Chapter 3 – Block Ciphers and the Data Encryption Standard

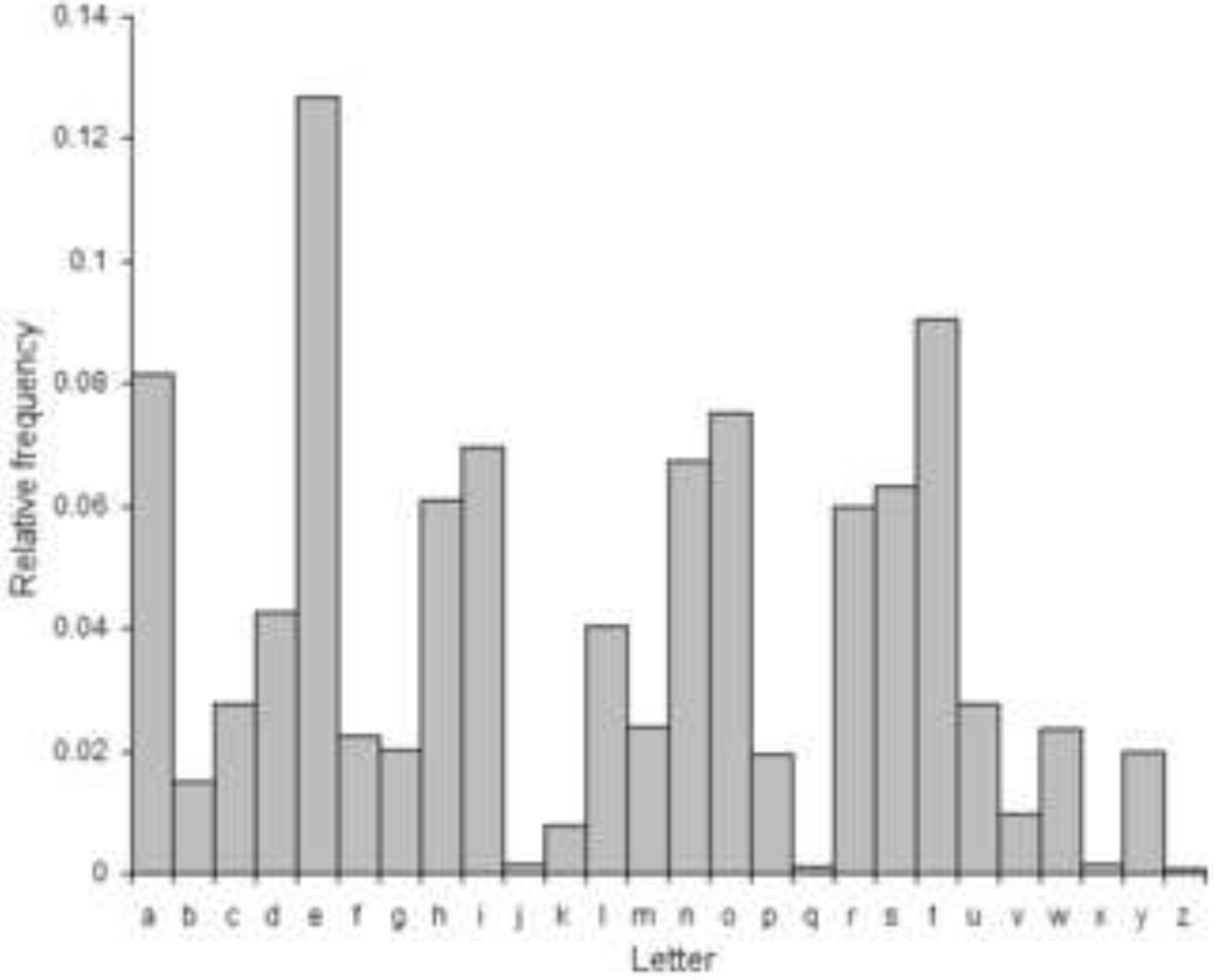
Character Frequencies

- In most languages letters are not equally common
  - in English **e** is by far the most common letter
- Have tables of single, double & triple letter frequencies
- Use these tables to compare with letter frequencies in ciphertext,
  - a monoalphabetic substitution does not change relative letter frequencies
  - do need a moderate amount of ciphertext (100+ letters)

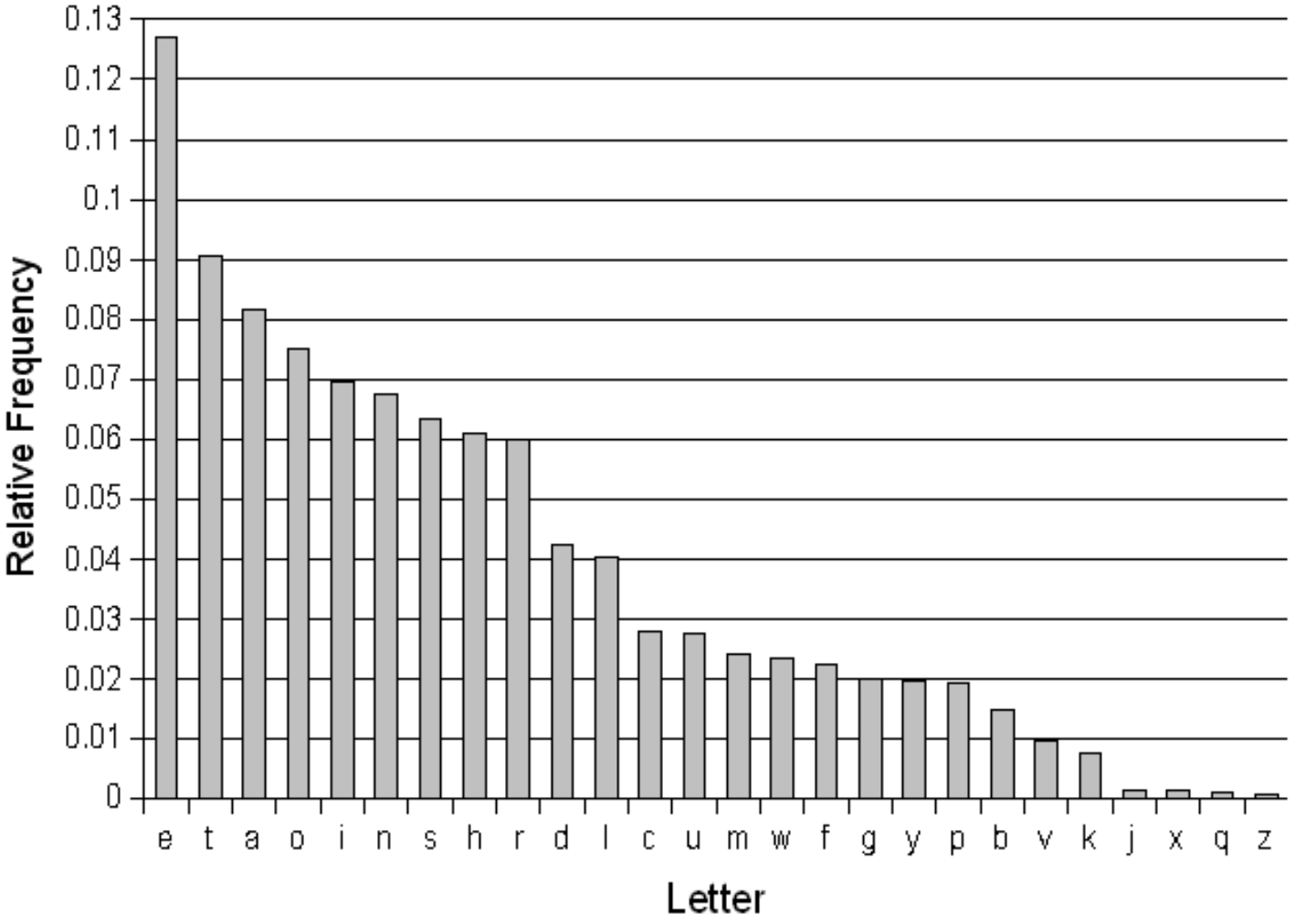
Letter Frequency Analysis

- Single Letter
  - A,B,C,D,E,…..
- Double Letter
  - TH,HE,IN,ER,RE,ON,AN,EN,….
- Triple Letter
  - THE,AND,TIO,ATI,FOR,THA,TER,RES,…

Letter Frequencies



Letter Frequencies



N-gram Frequencies

- **Digraph Frequency**
  - **th he an in er on re ed nd ha at en es of nt ea ti to io le is ou ar as de rt ve**
- **Trigraph Frequency**
  - **the and tha ent ion tio for nde has nce tis oft men**
  - **For more, see http://www.letterfrequency.org**

Modular Arithmetic Cipher

- Use a more complex equation to calculate the ciphertext letter for each plaintext letter
- $E_{(a,b)} : i \rightarrow a*i + b \bmod 26$
  - Need gcd(a,26) = 1
  - Otherwise, not reversible
  - So, $a \neq 2, 13, 26$
  - Caesar cipher: a=1, b=3

Block vs Stream Ciphers

- block ciphers process messages into 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
- hence are focus of course

Modern Block Ciphers

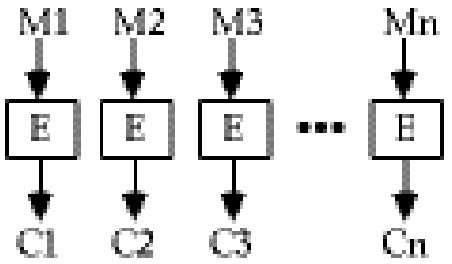
- will now look at modern block ciphers
- one of the most widely used types of cryptography algorithms
- provide strong secrecy and/or authentication services
- in particular will introduce DES (Data Encryption Standard)

Block Cipher Principles

- block ciphers look like an extremely large substitution
- would need table of 2^{64} entries for a 64-bit block
- arbitrary reversible substitution cipher for a large block size is not practical
  - 64-bit general substitution block cipher, key size 2^{64}!
- 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

- The message is broken into blocks,
  - Each of which is then encrypted
  - (Like a substitution on very big characters - 64-bits or more)



Substitution and Permutation

- In his 1949 paper Shannon also introduced the idea of substitution-permutation (S-P) networks, which now form the basis of modern block ciphers
  - An S-P network is the modern form of a substitution-transposition product cipher
  - S-P networks are based on the two primitive cryptographic operations we have seen before

Substitution

- A binary word is replaced by some other binary word
- The whole substitution function forms the key
- If use n bit words,
  - The key space is $2^n!$
- Can also think of this as a large lookup table, with n address lines (hence 2^n addresses), each n bits wide being the output value
- Will call them **s-boxes**

Cont.

