SAD Experiment 10

Aim:- Understanding the concepts of cryptography and guidelines for using encryption.

Theory -

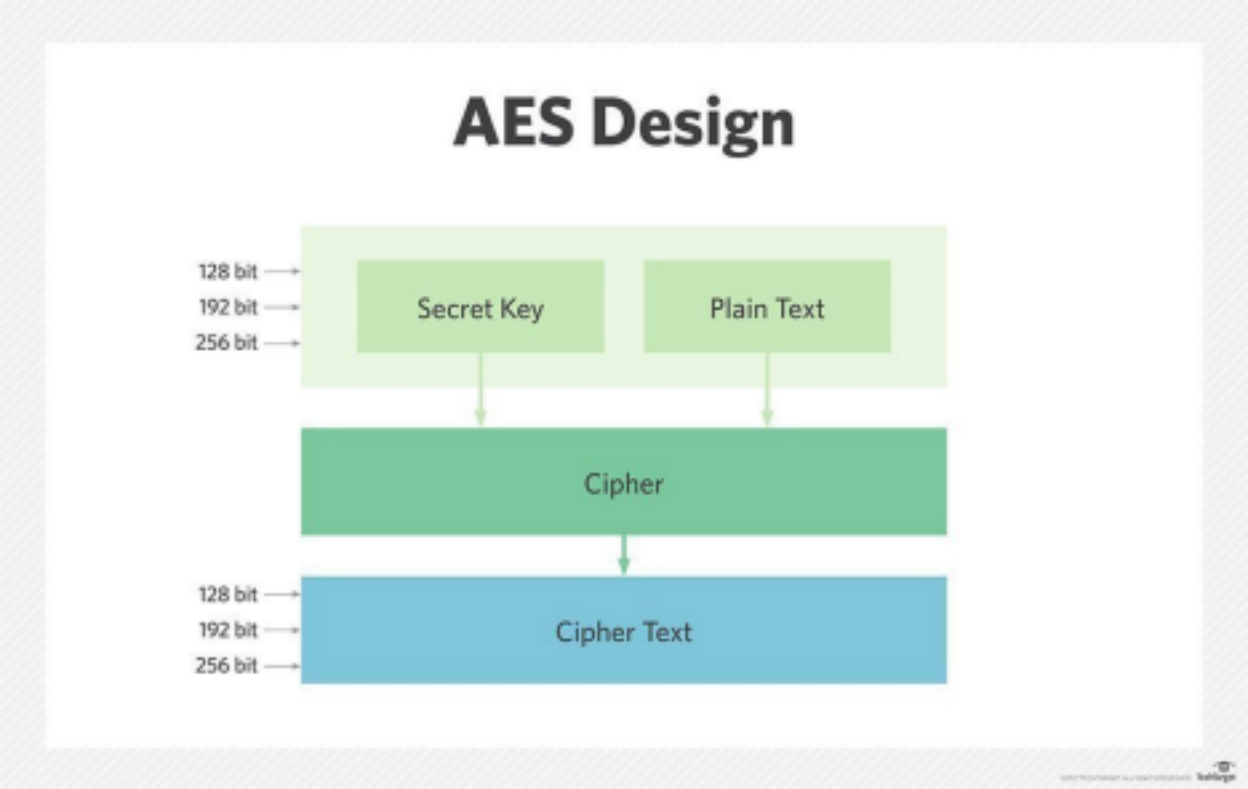
What is AES?

The Advanced Encryption Standard (AES) is a symmetric block cipher chosen by the U.S. government to protect classified information.AES is implemented in software and hardware throughout the world to encrypt sensitive data. It is essential for government computer security, cybersecurity and electronic data protection.The National Institute of Standards and Technology (NIST) started development of AES in 1997 when it announced the need for an alternative to the Data Encryption Standard (DES), which was starting to become vulnerable to brute-force attacks.NIST stated that the newer, advanced encryption algorithm would be unclassified and must be "capable of protecting sensitive government information well into the [21st] century." It was intended to be easy to implement in hardware and software, as well as in restricted environments -- such as a smart card -- and offer decent defenses against various attack techniques.AES was created for the U.S. government with additional voluntary, free use in public or private, commercial or noncommercial programs that provide encryption services. However, nongovernmental organizations choosing to use AES are subject to limitations created by U.S. export control.

