CNS SESSIONAL II IMPORTANT QUESTIONS

UNIT 3:

1)RSA cryptosystem with example

RSA Encryption Algorithm

RSA is a public key encryption algorithm developed by Rivert(R) , Shamir(S) and Adleman (A) in year 1977.

The RSA scheme is a block cipher in which the plaintext & ciphertext are integers between 0 and n-1 for some 'n'.

A typical size for 'n' is 1024 bits or 309 decimal digits.

RSA algorithm uses an expression with exponentials.

- Plaintext is encrypted in blocks, with each block having a binary value less than some number n.
- That is, the block size must be less than or equal to **log2(n) + 1**; in practice, the block size is **i** bits, where Encryption and decryption are of the following form, $2^i < n <= 2^{i+1}$
- for some plaintext block **M** and ciphertext block **C.**

$$C = M^e \bmod n$$

$$M = C^d \bmod n$$

$$M = C^d \bmod n = (M^e \bmod n)^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

- Both sender and receiver must know the value of **n**.
- The sender knows the value of **'e'** & only the receiver knows the value of **'d'** thus this is a public key encryption algorithm with a

Public key PU={e, n}

Private key PR={d, n}

- It is possible to find values **of e, d,** and **n** such that

  M^{ed} mod n = M for all M < n.

  It is relatively easy to calculate **M^{e} mod n** and **C^{d} mod n** for all values of **M < n**.

  It is infeasible to determine **d** given **e** and **n.**

Key Generation by Alice

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calcuate $\phi(n) = (p - 1)(q - 1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d = e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

RSA Algorithm-Encryption & Decryption

Encryption by Bob with Alice's Public Key

Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption by Alice with Alice's Public Key

Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

RSA Algorithm Example

- **Key Generation:**

Let p=7 and q=17 (both are primes and they are not equal)

n= p*q => n=7*17 => n=119

(n)=(7-1)*(17-1)=> 6*16=>96

Now calculate factors of 96 which is 2*2*2*2*2*3

Now select **e** such that **gcd((n),e)=1** (we need to select a value which is not equivalent to factors 2 and 3)

Take e=5

Calculate d such that **de≡1(mod (n))**

By congruency we can write it as

de(mod (n))=1

d*5 mod 96=1

Select 77

77*5 mod 96=1

385mod96=1

So d=77

Now

public key is PU={e,n}= {5,119}

private key is PR={d,n}=c{77,119}

Now we perform encryption by taking plaintext as **10**

Calculate ciphertext as

$C=M^e \bmod n$ = 10^5mod119

= 100000 mod 119

= 40

Thus the ciphertext generated is 40

Now we perform the decryption on the ciphertext generated C=40

We generate the plaintext as

$M= c^d \bmod n$ = 40^77 mod 119

= 10

We get our plaintext M=10

2) Explain about Rabin Cryptosystem with an example

Rabin Cryptosystem is an public-key cryptosystem invented by Michael Rabin. It uses asymmetric key encryption for communicating between two parties and encrypting the message.

Steps in Rabin cryptosystem

- Select two large distinct prime numbers p and q, such that p ☰ 3 mod 4, and q ☰ 3 mod 4(ex: p=139 and q=191).
- Calculate n = p*q.

n = 139*191
=26549

Encryption

- Obtain the public key n.

n=26549

- Convert the message's value to ASCII. Then convert it to binary, multiply it by itself, and convert the binary value back to decimal m.

m = "R" = 82 (R's ASCII value is 82)
= 82_{10} = 1010010_2;
= 1010010 **|** 1010010_2; → double extend
= 10578_{10}

- Encrypt with the following formula:

$c = m^2 \bmod n$

$c = 10578^2 \bmod 26549$

$c = 16598$

- Send C to the intended recipient.

Decryption

- Accept the sender's C.

c=16598

- Using Extended Euclidean GCD, specify a and b so that x*p + y*q = GCD(p, q).

x*p + y*q = 1

139x+191y=1

- Using the following formula, compute r and s:

$r = c^{(p+1)/4} \bmod p$

$s = c^{(q+1)/4} \bmod q$

$r = 16598^{(139+1)/4} \bmod 139 = 125$

$s=16598^{(191+1)/4} \bmod 191=118$

- Now, use the following formula to compute X and Y:

X = (x*p*r + b*q*s) mod p

Y = (y*p*r – b*q*s) mod q

X = 11*139*118 = 180422

Y=-8*191*125=-191000

- The four roots are as follows: m1=X, m2=-X, m3=Y, and m4=-Y.