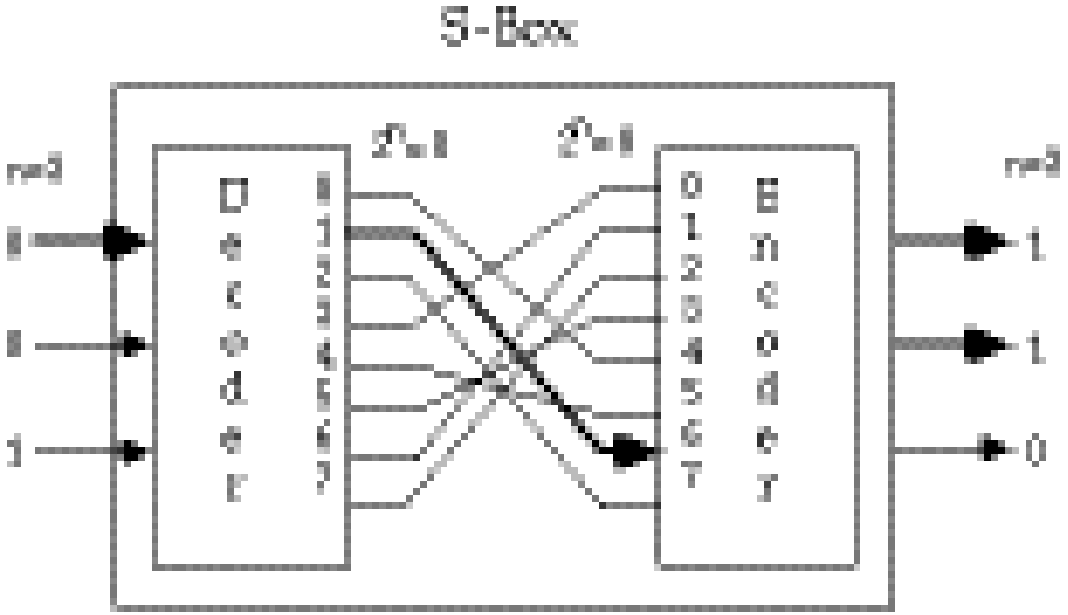


Fig 2.1 Substitution Operation

Permutation

- A binary word has its bits reordered (permuted)
- The re-ordering forms the key
- If use n bit words,
  - The key space is n! (Less secure than substitution)
- This is equivalent to a wire-crossing in practice
  - (Though is much harder to do in software)
- Will call these **p-boxes**

Cont.

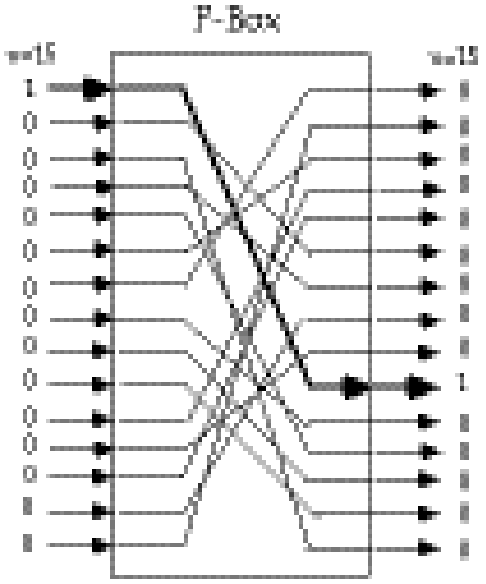


Fig 2.2 - Permutation or Transposition Function

Substitution-permutation Network

- Shannon combined these two primitives
- He called these **mixing transformations**
- A special form of product ciphers where
  - **S-boxes**
    - Provide **confusion** of input bits
  - **P-boxes**
    - Provide **diffusion** across s-box inputs

C. Shannon and Substitution-Permutation Ciphers

- in 1949 Shannon introduced idea of substitution-permutation (S-P) networks
  - modern substitution-transposition product cipher
- these form the basis of modern block ciphers
- S-P networks are based on the two primitive cryptographic operations we have seen before:
  - *substitution* (S-box)
  - *permutation* (P-box) (transposition)
- provide *confusion* and *diffusion* of message

Diffusion and Confusion

- Introduced by Claude Shannon to thwart cryptanalysis based on statistical analysis
  - Assume the attacker has some knowledge of the statistical characteristics of the plaintext
- cipher needs to completely obscure statistical properties of original message
- a one-time pad does this

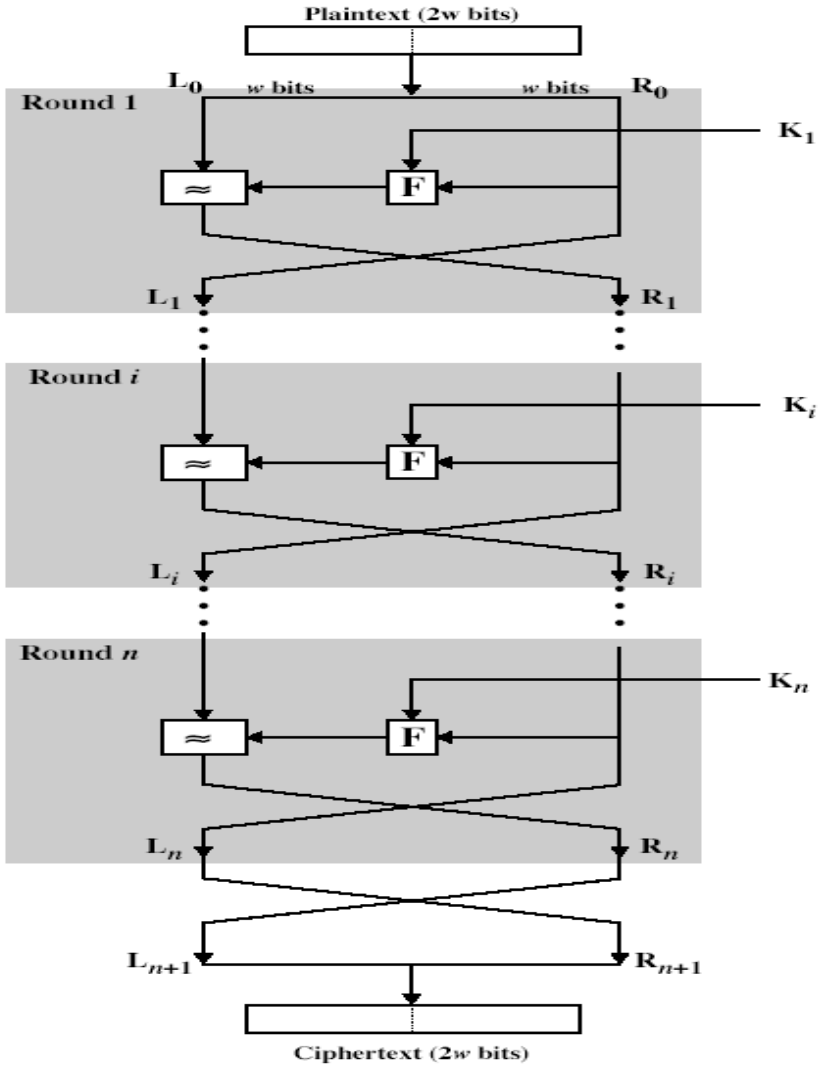
Diffusion and Confusion

- more practically Shannon suggested combining 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**
  - implements Shannon's substitution-permutation network concept
- 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