How AES encryption works

AES includes three block ciphers:
AES-128 uses a 128-bit key length to encrypt and decrypt a block of messages. AES-192 uses a 192-bit key length to encrypt and decrypt a block of messages. AES-256 uses a 256-bit key length to encrypt and decrypt a block of messages. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128, 192 and 256 bits, respectively.Symmetric, also known as secret key, ciphers use the same key for encrypting and decrypting. The sender and the receiver must both know -- and use -- the same secret key.The government classifies information in three categories: Confidential, Secret or Top Secret. All key lengths can be used to protect the Confidential
and Secret level. Top Secret information requires either 192- or 256-bit key lengths.There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. A round consists of several processing steps that include substitution, transposition and mixing of the input plaintext to transform it into the final output of ciphertext.

Image displaying the relationships between keys, ciphers and ciphertext in AES



AES uses 128-, 192- or 256-bit keys to encrypt and decrypt data.

The AES encryption algorithm defines numerous transformations that are to be performed on data stored in an array. The first step of the cipher is to put the data into an array, after which the cipher transformations are repeated over multiple encryption rounds.

The first transformation in the AES encryption cipher is substitution of data using a substitution table. The second transformation shifts data rows. The third mixes columns. The last transformation is performed on each column using a different part of the encryption key. Longer keys need more rounds to complete.

Data encryption standard (DES)

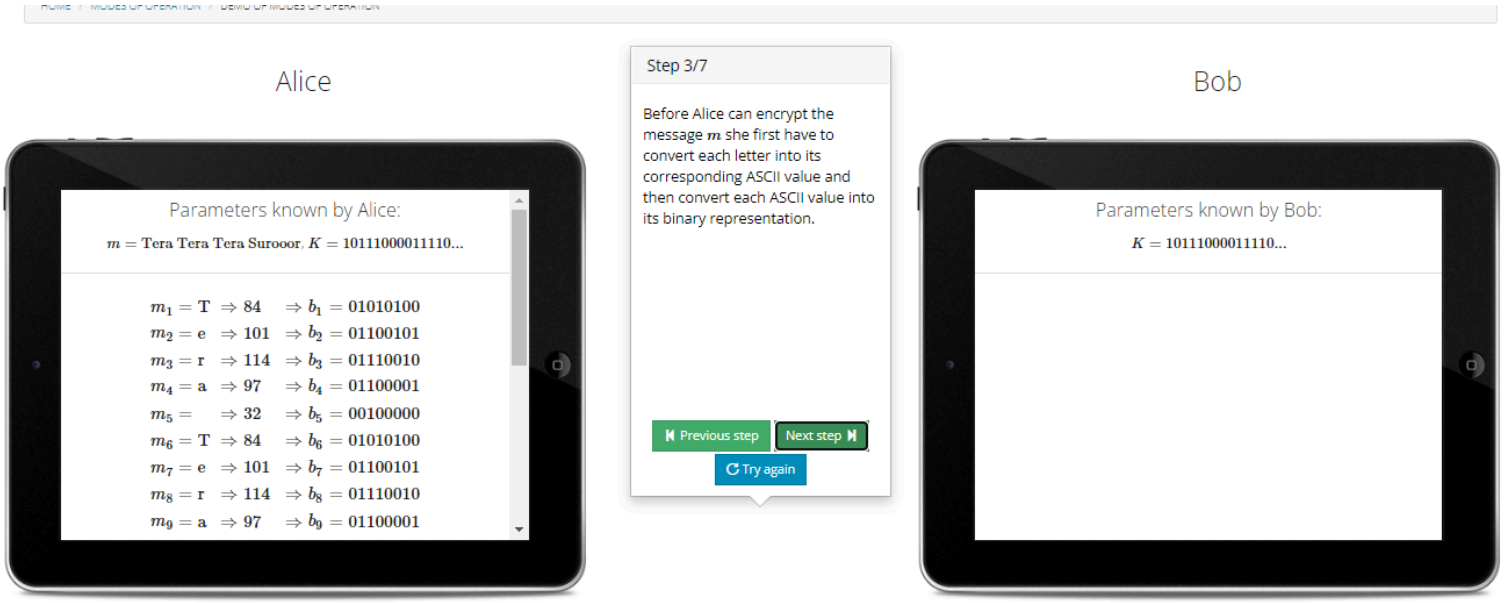
Data encryption standard (DES) has been found vulnerable to very powerful attacks and therefore, the popularity of DES has been found slightly on the decline. DES is a block cipher and encrypts data in blocks of size of 64 bits each, which means 64 bits of plain text go as the input to DES, which produces 64 bits of ciphertext. The same algorithm and key are used for encryption and decryption, with minor differences. The key length is 56 bits. The basic idea is shown in the figure.

