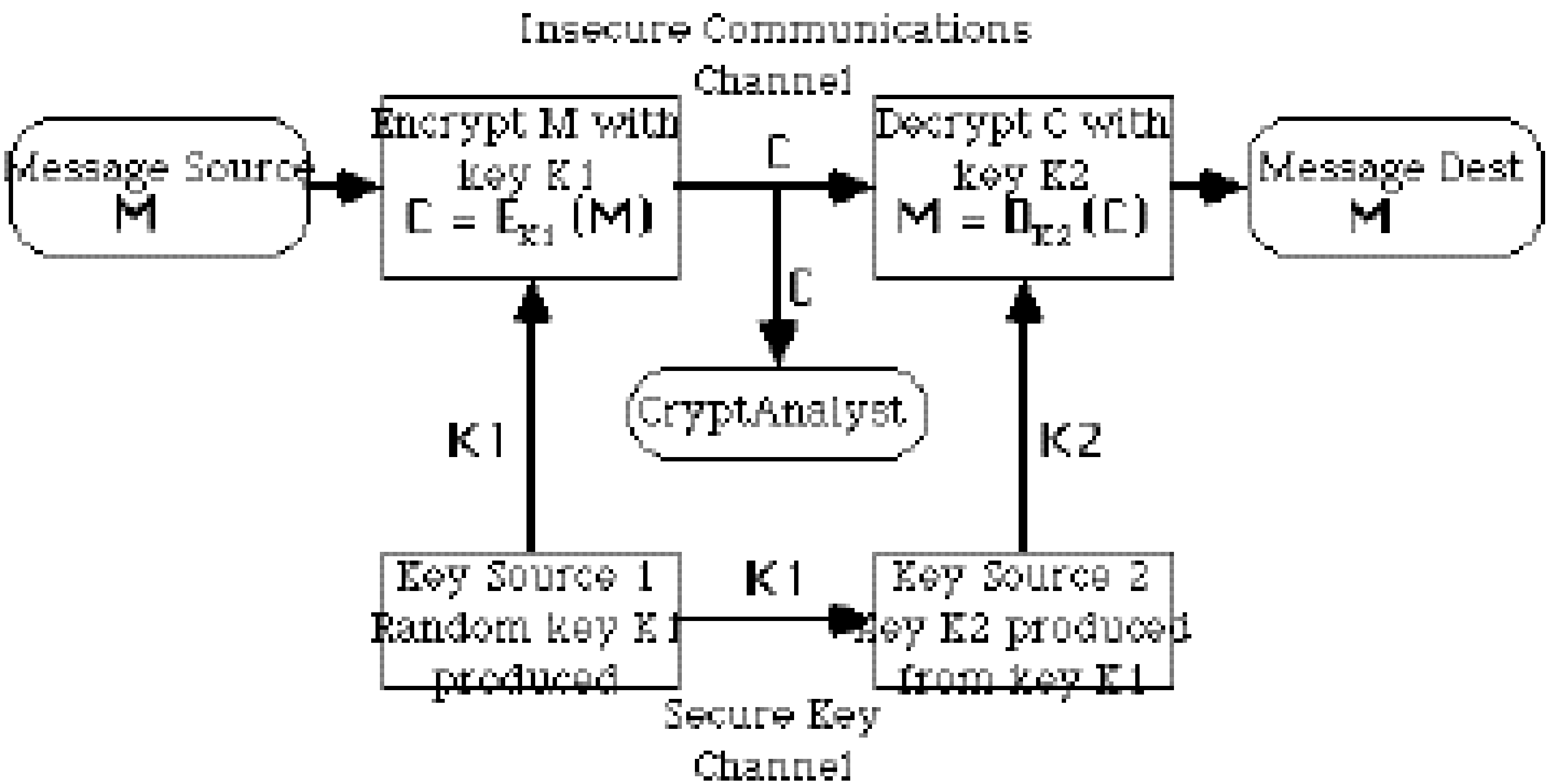
Modern Block Ciphers

- now look at modern block ciphers
- one of the most widely used types of cryptographic algorithms
- provide secrecy /authentication services
- focus on DES (Data Encryption Standard)
- to illustrate block cipher design principles

