x) = x^d$ in Table 7.2 and the functions $g(x) = (f(x))^{2^l} = x^{d2^l}$ it holds that $\delta_f = \delta_g$ and $N(f) = N(g)$.

When moving to other fields it is possible to obtain difference distribution tables with all 1 entries. As an example

Example 7.3.1 *The mapping $f(x) = x^2 \bmod p$, where p in a prime, is differentially 1-uniform, where the difference is induced by addition module p.*

The proof is given in Theorem 5.2.3. Note that the mapping in Example 7.3.1 is not a permutation. Also note that for a mapping, whose domain and range are of the same size, differentially 1-uniformity means that **all** entries in the difference distribution table for a non-zero input difference are 1. Therefore a permutation cannot be differentially 1-uniform, since different inputs never yields equal outputs.

As can be seen from Table 7.2 differential uniformity often requires the size of the domain and image of the functions to be odd. This is inconvenient in the design of block ciphers. The following result is useful.

Theorem 7.3.5 *Consider a mapping $F : GF(2^n) \rightarrow GF(2^n)$ and assume that F is differentially d-uniform. Then the mapping F_l obtained from F by discarding any l output bits is differentially $2^l \times d$-uniform.*

Proof: Consider the difference distribution table for F. All columns have a maximum entry of d. The difference distribution table for F_1, obtained from F by discarding one output bit, is the table for F where columns are added pairwise together. Therefore F_1 is differentially $2 \times d$-uniform. The result now follows by induction on l. □

7.4 Markov Ciphers and Differentials

In Section 7.3 we showed that it is possible to obtain an upper bound on the probabilities of any differential in an iterated cipher. However, this upper bound can only be used to prove resistance against a differential attack, when

p_{max}, the probability of the most likely one-round difference, is chosen to be small. For many practical ciphers p_{max} is relatively high. As an example, for the DES p_{max} is $1/4$, which means that for the DES with independent round keys the upper bound of the probability of any s-round differential, $s \geq 4$, is $2 \times p_{max}^2 = 1/8$. Thus a differential attack needs at least 8 chosen plaintext pairs. This lower bound on the complexity of a differential attack is, of course, too low to conclude resistance against a differential attack. We consider from now on an iterated cipher of block length n. For Markov ciphers more information about the probabilities of differentials can be obtained. The theory of Markov chains is well explored, see e.g. Feller [29]. Theorem 5.2.1 connects the theory of differential cryptanalysis with the theory of Markov chains.

Denote by $\mathbf{P}$ *the transition pobability matrix* of the homogeneous Markov chain, the $N \times N (N = 2^n - 1)$ matrix representing the *difference distribution table* for one round of the cipher. Let p_{ij}^s denote the probability that state j can be reached from state i in s steps. In other words for our case, that a certain difference i in the ciphertexts after t rounds, can result in another certain difference j in the ciphertexts after $t + s$ rounds. In the same way we let $\mathbf{P}^t$ denote the transition matrix with entries p_{ij}^t. If all probabilities p_{ij}^t are sufficiently small for some t, then we can expect the cipher to be secure against a differential attack after t rounds. Calculating $\mathbf{P}^t$ for $t = 2, 3, \ldots$ is usually computationally infeasible. For example, the transition matrices of the DES and IDEA are of size $2^{64} - 1$, and for the DES with $t = 15$, finding $\mathbf{P}^{15}$ is equivalent to finding the probabilities of all 15 round differentials. However, it is possible to study the asymptotic behaviour of the transition matrices. Note, that the transition matrix of a cipher is doubly stochastic, i.e., every row and every column sum to one. One row of $\mathbf{P}$ is a probability distribution, so every row sum to one [55]. A column of $\mathbf{P}$ is equivalent to a row of the transition matrix for the inverse of the round function and since it is bijective by definition, every column of $\mathbf{P}$ also sum to one.

Definition 7.4.1 ([29]) *A finite Markov chain is said to be* irreducible, *if for any* (i, j) *there exists an* r, *s.t.* $p_{ij}^r > 0$.

In our case, it means for example, that starting with a difference i in the plaintext, then any difference j is possible after r rounds.

Definition 7.4.2 ([29]) *A state i called* periodic *with period d_i, if $d_i > 1$ and $p_{ii}^s = 0$ except when $d_i \mid s$. A state i is called* aperiodic *with $d_i = 1$, if $p_{ii}^1 > 0$. The period of $\mathbf{P}$ is defined as $gcd(d_1, \ldots, d_N)$.*

It is seen that $\mathbf{P}$ is aperiodic, if there exists an i, s.t. $d_i = 1$

Definition 7.4.3 ([29]) *A Marcov chain, which is both aperiodic and irreducible is called* ergodic.

Theorem 7.4.1 ([29]) *If a Marcov chain is ergodic, then there exists an unique distribution $\{u_k\}$, s.t. for all (i, j)*

$$\lim_{n\to\infty} \; p_{ij}^n = u_j > 0 \tag{7.5}$$

Furthermore, if the transition matrix is doubly stochastic, $\{u_k\}$ is the uniform distribution.

In our case of differential cryptanalysis, the uniform distribution is when $u_k = 1/N$ for all k and Theorem 7.4.1 tells us, that if the transition matrix is irreducible and aperiodic, then the cipher is resistant against a differential attack after sufficiently many rounds. It does not give us an exact number of how many rounds. The aperiodicity of a transition matrix is sometimes easy to show, e.g. for the DES an input xor Γ = (1960000 1960000) results in an output xor Γ with probability $\frac{1}{234}$ i.e., Γ is aperiodic. Also IDEA has aperiodic states [55, Prop.4, p. 61]. To show irreducibility is a harder problem. Note, that the transition matrix $\mathbf{P}$ of a Markov cipher is irreducible, if there exists an r, s.t. $\mathbf{P}^r$ contains no zero entries. Moreover, it suffices to prove that $\mathbf{P}^{r_0}$ has a row or a column with no zero entries for some r_0 [1, 55].

A transition matrix $\mathbf{P}$ can be seen as a directed graph $\mathbf{G}$, where the nodes represent the states in the Markov chain, which has an edge from node i to node j, if a difference i can result in a difference j. O'Connor [87, 54] showed,

Theorem 7.4.2 *If the round function in an iterated cipher is selected uniformly from the set of all n-bit permutations S_{2^n} then*

$$\Pr(\mathbf{P} \textit{ is ergodic}) \to 1$$

for large n.

The proof of O'Connor's result makes use of a result from graph theory, saying that for large N if the number of edges is dominating $Nlog_2N$, where N is the number of nodes in the graph, then with a high probability $\mathbf{G}$ is strongly connected, i.e., from every node there is a path to all other nodes, which in our case means that $\mathbf{P}$ is ergodic. This result suggests, that most iterated ciphers will be resistant against differential attacks after sufficiently many rounds.

In Section 5.2.3 we described the hypothesis of stochastic equivalence. O'Connor examined the difference distribution table for randomly chosen permutations, where the difference operator is the XOR operation.

Theorem 7.4.3 (O'Connor [89]) *Let $\pi \in S_{2^m}$, the set of all m-bit permutations, be uniformly selected. Then for large m, the expected value of the largest entry in the XOR table of π is less than $2m$, and the expected fraction of the table that is zero is tending towards $e^{-\frac{1}{2}} \approx 0.6065$.*

It means that although in an ergodic m-bit Markov cipher the probabilities of all differentials are tending towards the uniform distribution, in an actual attack, a fixed key is used and some differential will have a probability of about $2m/2^m$ for large m. However, if the cipher is ergodic and has sufficiently many rounds, an attacker has no chance of knowing which differential has a high probability, since he does not know the key.

We will now consider Markov ciphers in the special case, where the iterated cipher has a Feistel structure.

7.4.1 Feistel ciphers

In this section we consider Feistel ciphers, cf. Definition 2.5.2, with block size $2n$. In [55] it is recommended that for an iterated block cipher, the transition matrix $\mathbf{P}$ is non-symmetric. It means, that it should not be possible that a difference α can result in difference β and at the same time a difference β can result in difference α. These combinations can be iterated any number of times, and if there exist combinations with a high probability the cipher may be vulnerable to differential attacks. Note that one round in a Feistel cipher includes the swapping of the halves after the evaluation of the F-function. For Feistel ciphers we have the following result.

Theorem 7.4.4 *If the round function F of a Feistel cipher is a permutation, see page* 153, *then the transition matrix* $\mathbf{P}$ *is non-symmetric.*

Proof: Assume that $(\alpha_L, \alpha_R) \rightarrow (\beta_L, \beta_R)$ and that $(\beta_L, \beta_R) \rightarrow (\alpha_L, \alpha_R)$, where $(\alpha_L, \alpha_R) \neq (0, 0)$. Because the right half of a ciphertext in one round in a Feistel cipher equals the left half of the ciphertext in the following round, $\alpha_R = \beta_L$ and $\alpha_L = \beta_R$. It follows that inputs to the F-function with difference α_L and inputs with difference α_R both have to lead to equal outputs. Since F is a permutation, this implies that $\alpha_L = \alpha_R = 0$. □

If the round function is not a permutation, then it is possible to construct 2-round iterative characteristics, like in Biham and Shamir's attack on the DES [7].

The result of O'Connor [87, 54], that most iterated ciphers are resistant to differential attacks, does not apply directly to the case of Feistel iterated ciphers, but we will show that the result also holds in that case.

Lemma 7.4.1 *In a Feistel cipher, the number of non-zero entries in one row of the transition matrix* $\mathbf{P}$ *is at most* $\sqrt{2^{2n}} = 2^n$. *If the round function F is chosen uniformly from S_{2^n}, then for large n the expected number of non-zero entries of one row is* $2^n \times e^{-\frac{1}{2}} \approx 2^{n-1}$.

Proof: The first part follows from the structure of Feistel ciphers. i.e., an input difference (β, α) leads always to an output difference (α, X), where α, β and X are n-bit values. Now the second part follows from Theorem 7.4.3. □

From Lemma 7.4.1 it follows that the expected number of non-zero entries in $\mathbf{P}$ is $2^{2n} \times 2^{n-1} = 2^{3n-1}$, when n is large. But since $2^{3n-1} > 2^{2n} \times log_2\ 2^{2n} = n \times 2^{2n+1}$ for n large, the result of 7.4.2 holds also for Feistel ciphers. Although Lemma 7.4.1. assumes that the round function of the cipher is a permutation, a similar result will hold for any uniformly selected round function. In other words the following result holds.

Theorem 7.4.5 *If the round function (permutation) in a Feistel cipher in selected uniformly from the set of all n-bit function, then*

$$\Pr(\mathbf{P}\ \textit{is ergodic}) \rightarrow 1$$

for large n.

As noted in the previous section it is not possible to calculate $\mathbf{P}^r$ when the block size is large. For Feistel ciphers with round permutations it is possible to derive the following property.

Theorem 7.4.6 *If the F-function of a Feistel cipher is a permutation, then the transition matrices* $\mathbf{P}^r, r \leq 5$, *have zero entries.*

Size (bits)		F perm.		F not perm.		All	
F-fct.	Cipher	No.	Markov	No.	Markov	No.	Markov
2	4	24	0.0%	232	41.4%	256	37.5%
3	6	8!	80.0%	2^{24} - 8!%	88.5%	2^{24}	88.4%

Table 7.3: The ratio of Markov to all ciphers in 4 and 6 bit ciphers (exhaustive search).

Proof: We prove only the case $r = 5$. The other cases are quite similar. In a 5-round differential an input difference $(\Gamma, 0)$ will never result in an output difference $(0, \Gamma)$ for any $\Gamma \neq 0$. The differential must have the following form.

$$\begin{array}{ccc} & (\Gamma, 0) & \\ 0 & \leftarrow & 0 \\ \Phi & \leftarrow & \Gamma \\ 0 & \leftarrow & \Phi \\ \Phi & \leftarrow & \Gamma \\ 0 & \leftarrow & 0 \\ & (0, \Gamma) & \end{array}$$

It is seen that it is necessary that $\Phi = 0$, but $0 \not\leftarrow \Gamma$ by assumption. □

This result suggests that if the F-function in a Feistel cipher is a permutation, it may take more rounds before $\mathbf{P}$ reaches the uniform distribution, since a similar result does not hold when the F-function is not a permutation.

To test Feistel ciphers for ergodicity we did an exhaustive search for all four and six bit Feistel ciphers. We split the tests in cases depending on whether the F-function is a permutation or not. The result of this experiment is summarised in Table 7.3. For 8-bit Feistel ciphers it is not possible to do an

exhaustive search in reasonable time, since the number of 8-bit Feistel ciphers is $16^{16} = 2^{64}$ and the number of ciphers with F a permutation is $16! \simeq 2^{44}$ We did 1000 tests, where in each test we chose a round function/permutation at random. Also we calculated the smallest number of rounds t, s.t. P^t contained no zero entries. The result of these tests is summarised in Table 7.4. It is seen from Table 7.4 that almost all 8 bit ciphers are Markov ciphers and that it takes about 6 rounds before P^t contains no zero entries, i.e., all nonzero differences are possible. The difference between using round functions and round permutations is visible but not significant.

Size (bits)		F perm.			F not perm.		
F-fct.	Cipher	No.	Markov	#Rounds	No.	Markov	#Rounds
4	8	1000	$> 99\%$	6.4	1000	$> 99\%$	6.3

Table 7.4: The ratio of Markov to all ciphers in 8 bit ciphers (1000 tests).

7.5 Resistance Against Linear Attacks

In this section we consider iterated block ciphers. It is possible to get a lower bound on all linear approximations of an iterated cipher expressed in terms of the most likely one-round approximation.

Definition 7.5.1 *For a boolean function $f : GF(2)^n + GF(2)$, the* **Walsh transform** *is defined*

$$F(w) = \sum_{x \in GF(2)^x} (-1)^{f(x)} \times (-1)^{x \cdot w} \tag{7.6}$$

where '$\cdot$' denotes the dot poduct.

The following theorem is well known, see e.g. [42, 71].

Theorem 7.5.1 *(Parseval's Theorem) For any boolean function, $f : GF(2)^n \rightarrow GF(2)$,*

$$\sum_{x \in GF(2)^x} (F(w))^2 = 2^{2n}$$

where F is the Walsh transform of f.

Parseval's theorem tells us that the squares of all Walsh transforms, $F(w)$, sum to a constant and the following result follows

Corollary 7.5.1 *Let $g : GF(2)^n \to GF(2)^m$. For any $b \in GF(2)^m$*

$$\sum_{a \in GF(2)^n} |Pr_X(a \cdot X = b \cdot g(X)) - 1/2|^2 = 1/4 \tag{7.7}$$

Proof: Let $f : GF(2)^n \to GF(2)$, $f(x) = b \cdot g(x)$. Then

$$\begin{aligned} F(a) &= \sum_{x \in GF(2)^n} (-1)^{f(x)} \times (-1)^{x \cdot a} \\ &= \#\{x : a \cdot x = f(x)\} - \#\{x : a \cdot x \neq f(x)\} \\ &= 2 \times \#\{x : a \cdot x = f(x)\} - 2^n \\ &= 2^{n+1} \times (\mathrm{Pr}_X(a \cdot x = f(x)) - 1/2) \end{aligned}$$

Since $\sum_{a \in GF(2)^n} (F(a))^2 = 2^{2n}$ we have completed the proof. □

In the following let X_i denote the ciphertext after i rounds of encryption, where X_0 is the plaintext. We consider first an r-round iterated m-bit cipher with round function g. Denote by p_g the probability of the best linear approximation of g, i.e.,

$$|p_g - 1/2| = \max_{k \in GF(2)^m} \max_{\alpha, \beta \neq 0} |\mathrm{Pr}_X(g(X, k) \cdot \beta = X \cdot \alpha) - 1/2| \tag{7.8}$$

Theorem 7.5.2 *Consider an r-round iterated cipher with independent round keys. Any s-round linear approximation, $s \geq 1$, has a probability p_L, such that*

$$|p_L - 1/2|^2 \leq |p_g - 1/2|^2.$$

Proof:

$$\begin{aligned}
&|\Pr_X(X_0 \cdot \alpha = X_s \cdot \beta) - \frac{1}{2}|^2 \\
&\leq \sum_{b_1 \in GF(2)^m} 2^2 \times |\Pr_X(X_0 \cdot \alpha = X_1 \cdot b_1) - 1/2|^2 \times \\
&\qquad\qquad |\Pr_X(X_1 \cdot b_1 = X_s \cdot \beta) - 1/2|^2 \\
&= |p_g - \frac{1}{2}|^2 \times 2^2 \times \sum_{b_1 \in GF(2)^m} |\Pr_X(X_0 \cdot \alpha = b_1 \cdot X_1) - 1/2|^2 \\
&= |p_g - \frac{1}{2}|^2
\end{aligned}$$

where we have upper bounded the first linear approximation in terms of p_g and used Corollary 7.5.1 to upper bound the remaining rounds. □

We consider now DES-like iterated ciphers with block size $2n$, cf. Definition 2.5.3, Recall that

$$\begin{aligned}
F(X, K_i) &= f(E(X)) + K_i) \\
f &: GF(2)^m \rightarrow GF(2)^n,\ m \geq n \\
E &: GF(2)^n \rightarrow GF(2)^m, \text{ an affine expansion mapping}
\end{aligned}$$

Since p_g is trivially one, we let pf denote the probability of the most likely non-trivial one round approximation, i.e.,

$$|p_f - 1/2| = \max_{k \in GF(2^m)} \max_{b \neq 0,\ a} |\Pr_X(F(X,k) \cdot b = X \cdot a) - 1/2| \qquad (7.9)$$

Theorem 7.5.3 *Consider an r-round iterated DES-like cipher with independent round keys. Any s-round linear approximation, $s \geq 4$, has a probability p_L such that*

$$|p_L - 1/2|^2 \leq 8|p_f - 1/2|^4.$$

Proof: We consider first $s = 4$ and the four round linear approximation in Figure 7.2, where $\alpha_L, \alpha_R, \beta_L, \beta_R \in GF(2)^n$ are fixed values. We omit the key k and let $F(X, k)$ be denoted by $F(X)$. Note that in the right half of the ciphertext after the first round although we have knowledge about the bits

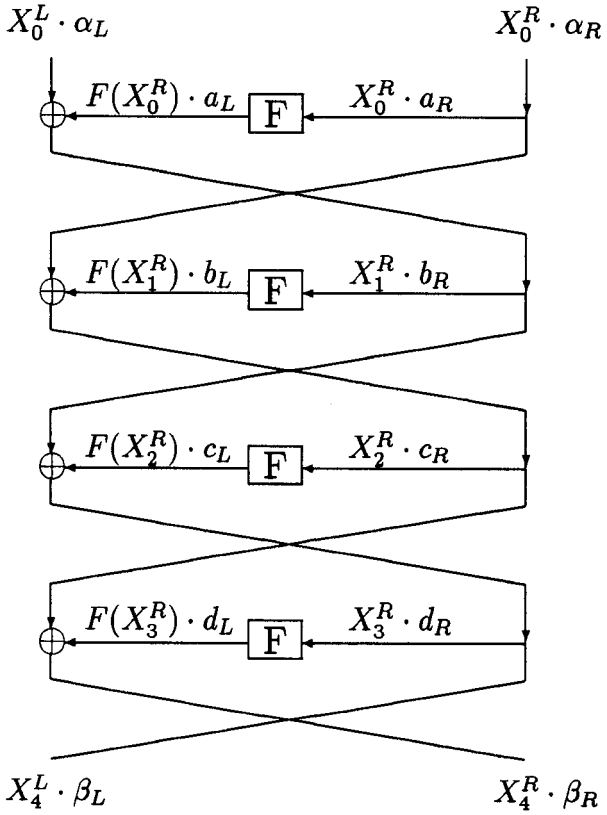


Figure 7.2: A four round linear approximation.

determined by α_L it doesn't necessarily mean that we use exactly those bits to approximate the F-function in the second round, so b_R is not in general equal to α_L. However, it follows from Figure 7.2 that the values of the left and right halves of α, β, a, b, c and d are dependent.

In Figure 7.3 we give the skeleton of a four round linear approximation, where we have omitted the X_i's and $F(X_i)$'s. We split the proof in two cases depending on the value of β_R.

1. $\beta_R = 0$.
There are no approximations in the last round. It follows that $\beta_L \neq 0$, $b_L = c_R$, $d_R = c_L \oplus \beta_L = 0$ and $b_R = \alpha_L \oplus \beta_L$. Also for a fixed value of $c_R = \gamma$ all values of the skeleton of Figure 7.3 are fixed, i.e.,

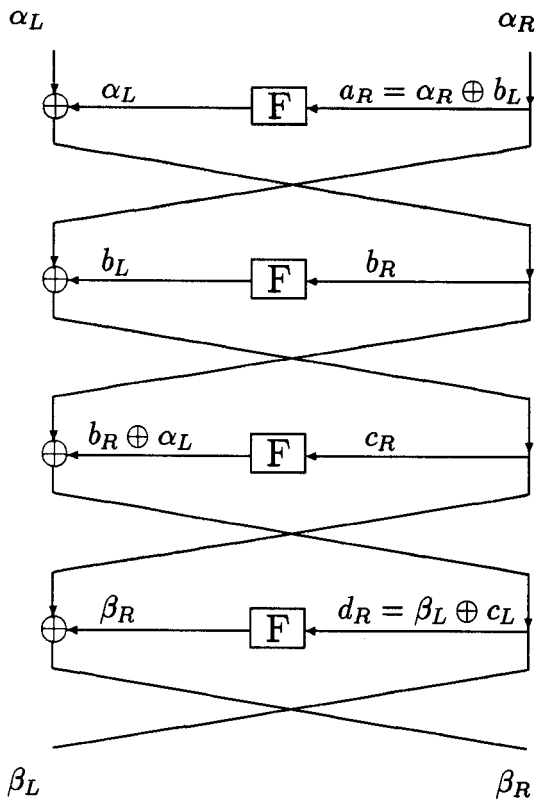


Figure 7.3: The skeleton of a four round linear approximation.

$$\begin{array}{ll} a_L = \alpha_L & a_R = \alpha_R \oplus \gamma \\ b_L = c_R = \gamma & b_R = \alpha_L \oplus \beta_L \\ c_L = \beta_L & c_R = \gamma \\ d_L = 0 & d_R = 0 \end{array}$$

In particular, if $c_R = 0$ then $b_L = 0$ and $b_R = 0$. In this case, there can be no approximation in the second round and the approximation in the first and third rounds are non-trivial, i.e.,

$$|\Pr_X(X_0 \cdot \alpha = X_4 \cdot \beta) - \tfrac{1}{2}|^2 \leq 4 \times |p_f - 1/2|^4.$$

If $c_R = \gamma \neq 0$ the approximations in the second and third rounds are non trivial and both upper bounded by $|p_f - 1/2|$. The approximation in the first

round is upper bounded using Corollary 7.5.1. More explicitly we obtain,

$$
\begin{aligned}
&|\Pr_X(X_0 \cdot \alpha = X_4 \cdot \beta) - \frac{1}{2}|^2 \\
&\leq \sum_{\gamma=0}^{2^m} |\Pr_X(X_0 \cdot \alpha = X_4 \cdot \beta, c_R = \gamma) - \frac{1}{2}|^2 \\
&= \sum_{\gamma=1}^{2^m} |\Pr_X(X_0 \cdot \alpha = X_4 \cdot \beta, c_R = \gamma) - \frac{1}{2}|^2 \\
&\quad + |\Pr_X(X_0 \cdot \alpha = X_4 \cdot \beta, c_R = 0) - \frac{1}{2}|^2 \\
&\leq \sum_{\gamma=1}^{2^m} |4 \times (\Pr_X(X_0 \cdot (\alpha_R \oplus \gamma) = F(X_0) \cdot \alpha_L) - \frac{1}{2}) \\
&\quad \times (\Pr_X(X_1 \cdot (\alpha_L \oplus \beta_L) = F(X_1) \cdot \gamma) - \frac{1}{2}) \\
&\quad \times (\Pr_X(X_2 \cdot \gamma = F(X_2) \cdot \beta_L) - \frac{1}{2})|^2 \\
&\quad + 4 \times |p_f - \frac{1}{2}|^4 \\
&\leq |p_f - \frac{1}{2}|^4 \times \sum_{\gamma=1}^{2^m} |4 \times \Pr_X(X_0 \cdot (\alpha_R \oplus \gamma) = F(X_0) \cdot \alpha_L) - \frac{1}{2}|^2 \\
&\quad + 4 \times |p_f - \frac{1}{2}|^4 \\
&= 8 \times |p_f - \frac{1}{2}|^4
\end{aligned}
$$

2. $\beta_R \neq 0$.
In this case it suffices to consider a three round approximation, depicted in Figure 7.4. Note that fixing a value of $c_L = \gamma$ all other values in the skeleton

of Figure 7.4 are fixed. We proceed in a way similar as above.

$$
\begin{aligned}
&|\Pr_X(X_1 \cdot \alpha = X_4 \cdot \beta) - \frac{1}{2}|^2 \\
&\leq \sum_{\gamma=0}^{2^m} |\Pr_X(X_1 \cdot \alpha = X_4 \cdot \beta, c_L = \gamma) - \frac{1}{2}|^2 \\
&= \sum_{\gamma=1}^{2^m} |\Pr_X(X_1 \cdot \alpha = X_4 \cdot \beta, c_L = \gamma) - \frac{1}{2}|^2 \\
&\quad + |\Pr_X(X_1 \cdot \alpha = X_4 \cdot \beta, c_L = 0) - \frac{1}{2}|^2 \\
&\leq \sum_{\gamma=1}^{2^m} |4 \times (\Pr_X(X_1 \cdot (\alpha_R \oplus \gamma) = F(X_1) \cdot \alpha_L) - \frac{1}{2}) \\
&\quad \times (\Pr_X(X_2 \cdot (\alpha_L \oplus \beta_R) = F(X_2) \cdot \gamma) - \frac{1}{2}) \\
&\quad \times (\Pr_X(X_3 \cdot (\gamma \oplus \beta_L) = F(X_3) \cdot \beta_R) - \frac{1}{2})|^2 \\
&\quad + 4 \times |p_f - \frac{1}{2}|^4 \\
&\leq |p_f - \frac{1}{2}|^4 \times \sum_{\gamma=1}^{2^m} |4 \times \Pr_X(X_1 \cdot (\alpha_R \oplus \gamma) = F(X_1) \cdot \alpha_L) - \frac{1}{2}|^2 \\
&\quad + 4 \times |p_f - \frac{1}{2}|^4 \\
&= 8 \times |p_f - \frac{1}{2}|^4
\end{aligned}
$$

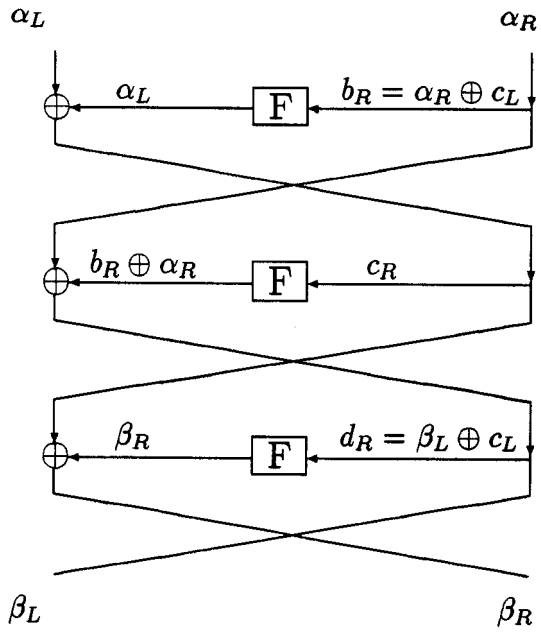


Figure 7.4: The skeleton of a three round linear approximation.

