



Internal Assessment Test 1 – Mar 2018- SCHEME OF EVALUATION

Sub:	Cryptography, Network Security & Cyber law				
Date:	12 / 03 / 2018	Duration:	90 mins	Sem:	VI
		Marks:	50		

Code:	15CS61
Branch:	ISE

Note: : Answer any 5

Total marks: 50

questions

Question #		Description	Marks Distribution		Max Marks
1	a)	Find the gcd of 81 and 57 using Euclidean Algorithm	3M	3M	10 M
	b)	HMAC Digital signature.	4M 3M	7M	
2	a)	Extended Euclid’s Algorithm Example.	7M 3M	10M	10M
3	a)	Illustrate how encryption is done using DES construction. Diagram	5M 5M	10M	10M
4	a)	RSA algorithm Finding d& e	2M	7M	10 M

		Encryption & Decryption	2M 3M		
	b)	Transposition cipher definition Example	2M 1M	3M	
5	a)	Diffie Hellman Key Exchange	5M	5M	10 M
	b)	Defense strategies and techniques. (4 techniques)	5M	5M	
6	a)	ElGamal algorithm Example.	3M 2M	5M	10 M
	b)	Mono alphabetic cipher (Ceaser cipher) Poly alphabetic cipher: Vignere cipher Hill cipher One-time pad	2M 1M 1M 1M	5M	
7		SHA 1 steps Diagram	5M 5M	10M	10M

Answers

1. A . Find the gcd of 81 and 57 using Euclidean Algorithm. (3)

$$81 = 1(57) + 24$$

$$57 = 2(24) + 9$$

$$24 = 2(9) + 6$$

$$9 = 1(6) + 3$$

$$6 = 2(3) + 0.$$

$$\text{gcd}(81, 57) = 3.$$

- b. Summarize HMAC and digital signature. (3)

HMAC (4)

specified as Internet standard RFC2104

MAC message authentication code

$$\text{MAC} = h(m \parallel k)$$

MAC is like a checksum

Source authentication-sender computes & sends

Msg integrity-checks computed & received MAC

uses hash function on the message:

$$\text{HMAC}_K = \text{Hash}[(K^+ \text{ XOR opad}) \parallel$$

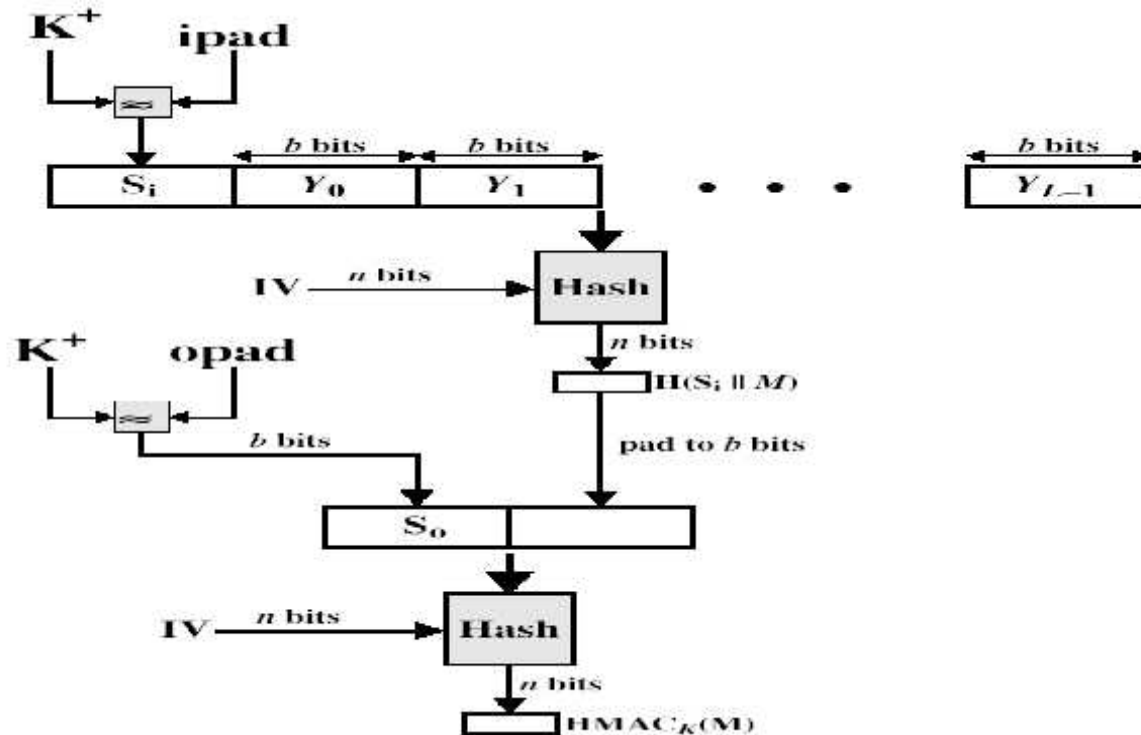
$$\text{Hash}[(K^+ \text{ XOR ipad}) \parallel M]]$$

where K^+ is the key padded out to size

and $opad$, $ipad$ are specified padding constants

overhead is just 3 more hash calculations than the message needs alone

any of MD5, SHA-1, RIPEMD-160 can be used



HMAC Security

know that the security of HMAC relates to that of the underlying hash algorithm

attacking HMAC requires either:

brute force attack on key used

birthday attack (but since keyed would need to observe a very large number of messages)

choose hash function used based on speed verses security constraints

Digital signature (3)

have looked at message authentication

but does not address issues of lack of trust

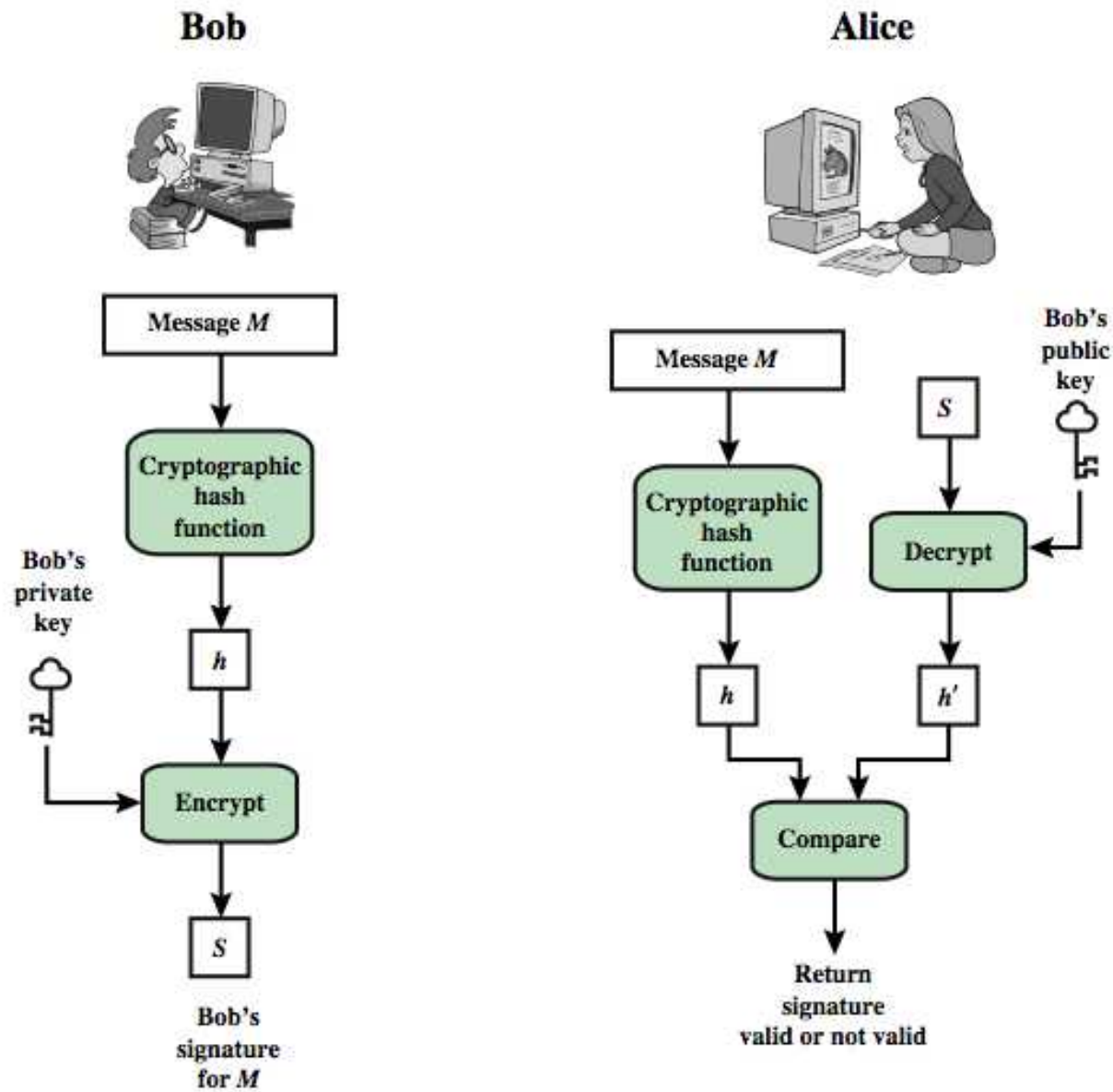
digital signatures provide the ability to:

verify author, date & time of signature

authenticate message contents

be verified by third parties to resolve disputes

hence include authentication function with additional capabilities



2. Label extended Euclid's Algorithm with an example

Used to find mod inverse

Algorithm: (inverse of $c \pmod b$)

computeinverse(b,c)

```
{
old1=1    new1=0
old2=0    new2=1
b'=b      c'=c
r=2
while(r>1){
q=b'/c'
r=b'%c'
temp1=old1-new1*q
old1=new1    new1=temp1
temp2=old2-new2*q
old2=new2    new2=temp2
b'=c'      c'=r
new1*b+new2*c=r
}
return new2 // new2 is the modulo inverse
}
```

E.g:

$b=79$ $c=12$ inverse of $12 \pmod{79}$

after iterations $(-5)*79+33*12=1$

$12*33=1+5*79 \equiv 1 \pmod{79}$

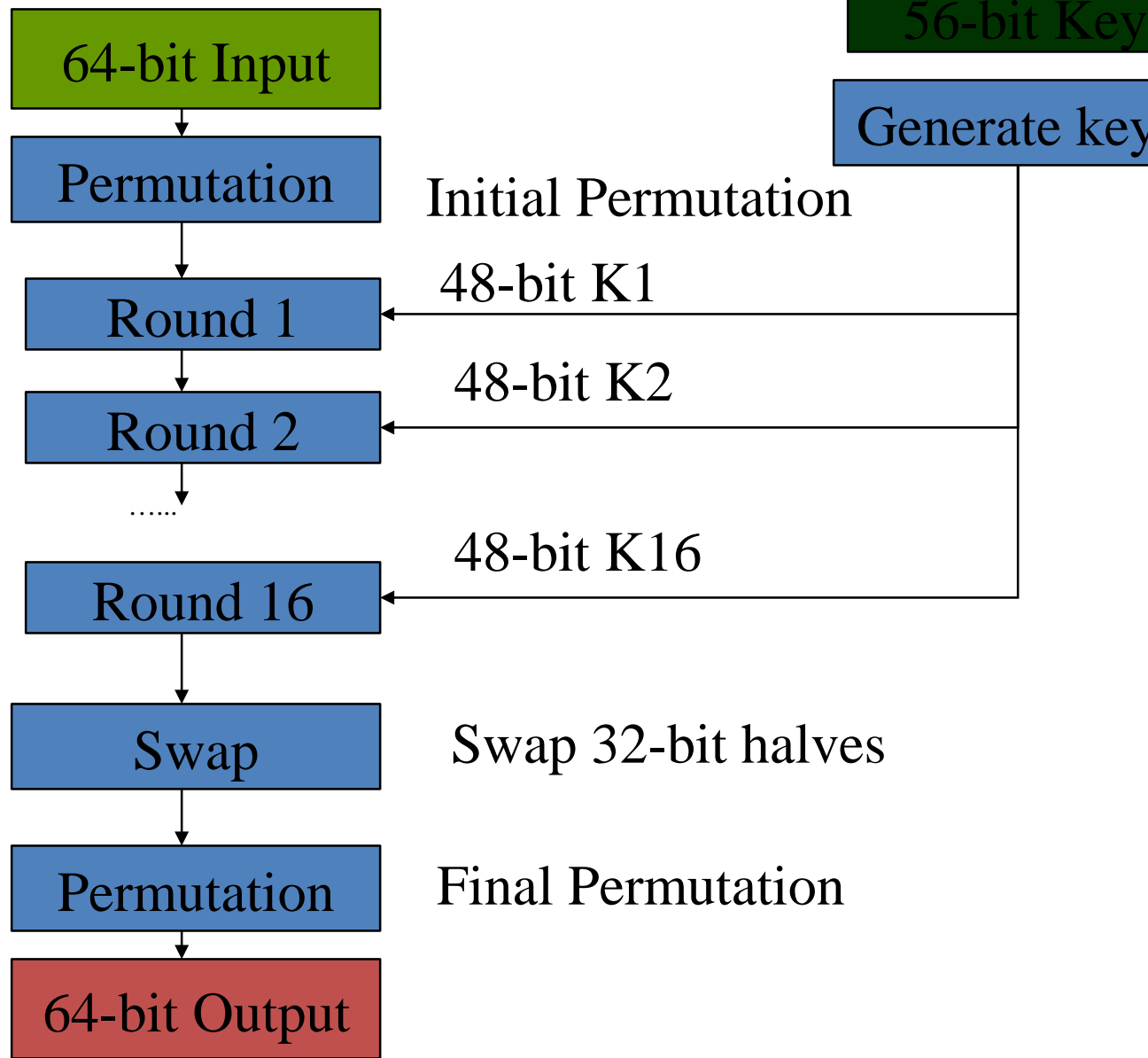
Thus the inverse of 12 modulo 79 is 33

3. Illustrate how encryption is done using DES construction.

DES (Data Encryption Standard)

