CBCS SCHEME

USN | | | | | | | | | | |

17TE71

Seventh Semester B.E. Degree Examination, Jan./Feb.2021
Cryptography and Network Security

Time: 3 hrs. Max. Marks: 100

Note: ***Answer any FIVE full questions, choosing ONE full question from each module.***

Module-1

1 a. Explain the procedure to calculate GCD using Euclid's algorithm. Determine the GCD of (24140, 16,762) using Euclid's algorithm. **(06 Marks)**

b. Encrypt the message "Work is workshop" using, play fair cipher with the keyboard "COMPUTER" and decrypt the cipher text to recover the original message. Give the rules for encryption and decryption. **(08 Marks)**

c. Develop a set of additive and multiplications tables for modulo 9. **(06 Marks)**

OR

2 a. Construct the finite field GF (2^4) multiplication table using the polynomial arithmetic modulo ($x^4 + x + 1$), show the calculation steps. **(06 Marks)**

b. Using extended Euclidean, find the multiplicative inverse of 550 mod 1769. **(06 Marks)**

c. Define the following:
(i) Groups, rings and fields.
(ii) Fermat's and Euler's theorem.
(iii) Cryptology, Cryptoanalysis, Cryptography. **(08 Marks)**

Module-2

3 a. Compare AES to DES for each of the following elements of DES :
(i) XOR of subkey material with the input of the f function.
(ii) XOR of the f function output with the left half of the block.
(iii) f function
(iv) Permutation P
(v) Swapping of half of the block. **(06 Marks)**

b. Consider the elliptic curve defined over $E_{2\,3}(1, 1)$. Let P = (3, 10) and Q = (9, 7). Find (P+Q) and 2P. **(08 Marks)**

c. Given p = 19, q = 23, m = 5 and e = 3. Use RSA algorithm to find n, $\phi(n)$, d and C(m). Also find M from decryption. **(06 Marks)**

OR

4 a. What are the 4 tasks performed in each round of AES cipher? Explain. **(06 Marks)**

b. Users A and B use the Diffie Hellman key exchange technique, a common prime q = 11 and a primitive root $\alpha = 7$ (i) If user A has private key $X_A = 3$. What is A's public key Y_A? (ii) If user B has private key $X_B = 6$. What is B's public key Y_B? What is the shared secret key? Write the algorithm as well? **(06 Marks)**

c. Given the plaintext [000102030405060708090A0B0C0D0E0F] and the key [01010101010101010101010101010101]. Show the (a) State matrix (b) Initial round key (c) Sub Bytes (d) Shift rows (e) Mix columns output states. **(08 Marks)**

Module-3

5 a. Explain MD5 algorithm steps. Compare it with SHA-1. **(08 Marks)**

b. Discuss the key components of digital signature algorithm. **(06 Marks)**

c. Explain the HMAC algorithm with a neat diagram. **(06 Marks)**

OR

6 a. Explain the Discrete Logarithm signature scheme. **(06 Marks)**

b. Describe SHA 512 algoritm in detail. **(06 Marks)**

c. Explain the following:

(i) Hash function and its requirements.

(ii) Role of compression function in Hash functions.

(iii) Difference between weak and strong collision resistance.

(iv) Advantages of HMAC over other hash based schemes. **(08 Marks)**

Module-4

7 a. Describe the four protocols defined by secure socket layer. **(06 Marks)**

b. Explain the Secure Shell (SSH) architecture. **(06 Marks)**

c. Explain the various phases of 802.11i. **(08 Marks)**

OR

8 a. Explain the parameters defined in SSL connection. **(06 Marks)**

b. Bring out the differences between SSL and TLS. **(06 Marks)**

c. Explain HTTPS elements encrypted connection initiation and connection closure. **(08 Marks)**

Module-5

9 a. Explain the services provided by PGP and the reasons for using PGP. **(06 Marks)**

b. Explain Encapsulating security pay load header. **(06 Marks)**

c. Explain the preparation of enveloped Data S/MIME entity. Write the functions of S/MIME and Enhanced Security Services of S/MIME. **(08 Marks)**

OR

10 a. Explain the IPsec architecture. **(06 Marks)**

b. Describe the following :

(i) Differences between Tunnel mode and Transport mode of IPsec.

(ii) Scope of ESP encryption and authentication. **(08 Marks)**

c. Explain IKE key determination protocol. **(06 Marks)**

* * * * *

Module 1

1 a **Pseudo Code of the Euclidean Algorithm:**

Step 1: Let **a, b** be the two numbers
Step 2: **a mod b = R**
Step 3: Let **a = b** and **b = R**
Step 4: Repeat Steps 2 and 3 until **a mod b** is greater than 0
Step 5: GCD = b
Step 6: Finish

q	r_1	r_2	r
1	24140	16762	7378
2	16762	7378	2006
3	7378	2006	1360
1	2006	1360	646
2	1360	646	68
9	646	68	34
2	68	34	0
	34	0	

1 b **Plain Text: Work is workshop**

Keyword: COMPUTER

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

Plain Text: WO RK IS WO RK SH OP
Cipher Text: VM KL DZ VM KL XD MU

Encryption Rule of Play-Fair Cipher:

Plaintext is encrypted two letters at a time.

a) If a pair is a repeated letter, insert filler like 'X'
b) If both letters fall in the same row, replace each with the letter to its right (circularly).
c) If both letters fall in the same column, replace each with the letter below it (circularly).
d) Otherwise, each letter is replaced by the letter in the same row but in the column of the other letter of the pair.

Decryption Rules of Play-Fair Cipher:

a) Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the left, with the first element of the row circularly following the last.
b) Two plaintext letters that fall in the same column are each replaced by the letter above, with the top element of the column circularly following the last.
c) Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter.

1 c

Additive Table of Mod 9:

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

Multiplicative Table of Mod 9:

X	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8
2	0	2	4	6	8	1	3	5	7
3	0	3	6	0	3	6	0	3	6
4	0	4	8	3	7	2	6	1	5
5	0	5	1	6	2	7	3	8	4
6	0	6	3	0	6	3	0	6	3
7	0	7	5	3	1	8	6	4	2
8	0	8	7	6	5	4	3	2	1

2 a The Finite Field $GF(2^4)$ multiplication table using the polynomial arithmetic modulo $(x^4 + x + 1)$ is: (To fill each cell in this table it needs to multiply and again apply modulo of $(x^4 + x + 1)$)

×	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	2	4	6	8	10	12	14	3	1	7	5	11	9	15	13
3	3	6	5	12	15	10	9	11	8	13	14	7	4	1	2
4	4	8	12	3	7	11	15	6	2	14	10	5	1	13	9
5	5	10	15	7	2	13	8	14	11	4	1	9	12	3	6
6	6	12	10	11	13	7	1	5	3	9	15	14	8	2	4
7	7	14	9	15	8	1	6	13	10	3	4	2	5	12	11
8	8	3	11	6	14	5	13	12	4	15	7	10	2	9	1
9	9	1	8	2	11	3	10	4	13	5	12	6	15	7	14
10	10	7	13	14	4	9	3	15	5	8	2	1	11	6	12
11	11	5	14	10	1	15	4	7	12	2	9	13	6	8	3
12	12	11	7	5	9	14	2	10	6	1	13	15	3	4	8
13	13	9	4	1	12	8	5	2	15	11	6	3	14	10	7
14	14	15	1	13	3	2	12	9	7	6	8	4	10	11	5
15	15	13	2	9	6	4	11	1	14	12	3	8	7	5	10

