Lecture 3

Encryption I

Suggested Readings:

- Chs 3 & 4 in KPS (recommended)
- Ch 3 in Stinson (optional)

[lecture slides are adapted from previous slides by Prof. Gene Tsudik]

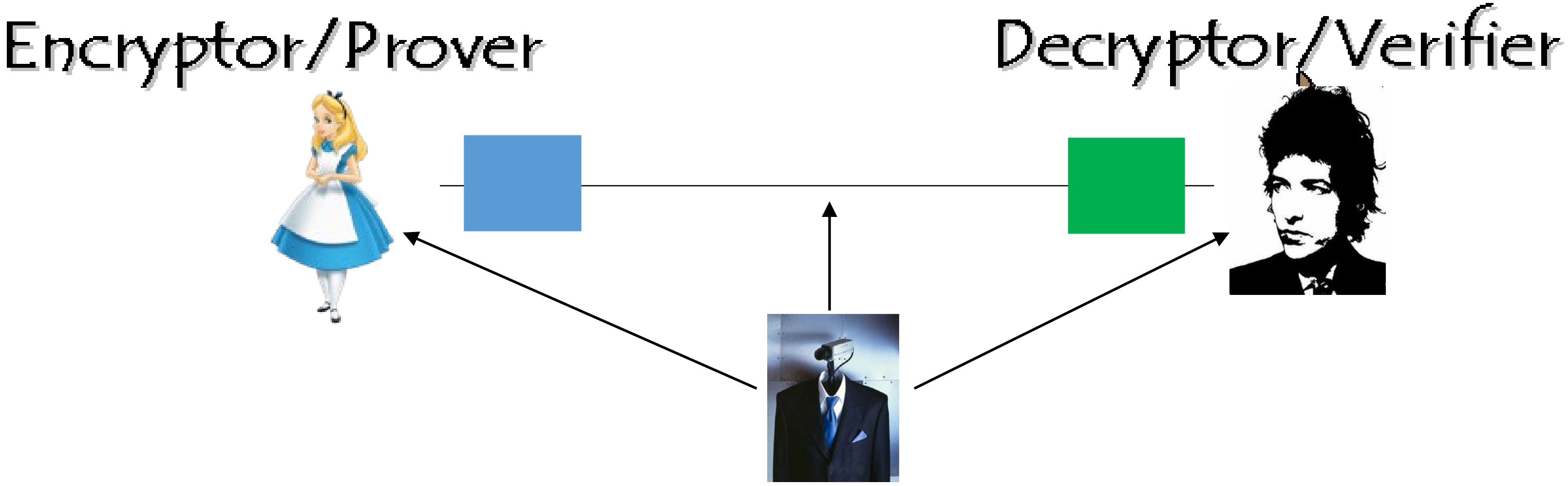
Crypto Basics

Crypto Attacks:

- ciphertext only
- known plaintext
- chosen plaintext
- chosen ciphertext
- brute force

Cryptosystem:

- *P -- plaintext*
- *C -- ciphertext*
- *K -- keyspace*
- *E -- encryption rules*
- *D -- decryption rules*