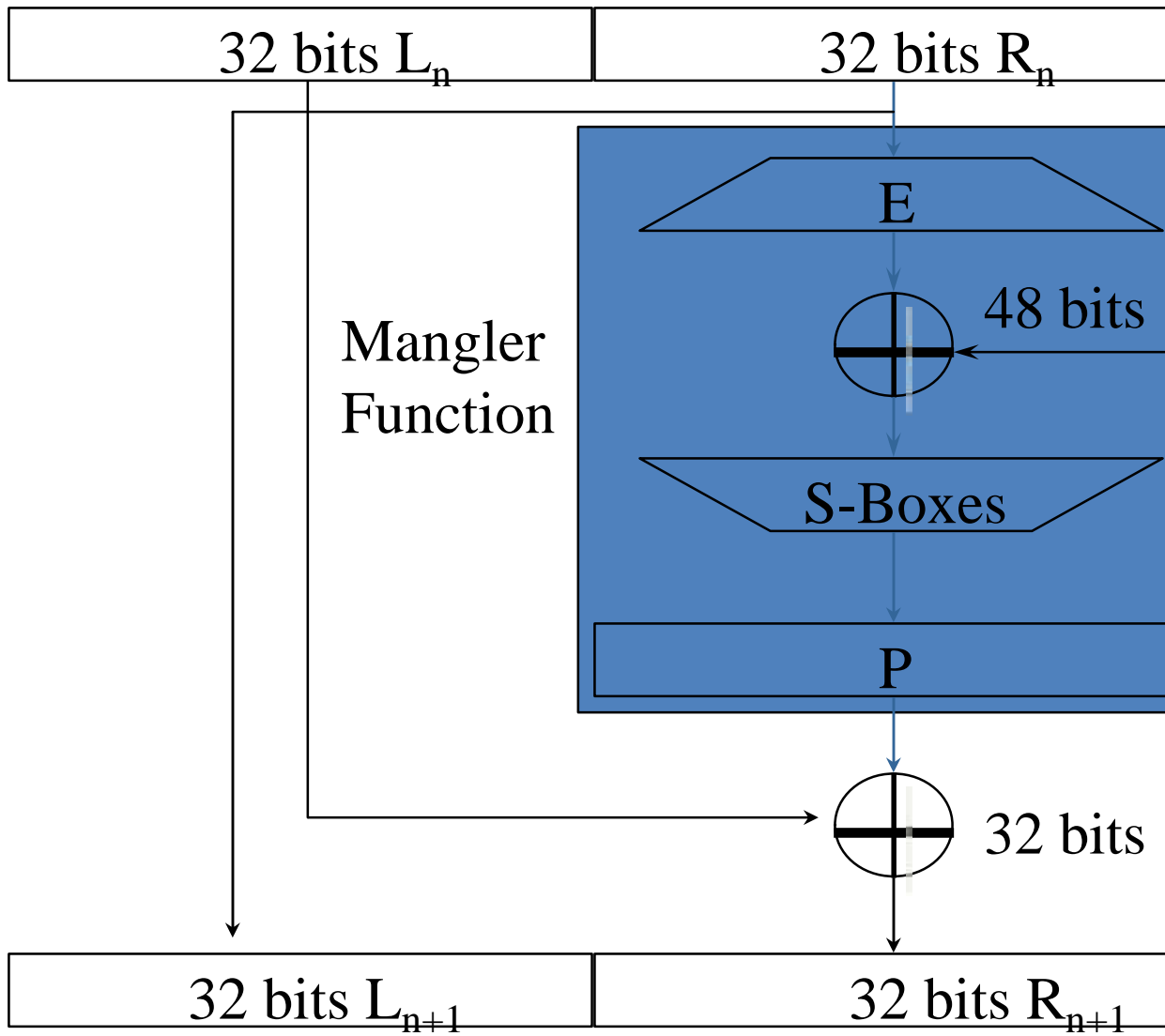
- Published as standard cipher for symmetric key cryptography by NIST(National Institute of Standards and Technology)

- Block size-64 bits • Key size-56/128 bits
- Stages:
- Initial Permutation
- 16 rounds of a given function
- 32 bit left-right swap
- Final permutation
- Stages in DES Encryption:



Single round of DES

One Round
Encryption



● In encryption

● $L_i = R_{i-1}$

- $R_i = L_{i-1} \text{ xor } f(R_{i-1}, K_i)$

- In Decryption

- $R_{i-1} = L_i$

- $L_{i-1} = R_i \text{ xor } f(L_i, K_i)$

Round function $f(R_{i-1}, K_i)$

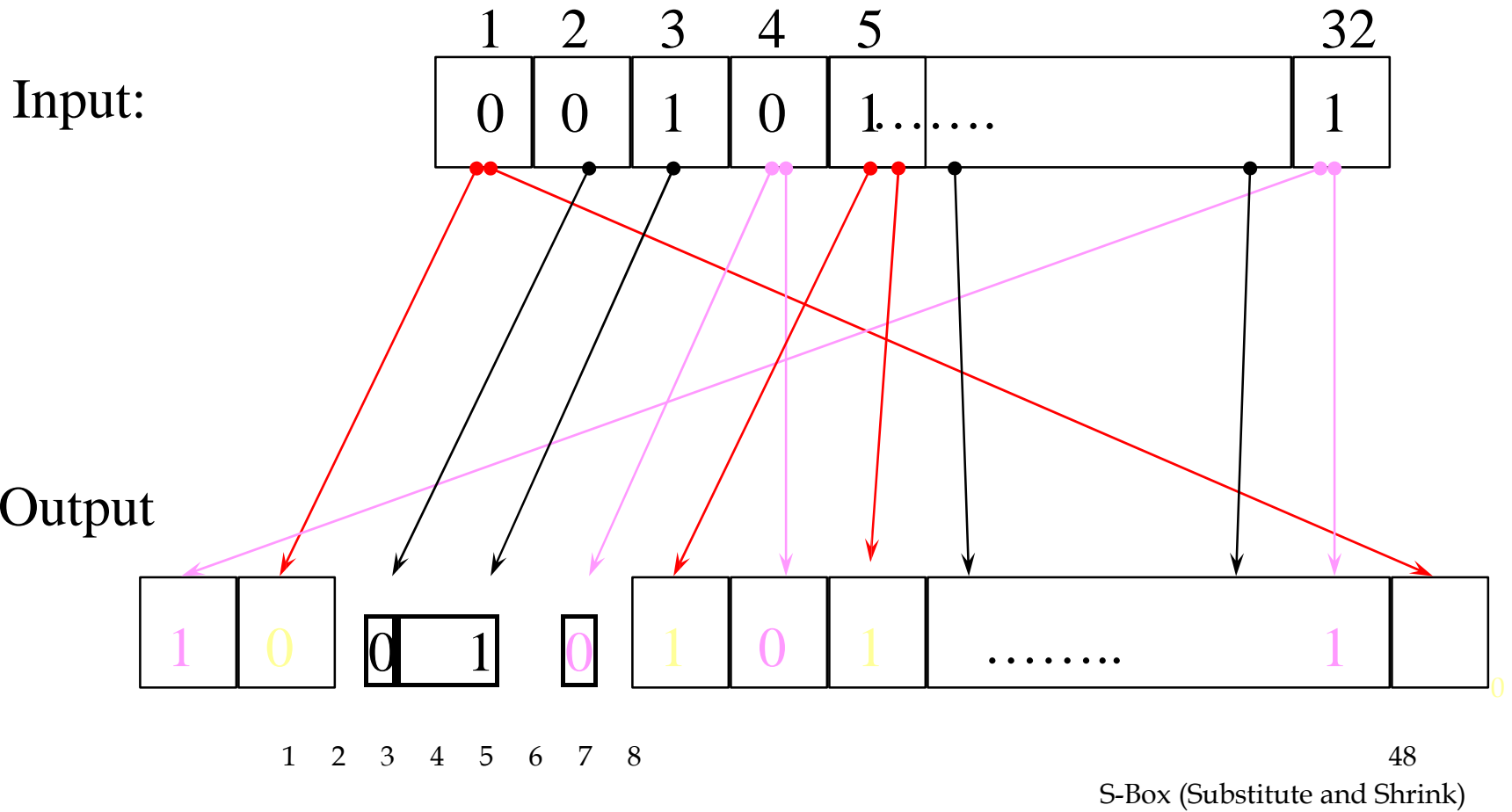
- Expansion-32 bit to 48 bit

- Xor with the round key- 48 bit key

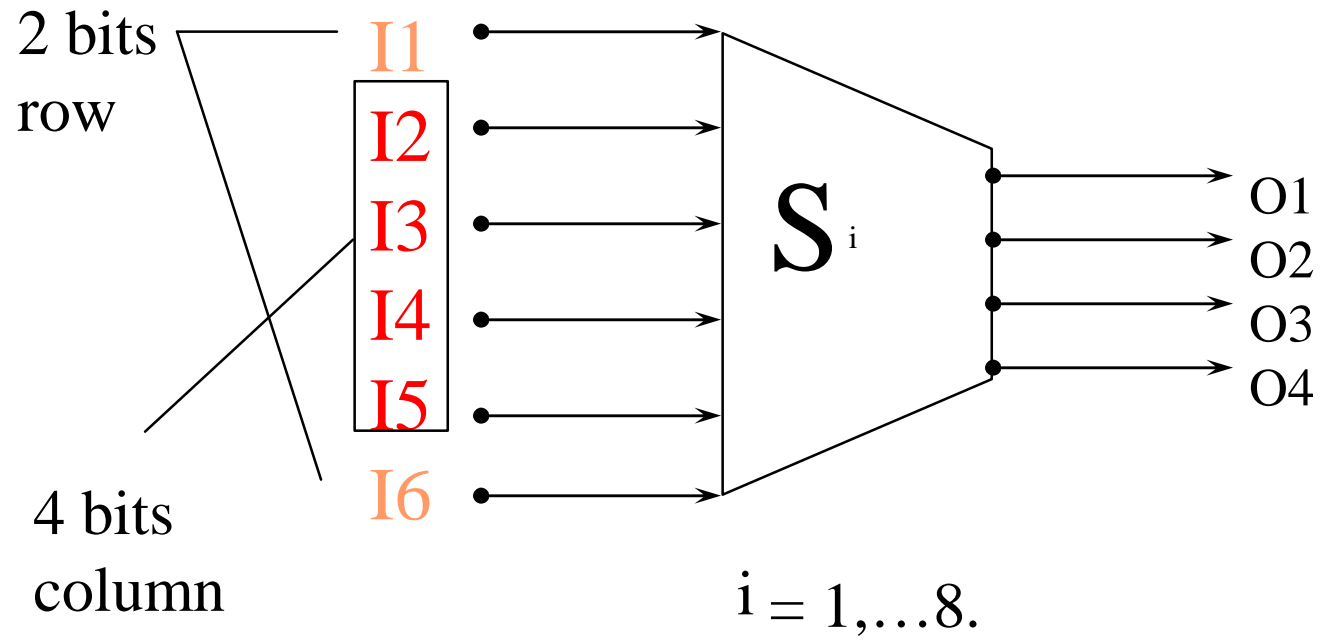
- Substitution

- Permutation

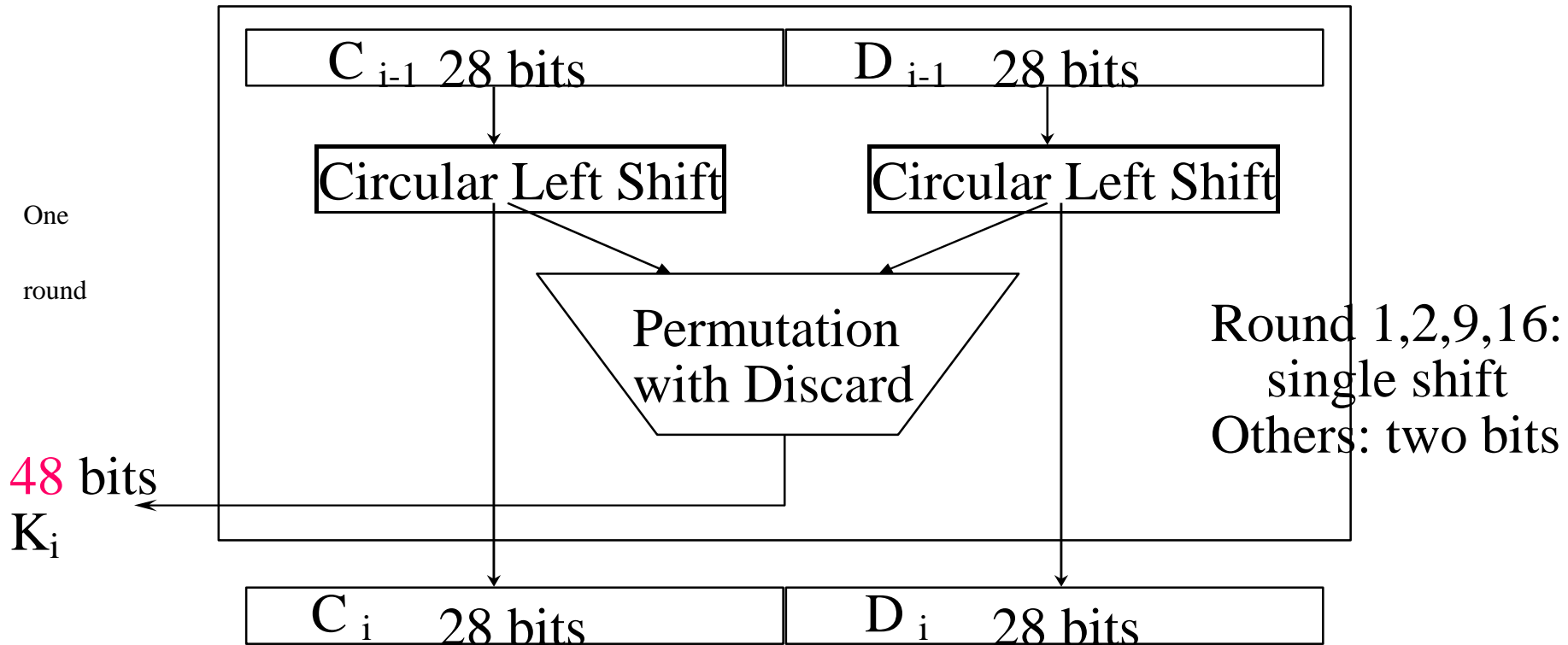
Bits Expansion (1-to-m)



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



Initial Permutation of DES key



4 (a) Perform encryption & decryption for the plaintext $M=7$ using RSA algorithm (Note: Select $p=11$, $q=3$)

- Select primes $p=11$, $q=3$.
- $n = pq = 11 \cdot 3 = 33$

$$\phi = (p-1)(q-1) = 10 \cdot 2 = 20$$

- Choose $e=3$

Check $\gcd(e, p-1) = \gcd(3, 10) = 1$ (i.e. 3 and 10 have no common factors except 1),

and check $\gcd(e, q-1) = \gcd(3, 2) = 1$

therefore $\gcd(e, \phi) = \gcd(e, (p-1)(q-1)) = \gcd(3, 20) = 1$

- Compute d such that $ed \equiv 1 \pmod{\phi}$

i.e. compute $d = e^{-1} \pmod{\phi} = 3^{-1} \pmod{20}$

i.e. find a value for d such that ϕ divides $(ed-1)$

i.e. find d such that 20 divides $3d-1$.

Simple testing ($d = 1, 2, \dots$) gives $d = 7$

Check: $ed-1 = 3 \cdot 7 - 1 = 20$, which is divisible by ϕ .

- Public key = $(n, e) = (33, 3)$

Private key = $(n, d) = (33, 7)$.

This is actually the smallest possible value for the modulus n for which the RSA algorithm works.

Now say we want to encrypt the message $m = 7$,

$$c = m^e \pmod{n} = 7^3 \pmod{33} = 343 \pmod{33} = 13.$$

Hence the ciphertext $c = 13$.

To check decryption we compute

$$m' = c^d \pmod{n} = 13^7 \pmod{33} = 7.$$

- (b) Quote how transposition cipher is used to convert a given message into cipher text.

A transposition or permutation cipher shuffles, rearranges or permutes the bits in a block of plaintext.

plaintext: begin operation at noon

b e g i n

o p e r a

t i o n a

t n o o n

Rows are rearranged as

Row 1->3, 2->5, 3->2, 4->1, 5->4

Resulting matrix on a t r a t i b e g i n o n n o p e

Columns are rearranged as

Columns 1->4, 2->3, 3->1, 4->2

Resulting matrix

a t n o

t i a r

g i e b

o n o n

p e o n

Ciphertext: atnotiargiebononpeon

5.a. Find the session key used by Alice and Bob using Diffie-Hellman key change. (note: prime $g = 23$, primitive root $a=5$, secret integer of $(XA)=4$ & $B(XB)=3$)

1. Alice and Bob agree to use a modulus $p = 23$ and base $g = 5$ (which is a primitive root modulo 23).
2. Alice chooses a secret integer $a = 4$, then sends Bob $A = g^a \bmod p$
 - o $A = 5^4 \bmod 23 = 4$
3. Bob chooses a secret integer $b = 3$, then sends Alice $B = g^b \bmod p$
 - o $B = 5^3 \bmod 23 = 10$
4. Alice computes $s = B^a \bmod p$
 - o $s = 10^4 \bmod 23 = 18$
5. Bob computes $s = A^b \bmod p$

o $s = 4^3 \bmod 23 = 18$

6. Alice and Bob now share a secret (the number 18).

Both Alice and Bob have arrived at the same value s , because, under mod p .

b. List the defense strategies and techniques.