- Now, Convert all of them to binary and divide them in half.

- Determine which half of the left and right halves are the same. Keep one half of that binary and convert it to decimal m. Obtain the ASCII character corresponding to the decimal value m. The resulting character conveys the message sent by the sender.

UNIT 4:

1) Explain SHA – 512 algorithm

1. Explain the working of SHA-512 Hash Algorithm with a neat diagram?

The Secure Hash Algorithm (SHA) was invented by the National Security Agency (NSA) and published in 1993 through the National Institute of Standard and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS PUB 180).

SHA depends on and shares the similar building blocks as the MD4 algorithm. The design of SHA introduced a new process which develop the 16-word message block input to the compression function to an 80-word block between other things.

The processing of SHA works as follows –

Step 1 – **Append padding bits** – The original message is padded and its duration is congruent to 448 modulo 512. Padding is continually inserted although the message already has the desired length. Padding includes a single 1 followed by the essential number of 0 bits.

Step 2 – **Append length** – A 64-bit block considered as an unsigned 64-bit integer (most essential byte first), and defining the length of the original message (before padding in step 1), is added to the message. The complete message's length is a multiple of 512.

Step 3 –**Initialize the buffer** – The buffer includes five (5) registers of 32 bits each indicated by A, B, C, D, and E. This 160-bit buffer can be used to influence temporary and final outcomes of the compression function. These five registers are initialized to the following 32-bit integers (in hexadecimal notation).

2. A = 67 45 23 01
3. B = ef cd ab 89
4. C = 98 ba dc fe
5. D = 10 32 54 76
6. E = c3 d2 e1 f0

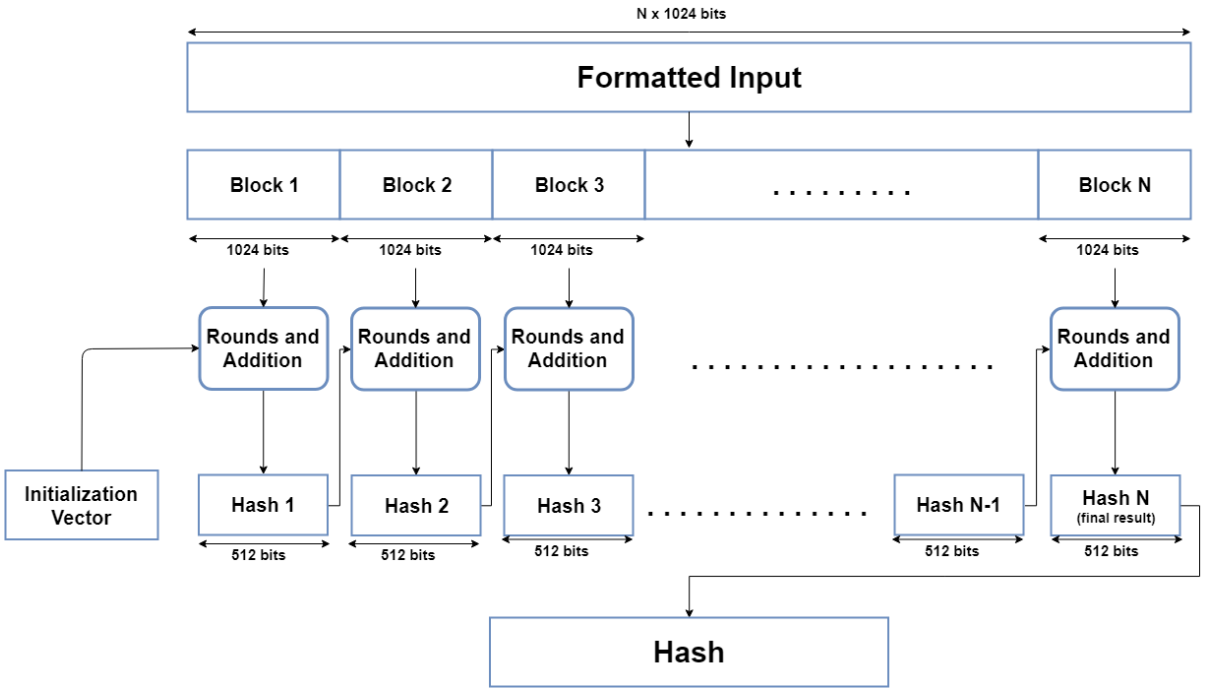
The registers A, B, C, and D are actually the same as the four registers used in MD5 algorithm. But in SHA-1, these values are saved in big-endian format, which define that the most essential byte of the word is located in the low-address byte position. Therefore the initialization values (in hexadecimal notation) occurs as follows –

