Security Analytics

Rakesh Verma
ReDAS laboratory
Computer Science Department
University of Houston

Lec. 1 contd & Lec. 2

Security Goals and Mechanisms
Introduction to Cryptography

How determined are attackers?

- Two case studies ("Ten laws of security"):
  - ➢Microsoft Xbox – hardware attack
  - ➢RSA – spearphishing emails
- Recent examples (1/22/18)
  - Fire and Fury book pdf
  - Hackers steal almost $400 million from cryptocurrency ICOs (ZDNet)
- Bottom line: Attackers can be extremely determined when the ``prize/thrill'' is huge

Outline

- Overview of Cryptography
- Classical Symmetric Cipher
- Modern Symmetric Ciphers (DES, AES)

(Adapted from slides accompanying book by William Stallings, Some slides are courtesy J. Kurose and K. Ross)

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

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

Secret Key vs. Secret Algorithm

- Secret algorithm: additional hurdle
- Hard to keep secret if used widely:
  - Reverse engineering, social engineering
- Commercial: published
  - Wide review, trust
- Military: avoid giving enemy good ideas

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

Unconditional vs. Computational Security

- **Unconditional security**
  - No matter how much computer power is available, the cipher cannot be broken
  - The ciphertext provides insufficient information to uniquely determine the corresponding plaintext
  - Only one-time pad scheme qualifies
- **Computational security**
  - The cost of breaking the cipher exceeds the value of the encrypted info
  - The time required to break the cipher exceeds the useful lifetime of the info

Brute Force Search

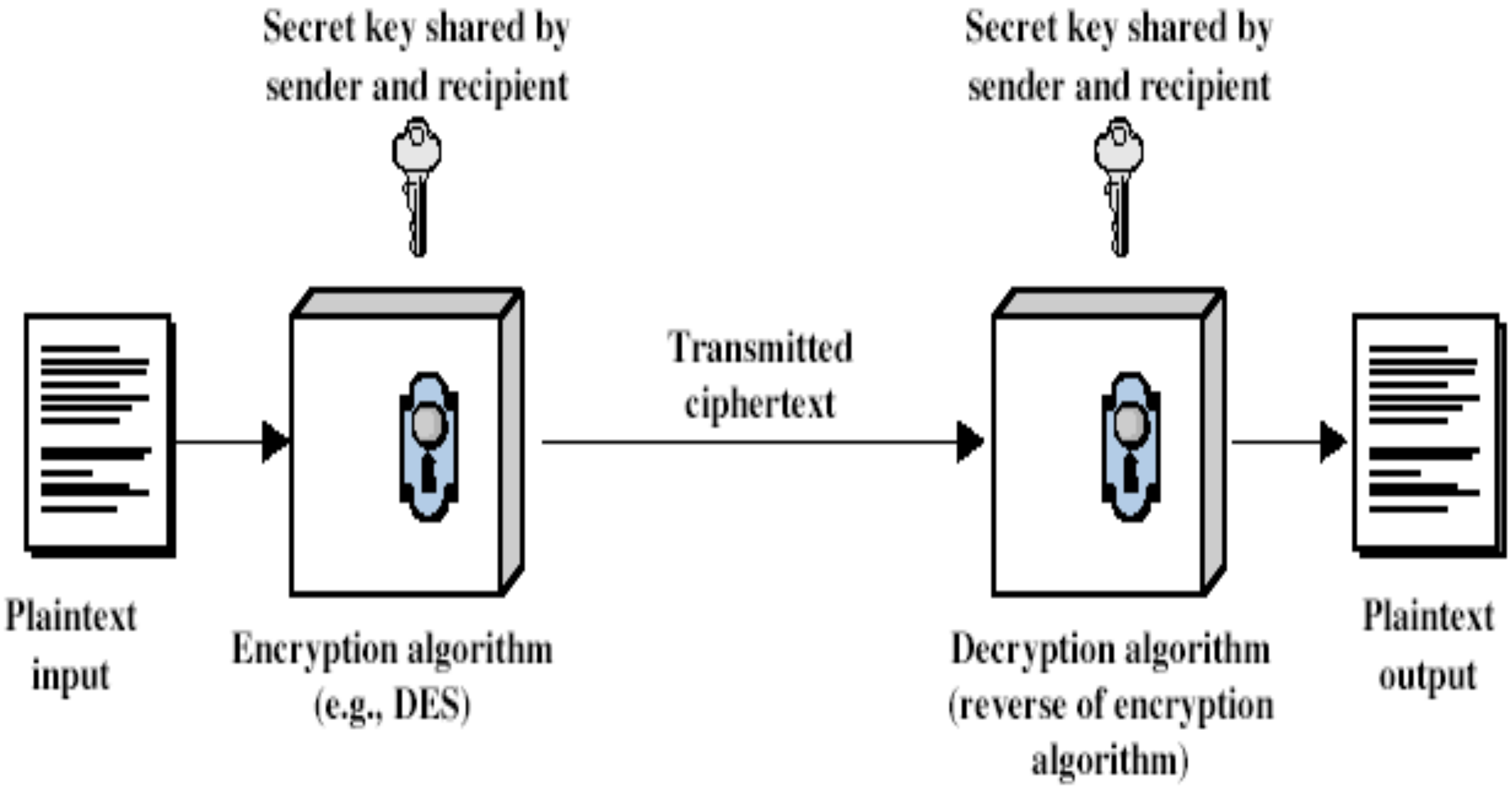
- Always possible to simply try every key
- Most basic attack, proportional to key size
- Assume either know / recognise plaintext

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} μs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs = 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)	$26! = 4 \times 10^{26}$	2×10^{26} $\mu s = 6.4 \times 10^{12}$ years	6.4×10^6 years

Outline

- Overview of Cryptography
- Classical Symmetric Cipher
  - Substitution Cipher
  - Transposition Cipher
- Modern Symmetric Ciphers (DES, AES)

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:

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

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

Cryptanalysis of Caesar Cipher

- Only have 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

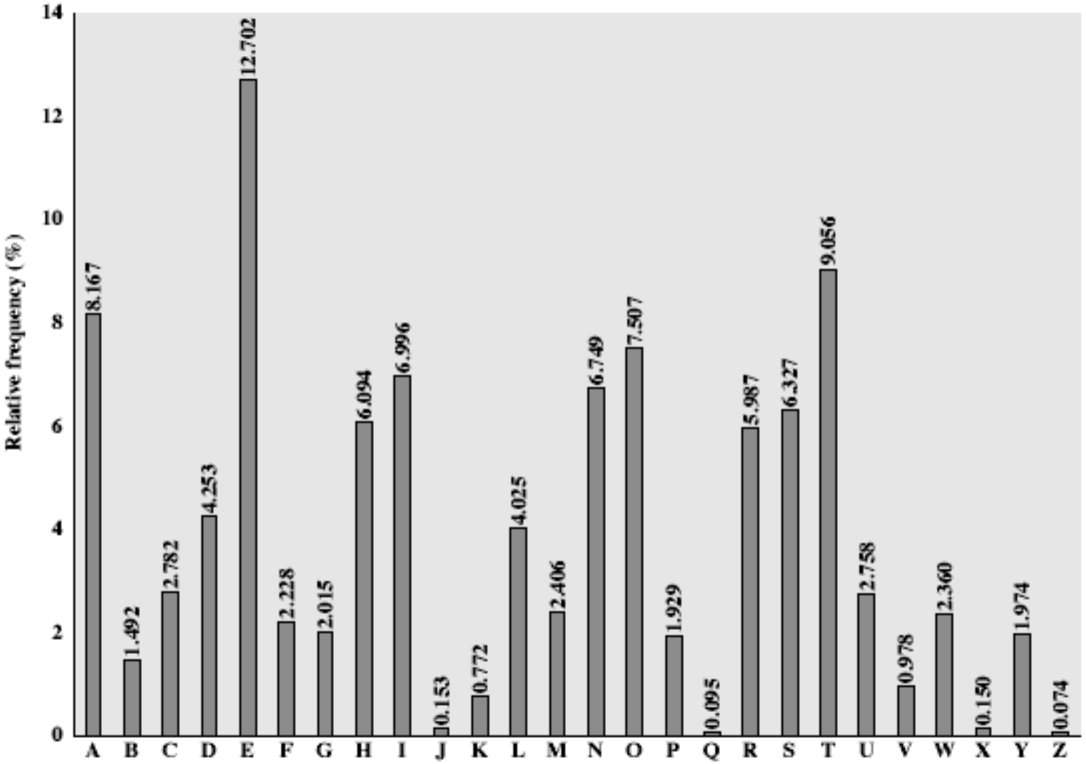
- Rather than just shifting the alphabet
- Could shuffle (jumble) the letters arbitrarily
- Each plaintext letter maps to a different random ciphertext letter
- Key is 26 letters long

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

Monoalphabetic Cipher Security

- Now have a total of $26! = 4 \times 10^{26}$ keys
- Is that secure?
- Problem is language characteristics
  - Human languages are **redundant**
  - Letters are not equally commonly used

English Letter Frequencies



Example Cryptanalysis

- Given ciphertext:

  ```
  UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
  VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
  EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
  ```

- Count relative letter frequencies (see text)
- Guess P & Z are e and t
- Guess ZW is th and hence ZWP is the
- Proceeding with trial and error finally get:

  ```
  it was disclosed yesterday that several informal but
  direct contacts have been made with political
  representatives of the viet cong in moscow
  ```

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

One-time Pad Analysis

- Let p(0) = x and p(1) = 1-x in plaintext message m
- Let's compute p(0) and p(1) for the ciphertext.
- We can get 0 in ciphertext if 0 in plaintext and 0 in key bit, or if 1 in plaintext and 1 in key bit
- So p(0) for ciphertext = $x/2 + (1-x)/2$ (why?) = ½
- Similarly, 1 in ciphertext if 0 in plaintex and 1 in key bit, or if 1 in plaintext and 0 in key bit
- So p(1) for ciphertext = $x/2 + (1-x)/2$ = ½
- Notice that the probability of a 0 or 1 in the ciphertext are the same. So ciphertext has "lost" the frequency information in the plaintext!

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

Outline

- Overview of Cryptography
- Classical Symmetric Cipher
- Modern Symmetric Ciphers (DES, AES)

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

Substitution-Permutation Ciphers

- Substitution-permutation (S-P) networks [Shannon, 1949]
  - modern substitution-transposition product cipher
- These form the basis of modern block ciphers
- S-P networks are based on the two primitive cryptographic operations
  - *substitution* (S-box)
  - *permutation* (P-box)
- provide *confusion* and *diffusion* of message

