CSC 416 – COMPUTER SECURITY AND CRYPTOGRAPHY

DR. S. A. ONASHOGA

DEPT OF COMPUTER SCIENCE

FEDERAL UNIVERSITY OF AGRICULTURE, ABEOKUTA

Review

- Overview of Cryptography
- Types of Cryptosystems
- Classical Symmetric Cipher
  - Substitution Cipher
  - Transposition Cipher
  - Product Cipher
- Modern Symmetric Ciphers (DES)

Basic Terminology

- **plaintext** - the original message
- **ciphertext** - the coded message
- **cipher** - algorithm for transforming plaintext to ciphertext
- **key** - info used in cipher known only to sender/receiver
- **encipher (encrypt)** - converting plaintext to ciphertext
- **decipher (decrypt)** - recovering ciphertext from plaintext
- **cryptography** - study of encryption principles/methods
- **cryptanalysis (codebreaking)** - the study of principles/ methods of deciphering ciphertext *without* knowing key
- **cryptology** - the field of both cryptography and cryptanalysis

Private-Key Cryptography

- **Private/secret/single key** cryptography uses **one** key
- Shared by both sender and receiver
- If this key is disclosed communications are compromised
- Also is **symmetric**, parties are equal
- Hence does not protect sender from receiver forging a message & claiming is sent by sender

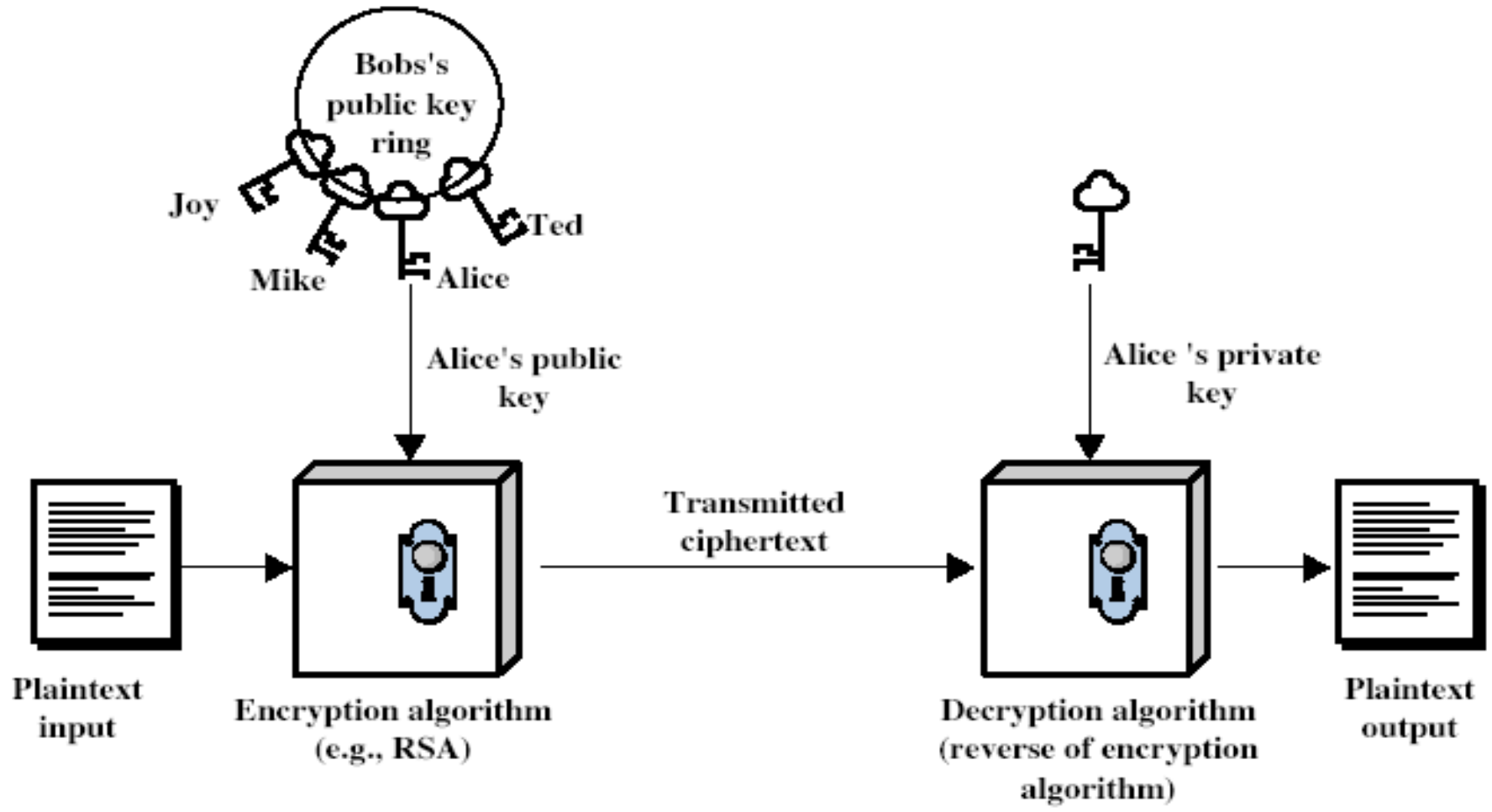
Public-Key Cryptography

- Probably most significant advance in the 3000 year history of cryptography
- Uses **two** keys – a public & a private key
- **Asymmetric** since parties are **not** equal
- Uses clever application of number theoretic concepts to function
- Complements **rather than** replaces private key crypto

Public-Key Cryptography

- **Public-key/two-key/asymmetric** cryptography involves the use of **two** keys:
  - a **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
  - a **private-key**, known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures**
- **Asymmetric** because
  - those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures

Public-Key Cryptography



Public-Key Characteristics

- Public-Key algorithms rely on two keys with the characteristics that it is:
  - computationally infeasible to find decryption key knowing only algorithm & encryption key
  - computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
  - either of the two related keys can be used for encryption, with the other used for decryption (in some schemes)