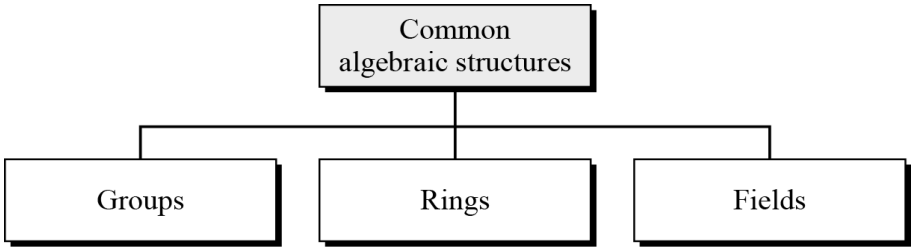
2 b

q	r_1	r_2	r	t_1	t_2	$t = t_1 - qt_2$
3	1759	550	109	0	1	−3
5	550	109	5	1	− 3	16
21	109	5	4	−3	16	−339
1	5	4	1	16	− 339	355
4	4	1	0	−339	355	−1759
	1	0		**355**	− 1759	

2 c (i) **GROUP RING AND FIELDS:**

The combination of the set and the operations that are applied to the elements of the set is called an algebraic structure. There are 3 common algebraic structures

a) Group
b) Rings
c) Fields



GROUP: A group (G) is a set of elements with a binary operation (•) that satisfies four properties (or axioms). It is denoted as **{*G*, •}.**

RING: A ring R, sometimes denoted by { $R, +, \times$ }is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in R the following axioms are obeyed.

FIELD: A **field** *F*, sometimes denoted by {*F*, +, *}, is a set of elements with two binary operations, called *addition* and *multiplication*, such that for all *a*, *b*, *c* in *F* the following axioms are obeyed.

(ii) **Fermat's Theorem:** Fermat's Theorem and Euler's Theorem are mostly used in public key cryptosystem. Fermat's Theorem and Euler's Theorem are helpful for quickly finding solution to exponentiations. Fermat's theorem has 2 versions of the theorem.

First Version: If 'P' is prime and 'a' is any integer but not divisible by 'P' then

$$a^{P-1} mod\ P = 1$$

Second Version: The second version removes the condition if 'P' is a prime and 'a' is any integer, then

$$a^{P} mod\ P = a$$

Euler's Theorem: Euler's theorem can be thought as a generalization of a Fermat's theorem. The modulus in the Fermat's theorem is a prime, but the modulus in Euler's theorem is an integer.

Like Fermat's theorem there are 2 versions of Euler's theorem

First Version: If 'a' and 'n' are relatively prime then

$$a^{\emptyset(n)} mod\ n = 1$$

Second Version: like Fermat's theorem, it removes the condition that 'a' and 'n' should be Co-prime.

$$a^{K.\emptyset(n)+1}\ mod\ n = a$$

(iii) **Cryptology:** The area of cryptography and cryptanalysis are called cryptology.

Cryptanalysis: The art of breaking the cipher text is known as cryptanalysis.

Cryptography: The art of keeping message secure is called cryptography.

Module-2

3 a Comparing AES to DES for the following Elements of DES:

Parameter	AES	DES
XOR of subkey material with the input of the f function	The subkey of length 128-bit is applied to the Complex function 'F'. The Complex function operates on complete 128-bit of the message and the XOR of subkey with the input of the function 'F' generates 128-bit output.	The subkey of length 48-bit is applied to the Complex function 'F'. The Complex function operates only on 32-bit of the message and the XOR of subkey with the input of the function 'F' generates 32-bit output.
XOR of the f function output with the left half of the block	This is not the case in AES. The output of the complex function 'F' is provided directly to the next round.	Yes, the output of the Complex function 'F' is XORed with the left half of the message block to generate the new right block of message.
F function	The round function itself can be treated as the complex function, which perform 4 operations.	The complex function consists of Expansion of D-box, XOR, S-Box and the straight S-Box.
Permutation P	There is no such concept of permutation P in AES.	There are 2 permutation choice being performed one is PC-1 which is operated on 64-bit Key and generates the output of 56-bit. The PC-2 operates on the 56-bit and produce an output of 48-bit.
Swapping of half of the block	Not Performed in AES	Yes the swapping of half of the block is performed at the end of each round.

3 b $\boldsymbol{P + Q}$:

$$\Delta = \left(\frac{y_Q - y_p}{x_Q - x_P}\right) mod\ p \Rightarrow \Delta = \left(\frac{7-10}{9-3}\right) mod\ 23 = \left(\frac{-3}{6}\right) mod\ 23$$

$\Delta = \left(\frac{-1}{2}\right) mod\ 23 = 11$

$x_R = (\Delta^2 - x_P - x_Q)\ mod\ p = (11^2 - 3 - 9) mod\ 23 = 109\ mod\ 23 = 17$

$y_R = (\Delta(x_P - x_R) - y_P)\ mod\ p = (11(3 - 17) - 10) mod\ 23$

$y_R = -164\ mod\ 23 = 20$

$P + Q = (17,20)$

$2P$:

$\Delta = \left(\frac{3{x_P}^2 + a}{2y_P}\right)\ mod\ p = \left(\frac{3(3^2) + 1}{2 \times 10}\right) mod\ 23 = \left(\frac{5}{20}\right) mod\ 23$

$\Delta = \left(\frac{1}{4}\right) mod\ 23 = 4^{-1} mod 23 = 6$

q	r_1	r_2	r	t_1	t_2	$t = t_1 - qt_2$
5	23	4	3	0	1	-5
1	4	3	1	1	-5	6
3	3	1	0	-5	6	-23
	1	0		6	-23	

$x_R = (\Delta^2 - 2x_P)\ mod\ p = (6^2 - 2 \times 3)\ mod\ 23 = 30\ mod\ 23 = 7$

$y_R = (\Delta(x_P - x_R) - y_P)\ mod\ p = (6(3 - 7) - 10)\ mod\ 23$

$y_R = (-34)\ mod\ 23 = 12$

$2P = (7,12)$

3 **c** $n = pq = 19 \times 23 = 437$

$\emptyset(n) = (p - 1) \times (q - 1) = 18 \times 22 = 396$

$e = 3$

$ed\ mod\ \emptyset(n) \equiv 1 => d = e^{-1}\ mod\ \emptyset(n) => d = 3^{-1}\ mod\ 396 =$ *Solution Doesn't exist*

$PU = \{3,437\}$ *and* $PR = \{343,437\}$

$C = M^e\ mod\ n => C = 5^3 mod\ 437 = 125$

$M = C^d\ mod\ n = 125^{Wrong}\ mod\ 527 =$ *No Soution* **(Wrong Question,** GCD($\emptyset(n)$,e)=1 **which is not satisfied here)**

4 **a** **Four Tasks are performed in each round of AES cipher:**

(i) SubBytes
(ii) ShiftRows
(iii) MixColumns
(iv) AddRoundKey

Substitute Bytes Transformation: AES defines a 16 × 16 matrix of byte values, called an S-box, that contains a permutation of all possible 256 8-bit values. Each individual byte of **State** is mapped into a new byte. The leftmost 4 bits of the byte are used as a row value and the rightmost 4 bits are used as a column value. These row and column values serve as indexes into the S-box to select a unique 8-bit output value. Example: the hexadecimal value {95} references row 9, column 5 of the S-box, which contains the value {2A}. S_box({95}) = {2A}

Here is an example of the SubBytes transformation:

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

→

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

ShiftRows Transformation: The first row of **State** is not altered. For the second row, a 1-byte circular left shift is performed. For the third row, a 2-byte circular left shift is performed. For the fourth row, a 3-byte circular left shift is performed.