Feistel Cipher Structure



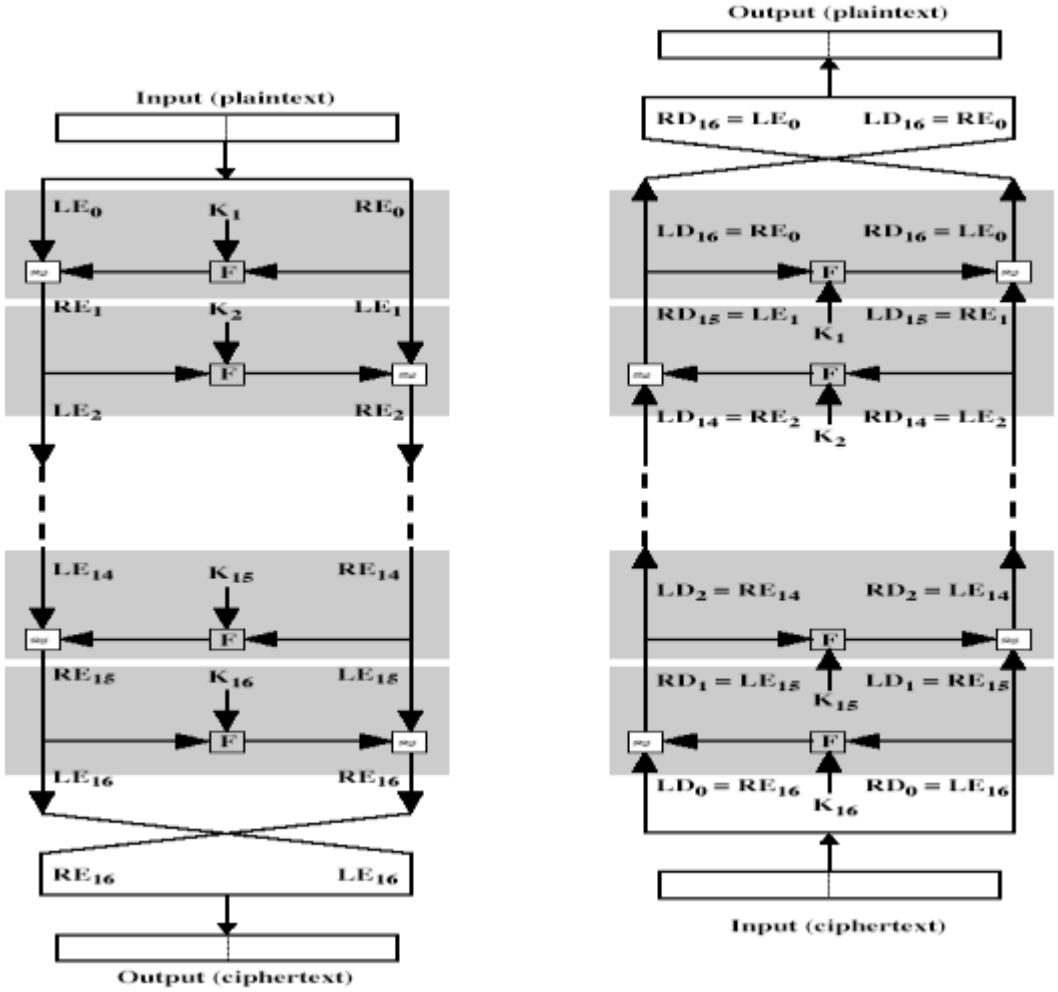
Feistel Cipher

- n sequential rounds
- A substitution on the left half L_i
  - 1. Apply a round function F to the right half R_i and
  - 2. Take XOR of the output of (1) and L_i
- The round function is parameterized by the subkey K_i
  - K_i are derived from the overall key K

Feistel Cipher Design Principles

- **block size**
  - increasing size improves security, but slows cipher
- **key size**
  - increasing size improves security, makes exhaustive key searching harder, but may slow cipher
- **number of rounds**
  - increasing number improves security, but slows cipher
- **subkey generation**
  - greater complexity can make analysis harder, but slows cipher
- **round function**
  - greater complexity can make analysis harder, but slows cipher
- **fast software en/decryption & ease of analysis**
  - are more recent concerns for practical use and testing

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

DES History

- IBM developed Lucifer cipher
  - by team led by Feistel
  - 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

History

- IBM LUCIFER 60's
  - Uses 128 bits key
- Proposal for NBS, 1973
- Adopted by NBS, 1977
  - Uses only 56 bits key
    - Possible brute force attack
  - Design of S-boxes was classified
    - Hidden weak points in in S-Boxes?
  - Wiener (93) claim to be able to build a machine at $100,00 and break DES in 1.5 days

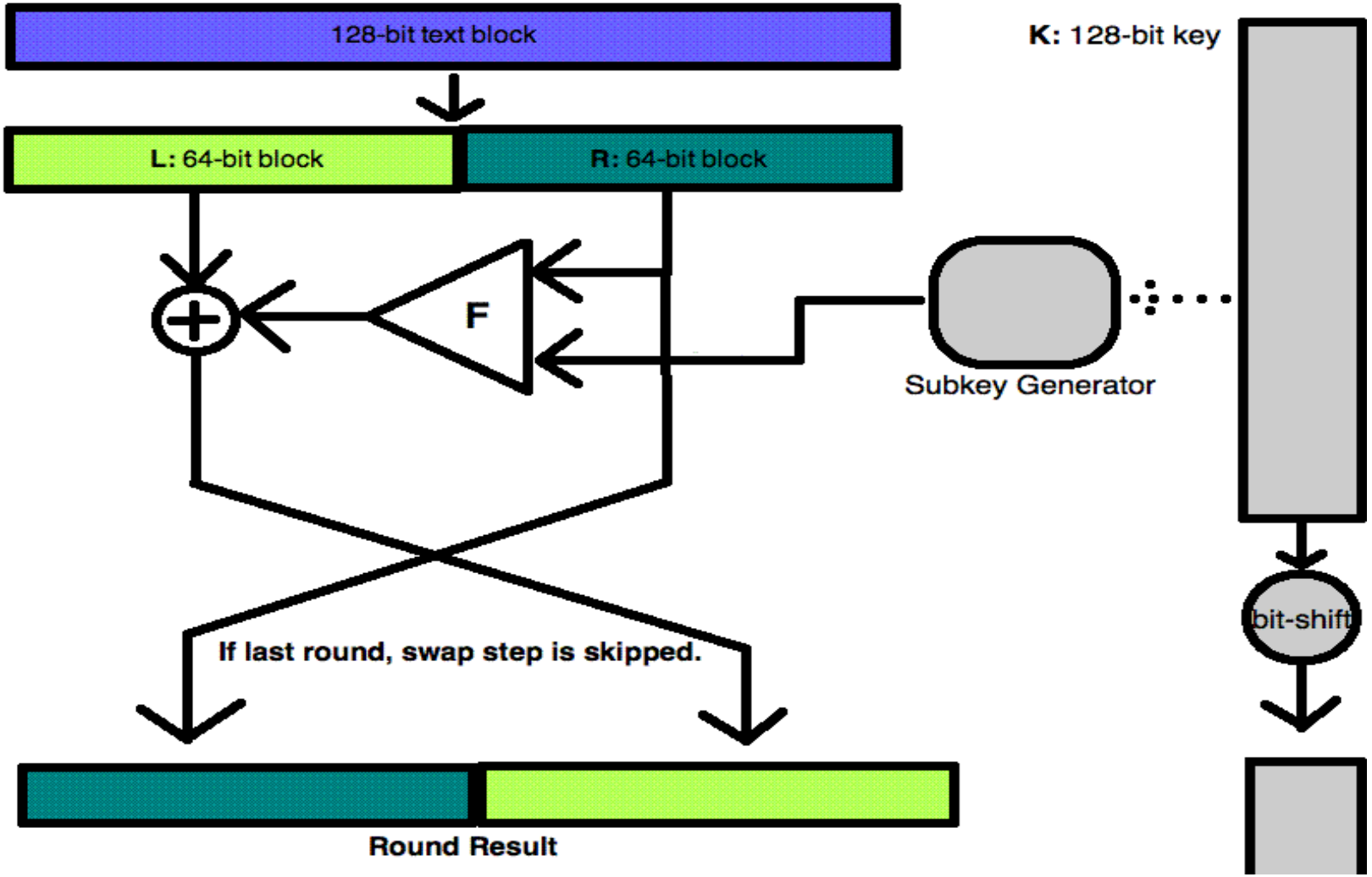
About Lucifer

- Created by Horst Feistel et al. at IBM
- One of the earliest block ciphers
  - “Father” of DES
- Limited commercial usage with banking software

About the Cipher

- Block Cipher
  - Plaintext blocks of 128 bits
  - 128 bit key
  - 72 bit sub-keys
  - Ciphertext blocks of 128 bits
- Feistel network
- 16 rounds per encryption

The Encryption Round



The Function

- XOR block with sub-key
- Block nibble swap
  - Based on bits of first sub-key byte
- Send nibbles to s-boxes
  - Left nibble to box 1, right nibble to box 2
- Bit permutation

Our Approach

- Separate the cipher from the practical program.
- Have separate “helper classes”.
    - Permutation Class
    - KeyHandler Class
    - SboxClass
- Keep readability high.
- High modularity for easy testing and optimization.
- Allow output as a hex string.
- Use Java - stick to what we know best!