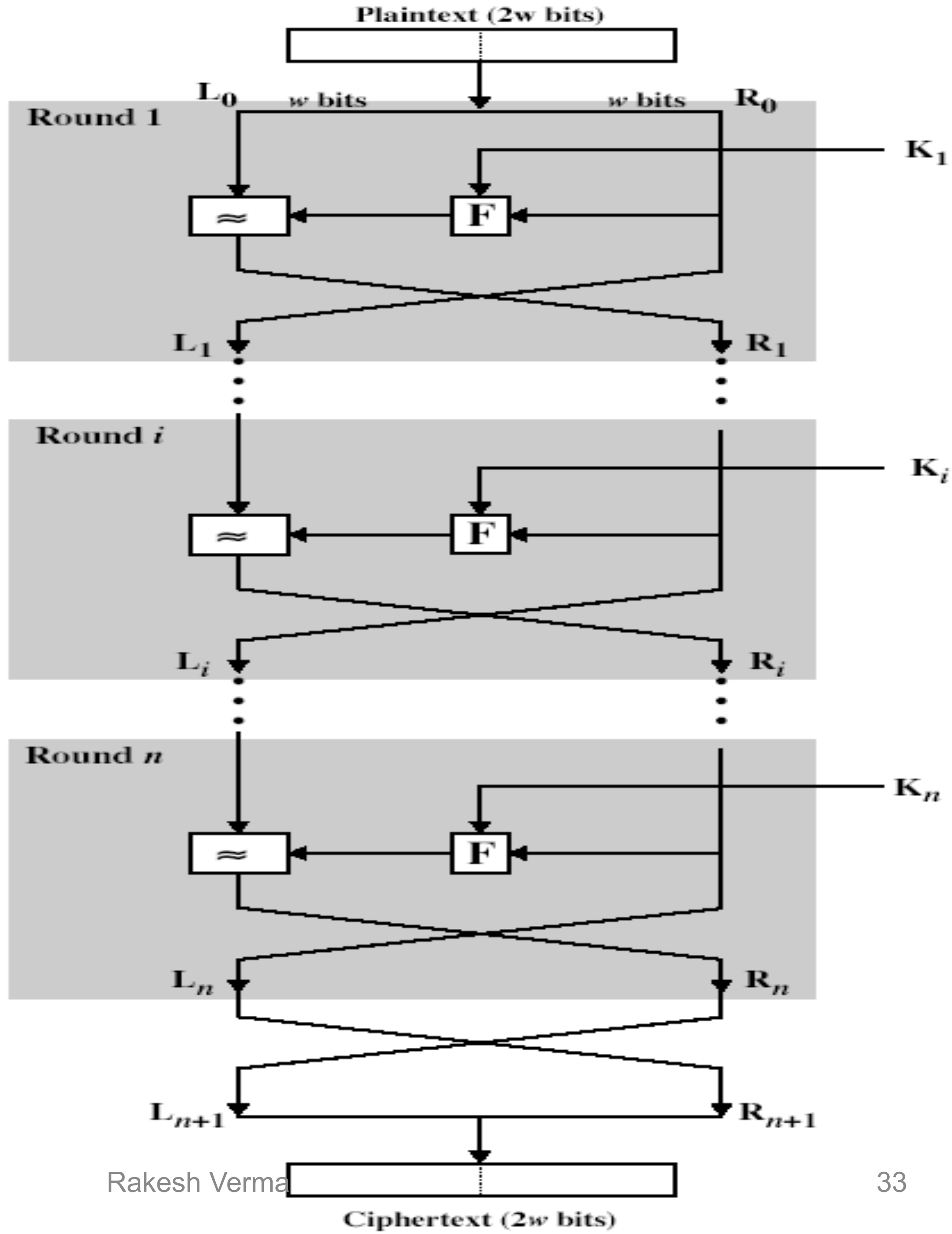
Confusion and Diffusion

- Cipher needs to completely obscure statistical properties of original message
- A one-time pad does this
- More practically Shannon suggested S-P networks 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

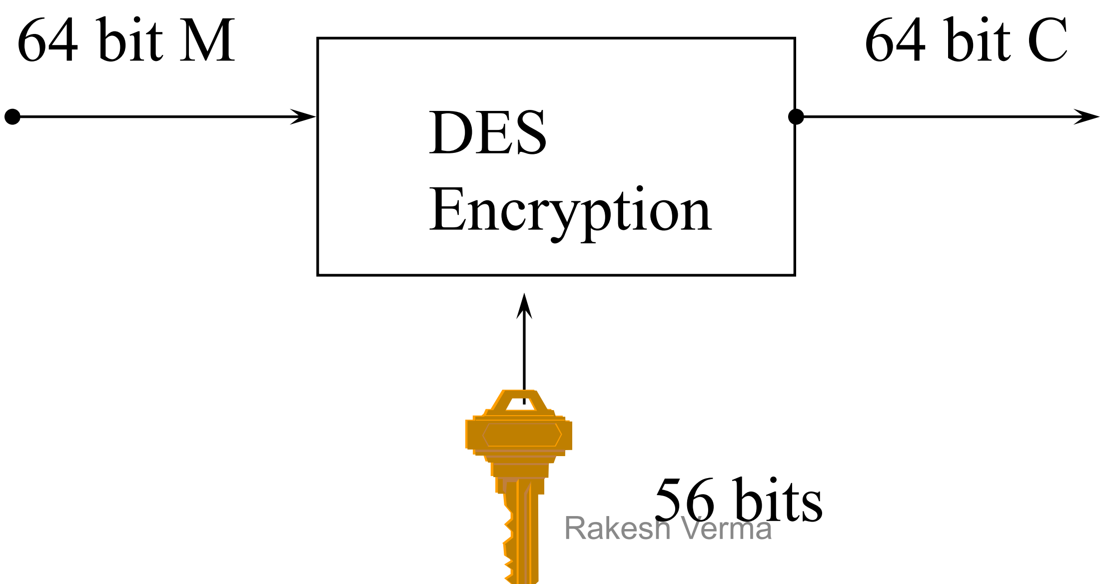
- **Feistel cipher** implements Shannon's S-P network concept
  - based on invertible product cipher
- Process through multiple rounds which
  - partitions input block into two halves
  - perform a substitution on left data half
  - based on round function of right half & subkey
  - then have permutation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