Now we consider the case where $s > 4$.

$$
\begin{aligned}
&|\mathrm{Pr}_X(X_0 \cdot \alpha = X_s \cdot \beta) - \frac{1}{2}|^2 \\
&\leq \sum_{\beta_s - 4} 2^2 \times |\mathrm{Pr}_X(X_0 \cdot \alpha = X_{s-4} \cdot b_{s-4}|^2 \times \\
&\qquad |\mathrm{Pr}_X(X_{s-4} \cdot b_{s-4} = X_s \cdot b_s)|^2 \\
&= 2^3 \times |p_f - \frac{1}{2}|^4 \times 2^2 \times \sum_{b_s - 4} |\mathrm{Pr}_X(X_0 \cdot (\alpha = b_{s-4} \cdot X_{s-4})|^2 \\
&= 8 \times |p_f - \frac{1}{2}|^4
\end{aligned}
$$

And we have completed the proof. □

Finally we note that a similar but different result appears in [84]

7.6 Ciphers Resistant to Differential and Linear Attacks

In the following we will show examples of ciphers resistant to both differential and linear attacks. We will use the differentially uniform mappings from Section 7.3.1, which we will see are also highly nonlinear. Note that when the nonlinearity of a function f, where $f : GF(2^n) \rightarrow GF(2^m)$, is $N(f)$, any linear approximation of f is bounded as follows (see also [36]),

$$|p_f - 1/2| \leq \frac{2^{m-1} - N(f)}{2^m}$$

In the following examples it is assumed that the round keys are independent.

7.6.1 Iterated cipher

Let $h(x) = x^{-1}$ in $GF(2^{64})$, where $h(x) = 0$ for $x = 0$, be the round function in an r-round iterated 64 bit cipher $E_K(\cdot)$. Each round take a 64 bit text input and a 64 bit round key, which are exclusive-or'ed to form the input to g, i.e., $g(X, K) = h(X \oplus K)$. Obviously h is a permutation and its own inverse and according to Table 7.2, h is differentially 4-uniform. Following Theorem 7.3.1 every s-round differential of $E_K(\cdot)$ for $s \geq 1$ has a probability of at most $4/2^{64} = 2^{-62}$.

The nonlinearity $N(h) = 2^{63} - 2^{32}$ according to Table 7.2 and by Theorem 7.5.2

$$|p_L - 1/2|^2 \leq |p_h - 1/2|^2 \leq (\frac{2^{32}}{2^{64}})^2 = 2^{-64}$$

It follows that h is highly resistant to both differential and linear attacks.

7.6.2 DES-like iterated cipher

Let $g(\mathbf{x}) = \mathbf{x}^5$ in $GF(2^{50})$ be the round function in a DES-like iterated 100 bit cipher. Each round take a 50 bit input and a 50 bit key, which exclusive-or'ed to form the input to F, i.e., $F(X, K) = f(X \oplus K)$. From Theorem 7.3.4 it follows that p_{max} of f is $\frac{4}{2^{50}}$ and f is a permutation. From Theorem 7.3.3

it follows that every s-round differential of this block cipher has probability less than or equal to 2^{-95} for $s \geq 3$.

According to Table 7.2 the nonlinearity $N(f)$ is $2^{49} - 2^{25}$. Therefore $|p_f - 1/2| \leq 2^{-25}$ and by Theorem 7.5.3

$$|p_L - 1/2|^2 \leq 8 \times |p_f - 1/2|^4 = 2^{-97}$$

It follows that this cipher is highly resistant to both differential and linear attacks.

7.7 Strong Key Schedules

In [100] ideas of how to improve the resistance of the DES to an exhaustive key search attack were given. The ideas given in this section are inspired by [100]. In [7] it is shown that the DES with independent round keys, i.e., a 768 bit key, is not essentially stronger than the DES with a 56 bit key. An attack using 2^{59} pairs of encryptions is presented, which finds the secret 768 bit key in time about 2^{61} encryptions. The improved attack on DES [7, Sect. 5] exploits the dependencies in the round keys and is not directly applicable to the DES with independent round keys. The complexity of an improved differential attack on the DES with independent round keys is not known to us. It seems, however, to require more than the 2^{47} chosen plaintexts used to attack the DES with dependent round keys as in [7]. In [64, 65, 66] a linear attack on the full 16-round DES is outlined. It finds 26 bits of the 56-bit key using 2^{45} known plaintexts. It is suggested to find the remaining 30 bits by exhaustive search. It is obvious that the existence of a linear attack finding the full round key of the last round would enable a possible attack on the DES with independent round keys, since the ciphertexts can then be decrypted one round with the obtained round key and a linear attack on the DES with 15 rounds can be performed. It seems though, that a linear attack on the round key in the last round of the DES will require many linear expressions [64, 65, 66], including expressions with a probability that requires many known plaintexts for the key to be uniquely determined.

The above speaks in favour of independent round keys in DES-like iterated ciphers. However, as an example, a 768 bit key for the DES is of no practical interest. The security gained seems, after all, to be small compared

to the big increase in the key size. We introduce new properties of a key schedule in a Feistel cipher.

Definition 7.7.1 *Consider an r-round iterated $2m$-bit block cipher with r round keys, each of length n bits. A* **strong key schedule** *has the following properties*

1. *Given any s bits of the r round keys, derived from an unknown master key, where $s < rn$, it is* 'hard' *to find any of the remaining $rn - s$ key bits from the s known bits.*

2. *Given some relation between two master keys it is* 'difficult' *to predict the relations between any of the round keys derived from the two master keys.*

The terms 'hard' and 'difficult' can be replaced by more precise definitions depending on the applications. Of course 'hard' cannot be harder than performing the key schedule for all keys, and 'difficult' cannot be more difficult than performing the key schedule for the two master keys.

The above properties will complicate differential and linear attacks and thwart the attacks based on simple relations discussed in Section 5.4.2.

A simple design of a strong key schedule

Let $E_K(\cdot)$ be an r-round Feistel cipher of block length $2m$ bits, using master key K for which the r round keys are of length n bits each and $n \leq 2m$.

1. Define an initial key schedule, which on input a master key K outputs r dependent round keys $\{K_i\} = K_1, \ldots, K_r$, s.t.

   (a) $E_{\{K_i\}}(\cdot)$ is secure against a known plaintext attack using encryptions of at most r known plaintexts, in the sense that an information deduction (see page 45) with a non-trivial information gain is not possible.

   (b) $E_{\{K_i\}}(\cdot)$ contains no simple relations where $g_1(P, K) = P \oplus \alpha$, α a constant, see Definition 5.4.3, page 90.

2. Define the round keys $\{RK_l\} = RK_1, \ldots, RK_r$ used for encryption as

$$RK_l = nMSB(E_{\{K_i\}}(IV \oplus l)),$$

where IV is a fixed value and $nMSB(X)$ denotes the n leftmost bits of X.

At a first glance it may seem strange and difficult to construct an initial key schedule yielding a cipher secure against a known plaintext attack and with no simple relations. However, for a 16 round cipher, as an example, it does not seem difficult to prove or at least be strongly convinced that the obtained cipher is secure against an attack using only 16 encryptions of known plaintexts and the condition on the simple relations is easy to meet. For a 16 round cipher the relation in (1b) would be $g_1(P) = P \oplus h$, h a hex digit, so this relation would not even hold for a cipher with the complementation property, the most well-known simple relation. As an example of such an initial key schedule, see the key schedules of the DES [90] and the LOKI ciphers [15, 14]. We can prove

Theorem 7.7.1 *The key schedule just defined is a strong key schedule, where 'hard' means as hard as a brute force attack on $E_{\{K_i\}}(\cdot)$ and 'difficult' means as difficult as one encryption of $E_{\{K_i\}}(\cdot)$. Furthermore the absence of weak keys in guaranteed and pairs of semi-weak keys are very unlikely to occur.*

Proof: By contradiction. Assume that property 1 of Definition 7.7.1 can be compromised faster than exhaustive search for all keys of $E_{\{K_i\}}(\cdot)$. This means, that given s bits of the set $\{RK_l\}$, which are ciphertext bits corresponding to less than r encryptions $E_{\{K_i\}}(IV \oplus l)$, it is possible in time less than brute force to find (bits of) ciphertexts, which were not given to us. But that yields a contradiction because of (1a).

Assume that property 2 of Definition 7.7.1 can be compromised faster than one encryption of $E_{\{K_i\}}(\cdot)$. This means, that we can find some relation between two master keys, K and K^*, s.t. $f(K) = K^*$ and some relation between two round keys, RK_l and RK_n^*, s.t. $g_2(RK_L) = RK_n^*$, where the total complexity of f and g_2 is less than that of one encryption of $E_{\{K_i\}}(\cdot)$. But that yields a contradiction because of (1b) and Definition 5.4.3, since then

$$E_{\{K_i\}}(P) = C \Rightarrow E_{f(\{K_i\})}(P \oplus (l \oplus n)) = g_2(C)$$

where $P = IV \oplus l$ and $C = RK_l$.

To prove the final statements we note that $RK_l \neq RK_n$ for $l \neq n$, i.e., there are no weak keys. Furthermore it is very unlikely that we can find pairs of semi-weak keys, K and K^*, s.t. $E_K(IV \oplus l) = E_{K^*}(IV \oplus (r+1-l))$ for all $l = 1, \ldots, r$. □

The above method applied to the DES may yield a DES-version with improved immunity to differential, linear and other attacks. However, this DES-version is only 16 times harder to break than the DES by exhaustive search of all keys and in view of [112] a larger master key is needed. A possibility would be to define the round keys as follows:

$$RK_i = 48MSB(DES_{K1}(DES^{-1}_{K2}(DES_{K1}(IV \oplus i)))),$$

i.e., use two-key triple DES to calculate the new round keys.

The above method involves encryptions in the generation of the round keys, but note that encryption with these ciphers is as fast as encryption with the same cipher using a conventional key schedule when the key is held constant (see also [100]).

7.8 A Test for Nonlinear Order

In [103] it was considered to cryptanalyzed the DES by the method of formal coding. The conclusion was that this is hardly possible. It was also shown that the nonlinear order of any of the 8 S-boxes in the DES is 5. An open question is: what is the order of the outputs for the full 16-round DES? In general, a cipher will be vulnerable to attacks like the method of formal coding if the nonlinear order of the outputs is too low. Higher order differentials can be used to determine a lower bound on the nonlinear order of a block cipher.

Test for nonlinear order
Input: $E_K(\cdot)$, a block cipher, a key K, plaintexts $x_1 \neq x_2$ and r, an integer.
Output: $i \leq r$, a minimum nonlinear order of E_K.
Let $a_1, a_2, \ldots, a_i$ be linearly independent.

1. Set $i = 1$
2. Compute $y_1 = \Delta_{a_1,\ldots,a_i} E_K(x_1)$ and $y_2 = \Delta_{a_1,\ldots a_i} E_K(x_2)$

3. If $y_1 = y_2$ output i and stop

4. If $i \geq r$ output i and stop

5. Set $i = i + 1$ and go to step (2)

If in step (3), $y_1 \neq y_2$ then the nonlinear order is greater than i according to Proposition 5.2.3. If $y_1 = y_2$ then the nonlinear order may be greater than i, because it is possible for other values of x_1' and x_2' that $y_1' \neq y_2'$. However, the above test must stop, since if the i'th derivative of f is constant, then the $(i + r)$'th derivative of f is zero for all $r > 0$. Also, note that computing an i'th order derivative of f, is equivalent to computing two times an $(i - 1)$'th order derivative of f. Therefore the values of y_1, y_2 can be stored and re-used in following steps.

To test a block cipher E pick a random key K and two random plaintexts and run the test for nonlinear order. If the output of the test is d then the nonlinear order of E_K is at least d. Repeat this procedure for as many keys and plaintexts as desired. The input r and the test in step (4) is necessary for block ciphers like the DES and r should be chosen not much greater than 32, since it takes about 2^r encryptions to check a nonlinear order of r.

7.9 Cascade Ciphers

In Section 6.1.6 we discussed the future of DES. We are in the situation, where we have a block cipher, that has proved to be very strong, the only problem being that the keys are too small and a simple brute-force attack has become possible. Thus, this section is motivated by the following general question: Given cryptosystem $\mathcal{X}$, which cannot in practice be broken faster than exhaustive key search, how can we build a new system $\mathcal{Y}$, such that

1. Keys in $\mathcal{Y}$ are significantly longer than keys in $\mathcal{X}$ (e.g. twice as long)

2. Given an appropriate assumption about the security of $\mathcal{X}, \mathcal{Y}$ is provably as hard to break as $\mathcal{X}$ under any natural attack (e.g. ciphertext only, known plaintext, etc.).

3. It can be convincingly argued that $\mathcal{Y}$ can in fact not be broken faster than exhaustive key search, and is therefore in fact much stronger than $\mathcal{X}$.

Possible answers to this question have already appeared in the literature. The best known example is two-key triple encryption, where we encipher using one key, decipher using a second key, and finally encipher using the first key. Van Oorschot and Wiener [111] have shown, refining an attack of Merkle and Hellman [76], that this construction is not optimal: in a known plaintext attack, it can be broken significantly faster than exhaustive key search.

We propose a new variant of two-key triple encryption, which has all the properties we require above.

7.9.1 Multiple encryption

In this section, we look at methods for enhancing cryptosystems based on the idea of encrypting plaintext blocks more than once. Following the notation of the introduction, we let $\mathcal{X}$ be the original system, and we let E_K resp. D_K denote encryption resp. decryption in $\mathcal{X}$ under key K. We assume that the key space of $\mathcal{X}$ consists of all k-bit strings, and that the block length of $\mathcal{X}$ is m. In a *cascade of ciphers* it is assumed that the keys of the component ciphers are independent. The following result was proved by Maurer and Massey [69].

Theorem 7.9.1 (The importance of being first.) *A cascade of ciphers is at least as hard to break as the first cipher.*

By restricting ourselves to the most powerful attack, the chosen plaintext attack, we can prove the following more general result.

Theorem 7.9.2 (The importance of being there.) *A cascade of ciphers is at least as hard to break in any attack as any of the component ciphers in the cascade in a chosen plaintext attack.*

Proof: Assume that we have an algorithm A, which on input the encryptions of n known or chosen plaintexts or on input just n ciphertexts, breaks

a cascade of N_c ciphers, $\mathcal{Y}$. We will use A to break any of the component ciphers in a chosen plaintext attack. Assume that $\mathcal{X}$ is the i'th cipher of the N_c ciphers in the cascade and that we can get encryptions of any chosen plaintext. Choose $N_c - 1$ keys at random for the ciphers exclusive $\mathcal{X}$. Whenever A asks for the encryption of a chosen or known plaintext P, we multiple encrypt P using the first $i-1$ keys, yielding PP. In a ciphertext only attack we choose a plaintext P ourselves. Then we get the encryption CC of PP in the chosen plaintext setting from $\mathcal{X}$. Now use the remaining $N_c - i$ keys to multiple encrypt CC, yielding C, which we input to A. Since by assumption, A breaks the cascade, it will output the N_c keys, amongst which we will get a candidate for the secret key of $\mathcal{X}$. We have proved that if we can break the cascade, we can break any of the component ciphers in a chosen plaintext attack. Thus, if a component cipher $\mathcal{X}$ is secure against a chosen plaintext attack, then a cascade of ciphers containing $\mathcal{X}$ is secure against any attack. □

A special case of a cascade of ciphers is when the component ciphers are equal, also called multiple encryption. In the following we consider different forms of multiple encryption.

Double Encryption

The simplest idea one could think of would be to encrypt twice using two keys K_1, K_2, i.e., let the ciphertext corresponding to P be $C = E_{K_2}(E_{K_1}(P))$. It is clear (and well-known), however, that no matter how K_1, K_2 are generated, there is a simple meet-in-the middle attack that breaks this system in a known plaintext attack using 2^k encryptions and 2^k blocks of memory, i.e. the same time complexity as key search in the original system. Even though the memory requirements may be unrealistic, it is clear that this is not a satisfactory improvement over $\mathcal{X}$.

Triple Encryption

Triple encryption with three independent keys K_1, K_2, and K_3, where the ciphertext corresponding to P is $C = E_{K_3}(E_{K_2}(E_{K_1}(P)))$, is also not a satisfactory solution for a similar reason as for double encryption. A simple meet-in-the-middle attack will break this in time about 2^{2k} encryptions and

space 2^k blocks of memory. Thus we do not get full return for our effort in tripling the key length - as stated in demand 3 in the introduction, we would like attacks to take time close to 2^{3k}, if the key length is $3k$. In addition to this, if $\mathcal{X} = DES$, then a simple triple encryption would preserve the complementation property, and preserve the existence of weak keys.

It is clear, however, that no matter how the three keys in triple encryption are generated, the meet-in-the-middle attack mentioned is still possible, and so the time complexity of the best attack against *any* triple encryption variant is not larger than 2^{2k}. It therefore seems reasonable to try to generate the three keys from two independent $\mathcal{X}$-keys K_1, K_2, since triple encryption will not provide security equivalent to more than 2 keys anyway.

Two-key Triple Encryption

One variant of this idea is well-known as two-key triple encryption, proposed by W. Tuchmann [110]: we let the ciphertext corresponding to P be $E_{K_1}(D_{K_2}(E_{K_1}(P)))$. Compatibility with a single encryption can be obtained by setting $K_1 = K_2$. As one can see, this scheme uses a particular, very simple way of generating the three keys from K_1, K_2. For two-key triple encryption there is a result similar to Theorem 7.9.2.

Theorem 7.9.3 *In a chosen plaintext/ciphertext attack two-key triple encryption is at least as hard to break as the underlying cipher.*

Proof: Assume that we have an algorithm B, which on input n chosen plaintexts, breaks a two-key triple encryption scheme, $\mathcal{Z}$, where $\mathcal{W}$ is the underlying cipher. Choose one key $K_{1,3}$ at random. Whenever B asks for the encryption of plaintext P, we encrypt P using the key $K_{1,3}$, yielding PP. Then we get the decryption CC of PP in the chosen ciphertext setting from $\mathcal{W}$. Now encrypt CC using again the key $K_{1,3}$ yielding C, which is input to B. Since by assumption B breaks the two-key triple scheme, it will output a candidate for the key in the second round, i.e., for the secret key of $\mathcal{W}$. □

Even though this result establishes some connection between the security of two-key triple encryption with the underlying cipher, it holds only for a chosen plaintext/ciphertext attack and still does not meet our second demand.

For the two-key triple encryption scheme, each of K_1 and K_2 only influences particular parts of the encryption process. Because of this, variants of the meet-in-the-middle attack are possible that are even faster than exhaustive search for K_1, K_2. In [76] Merkle and Hellman describes an attack on two-key triple DES encryption requiring 2^{56} chosen plaintext-ciphertext pairs and a running time of 2^{56} encryptions using 2^{56} words of memory. This attack was refined in [111] into a known plaintext attack on the DES, which on input n plaintext-ciphertext pairs finds the secret key in time $2^{120}/n$ using n words of memory. The attacks can be applied to any block cipher.

Since the attacks exploit that the keys used in the first and third encryption are equal, an initial attempt to thwart the attacks could be to let the third key be dependent on both the first and second key. Define encryption by $E_{K_1}(D_{K_2}(E_{K_3}(P)))$, where $K_3 = E_{K_1}(K_2) \oplus K_2$. Compatibility with a single encryption can still be obtained by setting $K_2 = D_{K_1}(0)$, in that way $K_2 = K_3$. By the security of the DM-scheme, see (3.1) on page 29, knowing K_1 (or K_2) does not give immediate knowledge about K_3 and vice versa and the scheme seems invulnerable to the attacks by Merkle and Hellman. However, we found no proof that this scheme is at least as secure as a single encryption.

We therefore propose what we believe to be stronger methods for generating the keys. Our main idea is to generate them *pseudorandomly* from 2 $\mathcal{X}$ keys using a generator based on the security of $\mathcal{X}$. In this way, an enemy trying to break $\mathcal{Y}$ either has to treat the 3 keys as if they were really random which means he has to break $\mathcal{X}$, according to Theorem 7.9.1; or he has to use the dependency between the keys – this mean breaking the generator which was also based on $\mathcal{X}$! Thus, even though we have thwarted attacks like Merkle-Hellman and van Oorschot-Wiener by having a strong interdependency between the keys, we can still, if $\mathcal{X}$ is secure enough, get a connection between security of $\mathcal{X}$ and $\mathcal{Y}$.

General Description of $\mathcal{Y}$

Let a block cipher $\mathcal{X}$ be given, as described above. The key length of $\mathcal{X}$ is denoted by k. By $E_K(P)$, we denote $\mathcal{X}$-encryption under K of block P, while $D_K(C)$ denotes decryption of C. We then define a new block cipher $\mathcal{Y}$

using a function G:

$$G(K_1, K_2) = (X_1, X_2, X_3)$$

which maps 2 $\mathcal{X}$-keys to 3 $\mathcal{X}$-keys. We display later a concrete example of a possible G-function. This is constructed from a few $\mathcal{X}$-encryptions. Keys in $\mathcal{Y}$ will consist of pairs (K_1, K_2) of $\mathcal{X}$-keys. Encryption in $\mathcal{Y}$ is defined by

$$E_{K_1,K_2}(P) = E_{X_3}(E_{X_2}(E_{X_1})P)))$$

where $(X_1, X_2, X_3) = G(K_1, K_2)$. Decryption is clearly possible by decrypting using the X_i's in reverse order.

Relation to the security of $\mathcal{X}$

We would like to be reasonably sure that we have taken real advantage of the strength of $\mathcal{X}$ when designing $\mathcal{Y}$. One way of stating this is to say that $\mathcal{Y}$ is at least as hard to break as $\mathcal{X}$. By Theorem 7.9.1, this would be trivially true if the three keys used in $\mathcal{Y}$ were statistically independent. This is of course not the case, since the X_i's are generated from only 2 keys. But if the generating function G has a pseudorandom property as stated below, then the X_i's are "as good as random" and we can still prove the result we want.

Definition 7.9.1 *Consider the following experiment: an enemy B is presented with three k-bit blocks X_1, X_2, X_3. He then tries to guess which of two cases has occurred:*

1. *The X_i's are chosen independently at random.*
2. *The X_i's are equal to $G(K_1, K_2)$, for randomly chosen K_1, K_2.*

Let p_1 be the probability that B guesses 1 given that case 1 occurs, and p_2 the probability that B guesses 1 given that case 2 occurs. The generator function G is said to be pseudorandom, *if for any B spending time equal to T encryption operations,*

$$|p_1 - P_2| \leq \frac{T}{V} ,$$

where V is the total number of possible values of the pair (K_1, K_2).

The intuition behind this is that B could always spend his time simply trying random pairs of keys, seeing if they could be a possible value of K_1, K_2, and guessing that he is in case 2 if he finds a solution. If case 2 really occurs, he finds the right value with probability at most T/V (we assume here that he would need at least one encryption to test a pair). In case 1 there is most likely no solution. Thus the definition says that if G is pseudorandom, there is no better method for B than this naive attack. Definition 7.9.1 is inspired by the complexity theoretic definition of a strong pseudorandom generator introduced by Blum and Micali [9].

In the rest of this subsection we consider attacks against $\mathcal{X}$ and $\mathcal{Y}$ in a fixed scenario with a given plaintext distribution and a given form of attack, such as known plaintext, chosen plaintext, etc. We do not specify these things further, because the reasoning below will work for any such scenario. The time unit will be encryptions in system $\mathcal{X}$.