Symmetric Cryptosystems

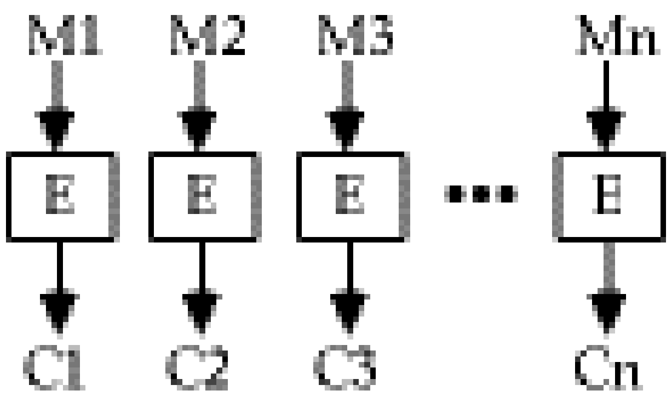


Symmetric (Private-Key) Encryption System

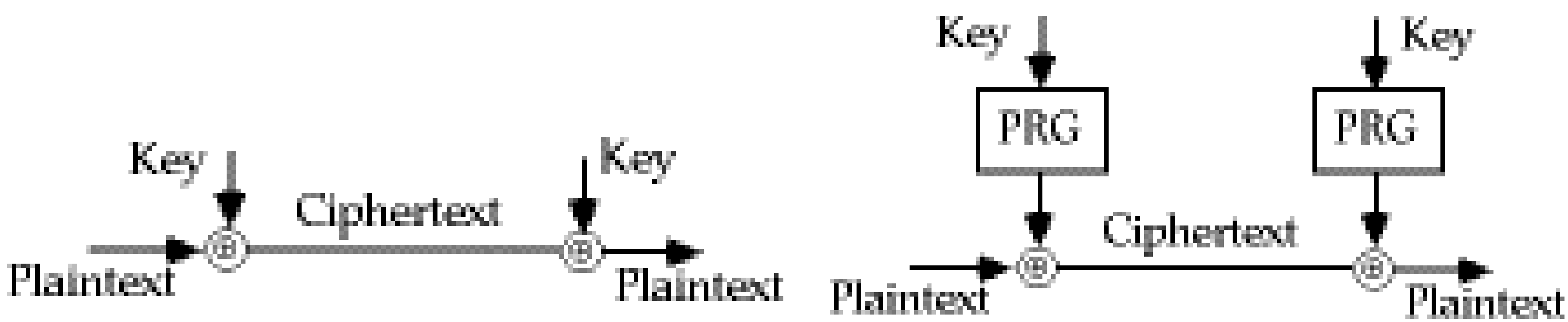
Block vs Stream Ciphers

- block ciphers process messages in blocks, each of which is then en/decrypted
- like a substitution on very big characters
  - 64-bits or more
- stream ciphers process messages a bit or byte at a time when en/decrypting
- many current ciphers are block ciphers
- broader range of applications

Block and Stream ciphers



Block Cipher

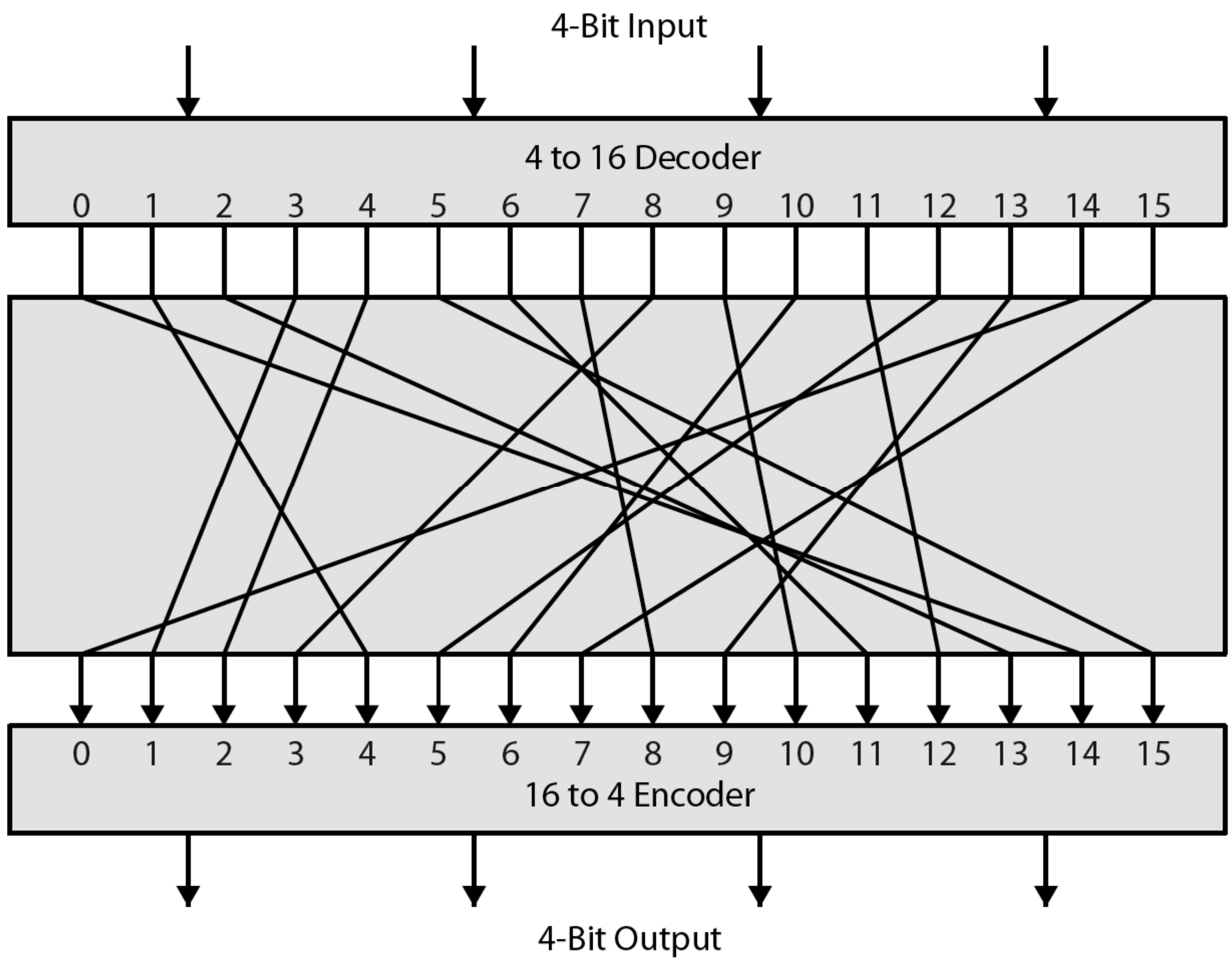


Stream Cipher

Block Cipher Principles

- most symmetric block ciphers are based on a **Feistel Cipher Structure**
- needed since must be able to **decrypt** ciphertext to recover messages efficiently
- block ciphers look like an extremely large substitution
- would need table of 2^{64} entries for a 64-bit block
- instead create from smaller building blocks
- using idea of a product cipher