Encryptor/Prover

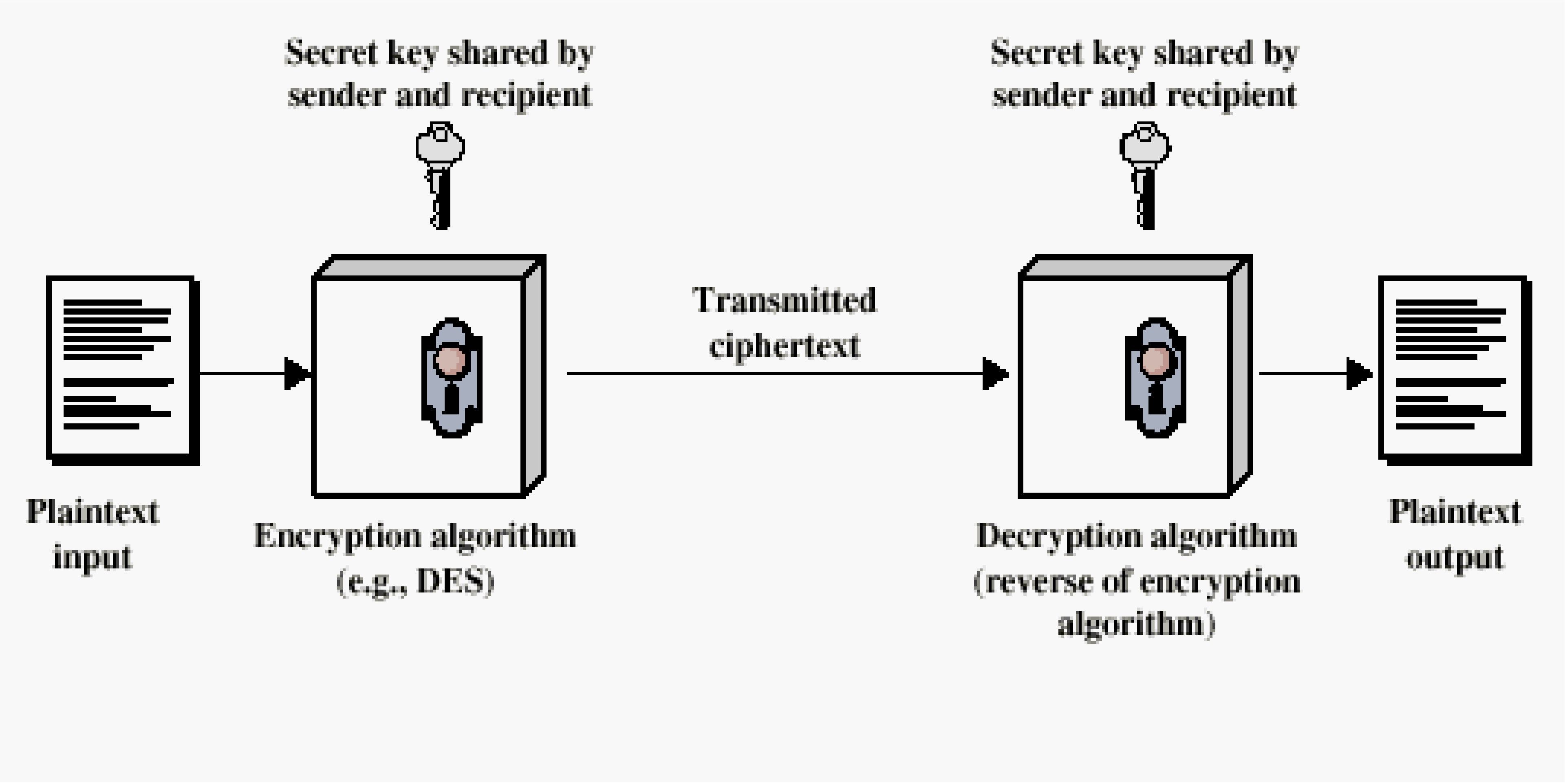
Decryptor/Verifier

Cryptosystems

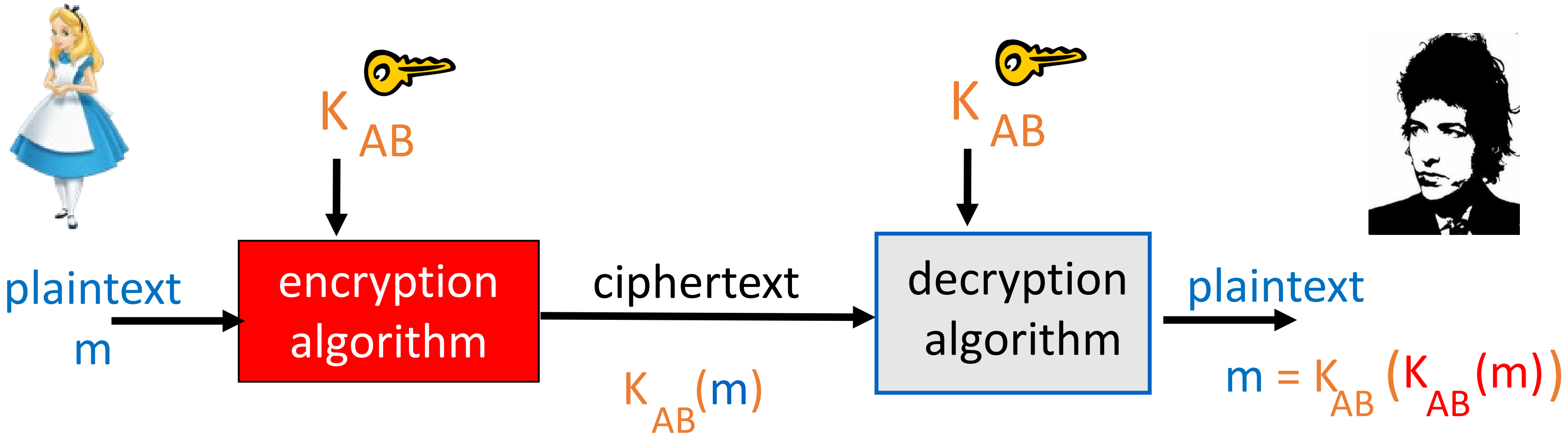
Classified along three dimensions:

- Type of operations used for transforming plaintext into ciphertext
  - Binary arithmetic: shifts, XORs, ANDs, etc.
    - Typical for **conventional/symmetric** encryption
  - Integer arithmetic
    - Typical for **public key/asymmetric** encryption
- Number of keys used
  - Symmetric or conventional (single key used)
  - Asymmetric or public-key (2 keys: 1 to encrypt, 1 to decrypt)
- How plaintext is processed:
  - One bit at a time – “stream cipher”
  - A block of bits – “block cipher”

Conventional/Symmetric Encryption Principles


Secret key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Conventional (Symmetric) Cryptography



K_{AB}

K_{AB}

plaintext m

encryption algorithm

ciphertext

$K_{AB}(m)$

decryption algorithm

plaintext

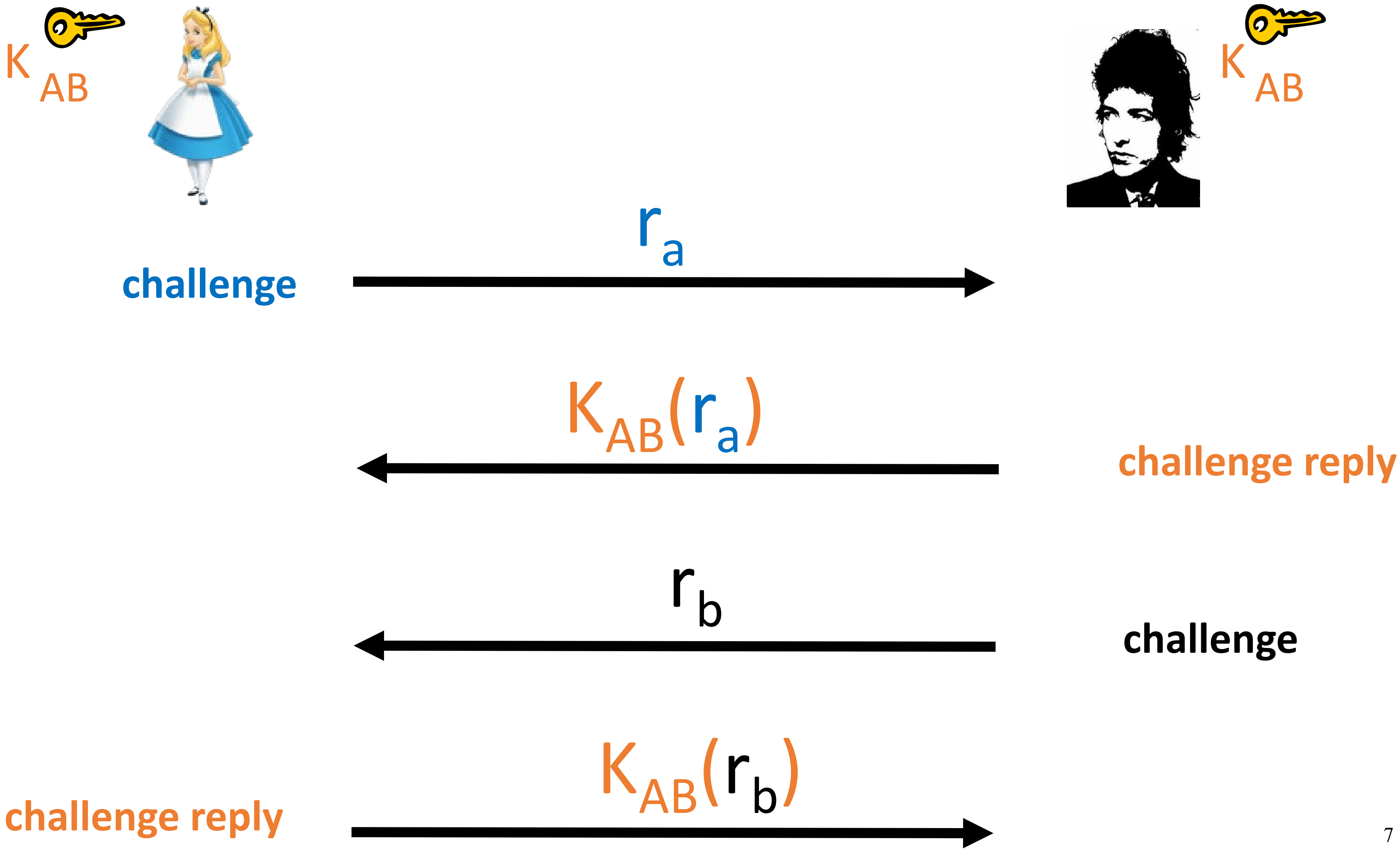
$m = K_{AB}(K_{AB}(m))$

- Alice and Bob share a key K_{AB} which they somehow agree upon (how?)
  - key distribution / key management problem
  - ciphertext is roughly as long as plaintext
  - examples: Substitution, Vernam OTP, DES, AES

Uses of Conventional/Symmetric Cryptography

- Message transmission (confidentiality):
  - Communication over insecure channels
- Secure storage: crypt on Unix
- Strong authentication: proving knowledge of a secret without revealing it:

