

# Cryptanalysis

## Block Ciphers 1

John Manferdelli  
JohnManferdelli@hotmail.com

© 2004-2010, John L. Manferdelli.

*This material is provided without warranty of any kind including, without limitation, warranty of non-infringement or suitability for any purpose. This material is not guaranteed to be error free and is intended for instructional use only*

# Block ciphers

- Complicated keyed invertible functions constructed from iterated elementary rounds.
  - Confusion: non-linear functions (ROM lookup)
  - Diffusion: permute round output bits

## Characteristics:

- Fast
- Data encrypted in fixed “block sizes” (64,128,256 bit blocks are common).
- Key and message bits non-linearly mixed in cipher-text

# Mathematical view of block ciphers

- $E(k, x) = y$ .
- $E: GF(2^m) \times GF(2^n) \rightarrow GF(2^n)$ , often  $m=n$ .
- $E(k, x)$  is a bijection in second variable.
- $E(k, x)$  in  $S_N$ ,  $N = 2^n$ .
- Each bit position is a balanced boolean function.
- $E$  is easy to compute but inverse function (with  $k$  fixed) is hard to compute without knowledge of  $k$ .
- Implicit function hard to compute.
- Intersection of algebraic varieties.

# A (very bad) block cipher

- Let  $M$  be an invertible  $n \times n$  matrix over  $\text{GF}(2)$ .
- Suppose  $k$  is an  $n$ -bit vector representing the key and  $p$  is an  $n$  bit vector representing the plaintext block
- Put  $c = M(p+k)$ .  $C$  is the ciphertext
- Example:

- $$M = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad k = (1,1,1)^T, \quad p = (1,0,1)^T, \quad c = (1,1,0)^T.$$

- Why is this so bad?
- Better (but still bad)
- Let  $R(k)$  be a rule that selects an invertible matrix from  $\text{GF}(2)^n \times \text{GF}(2)^n$ . Put  $c = R(k)p$ .
- Lesson: linear is bad

# Guiding Theorems

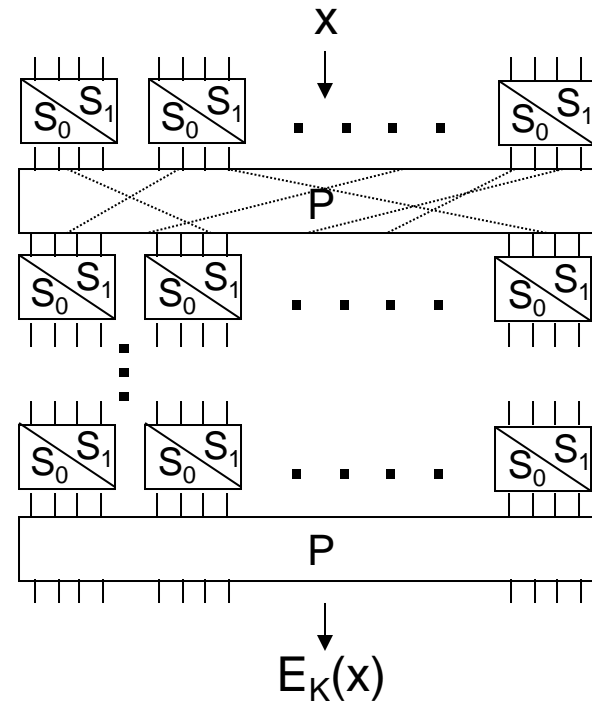
- Implicit Function Theorem: If  $f(x,y) = c$ , is a continuously differentiable function from  $F^n \times F^m$  into  $F^m$  and the  $m \times m$  Jacobian in the  $y$  variables is non-zero in a region, there is a function  $g$  from  $R^n$  to  $R^m$  such that  $F(x, g(x)) = c$ . When  $F$  is linear, this function is very easy to compute. Think of  $g$  as mapping the plaintext to the key (for fixed ciphertext).
- Functions in over finite fields are polynomials: If  $f$  is a function from  $k^n$  to  $k$ , where  $k$  is a finite field,  $f$  can be written as a polynomial in the  $n$  variables.
- Reduction in dimension: Generally (pathological exceptions aside), if  $f$  is a function from  $k^n$  to  $k$ , where  $k$  is a finite field, and  $f(x) = c$ , one variable can be written as a function of the other  $n-1$  variables. In other words, if  $g$  is a function from  $k^n$  to  $k$  subject to the constraint  $f(x) = c$ , then  $g$  can be rewritten as a function of  $n-1$  variables.

# Data Encryption Standard

- Federal History
  - 1972 study.
  - RFP: 5/73, 8/74.
  - NSA: S-Box influence, key size reduction.
  - Published in Federal Register: 3/75.
  - FIPS 46: January, 1976.
- DES
  - Descendant of Feistel's Lucifer.
  - Designers: Horst Feistel, Walter Tuchman, Don Coppersmith, Alan Konheim, Edna Grossman, Bill Notz, Lynn Smith, and Bryant Tuckerman.
- Brute Force Cracking
  - Key size controversy: USG wanted 48 bit keys, IBM wanted 64 bit keys. Result: 56-bit keys.
  - EFS DES Cracker: \$250K, 1998. 1,536 custom chips. Can brute force a DES key in days.
  - Deep Crack and distributed net break a DES key in 22.25 hours (dated)

# Horst Feistel: Lucifer

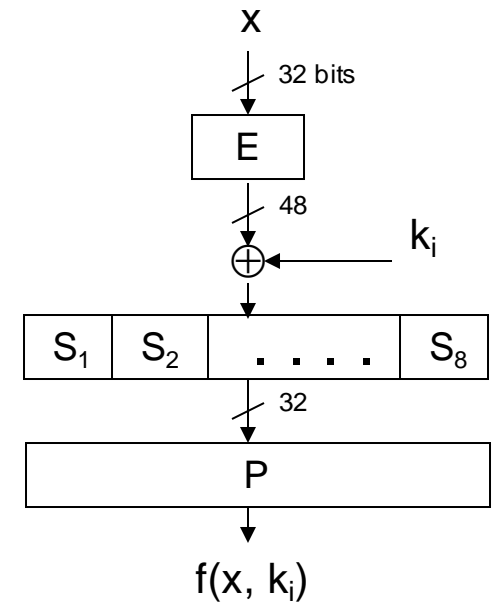
- First serious needs for civilian encryption (in electronic banking), 1970's
- IBM's response: Lucifer, an iterated SP cipher
- Lucifer (v0):
  - Two fixed, 4x4 s-boxes,  $S_0$  &  $S_1$
  - A fixed permutation  $P$
  - Key bits determine which s-box is to be used at each position
  - $8 \times 64/4 = 128$  key bits (for 64-bit block, 8 rounds)



Graphic by cschen@cc.nctu.edu.tw

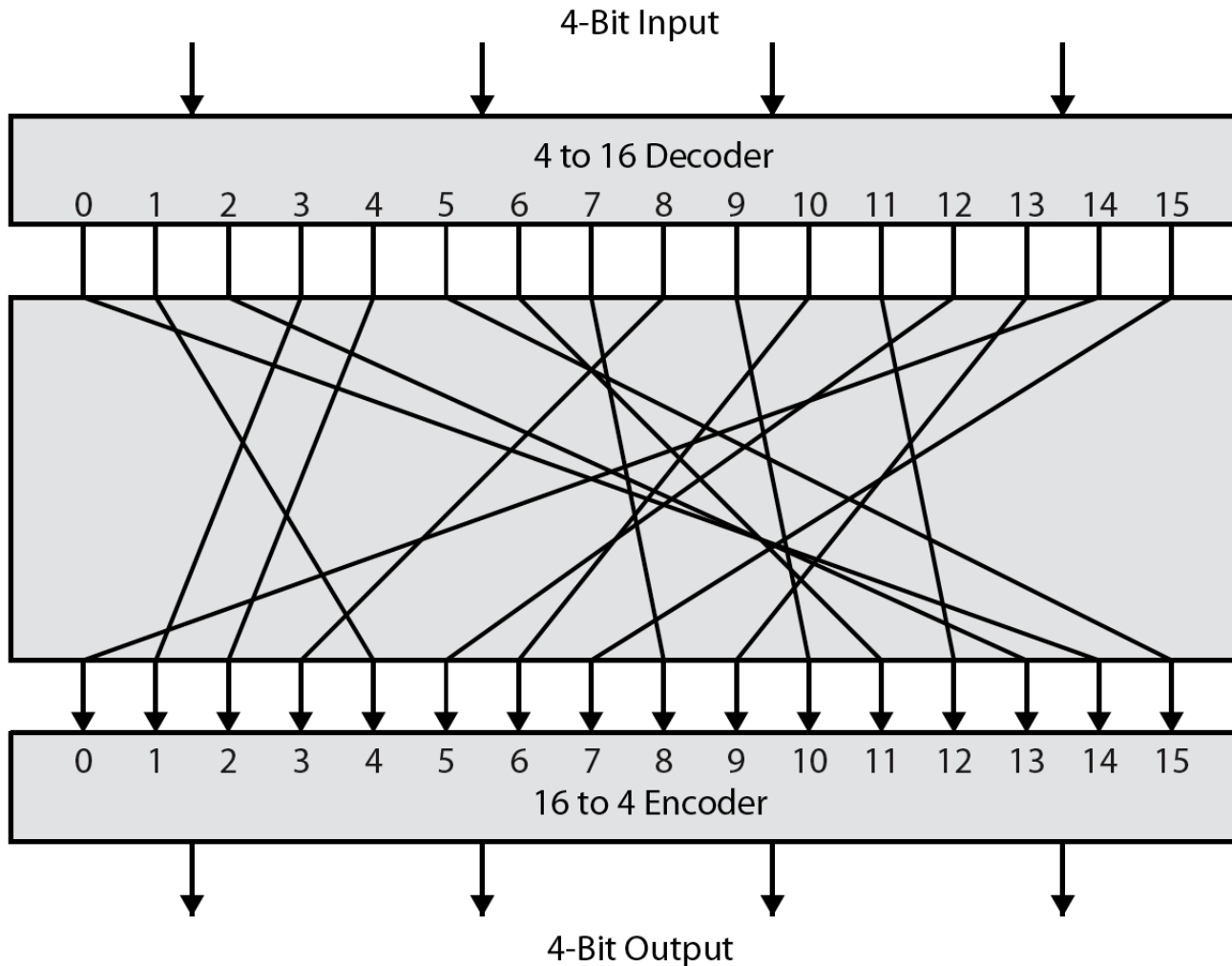
# From Lucifer to DES

- 8 fixed, 6x4 s-boxes (non-invertible)
- Expansion, E, (simple duplication of 16 bits)
- Round keys are used only for xor with the input
- 56-bit key size
- 16 x 48 round key bits are selected from the 56-bit master key by the “key schedule”.



