Assignment No. 2

Q.1 Explain Advanced Encryption Standards (AES) in detail.

⇒ i) Advanced Encryption Standard (AES) is a highly trusted encryption algorithm used to secure data by converting it into an unreadable format without the proper key.

ii) It is widely used today as it is much stronger than DES & triple DES despite being harder to implement.

iii) AES encryption uses various key lengths (128, 192 or 256 bits) to provide strong protection against unauthorized access.

iv) This data security measures is efficient & widely implemented in securing internet communication, protecting sensitive data & encrypting files.

Key features :-

- AES is Block Cipher
- The key size can be 128/192/256 bits
- Encrypts data in blocks of 128 bits each

v) It means it takes 128 bits as input & output 128 bits of encrypted cipher text.

vi) AES relies on the substitution-permutation network principle, which is performed using a series of linked operations that involve replacing & shuffling the input data.

E.
gad

Page No.:
Date :

KGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCE

*Working of AES Cipher :-

- AES performs operations on bytes of data rather than in bits. Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.

- The number of rounds depends on the key length as follow:-

N (number of Rounds)	key size
10	128
12	192
14	256

Encryption Process :- AES considers each block as a 16-byte (4 byte x 4 byte) grid in a column-major arrangement.

b0	b4	b8	b12
b1	b5	b9	b13
b2	b6	b10	b14
b3	b7	b11	b15

Each round comprises of 4 steps:-

a) subBytes
b) ShiftRows
c) Mix Column
d) Add Round key

Step 1:- Sub Bytes

This step implements the substitution. In this step each byte is substituted by another byte. It is performed using a lookup table also called the S-box. This substitution is done in a way that a byte never substituted by itself & also not substituted by another byte which is a compliment of the current byte. The result of this step is 16-byte (4×4) matrix like before.

Step 2:- Shift Rows

This step is just it sounds. Each row is shifted a particular number of times.

- The first row is not shifted.
- The second row is shifted once to the left.
- The third row is shifted twice to the left
- The fourth row is shifted thrice to the left.

b_1	b_2	b_3	b_4
b_5	b_6	b_7	b_8
b_9	b_{10}	b_{11}	b_{12}
b_{13}	b_{14}	b_{15}	b_{16}

→

b_1	b_2	b_3	b_4
b_6	b_7	b_8	b_5
b_{11}	b_{12}	b_9	b_{10}
b_{16}	b_{13}	b_{14}	b_{15}

Step 3:- Mix Columns

This step is a matrix multiplication. Each column is multiplied with a specific matrix & thus the position of each byte in the column is changed as a result.

This step is skipped in the last round.

$$\begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 3 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Step 4:- Add Round key.

- Now the resultant output of the previous stage is XOR-ed with the corresponding round key. Here, the 16 bytes are not considered as a grid but just as 128 bits of data.
- After all these rounds 128 bits of encrypted data are given back as output. This process is repeated until all the data to be encrypted undergoes this process.

Q.2] Explain SHA-1 algorithm.

⇒ i) SHA (Secure Hash Algorithm -1) is a cryptographic hash function developed by NSA.

ii) It produces a 160-bit (20-bytes) fixed size hash from an input of any length used for data integrity, digital signature, authentication, but now considered insecure due to collision vulnerabilities

iii) Padding the Message, The original message is padded so that its length becomes congruent to 448 mod 512.

iv) Breaking into Blocks, The padded message split into 512-bit blocks. Each 512-bit block is further divided into sixteen 32-bit words.

v) Message Schedule expansion, These 16 words are expanded into eighty 32-bit words using bitwise operations & shifts.

vi) Initialize Hash values:- Five 32-bit variables are initialized A, B, C, D, E with fixed constants.

vii) Main Compression function (80 Rounds)

- The algorithm processes each 512-bit block in 80 rounds, grouped in 4 stages of 20 rounds.
- Each stage uses a different logical function & Constant. Logical function include bitwise AND, OR, XOR & majority.
- Although E are updated in each round using the current message word, previous hash value & constants.

viii) Final Hash ~~Comparison~~ Compilation Computation:- After processing all blocks the final value of A, B, C, D, E are added to their initial value. The result is a 160-bit message digest (The SHA-1).

Q.3] Explain Symmetric key Distribution using Neetham-Schroeder authentication protocol.

⇒ Symmetric key Distribution :- SKD in Cryptography refers to the process of securely sharing a secret key between two parties so that they can use it for encryption & decryption of data. Since both parties need the same key, the primary challenge lies in ensuring the key is not intercepted during the distribution key.

Needham Schroeder :- It is a symmetric key protocol based on a two key transport protocols intended for use over an insecure network, both proposed by Roger Needham & Michael Schroeder.

- The Needham-Schroeder protocol based on a symmetric key protocol encryption algorithm. It forms the basis for the kerberos protocol. This protocol aims to establish a session key between two parties on a network typically to protect further communication.
- The Needham Schroeder Public-key protocol based on public key cryptographic/phy. This protocol is intended to provide mutual

authentication between two parties communicating on a network, but in its proposed form is insecure.