- 1. Accesscontrol-Authentication and Authorization
- Access control is a security technique that can be used to regulate who or what can view or use resources in a computing environment.
- Authentication: assurance that the entity is the one that claims to be
- E.g After login into system using password(Authentication) the job of the access controller is to answer authorization questions
- Whether Kavya is allowed to write into the file CS1561
- Access ctrl decision is based on:
- the subject or principal , kavya
- the object or resource CS1561
- the access mode or operation write

- Authentication/ Authorization
- ID card to enter into college
- Hall ticket to enter into exam hall to write exam
- Firewall: to protect n/w from outside world

2. Data Protection

- i) Confidentiality : assurance that the message is send and received only by authorized persons
- ii) Integrity : The data is not changed during transit
- Is achieved by Encryption, decryption and cryptographic checksum which is achieved by shared secret key
- the sender computes the checksum using one way function and sends the checksum along with the message and the receiver computes the checksum and cross check with the received check sum

3. Prevention and Detection

- Prevention strategies
- access control, encryption
- Detection strategies
- cryptographic checksum
- S/w security

- Black box testing: when the source code of the pgm is not easily available
- The goal is to determine whether the s/w has been carefully designed to handle unexpected or malicious i/p.
- White box testing: the security engineer has access to source code & can perform more elaborate testing
- Intrusion prevention technique- false +ve false -ve
- Intrusion detection technique- Anti virus products are signature based

4. Response, Recovery and Forensics

- Once an attack behavior is identified response measures should be taken and the system should be recovered to working state
- Cyber forensics is an emerging technique used to identify attacker using the fingerprint they left

6.a Enumerate ElGamal encryption and decryption with an example.

public-key cryptosystem related to D-H

uses exponentiation in a finite field

with security based difficulty of computing discrete logarithms, as in D-H

Setting up ElGamal

Let p be a large prime

By “large” we mean here a prime rather typical in length to that of an RSA modulus

Select a special number **g**

The number **g** must be a **primitive element** modulo **p**.

Choose a private key **x**

This can be any number bigger than 1 and smaller than **p-1**

Compute public key **y** from **x**, **p** and **g**

The public key **y** is **g** raised to the power of the private key **x** modulo **p**. In other words:

$$y = g^x \text{ mod } p$$

Step 1: Let $p = 23$

Step 2: Select a primitive element $g = 11$

Step 3: Choose a private key $x = 6$

Step 4: Compute $y = 11^6 \pmod{23}$
 $= 9$

Public key is 9

Private key is 6

ElGamal encryption

The first job is to represent the plaintext as a series of numbers modulo p . Then:

1. Generate a random number k
2. Compute two values C_1 and C_2 , where

$$C_1 = g^k \pmod{p} \quad \text{and} \quad C_2 = My^k \pmod{p}$$

3. Send the ciphertext C, which consists of the two separate values C_1 and C_2 .

To encrypt $M = 10$ using Public key **9**

1 - Generate a random number $k = 3$

2 - Compute $C_1 = 11^3 \bmod 23 = 20$

$$C_2 = 10 \times 9^3 \bmod 23$$

$$= 10 \times 16 = 160 \bmod 23 = 22$$

3 - Ciphertext $C = (20, 22)$

ElGamal decryption

$$C_1 = g^k \bmod p \quad C_2 = My^k \bmod p$$

$$M = C_1^{p-x-1} C_2 \bmod p$$

$$C_1=20 \quad c_2=22 \quad p=23 \quad x=6$$

$$20^{23-6-1} * 22 \bmod 23 = 10$$

To decrypt $C = (20, 22)$

1 - Compute $20^6 = 11^{18} \bmod 23 = 16$ ($3 \cdot 6 = 18$)

2 - Compute $22 / 16 = 10 \bmod 23$

3 - Plaintext = 10

b. Infer substitution cipher and its types with suitable example.

Types

- Mono alphabetic cipher
- Poly alphabetic cipher
- Caesar cipher
- earliest known substitution cipher
- by Julius Caesar
- It is a mono alphabetic cipher because each letter is always substituted for another unique letter
- first attested use in military affairs replaces each letter by 3rd letter on example:

meet me after the toga party
PHHW PH DIWHU WKH WRJD SDUWB

➤ can 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 = I N 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 =
O U T

➤ mathematically give each letter a number

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 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25

➤ then have Caesar (rotation) cipher as: $c = E(k, p) = (p + k) \bmod (26)$ $p = D(k, c) = (c - k) \bmod (26)$



ATTACK



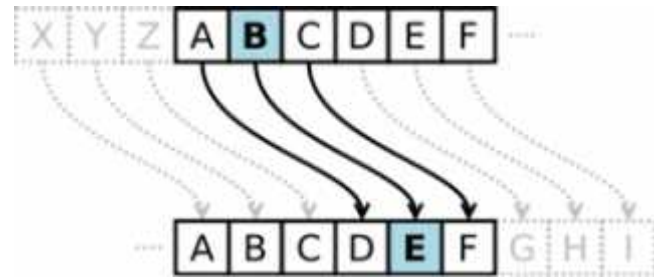
Soldier carrying the message "ATTACK"