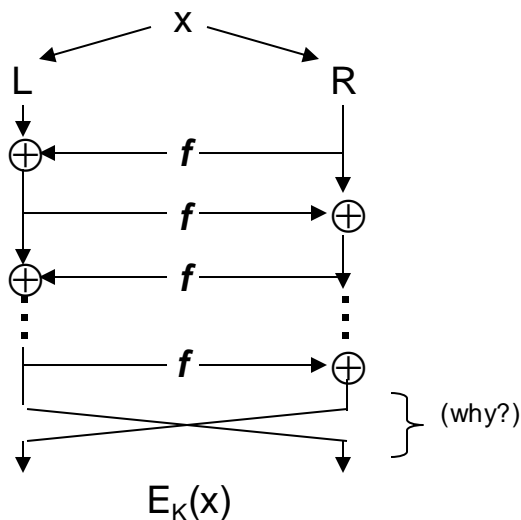


# What is a “safe” block cipher

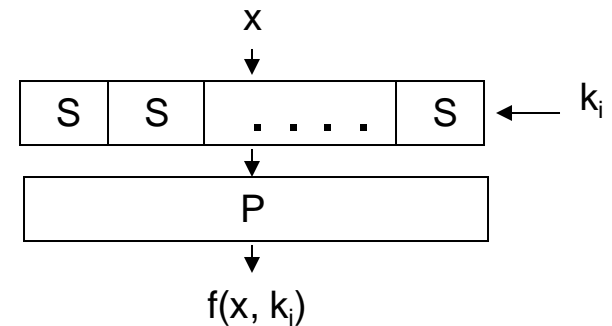


# Feistel Ciphers

- A straightforward SP cipher needs twice the hardware: one for encryption ( $S, P$ ), one for decryption ( $S^{-1}, P^{-1}$ ).
- Feistel's solution:

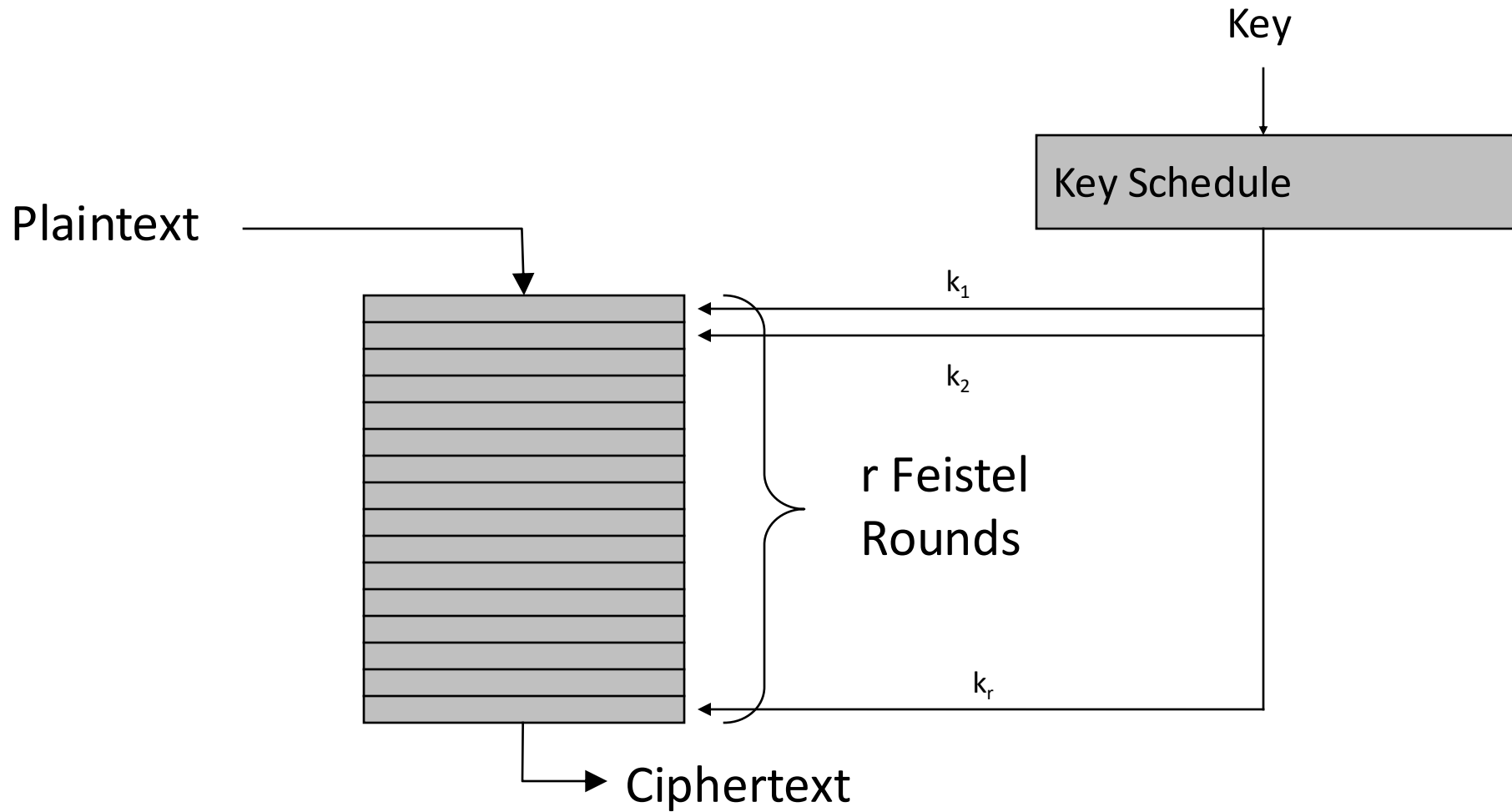


where the  
 $f$  function  
is SP:

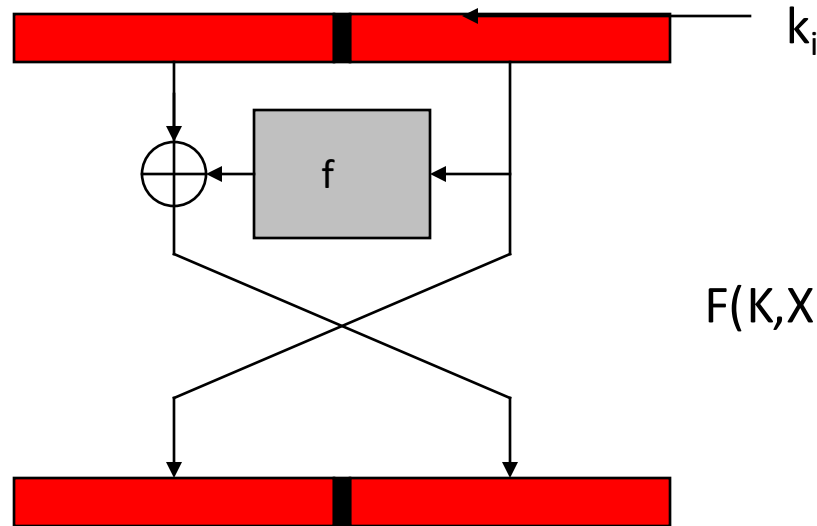


- Lucifer v1: Feistel SP cipher; 64-bit block, 128-bit key, 16 rounds.

# Iterated Feistel Cipher



# Feistel Round

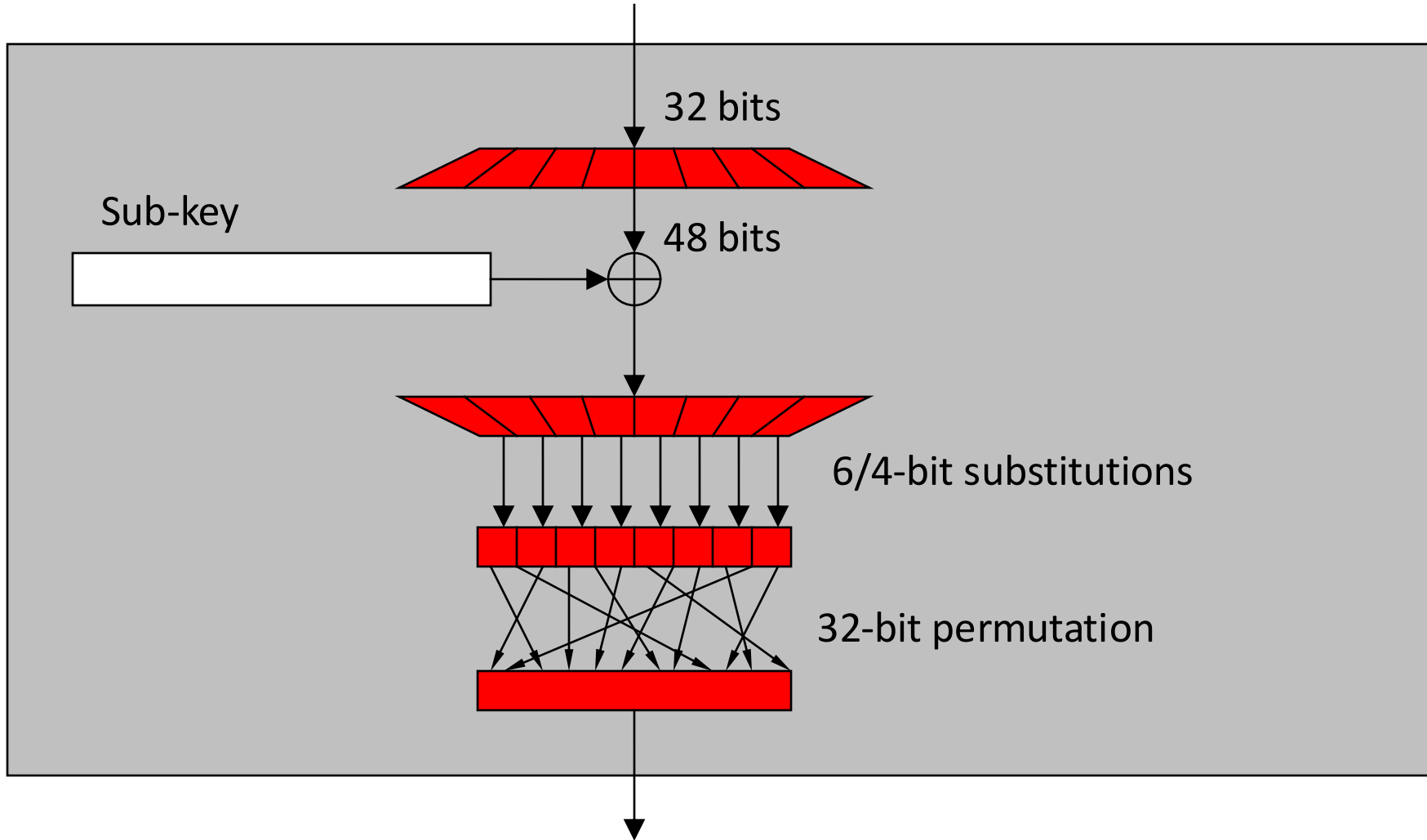


Graphic courtesy of Josh Benaloh

$F(K,X)$  = non-linear function

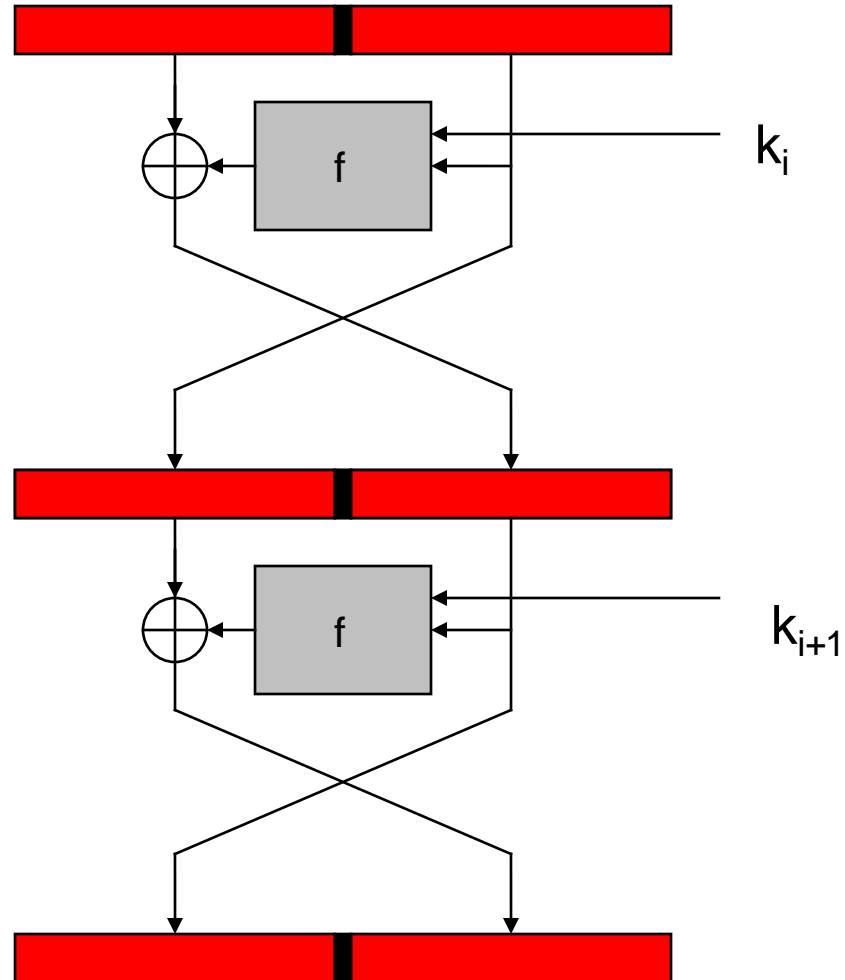
Note: If  $\sigma_i(L,R) = (L \oplus f(E(R) \oplus k_i), R)$  and  $\tau(L, R) = (R, L)$ , this round is  $\tau \sigma_i(L, R)$ .  
To invert: swap halves and apply same transform with same key:  
 $\sigma_i \tau \tau \sigma_i(L,R) = (L,R)$ .