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

→

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

MixColumns Transformation: Each byte of a column is mapped into a new value that is a function of all four bytes in that column. The transformation can be defined by the following matrix multiplication on **State.**

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Each element in the product matrix is the sum of products of elements of one row and one column. In this case, the individual additions and multiplications are performed in GF(2^8).

$$s'_{0,j} = (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$

$$s'_{1,j} = s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j} \oplus s_{3,j}$$

$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j})$$

$$s'_{3,j} = (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j})$$

AddRoundKey Transformation: The 128 bits of **State** are bitwise XORed with the 128 bits of the round key.

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$\oplus$

AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

$=$

EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D6

The Add round key transformation is as simple as possible and affects every bit of **State**. Due to the complexity of the round key expansion, plus the complexity of the other stages of AES, ensure security.

4 b $Y_A = \alpha^{X_A} \ mod \ q \Rightarrow \ Y_A = 7^3 \ mod \ 11 = 2$
$Y_B = \alpha^{X_B} \ mod \ q \Rightarrow \ Y_B = 7^6 \ mod \ 11 = 4$
$K_A = {Y_B}^{X_A} \ mod \ q \Rightarrow \ K_A = 4^3 \ mod \ 11 = 9$
$K_B = {Y_A}^{X_B} \ mod \ q \Rightarrow \ K_B = 2^6 \ mod \ 11 = 9$
$K_A = K_B = 9$

4 c **This Question is very lengthy and not possible to solve within 3 hours if not provided with the s-Box and also Mix column multiplication also require at least require 3hrs to calculate the result as it uses the $GF(2^8)$ multiplication and addition.**

AES Algorithm:

Plain Text = [0001 0203 0405 0607 0809 0A0B 0C0D 0E0F]
Key = [0101 0101 0101 0101 0101 0101 0101 0101]

(i) State Matrix:

00	04	08	0C
01	05	09	0D
02	06	0A	0E
03	07	0B	0F

(ii) Initial Round Key:

01	01	01	01
01	01	01	01
01	01	01	01
01	01	01	01

(iii) Sub Bytes:

00	04	08	0C

from **S-Box** →

63	F2	30	FE

01	05	09	0D
02	06	0A	0E
03	07	0B	0F

7C	6B	01	D7
77	6F	67	AB
7B	C5	2B	76

(iv) Shift rows:

63	F2	30	FE
6B	01	D7	7C
67	AB	77	6F
76	7B	C5	2B

(v) Mix Columns:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} 63 & F2 & 30 & FE \\ 6B & 01 & D7 & 7C \\ 67 & AB & 77 & 6F \\ 76 & 7B & C5 & 2B \end{bmatrix} = \begin{bmatrix} \_ & \_ & \_ & \_ \\ \_ & \_ & \_ & \_ \\ \_ & \_ & \_ & \_ \\ \_ & \_ & \_ & \_ \end{bmatrix}$$

_	_	_	_
_	_	_	_
_	_	_	_
_	_	_	_

Module-3

5 a **MD5:** MD5 is an improved version of MD4. Although more complex than MD4, it is similar in design and also produces a 128-bit hash. MD5 processes the 512-bit input, divided into 16 32-bit sub-blocks. Output of the algorithm is a 128-bit hash value which is a set of four 32-bit blocks.

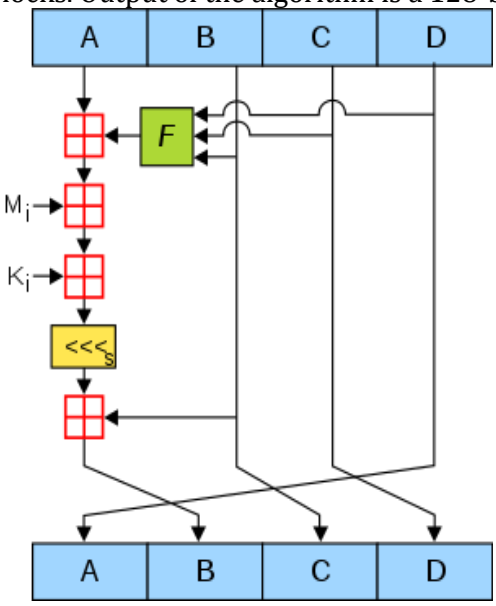


Figure: One MD5 operation

Description of MD5 (Algorithm): Message is **padded** so that its length is just 64 bits short of being a multiple of 512 (e.g. Length = 448 bit or 960bit etc). **Append message length:** padding is done by adding a single 1-bit to the end of the message, followed by as many zeros as are required. Then message's length is appended to the result. **Initialize MD buffer**. In MD5 total 4 variables are used known as MD buffer. Those are initialized as *A = 1234567, B = 89abcdef, C = fedcba98, D = 76543210.*

First the four variables are copied into different variables: *a gets A, b gets B, c* gets *C and d gets D.* The main loop has four rounds (MD4 had only three rounds) of 16 operations each.

Each operation performs a nonlinear function on three of *a, b, c and d* and then it adds that result to the fourth variable, a sub-block of the text and a constant. Then it rotates that result to the right a variable number of bits and adds the result to one of *a, b, c,* or *d*. Finally, the result replaces one of *a, b, c,* or *d*. There are four nonlinear functions, one used in each operation (a different one for each round).

$F(B,C,D) = (B \wedge C) \vee ((\neg B) \wedge D)$ $\quad H(B,C,D) = B \oplus C \oplus D$

$G(B,C,D) = (B \wedge D) \vee (C \wedge (\neg D))$ $\quad I(B,C,D) = C \oplus (B \vee (\neg D))$

Where $\wedge$ *is* ***AND***, $\vee$ *is* ***OR***, $\oplus$ *is* ***XOR***, $\neg$ *is* ***NOT***

*M*j represents the *j* th sub-block of the message (from 0 to 15)

<<<s represents a left circular shift of *s* bits.

The four operations can be represented as :

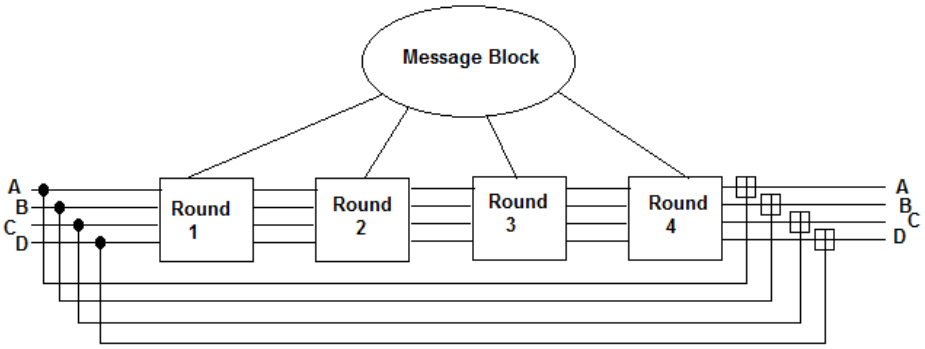
$FF(a,b,c,d,M_j,s,t_i) denotes\ a = b + \left((a + F(b,c,d) + M_j + t_i) <<< s\right)$

$GG(a,b,c,d,M_j,s,t_i) denotes\ a = b + \left((a + G(b,c,d) + M_j + t_i) <<< s\right)$

$HH(a,b,c,d,M_j,s,t_i) denotes\ a = b + \left((a + H(b,c,d) + M_j + t_i) <<< s\right)$

$II(a,b,c,d,M_j,s,t_i) denotes\ a = b + \left((a + I(b,c,d) + M_j + t_i) <<< s\right)$

Those constants t_i were chosen as follows:
In step i , t_i is the integer part of $2^{32} \times abs(\sin(i))$, where i is in radians
The main loop of MD5 can be shown as:



Main Loop of MD5

Comparison with SHA-1:

(i) MD5 uses four buffers whereas SHA-1 uses 5 buffers.
(ii) The algorithm is little modified as compared to MD5.
(iii) There is total 20 mathematical operations are being performed per each round in SHA-1 whereas in MD-5 each round is having 16 mathematical operation.
(iv) The key being used in MD5 for each mathematical operation in each round were unique, whereas the key being used for constant for a single round.

