13/7/2020

SWE 3002.

1. Fundamentals of security.

Security:

Goals of Information security.

→ C I A → Availability. (reliable & timely access user can use by authorised)
↓ ↳ Integrity (safe guard the accuracy & completion of access)
Confidentiality
(securing data)
(prevent the unauthorised users)
(disclosure of Information)

Security:

→ Internet was initially designed for connectivity
→ fundamental aspects of information must be protected
→ We can't keep ourselves isolated from the Internet.

prevention
reaction
detection

Attacks on different layers.

Open system Interconnection
OSI reference model

Open for communication with other model

Application	Layer 7: DNS, DHCP, HTTP, FTP, IMAP, LDAP, NTP, Radius, SSH, SMTP, SNMP, Telnet, TFTP.	[DNS Poisoning, Phishing, SQL injection, Spam scam]
Presentation		
~~Presentation~~ Session	Layer 5: SMB, NFS, Socks	
Transport	Layer 4 TCP. UDP.	[TCP attacks, Port attacks, SYN flooding, Sniffing]
Network	Layer 3: IPv4, IPv6...	
Data link	Layer 2: PPTP, Token ring	
Physical	ARP spoofing, MAC flooding.	

Intermediate

(COMMON TYPES OF ATTACKS)

→ Ping sweeps & port scans
→ Sniffing (capture the packet travel through network)
→ Man-in-the-middle attack.
→ Spoofing } end to end session layer
→ Hijacking }
→ Denial of service (DOS) & Distributed DOS (DDOS)
all service layers

14/7/2020

OSI Security architecture:

- Architecture focuses on security attacks, mechanisms, and services.

1) Security attack → Passive attack
→ Active attack
2) Security mechanism
3) Security services

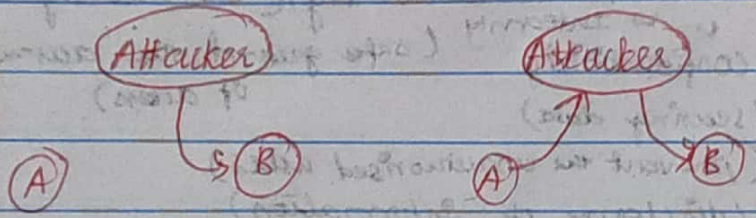
Security attack → action on our security

- Release of message content
Traffic analysis

passive attack

Active attack

Masquerade Replay



Modification of message DOS

attacker will send lot of request so it takes lots of time respond by user due to traffic it ends up in deny request

Eg: Viruses, malware, hacker concentrate on server side rather than on client side.

CCL

Security mechanism:

(X) design, detection, prevention, recover (X)

mechanism

Eg: Cryptography for security

→ Encipherment → data integrity → Digital signature → Authentication exchange → Traffic padding → Routing control → Notarization → Access control.

15/07/2020

Information System Security

Elementary cryptography:

⇒ classical method used in olden days.

Cryptography: Study of science

(plain text) Eg: message sent to someone which are converted into cipher text

ciphers:

(Shared) → Symmentric algo

Eg: DES

encript & decript use same key.

KE = KD

key encript key decript

Eg: Block ciphers: DES, AES, PRESENT etc...

Stream ciphers: As, Grain ...

(Unshared) Assymentric algo:

encript & decript are different

Eg: RSA

KE ≠ KD.

Eg: RSA, ECC

/

Electrical curve cryptography.

Basic Terminology:

→ plain text (original message to be encrypted) → Cipher text (msg converted to encrypted (unreadable)) → Encryption algorithm ↙ original message ↘ secret key.

→ encryption → decryption → decryption algorithm.

→ secret key: same key used for encryption & decryption → cipher or cryptographic system → Cryptography.

→ Crypt analysis → Cryptology. ↙ Cryptography + Cryptanalysis

Crypto System:

five tuple (P, C, K, E, D)

possible plain text | keyspace | cipher text | Encryption function | decryption function

small p, x. p = x.