Sample Encrypted Texts

- INPUT
-
- test
- sample input

- OUTPUT
-
- `be15cc3974c2f0ab55e38a881efafa23`
- `09b20463d448c50de9fc6ad609787a8d`

Timed Results (Version 1)

- 1,000,000 round encryption:

- **14716 msec encryption**
  1000000 repetitions
  1.47e-05 sec/encryption

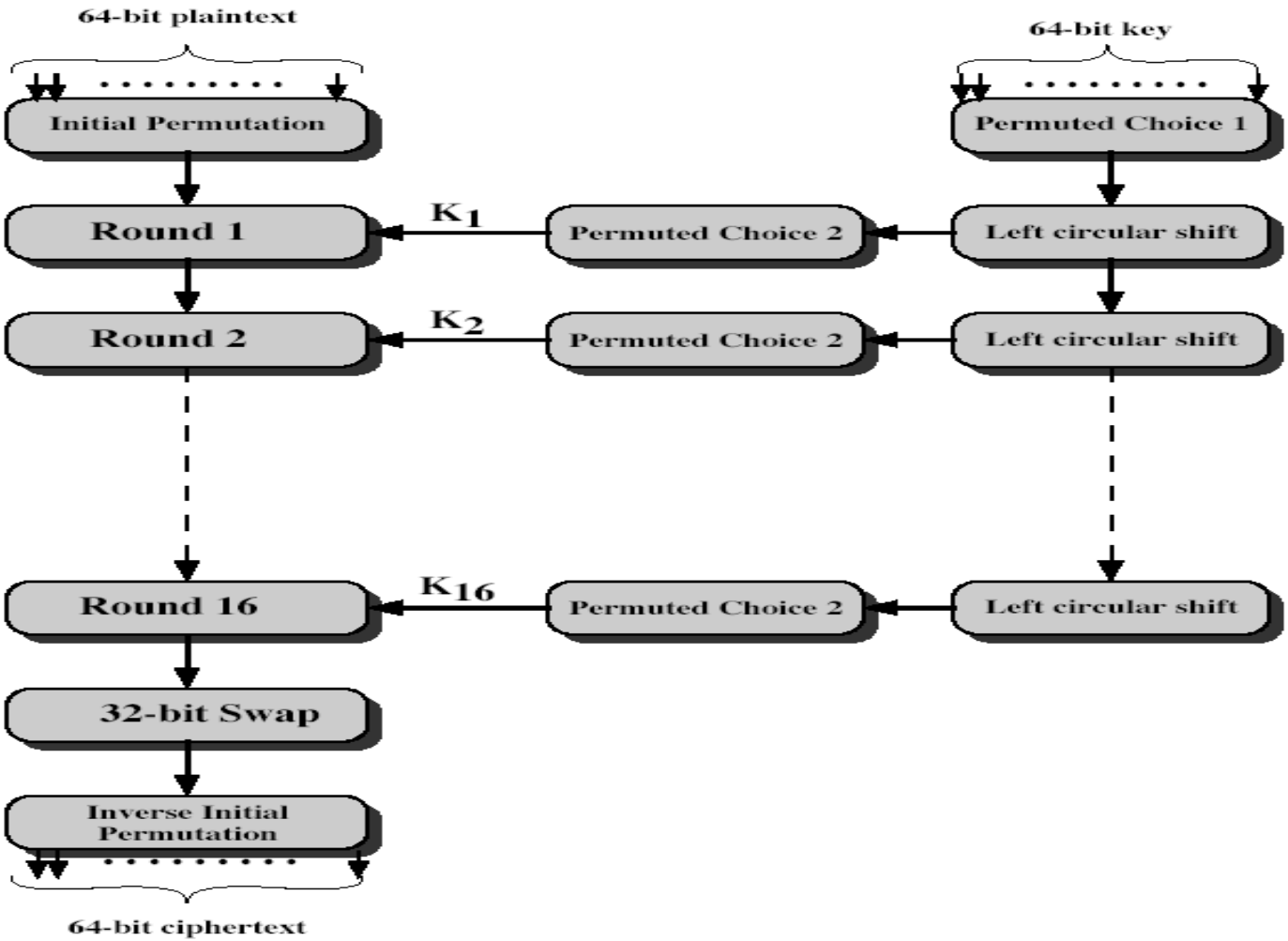
Questions?

Really? No Questions?

DES Design Controversy

- although DES standard is public
- was considerable controversy over design
  - in choice of 56-bit key (vs Lucifer 128-bit)
- subsequent events and public analysis show in fact design was appropriate
- DES has become widely used, especially in financial applications

DES Encryption



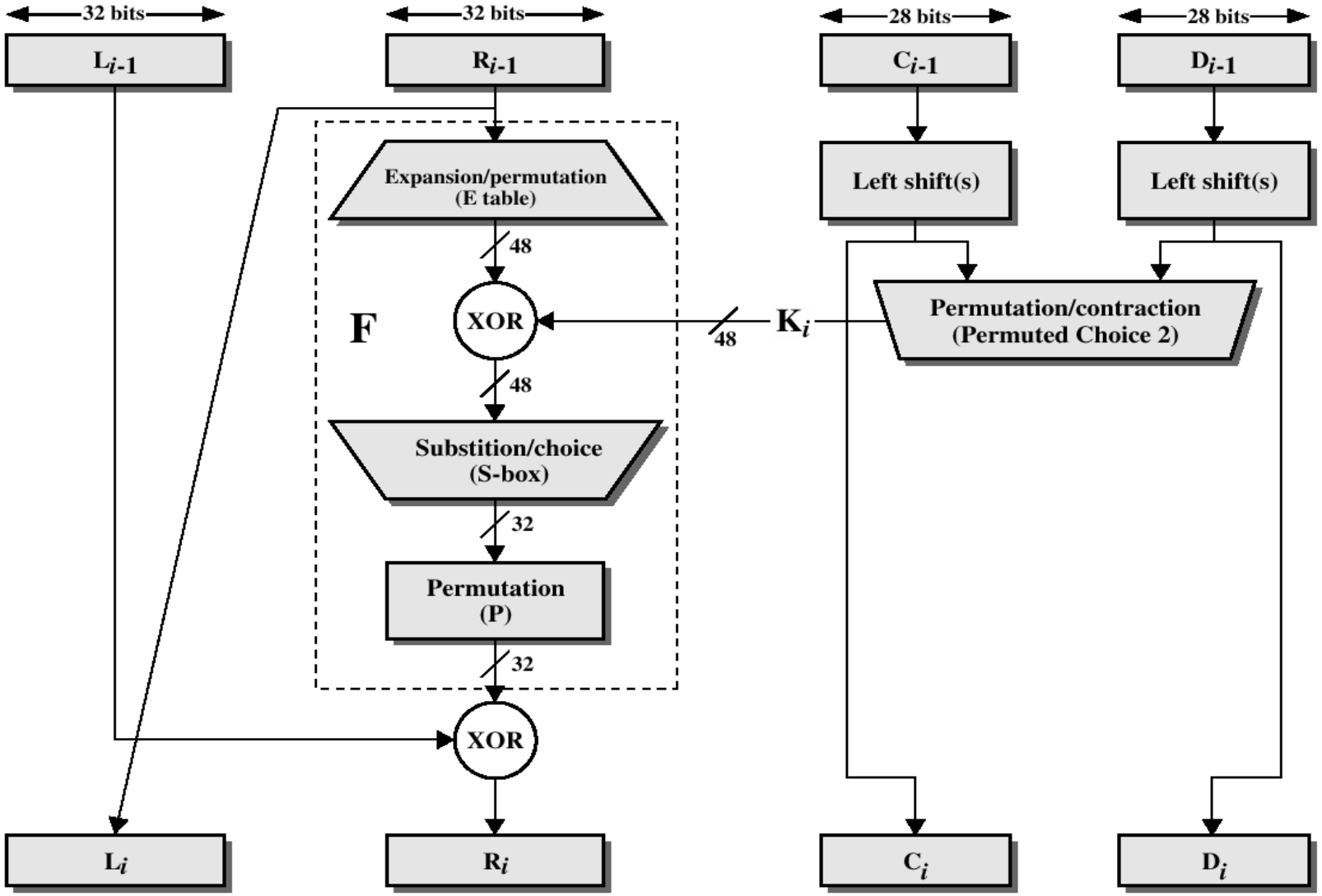


Figure 2.4 Single Round of DES Algorithm

Initial Permutation IP

- first step of the data computation
- IP reorders the input data bits
- quite regular in structure
- example:

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

Initial and Final Permutations

- Inverse Permutations

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Initial and Final Permutations