5 b Digital Signature Algorithm:

Global Public- Key Components:

p = a prime number L bits long, where $512 \leq L \leq 1024$ and is a multiple of 64.

q = a 160-bit prime factor of $p - 1$.

$g = h^{(p-1)/q}\ mod\ p$, Where $1 < h < (p-1)$ such that $h^{(p-1)/q}\ mod\ p > 1$

User's Private key:

$x =$ random number such that $0 < x < q$

User's Public Key:

$y = g^x\ mod\ p$

User's Pre-message secret Key:

$k =$ random number such that $0 < k < q$

Signing:

$r(signature) = \left(g^k mod\ p\right) mod\ q$

$s(signature) = \left(k^{-1}(H(m) + xr)\right) mod\ q$

Verifying:

$w = s^{-1}\ mod\ q$

$u_1 = [H(m)w] mod\ q$

$u_2 = (rw) mod\ q$

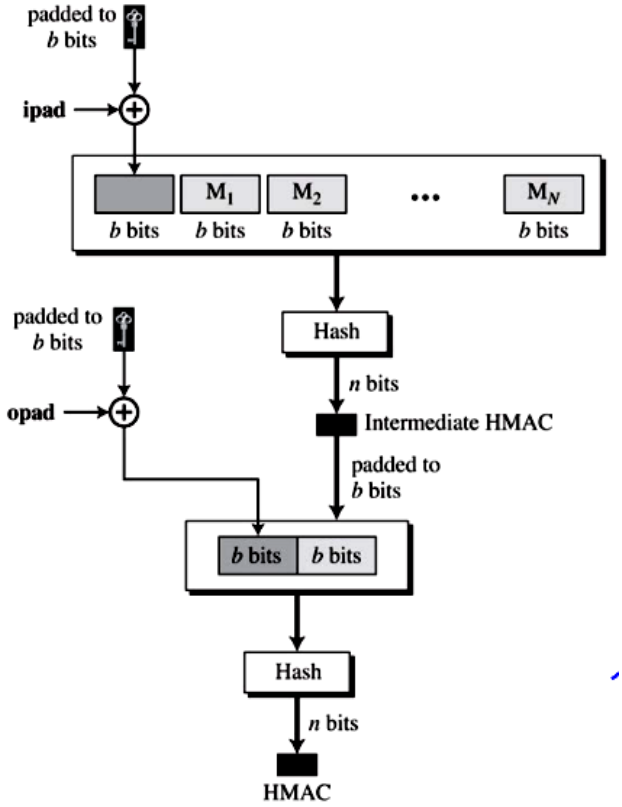
$v = [(g^{u_1} y^{u_2}) mod\ p] mod\ q$

$if \quad v = r$, the signature is verified

5 c HMAC:

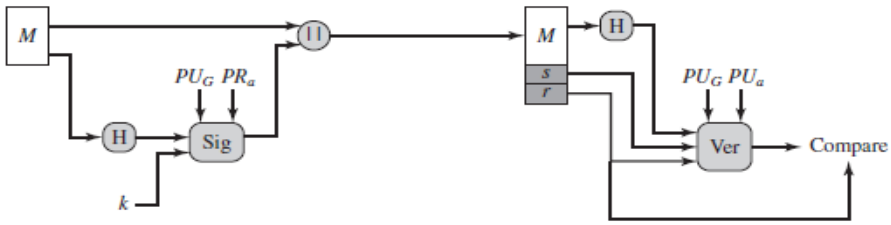
The message is divided into N blocks, each of b bits. The secret key is left-padded with 0's to create a b −bit key. The value of ipad is the $b/8$ repetition of the sequence 00110110. (36 in hexadecimal). The result is $N + 1$ blocks. The result of is hashed to create an n − bit digest. It is called as the

intermediate digest. The intermediate $n-$ bit HMAC is left padded with 0s to make a $b-$ bit block. Steps are repeated by a different constant opad (output pad). The value of 0pad is the $b/8$ repetition of the sequence 01011100. (5C in hexadecimal). The output is then prepended to the block hashed to create the final n-bit HMAC.



6 a Discrete Logarithm Signature Scheme:

DIGITAL SIGNATURE ALGORITHM (DSA):



Global Public- Key Components:

p = a prime number L bits long, where $512 \leq L \leq 1024$ and is a multiple of 64.

q = a 160-bit prime factor of p – 1.

$g = h^{(p-1)/q} \ mod \ p$, Where $1 < h < (p-1)$ such that $h^{(p-1)/q} \ mod \ p > 1$

User's Private key:

$x =$ random number such that $0 < x < q$

User's Public Key:

$y = g^x \ mod \ p$

User's Pre-message secret Key:

$k =$ random number such that $0 < k < q$

Signing:

$r(signature) = \left(g^k mod \ p\right) mod \ q$

$s(signature) = \left(k^{-1}(H(m) + xr)\right) mod \ q$

Verifying:

$w = s^{-1} \ mod \ q$

$u_1 = [H(m)w] mod \ q$

$u_2 = (rw) mod \ q$

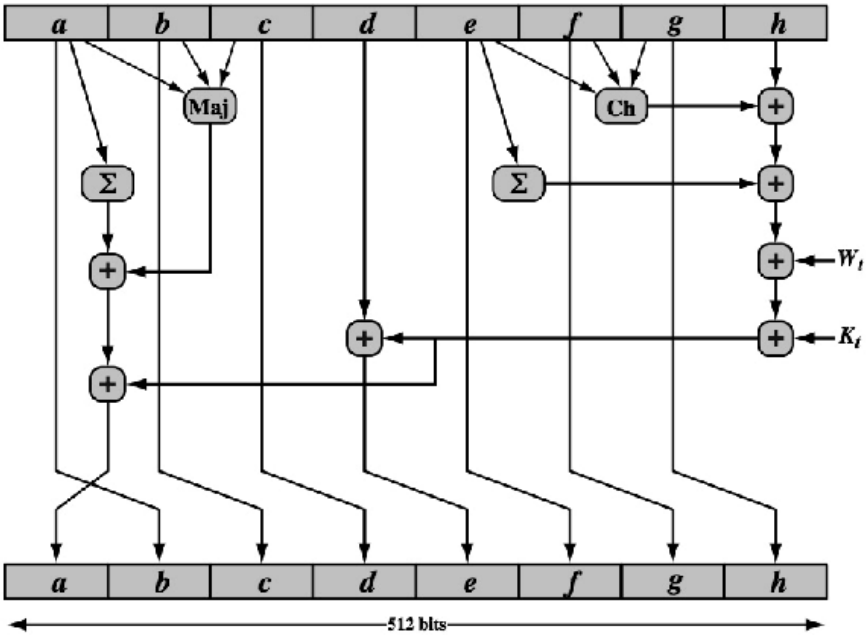
$v = [(g^{u_1} y^{u_2}) mod\ p] mod\ q$

$if \quad v = r$, the signature is verified

6 b SHA-512 Algorithm: (Out of Syllabus):