Example:- Here Alice (A) initiates the communication to Bob (B). S is a server trusted by both parties. In the communication:-

a) A & B are identities of Alice & Bob respectively.
b) NA & NB are nonces generated by A & B.
c) KAB is a symmetric generated key, which will be the session key of the session between A & B.
d) The protocol can be specified as follow in security protocol notation:- $A \rightarrow S: A, B, NA$
e) Alice sends a message to the server identifying herself & Bob, telling the server she wants to communicate with bob.

$S \rightarrow A: \{NA, KAB, B, \{KAB, A\}KBS\}KAS$

f) The server generates KAB & sends back to Alice a copy encrypted under KBS for Alice to forward to Bob & also a copy for Alice. Since Alice may be requesting keys for several different people, the nonce assures Alice that the message is fresh & that the server is replying to that particular message & the inclusion of Bob's name tells Alice who she is to share this key with.

$A \rightarrow B: \{KAB, A\}KBS$

Q.4 Explain Kerberos Authentication protocol.

=> Kerberos :- It provides a centralized authentication server whose functio is to authenticate users to servers to users. In Kerberos Authentication server & database is used for client Authentication. Kerberos runs a third party trusted server known as the Key Distribution center (KDC). Each user & service on the network is a principa

Components of kerberos :-

- Authentication Server (AS) :- The authentication server performs the initial authentication & tickets for Tikets Granting Service
- Database :- The authentication server verifies the access rights of users in the database.
- Ticket Granting Server (TGS) :- The Ticket Granting server issues the ticket for the server.

Working of kerberos :-

Step 1 :- The user logs in & requests access to network services. This triggers

a request for Ticket Granting Ticket (TGT).

Step 2 :- The Authentication Server (AS) verifies the users identity using the users password, if verified, it sends back to TGT & a session key, both encrypted with the user's password.

Step 3 :- The user decrypts the message using their password, retries the TGT & sends it to the Ticket Granting Server along with an authenticator.

Step 4 :- The TGS decrypts the TGT, validates the authenticator & if valid, creates a service ticket for the requested server.

Step 5 :- The user sends the service ticket & authenticator to the actual application server.

Step 6 :- The ~~Server~~ Server verifies both the ticket & the authenticator. If everything is correct, it grants access to the requested service.

Q.5 Explain the knapsack cryptosystem.

⇒ i) The knapsack cryptosystem is a type of public key cryptosystem based on the knapsack problem, a difficult computational puzzle.
ii) It involves a public key for encryption & a private key for decryption, allowing secure communication without a pre-shared secret.
iii) The security of knapsack systems relies on the difficulty of solving the knapsack problem, particularly the subset sum proble
iv) In essence, the knapsack cryptosystem concept leverages the computational difficu of the knapsack problem to secure communi-cation.
v) The public key allows encryption, while the private key enables decryption, making it a type of asymmetric encryption.

Q.6 Explain the properties of secure hash function

⇒ i) A secure Hash function is a cryptographi algorithm designed to take an input & return a fixed size string of bytes, typically a digest that appears random. For a hash function to be considered secure, it must satisfy several key properties.

ii) First determinism ensures that the same input always produces the same output. Pre-image resistance means that given a hash output, it is computationally infeasible to determine the original input.
iii) Second pre-image resistance ensures that it is difficult to find a different input that produces the same hash as a given input.
iv) Additionally, collision resistance is crucial. it should be infeasible to find any two distinct inputs that produces the same hash output.

Q.7] Explain CMAC algorithm.

⇒ i) The CMAC (cipher-based message Authentication Code) algorithm is used to ensure the integrity & authenticity of a message. It works by using a block cipher along with a secret key to generate a short, fixed size tag from the message.
ii) This tag is then sent along with the message. The receiver, who also has the secret key, runs the same CMAC process to generate a tag & checks if it matches the one recieved.
iii) If the tags match, the message is trusted, if not, it means the message was tempered with. CMAC is secure, efficient & commonly used in secure communications.

Date :

Q.8] Explain the Attacks on Digital Signature.

⇒ 1. Forgery Attack :- Tries to create fake sig
2. key-only Attack :- The attacker know public key & tries to guess signature
3. Known-message Attack :- The attacker uses priviously signed messages they choose to help them forge another signature.
4. Choosen Message Attack :- The attackers gets signature on messages they choose to help them forge another signature.
5. Reply Attack :- A valid signed message is captered & sent again to trick the ~~see~~ receiver.
6. MITM :- The attacker intercepts & change the message & signature during communication.

Q.9] Explain HMAC algorithm

⇒ HMAC (Hash-based Message Authentication Code) is an algorithm used to check message integrity & authenticity. It combines a secret key with the message, then applies a hash function to produce a fixed-size tag (MAC). This tag is sent along with the message. The receiver using the same secret key, runs the same process to generate a tag & compares it to the received one. If they match, the message is trusted.