7. word A = 67 45 23 01
8. word B = ef cd ab 89
9. word C = 98 ba dc fe
10. word D = 10 32 54 76
11. word E = c3 d2 e1 f0

Step 4 – **Process message in 512-bit blocks** – The compression function is divided into 20 sequential steps includes four rounds of processing where each round is made up of 20 steps.

The four rounds are structurally same as one another with the only difference that each round need a different Boolean function, which it can define as f1, f2, f3, f4 and one of four multiple additive constants Kt ($0 \le t \le 79$) which is based on the step under consideration.

Step 5 – **Output** – After processing the final 512-bit message block t (considering that the message is divided into t 512-bit blocks), and it can obtain a 160-bit message digest.



2)Explain how message integrity is different from message authentication

Message Integrity means checking the message's authenticity. It makes sure that the message has not been altered or tampered with. Message integrity means that a message has not been tampered with or modified.
There are many ways to verify the integrity of a message. Message Authentication Codes

- Signature Schemes
- Nonrepudiation
- Certificates
- Hash Functions

MAC

It is a combination of Key Value and a Cryptographic function used to encode or decode a text.

Working of Message Authentication Codes

The following steps explain the working of Message Authentication Codes for checking the message integrity:

1. The sender first uses the MAC Algorithm on the text message and generates an output called ciphertext.
2. Sender then combines ciphertext and key. This combination is called a MAC Code or Message Authentication Code.
3. Sender shares that Message Authentication Code with the receiver.

4. Once the receiver receives the Message Authentication Code, the receiver again runs the MAC Algorithm on the message to generate another Authentication Code. Then the receiver compares it with the Authentication code sent by the sender. If they match, then the message is verified; else, not.

3) Explain the Security services offered by a digital signature

A digital signature is a mathematical technique which validates the authenticity and integrity of a message, software or digital documents. It allows us to verify the author name, date and time of signatures, and authenticate the message contents. The digital signature offers far more inherent security and intended to solve the problem of tampering and impersonation (Intentionally copy another person's characteristics) in digital communications.

The important reason to implement digital signature to communication is:

- Authentication
- Non-repudiation
- Integrity

Authentication

Authentication is a process which verifies the identity of a user who wants to access the system. In the digital signature, authentication helps to authenticate the sources of messages.

Non-repudiation

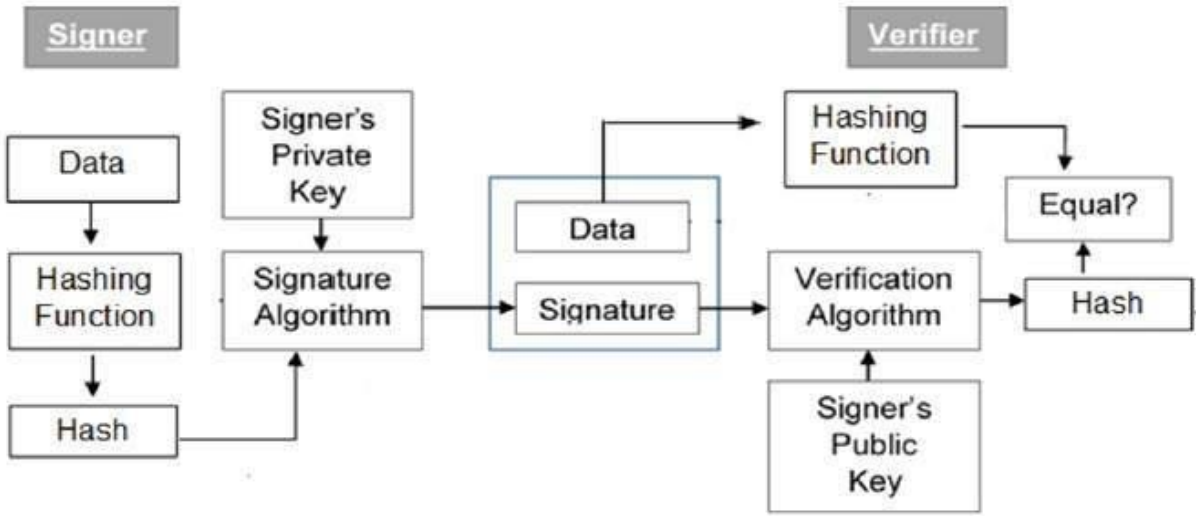
Non-repudiation means assurance of something that cannot be denied. It ensures that someone to a contract or communication cannot later deny the authenticity of their signature on a document or in a file or the sending of a message that they originated.