Challenge-Response Authentication Example



K_{AB}

K_{AB}

challenge r_a

$K_{AB}(r_a)$ challenge reply

r_b challenge

challenge reply $K_{AB}(r_b)$

Uses of Conventional/Symmetric Cryptography

- Message transmission (confidentiality):
  - Communication over insecure channels
- Secure storage: crypt on Unix
- Strong authentication: proving knowledge of a secret without revealing it:
  - Eve can obtain chosen <plaintext, ciphertext> pair
  - Challenge should be chosen from a large pool
- Integrity checking: fixed-length checksum for message via secret key cryptography
  - Send MAC along with the message MAC=H(K, m)

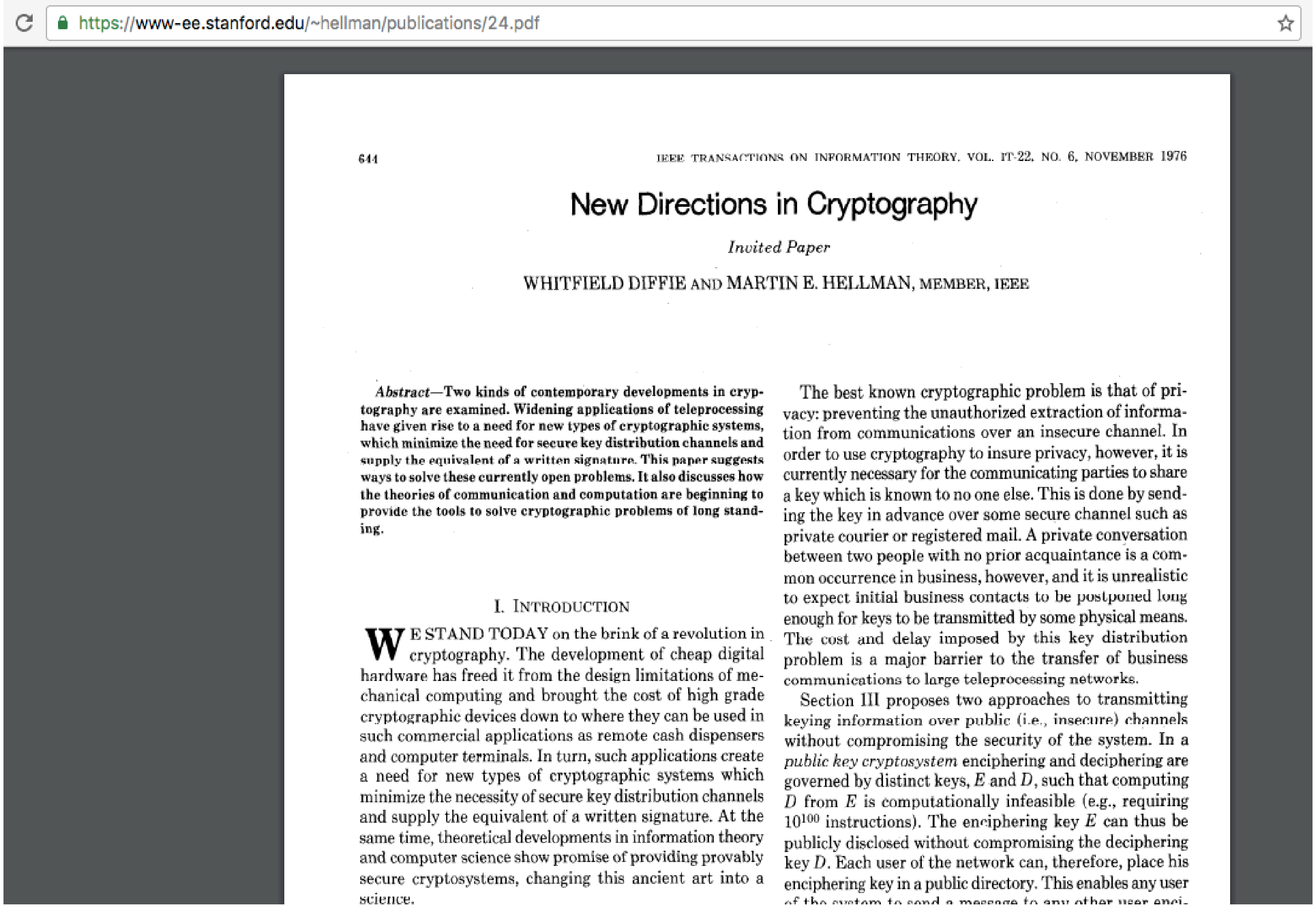
Conventional/Symmetric Cryptography

- Advantages
  - high data throughput
  - relatively short key size
  - primitives to construct various cryptographic mechanisms
- Disadvantages
  - key must remain secret at both ends
  - key must be distributed securely and efficiently
  - relatively short key lifetime

Public Key (Asymmetric) Cryptography

- Asymmetric cryptography
- Invented in 1974-1978 (Diffie-Hellman, Rivest-Shamir-Adleman)
  - Both win Turing awards (2002, 2015)!
- Two keys: private (SK), public (PK)
  - Encryption: with public key;
  - Decryption: with private key
  - Digital Signatures: Signing by private key; Verification by public key. i.e., “encrypt” message digest/hash -- $h(m)$ -- with private key
    - Authorship (authentication)
    - Integrity: Similar to MAC
    - Non-repudiation: can’t do with secret/symmetric key cryptography
- Much **slower** (~1000x) than conventional cryptography
    - Often used together with conventional cryptography, e.g., to encrypt session keys

Genesis of Public Key Cryptography: Diffie- Hellman Paper



https://www-ee.stanford.edu/~hellman/publications/24.pdf

644 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-22, NO. 6, NOVEMBER 1976

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

Abstract—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.