# DES Round Function

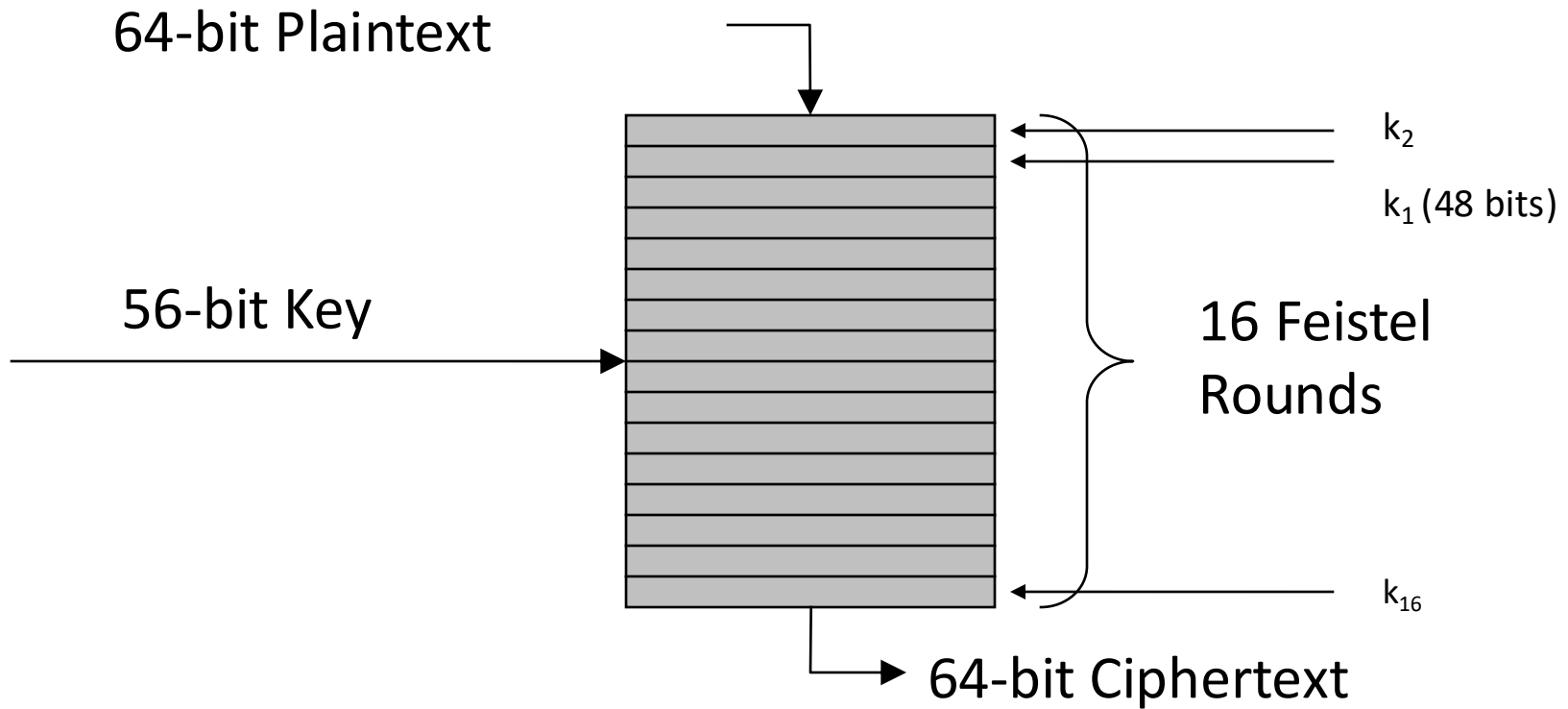


Slide courtesy of Josh Benaloh

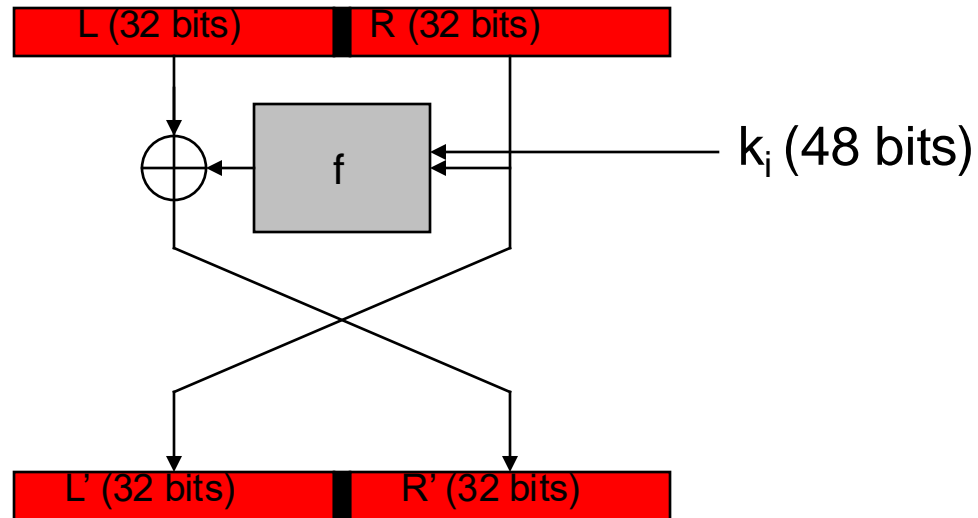
# Chaining Feistel Rounds



# DES



# DES Round



$F(K,X)$  = non-linear  
function



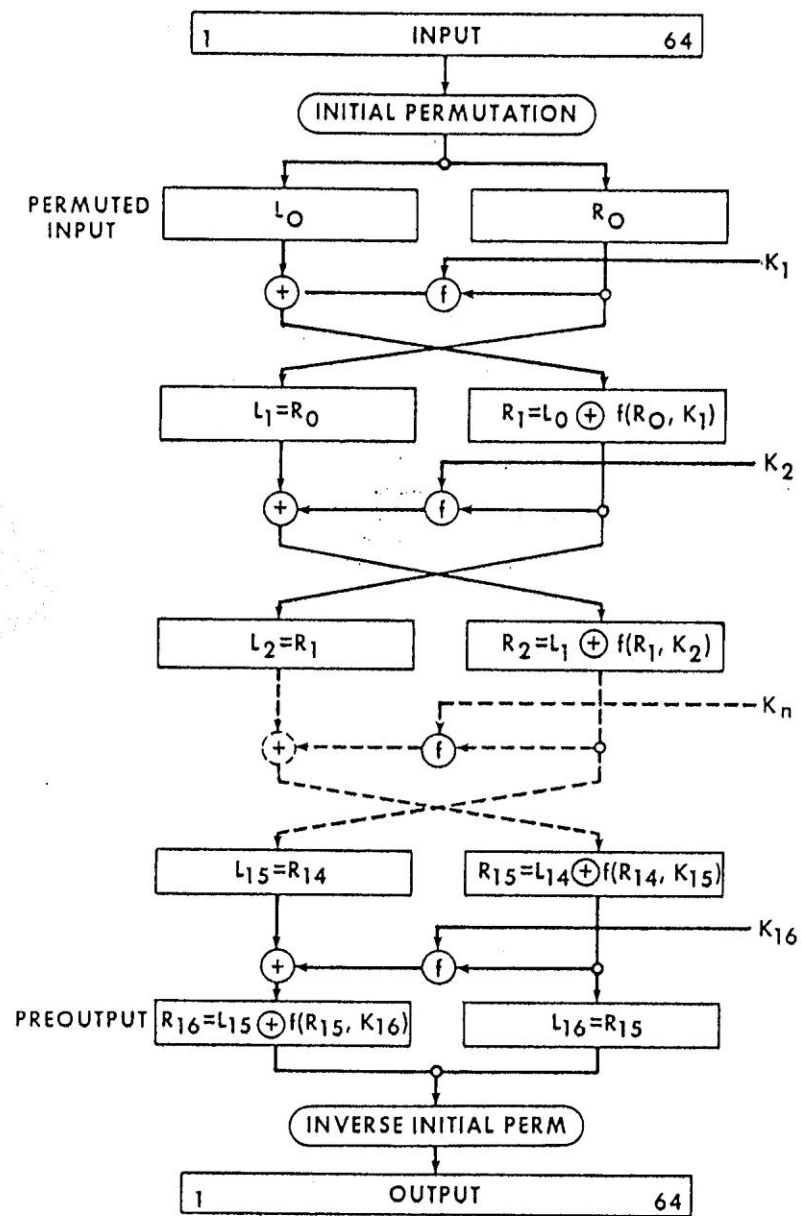
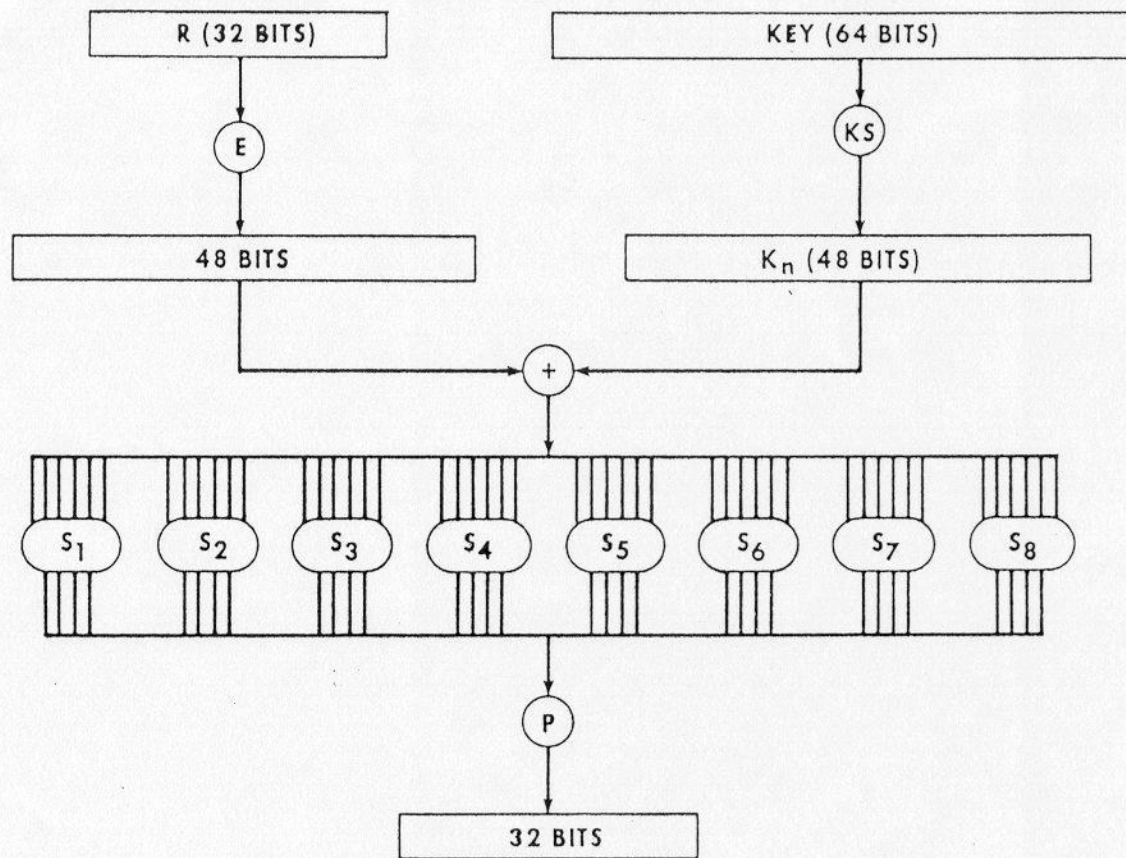


Figure 5.1. Electronic Codebook (ECB) Mode—Enciphering Computation.



$K_n$  CHANGES FOR  $N=1, 2, \dots, 16$

E=E FUNCTION  
KS=KEY SCHEDULE

Figure 5.2. Electronic Codebook (ECB) Mode—Calculation of  $f(R,K)$ .