$\forall K \in K$

Encryption rule: $\exists e_x \in K$ and

decryption rule: $\exists d_k \in D$

such that:

$(e_K : P \to C), (d_k : C \to P)$ and

$\forall x \in P, d_x(e_x(x)) = x$

17/7/2020 ISS

Classical ciphers → Stream, Block

plaintext is viewed as a sequence of elements
Eg: bits of characters

Stream
→ Substitution cipher: replacing each element of the plaintext with another element.
→ Transposition cipher (permutation): rearranging the order of the elements of the plaintext.
product cipher:

Substitution ciphers - Caesar cipher

plain text: $P = \{0, 1, 2, 3, \cdots, 25\}$
cipher set: $C = \{0, 1, 2 \cdots 25\}$
keyspace: $K = \{ \text{"} \}$
Encryption rule: $e_k(x) = (x+k) \bmod 26$,
decryption rule: $d_k(x) = (x-k) \bmod 26$
where $k \in K$ and $x \in P$

$Y_1, Y_2 \in C$
$d_k(Y_1) \neq d_k(Y_2)$.

Decryption rule:
$d_k(x) = (x-k) \bmod 26$

Encryption rule:
$e_k(x) = (x+k) \bmod 26$

20/07/2020 ISS

Crypt analysis:

→ Easy to compute →
(convert plain to text to cipher)

→ Brute force attack."

Key value shifted 0 to 25 (26) suits and try all possible numbers

→ Each k results in a unique mapping
$e_k : P \to C$ and $d_k : C \to P$.
→ The mappings are injective (one to one).

$y_1, y_2 \in C$

$d_k(y_1) \neq d_k(y_2)$

Cryptanalyze, assuming a caesar cipher:

Ciphertext: "LOMBEBSDISCKCCDBYXQKCSDC GOKUOCDVSXU"

k = 1

OMBEBSDISCKCCDBYXQKCDCGOKUOCDVSXUC

k = 2

MBEBSDISCKCCDBYXQKCDCGOKUOCDVSXUCO

k = 3

BEBSDISCKCCDBYXQKCDCGOKUOCDVSXUCOM

✓ k = 16

Monoalphabetic Substitution Cipher:

- Shuffle the letters & map each plaintext letter to a different random cipher text.
- we have a total of $26! = 4 \times 10^{26}$ keys.
- with so many keys, its secure against brute-force attacks.
- But not secure against some cryptanalytic attacks.
- problem is language characteristics.

n-0, n-1, n-2, ... n-26.

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

plain text : ATTACKATDAWN.

cipher text : ZXXZAZXKZRY.

How to Brute force attacks.

We should increase the size of length in ciphertext by including space or special characters

Eg: count relative letter frequencies.
guess {P, Z} = {e, t}

- Of double letters, ZW has highest frequency, so guess ZW = th and hence ZWP = the.
- proceeding with trial and error finally get:

It was disclosed yesterday that several informal but direct contacts have been made with political representative of

21/07/2020.

→ the key length is equal to length of our message.
→ Choose truly random autogenerated key (symbol, num, ...)
→ key used only one time after it destroyed by sender & receiver.

One time pad cipher : 10/26/2 (modulo)

A	B	C	D	E	F	G	H	I	J	K	L
0	1	2	3	4	5	6	7	8	9	10	11
M	N	O	P	Q	R	S	T	U	V	W	X
12	13	14	15	16	17	18	19	20	21	22	23
Y	Z										
24	25										

pattern

That's y it is strong & unbreakable.

A⊕B we main truth table to analyze the pattern.

Calculate Encryption & decryption using onetimepad.

Eg: plain text : HELLO.
key : DGHBC

plain. Text =	H	E	L	L	O
	7	4	11	11	14
key	D	G	H	B	C
	3	6	7	1	2

(11+1) mod 26
12 mod 26 = 12 (M)
(14+2) mod 26
16 mod 26 = 16 (Q)

"ciphertext : KKSMQ."