Initialization Vector

a = 0x6A09E667F3BCC908	b = 0xBB67AE8584CAA73B
c = 0x3C6EF372FE94F82B	d = 0xA54FF53A5F1D36F1
e = 0x510E527FADE682D1	f = 0x9B05688C2B3E6C1F
g = 0x1F83D9ABFB41BD6B	h = 0x5BE0CD19137E2179



6 c (i) <u>Requirements of Hash function:</u>

Requirement	Description
Variable input size	H can be applied to a block of data of any size.
Fixed output size	H produces a fixed-length output.
Efficiency	H(x) is relatively easy to compute for any given x
Preimage resistant (one-way property)	For any given hash value h, it is computationally infeasible to find y such that H(y) = h.
Second preimage resistant (weak collision resistant)	For any given block x, it is computationally infeasible to find $y \neq x$ with H(y) = H(x).
Collision resistant (strong collision resistant)	It is computationally infeasible to find any pair (x, y) such that H(x) = H(y).

A one-way hash function, H(M), operates on an arbitrary-length message, M and It returns a fixed-length hash value, *h*. *h* = H(M), where h is of length m(fixed). There are many functions which can take arbitrary-length input and return an output of fixed length but one-way hash functions have additional characteristics:

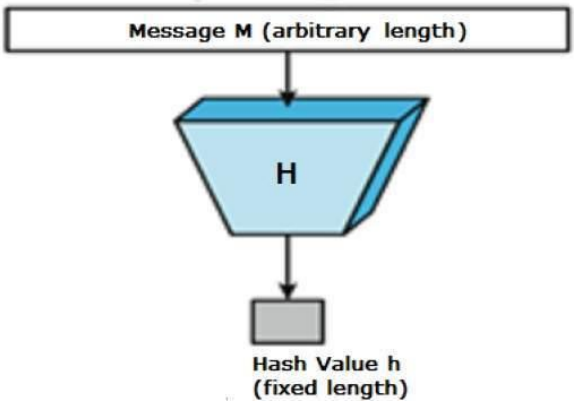
a) Given *M*, it is easy to compute *h*.

b) Given *h*, it is hard to compute *M* such that *H*(*M*)= *h*.

c) Given *M*, it is hard to find another message, M', such that *H*(*M*) =*H*(*M'*).

Important requirement of one-way hash function is to provide a "fingerprint" of *M* that is unique. Hash function value is often referred to as a **message digest.**

(ii) The hash function involves repeated use of compression function f.



(iii) Integrity protection requires collision resistant hash functions,

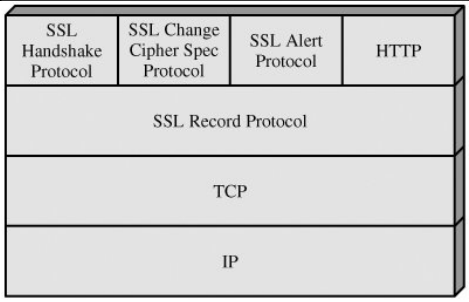
- ✓ 2nd pre-image resistance (weak collision resistance): Given an input x and h(x), it is computationally infeasible to find another input x' $x \neq x'$ with $h(x) \neq h(x')$
- ✓ Collision resistance (strong collision resistance): it is computationally infeasible to find any two inputs x and x', $x \neq x'$ with $h(x) \neq h(x')$.

(iv) **HMAC algorithm** stands for Hashed or Hash based Message Authentication Code. It is a result of work done on developing a MAC derived from cryptographic hash functions. HMAC is a great resistant towards cryptanalysis attacks as it uses the Hashing concept twice. HMAC consists of twin benefits of Hashing and MAC, and thus is more secure than any other authentication codes.

Module-4

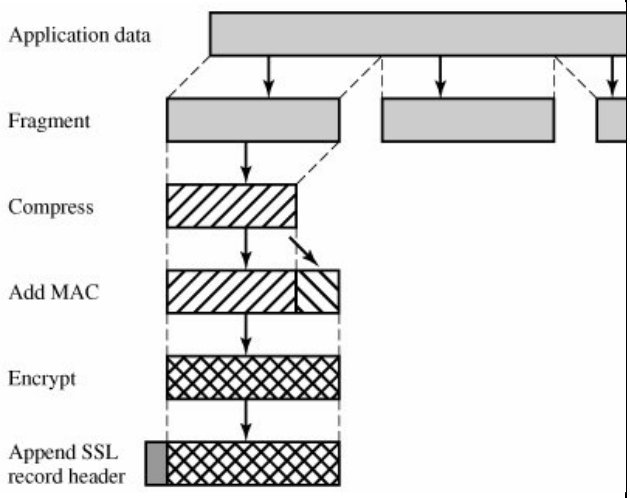
7 a Four Protocols defined by Secure Socket Layers are:

i. Record Protocol
ii. Handshake protocol
iii. Change of cipher spec protocol
iv. Alert Protocol



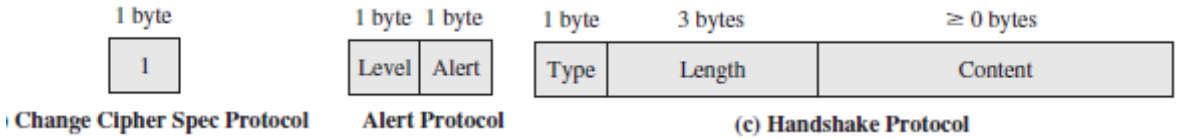
[Figure: SSL Protocol Stack]

SSL Record Protocol:



[Figure: SSL Record Protocol Operation]

Change of Cipher Spec Protocol: The change cipher spec protocol is one of the 3 SSL specific protocols that use the SSL record protocol and it is the simplest. This protocol consists of a single message which consists of a single byte with the value 1. The purpose of this message is to cause the pending state to be copied into the current state.



Change Cipher Spec Protocol **Alert Protocol** **(c) Handshake Protocol**

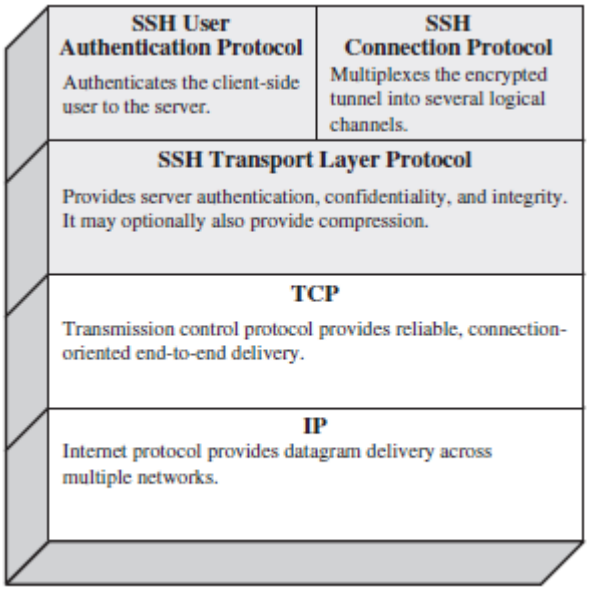
SSL Alert Protocol: SSL uses the alert protocol for reporting errors and abnormal conditions. Each message in this protocol consists of 2 bytes. The 1st byte takes the value warning (1) or fatal (2) to convey the severity of the message. If the level is fatal, SSL immediately terminates the connection.