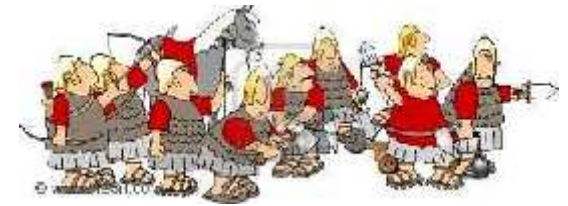
Message may be intercepted by enemy



ATTACK

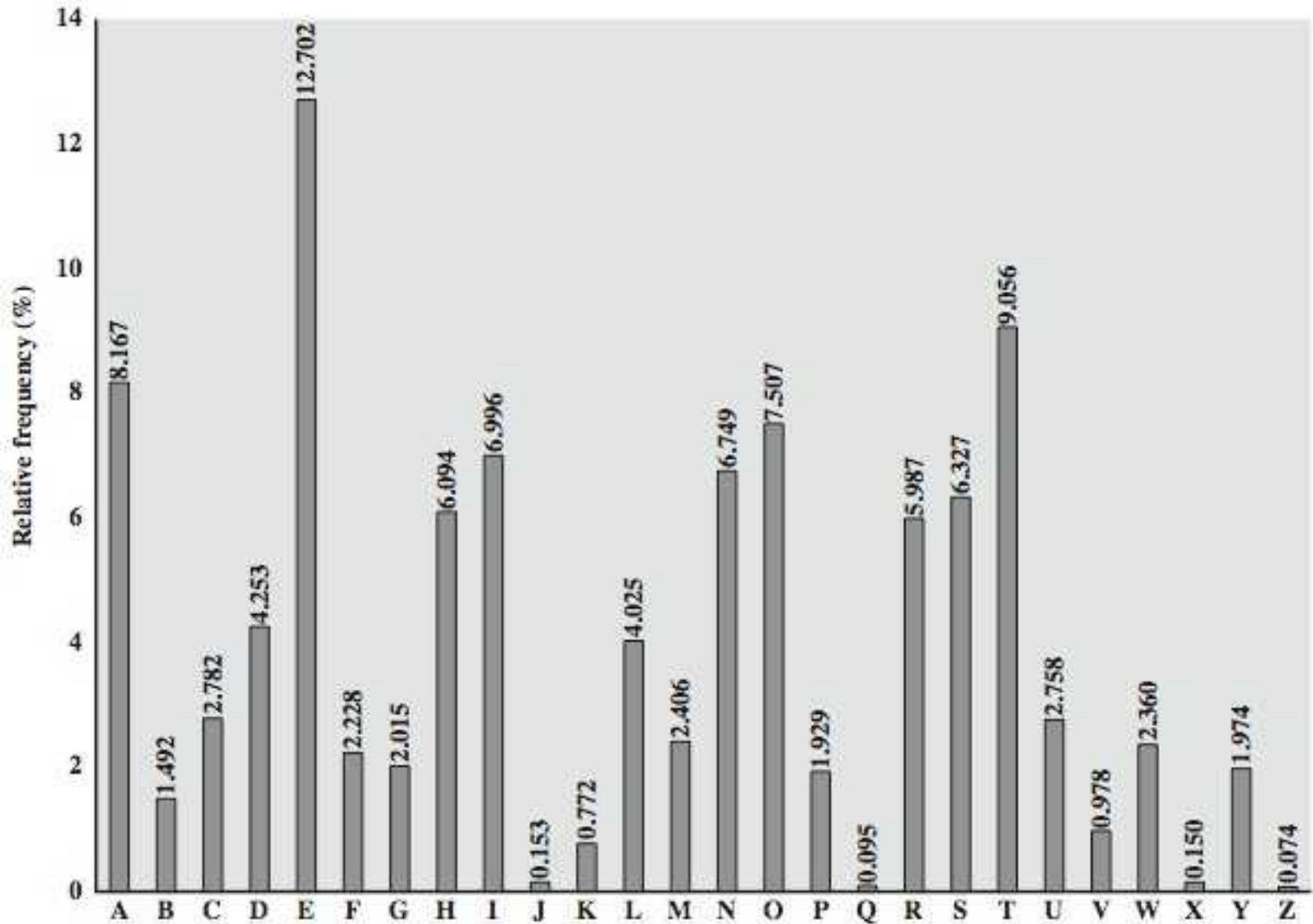


Soldier carrying the encrypted message "DWWDFN"



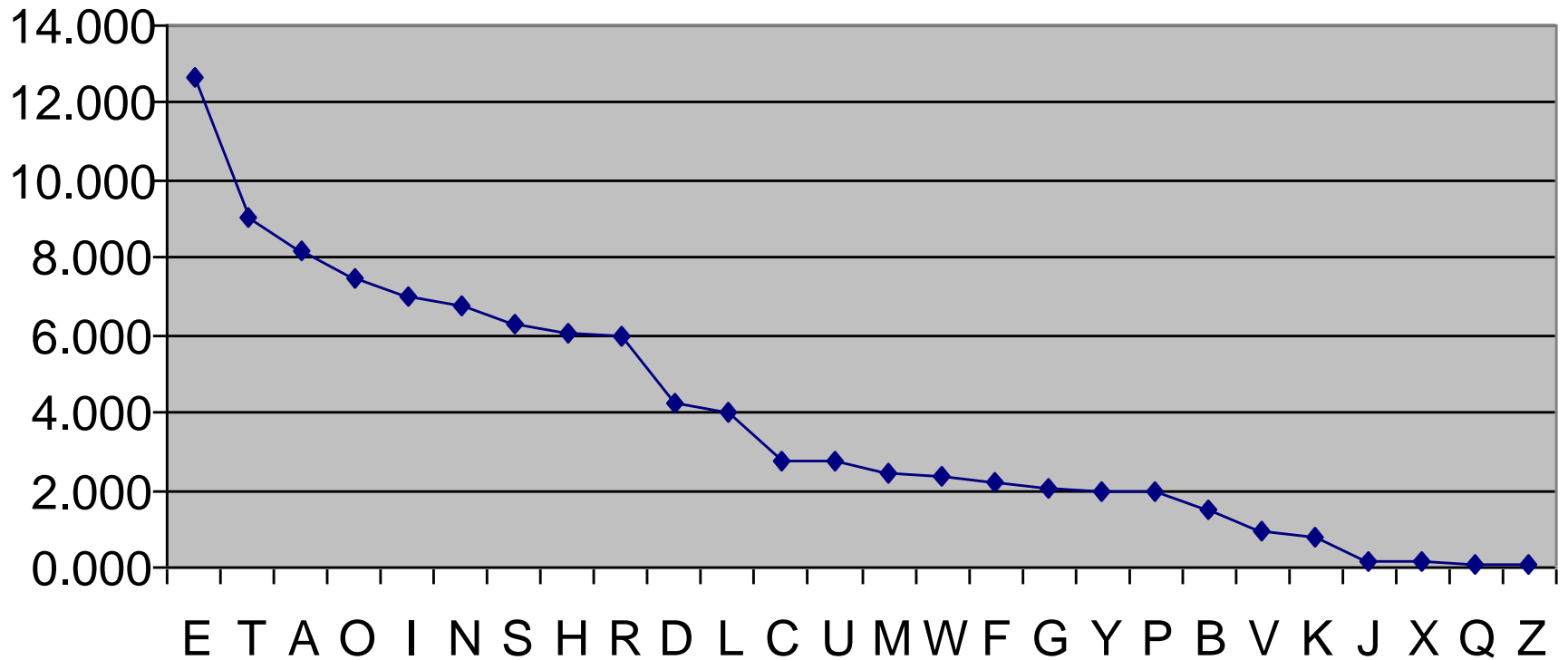
The encrypted message looks random and meaningless

