Chapter 4

Data Encryption Standard (DES)

General Notes:

- DES is by far the most popular private-key algorithm.
- It was published in 1975 and standardized in 1977.
- Expired in 1998.

4.1 Encryption

System Parameters:

→ block cipher.

→ 64 input/output bits.

→ 56 bits of key.

Principle: 16 rounds of encryption.

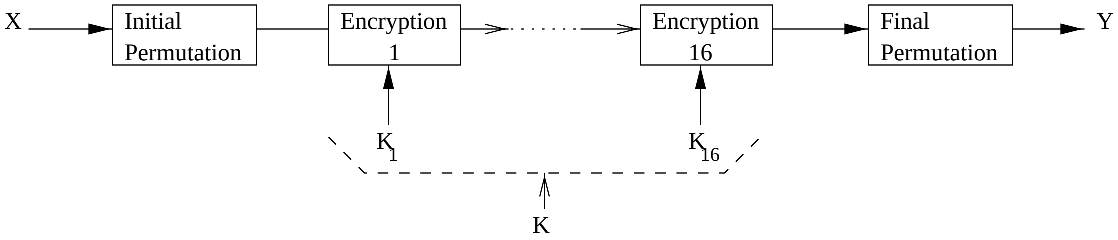


Figure 4.1: General Model of DES

4.1.1 Overview

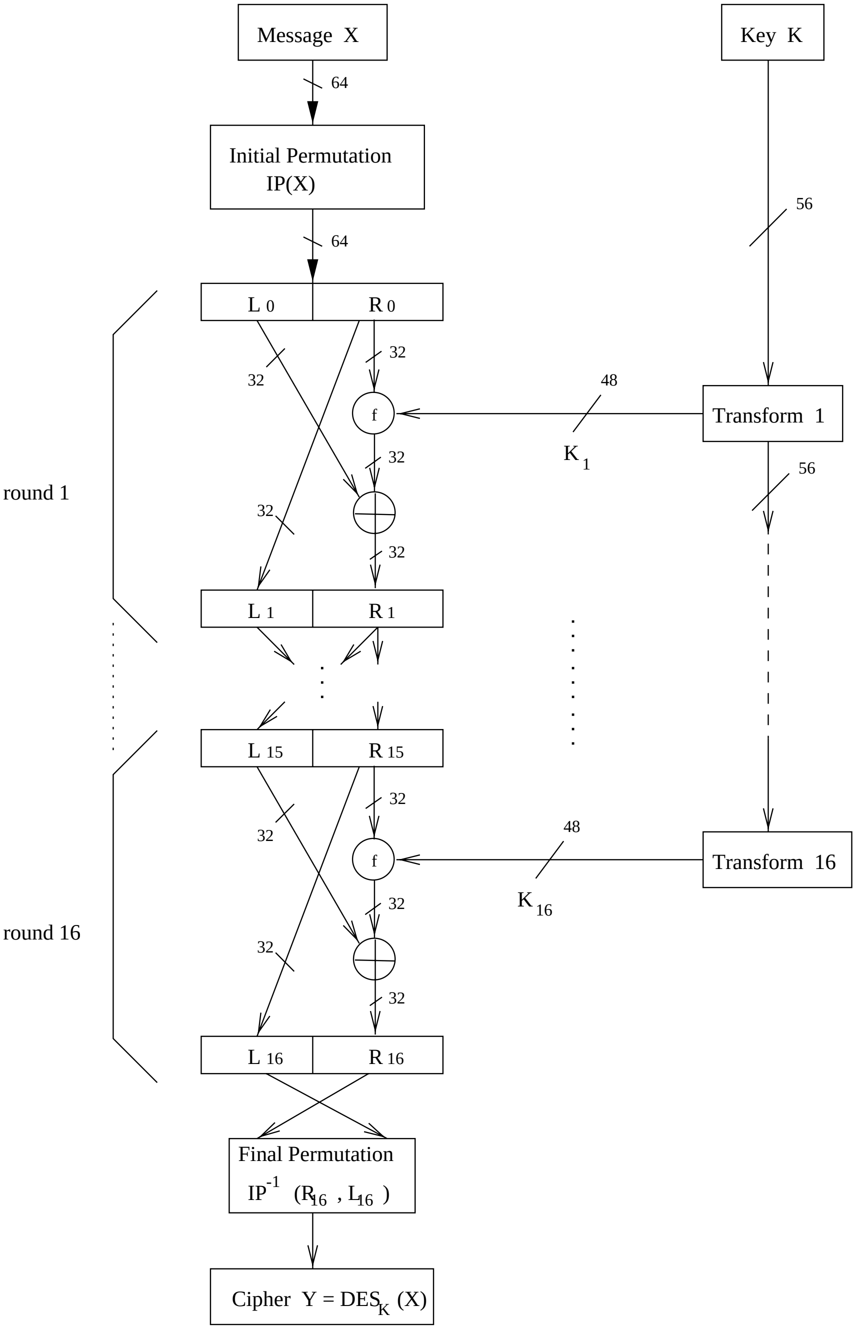


Figure 4.2: The Feistel Network

4.1.2 Permutations

a) Initial Permutation IP.

IP							
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

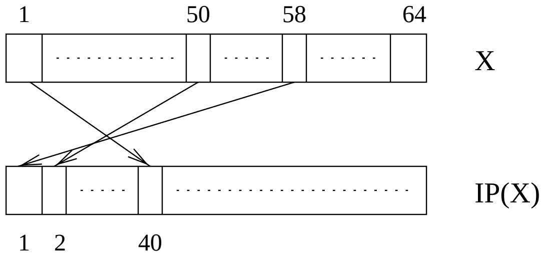


Figure 4.3: Initial permutation

b) Inverse Initial Permutation IP^{-1} (final permutation).

Note:

$IP^{-1}(IP(X)) = X.$

4.1.3 Core Iteration / f-Function

General Description:

$L_i = R_{i-1}.$

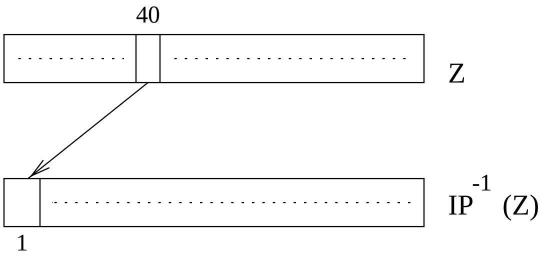


Figure 4.4: Final permutation

$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$.

The core iteration is the f-function that takes the right half of the output of the previous round and the key as input.

	E	bit	table		
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

S-boxes:

Contain look-up tables (LUTs) with 64 numbers ranging from $0 \ldots 15$.

Input: Six bit code selecting one number.

Output: Four bit binary representation of one number out of 64.