Public-Key Cryptosystems

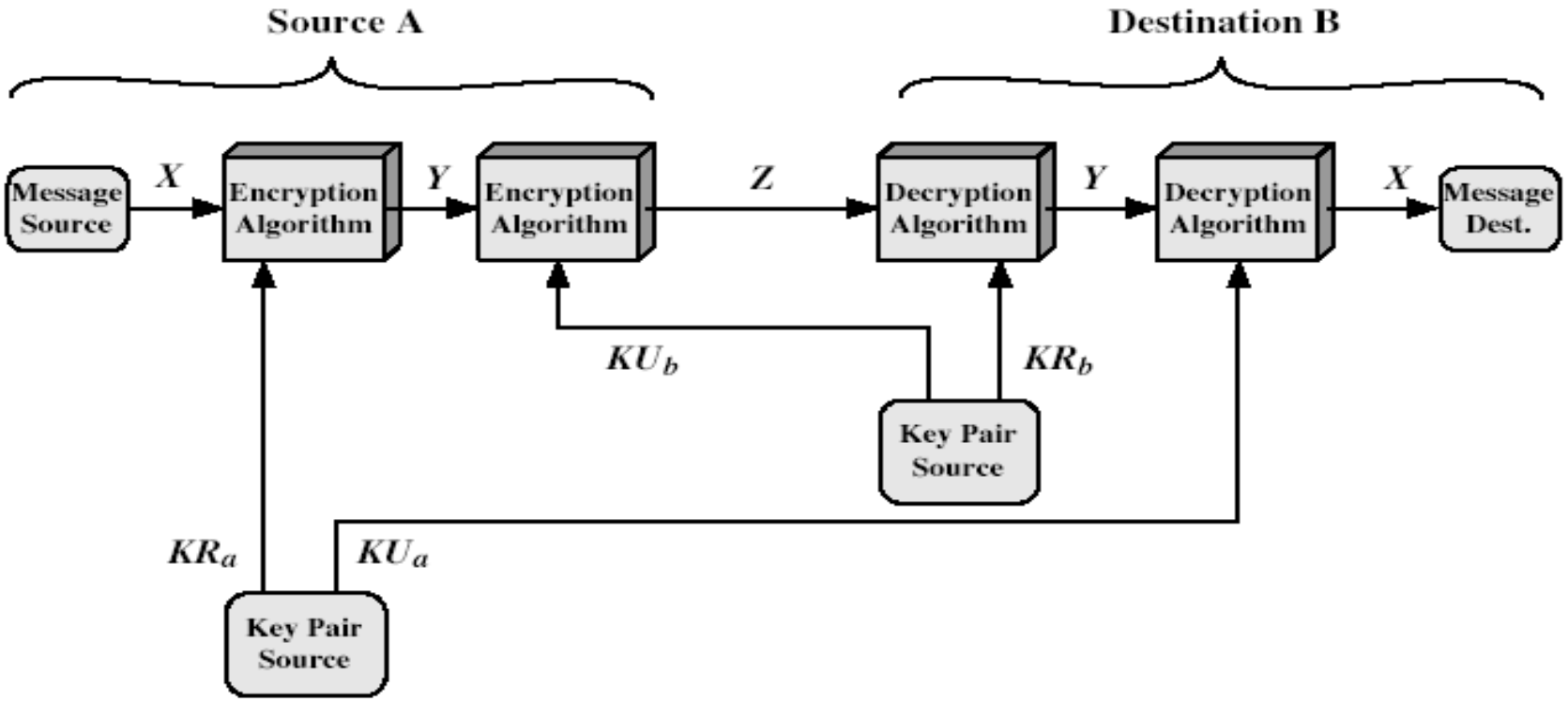


Figure 9.4 Public-Key Cryptosystem: Secrecy and Authentication

- Two major applications:
  - **encryption/decryption** (provide secrecy)
  - **digital signatures** (provide authentication)

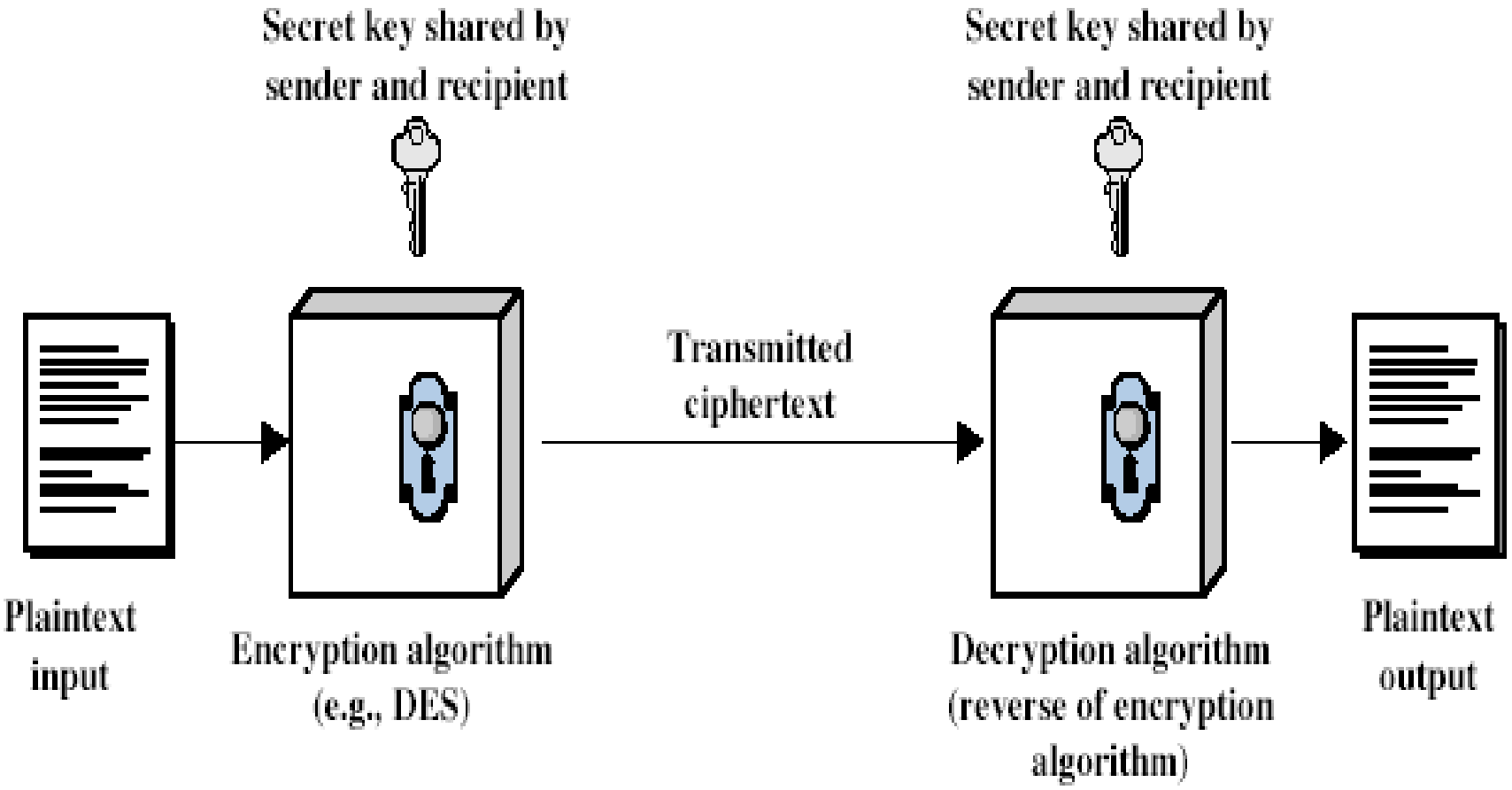
Classification of Cryptography

- Number of keys used
  - Hash functions: no key
  - Secret key cryptography: one key
  - Public key cryptography: two keys - public, private
- Type of encryption operations used
  - substitution / transposition / product
- Way in which plaintext is processed
  - block / stream

Cryptanalysis Scheme

- Ciphertext only:
  - Exhaustive search until "recognizable plaintext"
  - Need enough ciphertext
- Known plaintext:
  - Secret may be revealed (by spy, time), thus <ciphertext, plaintext> pair is obtained
  - Great for monoalphabetic ciphers
- Chosen plaintext:
  - Choose text, get encrypted
  - Useful if limited set of messages