Ideal Block Cipher



Claude Shannon and Substitution-Permutation Ciphers

- Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper
- form basis of modern block ciphers
- S-P nets are based on the two primitive cryptographic operations seen before:
  - *substitution* (S-box)
  - *permutation* (P-box)
- provide *confusion* & *diffusion* of message & key

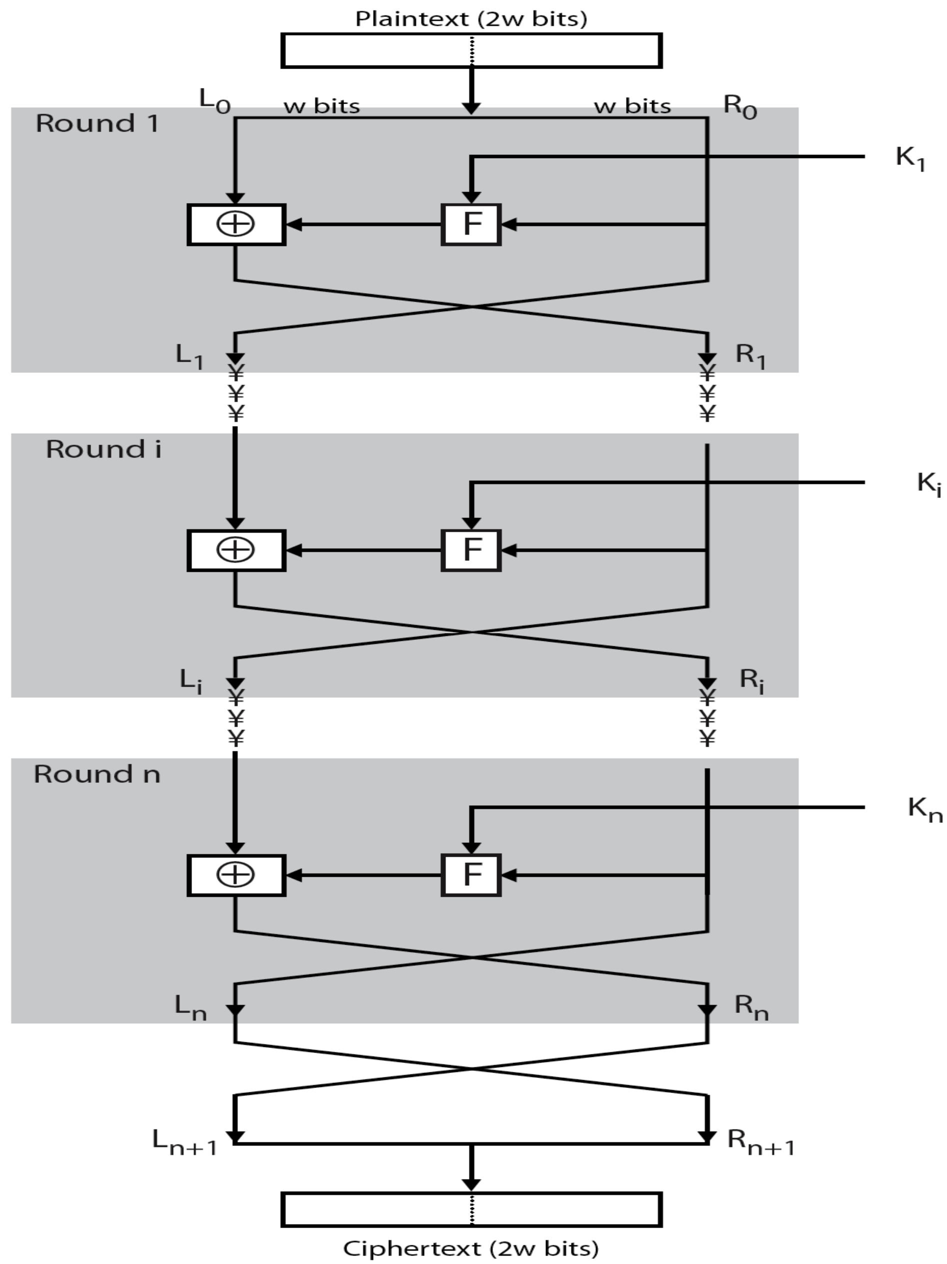
Confusion and Diffusion

- cipher needs to completely obscure statistical properties of original message
- a one-time pad does this
- more practically Shannon suggested combining S & P elements to obtain:
- **diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext
- **confusion** – makes relationship between ciphertext and key as complex as possible

Feistel Cipher Structure

- Horst Feistel devised the **feistel cipher**
  - based on concept of invertible product cipher
- partitions input block into two halves
  - process through multiple rounds which
  - perform a substitution on left data half
  - based on round function of right half & subkey
  - then have permutation swapping halves
- implements Shannon's S-P net concept