Integrity

Integrity ensures that the message is real, accurate and safeguards from unauthorized user modification during the transmission.

4) Describe signing and verification in Digital Signature

digital signature scheme is based on public key cryptography. The model of digital signature scheme is depicted in the following illustration –



The following points explain the entire process in detail –

- Each person adopting this scheme has a public-private key pair.
- Generally, the key pairs used for encryption/decryption and signing/verifying are different. The private key used for signing is referred to as the signature key and the public key as the verification key.
- Signer feeds data to the hash function and generates hash of data.
- Hash value and signature key are then fed to the signature algorithm which produces the digital signature on given hash. Signature is appended to the data and then both are sent to the verifier.
- Verifier feeds the digital signature and the verification key into the verification algorithm. The verification algorithm gives some value as output.
- Verifier also runs same hash function on received data to generate hash value.
- For verification, this hash value and output of verification algorithm are compared. Based on the comparison result, verifier decides whether the digital signature is valid.
- Since digital signature is created by 'private' key of signer and no one else can have this key; the signer cannot repudiate signing the data in future.

It should be noticed that instead of signing data directly by signing algorithm, usually a hash of data is created. Since the hash of data is a unique representation of data, it is sufficient to sign the hash in place of data. The most important reason of using hash instead of data directly for signing is efficiency of the scheme.

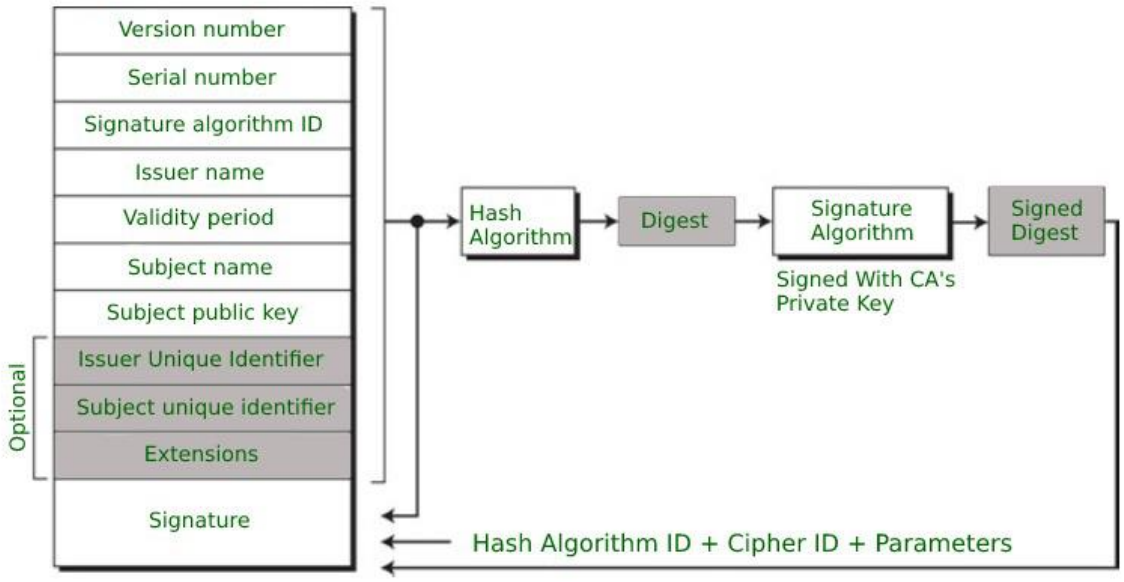
5) Authentication Procedures in X.509 Certificate

X.509 is a digital certificate that is built on top of a widely trusted standard known as ITU or International Telecommunication Union X.509 standard, in which the format of PKI certificates is defined.