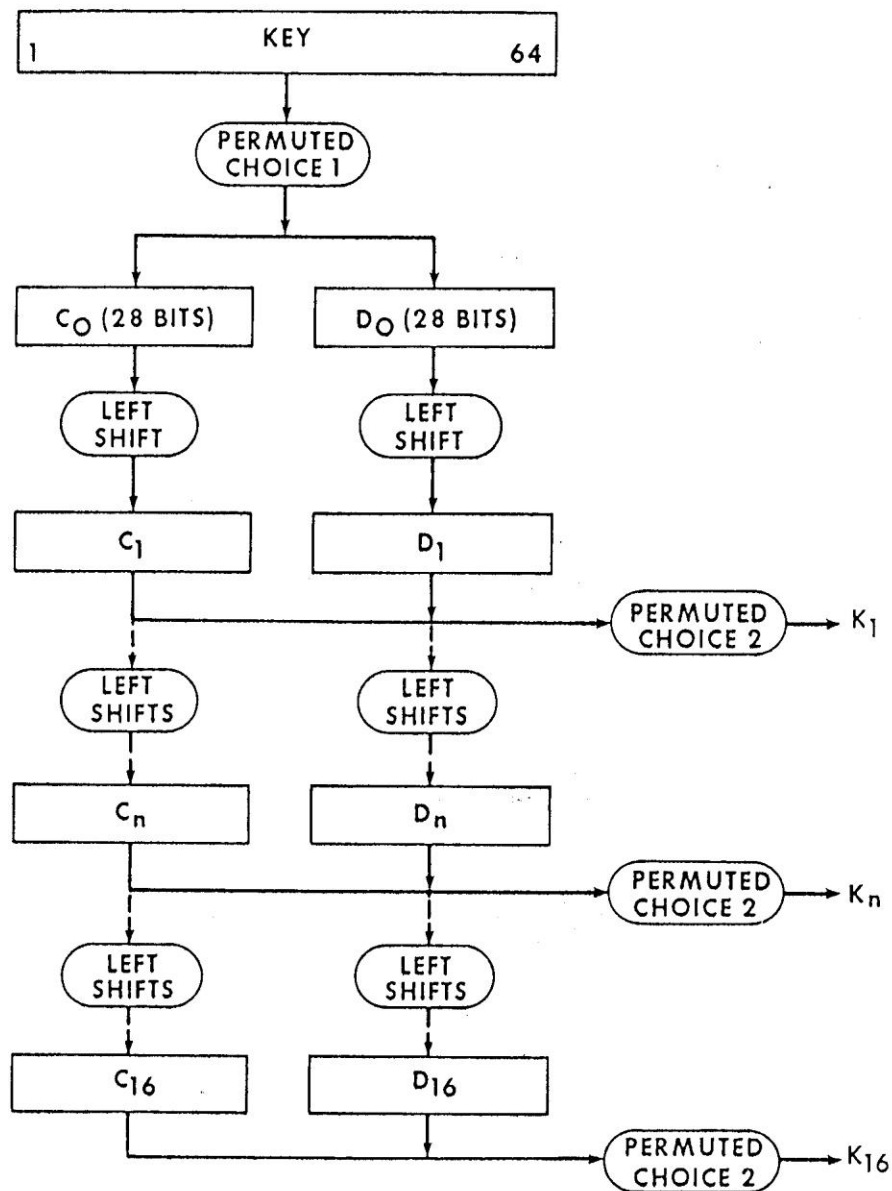


Figure 5.3. Electronic Codebook (ECB) Mode—Key Schedule (KS) Calculation.

# DES Described Algebraically

- $\sigma_i(L, R) = (L \oplus f(E(R) \oplus k_i), R)$ 
  - $k_i$  is 48 bit sub-key for round  $i$ .
  - $f(x) = P(S_1 S_2 S_3 \dots S_8(x))$ . Each  $S$  –box operates on 6-bit quantities and outputs 4 bit quantities.
  - $P$  permutes the resulting 32 output bits.
- $\tau(L, R) = (R, L)$ .
- Each round (except last) is  $\tau \sigma_i$ .
- Note that  $\tau \tau = \tau^2 = 1 = \sigma_i \sigma_i = \sigma_i^2$ .
- Full DES is:  $DES_K(x) = IP^{-1} \sigma_{16} \tau \dots \sigma_3 \tau \sigma_2 \tau \sigma_1 IP(x)$ .
- So, its inverse is:  $DES_K^{-1}(x) = IP^{-1} \sigma_1 \tau \dots \sigma_{14} \tau \sigma_{15} \tau \sigma_{16} IP(x)$ .

# DES Key Schedule

$$C_0 D_0 = PC_1(K)$$

$$C_{i+1} = \text{LeftShift}(\text{Shift}_i, C_i), D_{i+1} = \text{LeftShift}(\text{Shift}_i, D_i)$$

$$K_i = PC_2(C_i \parallel D_i)$$

$$\text{Shift}_i = \langle 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1 \rangle$$

- Note: Irregular Key schedule protects against related key attacks. [Biham, New Types of Cryptanalytic Attacks using Related Keys, TR-753, Technion]

# DES Key Schedule

pc1[64]

57	49	41	33	25	17	09	01	58	50	42	34	26	18	10	02
59	51	43	35	27	19	11	03	60	52	44	36	63	55	47	39
31	23	15	07	62	54	46	38	30	22	14	06	61	53	45	37
29	21	13	05	28	20	12	04	00	00	00	00	00	00	00	00

pc2[48]

14	17	11	24	01	05	03	28	15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02	41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56	34	53	46	42	50	36	29	32

# DES Key Schedule

## Key schedule round 1

10	51	34	60	49	17	33	57	2	9	19	42
3	35	26	25	44	58	59	1	36	27	18	41
22	28	39	54	37	4	47	30	5	53	23	29
61	21	38	63	15	20	45	14	13	62	55	31

## Key schedule round 2

2	43	26	52	41	9	25	49	59	1	11	34
60	27	18	17	36	50	51	58	57	19	10	33
14	20	31	46	29	63	39	22	28	45	15	21
53	13	30	55	7	12	37	6	5	54	47	23

# DES Data

S1 (hex)

e	4	d	1	2	f	b	8	3	a	6	c	5	9	0	7
0	f	7	4	e	2	d	1	a	6	c	b	9	5	3	8
4	1	e	8	d	6	2	b	f	c	9	7	3	a	5	0
f	c	8	2	4	9	1	7	5	b	3	e	a	0	6	d

S2 (hex)

f	1	8	e	6	b	3	4	9	7	2	d	c	0	5	a
3	d	4	7	f	2	8	e	c	0	1	a	6	9	b	5
0	e	7	b	a	4	d	1	5	8	c	6	9	3	2	f
d	8	a	1	3	f	4	2	b	6	7	c	0	5	e	9

S3 (hex)

a	0	9	e	6	3	f	5	1	d	c	7	b	4	2	8
d	7	0	9	3	4	6	a	2	8	5	e	c	b	f	1
d	6	4	9	8	f	3	0	b	1	2	c	5	a	e	7
1	a	d	0	6	9	8	7	4	f	e	3	b	5	2	c



# DES Data

S4 (hex)

7	d	e	3	0	6	9	a	1	2	8	5	b	c	4	f
d	8	b	5	6	f	0	3	4	7	2	c	1	a	e	9
a	6	9	0	c	b	7	d	f	1	3	e	5	2	8	4
3	f	0	6	a	1	d	8	9	4	5	b	c	7	2	e

S5 (hex)

2	c	4	1	7	a	b	6	8	5	3	f	d	0	e	9
e	b	2	c	4	7	d	1	5	0	f	a	3	9	8	6
4	2	1	b	a	d	7	8	f	9	c	5	6	3	0	e
b	8	c	7	1	e	2	d	6	f	0	9	a	4	5	3

S6 (hex)

c	1	a	f	9	2	6	8	0	d	3	4	e	7	5	b
a	f	4	2	7	c	9	5	6	1	d	e	0	b	3	8
9	e	f	5	2	8	c	3	7	0	4	a	1	d	b	6
4	3	2	c	9	5	f	a	b	e	1	7	6	0	8	d

# DES Data

S7 (hex)

```

4 b 2 e f 0 8 d 3 c 9 7 5 a 6 1
d 0 b 7 4 9 1 a e 3 5 c 2 f 8 6
1 4 b d c 3 7 e a f 6 8 0 5 9 2
6 b d 8 1 4 a 7 9 5 0 f e 2 3 c
    
```

S8 (hex)

```

d 2 8 4 6 f b 1 a 9 3 e 5 0 c 7
1 f d 8 a 3 7 4 c 5 6 b 0 e 9 2
7 b 4 1 9 c e 2 0 6 a d f 3 5 8
2 1 e 7 4 a 8 d f c 9 0 3 5 6 b
    
```

E

```

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
    
```

- Note: DES can be made more secure against linear attacks by changing the order of the S-Boxes: Matsui, On Correlation between the order of S-Boxes and the Strength of DES. Eurocrypt, 94.

# DES Data

P															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

- Note on applying permutations:** For permutations of bit positions, like P above, the table entries consisting of two rows, the top row of which is “in order” means the following. If t is above b, the bit at b is moved into position t in the permuted bit string. For example, after applying P, above, the first bit of the output string was at position 16 of the input string.

# Another cipher for the era: TEA

```
tea(unsigned K[4], unsigned& L, unsigned& R) {  
    unsigned d = 0x9e3779b9;  
    unsigned s = 0;  
    for(int i = 0; i < 32; i++) {  
        s += d;  
        L += ((R<<4)+K[0])^(R+s)^((R>>5)+K[1]);  
        R += ((L<<4)+K[2])^(L+s)^((L>>5)+K[3]);  
    }  
}
```

# S Boxes as Polynomials over GF(2)

1, 1:

56+4+35+2+26+25+246+245+236+2356+16+15+156+14+146+145+13+1  
35+134+1346+1345+13456+125+1256+1245+123+12356+1234+12346

1, 2:

C+6+5+4+45+456+36+35+34+346+26+25+24+246+2456+23+236+235+2  
34+2346+1+15+156+134+13456+12+126+1256+124+1246+1245+12456  
+123+1236+1235+12356+1234+12346

1, 3:

C+6+56+46+45+3+35+356+346+3456+2+26+24+246+245+236+16+15+1  
45+13+1356+134+13456+12+126+125+12456+123+1236+1235+12356+  
1234+12346

1, 4:

C+6+5+456+3+34+346+345+2+23+234+1+15+14+146+135+134+1346+1  
345+1256+124+1246+1245+123+12356+1234+12346

Legend: C+6+56+46 means  $1 \oplus x_6 \oplus x_5 x_6 \oplus x_4 x_6$

# Decomposable Systems

- $E_{k_1 || k_2}(x) = E'_{k_1}(x) || E''_{k_2}(x)$

m	t	$2^{mt}$	$m2^t$
2	32	$2^{64}$	$2^{33}$
4	16	$2^{64}$	$2^{18}$

- Good mixing and avalanche condition

# Feistel Ciphers defeat simple attacks

- After 4 rounds get flat statistics.
- Parallel system attack
- Even a weak round function can yield a strong Feistel cipher if iterated sufficiently.
  - Provided it's non-linear

# DES Attacks: Exhaustive Search

- Symmetry  $\text{DES}(\mathbf{k} \oplus \mathbf{1}, \mathbf{x} \oplus \mathbf{1}) = \text{DES}(\mathbf{k}, \mathbf{x}) \oplus \mathbf{1}$
- Suppose we know plain/cipher text pair  $(p, c)$

```
for (k=0; k<256; k++) {  
    if (DES(k, p) == c) {  
        printf("Key is %x\n", k);  
        break;  
    }  
}
```

- Expected number of trials (if  $k$  was chosen at random) before success:  $2^{55}$



# DES Attacks: Poor key hygiene

- Poor random number generator: 20 bits of entropy
  - $2^{20}$  vs  $2^{56}$
  - Second biggest real problem
  - First biggest: bad key management
- Symmetric ciphers are said to be secure in practice if no known attack works more efficiently than exhaustive search.
  - Note that the barrier is computational not information theoretic.

# Suppose you decide the keyspace is too small?

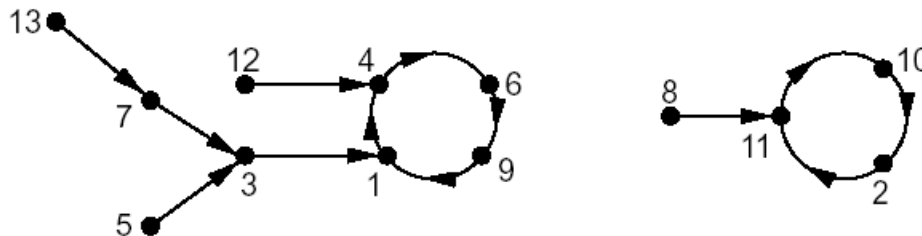
- Can you increase security by encrypting twice or more?
  - $E'(k_1 || k_2, x) = E(k_1, E(k_2, x))$
- Answer: Maybe.
- Three times is the charm (triple DES).
- If you do it twice, TMTO attack reduces it to little more than one key search time (if you have a lot of memory).

# What's the complexity of breaking a Block Cipher

- Suppose there are  $K$  keys ( $K=2^{56}$  for DES)
- Pick a plaintext  $p$  and sort the pairs  $(E(p,x), x)$  for  $x=0,1,\dots, K-1$
- Ask for  $E(p,k)=c$ .
- Lookup  $(c,x)$  in the table.
- $x$  is the key.
- $O(1)$  after precomputation!