Common formula:
(p+key) mod 26
(7+3) mod 26
10 mod 26 = 1
(4+6) mod 26
10 mod 26 = 10
(11+7) mod 26
18 mod 26 = 1

c.t = K K S m q

key = d g h b c

decryption : $\boxed{(c.t - key) \bmod 26}$

$(K - d) \bmod 26$

$(10 - 3) \bmod 26$

$7 = H$

$(10 - 6) \bmod 26$

$4 = E$

$(18 - 7) \bmod 26$

$11 = L$

$(12 - 1) \bmod 26$

$11 = L$

$(16 - 2) \bmod 26$

$14 - O.$

∴ plain text } = HELLO.

Eg: 2 : C.t = F C C E Q S H D

k = X D M O R N Y Z

F	C	C	E	Q	S	H	D
5	2	2	4	16	18	7	3
X	D	M	O	R	N	Y	Z
23	3	12	14	17	13	24	25

$(16-17) \bmod 26$

$1 \rightarrow B$

$(18-13) \bmod 26$

$5 \rightarrow F$

$(7-24) \bmod 26$

$17 \rightarrow R$

$(3-25) \bmod 26$

$22 \rightarrow W.$

$(5-23) \bmod 26$

$= 18 \rightarrow S$

$(2-3) \bmod 26$

$= 1 \rightarrow B$

$(2-12) \bmod 26$

$= 10 \rightarrow K$

$(4-14) \bmod 26$

$= 10 \rightarrow K$

∴ p.T = S B K K B F R W.

P.T : COME TODAY.
K : NCBTZQARX.

C	O	M	E	T	O	D	A	Y
2	14	12	4	19	14	3	0	24
N	C	B	T	Z	Q	A	R	X
13	2	1	19	25	16	0	17	23

C.T

15	16	13	23	18	4	3	17	21
P	Q	N	X	S	E	D	R	V.

disadvantage:

length is equal.

PLAYFAIR CIPHER:

Each & every character is chuffled

→ Encrypt pairs of letter (use digraph)

using 1 letter Eg: R ... (Caesar, one-time unigram or unigraph

using 2 " Eg: Rt ... digram " digraph

600 pair of letter combination possible instead of 26.

i) key table (5×5)

Key = keyword ; Message = Secret message

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

p.t = Secret message

substitution letter

playfair : Se cr et me sx sa ge

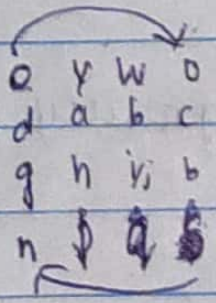
Encryption process → Same column → Same row

→ form a rectangle.

Same col: move each letter down one, reaching end of table, wrap around

Same row: move each letter right one, reaching end of table, wrap around.

form rect: Swap the letters with the one on the end of the rectangle.



Se cr et me sx sa ge = P.T

NO RD KU NF qz PC nd. = C.T

pt = meet me after yoga class

key : gravity falls

q r a v i q r a v i
t y f a l t y f l s
$ ß c d b c d e g
h k m n o
p q U W X

?.T=me et me af· te ry og ac la sx sx
.T=nd lb nd fd bl yc [redacted] rc fv gi gi
xo

q r a v i
t y f l s
b c d e h
K m n o P
q U W X Z.

OC ßl OC FD LB VC KV RD FV LZ LZ

pt : projection key : gravity falls.

Transposition Ciphers:

→ changing the position of each character is changed for each round.

→ position of character is changed but not the identity.

→ Algorithm is more secured by performing permutation

→ changing

1) Rail fence cipher 2) simple columnar transposition 3) simple columnar transposition technique with multiple rounds.

4) ~~Vernan~~ Vernam cipher (one time cipher) 5) Book cipher.

Rail fence:

→ Writing (Rules) plain text as sequence of diagonals.

→ Reading row by row to produce the cipher text.

eg: p.t = Attack at dawn. ; key = rails
= 2

(1 or 2 according to users)

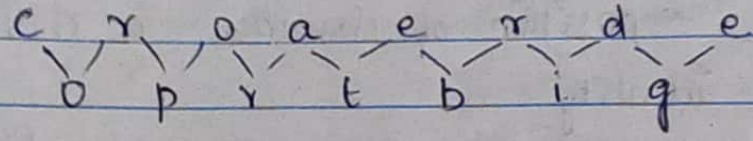
a t c a d w
t a k t a n

c.t : atcadwtaktan

F U P Y
O N R X

Cryptoxool

Eg: 2: pt: corporate bridge. key = 2



pt: defend the east wall of the castle.
key = 3.

d e t e t l f e s e
e n h a w l t c t
f d e s a o h a l

Simple columnar Technique:

① → Basic ② → Multiple rounds

Basic SCT:

Sequence of rows of a rectangle which read in columnar manner.