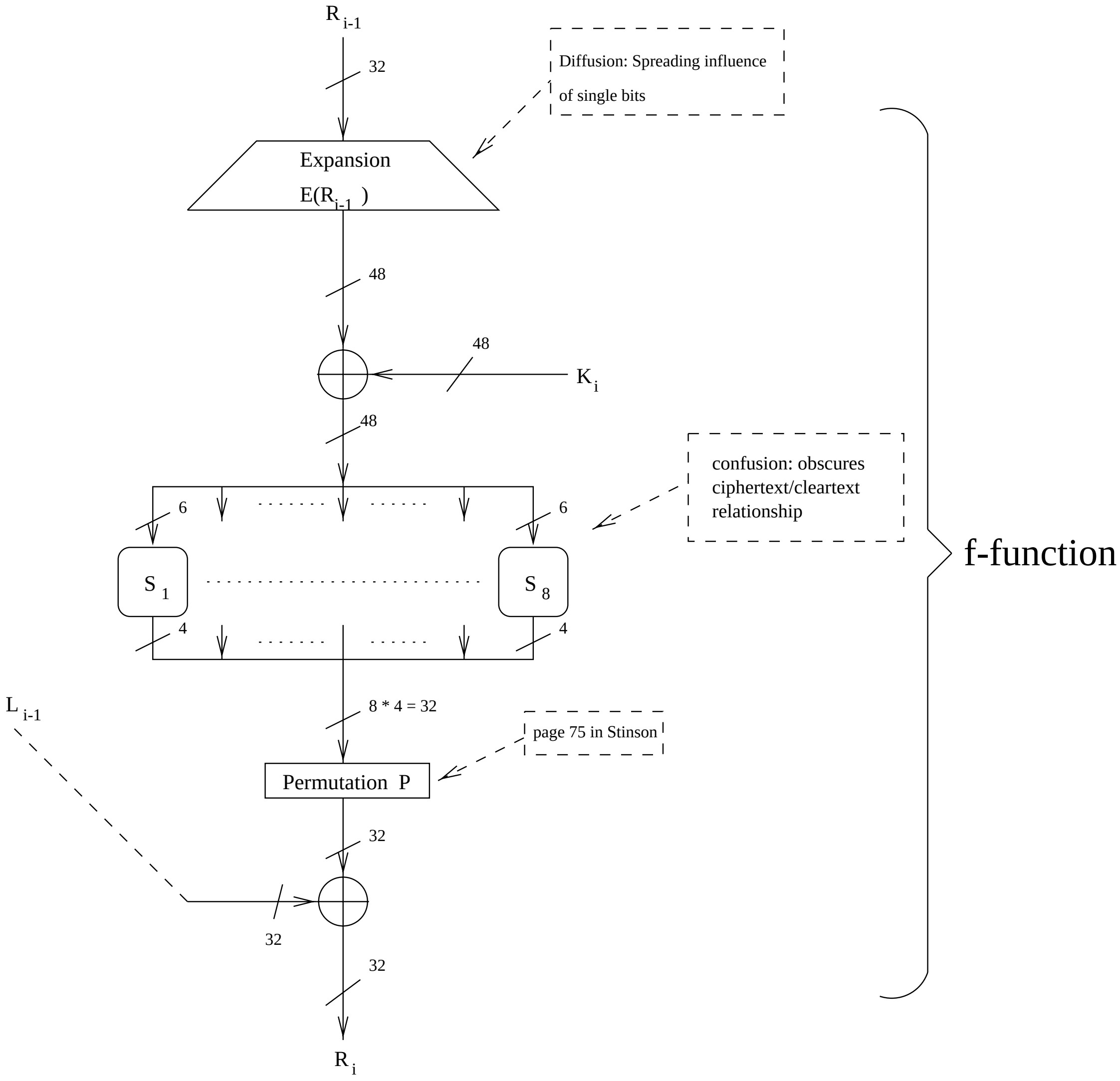


Figure 4.5: Core function of DES

Example:

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

S-Box 1

Input: Six bit vector with MSB and LSB selecting the row and four inner bits selecting column.

$b = (100101)$.

$\rightarrow$ row $= (11)_2 = 3$ (forth row).

$\rightarrow$ column $= (0010)_2 = 2$ (third column).

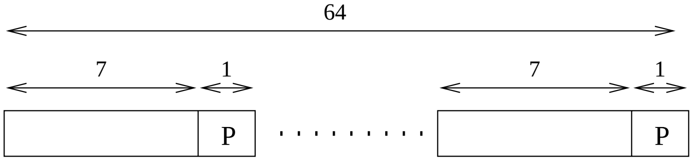
$S_1(37 = 100101_2) = 8 = 1000_2$.

Remark:

S-boxes are the most crucial elements of DES because they introduce a **non-linear** function to the algorithm, i.e., $S(a)$ XOR $S(b) \neq S(a$ XOR $b)$.

4.1.4 Key Schedule

Note:



P = parity bits

Figure 4.6: 64 bit DES block

In practice the DES key is artificially enlarged with odd parity bits. These bits are "stripped" in PC-1.