Other connections on the same session may continue, but no new connection on this session will be established.

SSL HANDSHAKE PROTOCOL: The handshake protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and the keys to be used to protect data sent in an SSL record. The handshake protocol is used before any application data is transmitted. The handshake protocol consists of a series of messages exchanged by client and server. Each message has 3 fields

(i) Type (1 Byte): one out of 10 messages.
(ii) Length (3 Bytes): Length of the message in bytes.
(iii) Content (≥ 0 Bytes): Parameters associated with that message.

7 b **SECURE SHELL (SSH):** Secure Shell (SSH) is a protocol for secure network communications designed to be relatively simple and inexpensive to implement. The initial version, SSH1 was focused on providing a secure remote logon facility to replace TELNET and other remote logon schemes that provided no security. SSH also provides a more general client/server capability and can be used for such network functions as file transfer and e-mail. SSH2 fixes a number of security flaws in the original scheme. SSH client and server applications are available for all most all the operating system. It has become the choice for remote login and X tunneling.



SSH Protocol Stack

SSH is organized as three protocols that run on top of TCP.

a) Transport Layer Protocol
b) User Authentication Protocol
c) Connection Protocol

Transport Layer Protocol Provides server authentication, data confidentiality, and data integrity. It may optionally provide compression.
User Authentication Protocol authenticates the user to the server.
Connection Protocol multiplexes multiple logical communications over a single SSH connection.

7 c **IEEE 802.11i PHASES OF OPERATION:**

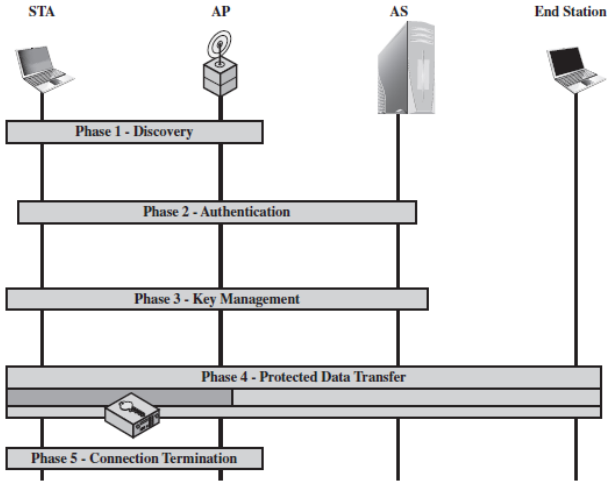


Figure IEEE 802.11i Phases of Operation

The operation of an IEEE 802.11i RSN can be broken into 5 phases. There are phases of operation for an RSN. The Network component involved are Wireless stations (STA), Access Point (AP), Authentication Server (AS), End station. The rectangle indicates the exchange of MAC protocol data unit (MPDU). Those 5 Phases are

i. Discovery
ii. Authentication
iii. Key Generation and distribution
iv. Protected data transfer
v. Connection Termination

8 a **SSL Connection Parameters:**

Sl. No.	Parameters	Description

	1	**Server and Client random**	It is the byte sequences that are chosen by the server and the client for each connection.
	2	**Server write MAC secret**	The secret key used in MAC operation on data sent by the server
	3	**Client Write MAC secret**	The secret key used in MAC operation on data sent by the client
	4	**Server Write Key**	The encryption key for data encryption done by the server and decryption by the client.
	5	**Client Write Key**	The encryption key for data encryption done by the client and decryption by the server.
	6	**Initialization vector**	For different modes of operations such as CBC, OFB, CFB the initialize vector is defined for each cipher key during negotiation, which is used for the 1st block exchange. The final cipher text from the block is used as the IV for the next block
	7	**Sequence Number**	Each party maintains separate sequence numbers for transmitted and received messages for each connection. The sequence number starts from 0 and increments. It must not exceed $2^{64} - 1$

8 b

Parameter	**SSL**	**TLS**
Abbreviation for	Secure socket Layer	Transport Layer Security
Chronology	Predecessor	Successor
Version	SSL v1, SSL v2, SSL v3	TLS v1.0, TLS v1.1, TLS v1.2, TLS v1.3
Developed by	Netscape	The internet Engineering Task Force (IETF)
Speed	It is faster	Slower due to the two-step communication process of handshaking and actual data transfer.
Compatibility	Doesn't support TLS	TLS v1.0 had an SSL fallback mechanism for backward-compatibility
Management	Less complex than TLS	More complex to manage than SSL
Browser support	Not supported by present day browsers	Most browsers support this protocol.
Related attacks	POODLE attack and DROWN attack	No such known attack associated with TLS v1.3
Challenges	Vulnerabilities allowed attacks like "man in middle", which led to discontinuation of this protocol.	Previous version of TLS had vulnerabilities which have been address in TLS v1.3

8 c **HTTPS:** HTTPS is the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server. Some search engines do not support HTTPS. Google provides HTTPS as an option: https://google.com. The principal difference seen by a user is that URL (uniform resource locator) addresses begin with https:// rather than http://. A normal HTTP connection uses port 80. If HTTPS is specified, port 443 is used, which Invokes SSL. When HTTPS is used, the following elements of the communication are encrypted:

(i) URL of the requested document
(ii) Contents of the document
(iii) Contents of browser forms (filled in by browser user)
(iv) Cookies sent from browser to server and from server to browser
(v) Contents of HTTP header

There is no fundamental change in using HTTP over either SSL or TLS, and both implementations are referred to as HTTPS.

Connection Initiation: The client initiates a connection to the server on the appropriate port and then sends the TLS client hello message to begin the TLS handshake. When the TLS handshake has

finished, the client may then client may initiate the first HTTP request. All HTTP data is to be sent as TLS application data. There are 3 levels of awareness of a connection in HTTPs.

(i) HTTP client requests a connection to an HTTP server by sending a connection request to the next lowest layer. The next lowest layer is the TCP but it also may be TLS/SSL

(ii) A session is established between a TLS client and the TLS server. This session can support one or more connections.

(iii) TLS request to establish a TCP connection.

Connection Closure: HTTP client and the server can indicate the closing of connection by including the following lines in an HTTP record: Connection: close. It indicates that the connection will be closed after this record is delivered. The closure of an HTTPS connection requires that TLS close the connection with the peer, which involves closing the TCP connection. Proper way to close a connection is for each side to use the TLS alert protocol to send a close_notify alert. TLS must initiate an exchange of closure alert before closing a connection.

Module-5

9 a Table Summary of PGP Services

Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and included with the message.
Compression	ZIP	A message may be compressed for storage or transmission using ZIP.
E-mail compatibility	Radix-64 conversion	To provide transparency for e-mail applications, an encrypted message may be converted to an ASCII string using radix-64 conversion.

PRETTY GOOD PRIVACY (PGP): PGP was invented by Phil Zimmermann in 1991 to provide e-mail with privacy, integrity and authentication. PGP can be used to create a secure e-mail message or to store a file securely for future retrieval.

Operational Description: The PGP provides 4 services: Authentication, Confidentiality, Compression, E-mail Compatibility.