Feistel Cipher Structure

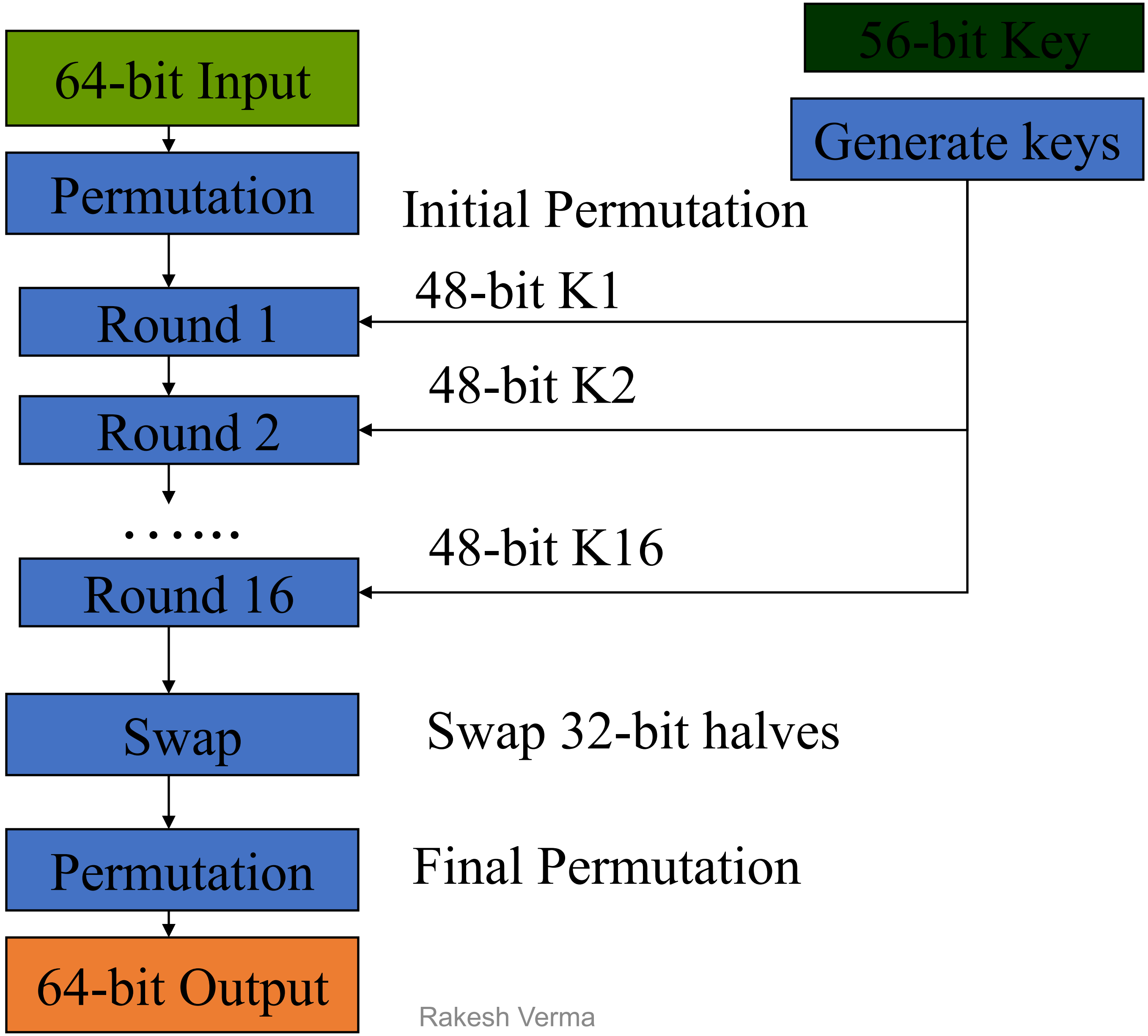


DES (Data Encryption Standard)

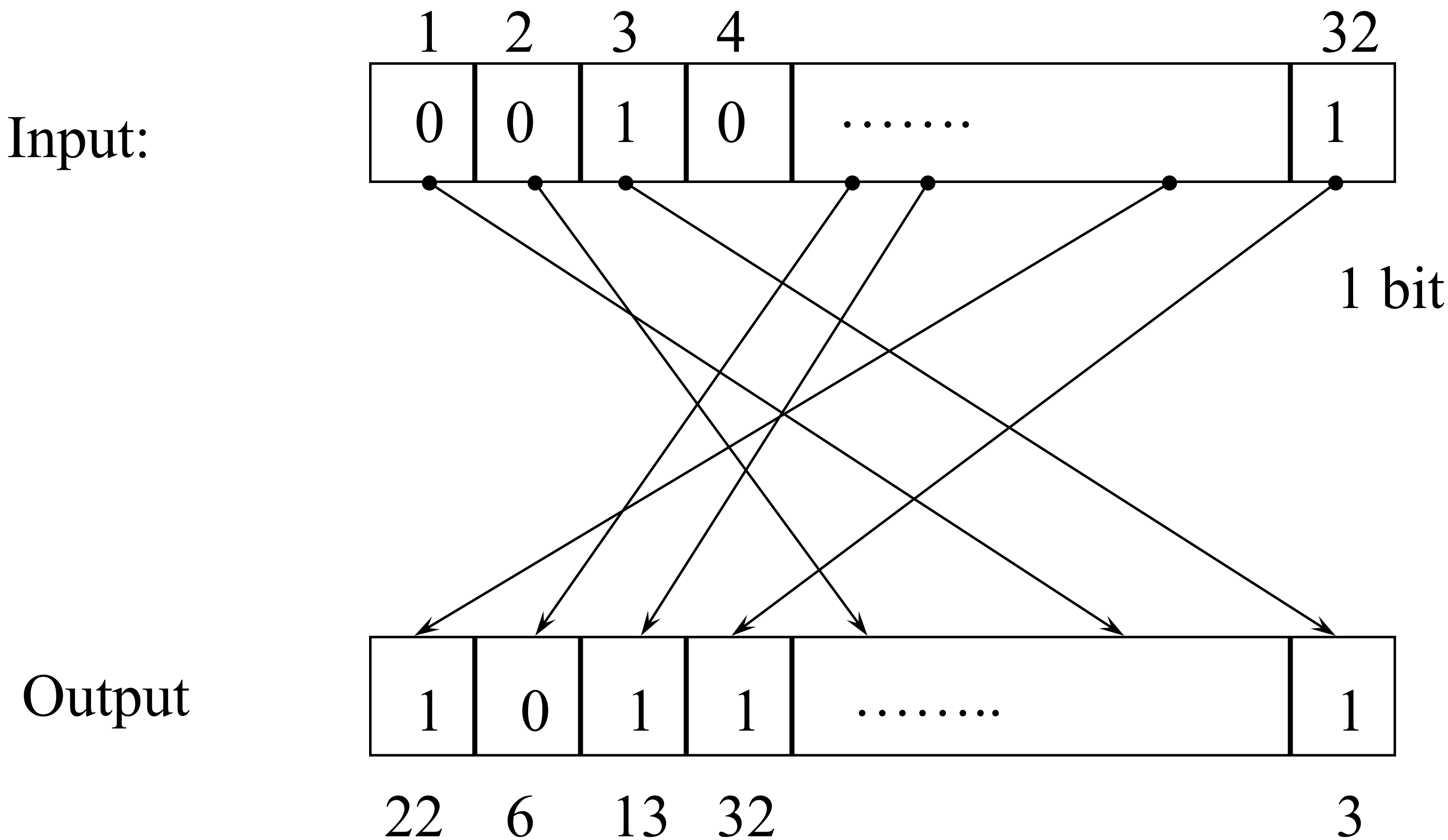
- Published in 1977, standardized in 1979.
- Key: 64 bit quantity=8-bit parity+56-bit key
  - Every 8th bit is a parity bit.
- 64 bit input, 64 bit output.