The next theorem shows the promised connection between security of $\mathcal{X}$ and $\mathcal{Y}$, i.e., in a given amount of time, an attack cannot do much better against $\mathcal{Y}$ than what is possible against $\mathcal{X}$.

Theorem 7.9.4 *Let p be the success probability of the bedst attack against $\mathcal{X}$ running in time T. Assume now that an attacker A against our new system $\mathcal{Y}$ runs in time T and has success probability $p + \epsilon$. If the function G used to construct $\mathcal{Y}$ is pseudorandom, then*

$$\epsilon \leq \frac{T}{V} ,$$

Proof: Let $\mathcal{Y}_0$ be the same system as $\mathcal{Y}$, but with independent keys X_i. By Theorem 7.9.1. using A against $\mathcal{Y}_0$ leads to an attack against $\mathcal{X}$ with the same success probability. Hence by assumption, A's success probability against $\mathcal{Y}_0$ will be at most p. But then we can use A to make an algorithm B that fits Definition 7.9.1: Given X_1, X_2, X_3, B uses these as keys in the triple encryption system and simulates A's attack. If A is successful, B will guess that the X_i's are generated from K_1, K_2, if not, B will guess that they are independent. Since in one case A will be attacking $\mathcal{Y}$, and in the other case $\mathcal{Y}_0$, it is clear that for this B, we have by Definition 7.9.1

$$\epsilon \leq |p_1 - p_2| \leq \frac{T}{V} ,$$

□

As an example of what the statement of the theorem means, consider an ideal case, where the best an attack against $\mathcal{X}$ can do, is to spend its time choosing random keys and test whether they fit with the information available. The success probability for time T would then be $T/2^k$ assuming a key can be tested in 1 encryption. Then the above theorem says that if G is pseudorandom, the success probability of any attack against $\mathcal{Y}$ running in time T can be at most $Y/2^k + T/2^{2k}$. This is larger than the original success probability against $\mathcal{X}$ by a factor of only $1 + 2^{-k}$.

A Concrete Two-key Triple Encryption Construction

We propose here a new construction for triple encryption, called **TEMK** for **T**riple **E**ncryption with **M**inimum **K**ey. In this construction the keys X_1, X_2, X_3 are all used for encryption. We define this construction of

$$G(K_1, K_2) = (X_1, X_2, X_3)$$

by

$$\begin{aligned} X_1 &= E_{K1}(D_{K2}(E_{K1}(IV_1))) \\ X_2 &= E_{K1}(D_{K2}(E_{K1}(IV_2))) \\ X_3 &= E_{K1}(D_{K2}(E_{K1}(IV_3))) \end{aligned}$$

where IV_i are three initial values, e.g. $IV_i = C + i$ where C is a constant. It is seen that two-key triple encryption is used.

Here, the reader may ask a (very legitimate) question: why are we using ordinary two-key triple DES here, when we have just spent half a paper arguing that it does not provide good enough security? The answer is that we are using two-key triple DES in a special situation where we can guarantee that for any particular pair of keys, the enemy will get at most a known plaintext attack with three known plaintexts. This follows from the fact that the three constants IV_1, IV_2, IV_3 are universally fixed, such that the pair of keys K_1, K_2 will never be applied to anything else than the IV_i's. The best known attack against two-key triple DES with three known plaintexts is the one by van Oorschot and Wiener [111], which has the complexity $2^{120}/3 \simeq 1.3 \times 2^{118}$. Since in our case the keys are only 112 bits, we conjecture

Scheme	Key size	# KS	# EN	Total	W.k.	C.p.
TEMK-DES	112	5	9	19	No	No
DES	56	1	0	2	Yes	Yes
Two-key triple-DES	112	2	0	4	Yes	Yes
Three-key triple-DES	168	3	0	6	Yes	Yes

Table 7.5: Comparison of the proposed scheme and the existing ones, all used with DES.

that this G is pseudorandom with the value $V = 2^{112}$. The most natural attack against pseudorandomness of G seems to be to guess either K_1 or K_2 and try to find the other value faster than exhaustive search.

The key scheduling in the above construction is slower than for the two-key triple encryption. In most software applications of the DES the key scheduling takes about twice the time of a single encryption. Using this estimate the key scheduling in the triple encryption scheme above takes time about 19 DES-encryptions. For comparison the key schedules for two-key triple DES and triple DES with three independent keys take 4 and 6 encryptions, respectively. In encryption with our new construction the key schedule should be performed once and the three round keys stored. In that way encryption with TEMK-DES is as fast as for other triple encryption schemes with fixed keys.

We conjecture that for the above construction, the fastest attack is a simple meet in the middle attack, which will be of time complexity at least 2^{2k}. In particular we conjecture that because of the strong interdependency between the X_i's, attacks like the ones from [76, 111] will not be possible. Finally we note that the absence of weak keys is guaranteed, since the three round keys are never equal and the complementation property does not hold. In Table 7.5 we give a schematic overview of the differences between our proposed scheme and the existing ones. KS and EN are the numbers of DES key schedules and DES encryptions respectively, needed in the key schedule of the triple encryption scheme. 'Total' is the total number of encryptions in the new key schedule using the above estimate. Finally we state if weak keys exist and if the complementation property holds.

Extensions

In the preceding sections we focused on triple encryption schemes. It is clear that our ideas can be extended to quadruple, quintuple, ... , n-fold schemes. Let $\mathcal{X}$ be a component cipher with a key size k. In general a $2i$-fold encryption scheme based on $\mathcal{X}$ is vulnerable to a meet in a middle attack using 2^{ik} words of memory taking time about 2^{ik} encryptions. Similarly, a $(2i+1)$-fold encryption scheme based on $\mathcal{X}$ is vulnerable to a meet in a middle attack using 2^{ik} words of memory taking time about $2^{(i+1)k}$. Therefore one does not get the security of the full key length. It is obvious that by generating the $2i$ $(2i+1)$ keys pseudorandomly, defined in a similar manner as Definition 7.9.1, from i $(i+1)$ keys one can prove a similar result as Theorem 7.9.4.

Chapter 8

Cryptanalysis of Hash Functions

In this chapter we consider cryptanalysis of hash functions based on a block cipher. In Section 8.1 we give attacks on a large class of hash functions based on a block cipher. We show that there exist attacks on all double block length hash functions of hash rate 1, according to Definition 3.2.7 and Definition 3.2.8, that is, iterated hash functions where in each round the block cipher is used twice, s.t. one message block requires one encryption of the underlying block cipher. This result suggests, that it is difficult to construct a more efficient scheme with security equivalent to the hash rate 1/2 scheme, the MDC-2 [77, 10], which is published as an ISO/IEC standard [38]. In Section 8.2 we apply our results to the specific double block length hash functions, the Parallel-DM [37], the PBGV hash function [96] and the LOKI-DBH function [15]. Also we give attacks (not related to the above) on the AR Hash Function proposed in [60], that breaks the scheme. In Section 8.3 we give a new kind of characteristics for iterated block ciphers, which can be used in differential attacks on hash functions based on a block cipher and give the best possible such characteristic for the DES. To our knowledge this is the best characteristic for the DES, where the input difference equals the output difference.

8.1 The Solving One-half Attack

In this section we consider a double block length hash function based on a block cipher according to Definition 3.2.8. For double block length hash functions of rate 1/2 or 1 the following upper bounds exist.

Theorem 8.1.1 (HLMW-93 [37]) *For a $2m$-bit iterated double block length hash function with hash rate $1/2$ or 1, the complexity of a free-start second preimage attack is upper bounded by about 2^m and the complexity of a free-start collision attack is upper bounded by about $2^{m/2}$. The attacks succeed with probability about 0.63.*

Hash functions obtaining these upper bounds as lower bounds for the free-start attacks are said to be *optimum* against a free-start attack [37]. The idea is, that given a specific initial value of the hash function the designer hopes that the complexities of real collision and real second preimage attacks are higher than the proven lower bounds for free-start attacks.

The basic idea behind the attacks in the proof of the Theorem 8.1.1 is to attack the two equations in Definition 3.2.8 separately. Such a method of "separately attacking the two functions" can also be used in real attacks, namely, the **solving-one-half** attacks, as we will show in the following.

Theorem 8.1.2 (Solving one-half attack (parallel)) *Consider a double block length hash function of rate 1 with hash round function of the form (8.1), where each h^i contains one encryption.*

$$\begin{cases} H_i^1 &= h^1(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2) \\ H_i^2 &= h^2(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2) \end{cases} \tag{8.1}$$

If for a fixed value of H_i^1 or H_i^2 or $H_i^1 \oplus H_i^2$), it takes T operations to find one pair of (M_i^1, M_i^2) for any given value of (H_{i-1}^1, H_{i-1}^2), such that the resulting 4-tuple $(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2)$ yields the fixed value for H_i^1 (or H_i^2 or $H_i^1 \oplus H_i^2$), then a second preimage attack on the hash function needs at most $(T+3) \cdot 2^m$ operations; and a collision attack on the hash function needs at most $(T+3) \cdot 2^{m/2}$ operations. The attacks succeed with probability about 0.63.

Proof: The second preimage attack: Let (H_0^1, H_0^2) be the given initial value and (H_n^1, H_n^2) be the hash code of a message M. We proceed as follows:

1. Compute forward the pair (H^1_{n-1}, H^2_{n-1}) from the given hash value (H^1_{n-2}, H^2_{n-2}) and a randomly chosen pair of messages (M^1_{n-1}, M^2_{n-1}).
2. Find the pair (M^1_n, M^2_n) from the pair (H^1_{n-1}, H^2_{n-1}) obtained above so that the 4-tuple $(H^1_{n-1}, H^2_{n-1}, M^1_n, M^2_n)$ yields the fixed value for H^1_n.
3. Compute the value for H^2_n from the 4-tuple $(H^1_{n-1}, H^2_{n-1}, M^1_n, M^2_n)$.

Repeat the above procedure 2^m times. Note that H^2_n is m bits long, so after obtaining 2^m values of H^2_n, with a high probability we hit the given value of H^2_n. Finally, note that step 1 takes two operations, step 2 takes T operations and step 3 takes one operation.

The collision attack: Let (H^1_0, H^2_0) be the given initial value. We shall find two different messages M and M', such that both messages yield the same hash code (H^1_n, H^2_n). Choose some random values and compute a value for H^1_n and fix it, then proceed in the same way as in the second preimage attack, i.e., perform steps 1, 2 and 3 above. Repeat this procedure $2^{m/2}$ times. Because H^2_n is m bits long, the "birthday argument" implies that some two values of the H^2_n will be the same with high probability. One can use the method of *distinguished points* by Quisquater and Delescaille, see Section 1.1.1, to find collisions using only negligible memory.

We note that the messages found in these attacks are of the same length. Therefore the MD-strengthening is of no importance. The case of collision attacks is obvious and for second preimage attacks, it follows that given a message M and the hash code $H(M)$ we can compute the hash code of the message M without the blocks containing the message length and proceed from there. □

The result of Theorem 8.1.2 is for the parallel form of a double block length hash function in which the two encryptions work side-by-side. Similar attacks can be applied to the serial form in which one encryption is computed after the other.

Theorem 8.1.3 (Solving one-half attack (serial)) *Consider a double block length hash finction of rate 1 with round unction of the form (8.2), where each h^i contains one encryption.*

$$\begin{cases} H^1_i &= h^1(H^1_{i-1}, H^2_{i-1}, M^1_i, M^2_i) \\ H^2_i &= h^2(H^1_{i-1}, H^2_{i-1}, M^1_i, M^2_i, H^1_i) \end{cases} \tag{8.2}$$

If for a fixed value of H_i^1, it takes T operations to and one pair of (M_i^1, M_i^2) for any given value of (H_{i-1}^1, H_{i-1}^2), such that the resulting 4-tuple $(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2)$ yields the fixed value for H_i^1, then a second preimage attack on the hash function needs at most $(T+3)\cdot 2^m$ operations; and a collision attack on the hash function needs at most $(T+3)\cdot 2^{m/2}$ operations. The attacks succeed with probability about 0.63.

8.1.1 Attacks on a large class of double block length hash functions of hash rate 1

In [95] it was shown that there exist basically two secure single block length hash functions. The DM-scheme (3.1) is one of them, the other one is the following

$$H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1} \oplus M_i \tag{8.3}$$

All other secure single block length hash functions can be transformed into either (3.1) or (8.3) by a linear transformations of the inputs M_i and H_{i-1} [95]. It means that for a double block length hash function one can obtain *optimum* security against free-start attacks if the scheme is equivalent to either two runs of (3.1) or two runs of (8.3) by a simple invertible transformation of the inputs M_i^1, M_i^2, H_{i-1}^1 and H_{i-1}^2. In the following we will show that all double block length hash functions of rate 1, for which (at least) one of two hash round functions has the form of either (3.Q page 29, or (8.3), the solving-one-half attack is applicable with $T \simeq 0$. First we write a double block length hash function, Definition 3.2.8, as follows

$$\left\{ \begin{array}{rcccl} H_i^1 & = & E_A(B) & \oplus & C \\ H_i^2 & = & E_R(S) & \oplus & T \end{array} \right. \tag{8.4}$$

where, for a rate 1 scheme, A, B and C are binary linear combinations of the m-bit vectors $H_{i-1}^1, H_{i-1}^2, M_i^1$ and M_i^2, and where R, S and T are some binary linear combinations of the vectors $H_{i-1}^1, H_{i-1}^2, M_i^1$ and M_i^2. For a rate $1/2$ scheme, A, B and C are binary linear combinations of the m-bit vectors $H_{i-1}^1, H_{i-1}^2, M_i$ and R, S and T are some combinations of the m-bit vectors $H_{i-1}^1, H_{i-1}^2, M_i$ and H_i^1.

We can write, in case of a rate 1 scheme, A, B and C in matrix-form as

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \end{bmatrix} \begin{bmatrix} H_{i-1}^1 \\ H_{i-1}^2 \\ M_i^1 \\ M_i^2 \end{bmatrix} \tag{8.5}$$

for some binary values a_i, b_i and $c_i (1 \leq i \leq 4)$.

We denote by L the 3×4 matrix in (8.5). We distinguish between cases depending on the rank of matrix L. We consider first all double block length hash functions, for which the matrix L has rank less than 3. This case includes the schemes for which one of the hash round functions is equivalent up to a linear transformation to either (3.1) or (8.3). We can prove the following result.

Theorem 8.1.4 *For the $2m$-bit iterated hash finction with rate 1, for which (at least) one of the round functions the matrix L of (8.5) has a rank of less than or equal to two, the complexity of a second preimage attack is upper bounded by about 3×2^m encryptions, and the complexity of a collision attack is upper bounded by about $3 \times 2^{m/2}$. The attacks succeed with probability about 0.63.*

Proof: We will show that the T of Theorem 8.1.3 is about zero. We assume w.l.o.g. that the hash round functions of type (8.5) is H_i^1 and that we are given the target (H_n^1, H_n^2). *Rank*$(L) = 1$: Tivial, since with the same intermediate hash values (H_{n-1}^1, H_{n-1}^2) used in the computation of the target H_n^1, there are at least 2^m possible values of (M_n^1, M_n^2) obtaining H_n^1. Thus, Theorem 8.1.3 holds with $T \simeq 0$.
Rank$(L) = 2$: We can rewrite (8.5) as follows

$$\begin{bmatrix} A \\ B \end{bmatrix} = N_1 \begin{bmatrix} H_{i-1}^1 \\ H_{i-1}^2 \end{bmatrix} \oplus N_2 \begin{bmatrix} M_i^1 \\ M_i^2 \end{bmatrix} \tag{8.6}$$

where N_1 and N_2 are 2×2 binary matrices. We distinguish between cases depending on the rank of N_2.

Rank$(N_2) \leq 1$: Then with the same intermediate hash values (H_{n-1}^1, H_{n-1}^2) used in the computation of the target H_n^1, there

are at least 2^m possible values of (M_n^1, M_n^2) obtaining H_n^1. Thus, Theorem 8.1.3 holds with $T \simeq 0$.

$Rank(N_2) = 2$: N_2 is invertible and we can rewrite (8.6) into

$$\begin{bmatrix} M_i^1 \\ M_i^2 \end{bmatrix} = N_2^{-1} \left[N_1 \begin{bmatrix} H_{i-1}^1 \\ H_{i-1}^2 \end{bmatrix} \oplus \begin{bmatrix} A \\ B \end{bmatrix} \right] \tag{8.7}$$

Given the target H_n^1 and by letting (A, B) be the same values used in the computation of the target H_n^1, we can find $(M_n^1, M_n^2)'$ for any values (H_{n-1}^1, H_{n-1}^2), such that we hit the target H_n^1. Thus, Theorem 8.1.3 holds with $T \simeq 0$ (time used to do the additions is negligible and the inversion of the matrix N_2 has to be done only once).

□

The conclusion of this section is that a double block length hash function of hash rate 1 which by a linear transformation of the inputs is equivalent to a secure single block length hash function, i.e., optimum with respect to free-start attacks, is vulnerable to the solving one half attack with a complexity about the same as the complexity of the free-start attacks, which means that one does not gain (much) security by doubling the block length.

8.1.2 Attacks on all double block length hash functions of hash rate 1

In the following we will consider any double block length hash functions of hash rate 1, that is, we consider equation (8.5)

Theorem 8.1.5 *For the double block length hash functions of hash rate 1, for which one of the m-bit hash round functions is of type (8.5), the complexity of a second preimage attack ia upper bounded by about 4×2^m and the complexity of a collision attack is upper bounded by about $4 \times 2^{m/2}$. However in three cases, the attack needs a pre-computed table with 2^m $2m$-bit values. For these cases only the complexities for the preimage attacks hold. The attacks succeed with probability about 0.63.*

Proof: We will show that the T of Theorem 8.1.3 is at most 1. The case

where $Rank(L) < 3$ is proved in Theorem 8.1.4.
$Rank(L) = 3$: Assume w.l.o.g. that first hash round function in this scheme has the form $H_i^1 = E_A(B) \oplus C$, where A, B and C are linearly independent. A and B can be expressed as in (8.6). We split the proof into two cases.

1. $Rank(N_2) = 1$. Let M_Z be the set $\{M_i^1, M_i^2, M_i^1 \oplus M_i^2\}$ and let $M_{ab} \in M_Z$ be the message variable contained in A and B. If C does not contain any of the messages in M_Z or contains only M_{ab}, Theorem 8.1.3 holds with $T \simeq 0$, since in this case we use the same intermediate values (H_{n-1}^1, H_{n-1}^2) used in the computation of the target H_n^1 (i.e., use the same messages $M_1, \ldots, M_{n-1}$). Since the rank of N_2 is one, for a fixed value of M_{ab} there are still 2^m possible values of (M_n^1, M_n^2) obtaining the hash code H_n^1.

   If C contains one message $M_c \in M_Z$, such that $M_c \neq M_{ab}$ then for any given (H_{n-1}^1, H_{n-1}^2), compute $E_A(B) = z$ for a random value of M_{ab}. Now use the correct value of the 2^m possible values of M_c to hit H_n^1, i.e., such that $C \oplus z = H_n^1$. In this case Theorem 8.1.3 holds with $T \simeq 1$.

2. $Rank(N_2) = 2$. H_i^1 can be written

$$\begin{aligned} H_i^1 &= E_A(B) \oplus C^0 \\ &= E_A(B) \oplus B \oplus C^1 \\ &= E_A(B) \oplus A \oplus C^2 \\ &= E_A(B) \oplus A \oplus B \oplus C^3 \end{aligned}$$

   Since the rank of L is 3 and the rank of N_2 is 2, either C^0, C^1, C^2 or C^3 does not contain any of the messages M^1, M^2 or $M^1 \oplus M^2$. Let C^i denote that value of C.

   (a) $C^i = C^0$. It is possible for any given value of (H_{n-1}^1, H_{n-1}^2) and thereby also for C^0, to find (M_n^1, M_n^2) such that the target H_n^1 is hit. Simply decrypt $D_A(C^0 \oplus H_n^1) = B$ using one of the two free message variables in A and using the other free message variable to adjust to the given (H_{n-1}^1, H_{n-1}^2) appearing in B. Again Theorem 8.1.3 holds with $T \simeq 1$.

(b) $C^i = C^1$. We first pre-compute (and sort) a table KT of 2^m triples (K_l, x_l, y_l), such that

$$K_l = E_{x_l}(y_l) \oplus y_l$$

for random values (x_l, y_l). Then for any given (H^1_{n-1}, H^2_{n-1}) compute $Q = C^1 \oplus H^1_n$. Look up $Q = K_j$ in table KT and set $A = x_j$ and set $B = y_j$ for A and B in equation (8.6). Since N_2 is invertible, by assumption, we find the values of (M^1_n, M^2_n), such that the target H^1_n is hit. Theorem 8.1.3 holds with $T \simeq 0$. We have assumed here that the time to sort a table of size 2^m is negligible compared to the time of 2^m encryptions.

(c) $C^i = C^2$. We first pre-compute (and sort) a table KT of 2^m triples (K_l, x_l, y_l), such that

$$K_l = E_{x_l}(y_l) \oplus x_l$$

for random values (x_l, y_l) and proceed similar as in the case where $C^i = C^1$.

(d) $C^i = C^3$. We first pre-compute (and sort) a table KT of 2^m triples (K_l, x_l, y_l), such that

$$K_l = E_{x_l}(y_l) \oplus x_l \oplus y_l$$

for random values (x_l, y_l) and proceed similar as in the case where $C^i = C^1$.

Note that in the last three cases the complexity of a collision attack is $2^{m/2}$ by Theorem 8.1.3 but the storage requirements are 2^m, i.e., nothing is gained compared to a brute force collision attack. □

We have shown that for all double block length hash functions of hash rate 1 based on a secret key block cipher, there exist second preimage attacks with complexity of about 4×2^m. For three classes of hash functions the attack needs a pre-computed table of size 2^m. A natural requirement for a double block length hash function is that the complexities of both second preimage and collision attacks are higher than the complexities for similar attacks on single block length hash functions. Our results show that there are no double block length hash functions of hash rate 1 meeting this requirement; however

in the cases where a pre-computed table is needed the space requirements are so large, so our results can also be seen as a classification of strong and weak double block length hash functions of hash rate 1.

In the following sections we apply our results to the schemes Parallel-DM, PBGV-scheme and the LOKI-DBH hash function.

8.2 Analysis of Specific Hash Functions

8.2.1 Parallel-DM

In [37], the **Parallel-DM**, a new hash function based on a secret-key block cipher was proposed. The Parallel-DM scheme is a $2m$-bit hash function based on an m-bit block cipher with an m-bit key. The security of Parallel-DM relies on the security of the DM-scheme in the way that by a linear transformation the scheme is equivalent to two parallel runs of the DM-scheme. Thereby the scheme obtains optimum security against free-start attacks. The scheme is defined (see also Figure 8.1)

$$\begin{cases} H_i^1 &= \mathrm{E}_{M_i^1 \oplus M_i^2}(H_{i-1}^1 \oplus M_i^1) \oplus H_{i-1}^1 \oplus M_i^1 \\ H_i^2 &= \mathrm{E}_{M_i^1}(H_{i-1}^2 \oplus M_i^2) \oplus H_{i-1}^2 \oplus M_i^2 \end{cases} \tag{8.8}$$

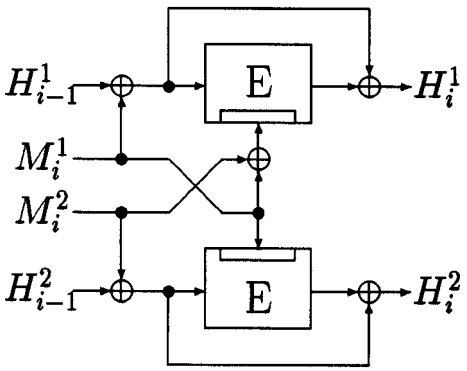


Figure 8.1: The $2m$-bit round function of the proposed Parallel-DM scheme.

To avoid trivial attacks MD-strengthening is used. First we apply the solving one half attack to the Parallel-DM.

Theorem 8.2.1 *There exists a second preimage attack on the Parallel-DM scheme which succeeds with probability* 0.63 *in time* 3×2^m. *There exists a collision attack on the Parallel-DM scheme which succeeds with probability* 0.63 *in time* $3 \times 2^{m/2}$.

Proof: Let A and B be two fixed (given or chosen) values such that $H_n^1 = E_B(A) \oplus A$. For any given value of (H_{n-1}^1, H_{n-1}^2), one can obtain on: pair of (M_n^1, M_n^2) where

$$M_n^1 = A \oplus H_{n-1}^1 \text{ and } M_n^2 = B \oplus M_n^1$$

such that the 4-tuple $(H_{n-1}^1, H_{n-1}^2, M_n^1, M_n^2)$ will yield the fixed value for H_n^1 in (8.8). Theorem 8.1.2 then implies that the complexity of a second preimage attack is about $3 \cdot 2^m$ (with $T = 0$) and the complexity of a collision attack is about $3 \cdot 2^{m/2}$. □

Note that attacks on the corresponding single block length hash function, the DM-scheme (3.1), has roughly the same complexities. There is also a second collision attack on the Parallel-DM. It has the same complexity and probability of success as the previous, but is different.

Theorem 8.2.2 *There exists a collision attack on the Parallel-DM scheme which succeeds with probability 0.63 in time* $2^{m/2}$.