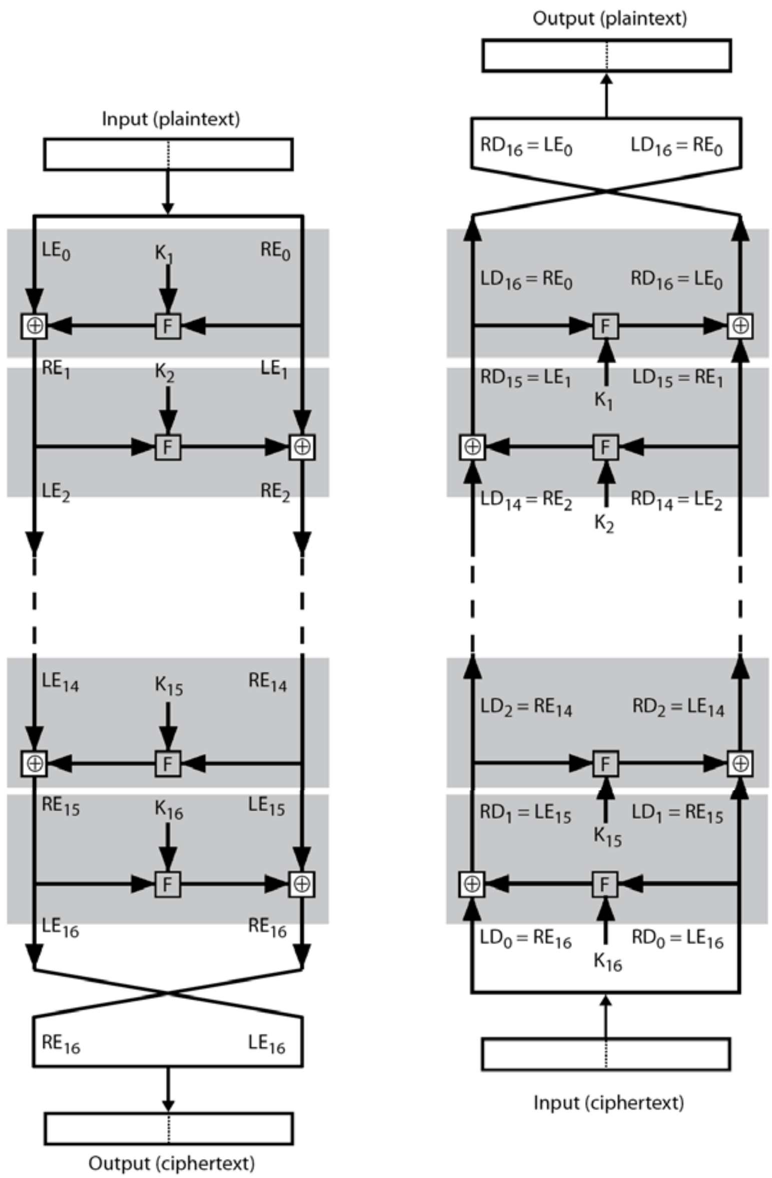
Feistel Cipher Structure



Feistel Cipher Design Elements

- block size
- key size
- number of rounds
- subkey generation algorithm
- round function
- fast software en/decryption
- ease of analysis

Feistel Cipher Decryption



Data Encryption Standard (DES)

- most widely used block cipher in world
- adopted in 1977 by NBS (now NIST)
  - as FIPS PUB 46
- encrypts 64-bit data using 56-bit key
- has widespread use
- has been considerable controversy over its security