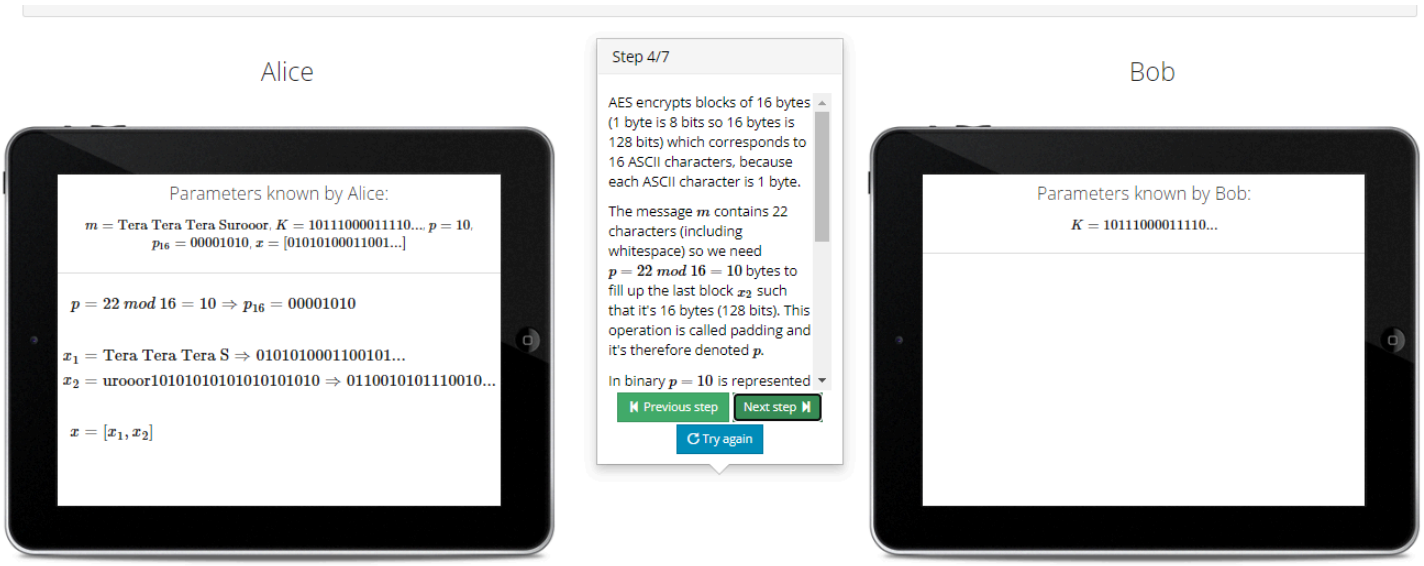
We have mentioned that DES uses a 56-bit key. Actually, the initial key consists of 64 bits. However, before the DES process even starts, every 8th bit of the key is discarded to produce a 56-bit key. That is bit positions 8, 16, 24, 32, 40, 48, 56,

and 64 are discarded. Thus, the discarding of every 8th bit of the key produces a 56-bit key from the original 64-bit key.

DES is based on the two fundamental attributes of cryptography: substitution (also called confusion) and transposition (also called diffusion). DES consists of 16 steps, each of which is called a round. Each round performs the steps of substitution and transposition. Let us now discuss the broad-level steps in DES.