Working of X.509 Authentication Service Certificate:

The core of the X.509 authentication service is the public key certificate connected to each user. These user certificates are assumed to be produced by some trusted certification authority and positioned in the directory by the user or the certified authority. These directory servers are only used for providing an effortless reachable location for all users so that they can acquire certificates. X.509 standard is built on an IDL known as ASN.1. With the help of Abstract Syntax Notation, the X.509 certificate format uses an associated public and private key pair for encrypting and decrypting a message.

Format of X.509 Authentication Service Certificate:



Generally, the certificate includes the elements given below:

- **Version number:** It defines the X.509 version that concerns the certificate.
- **Serial number:** It is the unique number that the certified authority issues.

- **Signature Algorithm Identifier:** This is the algorithm that is used for signing the certificate.
- **Issuer name:** Tells about the X.500 name of the certified authority which signed and created the certificate.
- **Period of Validity:** It defines the period for which the certificate is valid.
- **Subject Name:** Tells about the name of the user to whom this certificate has been issued.
- **Subject's public key information:** It defines the subject's public key along with an identifier of the algorithm for which this key is supposed to be used.
- **Extension block:** This field contains additional standard information.
- **Signature:** This field contains the hash code of all other fields which is encrypted by the certified authority private key.

UNIT 5

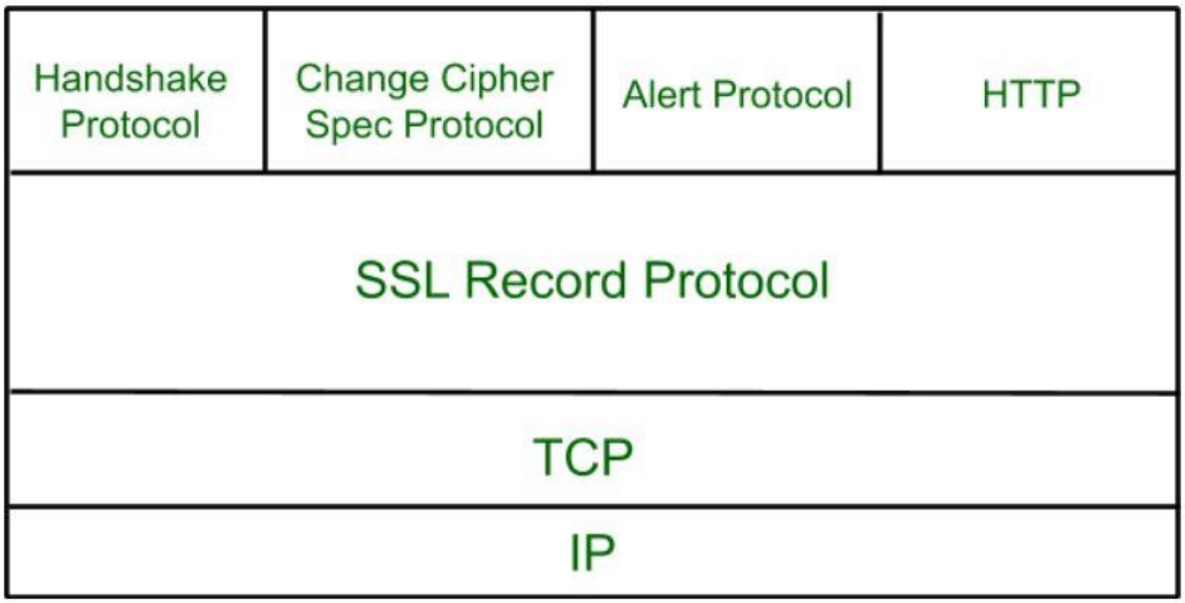
1)SSL Handshake Protocol with a neat diagram

Secure Socket Layer (SSL) provides security to the data that is transferred between web browser and server. SSL encrypts the link between a web server and a browser which ensures that all data passed between them remain private and free from attack.

Secure Socket Layer Protocols:

- SSL record protocol
- Handshake protocol
- Change-cipher spec protocol
- Alert protocol

SSL Protocol Stack:



SSL Record Protocol:

SSL Record provides two services to SSL connection.