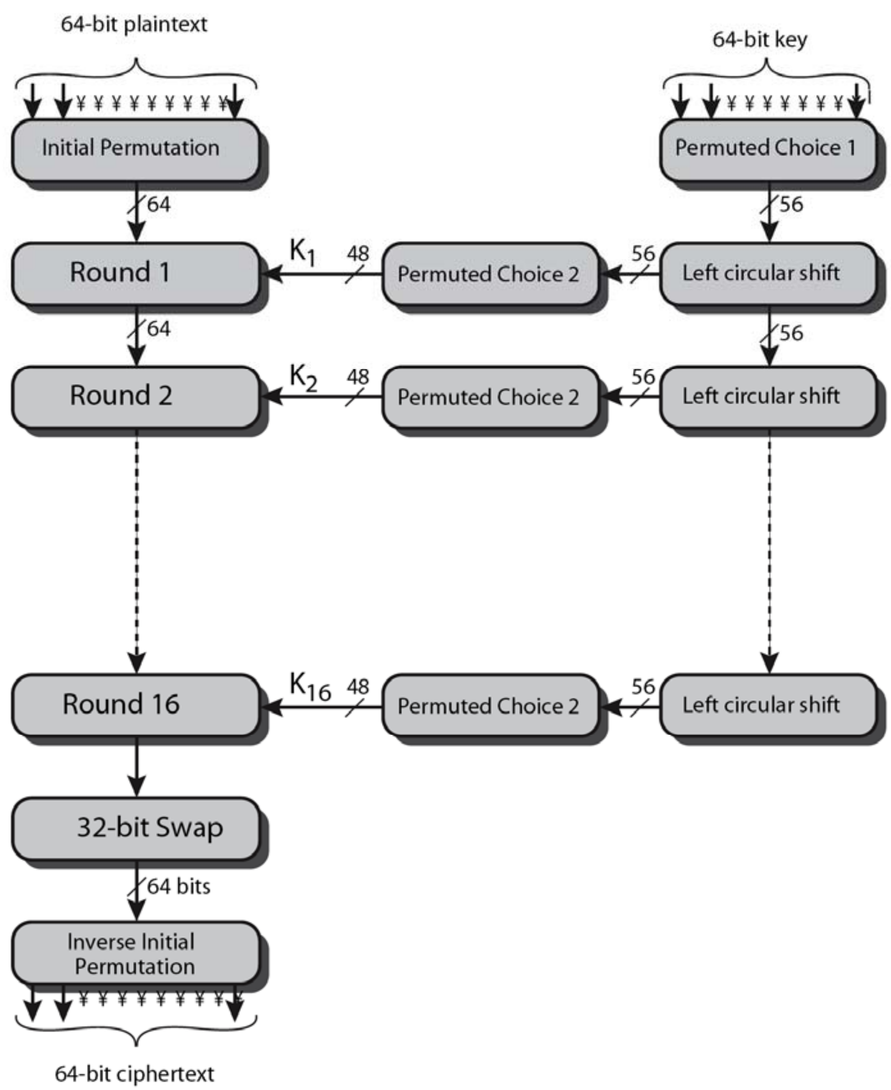
DES History

- IBM developed Lucifer cipher
  - by team led by Feistel in late 60's
  - used 64-bit data blocks with 128-bit key
- then redeveloped as a commercial cipher with input from NSA and others
- in 1973 NBS issued request for proposals for a national cipher standard
- IBM submitted their revised Lucifer which was eventually accepted as the DES

DES Design Controversy

- although DES standard is public
- was considerable controversy over design
  - in choice of 56-bit key (vs Lucifer 128-bit)
  - and because design criteria were classified
- subsequent events and public analysis show in fact design was appropriate
- use of DES has flourished
  - especially in financial applications
  - still standardised for legacy application use

DES Encryption Overview



Initial Permutation IP

- first step of the data computation
- IP reorders the input data bits
- even bits to LH half, odd bits to RH half
- quite regular in structure (easy in h/w)
- example:

```
IP(675a6967 5e5a6b5a) = (ffb2194d
004df6fb)
```