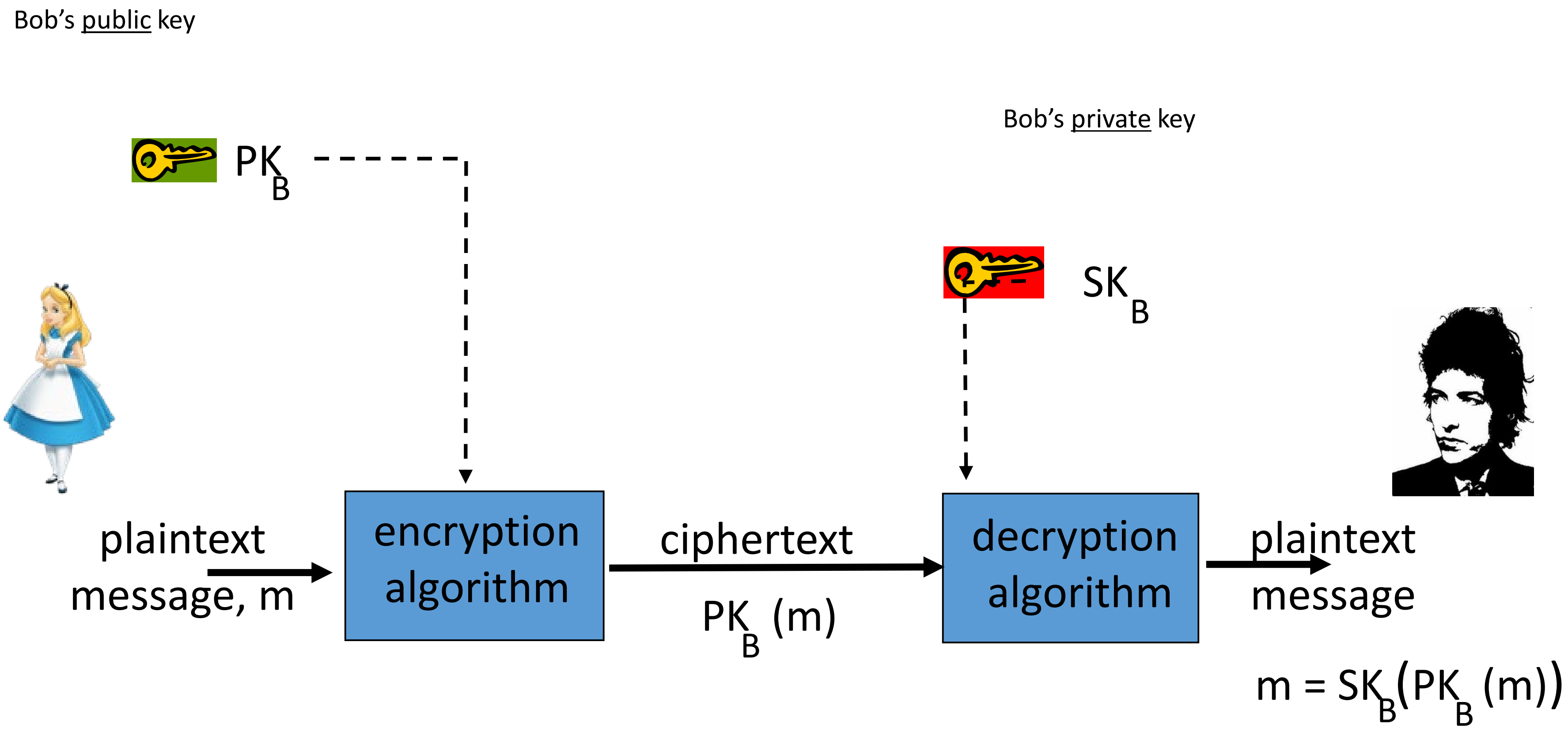
I. Introduction

WE STAND TODAY on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals. In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Section III proposes two approaches to transmitting keying information over public (i.e., insecure) channels without compromising the security of the system. In a *public key cryptosystem* enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infeasible (e.g., requiring 10^{100} instructions). The enciphering key E can thus be publicly disclosed without compromising the deciphering key D. Each user of the network can, therefore, place his enciphering key in a public directory. This enables any user of the system to send a message to any other user enci-

Public Key Cryptography



Bob's public key

PK_B

Bob's private key

SK_B

plaintext message, m

encryption algorithm

ciphertext

$PK_B(m)$

decryption algorithm

plaintext message

$m = SK_B(PK_B(m))$

Uses of Public Key Cryptography

- Data transmission (confidentiality):
  - Alice encrypts m_a using PK$_B$, Bob decrypts it to obtain m_a using SK$_b$.
- Secure Storage: encrypt with own public key, later decrypt with own private key
- Authentication:
  - No need to store secret**s**, only need *public* keys.
  - Secret/symmetric key cryptography: need to share *secret* key for every person one communicates with
- Digital Signatures (authentication, integrity, non-repudiation)

Public Key Cryptography

- Advantages
  - only the private key must be kept secret
  - relatively long life time of the key
  - more security services
  - relatively efficient digital signatures mechanisms
- Disadvantages
  - low data throughput
  - much larger key sizes
  - distribution/revocation of public keys
  - security based on conjectured hardness of certain computational problems

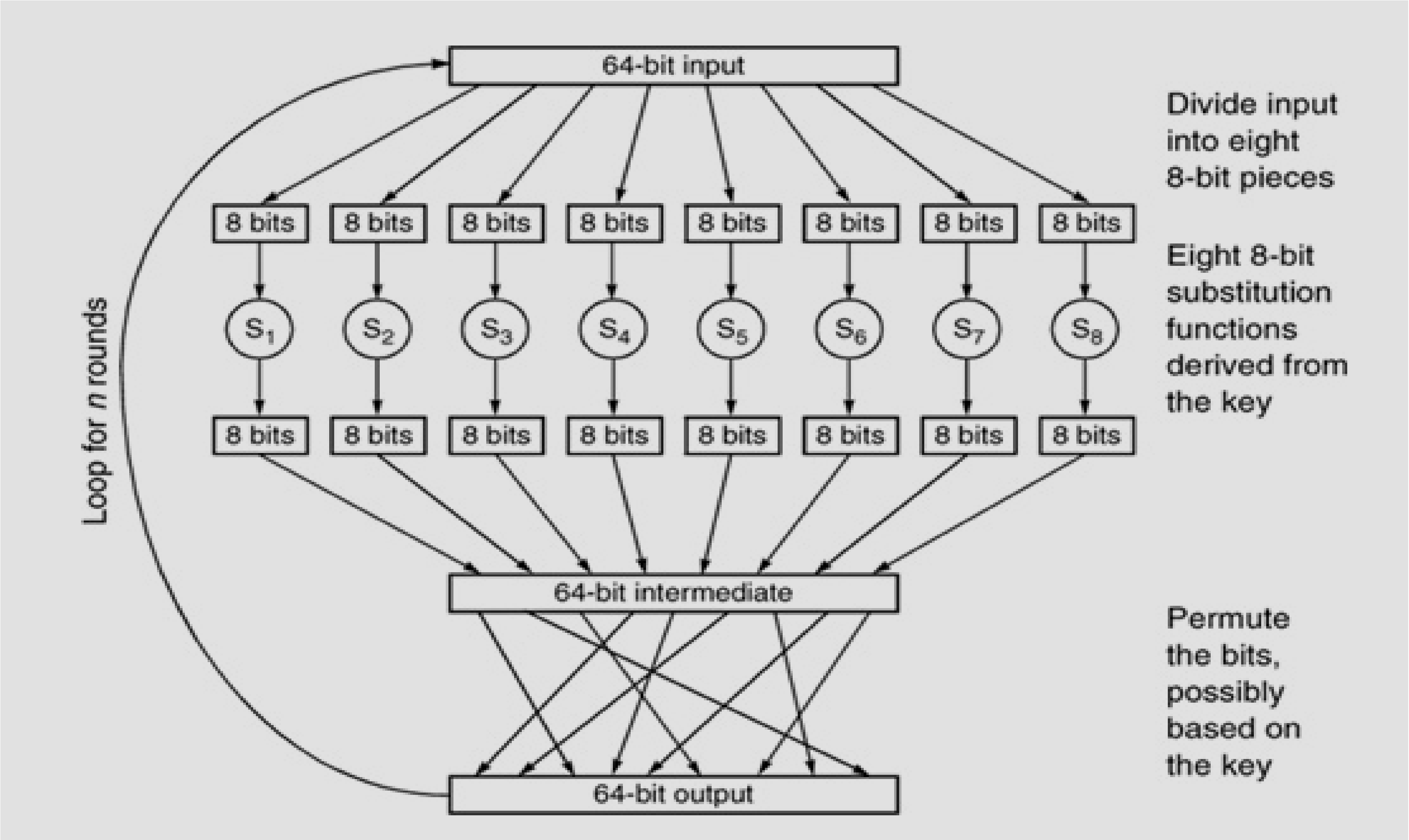
Comparison Summary

- Public key
  - encryption, signatures (esp., non-repudiation), and key management
- Conventional/symmetric
  - encryption and some data integrity applications
- Key sizes
  - Keys in public key crypto must be larger (e.g., 2048 bits for RSA) than those in conventional crypto (e.g., 112 bits for 3-DES or 256 bits for AES)
    - most attacks on “good” conventional cryptosystems are exhaustive key search (brute force)
    - public key cryptosystems are subject to “short-cut” attacks (e.g., factoring large numbers in RSA)