Symmetric Cipher Model



Requirements

- Two requirements for secure use of symmetric encryption:
  - a strong encryption algorithm
  - a secret key known only to sender / receiver

  $Y = E_K(X)$

  $X = D_K(Y)$
- Assume encryption algorithm is known
- Implies a secure channel to distribute key

Classical Substitution Ciphers

- Letters of plaintext are replaced by other letters or by numbers or symbols
- Plaintext is viewed as a sequence of bits, then substitution replaces plaintext bit patterns with ciphertext bit patterns

Caesar Cipher

- Earliest known substitution cipher
- Replaces each letter by 3rd letter on
- Example:

```
meet me after the toga party

PHHW PH DIWHU WKH WRJD SDUWB
```

Caesar Cipher

- Define transformation as:

  a b c d e f g h i j k l m n o p q r s t u v w x y z

  D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

- Mathematically give each letter a number

  a b c d e f g h i j k l m

  0 1 2 3 4 5 6 7 8 9 10 11 12

  n o p q r s t u v w x y Z

  13 14 15 16 17 18 19 20 21 22 23 24 25

- Then have Caesar cipher as:

  $C = E(p) = (p + k) \bmod (26)$

  $p = D(C) = (C - k) \bmod (26)$

Cryptanalysis of Caesar Cipher

- Only has 25 possible ciphers
  - A maps to B,..Z
- Given ciphertext, just try all shifts of letters
- Do need to recognize when have plaintext
- E.g., break ciphertext "GCUA VQ DTGCM"

Monoalphabetic Cipher

- Rather than just shifting the alphabet
- Could shuffle (jumble) the letters arbitrarily
- Each plaintext letter maps to a different random ciphertext letter
- Key is 26 letters long

```
Plain:  abcdefghijklmnopqrstuvwxyz
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN
Plaintext:  ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA
```

One-Time Pad

- If a truly random key as long as the message is used, the cipher will be secure - One-Time pad
- E.g., a random sequence of 0's and 1's XORed to plaintext, no repetition of keys
- Unbreakable since ciphertext bears no statistical relationship to the plaintext
- For **any** plaintext, it needs a random key of the same length
  - Hard to generate large amount of keys
- Have problem of safe distribution of key

Transposition Ciphers

- Now consider classical **transposition** or **permutation** ciphers
- These hide the message by rearranging the letter order, without altering the actual letters used
- Can recognise these since have the same frequency distribution as the original text

Rail Fence cipher

- Write message letters out diagonally over a number of rows
- Then read off cipher row by row
- E.g., write message out as:

```
m e m a t r h t g p r y

  e t e f e t e o a a t
```

- Giving ciphertext

```
MEMATRHTGPRYETEFETEOAAT
```

Product Ciphers

- Ciphers using substitutions or transpositions are not secure because of language characteristics
- Hence consider using several ciphers in succession to make harder, but:
  - Two substitutions make a more complex substitution
  - Two transpositions make more complex transposition
  - But a substitution followed by a transposition makes a new much harder cipher
- This is bridge from classical to modern ciphers

- Modern Symmetric Ciphers (DES)

Block vs Stream Ciphers

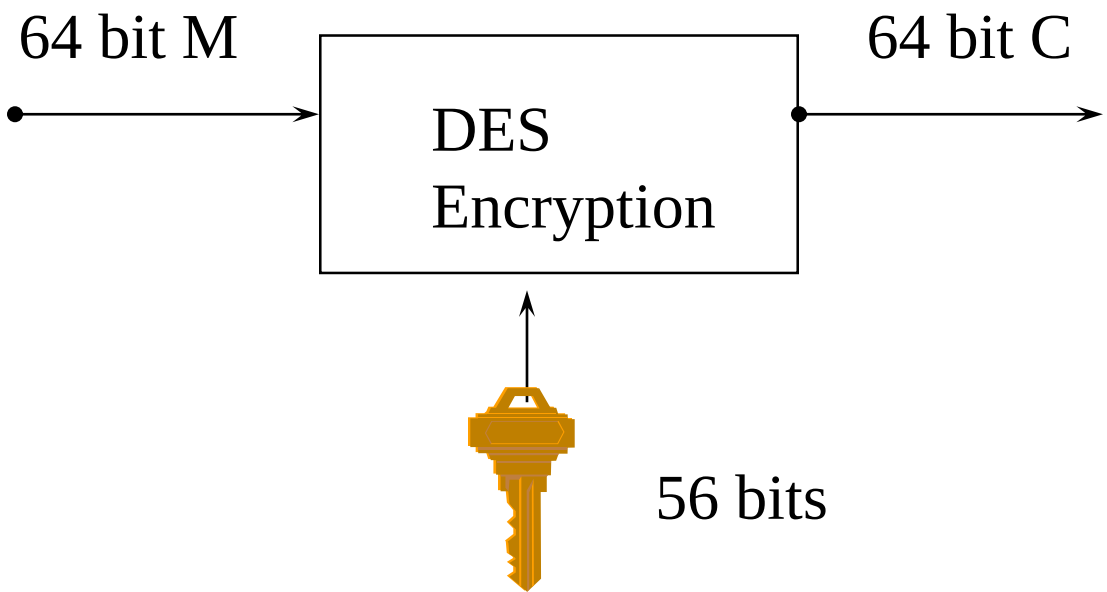
- Block ciphers process messages in into blocks, each of which is then en/decrypted
- Like a substitution on very big characters
  - 64-bits or more
- Stream ciphers process messages a bit or byte at a time when en/decrypting
- Many current ciphers are block ciphers, one of the most widely used types of cryptographic algorithms