Proof: In this attack we proceed as follows

1. For a given pair (H_{n-1}^1, H_{n-1}^2) set $M_n^1 = H_{n-1}^1 \oplus H_{n-1}^2$ and set $M_n^2 = 0$.

Now $H_n^1 = E_{M_n^1}(H_{n-1}^2) \oplus H_{n-1}^2 = H_n^2$. That is, the two m-bit values in the hash codes are equal. By repeating this attack $2^{m/2}$ times with probability $1 - (1 - 2^{-m/2})^{2^{m/2}} \simeq 0.63$ for one pair of k, l we have $H_n^1(k) = H_n^1(l) = H_n^2(k) = H_n^2(l)$. □

8.2.2 The PBGV hash function

This scheme was proposed in [96] and its round function is defined as follows.

$$H_i^1 = E_{M_i^1 \oplus M_i^2}(H_{i-1}^1 \oplus H_{i-1}^2) \oplus M_i^1 \oplus H_{i-1}^1 \oplus H_{i-1}^2 \quad (8.9)$$

$$H_i^2 = E_{M_i^1 \oplus H_{i-1}^1}(M_i^2 \oplus H_{i-1}^2) \oplus M_i^2 \oplus H_{i-1}^1 \oplus H_{i-1}^2 \quad (8.10)$$

Theorem 8.2.3 *There exists a second preimage attack on the PBGV scheme which succeeds with probability* 0.63 *in time* 4×2^m. *There exists a collision attack on the PBGV-scheme which succeeds with probability* 0.63 *in time* $4 \times 2^{m/2}$.

Proof: Let H_n^1 be a fixed (given or chosen) hash value and let K be a fixed (given or chosen) value of the key input in (8.9). For any given value of (H_{n-1}^1, H_{n-1}^2), let $d = H_{n-1}^1 \oplus H_{n-1}^2$, then one can obtain one pair of (M_n^1, M_n^2) where

$$M_n^1 = E_K(d) \oplus d \oplus H_i^1 \text{ and } M_n^2 = K \oplus M_n^1$$

such that the 4-tuple $(H_{n-1}^1, H_{n-1}^2, M_n^1, M_n^2)$ will yield the fixed value for H_n^1 in (8.9). Theorem 8.1.2 then implies that the complexity of a second preimage attack is about 4×2^m and the complexity of a collision attack is about $4 \times 2^{m/2}$. □

Note that the similar attacks have been reported before in [55, 93], but the above attack has a simpler form.

8.2.3 The LOKI DBH mode

The LOKI DBH Mode was proposed in [15]. It is a $2m$-bit iterated hash function, where the $2m$-bit round function is given by

$$\begin{cases} H_i^2 &= \mathbf{E}_{H_{i-1}^1 \oplus M_i^1}(H_{i-1}^1 \oplus M_i^2) &\oplus& H_{i-1}^1 \oplus H_{i-1}^2 \oplus M_i^2 \\ H_i^1 &= \mathbf{E}_{H_{i-1}^2 \oplus M_i^2}(H_{i-1}^1 \oplus M_i^1 \oplus H_i^2) &\oplus& H_{i-1}^1 \oplus H_{i-1}^2 \oplus M_i^1 \end{cases} \quad (8.11)$$

Note that (8.11) differs from the description used in [37], which is not correct.

Theorem 8.2.4 *There exists a second preimage attack on the LOKI DBH Mode scheme. The attack uses a precomputed table with* 2^m $2m$ *bit values and succeeds with probability* 0.63 *in time about* 3×2^m.

Proof: First note that H_i^2 has the form

$$H_i^2 = E_A(B) \oplus B \oplus H_{i-1}^2 \quad (8.12)$$

The matrices L and N_2 in the proof of Theorem 8.1.5 have rank three and two respectively. It follows that by pre-computing a table with 2^m $2m$ bit

values Theorem 8.1.3 holds with $T \simeq 0$, where we have assumed that the time needed to sort the table and all table look-ups equals $O(2^m)$. □

The above attack is the best attack reported in the literature to our knowledge. Note that this attack is also applicable to the first Quisquater-Girault hashing scheme from which the LOKI DBH was built [55, 37].

8.2.4 The AR hash function

In implementations of the DES the key schedule takes about twice the time of a single encryption in both hardware and software [4]. Therefore in hash functions based on a block cipher, where the chaining variable and/or the message variable is used as the key input to the DES, the key schedule has to be performed many times. In a hash function built from the DES used with fixed keys, the key schedules have to performed only once and faster schemes can be obtained. One such hash function, the AR hash function has been proposed by Algorithmic Research Ltd. It has been distributed in the ISO community [60] for informational purposes, but was not included in a standard. At the time of [60] it was in use in the German banking world.

In the following, $E_k(y)$ will denote the DES-encryption of block y using key k. The basic structure in AR hash can be described as a variant of the DES in CBC-mode, where the last 2 ciphertext blocks are added to the current input, and where the state consists of the last two "ciphertext" blocks computed. To do the entire function, the message is processed with two keys, yielding a result of 2 times 128 bits. This is then further compressed to get a result of 128 bits.

To define AR more precisely, we first divide the message m to be hashed into 8-byte blocks, denoted by $m_1, m_2, \ldots, m_n$ (0-padding is used on the last block if it is incomplete). We then define a series of 64-bit blocks o_{-1}, o_0, o_1, ..by

$$o_{-1} = o_0 = 0$$

and

$$o_i = m_i \oplus E_k(m_i \oplus o_{i-1} \oplus o_{i-2} \oplus \eta),$$

where k is an arbitrary DES key, and the constant η is defined by

$$\eta = 01\ 23\ 45\ 67\ 89\ AB\ CD\ EF$$

in hexadecimal notation. We now let $f_0(m,k), f_1(m,k), f_2(m,k)$ denote o_{n-2}, o_{n-1}, o_n respectively, In the actual hash function AR/DFP, two different keys k_1 and k_2 are used, specified as

$$k_1 = 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00, \quad k_2 = 2A\ 41\ 52\ 2F\ 44\ 46\ 50\ 2A$$

One then first computes

$$c_1 = f_1(m,k_1),\ c_2 = f_2(m,k_1),\ c_3 = f_1(m,k_2), c_4 = f_2(m,k_2)$$

and the hash value is now the concatenation of the two 8 byte blocks

$$G(G(c_1,c_2,k_1),\ G(c_3,c_4,k_1),k_1) \text{ and } G(G(c_1,c_2,k_2),\ G(c_3,c_4,k_2),k_2),$$

where G is the function defined by

$$G(x,y,k) = E_k(x \oplus y) \oplus E_k(x) \oplus E_k(y) \oplus y.$$

For convenience in the following, we will let $DFP(c_1,c_2,c_3,c_4,k)$ denote the final hash result.

Properties of f_1, f_2 and G

Let A and B be messages of length a multiple of 8 bytes, and let $A \parallel B$ be the concatenation of A and B. Choose a fixed, but arbitrary DES key k, and let $y = f_1(A,k)$, $z = f_2(A,k)$. Let m be an arbitrary 8-byte block. Let $C(A,m)$ be the three-block message

$$m \oplus \eta \oplus y \oplus z \parallel E_k(m) \oplus y \parallel E_k(m) \oplus z$$

Let $D(A,m)$ be the three-block message

$$m \oplus \eta \oplus y \oplus z \parallel m \oplus y \parallel m \oplus z$$

Let $E(A,m)$ be the three-block message

$$m \oplus \eta \oplus y \oplus z \parallel m \oplus y \parallel E_k^2(m) \oplus z$$

Then we have the following result, showing that it is very easy to find collisions for the functions f_1, f_2:

Lemma 8.2.1 *For arbitrary A, B, k, m as above, we have that*

$$f_i(A \parallel B, k) = f_i(A \parallel C(A, m) \parallel B, k),\ i = I, 2$$
$$f_2(A, k) = f_2(A \parallel E(A, m), k)$$

If k is a weak DES key, then we alao have

$$f_i(A \parallel B, k) = f_i(A \parallel D(A, m) \parallel B, k),\ i = 1, 2$$

Proof: By combining the definition of $C(A, m)$ and f_0, f_1, f_2 we obtain

$$\begin{aligned}
&f_0(A \parallel C(A, m), k) \\
&\quad = m \oplus \eta \oplus y \oplus z \oplus E_k(m \oplus \eta \oplus y \oplus z \oplus y \oplus z \oplus \eta) \\
&\quad = m \oplus \eta \oplus y \oplus z \oplus E_k(m) \\
&f_1(A \parallel C(A, m), k) \\
&\quad = E_k(m) \oplus y \oplus E_k(E_k(m) \oplus y \oplus m \oplus \eta \oplus y \oplus z \oplus E_k(m) \oplus z \oplus \eta) \\
&\quad = y \\
&f_2(A \parallel C(A, m), k) \\
&\quad = E_k(m) \oplus z \oplus E_k(E_k(m) \oplus z \oplus y \oplus m \oplus \eta \oplus y \oplus z \oplus E_k(m) \oplus \eta) \\
&\quad = z
\end{aligned}$$

This proves the first statement. The second and third are proved similarly, using for the third that if k is a weak key, then by definition it holds that $E_k(E_k(m)) = m$ for all m. □

By inspection of the definition of G, it is trivial to show the following lemma:

Lemma 8.2.2 *The functions G, DFP have the following properties for arbitrary c_1, c_2, k:*

$$G(c_1, c_2, k) = G(0, c_1, k) = E_k(0) \oplus c_1$$
$$G(c_1, c_2, k) = G(c_1 \oplus c_2, c_2, k)$$
$$G(c_1, 0, k) = E_k(0)$$
$$DFP(c_1, c_1, c_1, c_1, k) = (c_1, c_1),\ DFP(c_1, 0, c_2, 0) = 0$$

Thus, it is also very easy to find collisions for G and DFP. Although none of these properties imply directly a collision for the hash function itself, they will be useful in the following.

Attacks on AR Hash

Collision attack

This first attack on AR hash exploits the fact that for a weak key k of the DES there are 2^{32} fixpoints and they are easy to find, i.e., to find m s.t. $E_k(m) = m$, see Theorem 5.4.1.

If A is the empty message in Lemma 8.2.1, then $y = z = 0$. Let $X(m)$ be the 3-block message $m \oplus \eta \parallel E_{k_2} \parallel E_{k_2}(m)$. This means that by Lemma 8.2.1

$$\begin{aligned} f_1(X(m), k_2) &= 0 \\ f_2(X(m), k_2) &= 0 \end{aligned}$$

for any m. Let m be a fixpoint for k_1, then

$$\begin{aligned} f_1(X(m), k_1) &= E_{k_2}(m) \oplus E_{k_1}(E_{k_2}(m)) \\ f_2(X(m), k_1) &= E_{k_2}(m) \oplus E_{k_1}(E_{k_1}(E_{k_2}(m))) = 0 \end{aligned}$$

since k_1 is a weak key. The above four values are also the c_i values produced by hashing $X(m)$. But by Lemma 8.2.2, a G-value is invariant in the first argument if the second is 0, so it is clear that for fixpoints (for k_1) $m \neq m'$, $X(m)$ and $X(m')$ will be hashed to the same value. Finding two fixpoints for k_1 takes in time one DES encryption, which leads to:

Theorem 8.2.5 *There exists an algorithm, which finds in time one DES encryption, two different messages with the same AR hash value.*

The above attack can be extended to attacks that in time $n/2$ encryptions find n messages that hash to the same value, where $n \leq 2^{32}$. By contrast, a brute force attack that finds two messages that hashto the same value would require computation of about 2^{64} hash values.

Preimage attack

AR hash uses two fixed keys. In the following we consider arbitrary keys, where one key, k_1 is a weak key. Recall that the DES has 4 weak keys. The basic idea in this second attack on AR hash is to try to find a message which

takes the initial state back to itself, i.e., leads to a set of all-zero c-values. If Z is such a message, then clearly $AR(M) = AR(Z \parallel M) = AR(Z \parallel Z \parallel M) = \cdots$. It is also clear that once we have found such a Z, any message M can be attacked at no further cost.

In more detail, we try, inspired by Lemma 8.2.1, with Z of the form $Z = m_1 \oplus \eta \parallel m_2 \parallel m_2$. It is now easy to write down the equations that m_1, m_2 must satisfy in order for $f_1(Z, k_i) = f_2(Z, k_i) = 0, i = 1, 2$. We get the following:

$$E_{k_1}(m_1) \oplus m_1 = E_{k_1}^{-1}(m) \oplus m_2 \quad (8.13)$$

$$E_{k_2}(m_1) \oplus m_1 = E_{k_2}^{-1}(m) \oplus m_2 \quad (8.14)$$

It is difficult in general to say anything about the number of solutions to these equations, or how hard it is to find them. There is a special case, however, that is easier:

Put $m_2 = E_{k_2}(m_1)$, then (8.14) is always satisfied. Let m_1 be a fixpoint for k_1, then (8.13) is true if

$$E_{k_1}(E_{k_2}(m_1)) = E_{k_2}(m_1) \quad (8.15)$$

which is true if also $E_{k_2}(m_1)$ is a fixpoint for k_1. It is reasonable to assume that the mapping $E_{k_2}(\cdot)$ distributes fixpoints for k_1 uniformly. Therefore the probability that $E_{k_2}(m_1)$ is a fixpoint for k_1 is 2^{-32}. By running through all fixpoints for k_1 the probability that (8.15) is satisfied is

$$1 - (1 - 2^{-32})^{2^{32}} \simeq 1 - e^{-1} \simeq 0.63$$

Since checking whether a message is a fixpoint for a weak key takes half a DES encryption, the attack needs a total of $2 \times 2^{32} = 2^{33}$ DES encryptions. A similar attack appeared in [93]. To confirm the validity of the 0.63 probability, we did a computer simulation on a "scaled-down" version of DES, working with 32-bit blocks, thus making it easy to run through all fixpoints. The experiments confirmed the theory. The test ran through all 2^{16} fixpoints for 100 pairs of keys, where one key was a weak key in a 32 bit block version of DES. Out of 100 key pairs, the equation (8.15) had a solution for 62 pairs.

The above attack is quite feasible, and can be executed in at most a few days, even hours, using up to date hardware. Later in this section we give the results of an implementation of the attack on AR hash with the two keys

given in [60]. The above probability can be improved to almost 1 at the cost of a squared complexity. In this case we proceed as follows (where m_1 is not necessarily a fixpoint for k_1):

If we put $m_2 = E_{k_1}(m_1)$ then equation (8.13) is trivially satisfied, and (8.14) is satisfied as well, if

$$E_{k_1}(m_1) \quad = \quad E_{k_2}(m_1) \tag{8.16}$$

or

$$E_{k_2}(m_1) \oplus m_1 \quad = \quad E_{k_1}(m_1) \oplus E_{k_2}^{-1}(E_{k_1}(m_1)) \tag{8.17}$$

Symmetrically, we can put $m_2 = E_{k_2}(m_1)$. This means that (8.14) is now always satisfied, and that (8.13) is true if either $E_{k_1}(m_1) = E_{k_2}(m_1)$ (same condition as (8.16)) or if

$$E_{k_1}(m_1) \oplus m_1 \quad = \quad E_{k_2}(m_1) \oplus E_{k_1}^{-1}(E_{k_2}(m_1)) \tag{8.18}$$

Finally, since k_1 is a weak key, there is another possibility, namely to put $m_1 = m_2$. Once again, this trivially satisfies (8.13), and (8.14) is in this case satisfied, if

$$E_{k_2}^2(m_1) \quad = \quad m_1 \tag{8.19}$$

To summarise, if we can find a 64-bit block m_1 that satisfies (8.16), (8.17), (8.18) or (8.19) then we have a 3-block sequence Z that makes the attack successful. Checking if a block satisfies any of the equations requires at most 5 encryptions, so going through all possibilities for m_1 will require about $5 \cdot 2^{64} \simeq 2^{66}$ encryptions. The remaining question is of course if there are any solutions to the equations at all. Simply doing the 2^{66} encryptions is not feasible today (although it probably will become feasible in the not too distant future). Therefore the best we can do is to see if we can estimate the probability that solutions exist, assuming that the two keys k_1, k_2 are randomly chosen, but where k_1 is a weak key. Each of the 4 equations can be written in the form $h(m_1) = 0$, where h is some function that depends on the keys, and is built from a number of DES encryptions and decryptions. It is a generally accepted assumption that the DES in a context like this one behaves like a random function. This means that the 3 equations (8.16), (8.17) and (8.19) each have solutions with an independent probability of

$$1 - (1 - 2^{-64})^{2^{64}} \simeq 1 - e^{-1} \simeq 0.63$$

However, since (8.18) contains (8.15) as a special case this probability splits into two depending on whether fixpoints are examined or not, the probability that (8.18) has a solution therefore is

$$1 - ((1 - 2^{64})^{2^{64}-2^{32}} \times (1 - 2^{-32})^{2^{32}}) \simeq 1 - e^{-2}$$

Thus the probability over the choice of k_1, k_2 with k_1 weak that solutions do exist is about $1 - e^{-5} \simeq 0.99$.
In summary we have the following:

Theorem 8.2.6 *There exists two second preimage attacks on AR hash, which take time at most about 2^{33} and about 2^{66} DES encryptions, respectively. Under reasonable heuristic assumptions, the attacks can be shown to be successful for respectively about 63% and 99% of the possible choices of keys in AR hash. Both attacks can be done in a preprocessing phase, afier which each message can be attacked at no further cost.*

These attacks are much faster than a brute-force attack, which would require computation of about 2^{128} hash values.

For the keys chosen in *AR* hash we did an exhaustive search through all fixpoints for the weak key, $k_1 = 0$. We obtained

Theorem 8.2.7 *For AR hash there exists two 3-block messages Z_1 and Z_2, s.t. any message M can be prefixed with either Z_1 or Z_2 (or both) any number of times, yielding unchanged AR hash value, where*

$$Z_1 = \text{7a6199a238bb8643} \mid \text{8073d91a57ca1e2a} \mid \text{8073d91a57ca1e2a}$$
$$Z_2 = \text{02bb2604aafcbecf} \mid \text{6421e999f02ddfd6} \mid \text{6421e999f02ddfd6}$$

Conclusion

The weaknesses we have found in *AR* hash clearly make it very problematic to continue using the hash function as it is. The collision and preimage attacks can be thwarted by adding the message length to the message, however because of Theorem 8.2.7 collisions still can be obtained in constant time, because $Z_1 \parallel M$ and $Z_2 \parallel M$ would hash to the same value. So the question arises whether one can repair the function so that our attacks are prevented.

We have of course exploited the fact, that there are 2^{32} fixpoints for a weak DES key and that they are easy to find. However, avoiding weak keys still would enable a preimage attack, since equations (8.16), (8.17) and (8.18) can be set up independently of the nature of the keys. The probability for success for this attack is expected to be $1 - e^{-3} \simeq 95\%$. To confirm this we did another computer simulation on a "scaled-down" vision of DES. The test used 16 bit blocks and ran through all 2^{16} possible messages for 100 pairs of random keys. Out of 100 key pairs, for only 3 key pairs none of the equations (8.16), (8.17) and (8.18) had solutions thus confirming the theory.

Furthermore we made essential use of the fact that the initial state is all-zero, in particular that it consists of 4 blocks that are equal. Trying to prevent attacks only by changing the initial values is extremely dangerous and it is shown in [93] how to find collisions even in this case.

The Lemmas 8.2.1 and 8.2.2 show a number of problematic properties of f_1, f_2 and G that are independent of the initial state and of the chosen keys. Therefore, we believe that the basic design of f_1, f_2 and G should be reconsidered. One can perhaps guess that AR hash (or rather the f_1, f_2 functions) was designed starting from the standard MAC-mode for the DES (which uses a secret key), obtaining a hash function by using a known, fixed key, and adding some extra elements (the feedforward, etc.) to compensate for the weaknesses implied by the fact that the key is now known.

Our attacks can be seen as an illustration that constructing a hash function in this way from a MAC is not easy, and that it is perhaps a better strategy to build a hash function mode "from scratch".

8.3 Attacks based on Differential Cryptanalysis

In this section we consider hash functions based on an r-round block cipher with block size m bits and key size k bits. All secure hash functions of this kind use a mode with a feedforward of the plaintext [95]. In [93, 94] it is shown how to improve a differential attack when attacking a hash function based on a block cipher. The main idea is to look for a characteristic whose input difference equals the output difference, where for convenience we will assume that the exclusive-or is the appropriate difference. In that way a

feedforward of the plaintext will yield a collision for the hash round function. In other words, let $E_K(\cdot)$ be a block cipher used with key K and let P and P^* be two plaintext blocks. If $P \oplus P^* = E_K(P) \oplus E_K(P^*)$ then $P \oplus E_K(P) = P^* \oplus E_K(P^*)$. In the following we will assume that the attacker is given a key input to the block cipher, which can be an initial hash value or an intermediate hash value, and that he has full control over the plaintext input to the block cipher. Differential attacks on hash functions are similar to those on the block cipher itself with some important differences.

- The key is known. As stated in Section 6.1.1, characteristics for the DES vary depending on values of critical key bits. Since the attacker knows the key, he can optimise his choice of characteristic with respect to the key. Furthermore, it can be checked after every round of encryption, whether the pair of plaintexts analysed follow the extected values in the characteristic. Therefore for most pairs of plaintext only a few rounds of encryption have to be performed.

- A single right pair is sufficient for a collision. In many differential attacks more than one right pair is needed to uniquely determine the key.

- The characteristic used must be a full r-round characteristic. In differential attacks on iterated block ciphers, often a shorter characteristic is sufficient for a successful attack, see Section 5.2.

Unless the characteristic used in a differential attack on a hash function has a high probability, the method is not suitable for a preimage attack, in which case the attacker is given also the plaintext input, P, to the block cipher and for every characteristic he wants to use in the attack, there is only one choice for the collision message P^*. If the given message is long and/or many good characteristics have been found, the attacker is given the several plaintext inputs to the block cipher and has an increased probability of success. On the other hand if there exist many characteristics with high probabilities, the block cipher should probably not be used at all. We consider therefore mainly collision attacks, but we note that if one can find a characteristic of probability less than $2^{-(m-1)}$, where m is the block length and length of the hash code, a second preimage attack using only two encryptions would succeed with a higher probability than a brute force attack. We define

Definition 8.3.1 *The* **work factor** *of a differential attack on hash functions based on block ciphers is the estimated number of encryptions the attacker needs to do to find a right pair.*

In a differential attack if the work factor is larger than $2^{m/2}$, a better approach would be a brute-force collision attack based on the birthday paradox with complexity $2^{m/2}$.

8.3.1 Single block length hash functions based on DES-variants

Consider the hash function with the following hash round function

$$H_i = h(H_{i-1}, M_i) \quad = \quad E_{H_{i-1}}(M_i) \oplus M_i \tag{8.20}$$

This hash function has the same security level as the DM-scheme [55]. The only difference between the two is the interchanged role of the intermediate hash value and the message variable and since the DM-scheme is considered to be secure against a free-start attack then so is (8.20). Note also that the MDC-2 [10, 38] uses two parallel runs of (8.20). Therefore the attacks we are to describe can also be applied to double block length hash functions which uses two parallel runs of (8.20) and the complexities of the attacks are squared. The DES is the most widely used block cipher for encryption and naturally also the most used block cipher in hash functions based on block ciphers. We can use the 2-round iterative characteristic to do a differential collision attack on (8.20) based on DES variants with an odd number of rounds. Let $\Phi = 19600000_x$ and $\Gamma = 1b600000_x$ be the non-trivial xor half of the two characteristics used by Biham and Shamir to attack the DES [7]. As discussed in Section 6.1.1 the probabilities of these characteristics depend on certain values of two critical key bits. The probability for two rounds is $\frac{1}{146}$ for Φ if the key bits are equal and $\frac{1}{585}$ for Γ. If the key bits are different the probabilities are interchanged.

We construct an n-round characteristic, n odd, in an attack on a hash function based on an n-round version of the DES using either Φ or Γ as the different inputs in the left halves and with equal inputs in the right halves, i.e., using a input difference $(\Phi \mid 0)$ or $(\Gamma \mid 0)$. The probability $\frac{1}{146}$ can be

obtained for every two rounds for about one out of $2^{n/2-1}$ keys. There are $n/2$ rounds in the characteristic with different inputs to the F-function and the characteristic Φ is used if all $n/2$ pairs of critical key bits are equal and Γ is used if all $n/2$ pairs of critical key bits are different. It is assumed that the initial hash value (the key input to DES) given to the attacker enables him to construct characteristics with a probability of $\frac{1}{146}$ per two rounds. If the initial hash value does not meet that requirement he chooses a random message and hashes the message obtaining a new intermediate hash value and proceeds from there.

Let us consider a hash function (8.20) based on a 7-round version of the DES. The characteristic, say $(\Phi \mid 0)$, we use has a probability of $(\frac{1}{146})^3 \simeq 2^{-21.6}$ However we can start our characteristic in the second round; simply choose two ciphertexts after one round with difference $(0 \mid \Phi)$, then the difference in the plaintexts is exactly as we desire. The actual values of the plaintexts can be computed once a right pair is found. Furthermore we choose the inputs to the characteristic in the second round such that the probability after two complete rounds is one. The success of the combination $0 \leftarrow \Phi$ in the second round depends on only pairs of 14 bits and they can be calculated in a pre-processing phase. In the third round we obtain equal inputs to the F-function and in the fourth round the difference in the inputs to the F-function will again be Φ. It is seen that the inputs can be chosen such that the characteristic has probability one after 3 rounds of encryption, i.e., the probability of our characteristic is $(\frac{1}{146})^2 \simeq 2^{-14.4}$. We will assume that the number of pairs we have to try to get one right pair is the reciprocal value of the characteristic used, in this case about $2^{14.4}$ pairs. For most of these pairs we have to compute only 3 rounds of encryptions, namely in the second, third and fourth rounds after which only $2^{7.2}$ pairs are left. The work factor of the differential attack is therefore about $2^{14.4} \times \frac{3}{7} \times 2 \simeq 2^{14.2}$, where we note that the number of encryptions needed is twice the number of pairs needed in the attack.