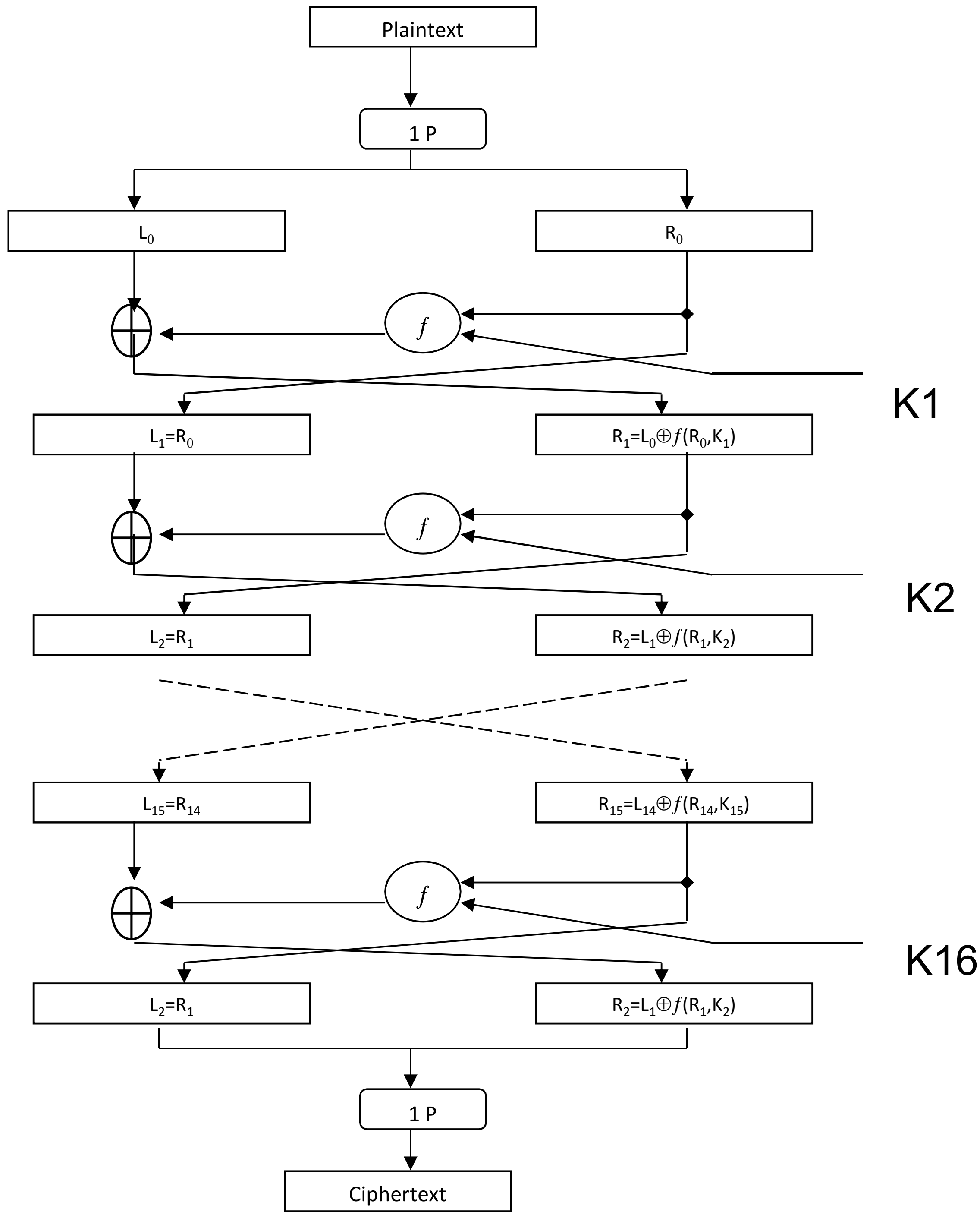
DES Round Structure

- uses two 32-bit L & R halves
- as for any Feistel cipher can describe as:

  $L_i = R_{i-1}$

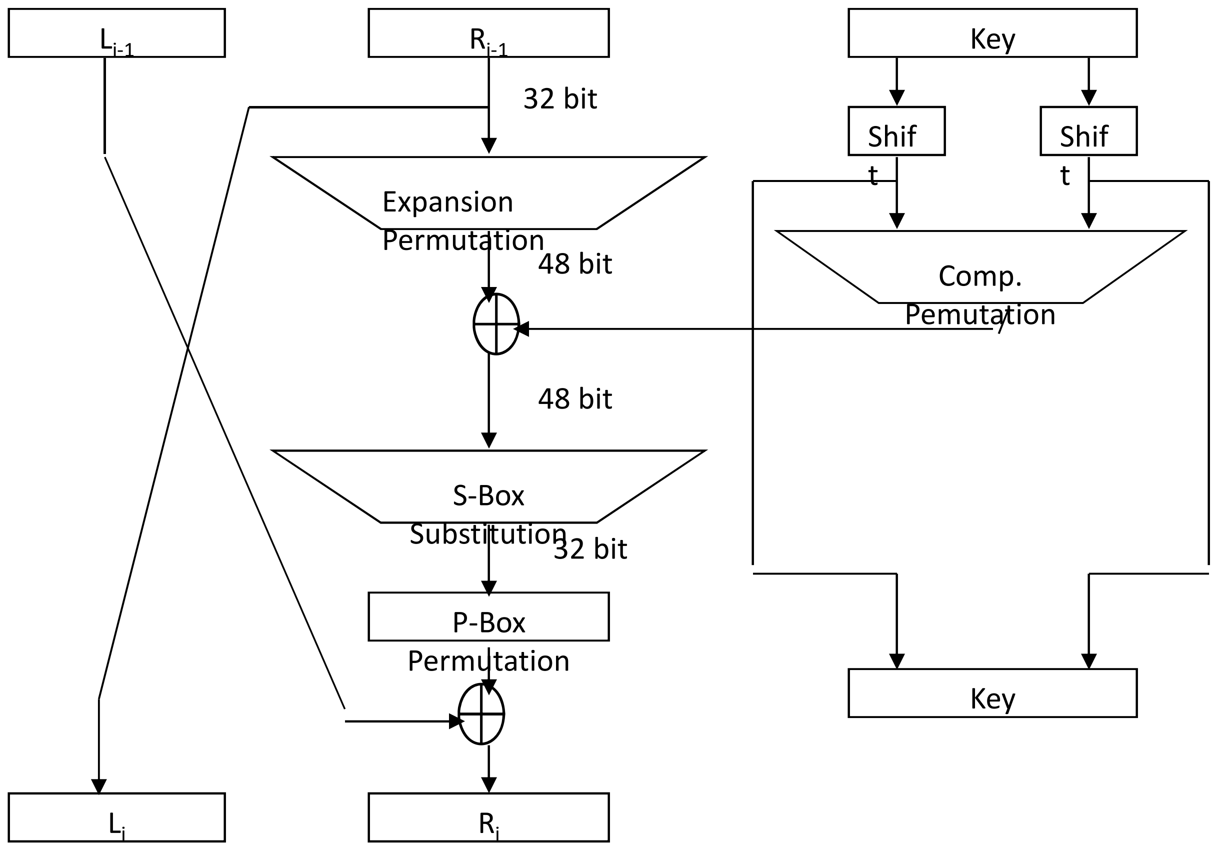
  $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

- F takes 32-bit R half and 48-bit subkey:
  - expands R to 48-bits using perm E
  - adds to subkey using XOR
  - passes through 8 S-boxes to get 32-bit result
  - finally permutes using 32-bit perm P