# Random mappings

- Let  $F_n$  denote all functions (mappings) from a finite domain of size  $n$  to a finite co-domain of size  $n$
- Every mapping is equally likely to be chosen,  $|F_n| = n^n$  the probability of choosing a particular mapping is  $1/n^n$
- Example.  $f : \{1, 2, \dots, 13\} \rightarrow \{1, 2, \dots, 13\}$



Graphic by Maithili Narasimha

- As  $n$  tends to infinity, the following are expectations of some parameters associated with a random point in  $\{1, 2, \dots, n\}$  and a random function from  $F_n$ :
  - (i) tail length:  $\sqrt{\frac{\pi n}{8}}$
  - (ii) cycle length:  $\sqrt{\frac{\pi n}{8}}$
  - (iii) rho-length:  $\sqrt{\frac{\pi n}{2}}$

# Time memory trade off (“TMTO”)

- If we can pre-compute a table of  $(k, E_k(x))$  for a fixed  $x$ , then given corresponding  $(x,c)$  we can find the key in  $O(1)$  time.
- Trying random keys takes  $O(N)$  time (where  $N$ , usually,  $2^k$ , is the number of possible keys)
- Can we balance “memory” and “time” resources?
- It is not a 50-50 proposition. Hellman showed we could cut the search time to  $O(N^{1/2})$  by pre-computing and storing  $O(N^{1/2})$  values.

# Chain of Encryptions

- Assume block length  $n$  and key length  $k$  are equal:  $n = k$
- Construct chain of encryptions:

$$SP = K_0$$

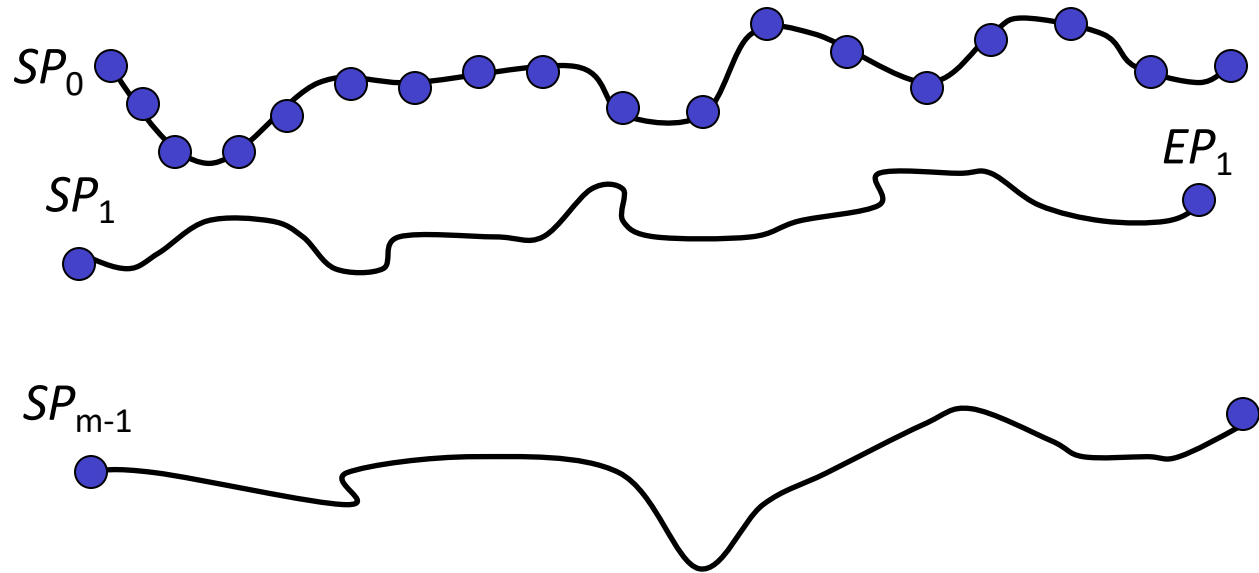
$$K_1 = E(P, SP)$$

$$K_2 = E(P, K_1)$$

⋮

⋮

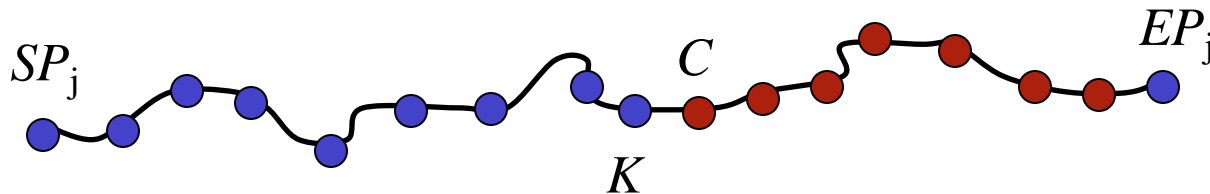
$$EP = K_t = E(P, K_{t-1})$$



- Pre-compute  $m$  encryption chains, each of length  $t + 1$
- Save only the start and end points

# TMTO Attack

- To attack a particular unknown key  $K$ 
  - For the same chosen  $P$  used to find chains, we know  $C$  where  $C = E(P, K)$  and  $K$  is unknown key
  - Compute the chain (maximum of  $t$  steps)
$$X_0 = C, X_1 = E(P, X_0), X_2 = E(P, X_1), \dots$$
- Suppose for some  $i$  we find  $X_i = EP_j$
- Since  $C = E(P, K)$  key  $K$  should lie before ciphertext  $C$  in chain!



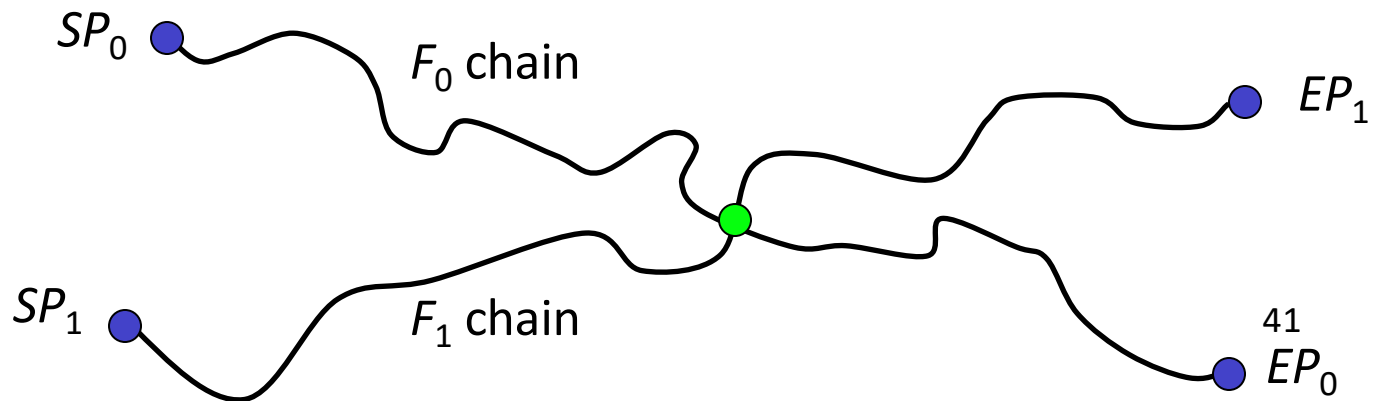
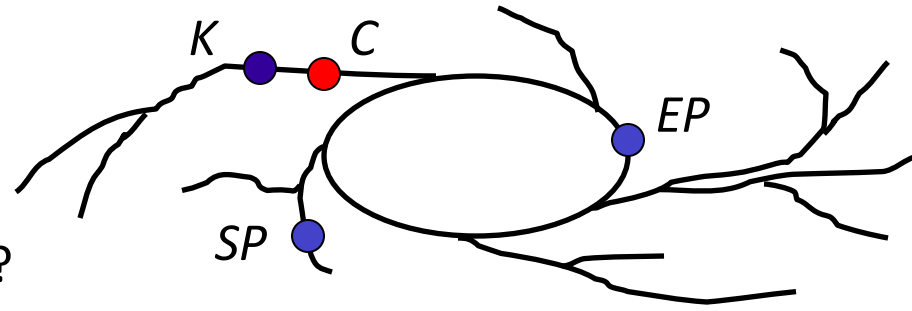
# DES TMTO

- Suppose block cipher has  $k = 56$
- Suppose we find  $m = 2^{28}$  chains each of length  $t = 2^{28}$  and no chains overlap (unrealistic)
- Memory:  $2^{28}$  pairs  $(SP_j, EP_j)$
- Time: about  $2^{28}$  (per attack)
  - Start at  $C$ , find some  $EP_j$  in about  $2^{27}$  steps
  - Find  $K$  with about  $2^{27}$  more steps
- Attack never fails!



# But things are a little more complicated

- Chains can cycle and merge
- False alarms, etc.
- What if block size not equal key length?
  - This is easy to deal with
- To reduce merging
  - Compute chain as  $F(E(P, K_{i-1}))$  where  $F$  permutes the bits
  - Chains computed using different functions can intersect, but they will **not** merge



Slide adapted from  
Mark Stamp

# TMTO in Practice

- Let
  - $m$  = random starting points for each  $F$  (# chains/table)
  - $t$  = Length of each chain
  - $r$  = number of “tables”, i.e., random functions
- Then  $mtr$  = total pre-computed chain elements
- Pre-computation is about  $mtr$  work
- Each TMTO attack requires
  - About  $mr$  “memory” and about  $tr$  “time”
- Choose  $m = t = r = 2^{k/3}$ ,  $mtr = 2^k$ .

# Success Probability

- Throw  $n$  balls into  $m$  urns
- What is expected number of urns that have at least one ball?
  - See Feller, *Intro. to Probability Theory*
- Why is this relevant to TMTO attack?
  - “Urns” correspond to keys
  - “Balls” correspond to constructing chains
- Assuming  $k$ -bit key and  $m, t, r$  defined as previously discussed
- Then, approximately,
 
$$P(\text{success}) = 1 - e^{-mtr/k}$$

$mtr$	$P(\text{success})$
0	0
$2^{k-5}$	0.03
$2^{k-4}$	0.06
$2^{k-3}$	0.12
$2^{k-2}$	0.22
$2^{k-1}$	0.39
$2^k$	0.63
$2^{k+1}$	0.86
$2^{k+2}$	0.98
$2^{k+3}$	0.99
$\infty$	1.00

# Group theory and DES

- What is the minimum length of a product of involutions from a fixed set required to generate  $S_n$ ?
- What does this have to do with the number of rounds in a cipher?
- How does this affect the increased security by “enciphering twice” with different keys?
- **Theorem** (Coppersmith and Grossman): If  $\sigma_K(L,R) = (L \oplus f(E(R) \oplus K), R)$ ,  $< \tau$ ,  $\sigma_K \geq A_N$ ,  $N = 2^n$ .
- **Note** (Netto): If  $a$  and  $b$  are chosen at random from  $S_n$  there is a good chance ( $\sim 3/4$ ) that  $\langle a, b \rangle = A_n$  or  $S_n$ .

# DES is not a group

- Set  $E_1(x) = \text{DES}_{0\text{x}\text{ffffffffffffffff}}(x)$ ,  $E_0(x) = \text{DES}_{0\text{x}0000000000000000}(x)$ .
- $F(x) = E_1(E_0(x))$ .
- There is an  $x$ :  $F^m(x) = x$ ,  $m \sim 2^{32}$ , a cycle length.
- If  $|F|=n$ ,  $m|n$ .
- Suppose DES is closed under composition so  $F = E_k = \text{DES}_k$ .
- $E_k^i = E_k^j$ ,  $E_k^{(j-i)} = I$ .  $0 \leq i < j \leq 2^{56}$ .
- Coppersmith found lengths of cycles for 33 plaintexts and the LCM of these cycle lengths  $> 2^{277}$ .

# If DES were a group...

- Suppose  $E_{K_1}(E_{K_2}(x)) = E_{K_3}(x)$ , that there are  $N$  possible keys, plaintexts and ciphertexts and that for a given plaintext-ciphertext pair there is only one possible key then there is a birthday attack that finds the key in  $O(N^{1/2})$ .
- Construct  $D_{K_1}(x)$  for  $O(N^{1/2})$  random keys,  $K_1$  and  $E_{K_2}(x)$  for  $O(N^{1/2})$  random keys,  $K_2$ . If there is a match,  $c = E_{K_1}(E_{K_2}(x))$ . This has the same effect as finding  $K_3$ .