9 b **ENCAPSULATING SECURITY PAYLOAD (ESP):** ESP can be used to provide the following services: Confidentiality, Data origin Authentication, Connectionless Integrity, Antireplay Services, Traffic Flow Confidentiality. ESP can work with a variety of encryption and authentication algorithms.

ESP Format: ESP Format contains the following field.

(i) **Security Parameters Index (SPI) (32 bits):** Identifies a security association.

(ii) **Sequence Number (32 bits):** A monotonically increasing counter value

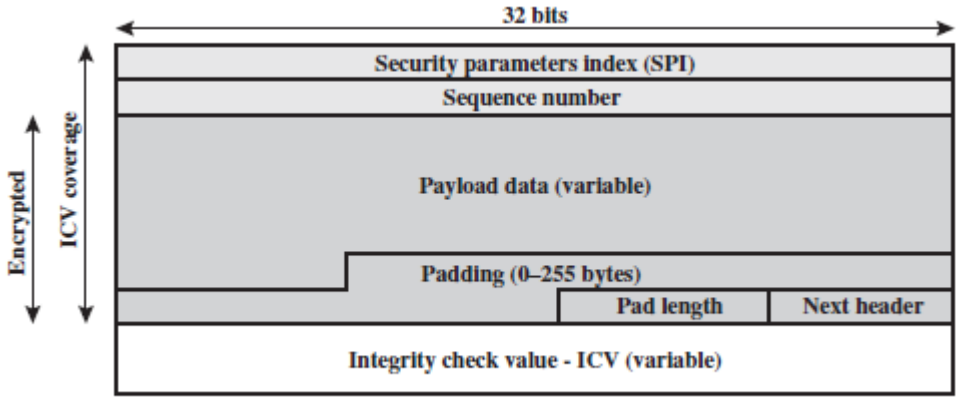
(iii) **Payload Data (variable):** This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.

(iv) **Padding (0–255 bytes):** If the encryption algorithm requires the plaintext to be multiple of some number of bytes, padding is used to expand the plaintext. Additional padding may be used to provide traffic flow confidentiality.

(v) **Pad Length (8 bits):** Indicates the number of pad bytes.

(vi) **Next Header (8 bits):** Identifies the type of data contained in the payload data field by identifying the first header in that payload (e.g., an extension header in IPv6, or an upper-layer protocol such as TCP).

(vii) **Integrity Check Value (ICV) (variable):** A variable-length field that contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field.



Header Format of ESP

9 c **<u>MULTIPURPOSE INTERNET MAIL EXTENSION:</u> (MIME):** Multipurpose internet mail extension (MIME) is an extension to the RFC 322. MIME Solves some of the problem appears when simple mail transfer protocol (SMTP) is used. Some of the problems are listed below:

a) SMTP can't transmit executable files or other binary objects
b) SMTP can't transmit the text data that includes national language characters.
c) SMTP server may reject mail message over a certain size.
d) SMTP gateway translates between ASCII and the character code EBCDIC (Extended Binary Coded Decimal Interchange code); but it don't use consistent mapping results the translation problem.
e) SMTP doesn't support non-textual data.
f) Wrapping lines longer than 76 characters.
g) Removal of trailing white spaces

MIME intended to resolve these problems.

<u>MIME Overview:</u> Five new message header fields are defined, those are

a) MIME version: The parameter value is 1.0 to ensure the standards.
b) Content Type: Describes the data contained in the body with sufficient details.
c) Content Transfer Encoding: Indicate the type of transformation used.
d) Content ID: Used to identify the MIME Entities uniquely.
e) Content Description: A text description of the object with the body, this is useful when the object is not readable (e.g. audio data)

S/MIME is similar to PGP. Both offer the ability to sign and encrypt the messages. S/MIME uses 3 public key algorithms

a) Digital Signature standard (DSS) is used for digital signature.
b) Diffie-Hellman algorithm for encrypting session key infact S/MIME uses a variant of Diffie-Hellman that provides encryption/decryption known as ElGmal.
c) RSA can be used for both signature and session key encryption.

S/MIME uses the MD5 and SHA-1 hash functions. MD5 for backward compatibility with older version of S/MIME. For message encryption 3 key triple DES is used, but it also supports 40 bit RC2.

<u>S/MIME: (Secure/Multipurpose Internet Mail Extension)</u> MIME stands for multipurpose Internet Mail extensions. As the name indicates, it is an extension to the internet mail. It helps the user to exchange different kinds of data files over the internet such as image, audio and video. MIME is required if text used are other than the ASCII. The traditional e-mail was following a format standard RFC 822, which is still in common use. The recent version of this format is RFC 5322 (Internet Message Format)

Secure/ Multipurpose Internet mail extension is a secure enhancement to the MIME Internet e-mail. S/MIME has emerged as the industry standard for commercial and organizational use.

10 a **<u>IP Sec architecture:</u>** The IP security architecture uses the concept of a security association (SA) as the basis for building security functions into IP. IP security policy is determined primarily by the interaction of 2 databases.

(i) Security association database (SAD)

(ii) Security policy Database (SPD)

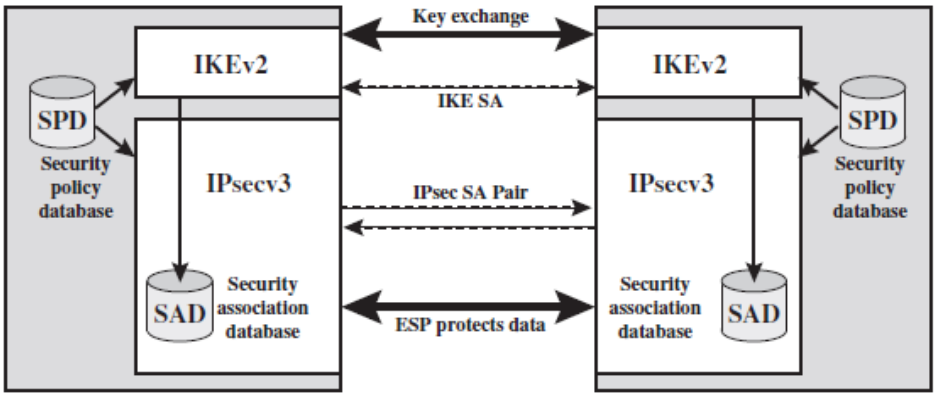
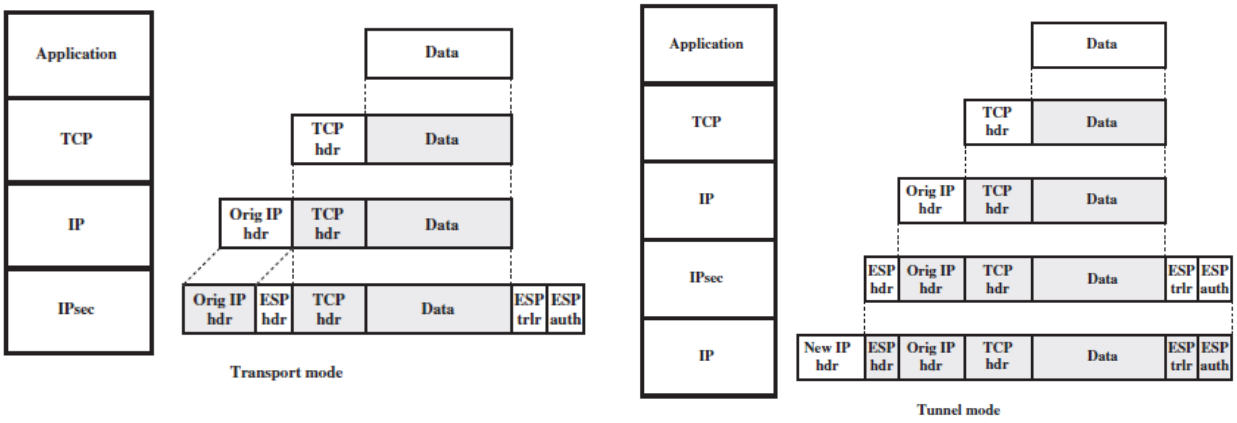
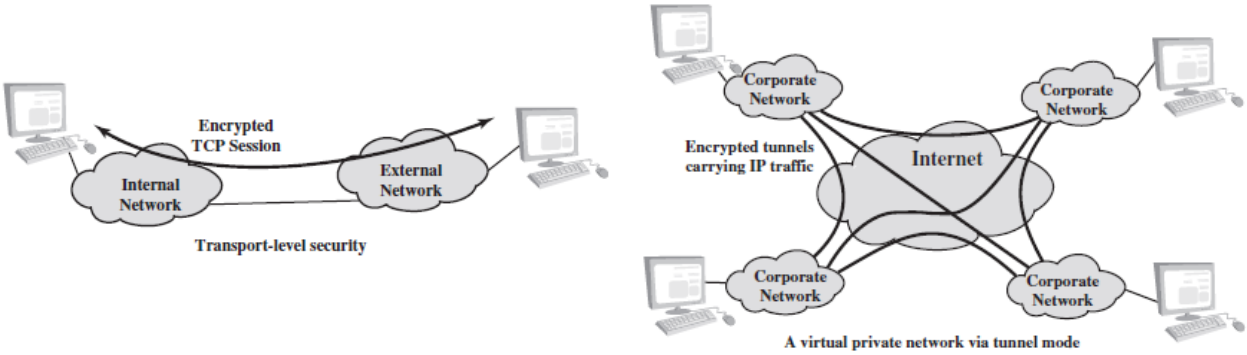


