1.

THREE ASPECTS OF SECURITY • The Open Systems Interconnection (OSI) security architecture offers a systematic framework for defining security attacks, security mechanisms and security services. • security attacks • Security attacks are categorized as either active attacks such as denial of service and message or file alteration, or passive attacks, such as unauthorized reading of a message or file and traffic analysis. • security mechanisms • A security mechanism is any process that is planned to detect, stop, or recover from a security attack.

THREE ASPECTS OF SECURITY • security services. • Security services contain authentication, access control, data confidentiality, data integrity, nonrepudiation, and availability.
ADVANTAGES OF INFORMATION SECURITY :

Reduce the possibility of IT system attacks and data breaches. ¬ Apply security measures to prevent unauthorised access to confidential data. ¬ Prevent service interruptions, such as denial-of-service assaults. ¬ Prevent unauthorised access to IT systems and networks. ¬ Reduce downtime as much as possible to maintain high output. ¬ Maintain company continuity by safeguarding information assets. ¬ Give customers peace of mind by protecting sensitive data from security risks.
DISADVANTAGES OF INFORMATION SECURITY :
Upgrade Frequently: To stay one step ahead of attackers, businesses must regularly upgrade its software, hardware, and security plan. ¬ Needs Continuous Learning: It is necessary to stay ahead of the threats because they are constantly evolving and new. ¬ Setting Up: security architectures and tools, such as firewalls, can be difficult and time-consuming. ¬ Slower Systems: As operating these security software consumes a lot of resources, Systems eventually start to become sluggish. • Continuous Monitoring of new threats is essential to an efficient cyber security plan. The only method for an organisation to identify dangers early on is through continuous system and network monitoring

2. Confidentiality, integrity and availability, also known as the CIA triad, is a model designed to guide policies for information security within an organization. The model is also sometimes referred to as the AIC triad (availability, integrity and confidentiality) to avoid confusion with the Central Intelligence Agency. Although elements of the triad are three of the most foundational and crucial cybersecurity needs, experts believe the CIA triad needs an upgrade to stay effective.

In this context, confidentiality is a set of rules that limits access to information, integrity is the assurance that the information is trustworthy and accurate, and availability is a guarantee of reliable access to the information by authorized people.

Confidentiality, integrity, availability

The following is a breakdown of the three key concepts that form the CIA triad:

- **Confidentiality** is roughly equivalent to privacy. Confidentiality measures are designed to prevent sensitive information from unauthorized access attempts. It is common for data to be categorized according to the amount and type of damage that could be done if it fell into the wrong hands. More or less stringent measures can then be implemented according to those categories.
- **Integrity** involves maintaining the consistency, accuracy and trustworthiness of data over its entire lifecycle. Data must not be changed in transit, and steps must be taken to ensure data cannot be altered by unauthorized people (for example, in a breach of confidentiality).
- **Availability** means information should be consistently and readily accessible for authorized parties. This involves properly maintaining hardware and technical infrastructure and systems that hold and display the information.

The three CIA triad principles

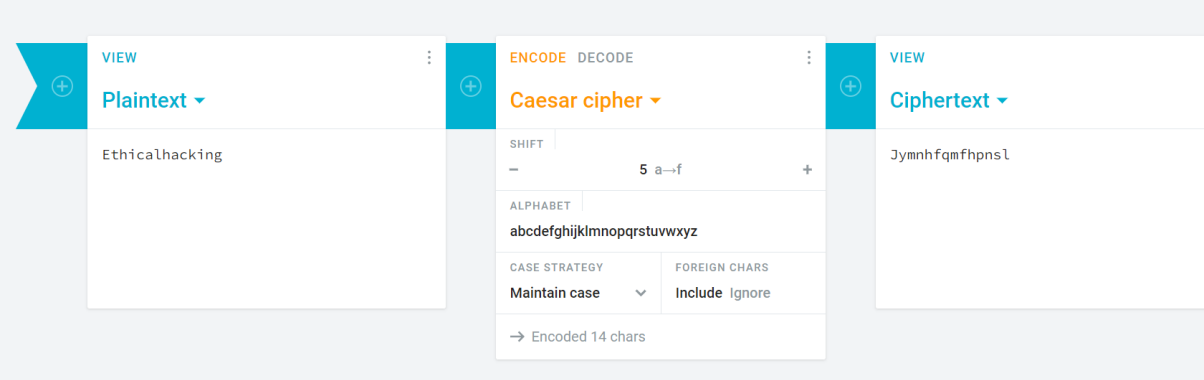
Why is the CIA triad important?