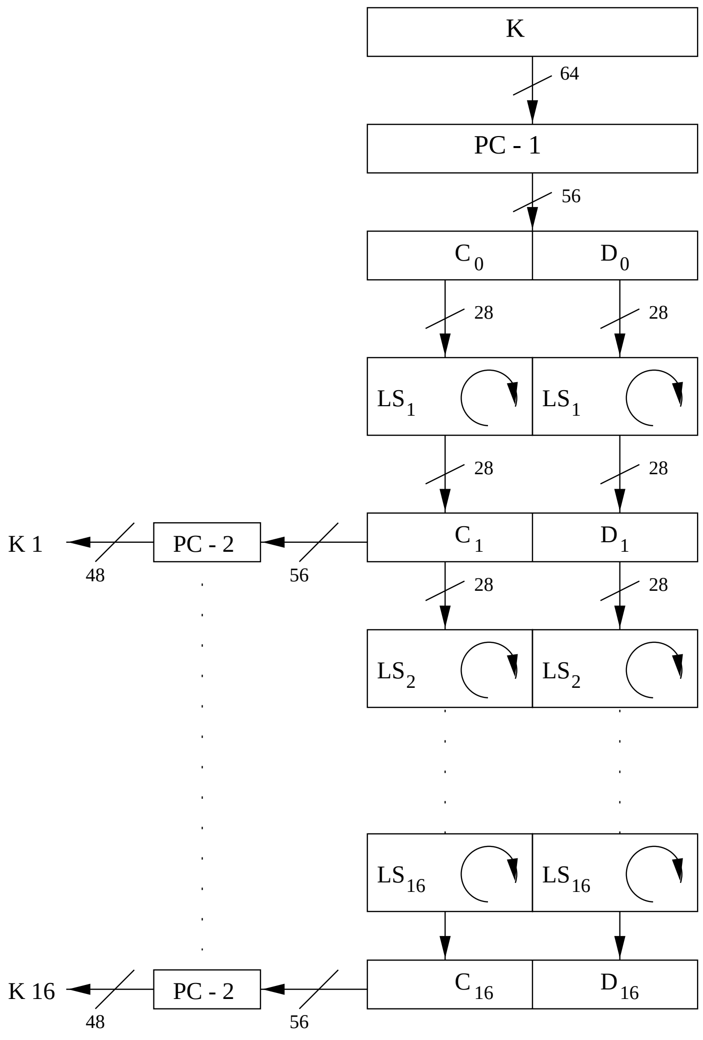


Figure 4.7: DES key scheduler

The cyclic Left-Shift (LS) blocks have two modes of operation:

a) for LS_i where $i = 1, 2, 9, 16$, the block is shifted once.

b) for LS_i where $i \neq 1, 2, 9, 16$, the block is shifted twice.

Remark:

The total number of cyclic Left-Shifts is $4 \cdot 1 + 12 \cdot 2 = 28$. As a results of this $C_0 = C_{16}$ and $D_0 = D_{16}$.

4.2 Decryption

One advantage of DES is that decryption is essentially the same as encryption. Only the key schedule is reversed. This is due to the fact that DES is based on a Feistel network.

Question: Why does decryption work essentially the same as encryption?

a) Find what happens in the initial stage of decryption!

$(L_0^d, R_0^d) = IP(Y) = IP(IP^{-1}(R_{16}, L_{16})) = (R_{16}, L_{16})$.

$(L_0^d, R_0^d) = IP(Y) = (R_{16}, L_{16})$.

$L_0^d = R_{16}$.

$R_0^d = L_{16} = R_{15}$.

b) Find what happens in the iterations!

What are (L_1^d, R_1^d) ?

$L_1^d = R_0^d = L_{16} = R_{15}$.

substitute into the above equation to get:

$R_1^d = L_0^d \oplus f(R_0^d, k_{16}) = R_{16} \oplus f(L_{16}, k_{16})$.

$R_1^d = [L_{15} \oplus f(R_{15}, k_{16})] \oplus f(R_{15}, k_{16})$.

$R_1^d = L_{15} \oplus [f(R_{15}, k_{16}) \oplus f(R_{15}, k_{16})] = L_{15}$.

in general: $L_i^d = R_{16-i}$ and $R_i^d = L_{16-i}$;

such that: $L_{16}^d = R_{16-16} = R_0$ and $R_{16}^d = R_0$.

c) Find what happens in the final stage!

$IP^{-1}(R_{16}^d, L_{16}^d) = IP^{-1}(L_0, R_0) \doteq IP^{-1}(IP(X)) = X \;\; q.e.d.$