Block Cipher Principles

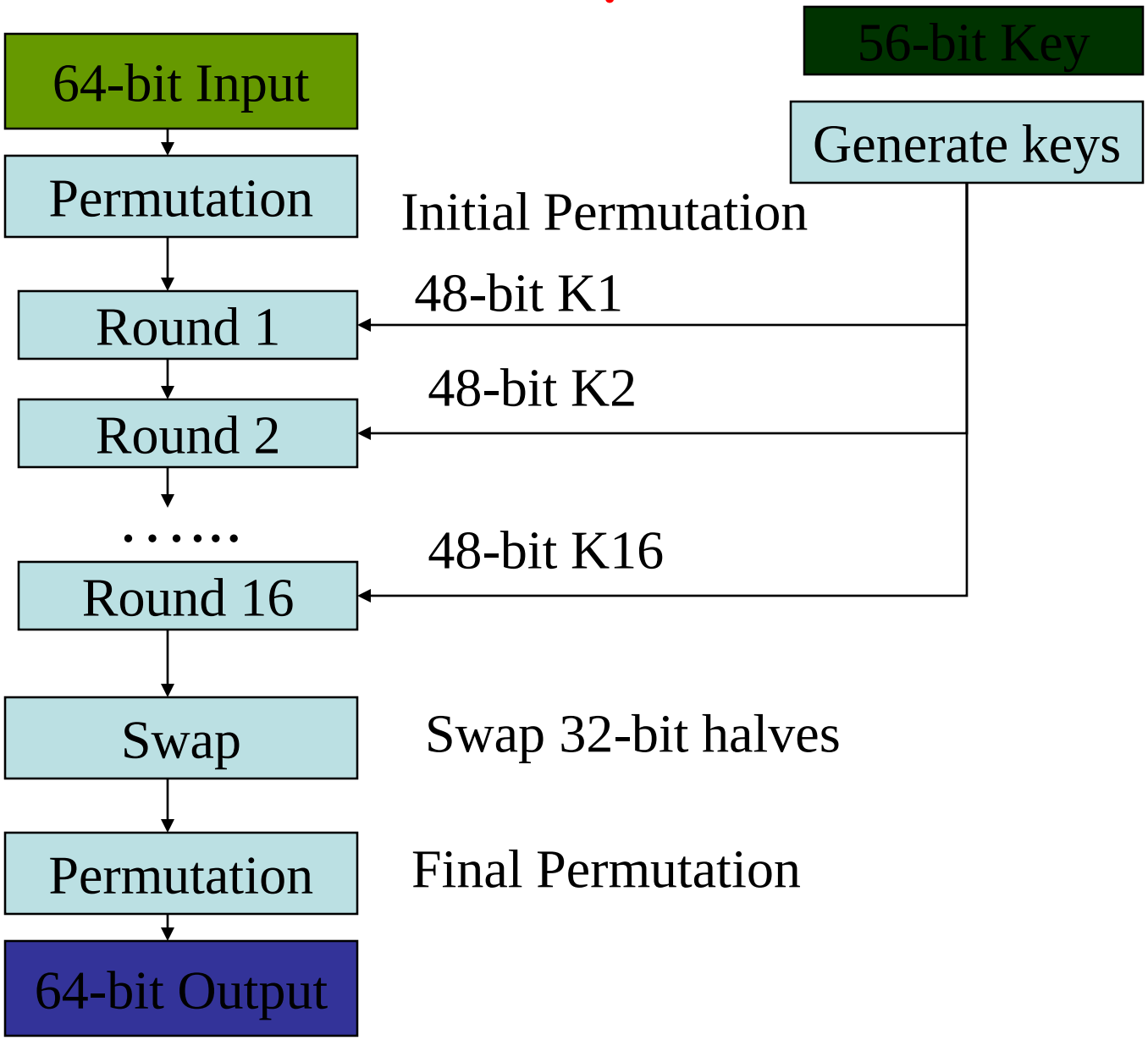
- Most symmetric block ciphers are based on a **Feistel Cipher Structure**
- Block ciphers look like an extremely large substitution
- Would need table of 2^{64} entries for a 64-bit block
- Instead create from smaller building blocks
- Using idea of a product cipher

DES (Data Encryption Standard)

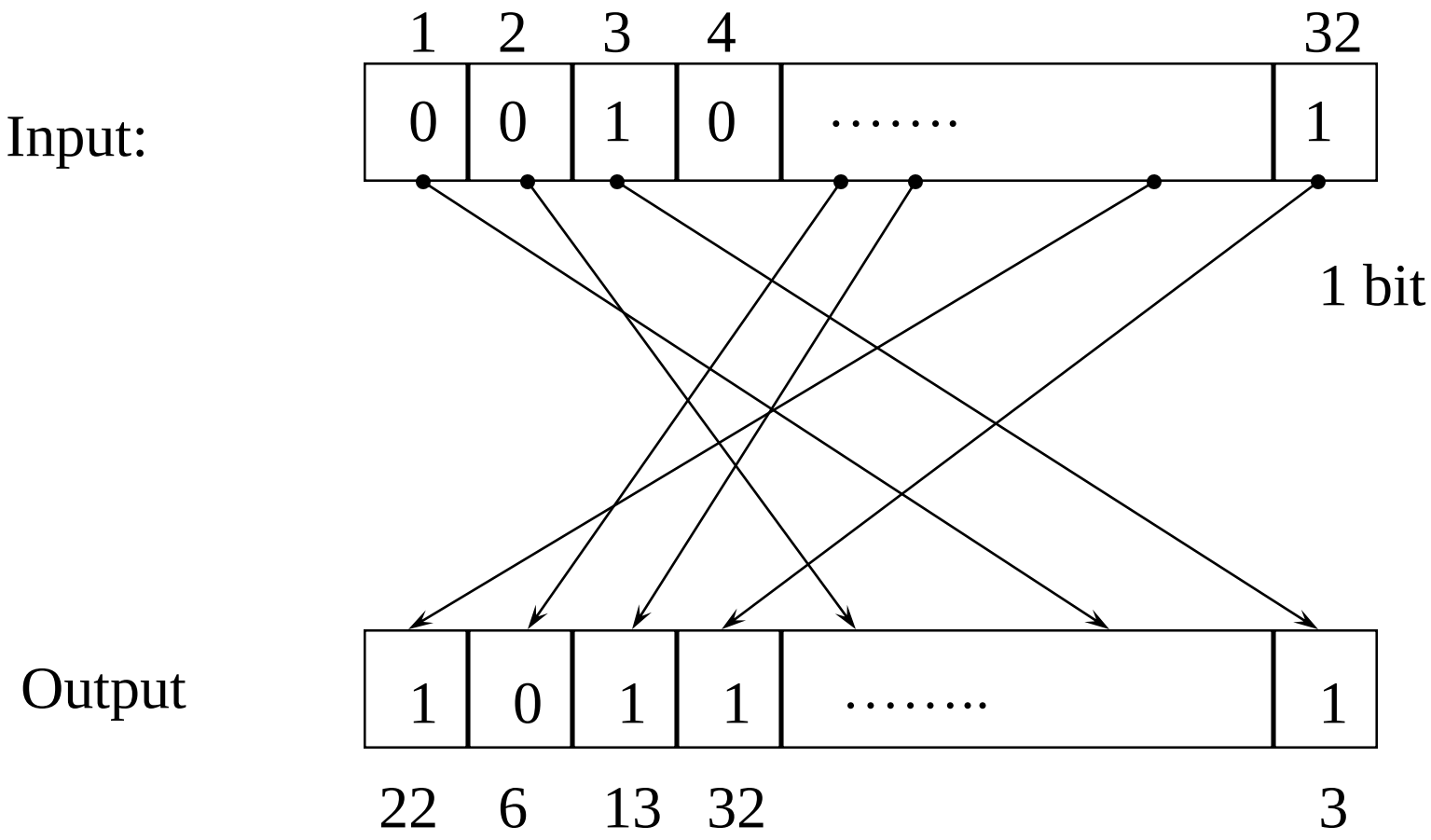
- Published in 1977, standardized in 1979.
- Key: 64 bit quantity=8-bit parity+56-bit key
  - Every 8th bit is a parity bit.
- 64 bit input, 64 bit output.