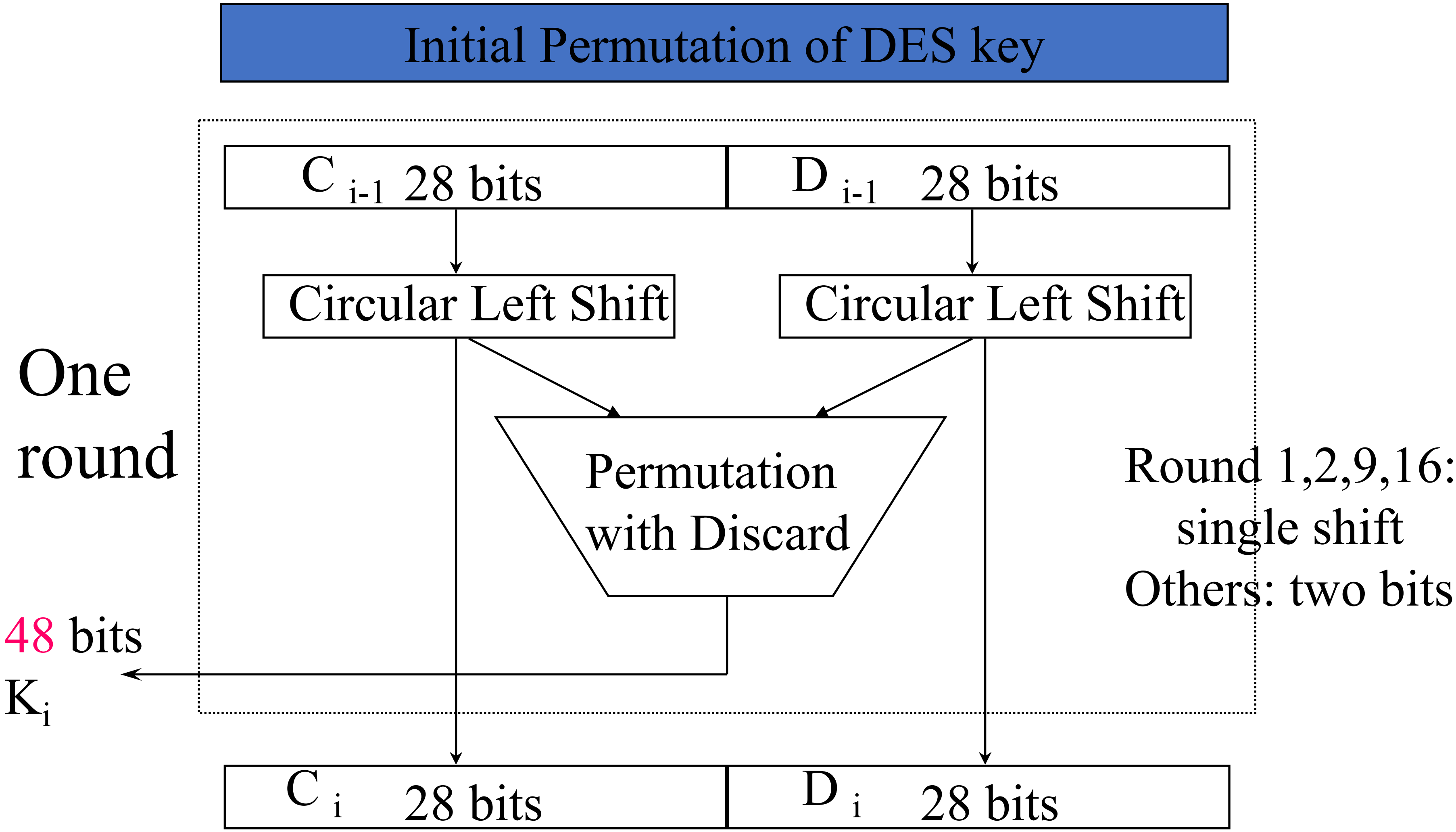
DES Top View


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)