English Letter Frequencies

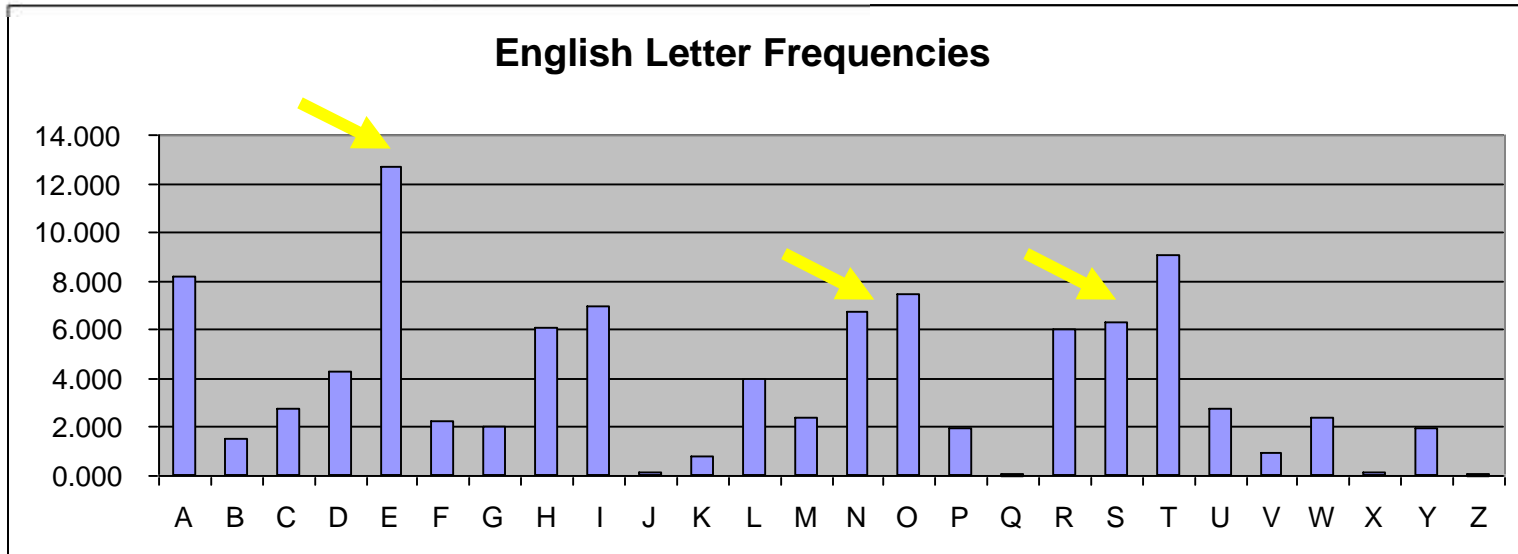


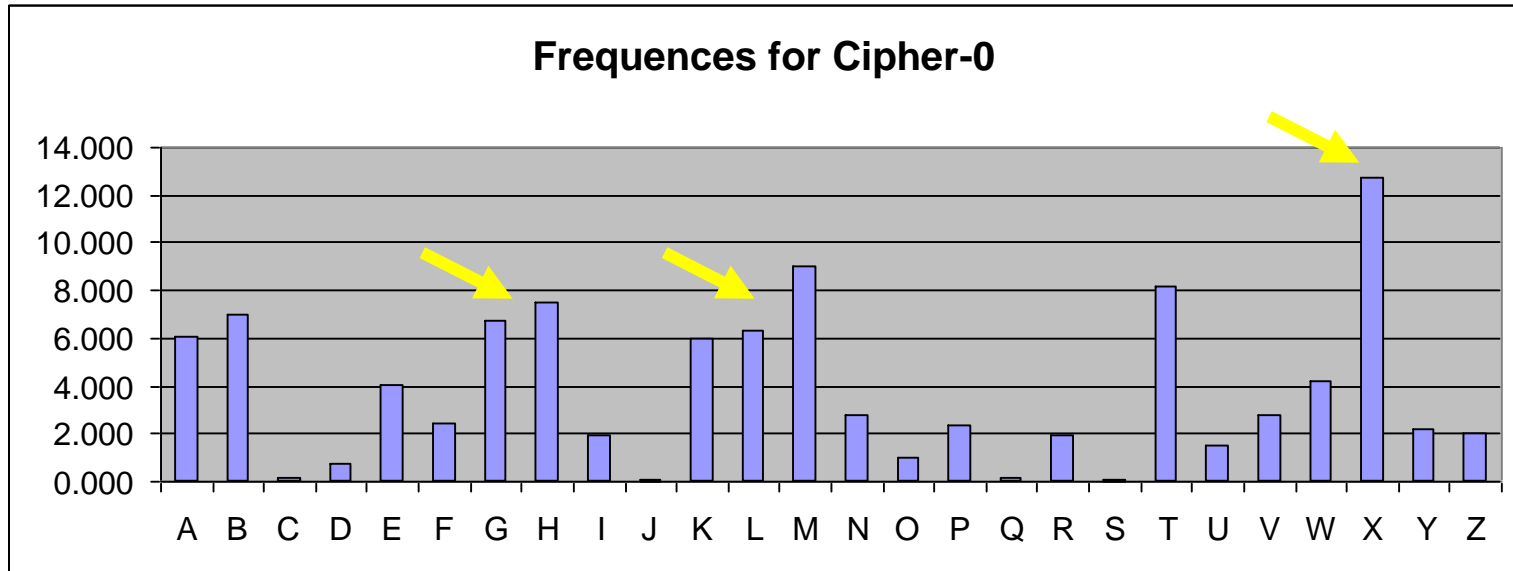
English Letter Frequencies

Sorted Relative Frequencies



What kind of cipher is this?





- key concept - monoalphabetic substitution ciphers do not change relative letter frequencies
- discovered by Arabian scientists in 9th century
- calculate letter frequencies for ciphertext
- compare counts/plots against known values
- if caesar cipher look for common peaks/troughs
 - peaks at: A-E-I triple, N-O pair, R-S-T triple troughs at: J-K, U-V-W-X-Y-Z
- for monoalphabetic must identify each letter
 - tables of common double/triple letters help (digrams and trigrams)

➤ amount of ciphertext is important – statistics!

➤ Example cryptanalysis

➤ given ciphertext:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

➤ count relative letter frequencies (see text)

➤ given ciphertext:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

➤ 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

Polyalphabetic cipher

In a polyalphabetic cipher, the ciphertext corresponding to a particular character in the plaintext is not fixed.

Vignere Cipher

- ▶ simplest polyalphabetic substitution cipher
- ▶ effectively multiple caesar ciphers
- ▶ key is multiple letters long $K = k_1 k_2 \dots k_d$
- ▶ i^{th} letter specifies i^{th} alphabet to use
- ▶ use each alphabet in turn
- ▶ repeat from start after d letters in message
- ▶ decryption simply works in reverse
- ▶ Attacker can deduce the key as it is repeated
- ▶ Example:
- ▶ write the plaintext out

- write the keyword repeated above it
- use each key letter as a caesar cipher key
- encrypt the corresponding plaintext letter
- eg using keyword *deceptive*

key: deceptivedeceptivedeceptive

3 4 2 4 15 19 2 1 4

plaintext: wearediscoveredsaveyourself ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ

Hint: shift w by 3 e by 4 a by 2....

Hill cipher

Invented by Lester Hill in 1929.

■ Inputs : String of English letters, A,B,...,Z.


An $n \times n$ matrix \mathbf{K} , with entries drawn from $0,1,\dots,25$.

(The matrix \mathbf{K} serves as the secret key.) Divide the input string into blocks of size n . Identify A=0, B=1, C=2, ..., Z=25.

■ Encryption: Multiply each block by \mathbf{K} and then reduce mod 26.

■ Decryption: multiply each block by the inverse of \mathbf{K} , and reduce mod 26.

■ $c = p \cdot k$



■ $p=c_k-1$

- **One-time pad**
- Arbitrarily long, random and non-repeating sequence of character=key
- called a One-Time pad (OTP)
- In Vigenere cipher the same key is repeated which is susceptible to the attacker
- But in one-time pad the key is not repeated
- is unbreakable since ciphertext bears no statistical relationship to the plaintext
- problems in key generation & safe distribution of key

7. Illustrate how SHA 1 is used to generate hash value.

SHA originally designed by NIST & NSA in 1993

was revised in 1995 as SHA-1

US standard for use with DSA signature scheme

standard is FIPS 180-1 1995, also Internet RFC3174

the algorithm is SHA, the standard is SHS

based on design of MD4 with key differences

produces 160-bit hash values

2005 results on security of SHA-1 raised concerns on its use in future applications

Secure Hash Algorithm Std-SHA-512

Digest Length=**160 bit**

I/P Text=512 bit

Sub Block size=32bit

$512/32=16$ total Sub blocks

No. Of Rounds=4

Iteration per round=**20**

Chaining Variable = $5 \times 32 = 160$

$K[t]$ constant= *Where $t=0$ to 79*

O/P-> four 32 bit blocks

1. **Padding**: Length of the message is 64 bits short of multiple of 512 after padding.
2. **Append** a 64-bit **length** value of original message is taken.
3. **Divide the input into 512-bit blocks**
4. **Initialise IV** 5-word (160-bit) buffer (A,B,C,D,E) to

(A=01 23 45 67,

B=89 AB CD EF,

C=FE DC BA 98,

D=76 54 32 10,

$E=C3\ D2\ E1\ F0)$

5. **Process Blocks** now the actual algorithm begins. message in 16-word (512-bit) chunks:

Copy IV into single register for storing temporary intermediate as well as the final results.

Divide the current 512-bit blocks into 16 sub-blocks, each consisting of 32 bits.

- ❑ Has No. Of Rounds=4, each round consisting of 20 *bit/step iteration* operations on message block & buffer
- ❑ expand 16 words into 80 words(20*4) by mixing & shifting. $K[t]$ constant= *Where $t=0$ to 79*
- ❑ Form new buffer value by adding output to input.