In Table 8.1 we give the work factors and the probabilities of the n-round characteristics in differential collision attacks on hash functions based on DES versions with n, odd, number of rounds. It is seen that the differential collision attack is better than a birthday collision attack on hash functions based on the DES with up to 11 rounds. With 13 rounds the two attacks have similar complexities. When the probability of the n-round characteristic is below 2^{-64} we cannot expect to get a right pair using a fixed key, since there

# Rounds (n)	Work factor ($\log_2$)	Probability ($\log_2$) of the n-round characteristic
7	14.2	-21.6
9	21.0	-28.8
11	27.9	-36.0
13	34.9	-43.2
15	41.9	-50.3
17	48.8	-57.5

Table 8.1: Complexities of a differential collision attack on DES-based hash functions for DES versions with a restricted odd number of rounds.

are only 2^{64} possible pairs with a given difference. If we do not get a right pair we could compute a new intermediate hash value to get a new key and proceed from there. Also we note that similar calculations were made in [95] but contained some errors. Finally, it should be stressed once again that the above attacks are not applicable to hash functions using the DES with an even number of rounds.

Preneel's proposal using DES

As stated in Section 6.1 the best characteristic for an attack on the DES is the 2-round iterative characteristic. However, since the halves in the DES are not swapped after the last round of encryption we cannot use the iterative characteristic, used to attack the DES itself in [7], to attack a hash function based on DES. An xor in the plaintexts $\Delta P = (X, 0)$ would lead to an xor in the ciphertexts $\Delta C = (0, X)$. In [93, 94] the following characteristic was proposed. Let X be an 32 bit xor, such that $0 \leftarrow X$ with probability p_1 and $X \leftarrow X$ with probability p_2. Then a 16-round characteristic with equal input xor and output xor can be built by using seven times the first combination and two times the second combination. In the remaining rounds equal inputs are obtained, i.e., the combination used is $0 \leftarrow 0$. The characteristic will have a total probability of $p_1^7 \times p_2^2$. We state the following result.

Lemma 8.3.1 (DES) *Two inputs with xor X for which $X \leftarrow X$ and $0 \leftarrow X$ must differ in the inputs to at least six S-boxes.*

Proof: Follows from the E-expansion and P-permutation of the DES. □

A similar result does not hold for inputs which differ in the inputs to six neighbouring S-boxes, however it follows by a closer look at the DES that only for inputs different in the inputs to S-boxes 5, 6, 7, 8, 1, and 2 we can obtain the desired characteristic. By consulting the difference distribution table it is easy to find that the one round characteristics $0 \leftarrow X$ and $X \leftarrow X$ where the inputs differ only in the inputs to S-boxes 5, 6, 7, 8, 1, and 2 will have probabilities of at most $2^{-17.6}$ and $2^{-16.9}$, respectively.

It may possible to build the 16-round characteristic where in the nontrivial rounds the two inputs are different in the inputs to six not neighbouring S-boxes. In that case there have to be different inputs to two times three S-boxes leading to equal outputs. But this can happen with a maximum probability of less than $\frac{1}{234} \times \frac{1}{341}$, (see Table 6.2). For the combination $X \leftarrow X$ the maximum probability will be $(\frac{16}{64})^6$.

For inputs different in the inputs to seven S-boxes the probability for any non-trivial one-round characteristic can be at most $(\frac{16}{64})^7$. The work factor for the differential attack will be about $(p_1^6 \times p_2)^{-1} \times \frac{3}{16} \times 2$, where we note that for most pairs we have to compute only about 3 rounds of encryptions in the same way as discussed earlier in this section.

Different inputs to	Probability (log_2) of the 16 round characteristic	Work Factor (log_2)
Six neighbouring S-boxes	-157	120
Six not neighbouring S-boxes	-138	109
Seven S-boxes	-126	96
Eight S-boxes	-144	110

Table 8.2: Maximum probabilities and minimum work factors for a differential collision attack using 16-round DES.

We summarise the maximum probabilities and work factors for a collision attack on the 16-round characteristic in Table 8.2.

From Table 8.2 it is seen that the Preneel's characteristic has a probability too low to be used a differential collision attack using the DES with 16 rounds. We are convinced, that by consulting the difference distribution table more

carefully or by doing exhaustive search for the combinations $X \leftarrow X$ and $0 \leftarrow X$ much lower probabilities will be found.

8.3.2 New characteristics for differential collision attacks

As we have seen it is not possible to use the characteristics which are optimal to attack the DES itself, to attack hash functions based on the DES with an even number of rounds. In this section we look for characteristics, whose input difference equals the output difference. Consider an r-round block cipher, r even. The basic idea is to build a characteristic, for which the difference in the two halves of the ciphertexts after $(r/2)$ rounds is equal, see Ψ in Figure 8.2. Then with the same probability the output difference of the two rounds will be some Λ (here we assume that the round keys are independent). It is easy to see that by choosing the values in the characteristic from the $(r/2)$'th round and backwards to the difference in the plaintexts equal to the values from the $(r/2+1)$'th round and to the difference in the ciphertexts, will yield a characteristic for which the differences in the plaintexts and in the ciphertexts are equal.

DES

The problem in finding good characteristics for the DES is, that when too many bits are different in the outputs of one round, because of the avalanche effect of DES, these differences spread to many S-boxes in the next round, which causes the overall probabilities to be low. To get equal outputs of the rounds we can use the 2-round iterative characteristic in the construction of Figure 8.2. In every round the combination $0 \leftarrow \Phi$ is used and the 16 round characteristic has a probability of $(\frac{1}{146})^8 \simeq 2^{-115}$. Although this is better than for the characteristic we analysed in the previous section it is still too low to hope for a successful attack using 16-round DES.

As stated by Theorem 6.1.1, inputs to the F-function different in the inputs to less than three S-boxes cannot lead to equal outputs of the F-function. And inputs different in the inputs to only one S-box, S(i) yield different inputs to at least two S-boxes (and not S(i)) in the following round. However, it is possible to find inputs different in the inputs to two neighbouring S-boxes,

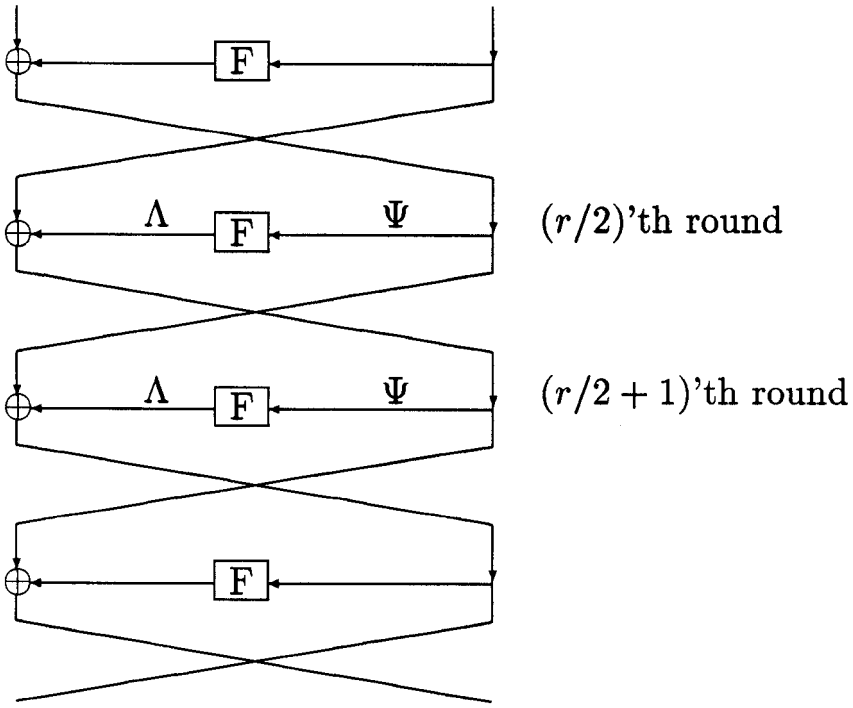


Figure 8.2: A characteristic to be used in a differential collision attack.

S(i) and S(i + 1), such that the outputs differ in only one bit, which via the P-permutation goes back to either S(i) or S(i + 1) in the following round. This phenomenon can be used to construct the 4-round iterative characteristic in Figure 8.3 for the DES, where $\Lambda \oplus \alpha = \Psi$ For every pair of neighbouring S-boxes in the DES there is exactly one possible value of α, which is due to the P-permutation and several possible values of Ψ and Λ. It is easy to find the best one for the DES by consulting the difference distribution table. With $\Lambda = e0000004_x, \Psi = e0000006_x$ and $\alpha = P(10000000_x) = 00000002_x$, one obtains a 4-round iterative characteristic with probability $(\frac{8\times10\times6\times10}{64^4})^2 \simeq 2^{-23.6}$. Like the 2-round iterative characteristics for the DES this probability splits into two per two rounds depending on the values of certain key bits. For the above characteristic these probabilities are between $2^{-23.0}$ and $2^{-24.2}$ respectively, where in the first case $\alpha \leftarrow \Lambda$ has probability $2^{-5.7}$ and $\alpha \leftarrow \Psi$ has probability $2^{-5.8}$. This characteristic can be concatenated with itself any number of times and by starting in the middle rounds it can be used to construct any s-round characteristic for s even, such that the difference

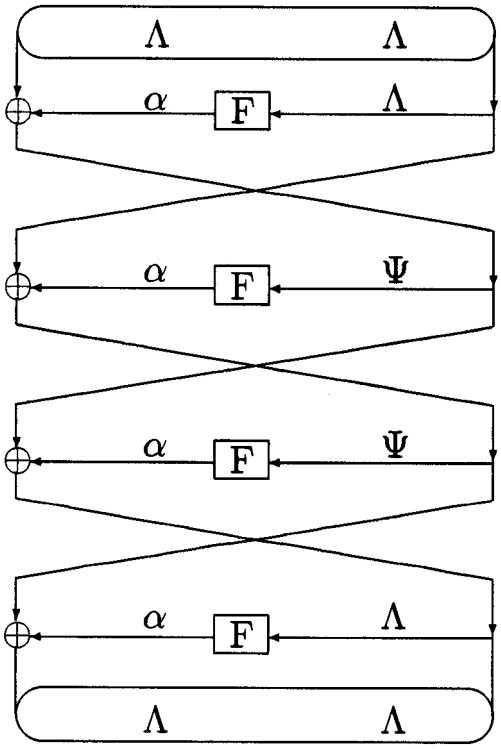


Figure 8.3: A 4-round iterative characteristic of DES-like ciphers.

in the plaintexts equals the difference in the ciphertexts. In characteristics with an even number of rounds, but not divisible by four, we maximise the probability by optimising the use of the combination $\alpha \leftarrow \Lambda$. In that way we obtain estimates calculated in the same way as for characteristic with odd number of rounds, see Table 8.3. Consider a 6-round variant of the DES. The characteristic we will use has the differences Λ in the inputs to the third and fourth rounds, therefore the difference in the plaintexts and in the ciphertexts will be $(\Lambda \mid \Lambda)$ with probability $2^{-34.4}$. Again we start the characteristic in the second round, i.e., we choose the pairs in the inputs to the F-function in the first and in the second round of the characteristic, such that the pair is a right pair after two rounds of encryption. The left halves of the plaintext pairs then automatically have the desired difference. In that way the probability of the characteristic increases to about $2^{-22.9}$. The work factor is $2^{22.9} \times \frac{2}{6} \times 2 \simeq 2^{22.3}$, since for most pairs only two rounds of encryption

# Rounds (n)	Work factor ($\log_2$)	Probability ($\log_2$) of the n-round characteristic
6	22.3	-34.4
8	33.5	-46.0
10	44.7	-57.4
12	55.9	-69.0
14	-	-80.4
16	-	-92.0

Table 8.3: Complexities of a differential collision attack on S^3-DES-based hash functions for s^3-DES versions with a restricted even number of rounds.

are needed, i.e., in the second and third rounds. From Table 8.3 it is seen that a differential collision attack is superior to the birthday collision attack in variants of the DES up to 8 rounds. It is seen that extended to 16 rounds our characteristic will have a probability of about 2^{-92}, which is much higher than for Preneel's proposal.

s^3-DES

In Section 6.4 we made an ad hoc search for characteristics to be used in differential attacks on the block cipher s^3-DES. Recall that s^3-DES is identical to the DES except for the S-boxes, which have been replaced. Our analysis showed that it is doubtful whether good characteristics exist for a successful differential attack. As for the DES there exist characteristics to be used differential collision attacks on the hash function (8.20) used with s^3-DES. With $\Lambda = 0001e000_x, \Psi = 0005e000_x$ and $\alpha = P(00004000_x) = 00040000_x$ of Figure 8.3, one obtains a 4-round iterative characteristic with probability $(\frac{10\times12\times12\times12}{64^4})^2 \simeq 2^{-19.8}$. Depending on certain values of the key, this probability splits into different values, the highest being 2^{-19}. Using this probability we obtain estimates on a differential collision attack on the hash function (8.20) using s^3-DES as the block cipher as listed in Table 8.4. It is interesting to note that although the block cipher s^3-DES seems to be more secure against a differential attack than the DES, differential attacks on hash functions using s^3-DES have lower complexities than the attacks on hash functions using DES.

# Rounds (n)	Work factor ($\log_2$)	Probability ($\log_2$) of the n-round characteristic
6	18.0	-28.2
8	27.6	-38.2
10	36.9	-47.2
12	46.1	-57.2
14	55.4	-66.3

Table 8.4: Complexities of a differential collision attack on s^3-DES-based hash functions for s^3-DES versions with a restricted even number of rounds.

The LOKI ciphers

For LOKI'89 the best known differential attack uses a 3-round iterative characteristic of probability $(28/4096)^2 \simeq 2^{-14.4}$, see [48]. The characteristic is

$$
\begin{array}{ccccl}
 & & (\Gamma, 0) & & \\
0 & \leftarrow & & 0 & \text{always} \\
\Gamma & \leftarrow & & \Gamma & \text{prob. } p = 28/4096 \\
\Gamma & \leftarrow & & \Gamma & \text{prob. } p = 28/4096 \\
 & & (\Gamma, 0) & &
\end{array}
$$

where $\Gamma = 00400000_x$.

For LOKI'91 the best known differential attack uses also the above 3-round iterative characteristic of probability 2^{-16}, see [14] and also Section 6.2.

This characteristic can be used to construct an s-round characteristic for s even, where the input difference equals the output difference for both LOKI'89 and LOKI'91. As in the case for the DES, we let the input differences in the two middle rounds be equal, here Γ. Table 8.5 gives the probabilities and work factors for a differential collision attack on the hash function (8.20) using the LOKI ciphers as the underlying block cipher. Note that the work factors are the number of encryptions needed by the underlying block cipher. Therefore the work factor in real time for LOKI with 10 rounds is higher than for LOKI with 8 rounds, despite the lower value for the former in Table 8.5. It is interesting to note that although the block cipher LOKI'91 is (probably) secure against a differential attack, differential

Cipher	# Rounds (n)	Work factor ($\log_2$)	Probability (logs) of the n-round characteristic
LOKI'89	6	14.3	-28.8
	8	28.4	-43.2
	10	28.0	-43.2
	12	42.2	-57.5
LOKI'91	6	16.0	-32.0
	8	31.6	-48.0
	10	31.3	-48.0
	12	47.0	-64.0

Table 8.5: Complexities of a differential collision attack on LOKI-based hash functions for LOKI versions with a restricted even number of rounds.

attacks on hash functions using the LOKI's have lower complexities than the attacks on hash functions using DES.

Chapter 9

Conclusions

The constructions of block ciphers have traditionally been based on ad hoc methods evolved from years of experience. No theory of how to build a secure block cipher has been developed. The design and cryptanalysis of ciphers are closely related, which is illustrated to a wide extent in this thesis.

There are three main applications for block ciphers, first of all and mainly, for encryption, but also as building blocks in cryptographic hash functions and digital signature schemes. We establish a new upper bound on the complexities of attacks on block ciphers, when used for encryption in the standard modes of operation.

During the last three years the two most important methods of cryptanalysing block ciphers have seen the light of the day, namely differential cryptanalysis and linear cryptanalysis. In this thesis lower bounds on the complexities of differential and linear attacks were given, and it was shown that there exist functions to be used in constructions of block ciphers provably resistant against the two attacks. The method of differential cryptanalysis was extended to include higher order differentials and partial differentials.

A third attack, based on simple relations was introduced. We defined and showed how to construct strong key schedules, that will thwart attacks based on simple relations and at the same time avoid so-called weak keys.

Extensive analyses were given of the specific block ciphers, DES, LOKI'91, s^2-DES, $x\text{DES}^1$, and $x\text{DES}^2$.

One way of enhancing the strength of a block cipher is by means of

multiple encryption. A new multiple encryption scheme was given, using a minimum number of key material, but provably as secure as single encryption under relevant and suitable assumptions.

Finally we cryptanalyzed hash functions based on block ciphers. At tacks on a large class of such hash functions were given. The results of our analysis are that it is difficult to construct hash functions based on block ciphers, that are both fast and with a high security level.

Appendix A

A Pictorial Illustration

The following five pages contain a pictorial illustration of conventional cryptography. The first figure shows a non encrypted picture, a plain-text, the following four figures show encryptions of the plaintext, using a substitution cipher, a transposition cipher, and using a product cipher in both the ECB and CBC modes.