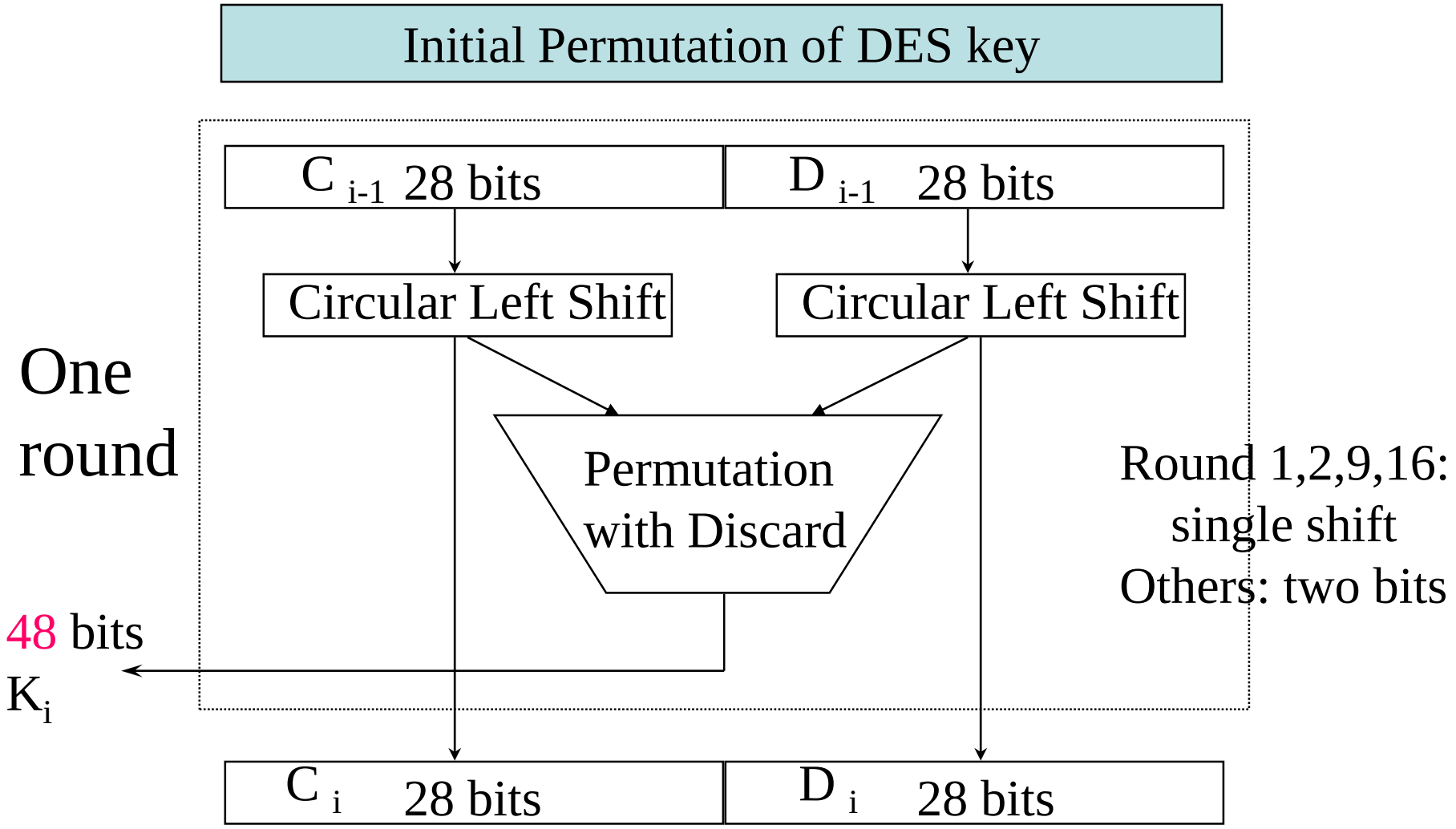
DES Top View


56-bit Key
64-bit Input
Generate keys
Permutation
Initial Permutation
48-bit K1
Round 1
48-bit K2
Round 2
. . . . . .
48-bit K16
Round 16
Swap
Swap 32-bit halves
Permutation
Final Permutation
64-bit Output

Bit Permutation (1-to-1)