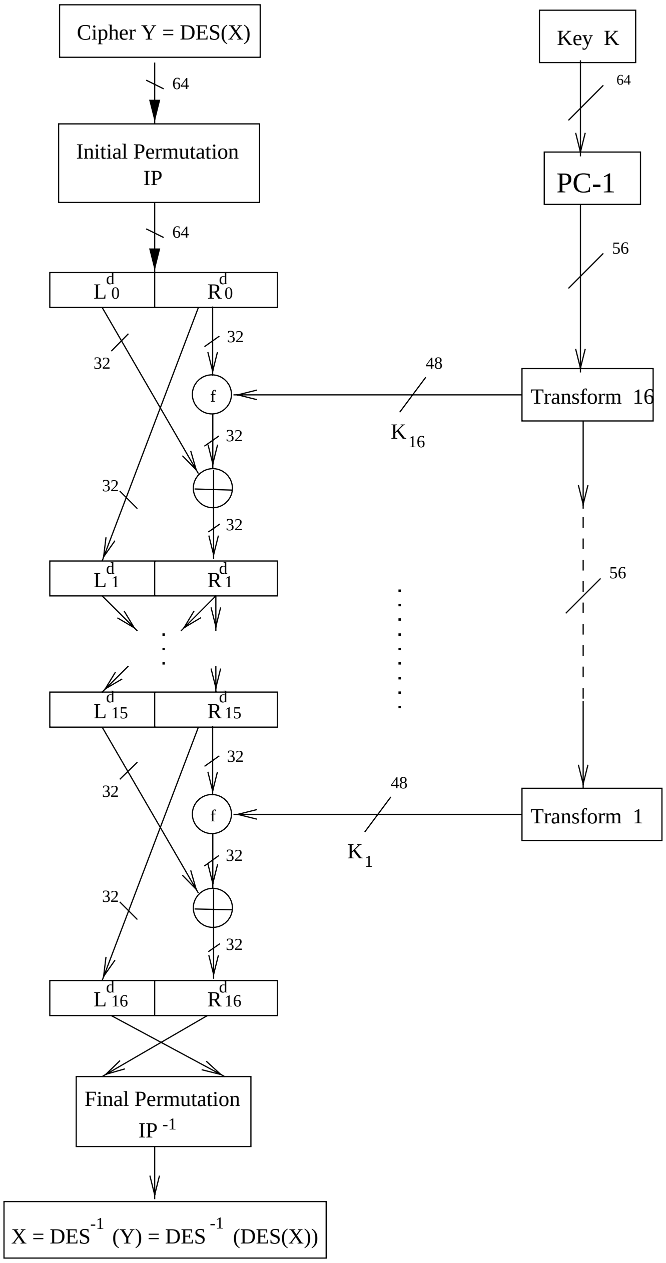


Figure 4.8: Decryption of DES

Reversed Key Schedule:

Question: Given K, how can we easily generate k_{16}?

$k_{16} = PC2(C_{16}, D_{16}) = PC2(C_0, D_0) = PC2(PC1(k))$.

$k_{15} = PC2(C_{15}, D_{15}) = PC2(RS_1(C_{16}), RS_1(D_{16})) = PC2(RS_1(C_0), RS_1(D_0))$.

4.3 Implementation

Note:

One design criteria for DES was fast hardware implementation.

4.3.1 Hardware

Since permutations and simple table look-ups are fast in hardware, DES can be implemented very efficiently [AM97, page 362].

Fastest Implementation:

⇒ 9 Gbit/s as 0.6 μm technology ASIC [WPR$^+$99] with 16 stage pipeline.

4.3.2 Software

Record: 130 Mbits/s by Biham [Bih97].

Typically: a few 10 Mbit/s.

4.4 Attacks

There have been two major points of criticism about DES from the beginning:

i) key size is too small,

ii) the S-boxes contained secret design criteria.