* all the characters of P.T msg row by row in a rectangle of predefined size.

* Read the msg in a columnar manner (col by col) * Resultant msg = C.T

* Reading msg can be any sequence no need to read in specific or particular pattern or maner.

Eg: corporate bridge : p.t.

choose predefined size :

total char in pt is 15 ∴ 5×3 or 3×5

ration

c o r p o c.T = crroairtdpegobe

① r a t e b

r i d g e decryption : vice versa

.T → write colby col → row by read.

Eg: 2: Gravity falls size: 3×4

G r a v

i t y f

a l l s c.T = Giarrtlaylvfs

Multiple rounds:

→ Repeat the procedure from step 1 to 3 many time as desired (more iteration is required)

Eg: nsropt : crroairtdpegobe. size : 3×5

c r r o a

i r t d p

e g o b e

∴ c.T = clerrgrtoodbape

Vernam cipher (one time pad cipher)

$c = (p+k) \bmod 26$

Simple columnar with multiple rounds:

pt: transposition key: 43152
size = ⑤
split pt into 5 5s

①②③④⑤
∴ pt = trans posit ionxx

4	3	1	5	2
n	a	t	s	r
i	s	p	t	o
x	n	i	x	o

c.t = nixasntpistxroo
—①

①②③④⑤
pt: nixas ntpis txroo

4	3	1	5	2
a	x	h	s	i
i	p	h	s	t
o	r	t	o	x

∴ c.t = aioxprnnt ssoitx.
—②

Cryptographic attack.

Stream cipher : 1 bit at a time

block cipher : AES DES.

classical or tradition Encrypting technique is

Stream : One Element at a time.

Block : We can access given block at a time
64, 128 or 256 bits at a time

→ Reverse encrypted text is simple and easy to breakable in Strean cipher.

→ Reverse encryted text is hard to break in block cipher

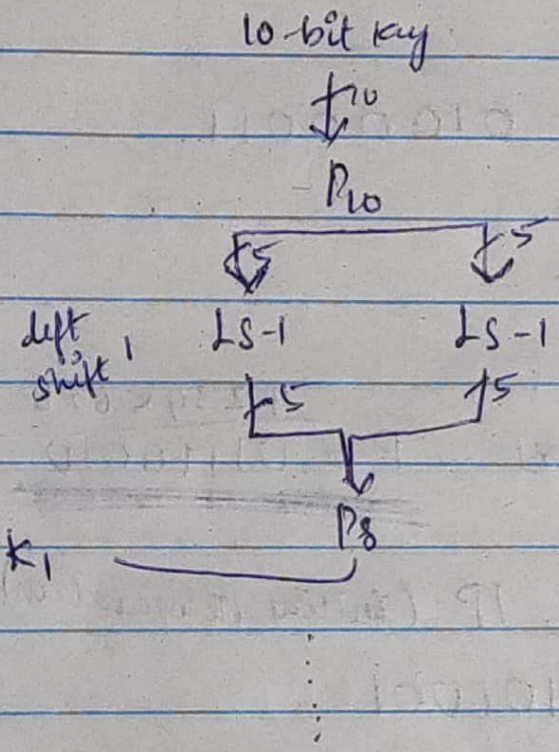
BLOCK CIPHER:

It is an encryption method that applies a deterministic algorithm along with a symmentric key to encrypt a block of text, rather than encrypting one bit at a time as in stream ciphers. For eg: AES encrypts 128 bit blocks with a key of predetermined length: 128, 192, or 256 bits.

S-DES (Simplified - Data Encryption Std)

key is based on permutation

If key = 10 bit it is splitted into 5, 5



Eg: S. DES key generation

10 bit key = K = 1010000010

① . Rearrage K using P10.