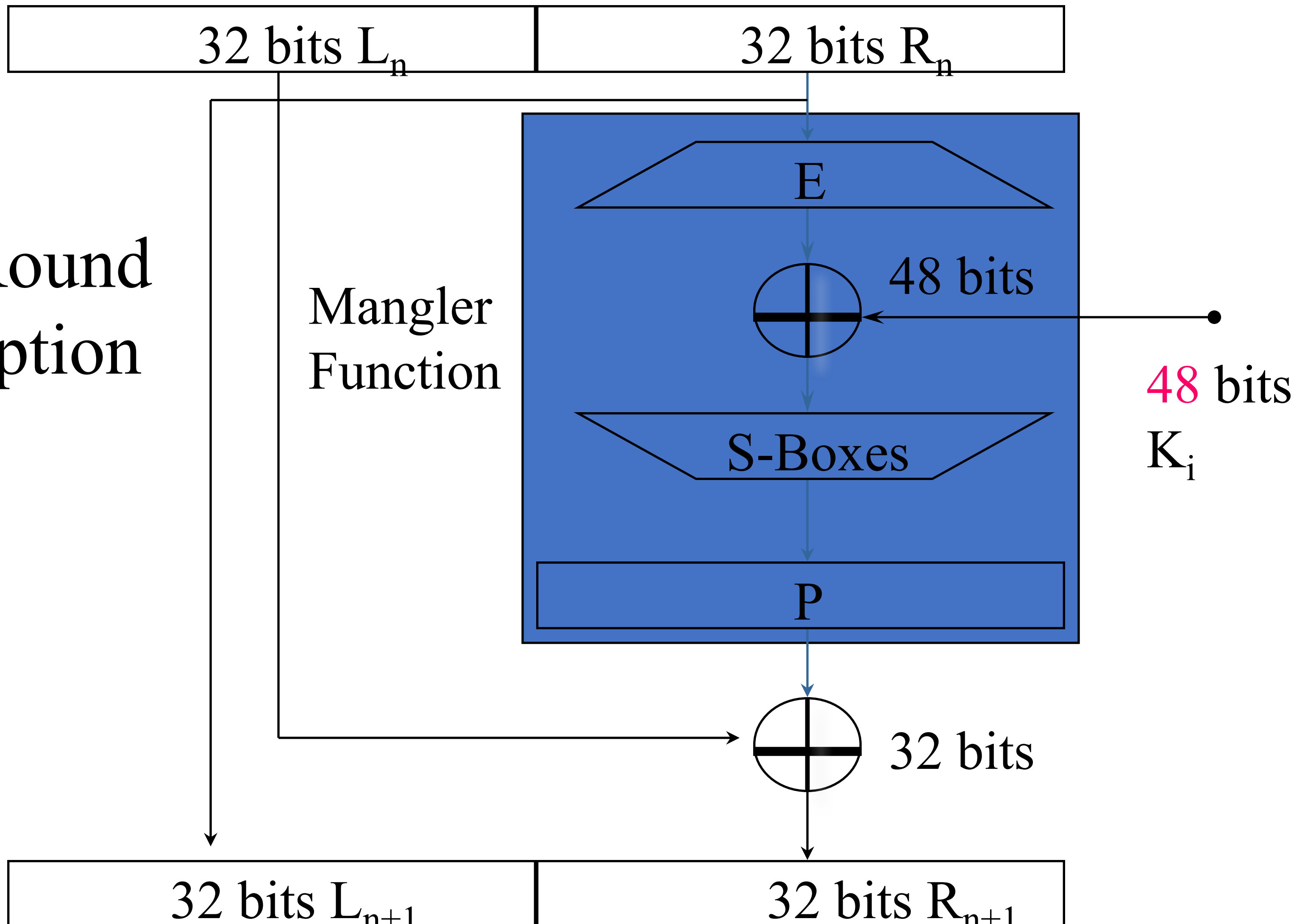


Per-Round Key Generation

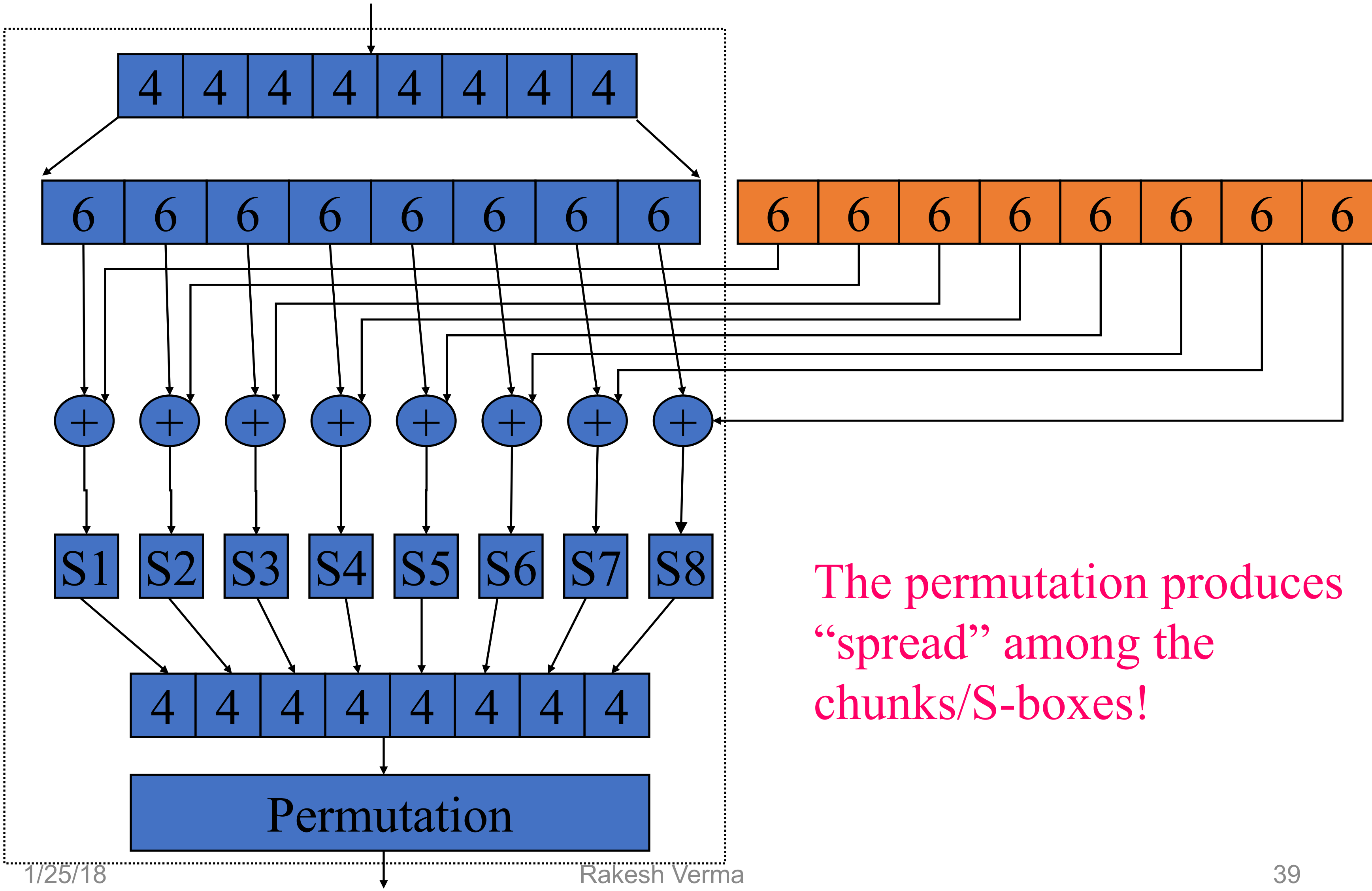


A DES Round

One Round
Encryption