Figure 1: The Plaintext

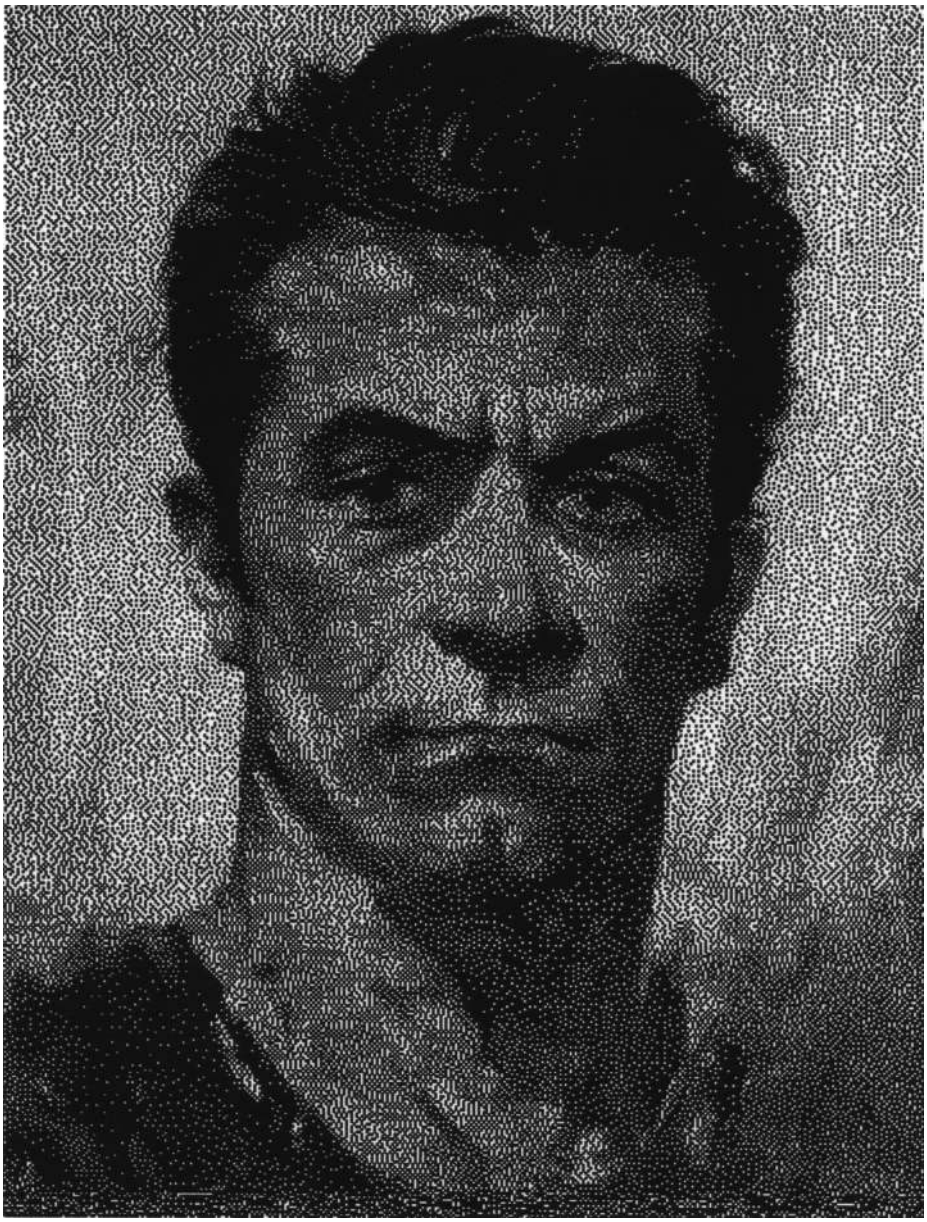


Figure 2: Substitution Cipher

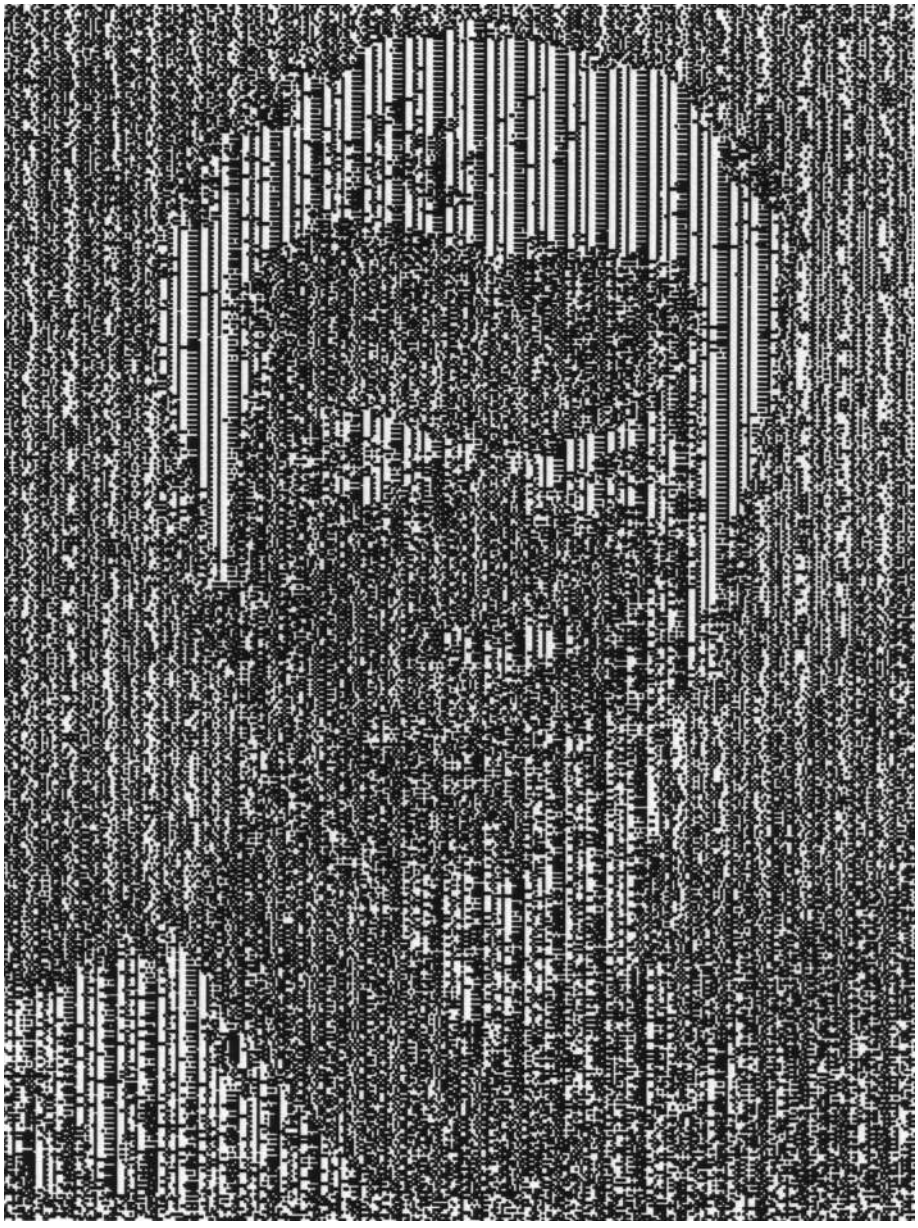


Figure 3: Transposition Cipher

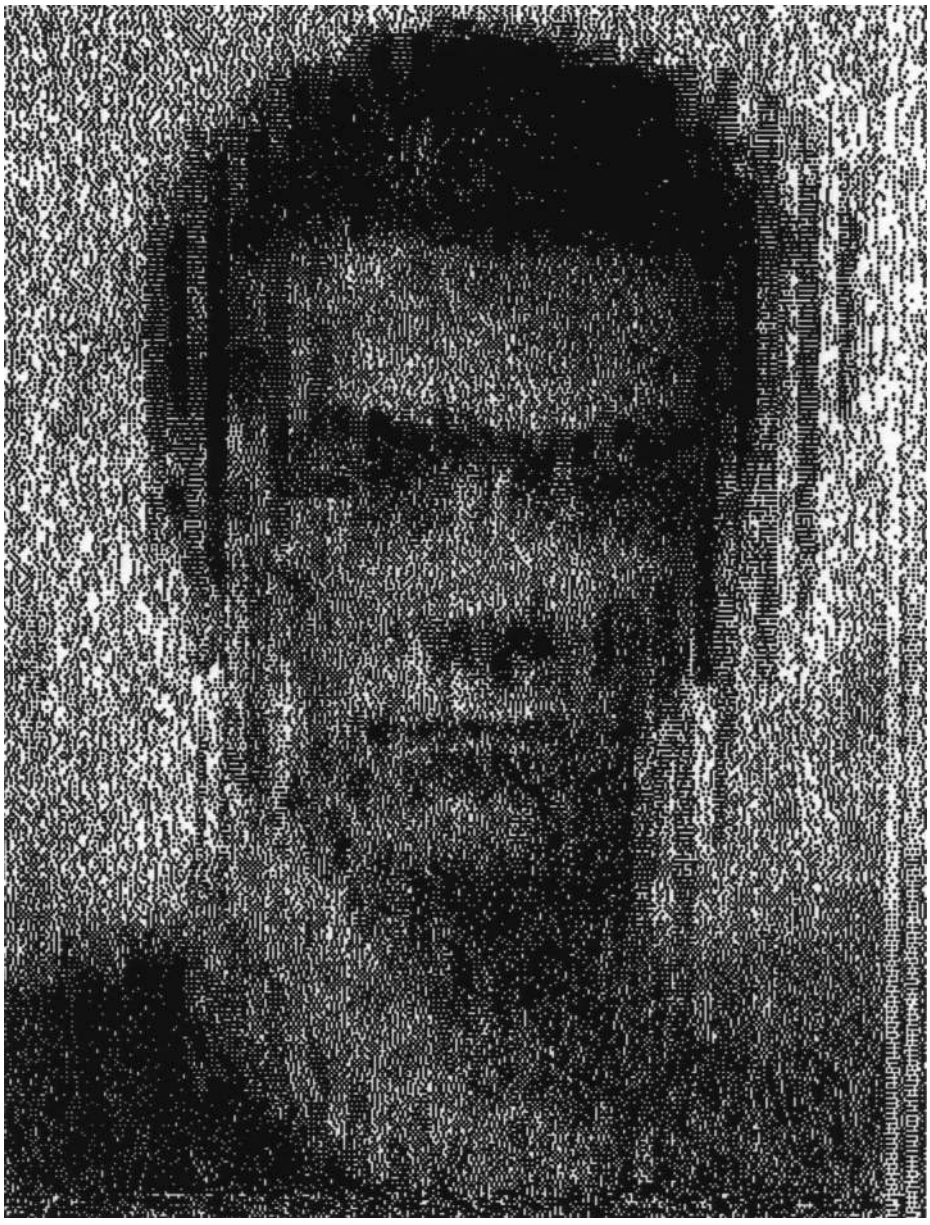


Figure 4: Product Cipher in ECB Mode

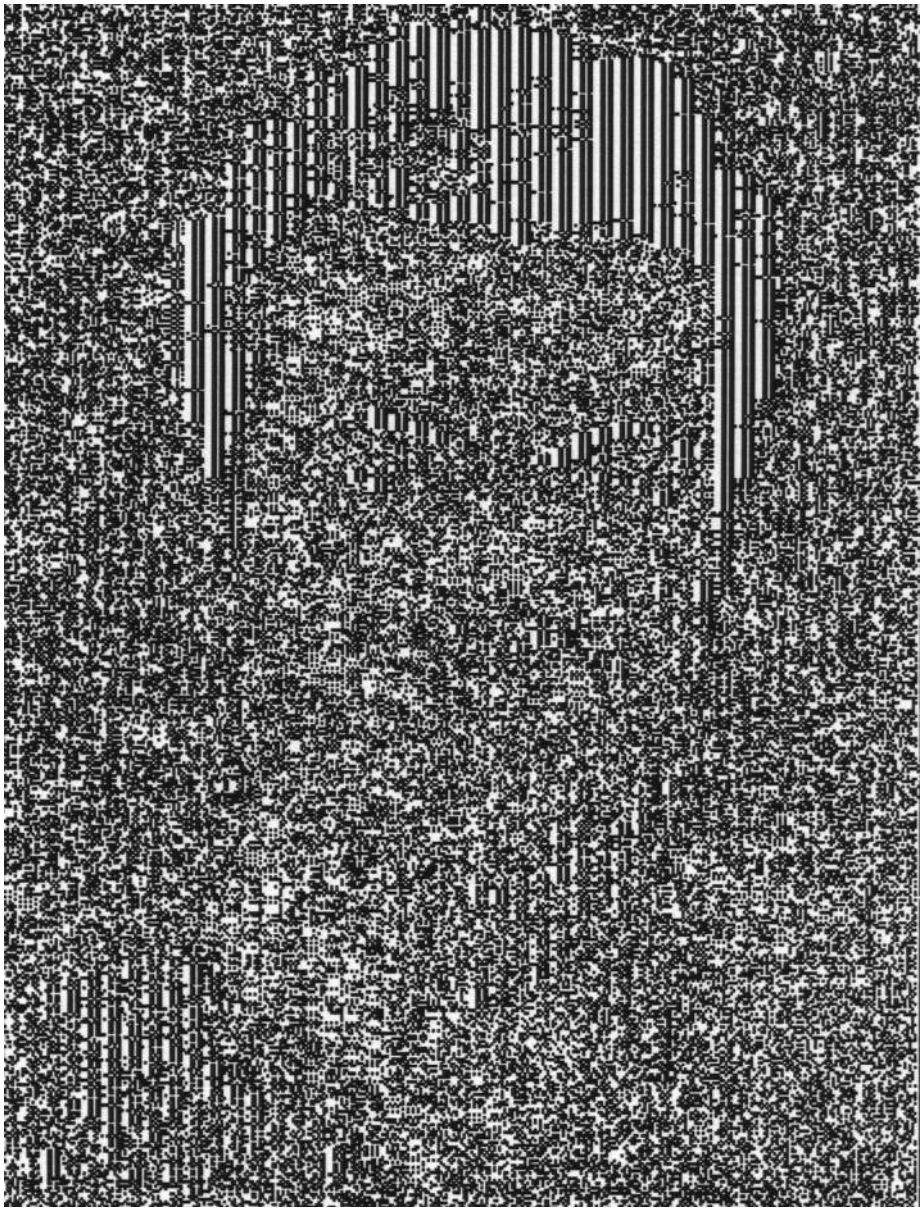
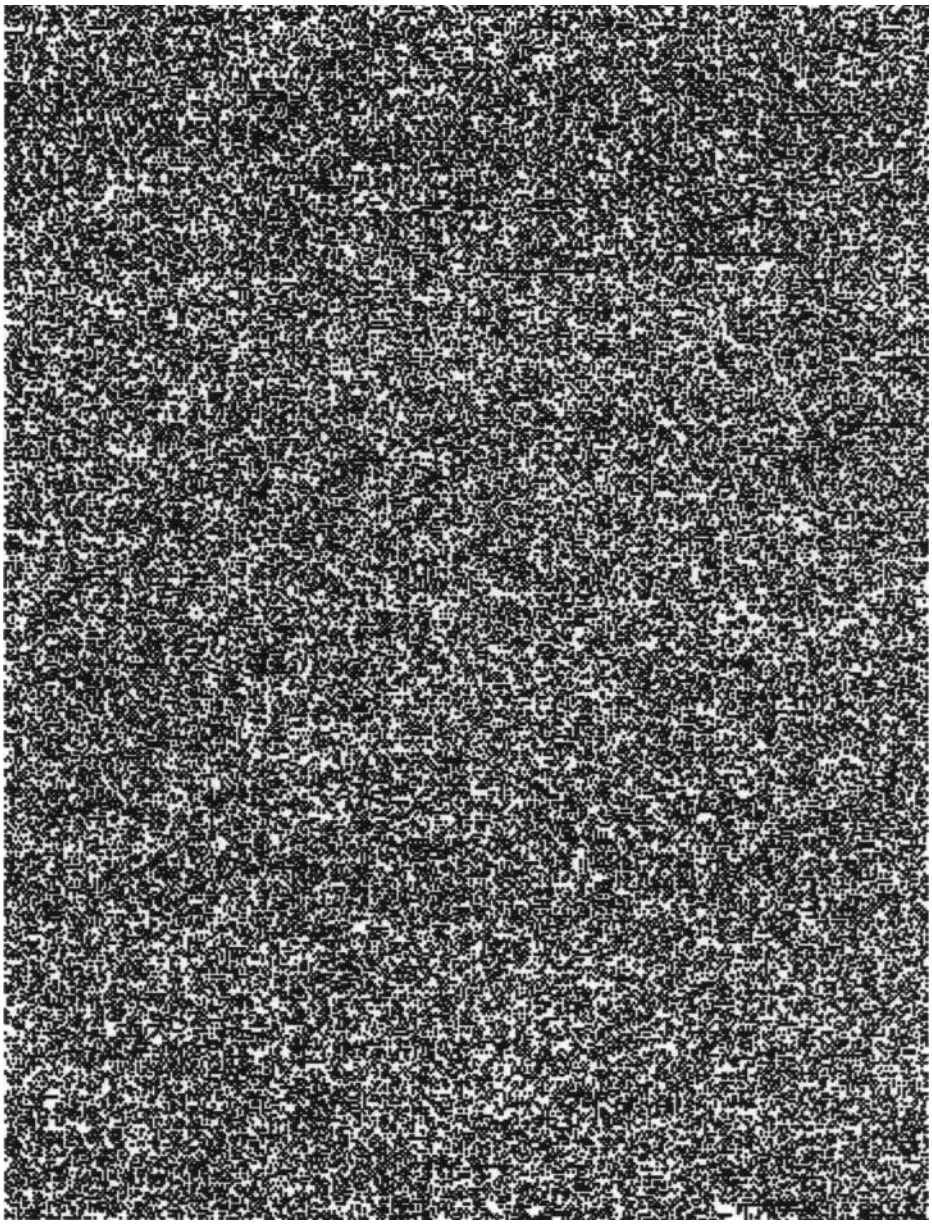


Figure 5: Product Cipher in CBC Mode



Appendix B

Tedious Proofs

B.1 Iterative Characteristics for the DES

In this section we give the proofs of the claims given in Section 6.1.1. Notation: Let Γ be an xor-sum of two inputs Y, Y^* to the F-function. Then $\Delta S(\Gamma)$ is the set of S-boxes, whose imputs are different after the E-expansion of Y and Y^*. Furthermore $\#\Delta S(\Gamma)$ denotes the number of S-boxes in $\Delta S(\Gamma)$. Example: Let $\Gamma = 0f000000_x$ (hex), then $\Delta S(\Gamma) = \{S1, S2, S3\}$ and $\#\Delta S(\Gamma) = 3$. Note that xor-addtiton is linear in both the E-expansion and the P-permutation of DES. In the proofs below the following tables and lemmata are used. Table B.1 shows thor each of the 8 S-boxes, which S-boxes are affected by the output of the particular S-box. Numbers with a subscript indicate that the particular bit affects one S-box directly and another S-box via the E-expansion. Example: If the output difference of $S1$ is 6_x (hex), then S-boxes 5 and 6 are directly affected and S-box 4 is affected after the E-expansion in the following round. Table B.2 shows the *reverse* of Table B.1, i.e., for every S-box it is shown which S-boxes from the preceding round affect the input, see also [21].

The next five lemmata follow form Tables B.1 and B.2.

Lemma B.1.1. *The six bits that make the input for an S-box, Si, come from six distinct S-boxes and not from Si itself.*

Lemma B.1.2. *The middle six input bits for two neighbouring S-boxes come*

S1	→	3	2	5	4	6		8	
S2	→	4	3	7	8	1		5	
S3	→	6	7	4	5	8		2	
S4	→	7		5	6	3		1	8
S5	→	2	3	4		7	6	1	
S6	→	1	2	8	7	3		5	
S7	→	8	1	3	4	6		2	
S8	→	2	1	7		4		6	5

Table B.1: Where the bits from an S-box go to.

S1	S2	S3	S4	S5	S6	S7	S8
4 2 5 6	8 3 7 5	1 4 6 7	2 5 8 3	1 2 6 4	8 7 1 3	5 4 8 2	6 3 1 7

Table B.2: Where the bits from an S-box come from.

from six distinct S-boxes.

Lemma B.1.3 *The middle ten input bits for three neighbouring S-boxes come from all 8 S-boxes. Six of the ten bits come from six distinct S-boxes and four bits come from the remaining two S-boxes.*

Lemma B.1.4 *The middle two bits in the input of an S-box Si, the inner input bits, come from two S-boxes, whose inner input bits cannot come from Si.*

Lemma B.1.5 *Let Φ be the difference in two inputs to the F-function and let Γ be the difference of two outputs of the F-function, such that $\Phi \rightarrow \Gamma$. If $\#\Delta S(\Phi) = \#\Delta S(\Gamma) = 2$ then for at least one S-box of $\Delta S(\Gamma)$ the inputs differ in only one bit.*

We are now ready to give the proof of the first claim in Section 6.1.1, page 99.

Theorem B.1.1 *For the xors, Γ and Φ, i.e., $\Gamma \rightarrow \Phi$ and $\Phi \rightarrow \Gamma$, the inputs to at least 5 S-boxes are different. That is, $\#\Delta S(\Gamma) + \#\Delta S(\Phi) \geq 5$.*

Proof:

1. $\#\Delta S(\Gamma) = 1$. The inputs to $\Delta S(\Gamma)$ differ in the inner input bits, i.e., at most two bits. Because of properties 2 and 4 of the DES S-boxes $\#\Delta S(\Phi) \geq 2$. The inputs of each S-box in $\Delta S(\Phi)$ differ in at most one bit each. Because of property 2 the outputs of Φ differ in at least four bits. Therefore $\Phi \not\rightarrow \Gamma$.

2. $\#\Delta S(\Gamma) = 2$. Because of the symmetry of the characteristic it follows a immediately that $\#\Delta S(\Gamma) \geq 2$. There are two cases to consider:

   (a) $\Delta S(\Gamma)$ are not neighbours. Because of properties 2 and 4 the outputs of both S-boxes in $\Delta S(\Gamma)$ will differ in at least two bits, making $\#\Delta S(\Phi) \geq 3$ according to Table B.1.

   (b) $\Delta S(\Gamma)$ are neighbours. From Lemma B.1.2 it follows that the outputs of $\Delta S(\Phi)$ differ in at most one bit each. Property 2 requires the inputs of $\Delta S(\Phi)$ to differ in at least two bits each. From Table B.1 it follows that the only way two neighbouring S-boxes in Γ can make the inputs of $\Delta S(\Phi)$ differ in at least two bits each, is when $\#\Delta S(\Phi) = 3$. This is however not possible for all pairs of neighbouring S-boxes. For example let $\Delta S(\Gamma) = \{S5, S6\}$, then it is possible to get $\Delta S(\Phi) = \{S1, S2, S3\}$ where for each S-box the inputs differ in two bits. But for $\Delta S(\Gamma) = \{S1, S2\}$ there will always be at least one S-box in $\Delta S(\Phi)$, whose inputs differ in only one bit.

3. $\#\Delta S(\Gamma) \geq 3$. Because of the symmetry of these xors $\#\Delta S(\Phi) \geq 2$. □

Next we will prove the second claim of Section 6.1.1, page 99, i.e., we will show that there is no 4-round iterative characteristic with a probability higher than the best 2-round iterative characteristic concatenated with itself. First we prove

Theorem B.1.2 *For a 4-round iterative characteristic with input sums* Γ, Φ *and* Ψ, *see Section 5.2.2,*

$$\#\Delta S(\Gamma) + \#\Delta S(\Phi) + \#\Delta S(\Psi) \geq 7.$$

Furthermore, for at least one of the input sums the inputs to three neighbouring S-boxes differ.

Proof: We are looking for input sums Γ, Φ and Ψ, such that $\Gamma \rightarrow \Phi$, $\Psi \rightarrow \Phi$ and $\Phi \rightarrow \Gamma \oplus \Psi$. Note that $\Delta S(\Gamma) \cap \Delta S(\Psi) \neq \emptyset$ and that if $\Delta S(\Gamma)$ are neighbours then so are $\Delta S(\Psi)$.

1. $\#\Delta S(\Gamma) = 1$. From the proof of Theorem B.1.1 we have $\#\Delta S(\Phi) \geq 2$, and each of the inputs to those S-boxes differs in exactly one bit.

   (a) $\#\Delta S(\Phi) = 2$. The S-boxes in $\Delta S(\Phi)$ are not neighbours and the inputs differ in one inner input bit, therefore each of the outputs differ in at least two bits. From a close look at Table B.1 it follows that if $\Delta S(\Gamma) = \{S7\}$ then it is possible to get $\#\Delta S(\Psi) = 3$, but then for one S-box $\in \Delta S(\Psi)$, not $S7$, the inputs differ in only one bit, an inner input bit. If $\Delta S(\Gamma) \neq \{S7\}$ then $\#\Delta S(\Psi) \geq 4$ and for at least one S-box, not $\Delta S(\Gamma)$, the inputs differ in onlyine bit. Therefore $\Psi \not\rightarrow \Phi$.

   (b) $\#\Delta S(\Phi) \geq 3$. The outputs for every S-box of $\Delta S(\Phi)$ differ in at least two bits. It follows easily from Table B.1 that $\#\Delta S(\Gamma \oplus \Psi) \geq 4$. Since $\Delta S(\Gamma) \subseteq \Delta S(\Psi)$, $\#\Delta S(\Psi) \geq 4$.

2. $\#\Delta S(\Gamma) = 2$. By the symmetry of the characteristic $\#\Delta S(\Psi) \geq 2$ and therefore $\#\Delta S(\Phi) \geq 3$. There are two cases to consider:

   (a) $\Delta S(\Gamma)$ are not neighbours. $\#\Delta S(\Phi) \geq 3$ because of properties 2 and 4, leaving only the possibility that $\#\Delta S(\Psi) = 2$ and $\#\Delta S(\Phi) = 3$. The S-boxes in $\Delta S(\Phi)$ must be neighbours. If not, let Si be an isolated S-box, different in the inputs in only inner bits. The outputs of Si differ in at least two bits, that must go to the inner bits of the two S-boxes in $\Delta S(\Gamma)$, since $\Delta S(\Gamma) = \Delta S(\Psi)$. But that is not possible according to Lemma B.1.4.

   (b) $\Delta S(\Gamma)$ are neighbours.

      i. $\#\Delta S(\Phi) = 1$. The outputs of $\Delta S(\Phi)$ differ in at least two bits. From Table B.1 it follows easily that for at least one S-box in $\Delta S(\Psi)$ not in $\Delta S(\Gamma)$ the inputs differ in only one bit and $\Psi \not\rightarrow \Phi$.

      ii. $\#\Delta S(\Phi) = 2$. Assume that $\#\Delta S(\Psi) = 2$. If $\Delta S(\Gamma) = \Delta S(\Psi)$ then the outputs of $\Delta S(\Phi)$ can differ in at most one bit each, according to Lemma B.1.2. But by Lemma B.1.5,

the inputs of at least one S-box in $\Delta S(\Phi)$ differ in only one bit, a contradiction by property 2. Therefore $\Delta S(\Gamma) \neq \Delta S(\Psi)$. Since $\Delta S(\Gamma) \cap \Delta S(\Psi) \neq \emptyset$ and $\Delta S(\Gamma)$ are neighbours we must have $\Delta S(\Gamma) = \{S(i-1), Si\}$ and $\Delta S(\Psi) = \{S(i), S(i+1)\}$ or vice versa. The outputs from $S(i-1)$ in Γ must be equal as must the outputs from $S(i+1)$ in Ψ. Therefore $\Gamma \oplus \Psi$ must have the following form (before the expansion):

$$S(i-1) \parallel S(i) \parallel S(i+1) = 0xyz \parallel 1**1 \parallel 0vw0,$$

where '$*$' is any bit, $xyz \neq 000$ and $vw \neq 00$. From Table B.2 it follows that $\Phi \not\rightarrow \Gamma \oplus \Psi$ for $\#\Delta S(\Phi) = 2$ and therefore $\#\Delta S(\Psi) \geq 3$.

iii. $\#\Delta S(\Phi) = 3$. Then $\#\Delta S(\Psi) = 2$. If $\Delta S(\Gamma) \neq \Delta S(\Psi)$ then the differences in the inputs to Φ is the effect of one S-box. For every S-box in $\Delta S(\Phi)$ the inputs differ in only one bit, therefore $\Phi \not\rightarrow \Gamma \oplus \Psi$. By similar reasoning we find that for both S-boxes in $\Delta S(\Gamma)$ the outputs have to differ. Furthermore $\Delta S(\Phi)$ are neighbours. Assume that they are not. Then the outputs of the isolated S-box differ in at least two bits, and from Table B.1 it follows that they affect at least 2 not neighbouring or 3 neighbouring S-boxes, a contradiction with $\Delta S(\Gamma) = \Delta S(\Psi)$.

	$\Delta S(\Gamma)$	$\Delta S(\Phi)$	$\Delta S(\Psi)$
Case A	2	2	3
Case B	2	3	2
Case C	3	1	3

3. $\#\Delta S(\Gamma) = 3$. Because of the symmetry in the characteristic we covered the cases where $\Delta S(\Psi) < 3$. Therefore $\#\Delta S(\Gamma) = \#\Delta S(\Psi) = 3$ and $\#\Delta S(\Phi) = 1$. $\Delta S(\Gamma)$ must be neighbours. Furthermore $\Delta S(\Gamma) = \Delta S(\Psi)$ otherwise $\Phi \not\rightarrow \Gamma \oplus \Psi$. □

Theorem B.1.3 *There are no 4-round iterative characteristics with a probability higher than* $(\frac{1}{234})^2$.

Proof: From the proof of Theorem B.1.2 we find that to have a 4-round

iterative characteristic, the inputs to seven S-boxes must be different in the three nonzero rounds. Furthermore for at least one round the inputs to three neighbouring S-boxes must be different. There are three cases to consider.
Case A: By Lemma B.1.5 we know that for at least one S-box in $\Delta S(\Phi)$ the inputs differ in only one bit. Furthermore for at least one of the three neighbouring S-boxes in $\Delta S(\Psi)$ the outputs must be equal, otherwise $\Gamma \not\to \Phi$. There are two cases to consider:

1. For both S-boxes in $\Delta S(\Phi)$ the inputs differ in only one bit. By property 2 the outputs differ in at least two bits each. For every three neighbouring S-boxes in Ψ we know the only two possible S-boxes of $\Delta S(\Phi)$ by Lemma B.1.3 and Table B.2. Example: If $\Delta S(\Psi) = \{S1, S2, S3\}$ then $\Delta S(\Phi) = \{S5, S6\}$. Furthermore the outputs of either $S1$ or $S3$ must be equal.
   We have eight triples of three neighbouring S-boxes in Ψ to examine and from Tables B.1 and B.2 it follows that there are only three possible values for $\Delta S(\Psi)$ and $\Delta S(\Phi)$. From the difference distribution table, see [7], we find that the best combination for $\Psi \to \Phi$ has probability $\frac{8\times12\times10}{64^3}$. But then the probability for a 4-round iterative characteristic $\Pr(4R) \leq \frac{1}{4^4} \times \frac{8\times12\times10}{64^3} < (\frac{1}{234})^2$.

2. For one of the S-boxes in $\Delta S(\Phi)$ the inputs differ in one bit, for the other S-box the inputs differ in two bits. For every three neighbouring S-boxes of Ψ there are only two possibilities for the S-box in $\Delta S(\Phi)$, whose inputs differ in only one bit. From a closer look at Table B.1 it follows that $\Delta S(\Phi)$ must be neighbours and there are only two possible values for $\Delta S(\Psi)$ and $\Delta S(\Phi)$. from the difference distribution table we find that the best combination for $\Psi \to \Phi$ has probability $\frac{12\times10\times4}{64^3}$. But then the probability for the 4-round iterative characteristic $\Pr(4R) \leq \frac{1}{4^4} \times \frac{12\times10\times4}{64^3} < (\frac{1}{234})^2$.

Case B: The three S-boxes in $\Delta S(\Phi)$ are neighbours. From the proof of Theorem B.1.2 we have $\Delta S(\Gamma) = \Delta S(\Psi)$. Then by Lemma B.1.2 the outputs of each of the three neighbouring S-boxes in $\Delta S(\Phi)$ can differ in at most one bit, therefore the inputs must differ in at least two bits each by property 2. Then it follows from Table B.2 that for each of the S-boxes in $\Delta S(\Gamma)$ the outputs must differ in two bits. For every triple of three neighbouring in $\Delta S(\Phi)$ there is only one possible way for the inputs to differ and only one possibility

for $\Delta S(\Gamma)$. The best combination of $\Delta S(\Gamma)$ and $\Delta S(\Phi)$ gives a probability for the 4-round iterative characteristic $\Pr(4R) \leq \frac{12\times12\times16\times(8\times4)^2}{64^7} < (\frac{1}{234})^2$.
Case C: From Theorem B.1.2 we have $\Delta S(\Gamma) = \Delta S(\Psi)$. The only possibility we have for a 4-round iterative characteristic of this kind is when $\Delta S(\Gamma) = \{S2, S3, S4\}$ and $\Delta S(\Phi) = \{S7\}$. The best combinations yields a probability for the 4-round iterative characteristic

$$\Pr(4R) \leq \frac{1}{4^4} \times \frac{14 \times 8 \times 8}{64^3} < (\frac{1}{234})^2$$

□

B.2 Key Enumeration in LOKI'91

In this section we give an enumeration of the keys in the chosen plaintext attack of Section 6.2. We use the same notation as in Section 6.2. Let $\mathcal{A}$ be a function from $GF(2)^{64}$ to itself

$$\mathcal{A} : K_L \parallel K_R \rightarrow K_R \parallel Rol_{25}(K_L)$$

As stated above, once we have tried the key $K = K_L \parallel K_R$ in step 5 of the algorithm without success, we don't have to try the keys

$$\mathcal{A}(K),\ \overline{K},\ \mathcal{A}(\overline{K})$$

The idea is to use $\mathcal{A}$ to construct a set of keys about half the size of the key space and s.t.