- Confidentiality
- Message Integrity

Handshake Protocol:

Handshake Protocol is used to establish sessions. This protocol allows the client and server to authenticate each other by sending a series of messages to each other. Handshake protocol uses four phases to complete its cycle.

Change-cipher Protocol: Change-cipher protocol consists of a single message which is 1 byte in length and can have only one value. This protocol's purpose is to cause the pending state to be copied into the current state.

Alert Protocol: This protocol is used to convey SSL-related alerts to the peer entity. Each message in this protocol contains 2 bytes.

2) Compare Security Facilities in TCP/IP Protocol Stack at Network, Transport and Application Levels

TCP/IP Model helps you to determine how a specific computer should be connected to the internet and how data should be transmitted between them

Application Layer

Application layer interacts with an application program, which is the highest level of OSI model. The application layer is the OSI layer, which is closest to the end-user. It means the OSI application layer allows users to interact with other software application.

The function of the Application Layers are:

- Application-layer helps you to identify communication partners, determining resource availability, and synchronizing communication.
- It allows users to log on to a remote host
- This layer provides various e-mail services

Transport Layer

Transport layer builds on the network layer in order to provide data transport from a process on a source system machine to a process on a destination system. It is hosted using single or multiple networks, and also maintains the quality of service functions.

Important functions of Transport Layers:

- It divides the message received from the session layer into segments and numbers them to make a sequence.
- Transport layer makes sure that the message is delivered to the correct process on the destination machine.

Internet Layer

An internet layer is a second layer of TCP/IP layes of the TCP/IP model. It is also known as a network layer. The main work of this layer is to send the packets from any network, and any computer still they reach the destination irrespective of the route they take.

Layer-management protocols that belong to the network layer are:

1. Routing protocols
2. Multicast group management
3. Network-layer address assignment.

3) services provided by PGP

PGP is short for Pretty Good Privacy, a security program that enables users to communicate securely by decrypting and encrypting messages, authenticating messages through digital signatures, and encrypting files.

Confidentiality and Authentication – The both services can be used for the same message. First, a signature is produced for the plaintext message and prepended to the message. Therefore the plaintext message plus signature is encrypted using CAST-128 (or IDEA or 3DES), and the session key is encrypted using RSA.

This sequence is desirable to the opposite encrypting the message and thus producing a signature of the encrypted message. It is usually more convenient to save a signature with a plaintext version of a message. Moreover, for the goals of third party verification, if the signature is implemented first, a third party need not be concerned with the symmetric key when testing the signature.

Compression – As a default, PGP restrict the message after using the signature but before encryption. This has the advantage of storing space both for e-mail transmission and for file storage.

E-mail compatibility – Some electronic mail systems only allows the use of blocks including ASCII text. When PGP is used, minimum part of the block to be transmitted is encrypted.

Segmentation – E-mail facilities are restricted to a maximum message length. For instance, some facilities accessible throughout the internet set a maximum length of 50,000 octets. Some message higher than that should be broken up into smaller segments, each of which is mailed independently.

4)short notes on IPSec elements

IPSec

IP Sec (Internet Protocol Security) is an Internet Engineering Task Force (IETF) standard suite of protocols between two communication points across the IP network that provide data authentication, integrity, and confidentiality.

Components

1. Encapsulating Security Payload (ESP)
2. Authentication Header (AH)
3. Internet Key Exchange (IKE)

Encapsulating Security Payload (ESP): It provides data integrity, encryption, authentication, and anti-replay. It also provides authentication for payload.

Authentication Header (AH): It also provides data integrity, authentication, and anti-replay and it does not provide encryption. The anti-replay protection protects against the unauthorized transmission of packets. It does not protect data confidentiality.

Internet Key Exchange (IKE): It is a network security protocol designed to dynamically exchange encryption keys and find a way over Security Association (SA) between 2 devices. The Security Association (SA) establishes shared security attributes between 2 network entities to support secure communication. The Key Management Protocol (ISAKMP) and Internet Security Association provides a framework for authentication and key exchange. ISAKMP tells how the setup of the Security Associations (SAs) and how direct connections between two hosts are using IPsec. Internet Key Exchange (IKE) provides message content protection and also an open frame for implementing standard algorithms such as SHA and MD5. The algorithm's IP sec users produce a unique identifier for each packet. This identifier then allows a device to determine whether a packet has been correct or not. Packets that are not authorized are discarded and not given to the receiver.