- Inverse Permutations

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

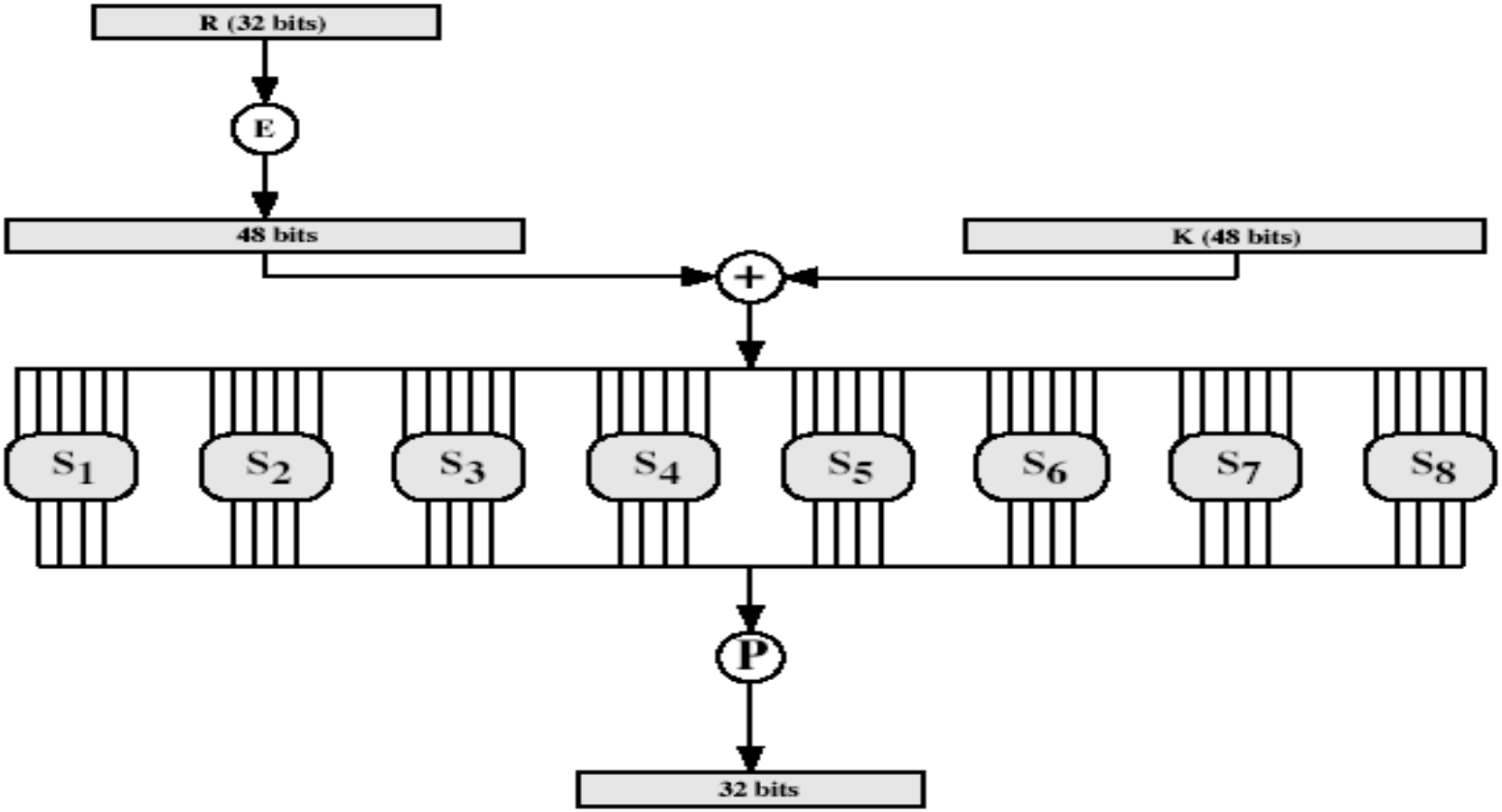
DES Round Structure

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

  $L_i = R_{i-1}$

  $R_i = L_{i-1}$ xor F(R_{i-1}, K_i)
- takes 32-bit R half and 48-bit subkey and:
  - expands R to 48-bits using Expansion Permutation E
  - adds to subkey
  - passes through 8 S-boxes to get 32-bit result
  - finally permutes this using 32-bit Permutation Function P

The round function F(R,K)



Substitution Boxes S

- 8 S-boxes (Table 3.3)
- Each S-Box mapps 6 to 4 bits
  - outer bits 1 & 6 (**row** bits) select the row
  - inner bits 2-5 (**col** bits) select the column
  - For example, in S1, for input 011001,
    - the row is 01 (row 1)
    - the column is 1100 (column 12).
    - The value in row 1, column 12 is 9
    - The output is 1001.
- result is 8 X 4 bits, or 32 bits

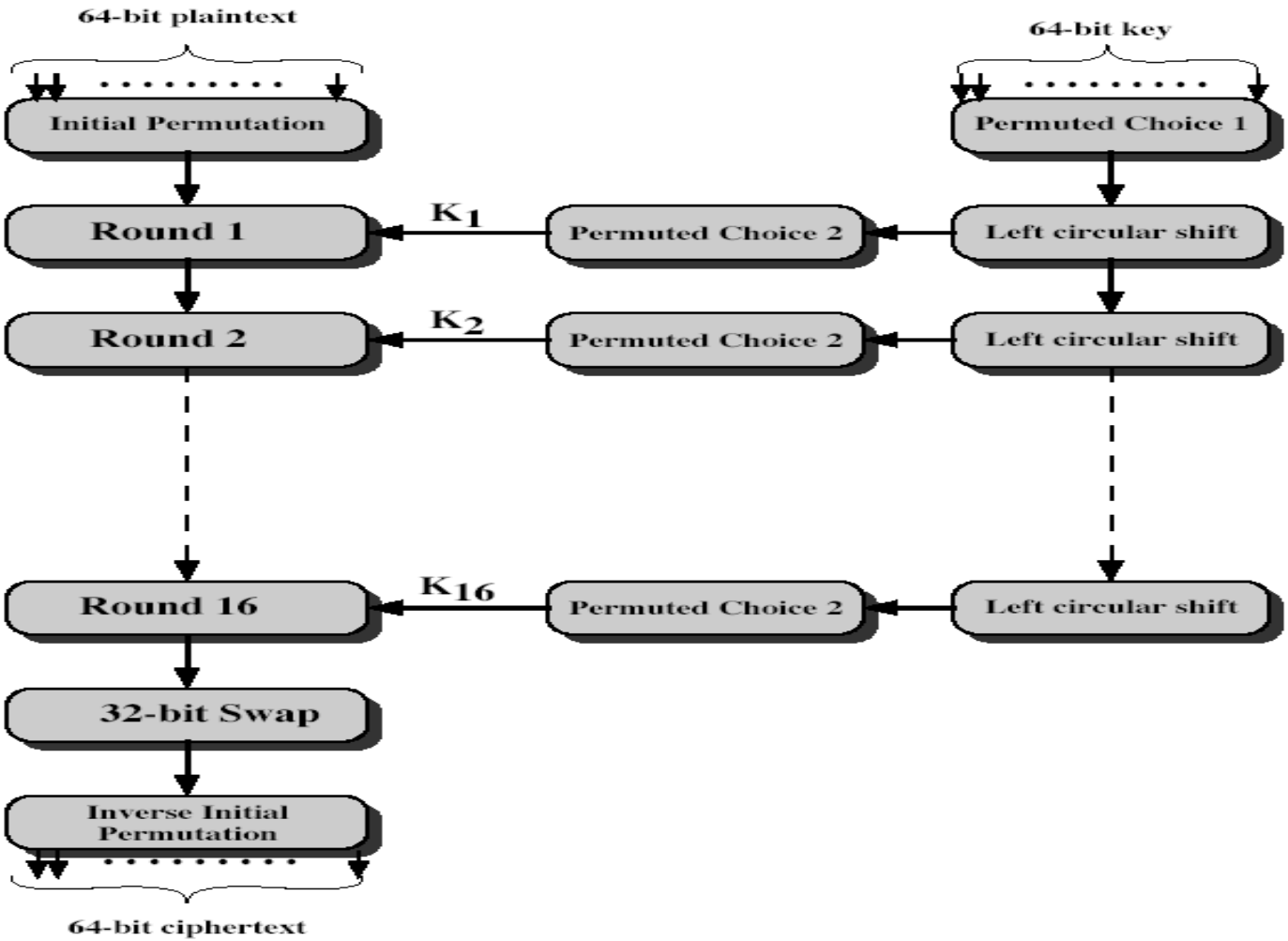
DES Key Schedule

- forms subkeys used in each round
- 1. initial permutation of the key PC1 (Table 3.4b)
- 2. divide the 56-bits in two 28-bit halves
- 3. at each round
  - 3.1. Left shift each half (28bits) separately either 1 or 2 places based on the left shift schedule (Table 3.4d)
    - Shifted values will be input for next round
  - 3.2. Combine two halfs to 56 bits, permuting them by PC2 (Table 3.4c) for use in function f
    - PC2 takes 56-bit input, outputs 48 bits

DES Decryption

- decrypt must unwind steps of data computation
- with Feistel design, do encryption steps again
- using subkeys in reverse order (SK16 … SK1)
- note that 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

DES Decryption (reverse encryption)



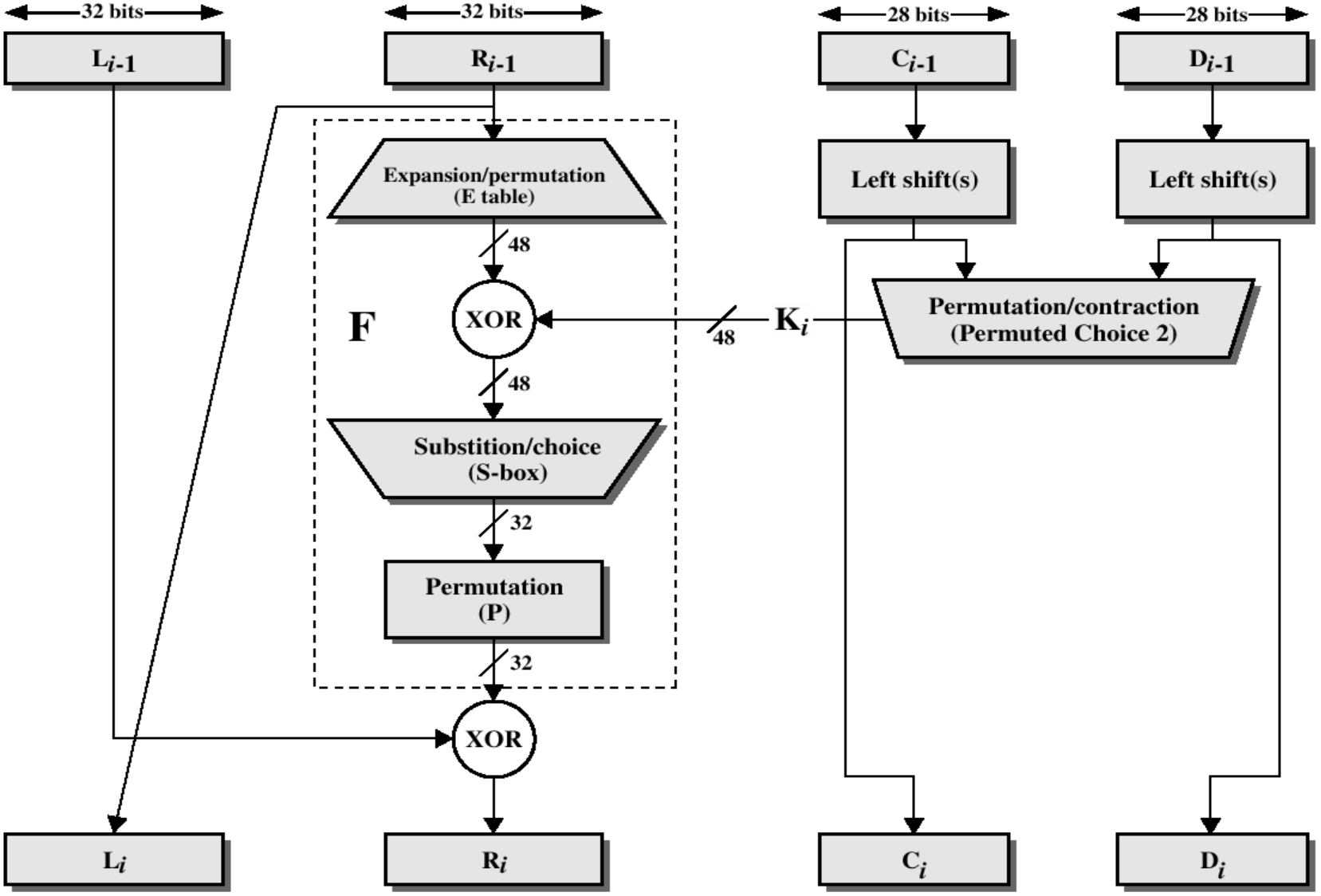


Figure 2.4 Single Round of DES Algorithm

Avalanche Effect

- key desirable property of encryption alg
- DES exhibits strong avalanche
- where a change of **one** input or key bit results in changing approx **half** output bits