“Modern” Block Ciphers

Data Encryption Standard (DES)

Generic Example of Block Encryption


64-bit input
8 bits
S_1
S_2
S_3
S_4
S_5
S_6
S_7
S_8
64-bit intermediate
64-bit output
Loop for *n* rounds
Divide input into eight 8-bit pieces
Eight 8-bit substitution functions derived from the key
Permute the bits, possibly based on the key

Feistel Cipher Structure

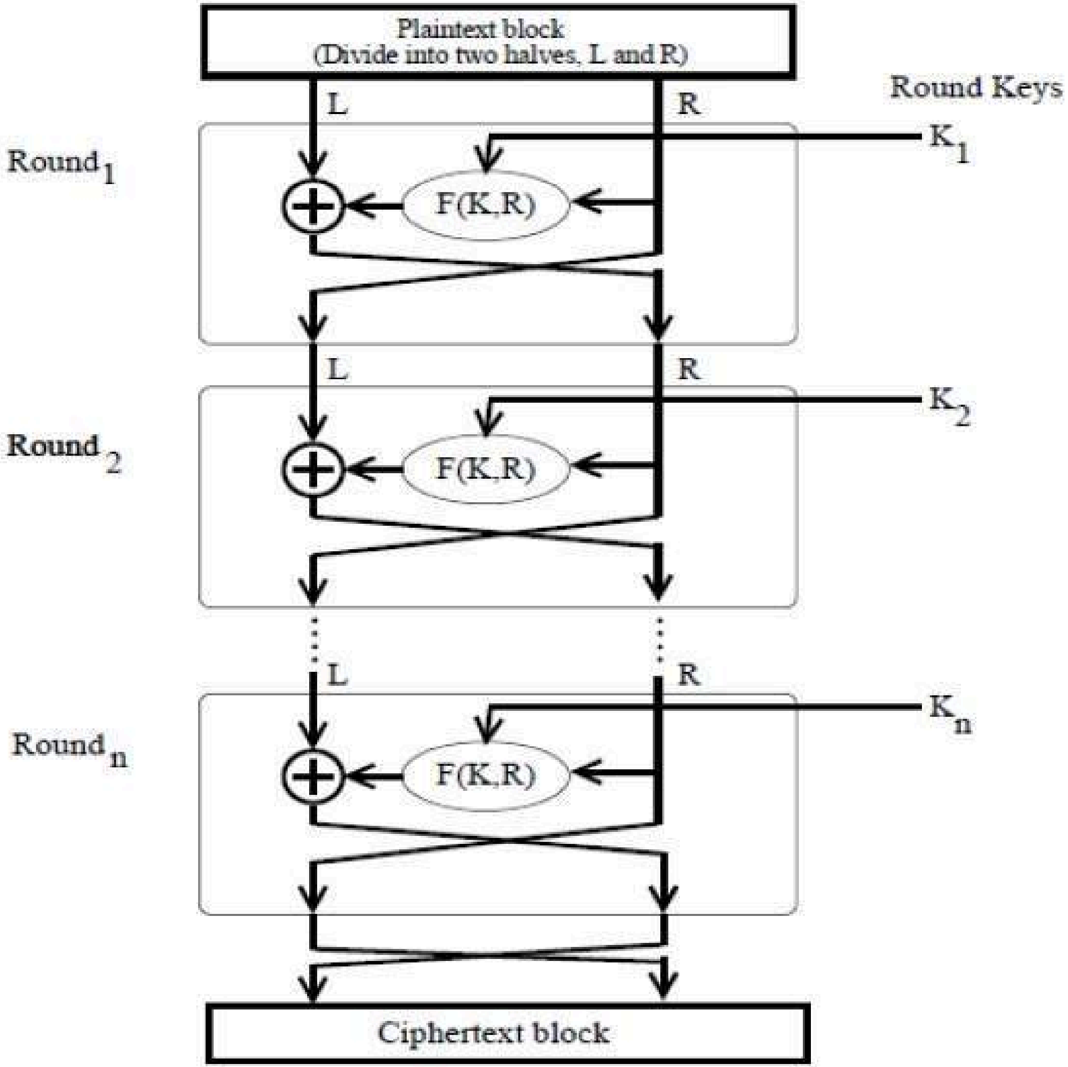
- Virtually all conventional block encryption algorithms, including DES, have a structure first described by Horst Feistel of IBM in 1973

- Specific realization of a Feistel Network depends on the choice of the following parameters and features:

Feistel Cipher Structure

- **Block Size:** larger block sizes mean greater security
- **Key Size:** larger key size means greater security
- **Number of Rounds:** multiple rounds offer increasing security
- **Subkey Generation Algorithm:** greater complexity leads to greater difficulty of cryptanalysis

Classic Feistel Network



“Round Keys” are generated from original key via subkey generation algorithm

Block Ciphers

- **Originated with early 1970's IBM effort to develop banking security systems**
- **First result was Lucifer, most common variant has 128-bit key and block size**
- **Was not secure in any of its variants**
- **Called a Feistel or product cipher**
- **F()-function is a simple transformation, does not have to be reversible**
- **Each step is called a round; the more rounds, the greater the security (to a point)**
- **Most famous example of this design is DES**