# DES Key Schedule

- $C_0 D_0 = PC_1(K)$
- $C_{i+1} = \text{LeftShift}(\text{Shift}_i, C_i)$ ,  $D_{i+1} = \text{LeftShift}(\text{Shift}_i, D_i)$ .
- $K_i = PC_2(C_i \parallel D_i)$
- $\text{Shift}_i = \langle 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1 \rangle$
- Note: Irregular Key schedule protects against related key attacks. [Biham, New Types of Cryptanalytic Attacks using Related Keys, TR-753, Technion]

# Weak Keys

- DES has:
  - Four weak keys  $k$  for which  $E_k(E_k(m)) = m$ .
  - Twelve semi-weak keys which come in pairs  $k_1$  and  $k_2$  and are such that  $E_{k_1}(E_{k_2}(m)) = m$ .
  - Weak keys are due to “key schedule” algorithm



# How Weak Keys Arise

- A 28 bit quantity has potential symmetries of period 1, 2, 4, 7, and 14.
- Suppose each of  $C_0$  and  $D_0$  has a symmetry of period 1; for example,  $C_0 = 0x00000000$ ,  $D_0 = 0x11111111$ . We can easily figure out a master key ( $K$ ) that produces such a  $C_0$  and  $D_0$ .
- Then  $DES_K(DES_K(x)) = x$ .

# Interlude: Useful Math for Boolean Functions

- Algebraic Representations
- Linear Functions
- Affine approximations
- Bent Functions: functions furthest from linear
- Hadamard transforms
- MDS, linear codes, RS codes
- Random Functions
- Correlation and Correlation Immunity
- Some Notation:
  - Let  $L_1(P) \oplus L_2(C) = L_3(K) \oplus c$  with probability  $p_i$
  - $e_i = |1 - p_i|$  called the “bias”

# Boolean Functions

- The distance between two boolean functions  $f$  and  $g$  is  $d(f,g)=\#\{X \mid f(X) \neq g(X)\}$ .
- *Distance*: For Boolean function  $f(X)$  and  $g(X)$ ,  $d(f,D)=\min_{[g(X) \in D]} d(f,g)$
- *Affine function*:  $h(x)=a_1x_1 \oplus a_2x_2 \oplus \dots \oplus a_nx_n \oplus c$
- $nl(f)$  denotes the minimum distance between  $f(X)$  and the set of affine functions  $D_{\text{affine}}$ .  $nl(f)=d(f, D_{\text{affine}})$ ,  $D_{\text{affine}}=RM(1,n)$ .
- *Balance*:  $f(X)$  is balanced iff there is an equal number of 0's and 1's in the output of  $f(X)$ .
- *Algebraic normal form (ANF)*:
- *Degree*:  $\deg(f)$ , the highest degree term in ANF.
  - Example:  $f(X)=x_1+x_2$ ,  $\deg(f)=1$ ,  $g(X)=x_1x_2$ ,  $\deg(g)=2$
- **Lagrange Interpolation Theorem**: Every function in  $n$  variables can be expressed as a polynomial (hence ANF).
- Degree is not the best measure of nonlinearity.  
 $f(x_1, \dots, x_n)=x_1 \oplus \dots \oplus x_n \oplus x_1 \dots x_n$  has high degree but differs from a linear function at only 1 of  $2^n$  possible arguments.

# Example: polynomial representation

- If  $f$  is boolean function on  $n$  variables  $x_1, x_2, \dots, x_n$  and  $\mathbf{a}=(a_1, a_2, \dots, a_n)$  then  $f(x_1, x_2, \dots, x_n)=\sum_{\mathbf{a}} g(\mathbf{a}) x_1^{a_1} x_2^{a_2} \dots x_n^{a_n}$  where  $g(\mathbf{a}) = \sum_{\mathbf{b} < \mathbf{a}} f(b_1, b_2, \dots, b_n)$ . Here  $\mathbf{b} < \mathbf{a}$  means the binary representation of  $\mathbf{b}$  does not have a 1 unless there is a corresponding 1 in the representation of  $\mathbf{a}$ .

- $g(0,0,0)=f(0,0,0)=1$
- $g(0,1,0)=f(0,0,0)+f(0,1,0)=0$
- $g(1,0,0)=f(0,0,0)+f(1,0,0)=1$
- $g(1,1,0)=f(0,0,0)+f(1,0,0)+f(0,1,0)+f(1,1,0)=0$
- $g(0,0,1)=f(0,0,0)+f(0,0,1)=0$
- $g(0,1,1)=f(0,0,0)+f(0,0,1)+f(0,1,0)+f(0,1,1)=1$
- $g(0,0,1)=g(1,0,1)=g(0,1,1)=g(1,1,1)=0$
- $f(x_1, x_2, x_3)= 1+x_1+x_2 x_3$

$x_1$	$x_2$	$x_3$	$f(x_1, x_2, x_3)$
0	0	0	1
1	0	0	0
0	1	0	1
1	1	0	0
0	0	1	1
1	0	1	0
0	1	1	0
1	1	1	1

# Best affine approximation of $f_1$

- $f_1$

0000	0001	0010	0011	0100	0101	0110	0111
1	0	0	1	0	1	1	0
1000	1001	1010	1011	1100	1101	1110	1111
0	1	1	0	0	1	1	0

- $\mathcal{W}(f)(w) = F(w) = 2^{-n} \sum_x (-1)^{f(x) \oplus (w, x)}$

- As polynomial:  $1 + x_4 + x_3 + x_2 + x_1 + x_2 x_1$

- Spectrum:

0000	0001	0010	0011	0100	0101	0110	0111
0.00	0.00	0.00	0.50	0.00	0.00	0.00	-0.50
1000	1001	1010	1011	1100	1101	1110	1111
0.00	0.00	0.00	-0.50	0.00	0.00	0.00	-0.50

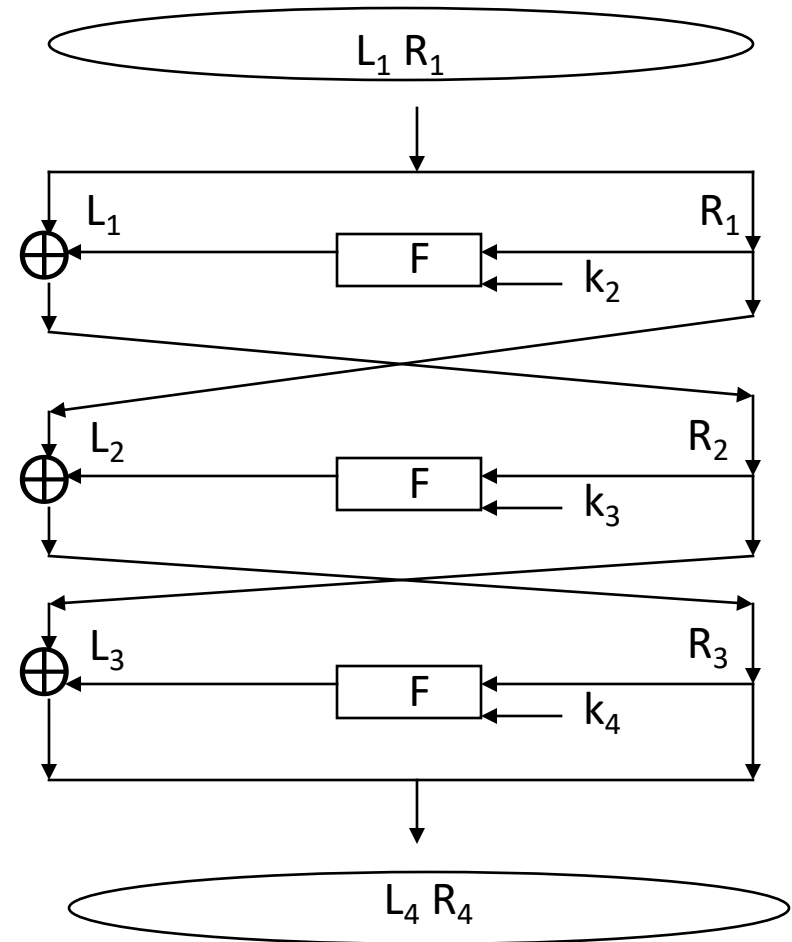
- $L(x) = x_3 + x_4$  is best linear approximation.  $\text{dist}(f_1, L(x)) = 8 (.5 + 1) = 12$ , so they disagree on  $16 - 12 = 4$  values

# Differential Characteristics

- Let  $P$  and  $P^*$  be inputs to a cipher and  $C$  and  $C^*$  be corresponding outputs with  $P \oplus P^* = P'$  and  $C \oplus C^* = C'$ .
- The notation  $P' \rightarrow C', p$  means the “input xor”,  $P'$  produces the “output xor”  $C'$  with probability  $p$ . Not all input/output xors are possible and the distribution is uneven. This can be used to find keys.  $P' \rightarrow C', p$  is called a *characteristic*.
- Notation:  $D_j(x', y') = \{u: S_j(u) \oplus S_j(u \oplus x') = y'\}$ .  $k_j \in x \oplus D_j(x', y')$
- For the characteristic  $0x34 \rightarrow d$  in S-box 1 from inputs  $1 \oplus 35 = 34$ ,  $D_1(34, d) = \{06, 10, 16, 1c, 22, 24, 28, 32\}$  and  $k_j \in \{7, 10, 17, 1d, 23, 25, 29, 33\} = 1 \oplus D_1(34, d)$

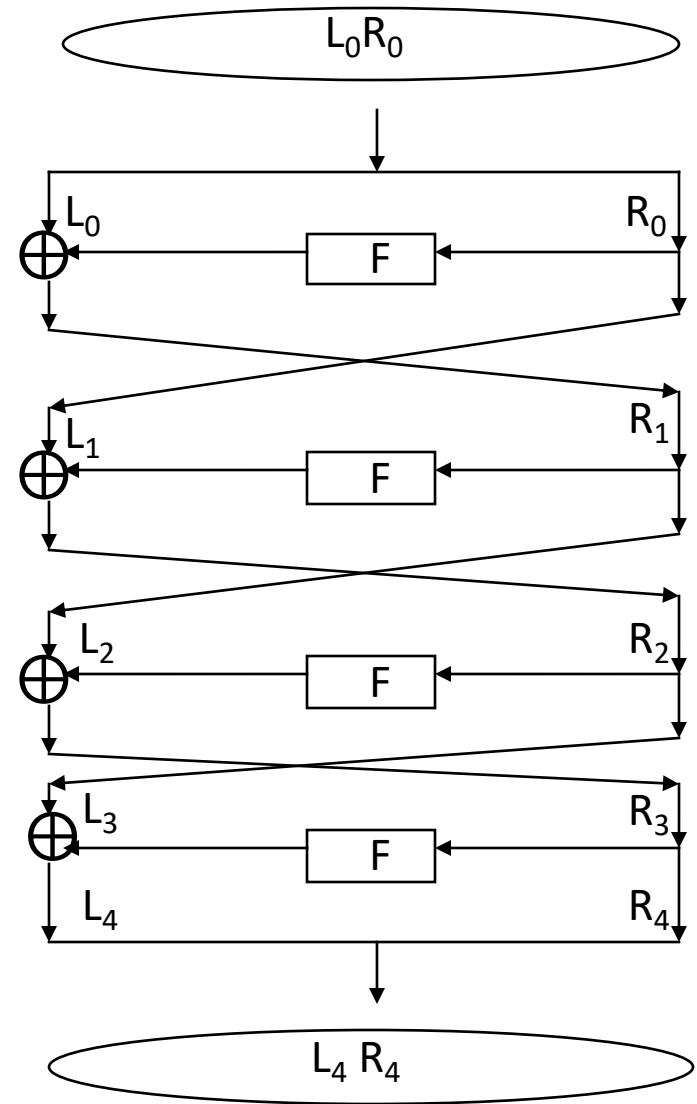
# Differential Cryptanalysis – 3 rounds

- $L_1 \oplus L_3 = f(k_2, R_1)$ . ..... (1)
- $L_4 \oplus L_3 = f(k_4, R_3)$ . ..... (2)
- $R_4 = R_3, L_2 = R_1, L_3 = R_2$ .
- $1 \& 2 \rightarrow L_4 \oplus L_1 = f(k_2, R_1) \oplus f(k_4, R_3)$ .
- $L_4 \oplus L_1 = f(k_2, R_1) \oplus f(k_4, R_3)$ . ..... (3)
- $L_4^* \oplus L_1^* = f(k_2, R_1^*) \oplus f(k_4, R_3^*)$ . .... (4)
- $3 \& 4 \rightarrow L_4' \oplus L_1' =$   
 $f(k_2, R_1^*) \oplus f(k_4, R_3^*) \oplus f(k_2, R_1^*) \oplus f(k_4, R_3^*)$ .
- $R_1 = R_1^* \rightarrow L_4' \oplus L_1' = f(k_4, R_3) \oplus f(k_4, R_3^*)$ .