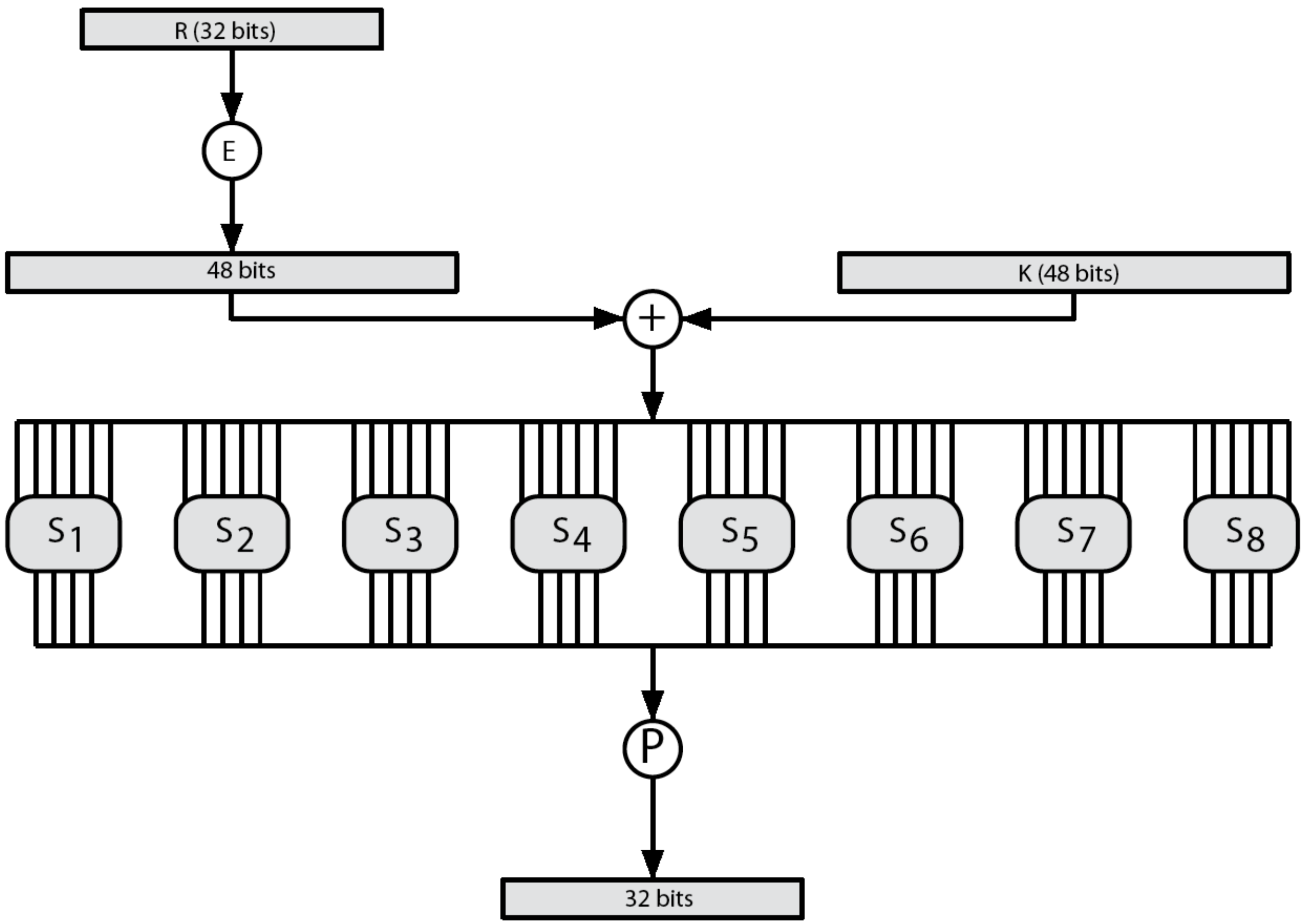


Plaintext
1 P
L_0
R_0
f
K1
$L_1=R_0$
$R_1=L_0\oplus f(R_0,K_1)$
f
K2
$L_2=R_1$
$R_2=L_1\oplus f(R_1,K_2)$
$L_{15}=R_{14}$
$R_{15}=L_{14}\oplus f(R_{14},K_{15})$
f
K16
$L_2=R_1$
$R_2=L_1\oplus f(R_1,K_2)$
1 P
Ciphertext