P10 = 1000001100

② Rearrage P10 using table

divide into 2 halves : 10000 01100

00001 11000

left shift -1

③ Rearrange the values into P8. (with table)

K1 = 10100100

④ left shift - 2 bits

00001 11000

Ans: 00100 00011

⑤ Rearrange the values with Ps to produce k_2.

$\therefore k_2 = 01000011$

ENCRYPTION:

8 bit plain text: P = 01110010 (12345678)

① Apply the IP (Initial permutation)

10101001

② Two halves L and R.

L = 1010 R = 1001

③ Expand & permutate R using

$\therefore$ R = 11000011

④ X-or input from step ③ with k_1

$\therefore k_1 = 10100100$

R = 11000011

O/p = 01100111

5) two halves L and R.

01100111 — from pre step

∴ L = 0110 R = 0111

Apply substitution boxes: S_0 & S_1

L = 0110 for S_0 = 0110 $B_{1,4}$ → row
$B_{2,3}$ → col

∴ Row = 00
col = 11 ---> 10.

for S_1 = 0111 R = 01, col = 11 ---> 11

o/p = 1011

6) apply P_4.

i/p = 1011

∴ o/p = 0111

7) X-or o/p from step 6 with L from step 3

0111 → previous step
1010 → 1st left half.
1101

8) 1101 → output of previous
1001 → original right halves

Shifting right becomes left & left to right

1001 1101

Round 2:

a) e/p with right halves : 1101

o/p - 11101011

10) XOR with K_2.

K_2 = 01000011
11101011
10101000

11) S-box.

S_0 = 1010 S_1 = 1000

R = 10 R = 10

C = 01 C = 00

o/p = 10 o/p = 11

(2) (3)

Apply P_4.

o/p = 0111

XOR o/p of step 12 with left half from step 8.

```
 0111
 1001
------
 1110
```

o/p from step 13 & right half from step 8

```
1110    1110 1101
```

Apply inverse IP : 01110111

∴ ciphertext = 01110111

S-DES:

Decryption:

$C \cdot T = IP^{-1}(fkey2(SW(fkey1(IP(P \cdot t)))))$

$P \cdot T = ip\ inverse(fk1(SW(fk2(ip((C \cdot t))))))$

$C \cdot T = \overset{1\,2\,3\,4\,5\,6\,7\,8}{01110111}$

Apply Initial permutation:

11101101 —— (1)

Assume the input from above (1) split into two halves:

$L = 1110 \quad R = 1101$ — (2)

Expand & permutate: using E

$R = \overset{1\,2\,3\,4}{1101}$

11101011 —— (3)

xor i/p from (3) with k_2

$$\begin{array}{r} 1110\ 1011 \\ 0100\ 0011 \\ \hline 1010\ 1000 \end{array}$$ —— (4)

5) i/p left half of step(4) into S-box S_0 & S_1 (right half)

00 – 0
01 – 1
10 – 2
11 – 3

$S_0 = L = 1010$.

row = 10 (2) col = 01 (1) = 2 (10).

$S_1 = R = 1000$

row = 10 (2) col = 00 (0) = 3 (11)

6) Rearrange o/p from 5 using P_4:

1 2 3 4
1 0 1 1
0 1 1 1 ——→ (6)

7) X-OR output from (6) with Original L from 2

0 1 1 1
1 1 1 0
1 0 0 1 →(7)

8) SW the halves and move to round 2.

L: 1001 R: 1101.