Strength of DES – Key Size

- 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
- brute force search looks hard
- recent advances have shown is possible
  - in 1997 on Internet in a few months
  - in 1998 on dedicated hardware (EFF) in a few days
  - in 1999 above combined in 22hrs!
- still must be able to recognize plaintext
- now considering alternatives to DES

Strength of DES – Timing Attacks

- attacks actual implementation of cipher
- use knowledge of consequences of implementation to derive knowledge of some/all subkey bits
- specifically use fact that calculations can take varying times depending on the value of the inputs to it

Strength of DES – Analytic Attacks

- now have several analytic attacks on DES
- these utilise some deep structure of the cipher
  - by gathering information about encryptions
  - can eventually recover some/all of the sub-key bits
  - if necessary then exhaustively search for the rest
- generally these are statistical attacks
- include
  - differential cryptanalysis
  - linear cryptanalysis
  - related key attacks

Differential Cryptanalysis

- one of the most significant recent (public) advances in cryptanalysis
- known in 70's with DES design
- Murphy, Biham & Shamir published 1990
- powerful method to analyse block ciphers
- used to analyse most current block ciphers with varying degrees of success
- DES reasonably resistant to it

Differential Cryptanalysis

- a statistical attack against Feistel ciphers
- uses cipher structure not previously used
- design of S-P networks has output of function *f* influenced by both input & key
- hence cannot trace values back through cipher without knowing values of the key
- Differential Cryptanalysis compares two related pairs of encryptions

Differential Cryptanalysis Compares Pairs of Encryptions

- Differential cryptanalysis is complex
- with a known difference in the input
- searching for a known difference in output

$$\Delta m_{i+1} = m_{i+1} \oplus m'_{i+1}$$

$$= \left[m_{i-1} \oplus \mathrm{f}\left(m_i, K_i \right) \right] \oplus \left[m'_{i-1} \oplus \mathrm{f}\left(m'_i, K_i \right) \right]$$

$$= \Delta m_{i-1} \oplus \left[\mathrm{f}\left(m_i, K_i \right) \oplus \mathrm{f}\left(m'_i, K_i \right) \right]$$

Differential Cryptanalysis

- have some input difference giving some output difference with probability p
- if find instances of some higher probability input / output difference pairs occurring
- can infer subkey that was used in round
- then must iterate process over many rounds

Differential Cryptanalysis

- perform attack by repeatedly encrypting plaintext pairs with known input XOR until obtain desired output XOR
- when found
  - if intermediate rounds match required XOR have a **right pair**
  - if not then have a **wrong pair**
- can then deduce keys values for the rounds
  - right pairs suggest same key bits
  - wrong pairs give random values
- larger numbers of rounds makes it more difficult
- Attack on full DES requires an effort on the order of 2^{47}, requiring 2^{47} chosen plaintexts to be encrypted

Linear Cryptanalysis

- another recent development
- also a statistical method
- based on finding linear approximations to model the transformation of DES
- can attack DES with 2^{47} known plaintexts, still in practise infeasible

Criteria for S-Boxes

- No output of any S-Box is too close to a linear function of the input bits
- Each row of an S-Box includes all 16 possible output bit combinations
- If two inputs to an S-box differ in one bit, the output bits differ in at least two bits
- If two inputs differ is the two middle bits, outputs must differ at least two bits
- Defend against differential analysis and provide good confusion properties

Block Cipher Design Principles

- basic principles still like Feistel in 1970's
- number of rounds
  - more is better, makes exhaustive search best attack
  - 16 rounds: brute force 2^{55}
  - differential analysis: $2^{55.1}$

Block Cipher Design Principles

- function F:
  - provides “confusion”, is nonlinear, avalanche
  - Strict Avalanche Criterion (SAC)
    - Any output bit i should change with p=1/2 when any single input bit j is inverted, for all i,j
    - Applies to both S-Boxes and the overall F function
- key schedule
  - No general rule has been discovered
  - complex subkey creation, key avalanche

Modes of Operation

- block ciphers encrypt fixed size blocks
- eg. DES encrypts 64-bit blocks, with 56-bit key
- need way to use in practise, given usually have arbitrary amount of information to encrypt
- four were defined for DES in ANSI standard **ANSI X3.106-1983 Modes of Use**
  - DES is the basic building block
- have **block** and **stream** modes

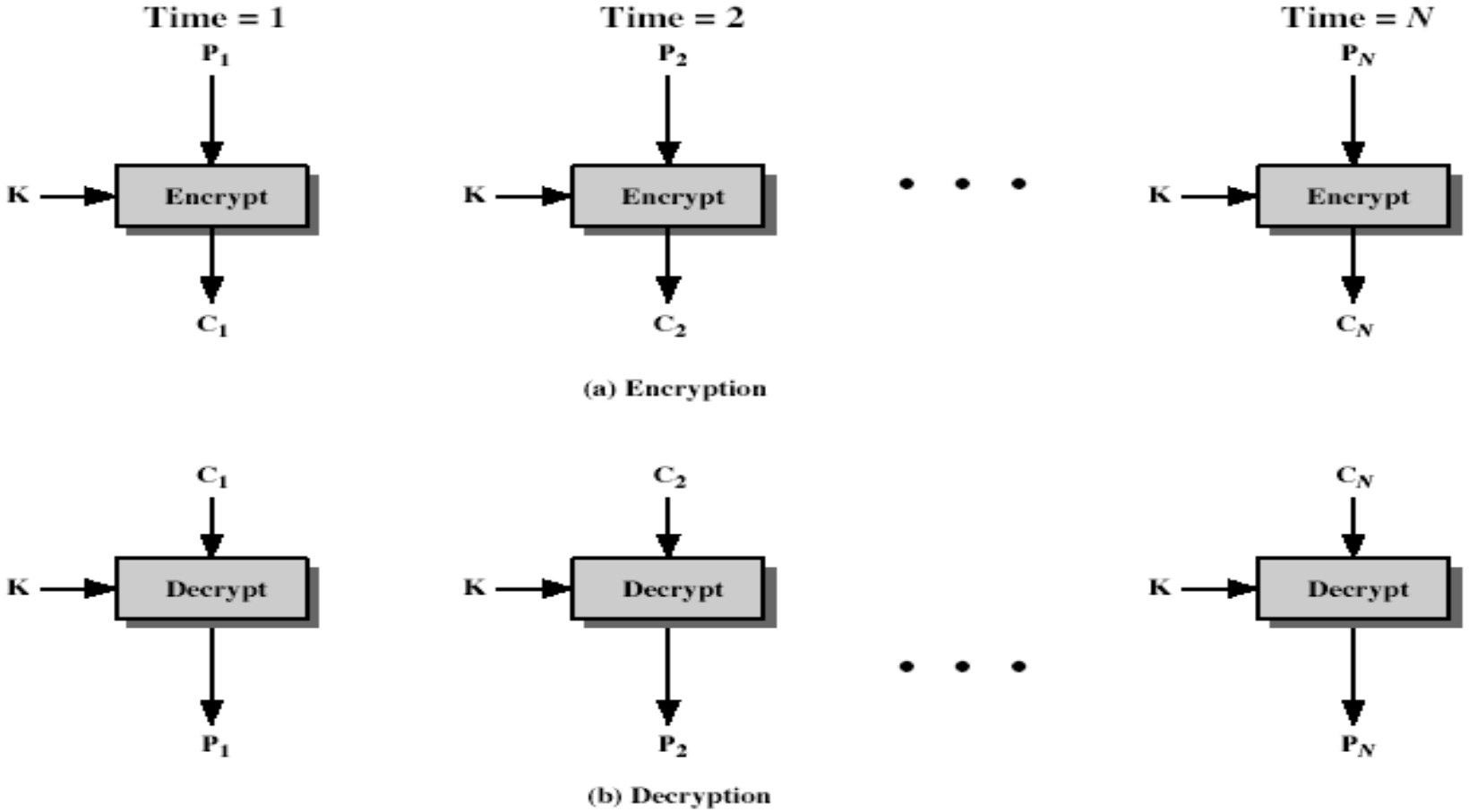
Electronic Codebook Book (ECB)