6. output hash value is the final buffer value

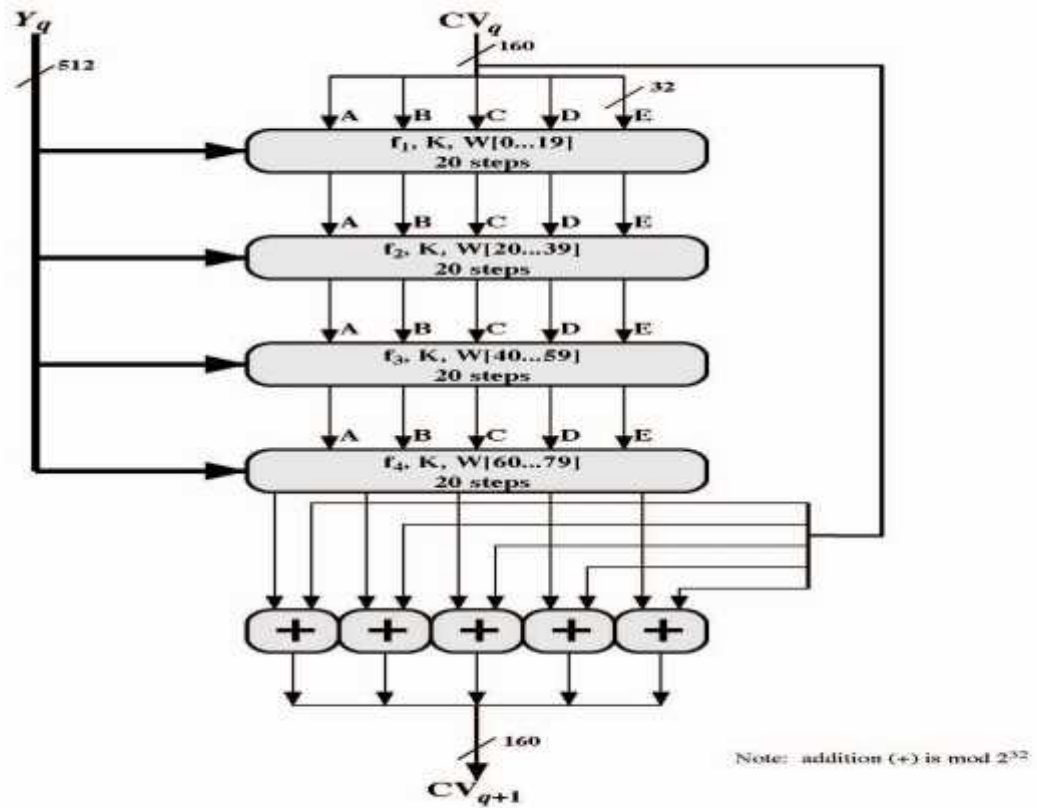
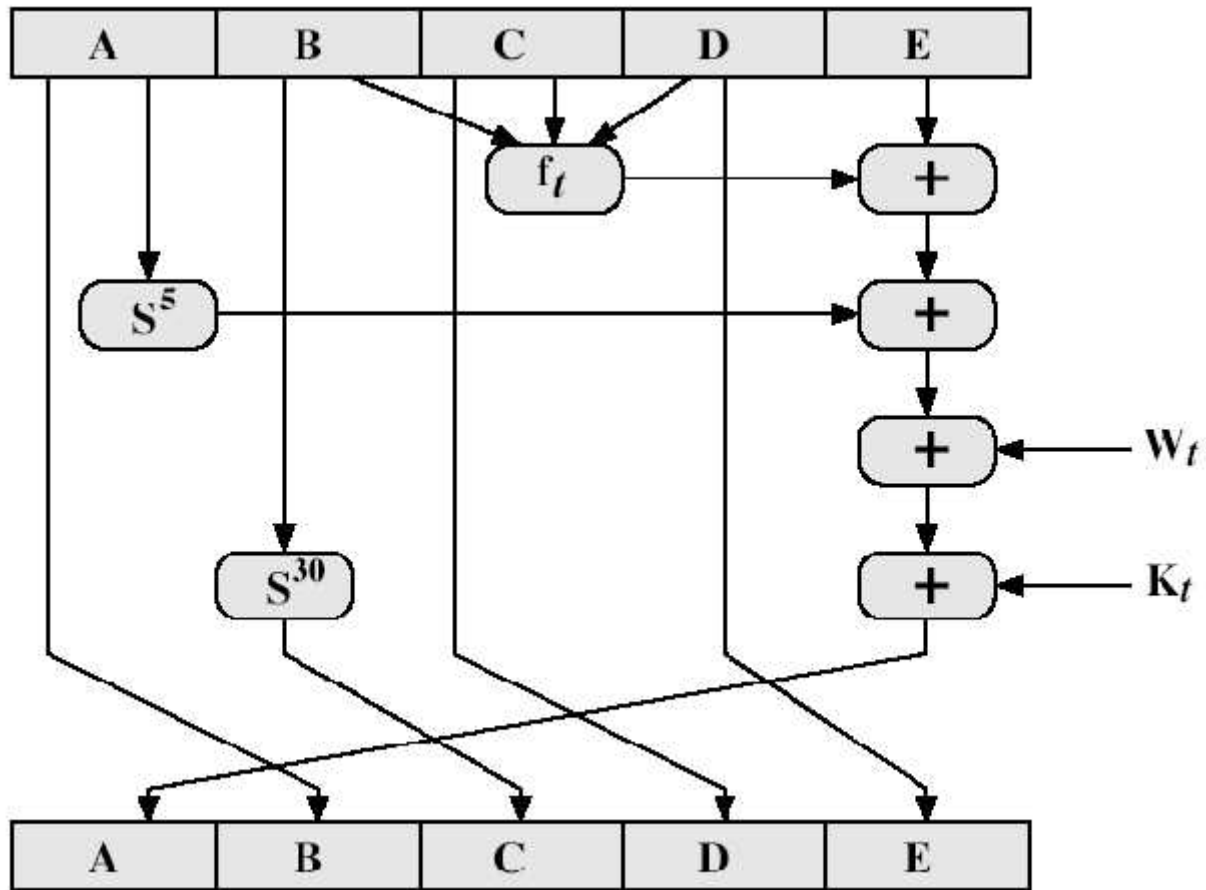


Figure 12.5 SHA-1 Processing of a Single 512-bit Block (SHA-1 Compression Function)

SHA-1 Compression Function



$ABCDE = (F[t] + E + S^5(A) + W[t] + K[t]), >>>$ Shift right by 1 bit for next iteration

each round has 20 steps which replaces the 5 buffer words thus:

$$(A, B, C, D, E) \leftarrow (E + f(t, B, C, D) + (A \ll 5) + W_t + K_t), A, (B \ll 30), C, D$$

ABCDE refer to the 5 words of the buffer

t is the step number

$f(t, B, C, D)$ is nonlinear function for round

W_t is derived from the message block

K_t is a constant value

S^t circular left shift of 32 bit sub-block by t bits

Process $F(t)$ in each SHA-1 round

□ where g can be expressed as:

ROUND 1: $(b \text{ AND } c) \text{ OR } ((\text{NOT } b) \text{ AND } (d))$ same as MD5

ROUND 2: $b \text{ XOR } c \text{ XOR } d$

ROUND 3: $(b \text{ AND } c) \text{ OR } (b \text{ AND } d) \text{ OR } (c \text{ AND } d)$

ROUND 4: $b \text{ XOR } c \text{ XOR } d$

Creation of 80-word input W_t

Adds redundancy and interdependence among message blocks

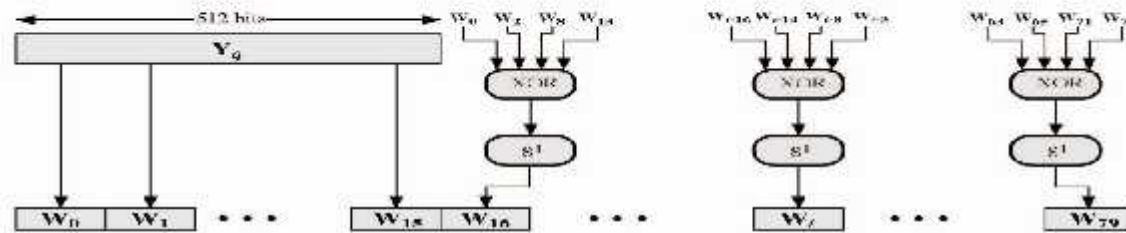


Figure 12.7 Creation of 80-word Input Sequence for SHA-1 Processing of Single Block