- the biggest block of bits in every key consists of 1's.
- for every key K, $\mathcal{A}(K)$ is also in the set.

Then let the enumeration of the keys be every second key from the above constructed set of keys. Later in this section we show that the enumeration obtained this way makes the total number of keys tried in the attack be very close to 2^{62}.
Let $\mathcal{A}list(K)$ be the set of 64 keys $\{K, \mathcal{A}(K), \mathcal{A}^2(K), \ldots\ldots, \mathcal{A}^{63}(K)\}$. Note that $\mathcal{A}^{64}(K) = K$. Define for $K = K_L \parallel K_R$

$$\mathcal{M}_K = \cup_{p,q}\{\mathcal{A}list(Rol_p(K_L) \parallel Rol_q(K_R)) \cup \mathcal{A}list(Rol_p(K_p) \parallel Rol_q(K_L))\}$$

for $p = 0, 1, 2, 3$ and $q = 0, 8, 16, 24$.

Lemma B.2.1 *For $K = K_L \parallel K_R$, $\mathcal{M}_K$ is the set of all keys of the forms:*

$$Rol_x(K_L) \parallel Rol_y(K_R)$$
$$Rol_x(K_R) \parallel Rol_y(K_L)$$

for all $x, y \in \{0, 1, \ldots\ldots, 31\}$

Proof: For fixed K there are $2 \times 32 \times 32 = 2^{11}$ keys of the above form. Since $\mathcal{A}list$ produces 64 keys, the total number of keys in $\mathcal{M}_K$ is $2 \times 16 \times 64 = 2^{11}$. Therefore it suffices to show that the pairs of rotations of the keys in $\mathcal{M}_K$ are distinct, i.e., that $Rol_a(K_L) \parallel Rol_b(K_R)$ does not appear twice for any a, b. It is obvious that $Rol_a(K_L) \parallel Rol_b(K_R)$ does not appear twice in one $\mathcal{A}list$. There are two cases to consider, $Rol_a(K_L) \parallel Rol_b(K_R)$ appears in

1. $\dfrac{\mathcal{A}list(Rol_p(K_L) \parallel Rol_q(K_R)) \text{ and } \mathcal{A}list(Rol_{p'}(K_L) \parallel Rol_{q'}(K_R))}{Rol_a(K_L) \parallel Rol_b(K_R) = Rol_{p+25n}(K_L) \parallel Rol_{q+25n}(K_R) \wedge}$
   $Rol_a(K_L) \parallel Rol_b(K_R) = Rol_{p'+25n}(K_L) \parallel Rol_{q'+25n}(K_R) \Rightarrow$
   $p + 25n = p' + 25n \bmod 32 \wedge q + 25n = q' + 25n \bmod 32 \Rightarrow$
   $p - p' = q - q' \bmod 32 \Rightarrow (p, q) = (p', q')$,
   since $p - p' \in \{-3, -2, -1, 0, 1, 2, 3\}$ and $q - q' \in \{0, 8, 16, 24\}$.

2. $\dfrac{\mathcal{A}list(Rol_p(K_L) \parallel Rol_q(K_R)) \text{ and } \mathcal{A}list(Rol_{p'}(K_R) \parallel Rol_{q'}(K_L))}{Rol_a(K_L) \parallel Rol_b(K_R) = Rol_{p+25n}(K_L) \parallel Rol_{q+25n}(K_R) \wedge}$
   $Rol_a(K_L) \parallel Rol_b(K_R) = Rol_{q'+25n}(K_L) \parallel Rol_{p'+25+25n}(K_R) \Rightarrow$
   $p + 25n = q' + 25n \bmod 32 \wedge$
   $q + 25n = p' + 25 + 25n \bmod 32 \Rightarrow$
   $p + p' + 25 = q + q' \bmod 32$
   A contradiction, since $p + p' + 25 \in \{25, 26, \ldots, 31\}$ and
   $q + q' \bmod 32 \in \{0, 8, 16, 24\}$ □

Let Ka and Kb be two 32-bit key halves, s.t. Ka and Kb are no rotations of each other, i.e., $Rol_x(Ka) \neq Kb$ for any $x, 0 < x < 32$.
For $K = Ka \parallel Kb$, $\mathcal{M}_K$ is a set of distinct keys except in the cases where $Rol_x(Ka) = Ka$ for some x and/or $Rol_y(Kb) = Kb$ for some y.

Lemma B.2.2 *Let H be a 32-bit key. There are $2^{gcd(n,32)}$ possible values*

of H, such that $H = Rol_n(H)$, *where* $0 < n < 32$.

From Lemma B.2.2 it follows for $K = K_L \parallel K_R$, where K_L and K_R are no rotations of each other, that

Lemma B.2.3 *There at most* 2^{49} *keys for which the elements in* $\mathcal{M}_K$ *are not distinct.*

Proof: Assume we have two equal keys K' and $K*$ from $\mathcal{M}_K$. Then

$$K' = Rol_a(K_L) \parallel Rol_b(K_R),\ K* = Rol_c(K_L) \parallel Rol_d(K_R)$$

Clearly from the proof of Lemma B.2.1 $(a, b) \neq (c, d)$. Then

$$Rol_a(K_L) = Rol_c(K_L) \wedge Rol_b(K_R) = Rol_d(K_R) \Rightarrow$$
$$Rol_{a-c}(K_L) = K_L \wedge Rol_{b-d}(K_R) = K_R$$

If $a = c$ then there are 2^{32} possible values for K_L, but then there are at most 2^{16} possible values for K_R according to Lemma B.2.2, since $(a, b) \neq (c, d)$. If $b = d$ then $a \neq c$ and we get a total number of $2 \times 2^{32} \times 2^{16} = 2^{49}$ keys. □

The following algorithm makes a list of 32 bit strings, where no two strings are rotations of each other and where the biggest block of bits in every string consists of 1's.

ALGORITHM - No-rotations-of-keys (NRK)
For all positive $k \leq 32$, list all k-tuples $(a_1, a_2, \ldots, a_k)$, s.t.

1. $\sum_{i=1}^{k} a_i = 32$
2. $a_i \geq 1$ for $0 < i \leq k$
3. $\sum_{i=1}^{k} a_i \times 32^{k-i} \geq \sum_{i=1}^{k} a_{i+n \bmod (k+1)} \times 32^{k-i}$, for all $n \leq k$

Method: For every k-tuple $(a_1, \ldots\ldots, a_k)$ output the 32-bit key, where the a_1 MSB's are 1-bits, the next a_2 bits are 0-bits and so on.

Lemma B.2.4 *No two keys in the output from (NRK) are rotations of each other.*

Proof: Because of the inequality in 3. above if $k > 1$, then k is even. Therefore for $k > 1$ the a_k LSB are 0-bits and furthermore $a_1 \geq a_i$ for $i \leq k$. Let A and A' be two 32 bit keys from (NRK), s.t. $A = Rol_x(A')$ for some fixed x. Write A and A' as tuples $(a_1, \ldots, a_m)$ and $(a'_1, \ldots, a'_l)$ according to the method in (NRK). Clearly $l = m$ otherwise A cannot be a rotation of A'. Because $A = Rol_x(A')$ we have for some i

$$a_{1+n} = a'_{i+n \bmod (m+1)}, \quad 0 < n \leq m$$

Especially we have $a_1 = a'_i$ and $a_{m-i+2} = a'_1$. Because of the inequality in 3. above we have

$$a'_i \leq a'_1 \wedge a_1 \leq a_{m-i+2}$$

Therefore $a'_i = a'_1 \Rightarrow a_1 = a'_1$. Now $a_1 = a_{m-i+2} \Rightarrow a_2 \geq a_{m+i+3}$. Similar as before

$$a_2 = a'_{i+1} \leq a'_2 = a_{m-i+3} \Rightarrow a'_2 = a_2$$

By induction we obtain $A = A'$ □

ALGORITHM - Enumeration (EN)

1. $i = 1$
2. Let K_L be the i'th output from (NRK)
3. For $j = 1$ to i do
   (a) Let K_R be the j'th output from (NRK)
   (b) For $K = K_L \parallel K_R$ output the first and then every second key from all $\mathcal{A}lists$ in $\mathcal{M}_K$
   (c) For $K = \overline{K_L} \parallel K_R$ do as in 3b
4. Set $i = i + 1$ and goto 2

We are left to check whether the set

$$KS = \cup_{K_i}\{Ki, \mathcal{A}(Ki), \overline{Ki}, \overline{\mathcal{A}(Ki)}\}$$

where the Ki's are the keys output from (EN), contains the entire key space. Let $K^* = K_L^* \parallel K_R^*$ be an arbitrary key. Rotate K_L^* and K_R^* such that the biggest blocks of bits (0's or 1's) are the MSB. Let $K(j) = K'_L \parallel K'_R$ be that key.

If the MSB in both K'_L and K'_R are 1's then they are both output from (NRK). Then at some point $K(j)$ or $K(l) = K'_R \parallel K'_L$ say $K(j)$, is the key considered in step 3(b) of (EN). Let $K(n), 0 < n < 2^{10}$ be all keys output in step 3(b) when $K = K(j)$. Then $L = \{K(n), \mathcal{A}(K(n))\}, 0 < n \leq 2^{10}$ are all rotations of the key halves in $K(j)$ according to Lemma B.2.1. Therefore $K^* \in L \in KS$.
If MSB in both K'_L and K'_R are 0's, then at some point either $\overline{K(j)}$ or $\overline{K(l)}$ is the key considered in step 3(b). Let $K(n)$ be as before, when $K = \overline{K(j)}$ Then $L = \{K(n), \mathcal{A}(K(n))\}, 0 < n \leq 2^{10}$ are all rotations of the key halves in $K(j)$ according to Lemma B.2.1. Therefore $\overline{K^*} \in L \in KS \Rightarrow K^* \in \overline{L} \in KS$.
If the MSB in K'_L and K'_R are 1's and 0's resp. or vice versa similar arguments hold for step 3(c).
We have implemented (NRK) on a SUN-Spare workstation. The number of key halves output from (NRK) is $2^{26} + 2068$. It means that the number of keys output from (NRK) in 2. and 3(a) above is about

$$\frac{(2^{26} + 2068)^2}{2} \simeq 2^{51} + 2^{37}.$$

Every second key from $\mathcal{M}_K$ gives 2^{10} keys for each K. The total number of keys in the enumeration therefore is about

$$(2^{51} + 2^{37}) \times 2 \times 2^{10} = 2^{62} + 2^{48}.$$

We have given an enumeration of the keys, s.t. the total number of iterations of step 5 in the algorithm of the chosen plaintext attack is close to 2^{62}. The time used in the enumeration (EN) is negligible compared to the 1.07×2^{62} full 16 rounds of encryption of LOKI'91, since it runs only once per every 2×2^{10} runs of step 5 in the algorithm of the chosen plaintext attack.

Appendix C

The Data Encryption Standard

The Data Encryption Standard was published as a Federal Information Processing Standard in January 1977 [90]. In this section we give a full description of the algorithm.

The algorithm encrypts a 64 bit plaintext block P into a 64 bit ciphertext block C using a 64 bit key K. The bits are numbered from 1 to 64, s.t. the most significant bit has the lowest index, i.e., $P = p_1, \ldots, p_{64}$, where the p_i's are bits. The key K is the input to the key schedule algorithm (described below), which outputs 16 round keys, each of 48 bits. The plaintext is first permuted using the permutation IP, Table C.2, s.t.

$$\begin{aligned} P &= p_1, \ldots, p_{64} \\ IP(P) &= p_{58}, p_{50}, \ldots, p_7 \end{aligned}$$

thereafter $IP(P)$ is divided into two halves P^L and P^R of 32 bits each. Given a permuted plaintext $IP(P) = (P^L, P^R)$ and the round keys $(K_1, K_2, \ldots, K_{16})$ the ciphertext $C = (C_L, C_R)$ is computed in 16 rounds. Set $C_0^L = P^L$ and $C_0^R = P^R$ and compute for $i = 1, 2, \ldots, 16$

$$(C_i^L, C_i^R) = (C_{i-1}^R, F(C_{i-1}^R, K_i) \oplus C_{i-1}^L)$$

Set $C_i = (C_i^L, C_i^R)$ for $i = 1, \ldots, 15$, and $C_{16} = (C_{16}^R, C_{16}^L)$. The ciphertext is then permuted using the inverse of IP, i.e., the ciphertext is $C = IP^{-1}(C_{16})$. Note that the halves are not swapped after the last evaluation of F. In that way the decryption algorithm equals the encryption algorithm, except that

the round keys are used in reverse order, i.em., K_{16} is used in the first round, K_{15} in the second and so on.

The function F is depicted in Figure C.1. The 32 bit text input X is expanded by the E-expansion, Table C.3. Next, the exclusive-or of $E(X)$ and a 48 bit round key is divided into 8 blocks of six bits each. The 8 blocks are used as the inputs to the 8 S-boxes in Table C.1 in the following way. Let $b_1, b_2, b_3, b_4, b_5, b_6$ be six input bits to an S-box. The integer corresponding to the bits b_1b_6 selects one of four rows in the S-box and the integer $b_2b_3b_4b_5$ selects one of sixteen columns. The 32 bits output from the S-boxes are concatenated and permuted using the permutation P, Table C.4.

The key schedule algorithm takes a 64 bit key K as input. Initially the parity bits of K are removed and the remaining 56 bits are permuted, using the permuted choice function PC-1, Table C.5. The result PC-1(K) is divided into two halves C_0 and D_0. The halves are then rotated successively to the left in the following way, $C_i = LS_i(C_{i-1})$ and $D_i = LS_i(D_{i-1})$, where the number of shifts LS_i is listed in Table C.7. The 16 round keys K_i, $i = 1, \ldots, 16$ are computed as follows K_i = PC-2(C_i, D_i), where PC-2 is the permuted choice function in Table C.6.

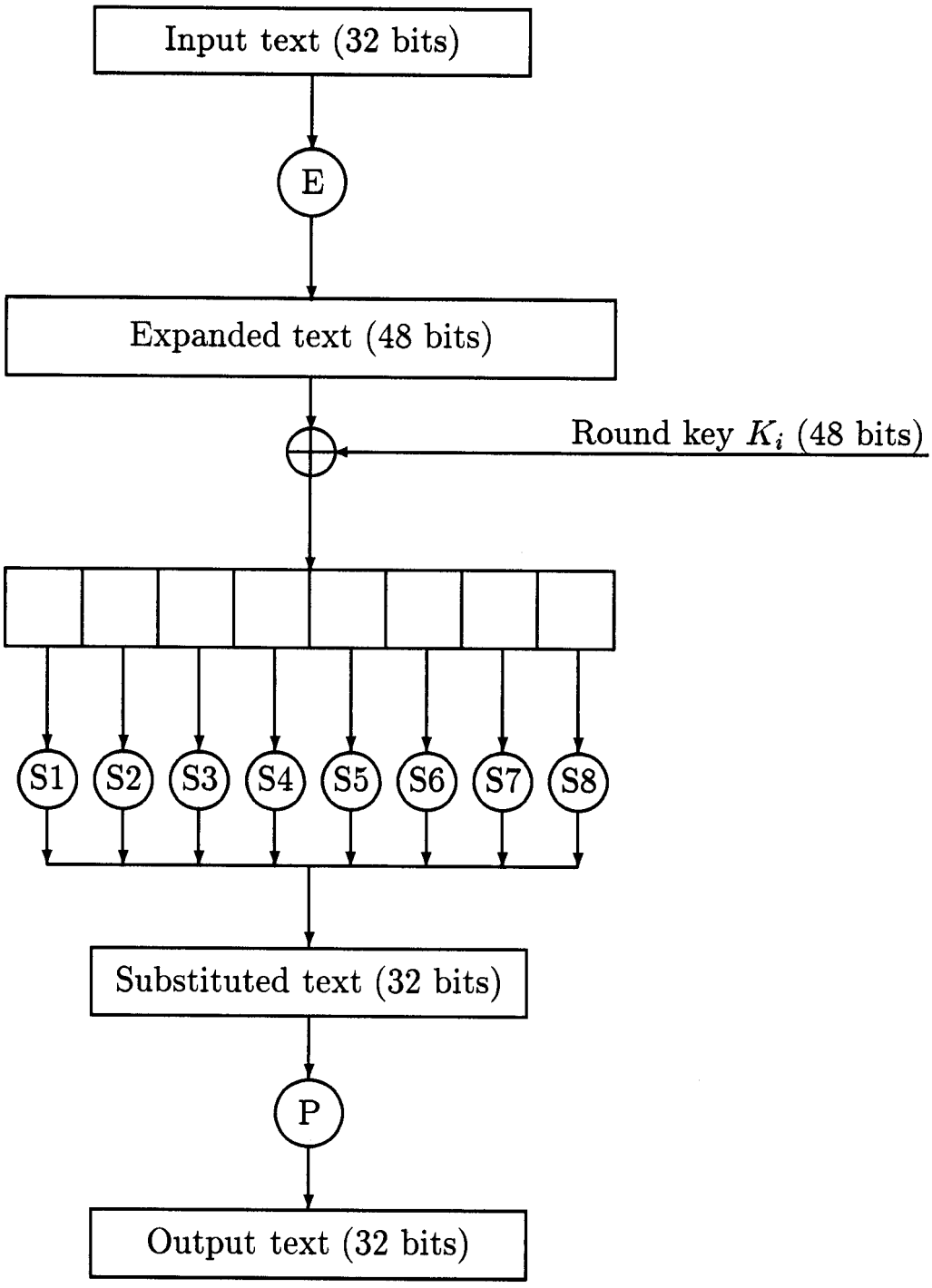


Figure C.1: The F-function in DES.

S1:	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S2:	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S3:	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S4:	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S5:	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S6:	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S7:	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S8:	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Table C.1: The 8 S-boxes of the DES.

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Table C.2: The initial permutation IP.

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Table C.3: The E-expansion.

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Table C.4: The P-permutation.

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Table C.5: The initial permuted choice PC-1.

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Table C.6: The permuted choice table PC-2.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LS_i	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Table C.7: The circular shifts in the key schedule of DES.

Appendix D

LOKI'91

In 1990 Brown, Seberry and Pieprzyk [15] proposed a new encryption primitive, called LOKI, later renamed LOKI'89, and a redesign, LOKI'91 was proposed in [14]. In this section we give a full description of the LOKI'91.

The algorithm encrypts a 64 bit plaintext block P into a 64 bit ciphertext block C using a 64 bit key K. The bits are numbered from 64 to 1, i.e., the opposite way as in the description of the DES, s.t. $P = p_{64}, \ldots, p_1$, where the p_i's are bits. The key K is the input to the key schedule algorithm (described below), which outputs 16 round keys, each of 32 bits. A plaintext P is divided into two halves P^L and P^R of 32 bits each. Given a plaintext $P = (P^L, P^R)$ and the round keys $(K_1, K_2, \ldots, K_{16})$ the ciphertext $C = (C_L, C_R)$ is computed in 16 rounds. Set $C_0^L = P^L$ and $C_0^R := P^R$ and compute for $i = 1, 2, \ldots, 16$

$$(C_i^L, C_i^R) = (C_{i-1}^R, F(C_{i-1}^R, K_i) \oplus C_{i-1}^L$$

Set $C_i = ((C_i^L, C_i^R)$ for $i = 1, \ldots, 15$, and $C_{16} = (C_{16}^L, C_{16}^R)$. The ciphertext is $C = C_{16}$.

The function F is depicted in Figure D.1. The 32 bit text input X is exclusive-or'ed to a 32 bit round key and is thereafter expanded to 48 bits by the E-expansion, Table D.2. Next the bits are divided into 4 blocks of 12 bits each. Each block is substituted by a 8 bit value by SB the only S-box in LOKI'91, in the following way.

Let $b_{12}, b_{11}, \ldots, b_2, b_1$ be twelve input bits to an S-box. The integer $r = b_{12}, b_{11}, b_2, b_1$ and the integer $c = b_{10}, b_9, \ldots, b_4, b_3$ are formed. The integer

r selects one of sixteen irreducible polynomials, pol_r, from Table D.1. The S-box operates as follows

$$SB(r, c) = (c + ((r * 17) \oplus ff_x)\&ff_x)^{31} \text{mod } pol_r$$

where ff_x is a byte with eight 1-bits, '+' and '$*$' refer to arithmetic addition and multiplication, $\oplus$ is addition modulo 2, & is bitwise and-operation, and the exponentiation is performed in $GF(2^8)$.

The four times eight bits output from the S-boxes are concatenated and permuted using the permutation P, Table D.3.

The key schedule algorithm takes a 64 bit key K as input. The key is divided into two 32 bit halves K_L, K_R and the 16 round keys $K(i), i = 1, \ldots, 16$, are derived as follows:

1. $i = 1$
2. $K(i) = K_L; i = i + 1$
3. $K_L = Rol_{13}(K_L)$
4. $K(i) = K_L; i = i + 1$
5. $K_L = Rol_{12}(K_L)$
6. $Swap(K_L, K_R)$
7. go to 2.

where $Rol_n(X)$ is a bitwise rotation of X, n positions to the left and $Swap(X, Y)$ swaps X and Y.

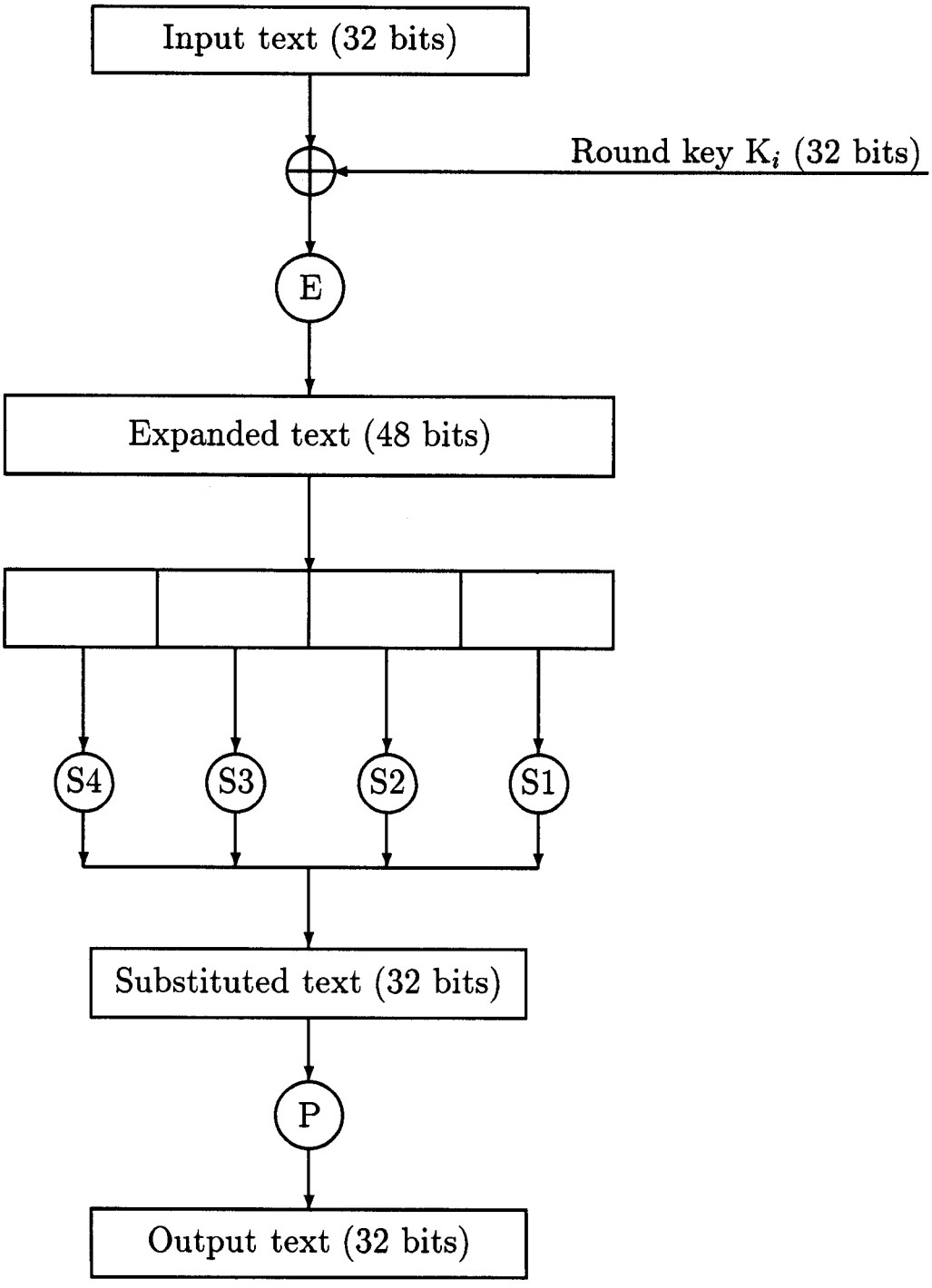


Figure D.1: The F-function in LOKI

r	1	2	3	4	5	6	7	8
pol_r	375	379	391	395	397	415	419	425
r	9	10	11	12	13	14	15	16
pol_r	433	455	451	463	471	477	487	499

Table D.1: The 16 irreducible polynimials in decimal notation

3	2	1	0	31	30	29	28	27	26	25	24
27	26	25	24	23	22	21	20	19	18	17	16
19	18	17	16	15	14	13	12	11	10	9	8
11	10	9	8	7	6	5	4	3	2	1	0

Table D.2: The E-expansion

31	23	15	7	30	22	14	6
29	21	13	5	28	20	12	4
27	19	11	3	26	18	10	2
25	17	9	1	24	16	8	0

Table D.3: The permutation P.