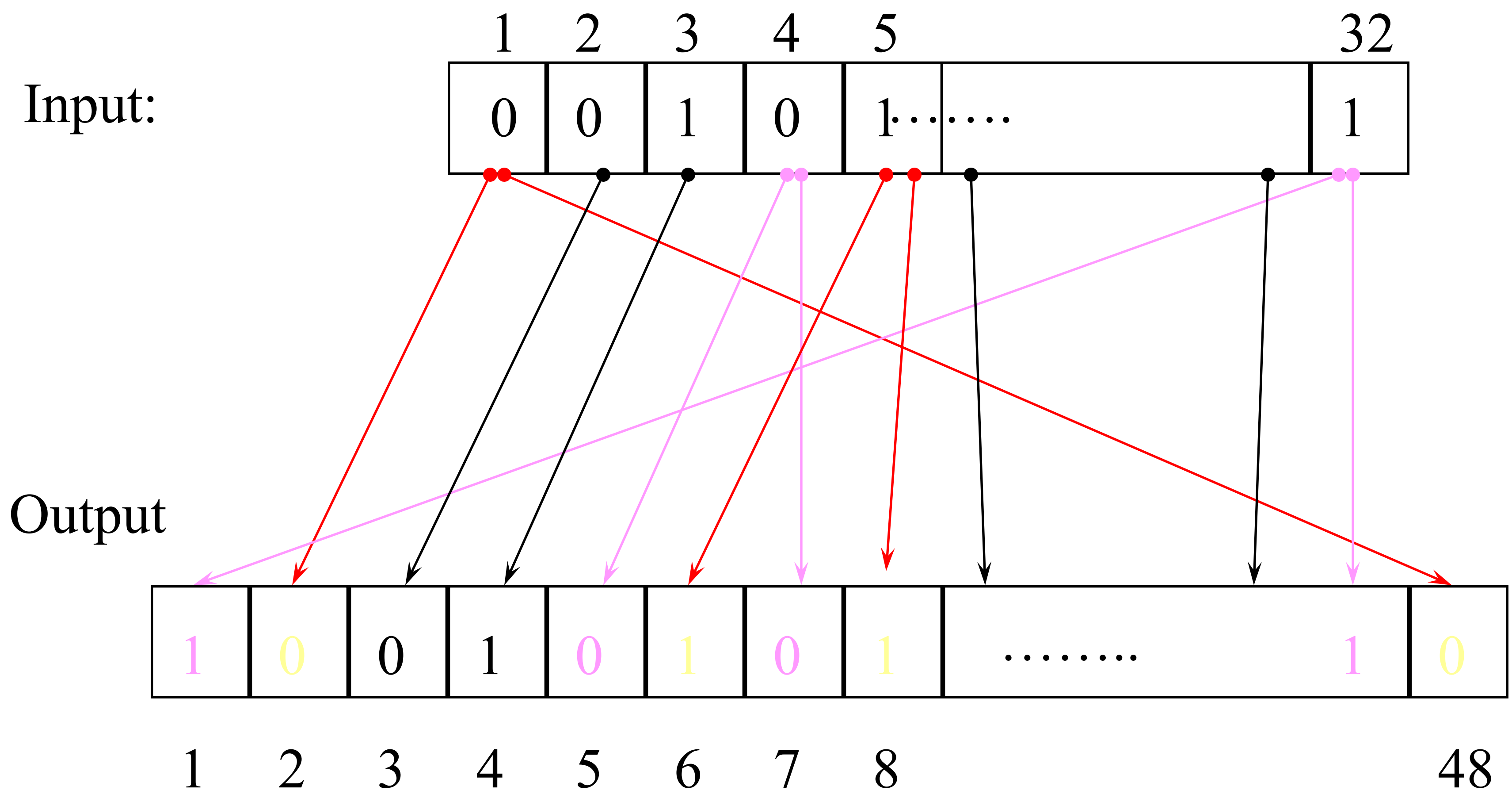


Mangler Function



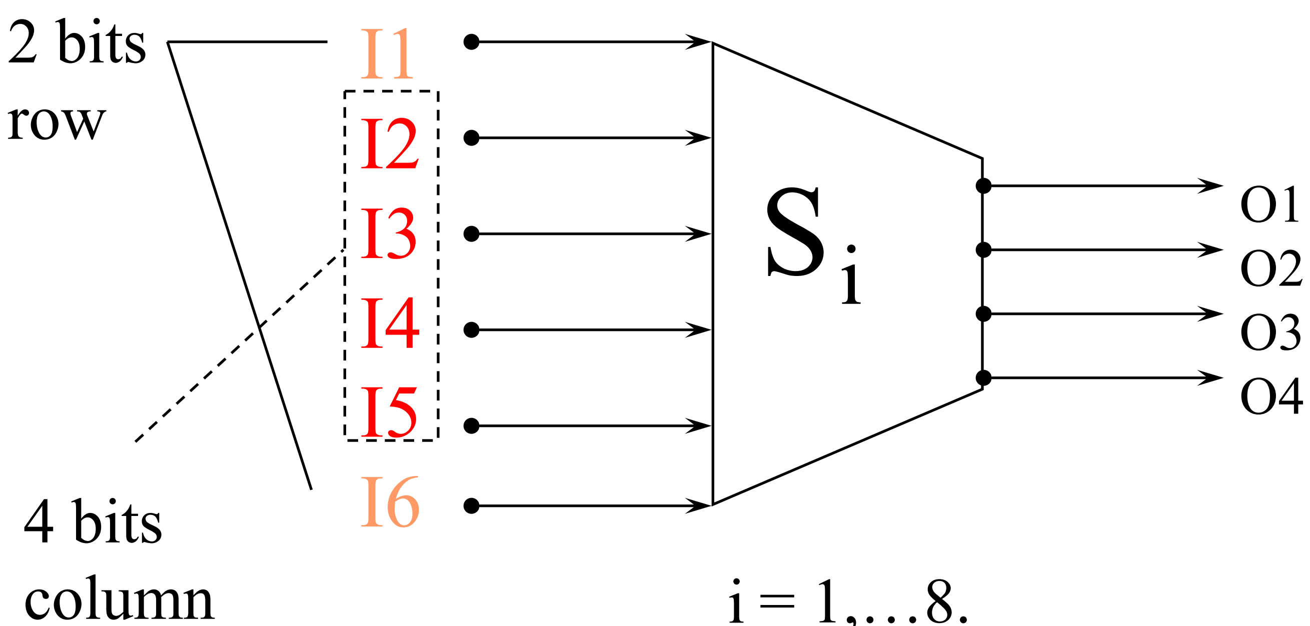
The permutation produces "spread" among the chunks/S-boxes!