- message is broken into independent blocks which are encrypted
- each block is a value which is substituted, like a codebook, hence name
  - Each DES is a very complex 64-bit to 64-bit substitution
- each block is encoded independently of the other blocks

$$C_i = DES_{K1}(P_i)$$

- uses: secure transmission of single values
  - Repeated input blocks have same output
  - Not secure for long transmission

Electronic Codebook Book (ECB)



Advantages and Limitations of ECB

- repetitions in message may show in ciphertext
  - if aligned with message block
  - particularly with data such graphics
  - or with messages that change very little, which become a code-book analysis problem
- weakness due to encrypted message blocks being independent
- main use is sending a few blocks of data

Cipher Block Chaining (CBC)

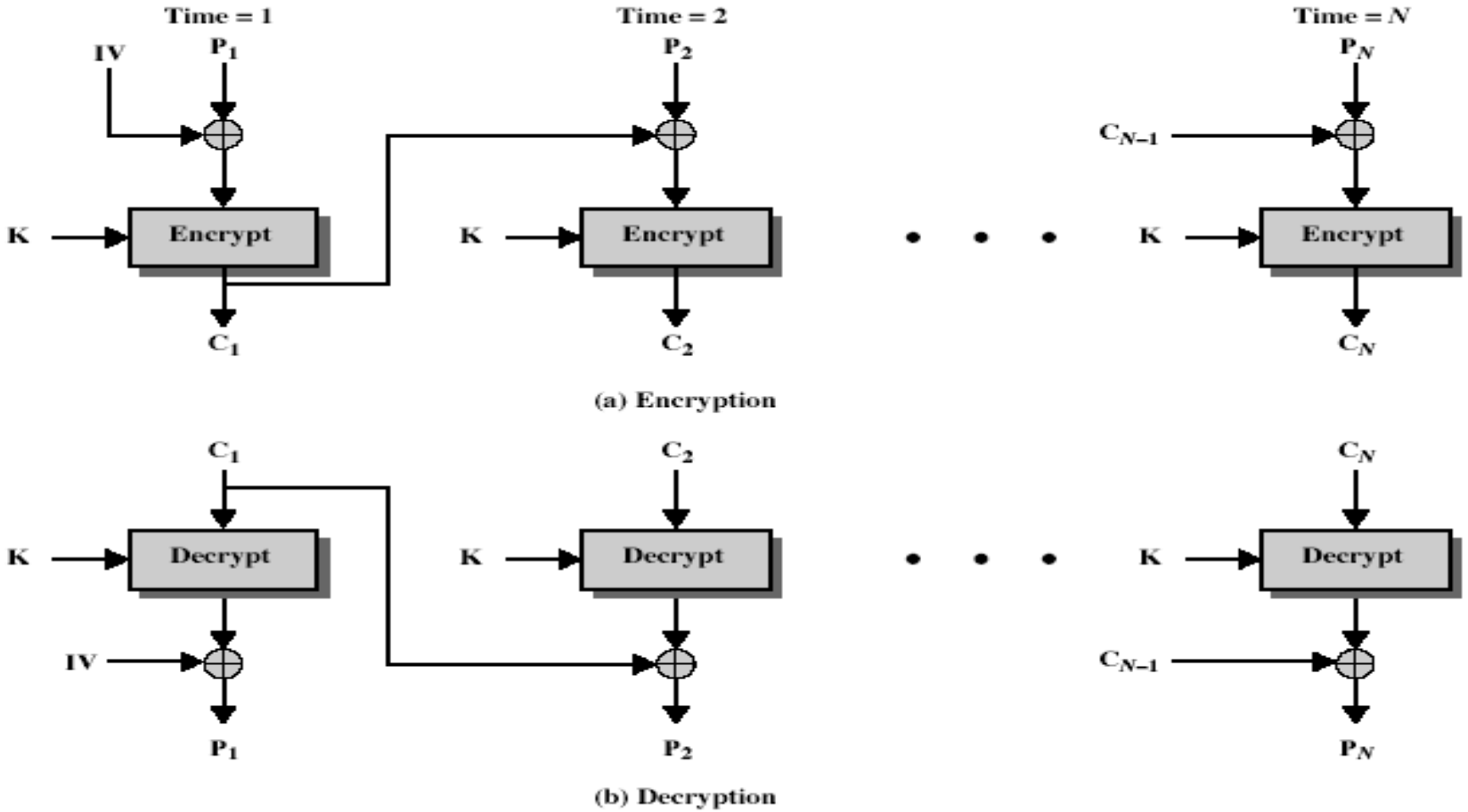
- message is broken into blocks
- but these are linked together in the encryption operation
- each previous cipher blocks is chained with current plaintext block, hence name
- use Initial Vector (IV) to start process

$$C_i = DES_{K1}(P_i \text{ XOR } C_{i-1})$$

$$C_{-1} = IV$$

- uses: bulk data encryption, authentication

Cipher Block Chaining (CBC)



Advantages and Limitations of CBC

- each ciphertext block depends on **all** message blocks
- thus a change in the message affects all ciphertext blocks after the change as well as the original block
- need **Initial Value** (IV) known to sender & receiver
  - however if IV is sent in the clear, an attacker can change bits of the first block, and change IV to compensate
  - hence either IV must be a fixed value (as in EFTPOS) or it must be sent encrypted in ECB mode before rest of message

Cipher FeedBack (CFB)

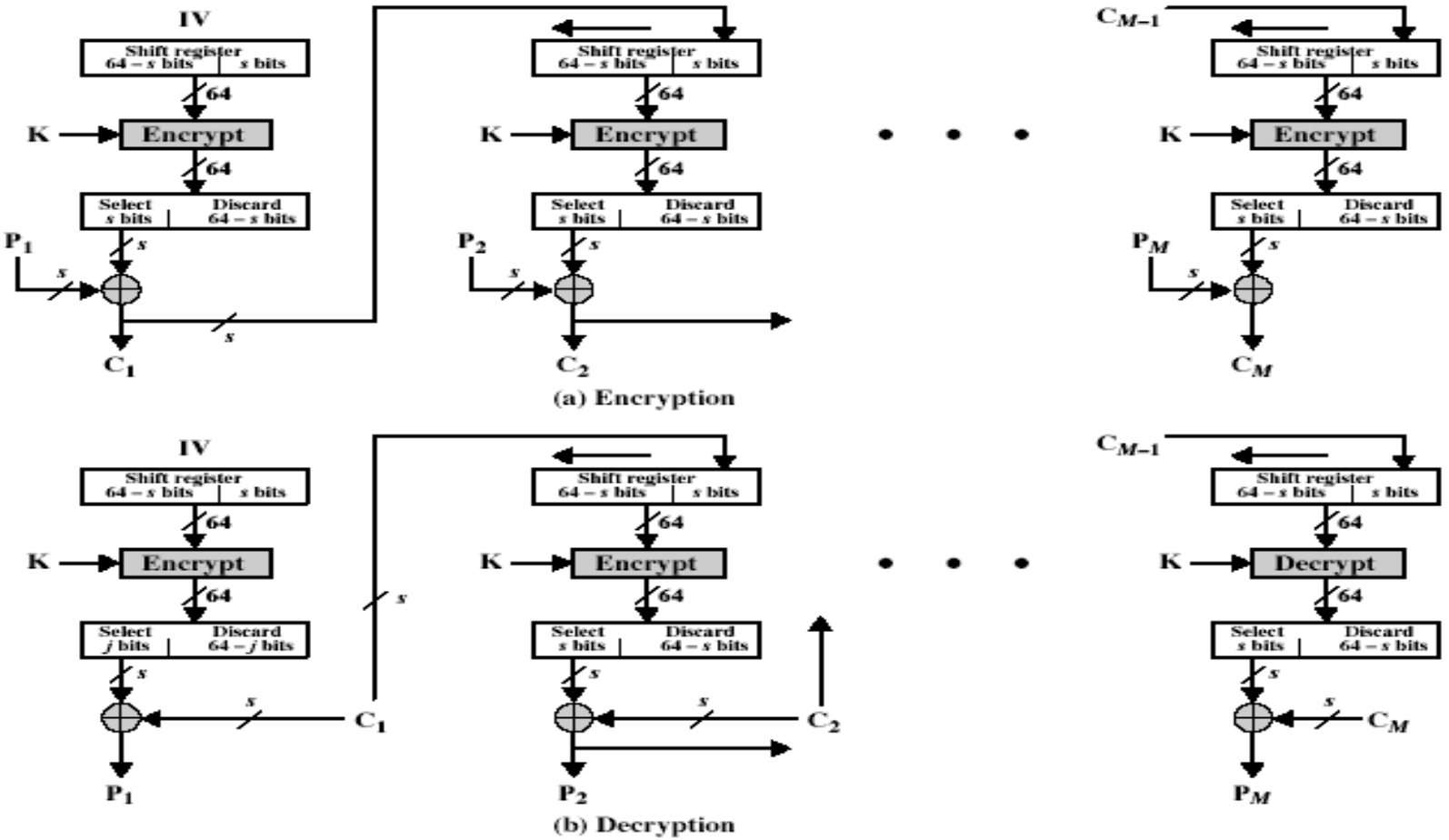
- message is treated as a stream of bits
- added to the output of the block cipher
- result is feed back for next stage (hence name)
- standard allows any number of bit (1,8 or 64 or whatever) to be feed back
  - denoted CFB-1, CFB-8, CFB-64 etc
- is most efficient to use all 64 bits (CFB-64)

$$C_i = P_i \text{ XOR } DES_{K1}(C_{i-1})$$

$$C_{-1} = IV$$

- uses: stream data encryption, authentication

Cipher FeedBack (CFB)



Advantages and Limitations of CFB

- appropriate when data arrives in bits/bytes
- most common stream mode
- note that the block cipher is used in **encryption** mode at **both** ends
- errors propagate for several blocks after the error
  - Must use over a reliable network channel