Conventional Encryption Standard

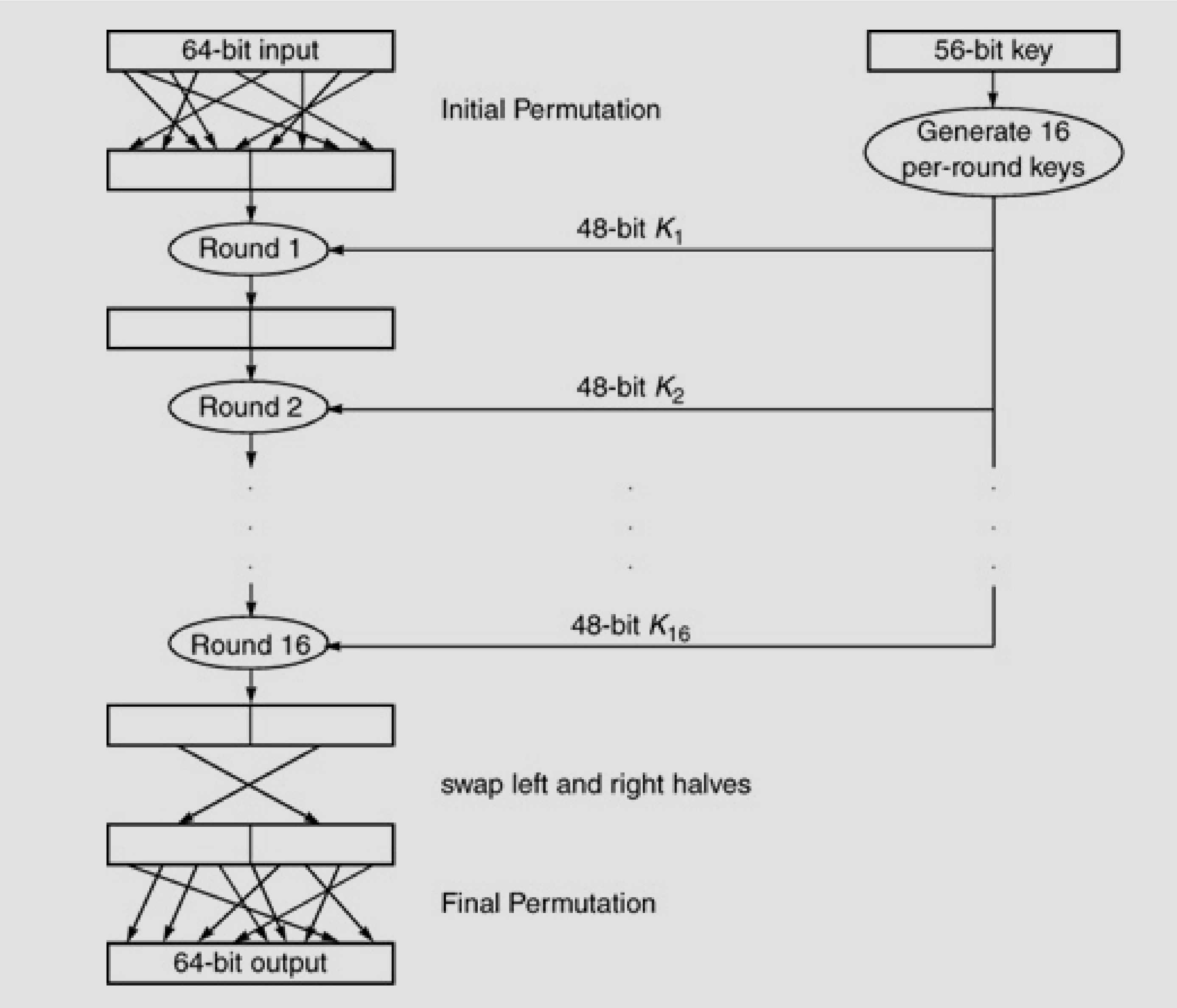
Data Encryption Standard (DES)

- Most widely used encryption method in 1970s/80s/90s
  - AES took over in early 2000s
- Block cipher (in native ECB mode)
- Plaintext processed in 64-bit blocks
- Key is 56 bits

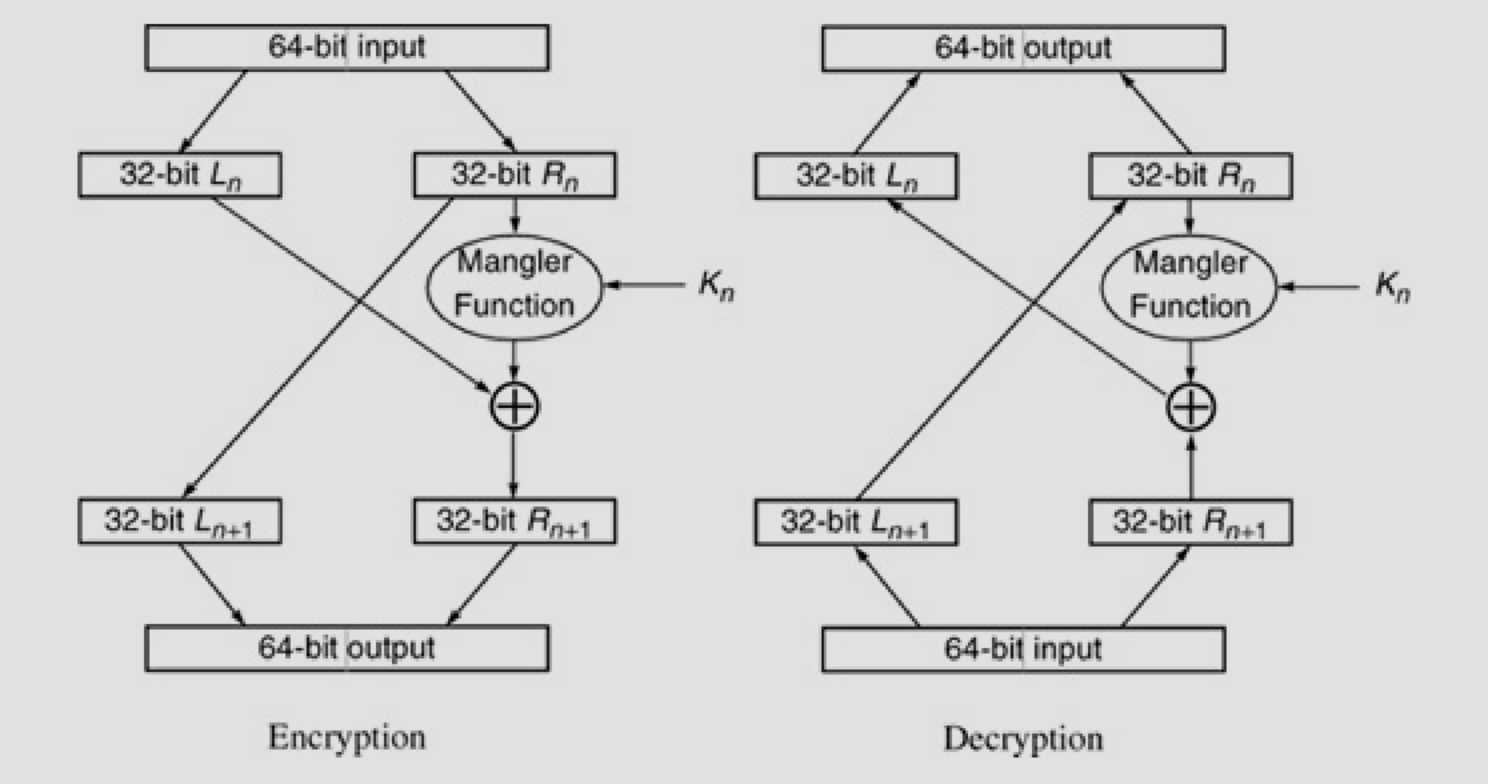
Data Encryption Standard (DES)

- 64 bit input block
- 64 bit output block
- 16 rounds
- 64 (effective 56) bit key
- Key schedule computed at startup
- Aimed at bulk data
- > 16 rounds does not help
- Other S-boxes usually hurt …