# Simplified DES

- $L_{i+1} = R_i$ , each 6 bits.
- $R_{i+1} = L_i \oplus f(R_i, K_i)$
- $K$  is 9 bits.
- $E(x) = (x_1 \ x_2 \ x_4 \ x_3 \ x_4 \ x_3 \ x_5 \ x_6)$
- $S_1$ 
  - 101 010 001 110 011 100 111 000
  - 001 100 110 010 000 111 101 011
- $S_2$ 
  - 100 000 110 101 111 001 011 010
  - 101 011 000 111 110 010 001 100
- $K_i$  is 8 bits of  $K$  starting at  $i^{\text{th}}$  bit.





# Differential Cryptanalysis – 3 rounds

$$L_4 \oplus R_2 = f(R_3, k_4) \text{ and } L_1 \oplus R_2 = f(R_1, k_1)$$

$$\text{So, } L_1 \oplus L_4 = f(R_3, k_4) \oplus f(R_1, k_1)$$

$$\text{If, } R_1^* = R_1, L_1' \oplus L_4' = f(R_3, k_4) \oplus f(R_3^*, k_4)$$

$$L_1, R_1 : 000111 \ 011011$$

$$L_1^*, R_1^* : 101110 \ 011011$$

$$L_1', R_1' : 101001 \ 000000$$

$$L_4, R_4 : 100101 \ 000011$$

$$L_4^*, R_4^* : 011000 \ 100100$$

$$L_4', R_4' : 111101 \ 100111$$

$$E(R_4) : 0000 \ 0011$$

$$E(R_4') : 1010 \ 1011$$

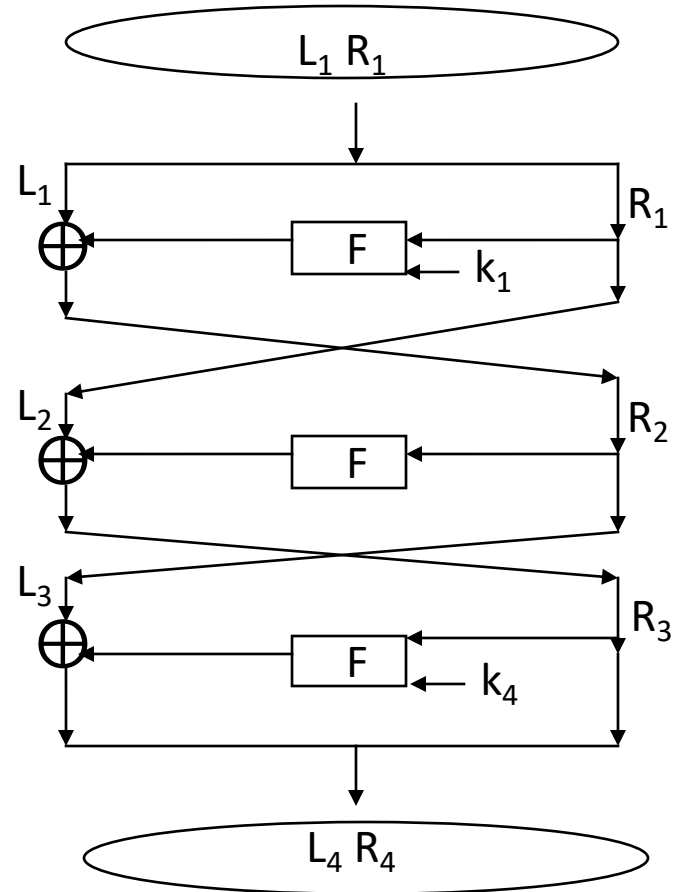
$$L_4' \oplus L_1' : 111 \ 101 \oplus 101 \ 001 = 010 \ 100.$$

$$S_1' : 1010 \rightarrow 010(1001, 0011).$$

$$S_2' : 1011 \rightarrow 100(1100, 0111).$$

$$(E(R_4) \oplus k_4)_{1..4} = 1001, 0011, \ k_4 = 1001, 0000.$$

$$(E(R_4) \oplus k_4)_{5..8} = 1100, 0111, \ k_4 = 1111, 0100.$$



# Differential Cryptanalysis, 4 rounds

Pick

$L_0', R_0' : 011010 \ 001100.$

Then

$E(R_0') : 0011 \ 1100.$

$0011 \rightarrow 011$  with  $p = \frac{3}{4}$

$1100 \rightarrow 010$  with  $p = \frac{1}{2}$

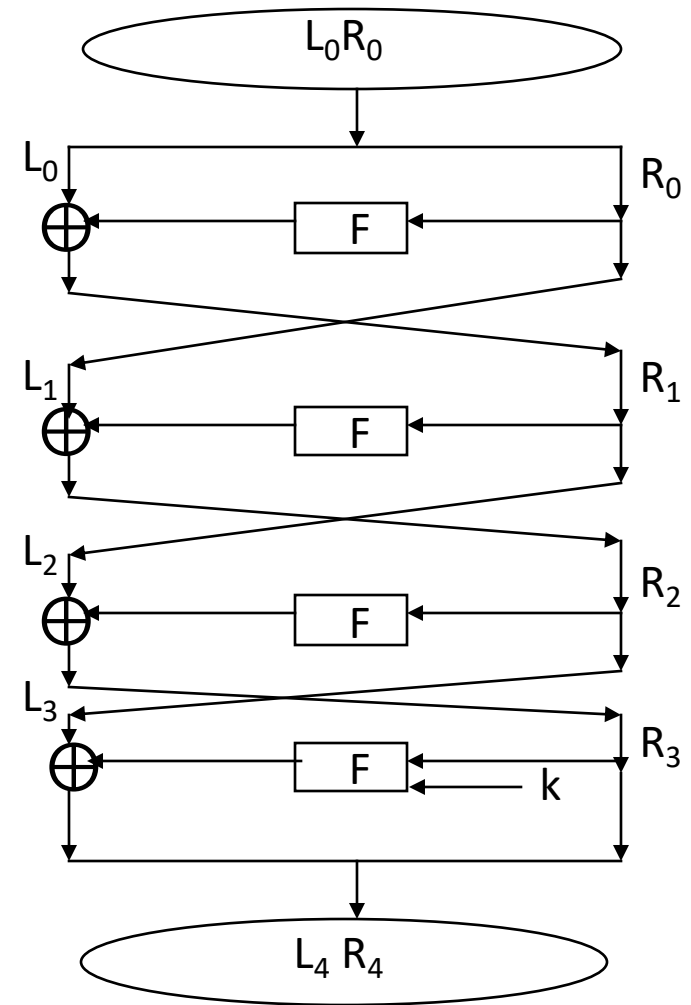
So

$L_1' = 011 \ 010$  with  $p = \frac{3}{8},$

and

$L_1', R_1' : 001100, \ 000000$  with  $p = \frac{3}{8}.$

- $R_4' \oplus L_1' = f(L_4, k) \oplus f(L_4^*, k)$
- $\frac{3}{8}$  of the pairs with the input differential  $(L_0', R_0'),$  produce a conforming  $(L_1', R_1').$   $\frac{5}{8}$  scatter  $(L_1', R_1')$  at random.



# Estimating cost of Differential Attack

- Given  $m$  pairs of text,  $p$  the probability of a right pair,  $k$  the number of keys,  $\gamma$  the number of suggested keys per right pair and  $\lambda$  the ratio of non-discarded pairs to the total number of pairs.
- Average count is  $\frac{\lambda \gamma m}{k}$
- $SN = \frac{mp}{\frac{\gamma \lambda m}{k}} = \frac{kp}{\gamma \lambda}$
- Right pairs are binomially distributed and for small  $p$  can be Poisson approximated by  $X \sim P(m, p)$

# Comments on Differential Cryptanalysis of DES

# Rounds	Needed pairs	Analyzed Pairs	Bits Found	# Char rounds	Char prob	S/N	Chosen Plain
4	$2^3$	$2^3$	42	1	1	16	$2^4$
6	$2^7$	$2^7$	30	3	1/16	$2^{16}$	$2^8$
8	$2^{15}$	$2^{13}$	30	5	1/10486	15.6	$2^{16}$
16	$2^{57}$	$2^5$	18	15	$2^{-55.1}$	16	$2^{58}$

# DES S-Box Design Criteria

- No S-box is linear or affine function of its input.
- Changing one bit in the input of an S-Box changes at least two output bits.
- S-boxes were chosen to minimize the difference between the number of 1's and 0's when any input bit is held constant.
- $S(X)$  and  $S(X \oplus 001100)$  differ in at least 2 bits
- $S(X) \oplus S(X \oplus 11xy00)$

# Comments on effect of components on Differential Cryptanalysis

- E
  - Without expansion, there is a 4 round iterative characteristic with  $p=1/256$
- P
  - Major influence. If  $P=I$ , there is a 10-round characteristic with  $p=2^{-14.5}$  (but other attacks would be worse).
- S Box order
  - If S1, S7 and S4 were in order, there would be a 2 round iterative characteristic with  $p=1/73$ . However, Matsui found an order (24673158) that is better and also better against Linear crypto. Optimum order for LC resistance: 27643158.
- S properties
  - S boxes are nearly optimum against differential crypto

# Linear Cryptanalysis

- Basic idea:
  - Suppose  $\alpha_i(P) \oplus \beta_i(C) = g_i(k)$  holds with  $g_i$ , linear, for  $i = 1, 2, \dots, m$ .
  - Each equation imposes a linear constraint and reduces key search by a factor of 2.
  - Guess  $(n-m-1)$  bits of key. There are  $2^{(n-m-1)}$ . Use the constraints to get the remaining keys.
- Can we find linear constraints in the “per round” functions and knit them together?
- No! Per Round functions do not have linear constraints.

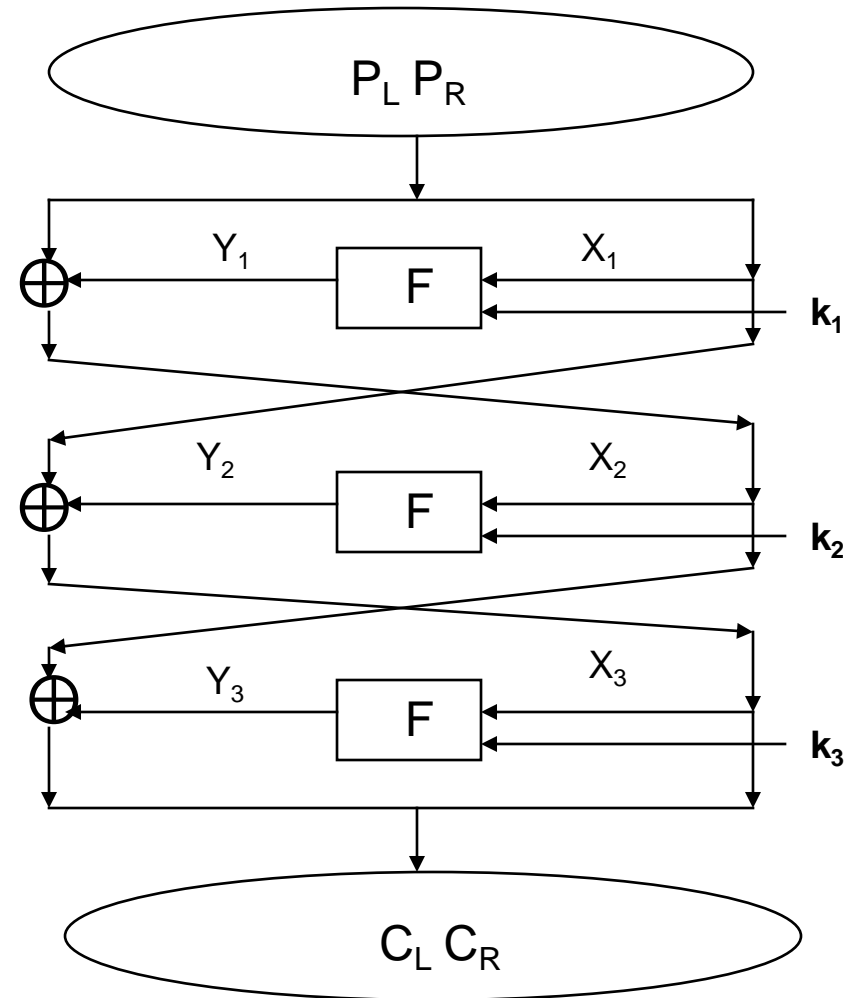
# Linear Cryptanalysis

- Next idea
  - Can we find  $\alpha(P) \oplus \beta(C) = L(k)$  which holds with  $L$ , linear, with probability  $p$ ?
  - Suppose  $\alpha(P) \oplus \beta(C) = L(k)$ , with probability  $p > .5$ .
  - Collect a lot of plain/cipher pairs.
  - Each will “vote” for  $L(k)=0$  or  $L(k)=1$ .
  - Pick the winner.
- $p = 1/2 + \epsilon$  requires  $c \epsilon^{-2}$  texts (we’ll see why later).
- $\epsilon$  is called “bias”.