Appendix E

Dansk resume

Denne licentiatafhandling omhandler kryptoanalyse, anvendelser og design af konventionelle blokkryptosystemer. Hovedvægten er lagt på en speciel og vigtig klasse af bloksystemer, de såkaldte *Feistel*-systemer. Disse systemer består af et antal runder, hvor man i hver runde bruger en kryptografisk svag funktion, som ved iteration forventes af were stærk.

Anvendelser

Den mest udbredte anvendelse af et blokkryptosystem er til kryptering. De forskellige måder at bruge et bloksystem til kryptering gennemgåes og analyseres, og vi introducerer et nyt klassifikationssystem af mulige angreb på bloksystemer. Dette udmønter sig i en ny (teoretisk) øvre grænse for kompleksiteten af angreb på bloksystemer.

En anden anvendelse af bloksystemer er som bygge-klods i kryptografiske hashfunktioner, hvilke vi gennemgår og analyserer.

Endelig analyseres en tredje anvendelse af bloksystemer, som byggeklods i digitale signatursystemer. Specielt vises, at et system foreslået af R. Merkle er både sikkert og praktisk med passende og rimelige antagelser om bloksystemet.

Kryptoanalyse

De vigtigste kendte angreb pa bloksystemer, lineaer og differentiel kryptoanalyse gennemgås og forbedringer til begge metoder gives. Endvidere introducerer vi et nyt angreb baseret på såkaldte *simple relationer* og anvender det på et australsk foreslået bloksystem. Differentiel kryptoanalyse bruger såkaldte *differentialer* (A, B), d.v.s. et par af klartekster med differens A, som efter et vist antal runder resulterer i differensen B med en ikke ubetydelig sandsynlighed. Derved kan (dele af) den hemmelige nøgle findes. Vi giver nye metoder til, hvordan de bedste sådanne differentialer kan bestemmes. Endvidere introducerer vi *højere ordens differentialer* og *delvise differentialer* og viser, at begge har brugbare anvendelser. Ovennævnte angreb udføres på fem specifikke bloksystemer, DES, LOKI'91, s^2-DES, xDES^1 og xDES^2.

Angreb på hashfunktioner baseret på bloksystemer klassificeres og nye angreb på en stor klasse af hashfunktioner introduceres, og anvendes på tre eksisterende systemer, som vises ikke at være så gode, som tidligere antaget. En fjerde hashfunktion, AR Hash, tilhørende en anden klasse af hashfunktioner, studeres. Systemet er hurtigere end de kendte standarder og blev brugt i den tyske bankverden. Vi viser, at systemet er totalt usikkert.

Design

Principper for design af nye bloksystemer diskuteres. Eksisterende systemer er baseret på ad hoc metoder, og ingen egentlig teori eksisterer på området. Vi giver nye nedre grænser for kompleksiteterne af angreb på bloksystemer ved lineær og differentiel kryptoanalyse, og viser, at der eksisterer funktioner, som kan bruges til konstruktion af kryptosystemer, bevisligt sikre mod de to angreb. Endvidere defineres såkaldte *stærke nøgleskemaer*. I et rundebaseret kryptosystem anvendes som oftest et nøgleskema, som med inddata en relativ kort nøgle giver en række rundenøgler. Et kryptosystem med et stærkt nøgleskema vises at være immun overfor angreb baseret på simple relationer. En simpel metode til konstruktion af stærke nøgleskemaer angives. En udbredt metode til forøgelse af sikkerheden af et konventionelt kryptosystem er ved *gentagne krypteringer*, d.v.s. hvor en klartekst krypteres adskillige gange med samme kryptosystem, men hver gang med en forskellig nøgle. Ved denne metode er det nødvendigt at sikre sig, at sikkerheden for et sys-

tem ved gentagen kryptering ikke er lavere end sikkerheden for systemet ved en enkelt kryptering. Vi analyserer eksisterende metoder for gentagen kryptering og giver et nyt forslag til et system, som er bevisligt ligeså sikkert som engangskryptering og med et minimalt forbrug af forskellige nøgler.

Dele af arbejdet i denne afhandling er skrevet som separate artikler. I samarbejde med lektor Ivan B. Damgård artiklerne [19, 20], med Kaisa Nyberg artiklerne [85, 86], med Xuejia Lai artiklerne [53, 57] og med Luke O'Connor artiklen [54]. Endvidere har undertegnede selv skrevet artiklerne [47, 48, 49, 50, 51, 52].

List of Figures

List of Tables

Bibliography

[1] R. Ash. *Infomation Theory.* Interscience Publishers, 1965.

[2] T. Beth and C. Ding. On almost perfect nonlinear permutations. In T. Helleseth, editor, *Advanced in Cryptology - Proc. Eurocrypt'93, LNCS 765*, pages 65–76. Springer Verlag, 1993.

[3] E. Biham. New types of cryptanalytic attacks using related keys. In T. Helleseth, editor, *Advances in Cryptology - Proc. Eurocrypt'98, LNCS 765*, pages 398–409. Springer Verlag, 1993.

[4] E. Biham. On Matsui's linear cryptanalysis. In *Advances in Cryptology - Proc. Eurocrypt'94.* Springer Verlag, 1994. To appear.

[5] E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.

[6] E. Biham and A. Shamir. Differential cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer. In J. Feigenbaum, editor, *Advances in Cryptology - Proc. Crypto'91, LNCS 576*, pages 156–171. Springer Verlag, 1992.

[7] E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard.* Springer Verlag, 1993.

[8] E. Biham and A. Shamir. Differential cryptanalysis of the full 16-round DES. In E.F. Brickell, editor, *Advances in Cryptology - Proc, Crypto'92, LNCS 740*, pages 487–496. Springer Verlag, 1993.

[9] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, pages 856–864, 1984.

[10] B.O. Brachtl, D. Coppersmith, M.M. Hyden, SM. Matyas, C.H. Meyer, J. Oseas, S. Pilpel, and M. Schilling. Data authentication using modification detection codes based on a public one way encryption function. U.S. Patent Number 4,908,861, March 13 1990.

[11] E.F. Brickell, D. Denning, S.T. Kent, D.P. Maher, and W. Tuchmann. Skipjack review - interim report. On sci.crypt (Internet) August, 1993.

[12] E.F. Brickell, J.H. Moore, and M.R. Purtill. Structure in the S-boxes of the DES. In A.M. Odlyzko, editor, *Advances in Cryptology - Crypto'86, LNCS 263*, pages 3–8. Springer Verlag, 1987.

[13] E.F. Brickell and A.M. Odlyzko. Cryptanalysis: A survey of recent results. In G.J. Simmons, editor, *Contemporary Cryptology - The Science of Information Integrity*, chapter 10, pages 501–540. IEEE Press, 1992.

[14] L. Brown, M. Kwan, J. Pieprzyk, and J. Seberry. Improving resistance against differential cryptanalysis and the redesign of LOKI. In H. Imai, R.L. Rivest, and T. Matsumoto, editors, *Advances in Cryptology - Proc. AsiaCrypt'91, LNCS 453*, pages 36–50. Springer Verlag, 1993.

[15] L. Brown, J. Pieprzyk, and J. Seberry. LOKI - a cryptographic primitive for authentication and secrecy applications. In J. Seberry and J. Pieprzyk, editors, *Advances in Cryptology - Proc, AusCrypt'90, LNCS 453*, pages 229–236. Springer Verlag, 1990.

[16] D. Coppersmith. The real reason for Rivest's phenomenon. In H.C. Williams, editor, *Advances in Cryptology - Proc. Crypto '85, LNCS 218*, pages 535–536. Springer Verlag, 1986.

[17] T. Cusick and M. Wood. The REDOC-II cryptosystem. In A.J. Menezes and S.A. Vanstone, editors, *Advances in Cryptology - Proc. Crypto'90, LNCS 537*, pages 545–563. Springer Verlag, 1991.

[18] I.B. Damgård. A design principle for hash functions. In G. Brassard, editor, *Advances in Cryptology - Proc. Crypto'89, LNCS 435*, pages 416–427. Springer Verlag, 1990.

[19] I.B. Damgård and L.R. Knudsen. The breaking of the AR hash function. In T. Helleseth, editor, *Advances in Cryptology - Proc. Eurocrypt '93, LNCS*, pages 286–292. Springer Verlag, 1993.

[20] I.B. Damgård and L.R. Knudsen. Methods for enhancing the strength of conventional cryptosystems. m s Internal document, 1994.

[21] D.W. Davies. Some regular properties of the DES. In *Advances in Cryptology - Proc. Crypto 82*, pages 89–96. Plenum Press, 1983.

[22] D.W. Davies and W.L. Price. *Security fov Computer Networks.* John Wiley & Sons, 1989.

[23] D.E. Denning. *Cryptography and Data Security.* Addison-Wesley 1982.

[24] Y. Desmedt, J.-J. Quisquater, and M. Davio. Dependence of output on input in DES: Small avalanche characteristics. In G.R. Blakley and D. Chaum, editors, *Advances in Cryptology - Proc. Crypto'84, LNCS 196*, pages 359–376. Springer Verlag, 1985.

[25] J. Detombe and S. Tavares. Constructing large cryptographically strong S-boxes. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology - Proc. AusCrypt 92, LNCS 718*, pages 165–181. Springer Verlag, 1993.

[26] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. on Information Theory*, IT-22(6):644–654, 1976.

[27] J.H. Evertse. Linear structures in blockciphers. In D. Chaum and W.L. Price, editors, *Advances in Cryptology - Proc. Eurocrypt'87, LNCS 304*, pages 249–266, 1988.

[28] H. Feistel. Cryptography and computer privacy. *Scientific American*, 228(5):15–23, 1973.

[29] W. Feller. *Probability Theory and its Applications.* John Wiley & Sons, 1950.

[30] P. Flajolet and A.M. Odlyzko. Random mapping statistics. In J.-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology - Eurocrypt '89, LNCS 434*, pages 329–354. Springer Verlag, 1990.

[31] S. Goldwasser, S. Micali, and R.L. Rivest. A "paradoxical" solution to the signature problem. In *Proc. 25th IEEE Symposium on Foundations of Computer Science*, pages 441–448, 1984.

[32] S. Goldwasser, S. Micali, and R.L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[33] M. Hellman. A cryptanalytic time-memeory trade-off. *IEEE Trans. on Information Theory*, IT-26(4):401–406, 1980.

[34] M.E. Hellman and S.K. Langford. Differential–linear cryptanalysis. In Y. G. Desmedt, editor, *Advances in Cryptology - Proc. Crypto'94, LNCS 839*, pages 26–39. Springer Verlag, 1994.

[35] M.E. Hellman, R. Merkle, F. Schroeppel, L. Washington, W. Diffie, S. Pohlig, and P. Schweitzer. Results of an initial attemt to crypt analyse the NBS Data Encryption Standard. Technaicl report, Stanford University, U.S.A., September 1976.

[36] H.M. Heyes and S.E. Tavares. The design of product ciphers resistrant to differnetial and linear cryptanalysis. Presented at the rump session of Crypto'93, 1993.

[37] W. Hohl, X. Lai, T. Meier, and C. Waldvogel. Security of iterated hash function based on block ciphers. In D. R. Stinson, editor, *Advances in Cryptology - Proc. Crypto'93, LNCS 773*, pages 379–390. Spriger verlag, 1993.

[38] ISO-10118. Information technology — security techniques – hash-functions, part 1: general and part 2: Hash-functions using an n-bit block cipher algorithm. ISO/IEC, 1994.

[39] C.J.A. Jansen and D.E. Boekee. Modes of blockcipher algorithns and their protection against active eavesdropping. In D. Chaum and W.L. Price, editors, *Advances in Cryptology - Proc. Eurocrypt'87, LNCS 304*, pages 327–347. Springer Verlag, 1988.

[40] D. Kahn. *The Codebreakers.* MacMillan, 1967.

[41] J.B. Kam and G.I. Davida. A stryctured design of substitution-permutation encryption networks. *IEEE Transaction on Computers*, C-28(10):747–753, 1979.

[42] M.G. Karpocsky. *Finite Orthogonal Series in the Design of Digital Devices.* Wiley, New York, 1976.

[43] K. Kim. Constryciton of Des-like S-boxes based on boolean functions satisfying the sac. In H. Imai, R.L. Rivest, and T. Matsumoto, editors, *Advances in Cryptology - Proc. AsiaCrypt'91, LNCS 453*, pgaes 59–72. Springer Verlag, 1993.

[44] K. Kim, S. Park, and S. Lee. The reconstruction of s^2-DES s-boxes and their immunity to differential cryptanalysis. Presented at Korea-Japan Joint Workshop on Information Security and Cryptology (JW-ISC'93), Seoul, 1993.

[45] K. Kim, S. Park, and S. Lee. The reconstruction of s^2-DES s-boxes and their immunity to differential cryptanalysis. Unpublished manuscript, preliminary draft, 1993.

[46] L.R. Knudsen. Differential cryptanalysis of DES and LOKI. Master's thesis, Århus University, 1991. (In danish).

[47] L.R. Knudsen. Cryptanalysis of LOKI' 91. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology, AusCrypt 92, LNCS 718.* Springer Verlag, 1993.

[48] L.R. Knudsen. Cryptanalysis of LOKI. In H. Imai, R.L. Rivest, and T. Matsumoto, editors, *Advances in Cryptology - Proc. AsiaCrypt'91, LNCS 458*, pages 22–35. Springer Verlag, 1993.

[49] L.R. Knudsen. Iterative characteristics of DES and s^2-DES. In E.F. Brickell, editor, *Advances in Cryptology - Proc. Crypto'92, LNCS 740*, pages 497–511. Springer Verlag, 1993.

[50] L.R. Knudsen. Practically secure Feistel ciphers. In *Advances in Cryptology - Proc. Algorithm Workshop, Cambridge, U.K.* Springer Verlag, 1993. To appear.

[51] L.R. Knudsen. Applications of higher order and partial differentials. Internal document, 1994.

[52] L.R. Knudsen. New potentially weak keys for DES and LOKI. In *Advances in Cryptology - Proc. Eurocrypt '94.* Springer Verlag, 1994. To appear.

[53] L.R. Knudsen and X. Lai. New attacks on all double block length hash functions of hash rate 1, including the parallel-DM. In *Advances in Cryptology - Proc. Eurocrypt'94.* Springer Verlag, 1994. To appear.

[54] L.R. Knudsen and L.J. O'Connor. A survey of results related to differential cryptanalysis. In preparation, 1994.

[55] X. Lai. *On the Design and Security of Block Ciphers.* PhD thesis, ETH, Zurich, Switzerland, 1992.

[56] X. Lai. Higher order derivatives and differential cryptanalysis. In *Proc. "Symposium on Communication, Coding and Cryptography", in honor of James L. Massey on the occasion of his 60'th birthday, Feb. 10-13, 1994, Monte-Verita, Ascona, Switzerland*, 1994. To appear.

[57] X. Lai and L.R. Knudsen. Attacks on double block length hash functions. In *Advances in Cryptology - Proc. Algorithm Workshop, Cambridge, U.K.*, pages 157–165. Springer Verlag, 1994. To appear.

[58] X. Lai, J.L. Massey, and S. Murphy. Markov ciphers and differential cryptanalysis. In D.W. Davies, editor, *Advances in Cryptology - Proc. Eurocrypt'91, LNCS 547*, pages 17–38. Springer Verlag, 1992.

[59] R. Lidl and H. Niederreiter. Finite fields. In *Encyclopedia of Mathematics and its applications*, volume 20. Addison-Wesley, Reading, Massachusetts, 1983.

[60] Algorithmic Research Ltd. Ar fingerprint function. ISO-IEC/JTC1/SC27/WG2 N179, working document, 1992.

[61] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal of Computing*, 17(2):373–386, 1988.

[62] S.S. Magliveras and N.D. Memon. Algebraic properties of cryptosystem PGM. *Journal of Cryptology*, 5(3):167–184, 1992.

[63] J.L. Massey. Cryptography: Fundamentals and applications. Copies of transparencies, Advanced Technology Seminars, 1993.

[64] M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology - Proc. Eurocrypt'93, LNCS 765*, pages 386–397. Springer Verlag, 1993.

[65] M. Matsui. Linear cryptanalysis method of DES cipher (I). Private communications, 1993.

[66] M. Matsui. The first experimental cryptanalysis of the Data Encryption Standard. In Y. G. Desmedt, editor, *Advances in Cryptology - Proc. Crypto'94, LNCS 839*, pages 1–11. Springer Verlag, 1994.

[67] M. Matsui. On correlation between the order of S-boxes and the strength of DES. In *Abstracts of Eurocrypt'94*. Springer Verlag, 1994.

[68] M. Matsui and A. Yamagishi. A new method for known plaintext attack of FEAL cipher. In R. A. Rueppel, editor, *Advances in Cryptology - Eurocrypt'92, LNCS 658*, pages 81–91. Springer Verlag, 1993.

[69] U. Maurer and J.L. Massey. Cascade ciphers: The importance of being first. *Journal of Cpyptology*, 6(1):55–61, 1993.

[70] U.M. Maurer. New approaches to the design of self-synchronizing stream ciphers. In D.W. Davies, editor, *Advances in Cryptology - Eurocrypt'91, LNCS 547*, pages 458–471. Springer Verlag, 1991.

[71] W. Meier and 0. Staffelbach. Nonlinearity criteria for cryptographic functions. In J.-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology - Eurocrypt'89, LNCS 434*, pages 549–562. Springer Verlag, 1990.

[72] R. Merkle. A digital signature based on a conventional encryption function. In C. Pomerance, editor, *Advances in Cryptology - Crypto'87, LNCS 293*, pages 369–378. Springer Verlag, 1988.

[73] R. Merkle. A certified digital signature. In G. Brassard, editor, *Advances in Cryptology - Crypto'89, LNCS 435*, pages 218–238. Springer Verlag, 1990.

[74] R. Merkle. One way hash functions and DES. In G. Brassard, editor, *Advances in Cryptology - Crypto'89, LNCS 435*, pages 428–446. Springer Verlag, 1990.

[75] R. Merkle. Fast software encryption functions. In A.J. Menezes and S.A. Vanstone, editors, *Advances in Cryptology - Proc. Crypto'90, LNCS 537*, pages 476–501. Springer Verlag, 1991.

[76] R. Merkle and M. Hellman. On the security of multiple encryption. *Communications of the ACM*, 24(7):465–467, 1981.

[77] C. Meyer and M. Schilling. Secure program load with manipulation detection code. In *Proceedings of SECURICOM 1988*, pages 111–130, 1988.

[78] C.J. Mitchell, F. Piper, and P. Wild. Digital signatures. In G.J. Simmons, editor, *Contemporary Cryptology - The Science of Information Integrity*, chapter 6, pages 325–378. IEEE Press, 1992.

[79] S. Miyaguchi. The FEAL cipher family. In A.J. Menezes and S.A. Vanstone, editors, *Advances in Cryptology - Proc. Crypto'90, LNCS 537*, pages 627–638. Springer Verlag, 1990.

[80] J.H. Moore and G.J. Simmons. Cycle structure of the DES with weak and semiweak keys. In A.M. Odlyzko, editor, *Advances in Cryptology - Crypto'86, LNCS 263*, pages 9–32. Springer Verlag, 1987.

[81] S. Murphy, K. Paterson, and P. Wild. A weak cipher that generates the symmetric group. *Journal of Cryptology*, 7(1):61–65, 1994.

[82] J. Nechvatal. Public key cryptography. In G.J. Simmons, editor, *Contemporary Cryptology - The Science of Information Integrity*, chapter 4, pages 177–288. IEEE Press, 1992.

[83] K. Nyberg. Differentially uniform mappings for cryptography. In T. Helleseth, editor, *Advances in Cryptology - Proc. Eurocrypt'93, LNCS 765*, pages 55–64. Springer Verlag, 1993.

[84] K. Nyberg. Linear approximations of block ciphers. In *Advances in Cryptology - Proc. Eurocrypt'94*. Springer Verlag, 1994. To appear.

[85] K. Nyberg and L.R. Knudsen. Provable security against differential cryptanalysis. In E.F. Brickell, editor, *Advances in Cryptology - Proc. Crypto'92, LNCS 740*, pages 566–574. Springer Verlag, 1993.

[86] K. Nyberg and L.R. Knudsen. Provable security against a differential attack. *The Journal of Cryptology*, 1994. To appear.

[87] L.J. O'Connor. Differential cryptanalysis, Markov chains and random graph theory. Private communication.

[88] L.J. O'Connor. *An analysis of product ciphers using boolean functions.* PhD thesis, Department of Computer Science, University of Waterloo, 1992.

[89] L.J. O'Connor. On the distribution of characteristics in bijective mappings. In T. Helleseth, editor, *Advances in Cryptology - Eurocrypt 93, LNCS 765*, pages 360–370. Springer Verlag, 1994.

[90] National Bureau of Standards. Data encryption standard. Federal Information Processing Standard (FIPS), Publication 46, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., January 1977.

[91] National Bureau of Standards. DES modes of operation. Federal Information Processing Standard (FIPS), Publication 81, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., December 1980.

[92] B. Preneel. Announced at Crypto'93. During the presentation of [95]

[93] B. Preneel. *Analysis and Design of Cryptographic Hash Fuctions.* PhD thesis, Katholieke Universiteit Leuven, January 1993.

[94] B. Preneel. Differential cryptanalysis of hash functions based on block ciphers. In *Proc. of 1'st ACM Conference on Computer and Communications Security*, Nov. 3-5 1993.

[95] B. Preneel. Hash functions based on block ciphers: A synthetic approach. In D.R. Stinson, editor, *Advances in Cryptology - Proc. Crypto'93, LNCS 773*, pages 368–378. Springer Verlag, 1993.

[96] B. Preneel, A. Bosselaers, R. Govaerts, and J. Vandewalle. Collision-free hashfunctions based on blockcipher algorithms. In *Proceedings of 1989 International Carnahan Conference on Security Technology*, pages 203–210, 1989.

[97] B. Preneel, M. Nuttin, V. Rijmen, and J. Buelens. Differential cryptanalysis of the CFB mode. In D.R. Stinson, editor, *Advances in Cryptology - Proc. Crypto'93, LNCS 773*, pages 212–223. Springer Verlag, 1993.

[98] J.-J. Quisquater and J.-P. Delescaille. How easy is collision search. Applications to DES. In J.-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology - Eurocrypt '89, LNCS 434*, pages 429–433. Springer Verlag, 1990.

[99] J.-J. Quisquater and J.-P. Delescaille. How easy is collision search. New results and applications to DES. In G. Brassard, editor, *Advances in Cryptology - Cyypto'89, LNCS 435*, pages 408–413. Springer Verlag, 1990.

[100] J.-J. Quisquater, Y. Desmedt, and M. Davio. The importance of 'good' key scheduling schemes. In H.C. Williams, editor, *Advances in Cryptology - Proc. Crypto'85, LNCS 218*, pages 537–542. Springer Verlag, 1986.

[101] J. Rompel. One-way functions are necessary and sufficient for secure signature. In *STOC*, pages 387–394, 1990.

[102] I. Schaumüller-Bichl. *Zur Analyze des Data Encryption Standard und Synthese Verwandter Chiffriersysteme.* PhD thesis, Linz University, May 1981.

[103] I. Schaumüller-Bichl. The method of formal coding. In *Cryptography - Proc., Burg Feuerstein, 1992, LNCS 149*, pages 235–255. Springer Verlag, 1982.

[104] I. Schaumüller-Bichl. On the design and analysis of new cipher systems related to the DES. Technical report, Linz University, 1983.

[105] B. Schneier. *Applied Cryptography.* Wiley & Sons, 1994.

[106] R. Sedgewick, T.G. Szymanski, and A.C. Yao. The complexity of finding cycles in periodic functions. *SIAM Journal of Computing*, 11:376–390, 1982.

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

[108] M.E. Smid and D.K. Branstad. The Data Encryption Standard: Past and future. In G.J. Simmons, editor, *Contemporary Cryptology - The Science of Information Integrity*, chapter 1, pages 43–64. IEEE Press, 1992.

[109] A. Sorkin. LUCIFER: a cryptographic algorithm. *Cryptologia*, 8(1):22–35, 1984.

[110] W. Tuchman. Hellman presents no shortcut solutions to DES. *IEEE Spectrum*, 16(7):40–41, July 1979.

[111] P.C. van Oorschot and M.J. Wiener. A known-plaintext attack on two-key triple encryption. In I.B. Damgård, editor, *Advances in Cryptology - Proc. Eurocrypt'90, LNCS 473*, pages 318–325. Springer Verlag, 1990.

[112] M.J. Wiener. Efficient DES key search. Technical Report TR-244, School of Computer Science, Carleton University, Ottawa, Canada, May 1994. Presented at the Rump Session of Crypto'93.

[113] Y. Zheng. *Principles for Designing Secure Block Ciphers and One-way Hash Functions.* PhD thesis, Yokohama National University, Japan, 1990.

[114] Y. Zheng, T. Matsumoto, and H. Imai. On the construction of block ciphers provably secure and not relying on any unproved hypothesis. In G. Brassard, editor, *Advances in Cryptology - Proc, Crypto'89, LNCS 435*, pages 461–480. Springer Verlag, 1990.

[115] Y. Zheng, T. Matsumoto, and H. Imai. Duality between two cryptographic primitives. In *Proc. 8th International Conference on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, LNCS 508*, pages 379–390. Springer Verlag, 1991.

Index