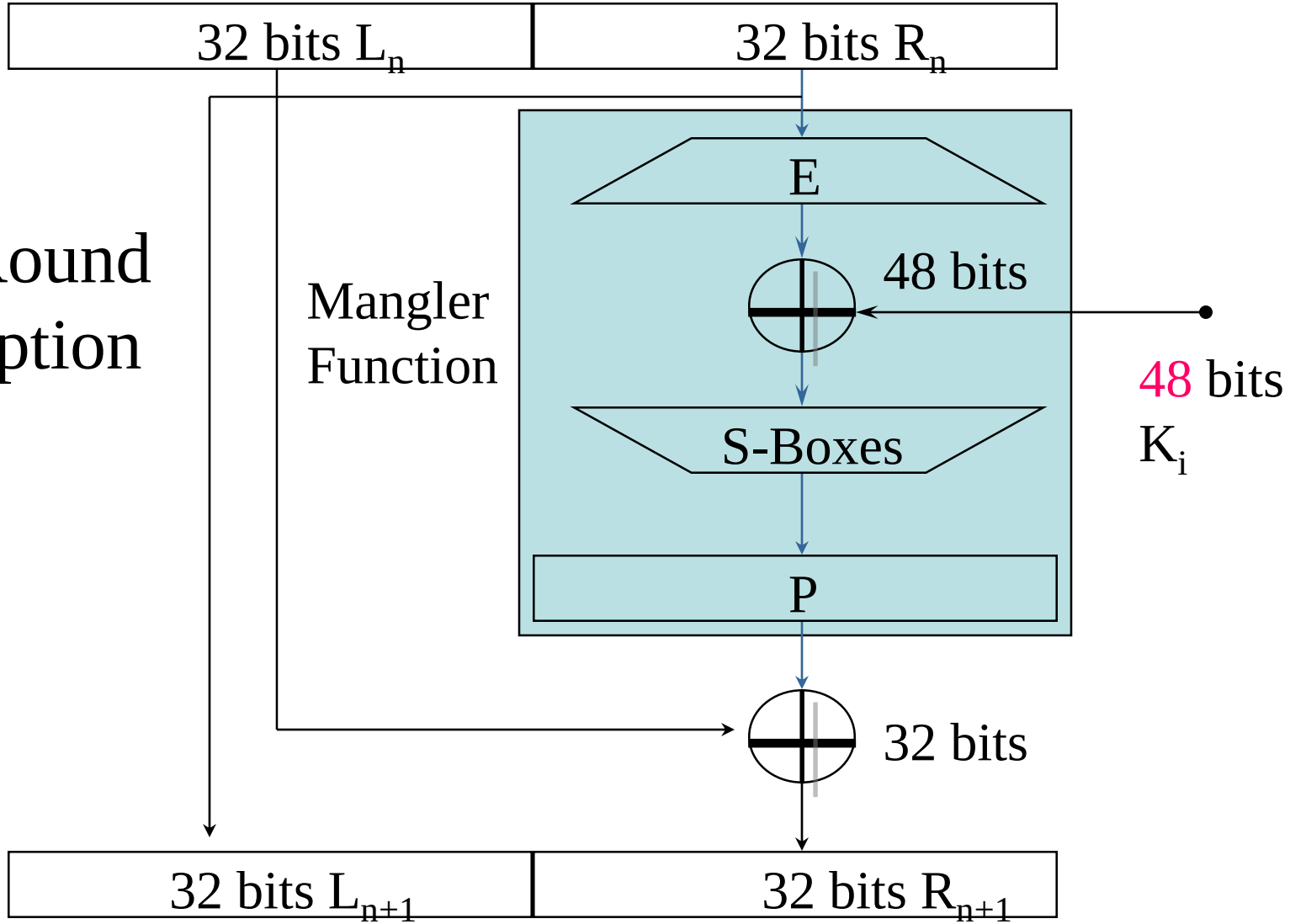


Per-Round Key Generation

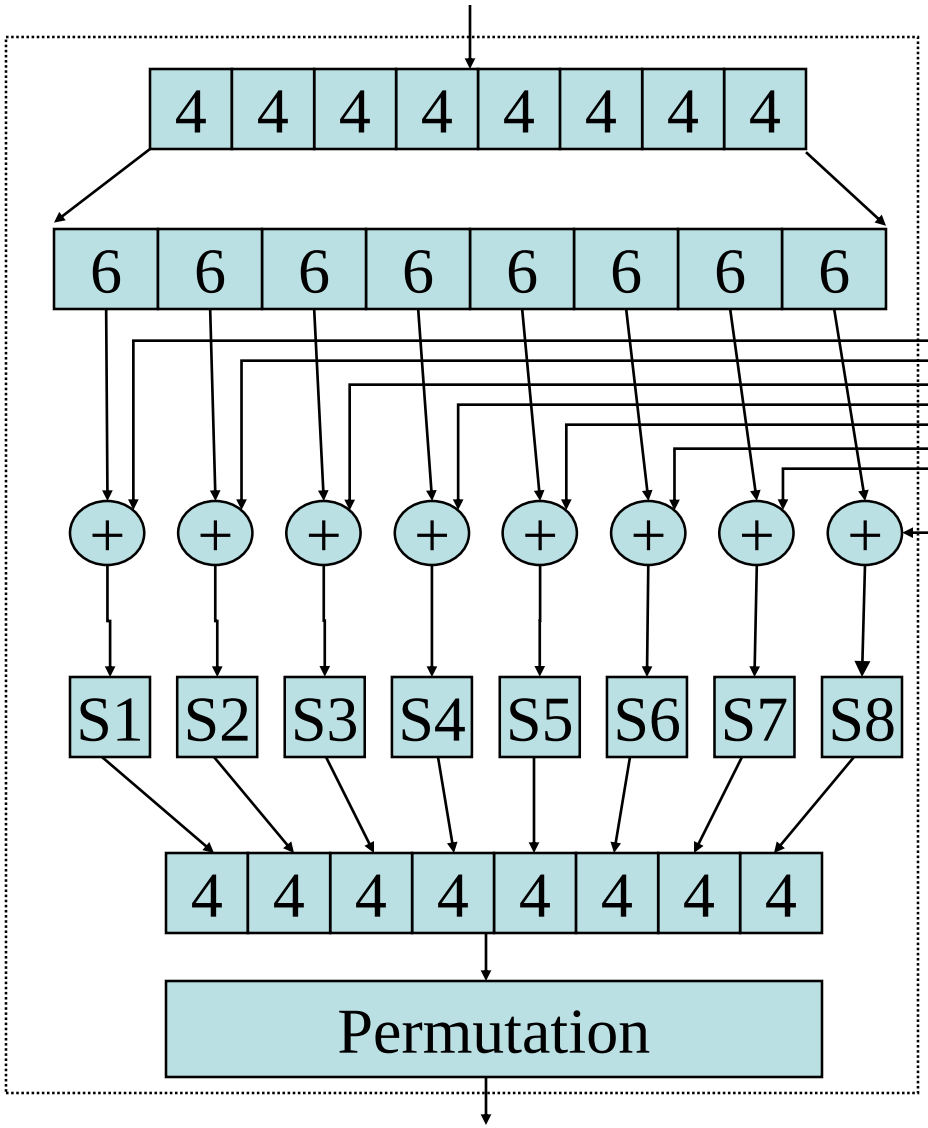


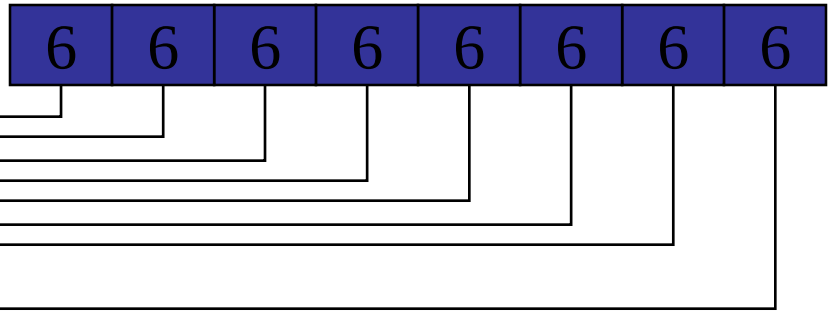
A DES Round

One Round Encryption



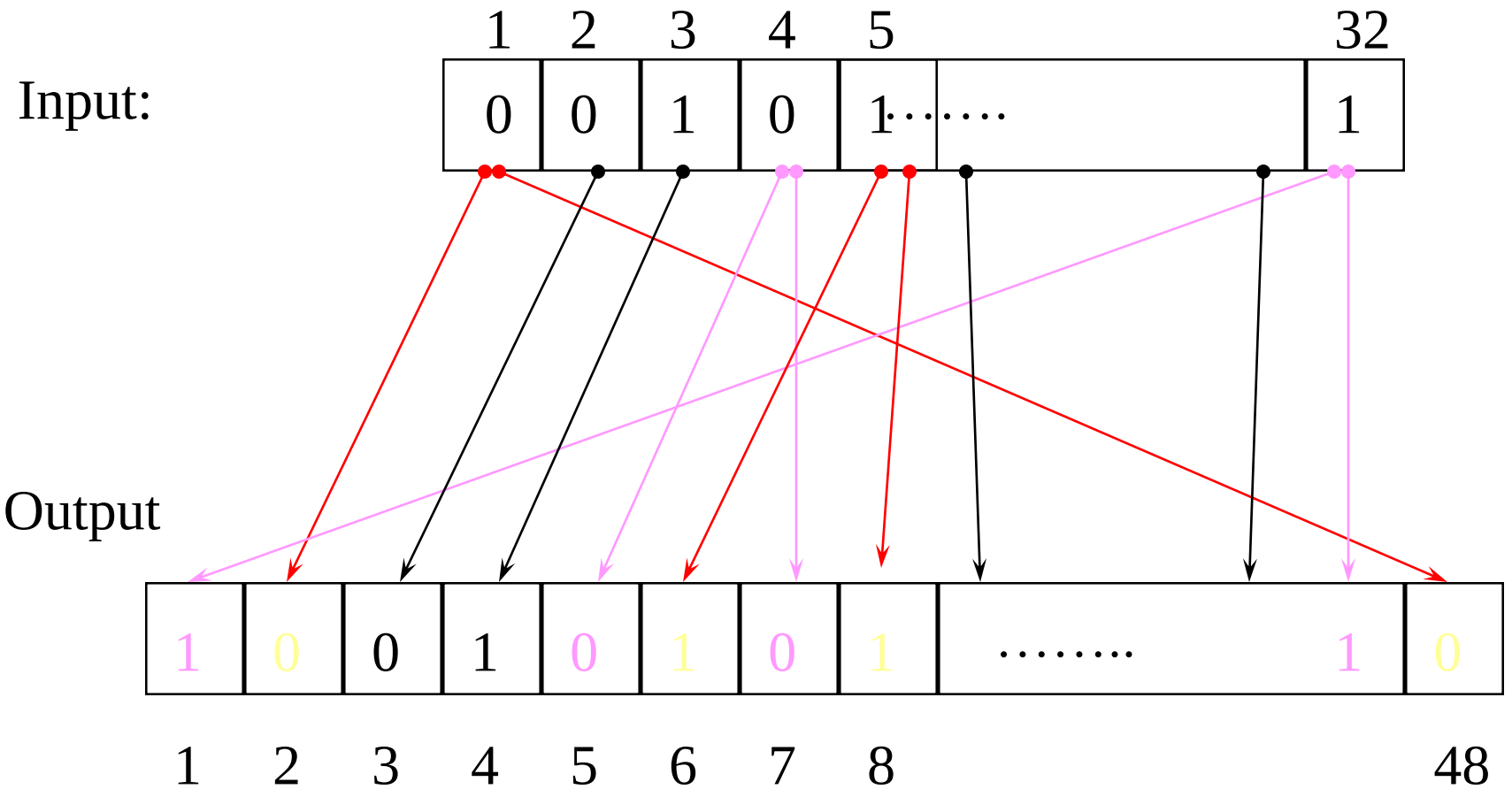
Mangler Function





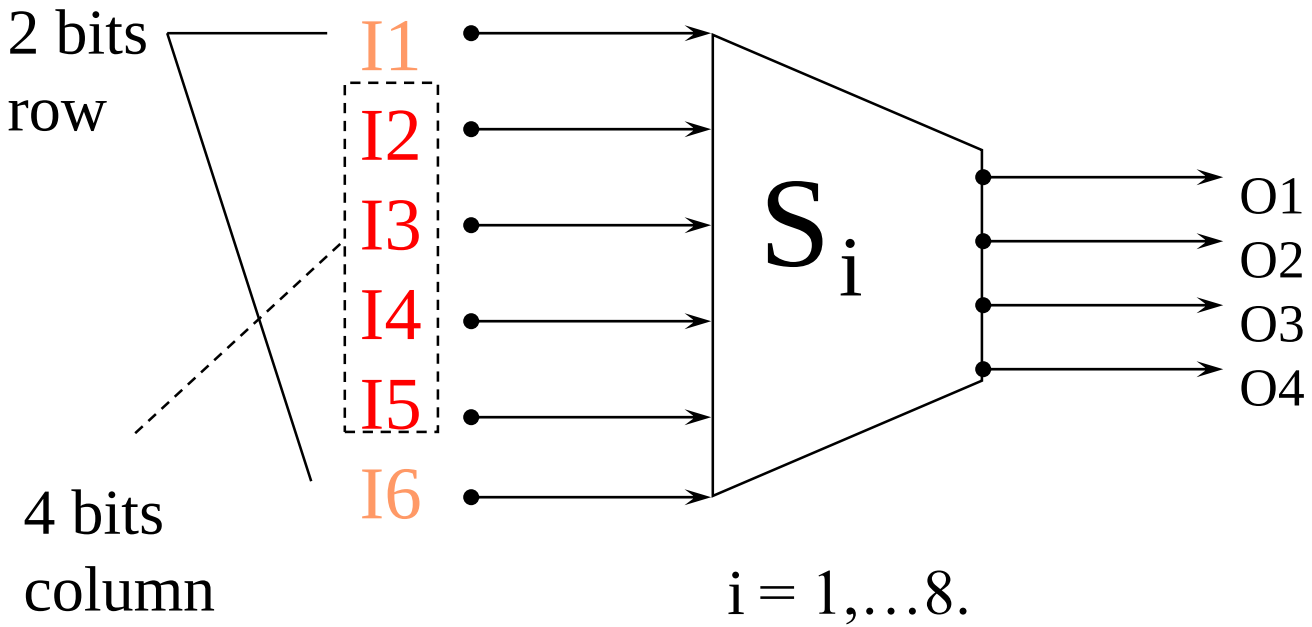
The permutation produces "spread" among the chunks/S-boxes!

Bits Expansion (1-to-m)



S-Box (Substitute and Shrink)