Figure IPsec Architecture

10 b (i) Difference between Tunnel and Transport Mode:

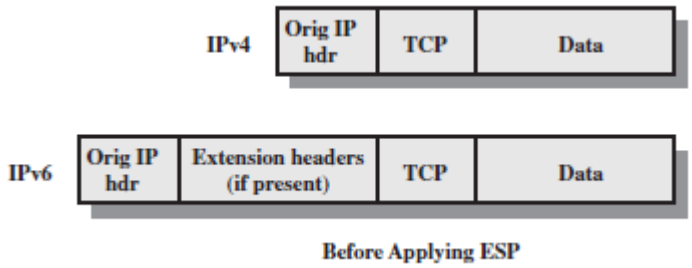


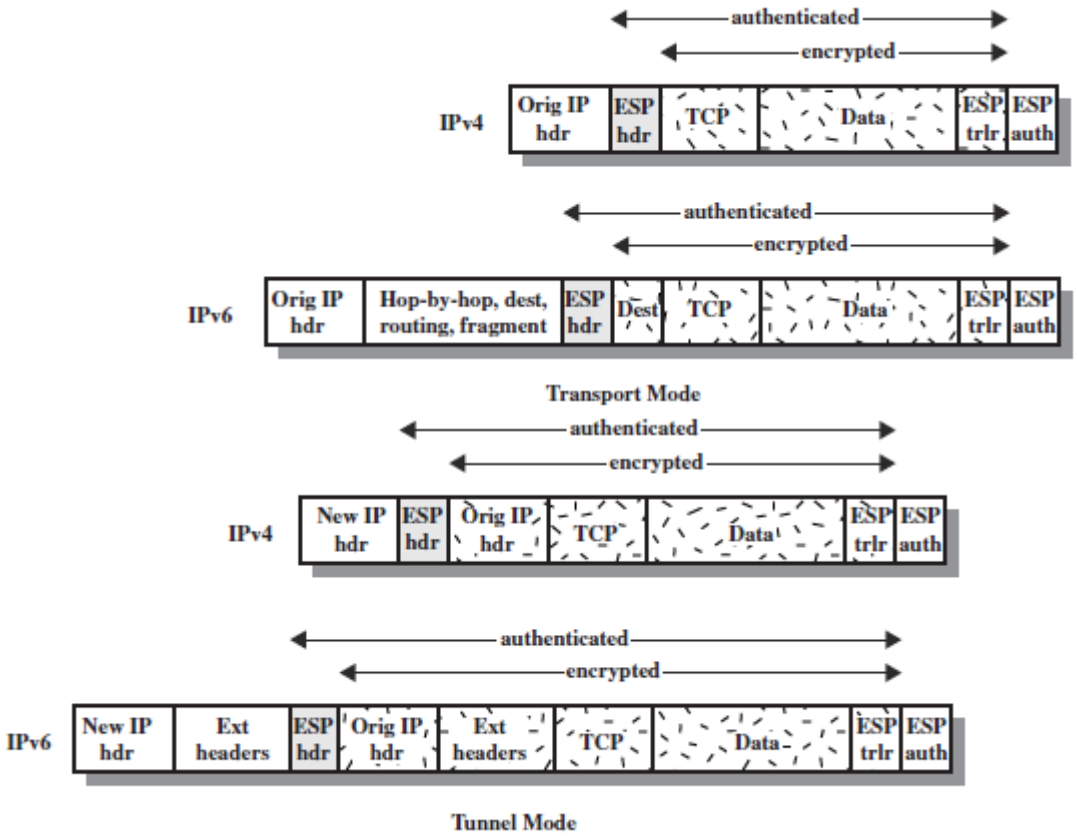
<u>Transport and Tunnel Mode in ESP:</u> The transport mode provides encryption directly between 2 hosts.



Tunnel mode operation can be used to set up a virtual private network (VPN). In the below example, 4 private networks interconnected across the Internet. Hosts on the internal Networks use the internet for transport of data.

(ii) Scope of Authentication and Encryption:





10 c **<u>IKE Key determination Protocol:</u>**

IKE key determination is a refinement of the Diffie-Hellman key exchange algorithm. Diffie-Hellman involves the interaction between users A and B. There is prior agreement on two global parameters: q and large prime number and α, a primitive root of q. A selects a random integer X_A as its private key and transmits to B its public key $Y_A = \alpha^{X_A} \bmod q$. Similarly, B selects a random integer X_B as its private key and transmits to A its public key $Y_B = \alpha^{X_B} \bmod q$. Each side can now compute the secret session key: $K = (Y_B)^{X_A} \bmod q = (Y_A)^{X_B} \bmod q = \alpha^{X_A \alpha^{X_B}} \bmod q$

The Diffie-Hellman algorithm has two attractive features:

(i) Secret keys are created only when needed. There is no need to store secret keys for a long period of time, exposing them to increased vulnerability

(ii) The exchange requires no pre-existing infrastructure other than an agreement on the global parameters.

The default automated key management protocol for IPsec is referred to as ISAKMP/Oakley and consists of the following elements:

(i) **Oakley Key Determination Protocol:** Oakley is a key exchange protocol based on the Diffie-Hellman algorithm but providing added security. Oakley is generic i.e. it does not dictate specific formats.

(ii) **Internet Security Association and Key Management Protocol (ISAKMP):** ISAKMP provides a framework for Internet key management and provides the specific protocol support, including formats, for negotiation of security attributes.