1) In the first step, the 64-bit plain text block is handed over to an initial Permutation (IP) function.
2) The initial permutation is performed on plain text.
3) Next, the initial permutation (IP) produces two halves of the permuted block; saying Left Plain Text (LPT) and Right Plain Text (RPT).
4) Now each LPT and RPT go through 16 rounds of the encryption process.
5) In the end, LPT and RPT are rejoined and a Final Permutation (FP) is performed on the combined block
6) The result of this process produces 64-bit ciphertext.


Alice
Parameters known by Alice:
m = Tera Tera Tera Surooor, K = 10111000011110...
m_1 = T ⇒ 84 ⇒ b_1 = 01010100
m_2 = e ⇒ 101 ⇒ b_2 = 01100101
m_3 = r ⇒ 114 ⇒ b_3 = 01110010
m_4 = a ⇒ 97 ⇒ b_4 = 01100001
m_5 = ⇒ 32 ⇒ b_5 = 00100000
m_6 = T ⇒ 84 ⇒ b_6 = 01010100
m_7 = e ⇒ 101 ⇒ b_7 = 01100101
m_8 = r ⇒ 114 ⇒ b_8 = 01110010
m_9 = a ⇒ 97 ⇒ b_9 = 01100001
Step 3/7
Before Alice can encrypt the message m she first have to convert each letter into its corresponding ASCII value and then convert each ASCII value into its binary representation.
Previous step
Next step
Try again
Bob
Parameters known by Bob:
K = 10111000011110...


Alice
Parameters known by Alice:
m = Tera Tera Tera Surooor, K = 10111000011110..., p = 10, p_{16} = 00001010, x = [01010100011001...]
$p = 22\ mod\ 16 = 10 \Rightarrow p_{16} = 00001010$
x_1 = Tera Tera Tera S ⇒ 0101010001100101...
x_2 = urooor1010101010101010101 ⇒ 0110010101110010...
$x = [x_1, x_2]$
Step 4/7
AES encrypts blocks of 16 bytes (1 byte is 8 bits so 16 bytes is 128 bits) which corresponds to 16 ASCII characters, because each ASCII character is 1 byte.
The message m contains 22 characters (including whitespace) so we need $p = 22\ mod\ 16 = 10$ bytes to fill up the last block x_2 such that it's 16 bytes (128 bits). This operation is called padding and it's therefore denoted p.
In binary $p = 10$ is represented
Previous step
Next step
Try again
Bob
Parameters known by Bob:
K = 10111000011110...


Alice
Parameters known by Alice:
m = Tera Tera Tera Surooor, K = 10111000011110..., p = 10, p_{16} = 00001010, x = [01010100011001...], IV = 00101001111000..., y = [00101001111000.....]
Encrypts the plaintext blocks x:
$y_0 = IV$
$z_i = E_K(y_{i-1})$
$y_i = x_i \oplus z_i$
$y_0 = IV$ = 00101001111000...
$z_1 = E_K(y_0)$ = 01000100001110...
Step 5/7
Alice first chooses the random 128-bits initialization vector IV.
She then uses the key K, the initialization vector IV and the AES encryption function E_K to encrypt the blocks x.
Finally she sends the ciphertext blocks y to Bob.
Previous step
Next step
Try again
Bob
Parameters known by Bob:
K = 10111000011110..., IV = 00101001111000..., y = [00101001111000.....]
Receives the ciphertext blocks y

Alice

Parameters known by Alice:

$m = \text{Tera Tera Tera Surooor}$, $K = 10111000011110...$, $p = 10$, $p_{16} = 00001010$, $x = [01010100011001...]$, $IV = 00101001111000...$, $y = [00101001111000.....]$

Step 6/7

Bob uses the key K, the initialization vector IV and the AES encryption function E_K to decrypt the blocks y.