- 48 bits ==> 32 bits. (8*6 ==> 8*4)
- 2 bits used to select amongst 4 substitutions for the rest of the 4-bit quantity



S-Box Examples

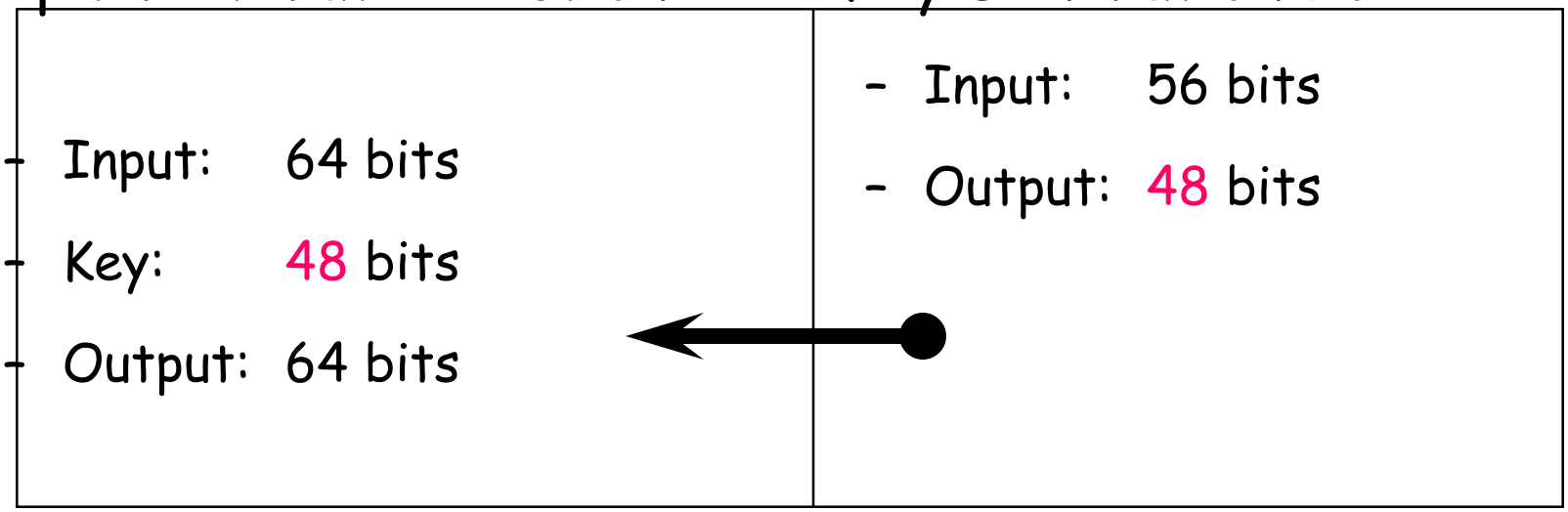
Each row and column contain different numbers.

	0	1	2	3	4	5	6	7	8	9…. 15
0	14	4	13	1	2	15	11	8	3	
1	0	15	7	4	14	2	13	1	10	
2	4	1	14	8	13	6	2	11	15	
3	15	12	8	2	4	9	1	7	5	

Example: input: 100110 output: ???