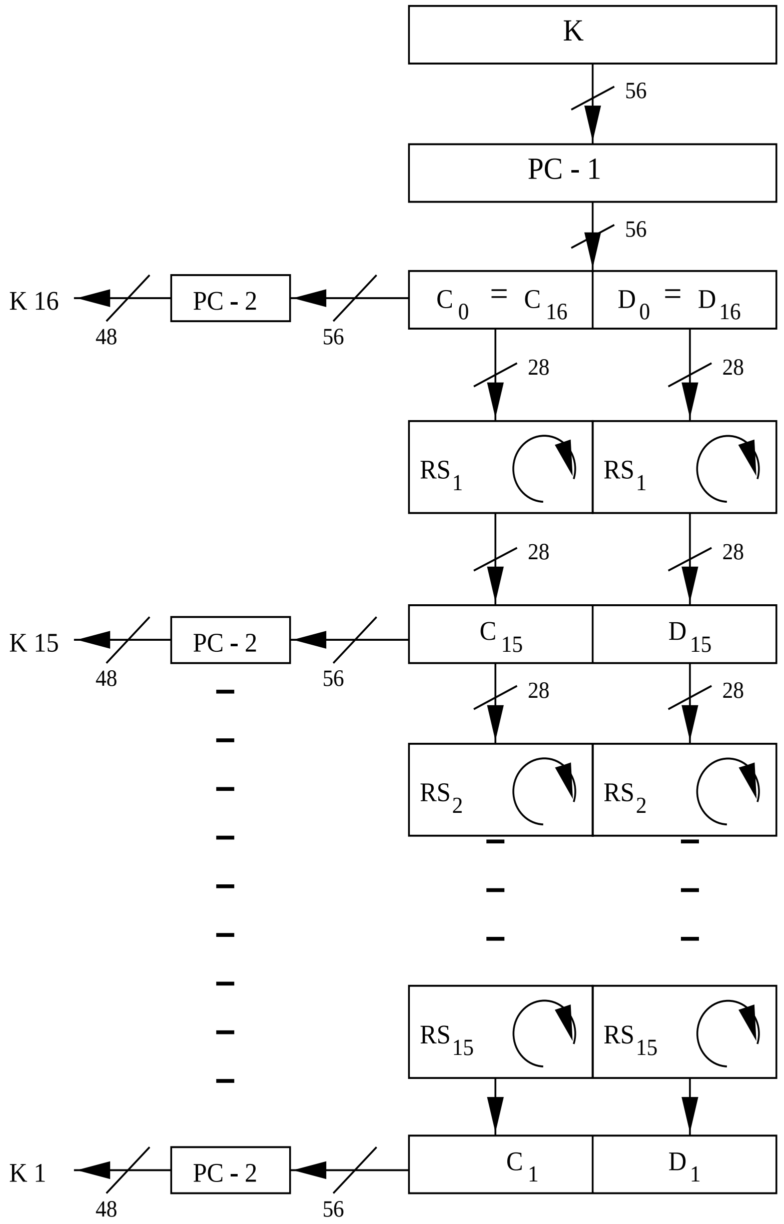


Figure 4.9: Reversed key scheduler for decryption of DES

4.4.1 Exhaustive Key Search

Known Plaintext Attack:

known: X and Y.

unknown: K, such that $Y = DES_k(X)$.

idea: test all 2^{56} possible keys → $DES_{k_i}(X) \overset{?}{=} Y$; $i = 0, 1, \ldots, 2^{56} - 1$.

4.4.2 Differential Cryptanalysis

Proposed by Biham/Shamir in 1990.

Principle:

To consider differences in plain and ciphertext pairs and deduce the likelihood of certain keys.

16-round DES requirements:

With chosen plaintext, 2^{47} (X,Y) pairs are needed.

With known plaintext, 2^{55} (X,Y) pairs are needed.

2^{37} arithmetic operations are needed.

Since each (X,Y) pair is 128 bits long, large storage is needed which makes this attack highly impractical!

Remark: The DES S-boxes are optimized against differential cryptanalysis.

4.4.3 Linear Cryptanalysis

Proposed by Matsui in 1993 and presented at CRYPTO'94.

Principal:

To consider differences in plain and ciphertext pairs and deduce the likelihood of certain key bits.

The actual attack was implemented:

→ with 2^{43} known plaintexts, the key was recovered in 50 days.

→ using 12 HP RISC workstations running at 99MHz.

Remark: The S-box design of DES is not optimized for this attack.

Date	Proposed/implemented attack
1977	Diffie & Hellman, estimate cost of key search machine (underestimate)
1990	Biham & Shamir propose differential cryptoanalysis (2^{47} chosen ciphertexts)
1993	Mike Wiener proposes detailed hardware design for key search machine: average search time of 36 h @ $100,000
1993	Matsui proposes linear cryptoanalysis (2^{43} chosen ciphertexts)
Jun. 1997	DES Challenge I broken, distributed effort took 4.5 months
Feb. 1998	DES Challenge II–1 broken, distributed effort took 39 days
Jul. 1998	DES Challenge II–2 broken, key-search machine built by the Electronic Frontier Foundation (EFF), 1800 ASICs, each with 24 search units, $250K, 15 days average (actual time 56 hours)
Jan. 1999	DES Challenge III broken, distributed effort combined with EFF's key-search machine, it took 22 hours and 15 minutes.

Table 4.1: History of full-round DES attacks

4.5 DES Alternatives

There exists a wealth of other block ciphers. A small collection of as of yet unbroken ciphers is:

Algorithm	*Year*	*Inventor*	*X/Y bits*	*Key*	*Core Operation*
AES	2000+	?	128	128/192/256	?
Triple DES			64	112	S-box
IDEA	90/92	Lai/Massey	64	128	modulo arithmetic
Cast	93	Adams/Tavares	64	64	variable S-boxes
Safer	94	Massey	64	64/128	modulo arithmetic

For further reading, consult Chapters 13 and 14 in [Sch93].