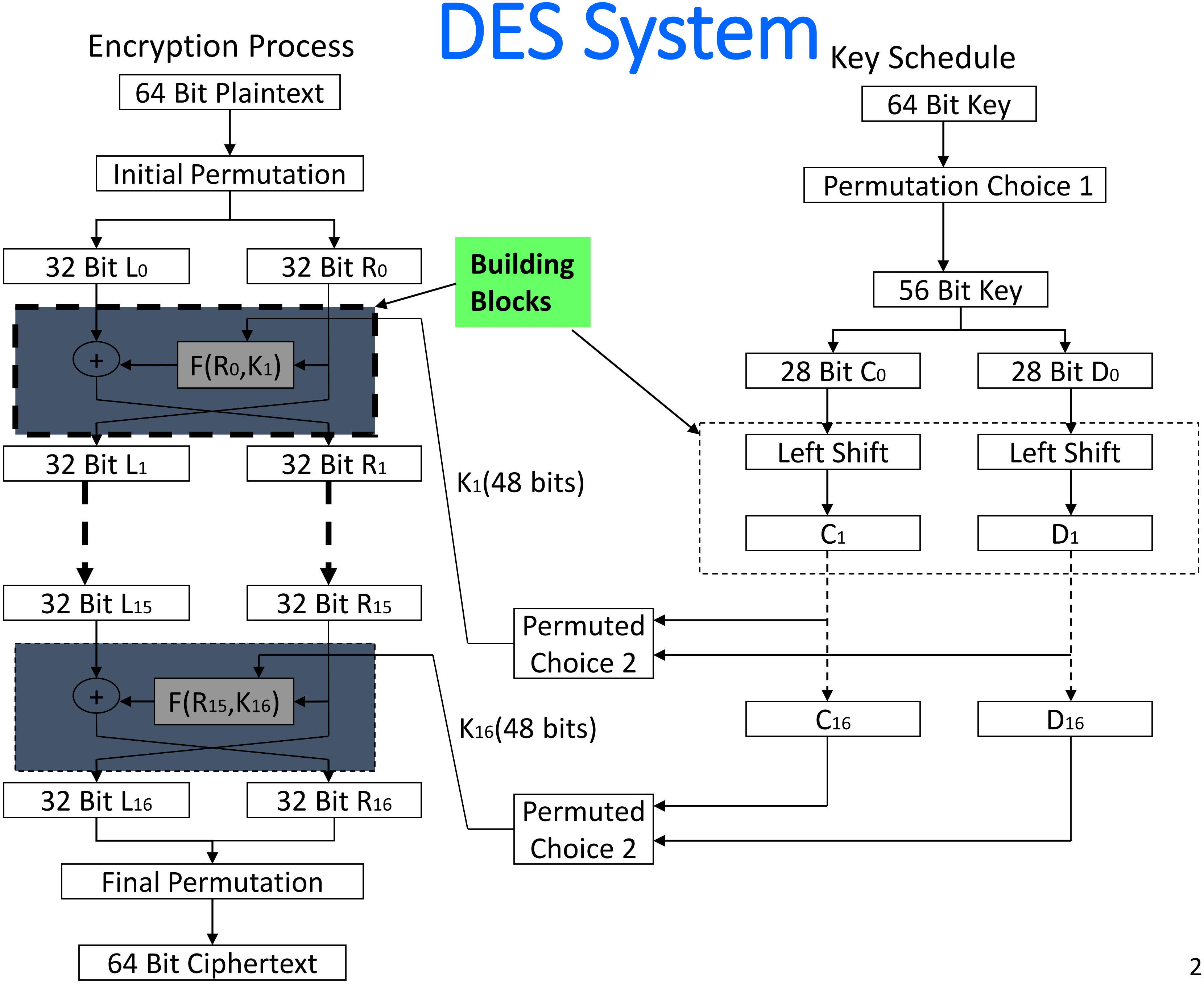
Basic Structure of DES



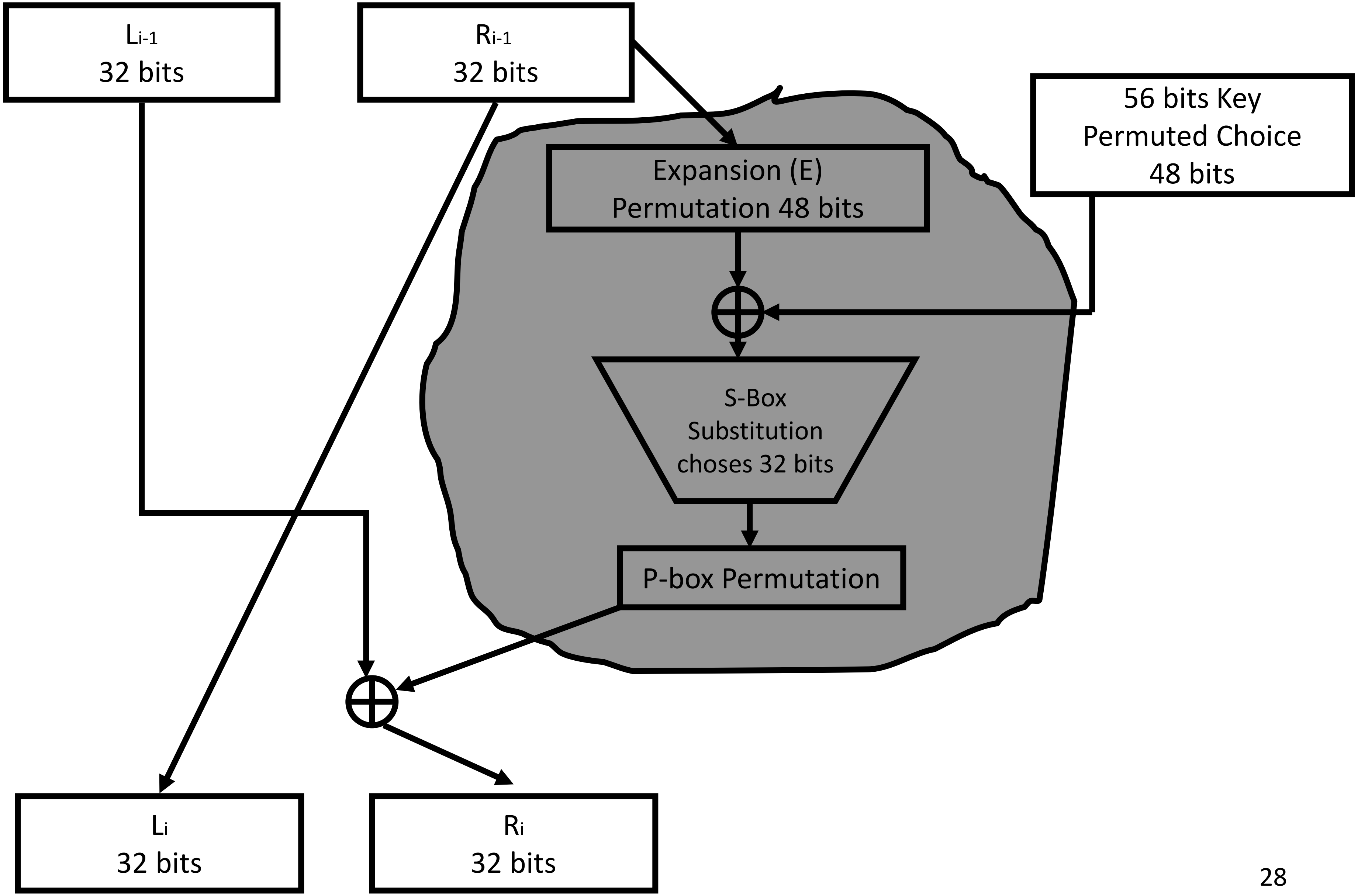
Encryption vs Decryption in DES



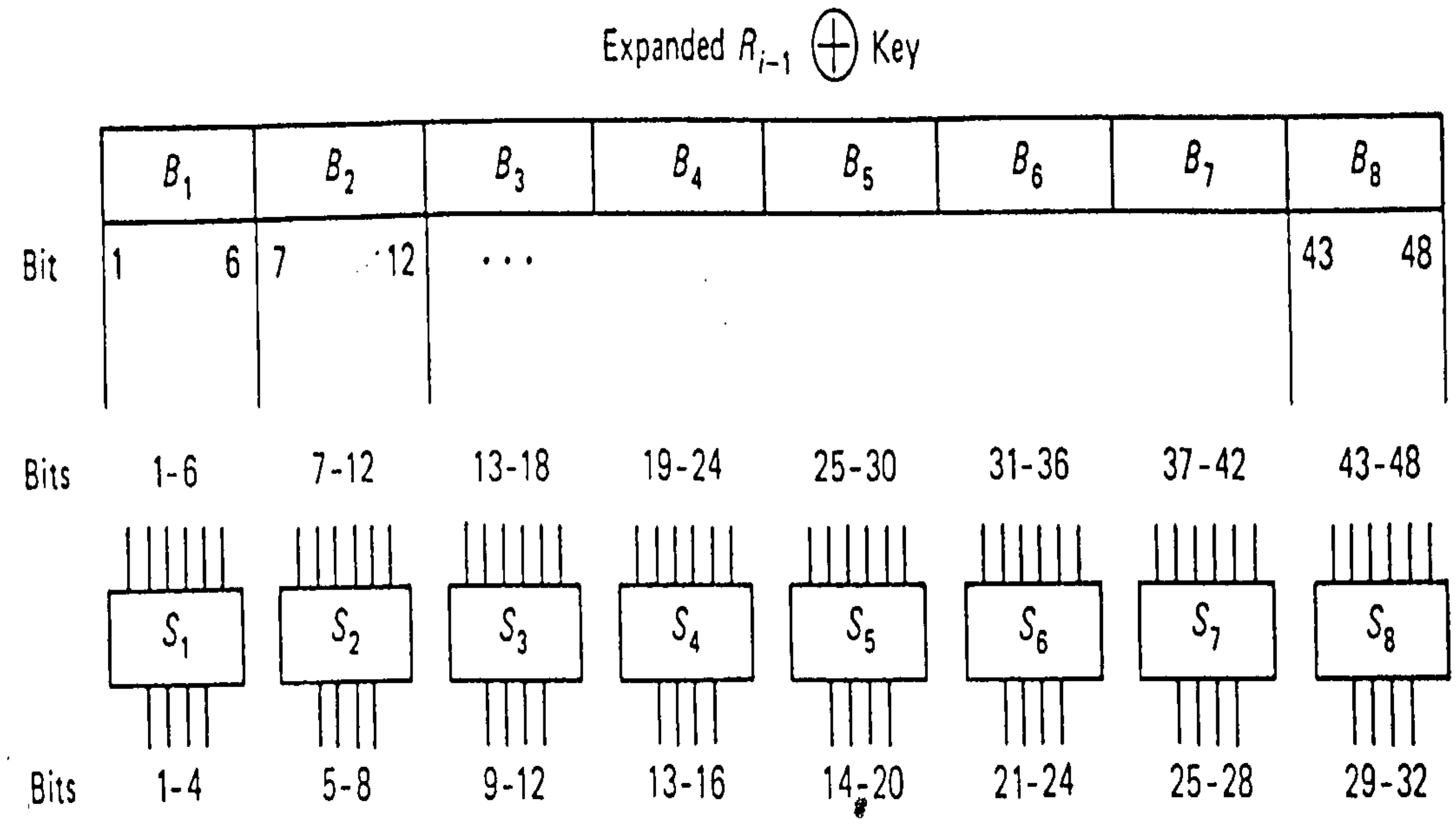
DES System


Encryption Process
64 Bit Plaintext
Initial Permutation
32 Bit L0
32 Bit R0
+
F(R0,K1)
32 Bit L1
32 Bit R1
32 Bit L15
32 Bit R15
+
F(R15,K16)
32 Bit L16
32 Bit R16
Final Permutation
64 Bit Ciphertext
Building Blocks
K1(48 bits)
K16(48 bits)
Permuted Choice 2
Permuted Choice 2
Key Schedule
64 Bit Key
Permutation Choice 1
56 Bit Key
28 Bit C0
28 Bit D0
Left Shift
Left Shift
C1
D1
C16
D16

Function F

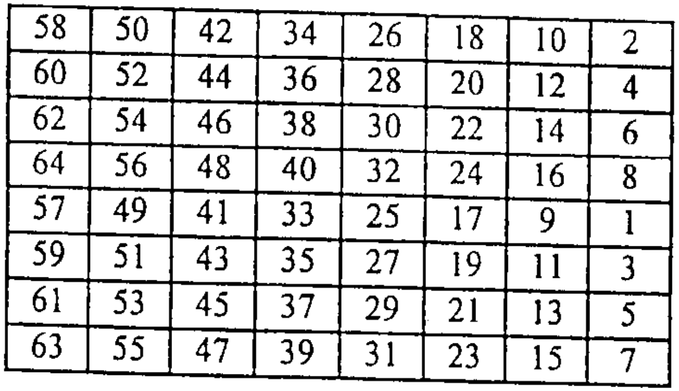

L_{i-1}
32 bits
R_{i-1}
32 bits
56 bits Key