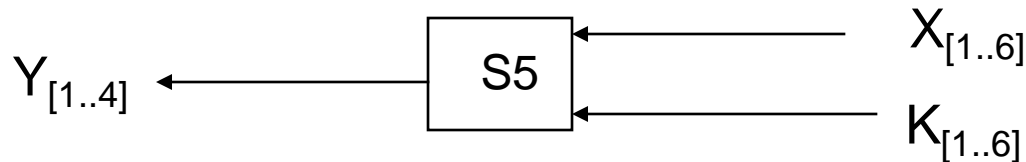
# Linear Cryptanalysis Notation

- Matsui numbers bits from right to left, rightmost bit is bit 0. FIPS (and everyone else) goes from left to right starting at 1. I will use the FIPS conventions. To map Matsui positions to everyone else's:
  - $M(i) = 64 - EE(i)$ . For 32 bits make the obvious change.
- Matsui also refers to the two portions of the plaintext and cipher-text as  $(P_H, P_L)$ ,  $(C_H, C_L)$ , we'll stick with  $(P_L, P_R)$ ,  $(C_L, C_R)$ .



# Linear and near linear dependence

- Here is a linear relationship over  $GF(2)$  in  $S5$  that holds with probability  $52/64$  (from  $NS_5(010000,1111)=12$ ):



- $X[2] \oplus Y[1] \oplus Y[2] \oplus Y[3] \oplus Y[4] = K[2] \oplus 1$ .
- Sometimes written:  $X[2] \oplus Y[1,2,3,4] = K[2] \oplus 1$ .
- You can find relations like this using the “Boolean Function” techniques we describe a little later
- After applying  $P$ , this becomes  

$$X[17] \oplus F(X,K)[3,8,14,25] = K[26] \oplus 1$$

# Linear Cryptanalysis of 3 round DES

$$X[17] \oplus Y[3,8,14,25] = K[26] \oplus 1, \quad p = 52/64$$

- Round 1

$$X_1[17] \oplus Y_1[3,8,14,25] = K_1[26] \oplus 1$$

$$P_R[17] \oplus P_L[3,8,14,25] \oplus R_1[3,8,14,25] = K_1[26] \oplus 1$$

- Round 3

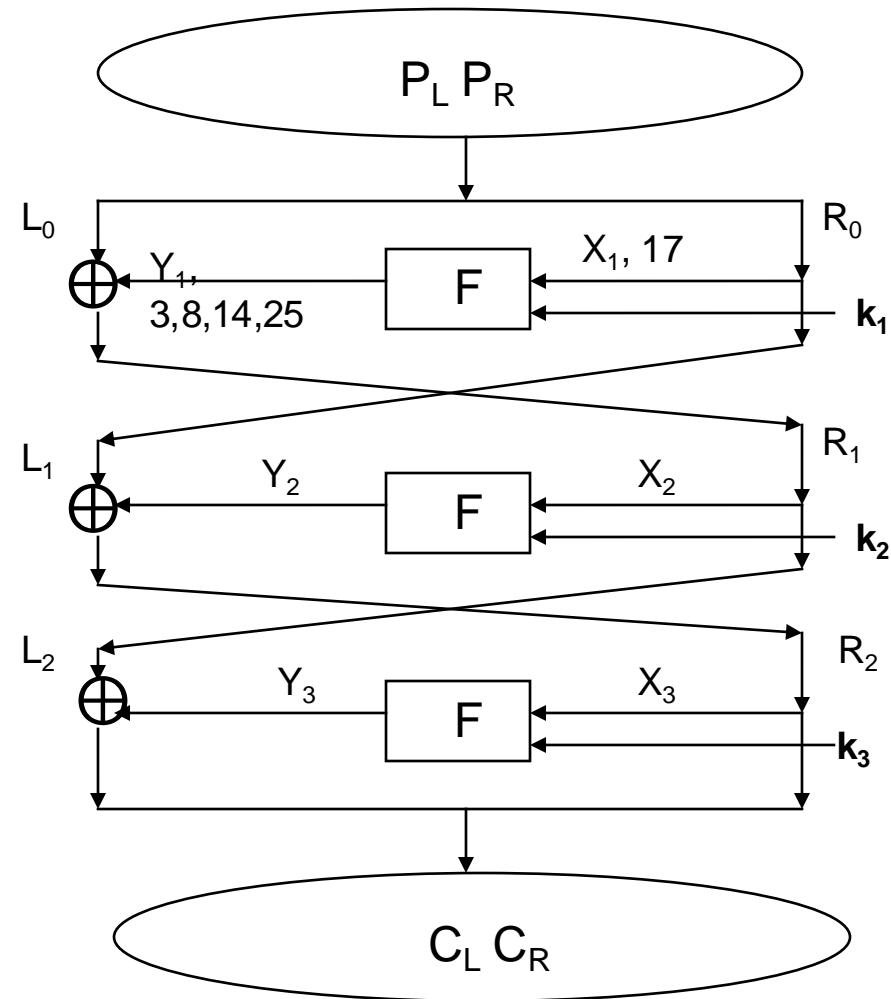
$$X_3[17] \oplus Y_3[3,8,14,25] = K_3[26] \oplus 1$$

$$R_1[3,8,14,25] \oplus C_L[3,8,14,25] \oplus C_R[17] = K_3[26] \oplus 1$$

- Adding the two get:

$$P_R[17] \oplus P_L[3,8,14,25] \oplus C_L[3,8,14,25] \oplus C_R[17] = K_1[26] \oplus K_3[26]$$

Holds with  $p = (52/64)^2 + (12/64)^2 = .66$



# Piling up Lemma

- Let  $X_i$  ( $1 \leq i \leq n$ ) be independent random variables whose values are 0 with probability  $p_i$ . Then the probability that  $X_1 \oplus X_2 \oplus \dots \oplus X_n = 0$  is

$$\frac{1}{2} + 2^{n-1} \prod_{[1,n]} (p_i - 1/2)$$

Proof:

By induction on  $n$ . It's tautological for  $n=1$ .

Suppose  $\Pr[X_1 \oplus X_2 \oplus \dots \oplus X_{n-1} = 0] = q = \frac{1}{2} + 2^{n-2} \prod_{[1,n-1]} (p_i - 1/2)$ .

Then  $\Pr[X_1 \oplus X_2 \oplus \dots \oplus X_n = 0] = qp_n + (1-q)(1-p_n) = \frac{1}{2} + 2^{n-1} \prod_{[1,n]} (p_i - 1/2)$  as claimed.

# Mathematics of biased voting

- Central Limit Theorem. Let  $X, X_1, \dots, X_n$  be independent, identically distributed random variables and let  $S_n = X_1 + X_2 + \dots + X_n$ . Let  $m = E(X)$  and  $\sigma^2 = E((X - \mu)^2)$ . Finally set  $T_n = (S_n - n\mu)/(\sigma\sqrt{n})$ ,  $n(x) = 1/(\sqrt{2\pi}) \exp(-x^2/2)$  and

$$N(a,b) = \int_{[a,b]} n(x) dx.$$

Then

$$\Pr(a \leq T_n \leq b) = N(a,b).$$

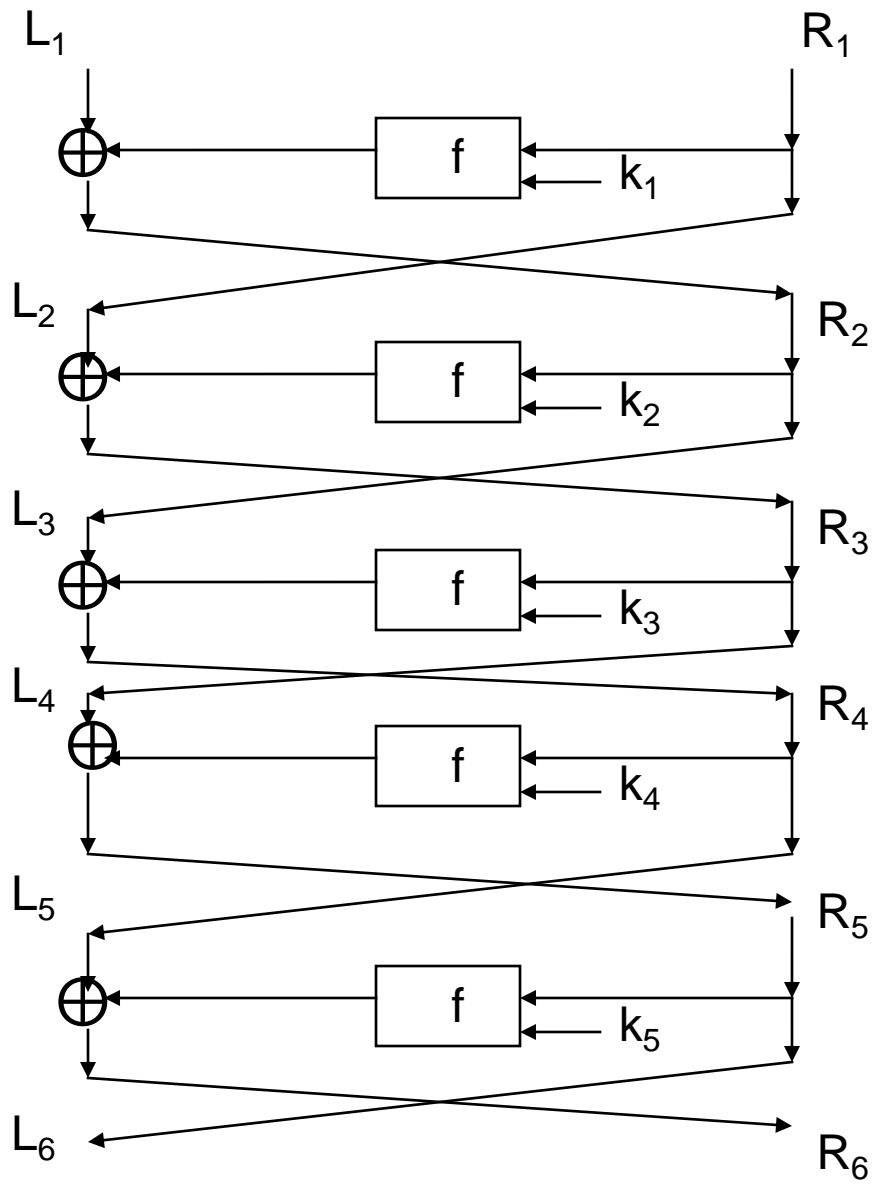
- $N$  is the Normal Distribution; it is symmetric around  $x=0$ .
- $N(-\infty, 0) = \frac{1}{2}$ .
- $N(-.5, .5) = .38$ ,  $N(-.75, .75) = .55$ ,  $N(-1, 1) = .68$ ,
- $N(-2, 2) = .9546$ ,  $N(-3, 3) = .9972$

# Application of CLT to LC

- $p = \frac{1}{2} + \epsilon$ ,  $1-p = \frac{1}{2} - \epsilon$ . Let  $L(k, P, E_k(P)) = 0$  be an equation over  $GF(2)$  that holds with probability  $p$ . Let  $X_i$  be the outcome (1 if true, 0 if false) of an experiment picking  $P$  and testing whether  $L$  holds for the real  $k$ .
- $E(X_i) = p$ ,  $E((X_i - p)^2) = p(1-p)^2 + (1-p)(0-p)^2 = p(1-p)$ . Let  $T_n$  be as provided in the CLT.
- Fixing  $n$ , what is the probability that more than half the  $X_i$  are 1 (i.e.- What is the probability that  $n$  random equations vote for the right key)?
- This is just  $\Pr(T_n \sigma - \epsilon \sqrt{n} / \sqrt{1/4 - \epsilon^2})$ . If  $n = d^2 \epsilon^{-2}$ , this is just  $\Pr(T_n \sigma - d / \sqrt{1/4 - \epsilon^2})$  or, if  $\epsilon$  is small  $\Pr(T_n \sigma - 2d)$ .
- Some numerical values:  $d = .25$ ,  $N(-.5, \infty) = .69$ ,  $d = .5$ ,  $N(-1, \infty) = .84$ ,  $d = 1$ ,  $N(-2, \infty) = .98$ ,  $d = 1.5$ ,  $N(-3, \infty) = .999$ .

# End

Thank you, IBM, and collaborators, for a great cipher.



...