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

Abstract view of block ciphers

- Composition of M functions, which are applied to the plaintext and produce the corresponding ciphertext

$$c = E_k(p) = F_M \circ F_{M-1} \circ \cdots \circ F_2 \circ F_1(p) \; \ldots. \quad (1)$$

- Each function is Boolean, either linear or non-linear
- Every linear function in Equation (1) has the form

$$F_i(x_1, x_2, \cdots, x_l) = a_1 \cdot x_1 \oplus a_2 \cdot x_2 \oplus \cdots \oplus a_l \cdot x_l \;, \quad (2)$$

- Every non-linear function in Equation (1) is of the form

$$\mathcal{F}_i(x_1, x_2, \cdots, x_l) = \bigoplus_{j=1}^{l} a_j \prod_{z \in Z} x_z$$

AES

- DES was not too strong a cipher
- Triple-DES has a small block size so it is slow
- NIST issued a call for ciphers in 1997
- 15 candidates were accepted in 1998
- 5 shortlisted in 1999
- Rijndael selected as AES in 2000
- Issued as FIPS PUB 197 in 2001

AES

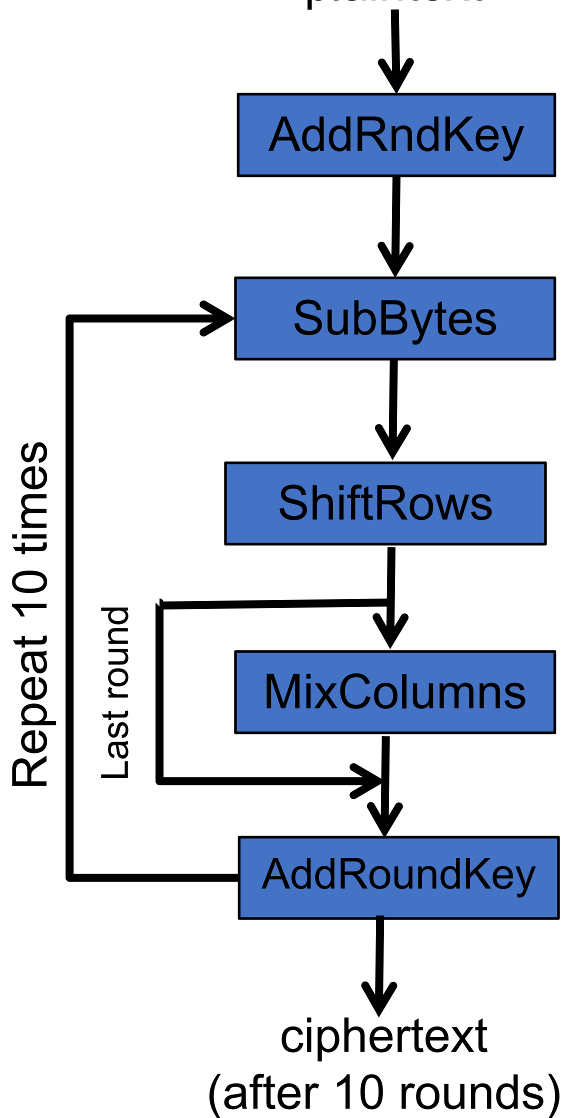


Table 1. Block Cipher Specifications

N	No. of rounds in the cipher
l	Size of round input/output
m	No. of non-overlapped parts in round input/output
w	Size of each part
M	No. of functions invoked during execution
$\mathcal{H}$	Differential Characteristic of the S-boxes

Table 2. Specifications for some Ciphers

	N	**l**	**m**	**w**	**M**	$\mathcal{H}$
AES	10	128	16	8	40	1
CLEFIA	18	128	16	8	74	$2^{1.36}$, $2^{1.02}$
SMS4	32	128	16	8	128	$2^{1.017}$

AES summarized

- Initial key whitening after which each round splits its input into 16 parts of one byte each (w = 8).
- First nine rounds have four operations: AddRoundKey (ARK), SubBytes (SB), ShiftRows (SR), and MixColumns (MC).
- Final round is similar except that it does not have the MixColumns operation.
- ARK, SR, and MC functions are linear, while the SB is a non-linear function.
- A composition of these 4 functions, repeated 10 times, is applied to the plaintext to generate the ciphertext.

AES Details

- Key expanded into array of 32-bit words
  - Each round uses 4 words (round key)
- ARK – a form of Vernam cipher
- SB – simple substitution of each byte
  - Uses one table of 16 x 16 bytes, permutation of all 256 8-bit values
  - each byte of state is replaced by byte indexed by row (left 4-bits) & column (right 4-bits)

  e.g. byte {95} is replaced by byte in row 9 column 5
  which has value {2A}
- MC – each column processed separately, each byte replaced by a value dependent on all 4 bytes in column

RSA

- Alice chooses two large primes p and q, computes n = pq
- Chooses e relatively prime to m = φ(n) = (p-1)(q-1), Euler totient function,
- Publishes (e, n) as public key
- Computes d as inverse of e mod φ(n), (d, m) is the decryption key (Alice keeps it secret)
- Bob does the same
- Bob sends a message m to Alice by using her public key
  - Computes $c = m^e \bmod n$ (assuming $m < n$)
  - Alice decrypts by computing $c^d \bmod n$