DES Standard

- Cipher Iterative Action :
  - Input: 64 bits
  - Key: 48 bits
  - Output: 64 bits
- Key Generation Box :
  - Input: 56 bits
  - Output: 48 bits



One round (Total 16 rounds)

DES Box Summary

- Simple, easy to implement:
  - Hardware/gigabits/second, software/megabits/second
- 56-bit key DES may be acceptable for non-critical applications but triple DES (DES3) should be secure for most applications today
- Supports several operation modes (ECB CBC, OFB, CFB) for different applications

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 a huge cluster of computers over the Internet in a few months
  - in 1998 on dedicated hardware called "DES cracker" by EFF in a few days ($220,000)
  - in 1999 above combined in 22hrs!
- Still must be able to recognize plaintext
- No big flaw for DES algorithms

DES Replacement

- Triple-DES (3DES)
  - 168-bit key, no brute force attacks
  - Underlying encryption algorithm the same, no effective analytic attacks
  - Drawbacks
    - Performance: no efficient software codes for DES/3DES
    - Efficiency/security: bigger block size desirable
- Advanced Encryption Standards (AES)
  - US NIST issued call for ciphers in 1997
  - Rijndael was selected as the AES in Oct-2000

Brute Force Search

- Always possible to simply try every key
- Most basic attack, proportional to key size
- Assume either know / recognise plaintext

Key Size (bits)	Number of Alternative Keys	Time required at 1 encryption/μs	Time required at 10^6 encryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\ \mu s = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\ \mu s = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\ \mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\ \mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\ \mu s = 6.4 \times 10^{12}$ years	6.4×10^{6} years

Outlines

- Strength/weakness of DES, AES
- Public Key Cryptography
- Modular Arithmetic
- RSA

Modular Arithmetic

- Public key algorithms are based on modular arithmetic.
- Modular addition.
- Modular multiplication.
- Modular exponentiation.

Modular Addition

- Addition modulo (mod) *K*
  - Poor cipher with (d_k+d_m) mod *K*, e.g., if *K*=10 and d_k is the key.

+	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	1	2	3	4	5	6	7	8	9	0
2	2	3	4	5	6	7	8	9	0	1
3	3	4	5	6	7	8	9	0	1	2

- Additive inverse: addition mod *K* yields 0.
- "Decrypt" by adding inverse.

Modular Multiplication

- Multiplication modulo *K*

*	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	1	2	3	4	5	6	7	8	9	1
2	0	2	4	6	8	0	2	4	6	8
3	0	3	6	9	2	5	8	1	4	7

- Multiplicative inverse: multiplication mod *K* yields 1
- Only some numbers have inverse

Modular Multiplication

- Only the numbers *relatively prime* to *n* will have mod *n* multiplicative inverse
- *x*, *m* relative prime: no other common factor than 1
  - Eg. 8 & 15 are relatively prime - factors of 8 are 1,2,4,8 and of 15 are 1,3,5,15 and 1 is the only common factor

Totient Function

- Totient function *ø(n)*: number of integers less than *n* relatively prime to *n*
  - if *n* is prime,
    - *ø(n)=n-1*
  - if *n=p*q*, and *p*, *q* are primes, p != q
    - *ø(n)=(p-1)(q-1)*
  - E.g.,
    - ø(37) = 36
    - ø(21) = (3-1)×(7-1) = 2×6 = 12

Modular Exponentiation

x^y	0	1	2	3	4	5	6	7	8	9
0		0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1
2	1	2	4	8	6	2	4	8	6	2
3	1	3	9	7	1	3	9	7	1	3
4	1	4	6	4	6	4	6	4	6	4
5	1	5	5	5	5	5	5	5	5	5
6	1	6	6	6	6	6	6	6	6	6
7	1	7	9	3	1	7	9	3	1	7
8	1	8	4	2	6	8	4	2	6	8
9	1	9	1	9	1	9	1	9	1	9

Modular Exponentiation

- $x^y \bmod n = x^{y \bmod \phi(n)} \bmod n$
- if $y = 1 \bmod \phi(n)$ then $x^y \bmod n = x \bmod n$

RSA (Rivest, Shamir, Adleman)

- The most popular one.
- Support both public key encryption and digital signature.
- Assumption/theoretical basis:
  - Factoring a big number is hard.
- Variable key length (usually 512 bits).
- Variable plaintext block size.
  - Plaintext must be "smaller" than the key.
  - Ciphertext block size is the same as the key length.

What Is RSA?

- To generate key pair:
  - Pick large primes (>= 256 bits each) *p* and *q*
  - Let *n* = *p***q*, keep your *p* and *q* to yourself!
  - For public key, choose *e* that is relatively prime to *ø(n) =(p-1)(q-1)*, let pub = <*e*,*n*>
  - For private key, find *d* that is the multiplicative inverse of *e* mod *ø(n)*, i.e., *e***d* = 1 mod *ø(n)*, let priv = <*d*,*n*>

RSA Example

1. Select primes: $p=17$ & $q=11$
2. Compute $n = pq = 17 \times 11 = 187$
3. Compute $\phi(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. Select e : $\gcd(e,160)=1$; choose $e=7$
5. Determine d: $de=1 \mod 160$ and $d < 160$ Value is $d=23$ since $23 \times 7 = 161 = 10 \times 160 + 1$
6. Publish public key $KU=\{7,187\}$
7. Keep secret private key $KR=\{23,17,11\}$

How Does RSA Work?

- Given pub = $\langle e, n \rangle$ and priv = $\langle d, n \rangle$
  - encryption: $c = m^e \bmod n$, $m < n$
  - decryption: $m = c^d \bmod n$
  - signature: $s = m^d \bmod n$, $m < n$
  - verification: $m = s^e \bmod n$
- given message $M = 88$ (nb. $88<187$)
- encryption:

  $C = 88^7 \bmod 187 = 11$
- decryption:

  $M = 11^{23} \bmod 187 = 88$

Why Does RSA Work?

- Given pub = $\langle e, n \rangle$ and priv = $\langle d, n \rangle$
  - $n = p*q$, $\phi(n) = (p-1)(q-1)$
  - $e*d = 1 \bmod \phi(n)$
  - $x^{e*d} = x \bmod n$
  - encryption: $c = m^e \bmod n$
  - decryption: $m = c^d \bmod n = m^{e*d} \bmod n = m \bmod n = m$ (since $m < n$)
  - digital signature (similar)

Is RSA Secure?

- Factoring 512-bit number is very hard!
- But if you can factor big number *n* then given public key <*e*,*n*>, you can find *d*, hence the private key by:
  - Knowing factors *p*, *q*, such that, n = *p***q*
  - Then *ø(n)* =*(p-1)(q-1)*
  - Then *d* such that *e***d* = 1 mod *ø(n)*
- Threat
  - Moore's law
  - Refinement of factorizing algorithms
- For the near future, a key of 1024 or 2048 bits needed

Symmetric (DES) vs. Public Key (RSA)

- Exponentiation of RSA is expensive !
- AES and DES are much faster
  - 100 times faster in software
  - 1,000 to 10,000 times faster in hardware
- RSA often used in combination in AES and DES
  - Pass the *session key* with RSA