1101 1001 — (8)

9) Expand & permutate 8

R = 1001

11000011 — (9)

10) xor o/p of (9) with K_1

$$\begin{array}{r} 1100\ 0011 \quad -(9) \\ 1010\ 0100 \quad -K_1 \\ \hline 0110\ 0111 \end{array}$$

11) Input to S-boxes:

$$S_0 \begin{cases} I = 0110 \\ row = 00 \quad col = 11 \quad i = 10 \\ \quad\quad 0 \qquad\quad 3 \qquad\quad 2 \end{cases}$$

$$S_1 \begin{cases} R = 0111 \\ row = 01 \quad col = 11 \quad i = 11 \\ \quad\quad 1 \qquad\quad 3 \qquad\quad 3 \end{cases}$$

12) Rearrange the output from (11) using P_4:

1011 — (11)

o/p 0111 — (P_4 table)

13) X-OR o/p of step 12 with left half (8)

$$\begin{array}{r} 0111 \\ 1101 \\ \hline 1010 \end{array} \leftarrow (13)$$

14) O/P of 13 & right half of (8) into inverse

10101001

01110010 —→ (P.T)

——x——

Eg: 2: P.t = 11010101 key = 0111010001

key generation: 1 2 3 4 5 6 7 8 9 10
0 1 1 1 0 1 0 0 0 1

1) using P_{10} rearrange key.

1010110001 → (1)

2) divide into two halves:

10101 10001

left 01011 00011 → (2)
shift 1 2 3 4 5 5 4 8 9 10

3) rearrange the half with P_8

$k_1 = 00010111$

left shift 2 bits : ②

01011 00011
01101 01100.
12345 678910

$K_2 = 01101100$

Encryption :

pt : 11010101
12345678

Apply IP for PT.

ip.PT = 11011100 → ①

split into two halves.

L = 1101 R = 1100 → ②
1234

Expand & permutate using R

01101001 → ③

X-OR input from ③ with K_1

01101001 → ③
00010111 → ④

01111110 → ⑤

④ split into 2 half.

01111110

L = 0111 R = 1110 → ⑤

Apply S_0 & S_1

S_0 : L = 0111

row = 01 = 01

col = 11 = 11

S_1 : R = 1110

row = 10 = 10

col = 11 = 11

∴ O/p = 0000 → ⑥

Apply P_4 in ⑥.

0000

0000 → ⑦

Apply X-OR in ⑦ with original left half ②.

0000

1101

1101 → ⑧.

1101 → 8

1100 → right half. shifting.

1100 1101 → ⑨.

Round 2.

O/p of (9) right half.

1234
1101

Expand & permutate apply on (9).

11101011 → (10).

X-OR (10) with K2:

```
 11101011
 01101100
----------
 10000111 → (11)
```

Apply (11) in s boxes

L = 1000 R = 0111

So : L = 1000 S_1 : R = 0111

row : 10 row : 01
col : 00 col : 11
00 11

0011 → (12)
1234

Apply P_4 in (12).

0110 → (13)

X-OR (13) with left half (9)

```
 0110
 1100
------
 1010 → (14)
```

o/p of (14) & right half from (9).

10101101 → (15)
1 2 3 4 5 6 7 8

Apply inverse on (15)

~~01010101~~ 01110011

∴ cipher text = ~~01010101~~ 01110011

—x—

Apply IP: C.T = ~~01010101~~ 01110011
1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8

~~10000101~~ 10101101 → (1).

(1) split into 2 halves.

L = 1010 R = 1101.

E/p using R.

R = 1101

11101011

X-OR (3) with K2.

```
  11101011
  01101100
  --------
  10000111
```

5) S-boxes.

$S_0 : L = 1000$

row = 10 → 2

col = 00 → 0

00 → 0

$S_1 : R = 0111$

row = 01 → 1

col = 11 → 3

11

0011

00 0
01 1
10 2
11 3

6) Apply P_4 on (5)

0011 → 0110.

1 2 3 4

7) X-OR (6) with original 1 (2)

0110
1010

1100

8) SW the values & move to round 2.

$\therefore L = 1100 \quad R = 1101$

(1) ↔ (2)

1101 1100

9) E/p on (8)

R = 1101

01101001

x.or ⑨ with k1.

```
01101001
00010111
--------
01111110.
```

Input to s.boxes.

L = 0111 . R = 1110.

row = 01 (1) row = 10 2

col = 11 (3) col = 11 3

00 00.

0000

Apply Py on ⑪.

```
0000
1234   0000.
```

x.or 11 with left half 8.

```
0000
1101
----
1101
```

⑬ & right half of ⑧ then inverse.

```
1101 1100.
12345678
```

11010101 is the plain text.