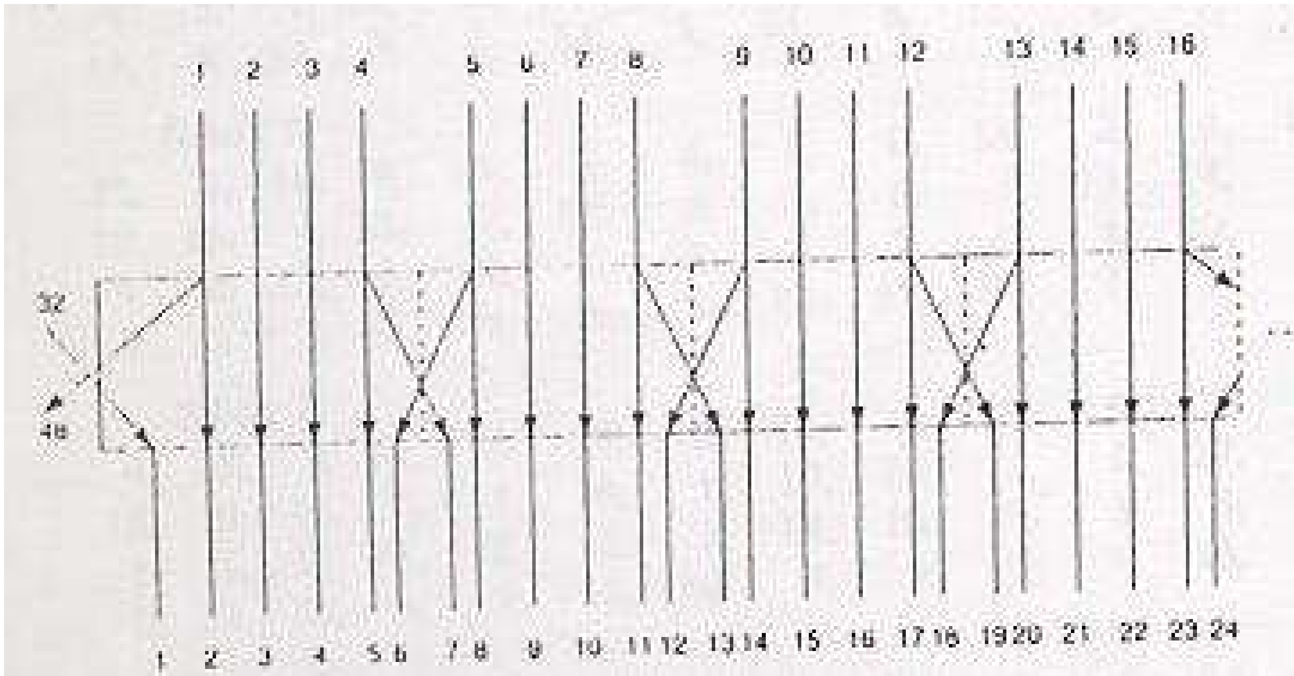
One round of DES



DES Round Structure



Expanded permutation



Substitution Boxes S

- have eight S-boxes which map 6 to 4 bits
- each S-box is actually 4 little 4 bit boxes
  - outer bits 1 & 6 (**row** bits) select one row of 4
  - inner bits 2-5 (**col** bits) are substituted
  - result is 8 lots of 4 bits, or 32 bits
- row selection depends on both data & key
  - feature known as autoclaving (autokeying)
- example:
  - `S(18 09 12 3d 11 17 38 39) = 5fd25e03`

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S1 0:	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7
1:	0	F	7	4	E	2	D	1	A	6	C	B	9	5	3	8
2:	4	1	E	8	D	6	2	B	F	C	9	7	3	A	5	0
3:	F	C	8	2	4	9	1	7	5	B	3	E	A	0	6	D
S2 0:	F	1	8	E	6	B	3	4	9	7	2	D	C	0	5	A
1:	3	D	4	7	F	2	8	E	C	0	1	A	6	9	B	5
2:	0	E	7	B	A	4	D	1	5	8	C	6	9	3	2	F
3:	D	8	A	1	3	F	4	2	B	6	7	C	0	5	E	9
S3 0:	A	0	9	E	6	3	F	5	1	D	C	7	B	4	2	8
1:	D	7	0	9	3	4	6	A	2	8	5	E	C	B	F	1
2:	D	6	4	9	8	F	3	0	B	1	2	C	5	A	E	7
3:	1	A	D	0	6	9	8	7	4	F	E	3	B	5	2	C
S4 0:	7	D	E	3	0	6	9	A	1	2	8	5	B	C	4	F
1:	D	8	B	5	6	F	0	3	4	7	2	C	1	A	E	9
2:	A	6	9	0	C	B	7	D	F	1	3	E	5	2	8	4
3:	3	F	0	6	A	1	D	8	9	4	5	B	C	7	2	E
S5 0:	2	C	4	1	7	A	B	6	8	5	3	F	D	0	E	9
1:	E	B	2	C	4	7	D	1	5	0	F	A	3	9	8	6
2:	4	2	1	B	A	D	7	8	F	9	C	5	6	3	0	E
3:	B	8	C	7	1	E	2	D	6	F	0	9	A	4	5	3
S6 0:	C	1	A	F	9	2	6	8	0	D	3	4	E	7	5	B
1:	A	F	4	2	7	C	9	5	6	1	D	E	0	B	3	8
2:	9	E	F	5	2	8	C	3	7	0	4	A	1	D	B	6
3:	4	3	2	C	9	5	F	A	B	E	1	7	6	0	8	D
S7 0:	4	B	2	E	F	0	8	D	3	C	9	7	5	A	6	1
1:	D	0	B	7	4	9	1	A	E	3	5	C	2	F	8	6
2:	1	4	B	D	C	3	7	E	A	F	6	8	0	5	9	2
3:	6	B	D	8	1	4	A	7	9	5	0	F	E	2	3	C
S8 0:	D	2	8	4	6	F	B	1	A	9	3	E	5	0	C	7
1:	1	F	D	8	A	3	7	4	C	5	6	B	0	E	9	2
2:	7	B	4	1	9	C	E	2	0	6	A	D	F	3	5	8
3:	2	1	E	7	4	A	8	D	F	C	9	0	3	5	6	B

P-box permutation

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

P-Box Permutasyonu

DES Key Schedule

- forms subkeys used in each round
  - initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
  - 16 stages consisting of:
    - rotating **each half** separately either 1 or 2 places depending on the **key rotation schedule** K
    - selecting 24-bits from each half & permuting them by PC2 for use in round function F
- note practical use issues in h/w vs s/w

DES Decryption

- decrypt must unwind steps of data computation
- with Feistel design, do encryption steps again using subkeys in reverse order (SK16 … SK1)
  - IP undoes final FP step of encryption
  - 1st round with SK16 undoes 16th encrypt round
  - ….
  - 16th round with SK1 undoes 1st encrypt round
  - then final FP undoes initial encryption IP
  - thus recovering original data value

Avalanche Effect

- key desirable property of encryption alg
- where a change of **one** input or key bit results in changing approx **half** output bits
- making attempts to “home-in” by guessing keys impossible
- DES exhibits strong avalanche