Permuted Choice
48 bits
Expansion (E)
Permutation 48 bits
S-Box
Substitution
choses 32 bits
P-box Permutation
L_i
32 bits
R_i
32 bits

DES Substitution Boxes Operation



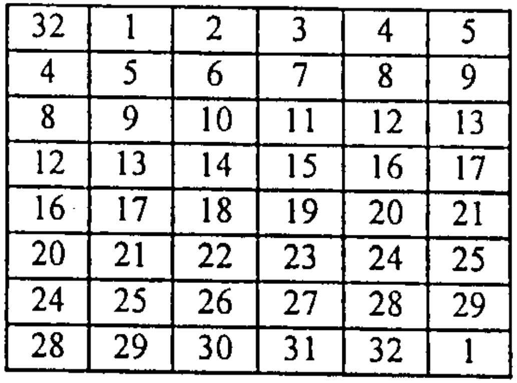
Operation Tables of DES (IP, IP^{-1}, E and P)

Initial Pemutation (IP)



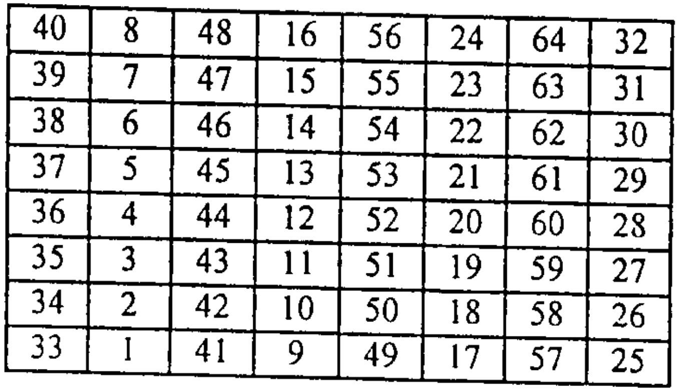
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Bit-Selection Table E



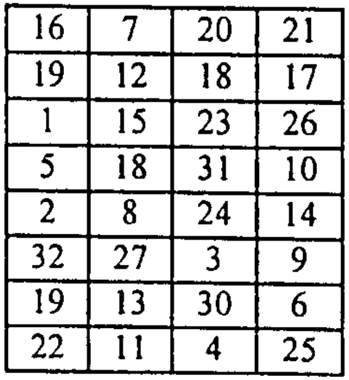
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Inverse Initial Pemutation (IP^{-1})



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

Permutation P



16	7	20	21
19	12	18	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