Output FeedBack (OFB)

- message is treated as a stream of bits
- output of cipher is added to message
- output is then feed back (hence name)
- feedback is independent of message
- can be computed in advance

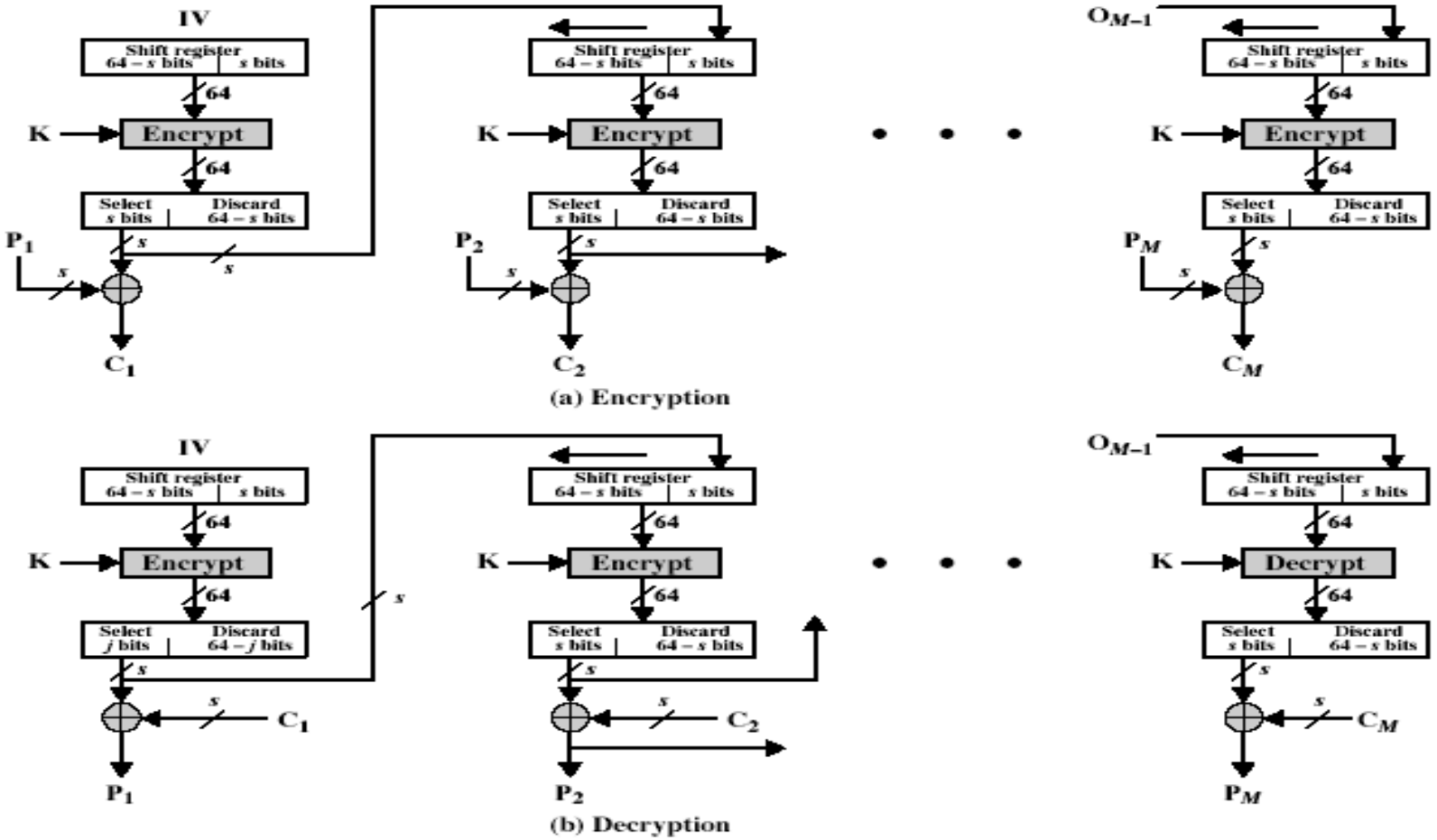
$$C_i = P_i \text{ XOR } O_i$$

$$O_i = DES_{K1}(O_{i-1})$$

$$O_{-1} = IV$$

- uses: stream encryption over noisy channels

Output FeedBack (OFB)



Advantages and Limitations of OFB

- used when error feedback a problem or where need to encryptions before message is available
- superficially similar to CFB
- but feedback is from the output of cipher and is independent of message
  - Errors do not propagate
- sender and receiver must remain in sync, and some recovery method is needed to ensure this occurs
- Because the "random" bits are independent of the message, they must **never** be used more than once
  - otherwise the 2 ciphertexts can be combined, cancelling these bits)

Counter (CTR)

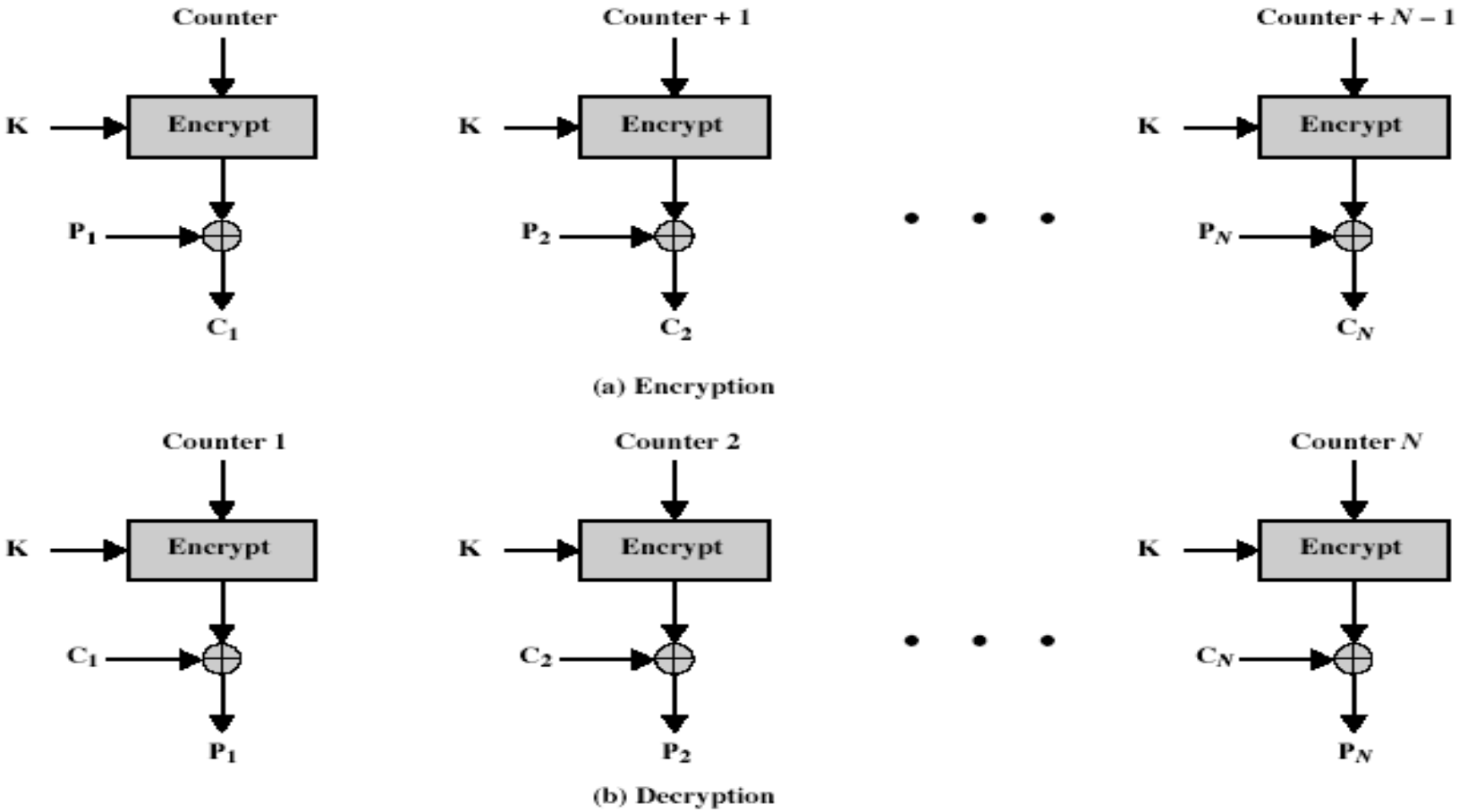
- a “new” mode, though proposed early on
- encrypts counter value rather than any feedback value
- must have a different key & counter value for every plaintext block (never reused)

$$C_i = P_i \text{ XOR } O_i$$

$$O_i = DES_{K1}(i)$$

- uses: high-speed network encryptions

Counter (CTR)


Counter
Counter + 1
Counter + N – 1
K
Encrypt
P1
C1
P2
C2
PN
CN
(a) Encryption
Counter 1
Counter 2
Counter N
(b) Decryption

Advantages and Limitations of CTR

- efficiency
  - can do parallel encryptions
  - in advance of need
  - good for bursty high speed links
- random access to encrypted data blocks
  - Do not have to decode from the beginning
- provable security (good as other modes)
- but must ensure never reuse key/counter values, otherwise could break (cf OFB)

Summary

- have considered:
  - block cipher design principles
  - DES
    - details
    - strength
  - Differential & Linear Cryptanalysis
  - Modes of Operation
    - ECB, CBC, CFB, OFB, CTR