Previous step | Next step

Try again

Bob

Parameters known by Bob:

$K = 10111000011110...$, $IV = 00101001111000...$, $y = [00101001111000.....]$, $x = [01010100011001...]$

Decrypts the ciphertext blocks y:

$$y_0 = IV$$
$$z_i = E_K(y_{i-1})$$
$$x_i = y_i \oplus z_i$$

$$y_0 = IV = 00101001111000...$$
$$z_1 = E_K(y_0) = 01000100001110...$$
$$x_1 = y_1) \oplus z_1 = 01010100011001...$$

HOME / MODES OF OPERATION / DEMO OF MODES OF OPERATION

Alice

Parameters known by Alice:

$m = \text{Tera Tera Tera Surooor}$, $K = 10111000011110...$, $p = 10$, $p_{16} = 00001010$, $x = [01010100011001...]$, $IV = 00101001111000...$, $y = [00101001111000.....]$

Step 7/7

Bob converts the first byte $b_1 = 01010100$ in the first block x_1 to its integer representation 84 and then to its letter representation T.

Then he converts the rest of the bytes in the blocks x.

Because the decimal value of the last byte $p_{16} = 00001010$ in the last block x_2 is $p = 10$ and between 1 and 15, Bob know that the padding byte p_{16} has been added 10 times at the end of the last block x_2.

Previous step | Next step

Try again

Bob

Parameters known by Bob:

$K = 10111000011110...$, $IV = 00101001111000...$, $y = [00101001111000.....]$, $x = [01010100011001...]$, $p_{16} = 00001010$, $p = 10$, $m = \text{Tera Tera Tera Surooor}$

$p_{16} = 00001010 \Rightarrow p = 10$

$x_1 = 0101010001100101... \Rightarrow$ Tera Tera Tera S

$x_2 = 0110010101110010... \Rightarrow$ urooor10101010101010101010

$m = \text{Tera Tera Tera Surooor}$

Online RSA Key Generator

Key Size 1024 bit

Generate New Keys

Generated in 143 ms

Async

Private Key

```
-----BEGIN RSA PRIVATE KEY-----
MIICWgIBAAKBgG1XONCiQk+Ib/8EFcrUyd5ZIyFoy5OqLKlbPh4/JMXAtC9/rakq
g2BCd8/mOqRG+zL6kl/QV1c/dzYxOdJ7uPn9vokczOM3Y0ulbMbtNSlDspO6u+Zo
5j2ecSSa2HqA6bR7EEBVqDcCNegu9tZhxgPLw+Hd5DklLZamSysju7m9AgMBAAEC
gYBiHM0PcTndUgllJI5FdhE6lnQCi515AE+ZHBWuYGynwhYtNP1oeYxveNU3yVg/
hq3rfajwUzu8P2fCSgOfmoNNKD3KotKCUQ5udXke6y8sxfS5SDb5ahp3cSf2cJT5
VQrrO6LXfG6WfZGPcjSnFR+eggxQPG5XJQ3KbQp5JINsJQJBAMAuZNXABCNSVw9a
LI47NA2nE3i3PROsNq8rVsILu8hofUF/7di3L0ESN9acfsTc1JRf0rlbGYNwmxD7
MaDKcp8CQQCRpm9iqGTgaFpTaMPAXKies/TnG9XUy0Y8jKDaC6RPRNJ/YE
```

Public Key

```
-----BEGIN PUBLIC KEY-----
MIGeMA0GCSqGSIb3DQEBAQUAA4GMADCBiAKBgG1XONCiQk+Ib/8EFcrUyd5ZIyFo
y5OqLKlbPh4/JMXAtC9/rakqg2BCd8/mOqRG+zL6kl/QV1c/dzYxOdJ7uPn9vokc
zOM3Y0ulbMbtNSlDspO6u+Zo5j2ecSSa2HqA6bR7EEBVqDcCNegu9tZhxgPLw+Hd
5DklLZamSysju7m9AgMBAAE=
-----END PUBLIC KEY-----
```

RSA Encryption Test

Text to encrypt:

Give me some sunshine , give me some rain , give me another chance i wanna grow up once again

Encrypt / Decrypt

Encrypted:

RSA Encryption Test

Text to encrypt:

Encrypt / Decrypt

Encrypted:

```
WcdukKMwIlc4RZewtpG2KbVgW9i787cmUMpbepKRQ0kFAwrnY8YfbqABZuqSDgtQcBA6ttzrBgN4DYvTFSYC50hDf8wCBjmR0bCpwL9glqD/6HfqYEmHUC2vs1MeL2JwmIpy+UQWyDolJAxNhd9pVUCyP9VMsPzn+uihEaEWAoc=
```