With each letter representing a foundational principle in cybersecurity, the importance of the CIA triad security model speaks for itself. Confidentiality, integrity and availability together are considered the three most important concepts within information security.

Considering these three principles together within the framework of the "triad" can help guide the development of security policies for organizations. When evaluating

needs and use cases for potential new products and technologies, the triad helps organizations ask focused questions about how value is being provided in those three key areas.

3.



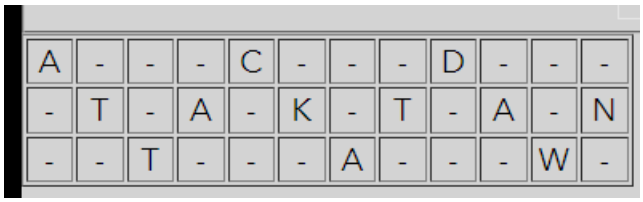
Input:Ethical hacking

Output: Jymnhfqmfhpnsl

4.
input: attackatdawn

Key-3

Output: ACDTAKTANTAW



5. Hill Cipher is a polygraphic substitution cipher that operates on blocks of letters. In this case, you want to use a 2x2 matrix as the key for encryption and decryption. Let's start with the encryption process and then move on to decryption.

Encryption:

1. Assign numbers to the letters using the alphabet, starting from 0:
   - S -> 18
   - H -> 7

    - O -> 14
    - R -> 17
2. Create the 2x2 key matrix using the key "HILL":

```
H I

L L
```

Hill Cipher is a polygraphic substitution cipher that operates on blocks of letters. In this case, you want to use a 2x2 matrix as the key for encryption and decryption. Let's start with the encryption process and then move on to decryption.

Encryption:

1. Assign numbers to the letters using the alphabet, starting from 0:
    - S -> 18
    - H -> 7
    - O -> 14
    - R -> 17
2. Create the 2x2 key matrix using the key "HILL":

- ```
  H  I
  L  L
  ```

- Multiply the key matrix by the column vector of the plaintext:

```
| 18 |    | 7 |
| 14 |  x | 8 |
```

Perform matrix multiplication:

- ```
  | 18*7 + 14*8 |   | 126 + 112 |   | 238 |
  | 18*8 + 14*8 | = | 144 + 112 | = | 256 |
  ```

- Take the modulo 26 of each result to get the ciphertext values:

```
4. 238 % 26 = 12 (C)
5. 256 % 26 = 2  (C)
```
6.

So, the encrypted ciphertext for "SHOR" using the Hill Cipher with the key "HILL" is "CC".

Decryption:

To decrypt, we need to find the inverse of the key matrix, which can be a bit complex for a general 2x2 matrix. However, for the given key matrix "HILL," it happens to be a special case where the determinant is non-zero and the inverse can be found easily.

1. Calculate the determinant of the key matrix:

- `Det(HILL) = (H * L) - (I * L) = (7 * 12) - (8 * 14) = 84 - 112 = -28`

- Find the multiplicative inverse of the determinant modulo 26 (since we're working with the alphabet):

- `Multiplicative Inverse of -28 mod 26 = Multiplicative Inverse of 24 mod 26 = 19`

- Find the adjugate matrix of the key matrix (swap the diagonals and negate the off-diagonals):

- `Adj(HILL) = | L  -I |`
  `| -L   H |`

- Multiply the adjugate matrix by the multiplicative inverse of the determinant:

- `Inv(HILL) = 19 * | L  -I | = | 19L  -19I |`
  `| -L   H   |   | -19L  19H |`

- Now, encrypt the ciphertext "CC" using the inverse key matrix:

```
| 19L  -19I |   | 2 |
| -19L  19H | x | 2 |
```

Perform matrix multiplication:

- `| 19*2 + (-19)*2 |   | 38 - 38 |   | 0 |`
  `| (-19)*2 + 19*2 | = | (-38) + 38 | = | 0 |`

- Take the modulo 26 of each result to get the plaintext values:

6. `0 % 26 = 0 (A)`
7. `0 % 26 = 0 (A)`
8.

So, the decrypted plaintext for "CC" using the Hill Cipher with the key "HILL" is "AA," which is the original plaintext "SHOR."

6. The first step is to generate the sub-keys.

This is called Key Generation or Key Expansion: The input key, K, is split into 2 words, w0 and w1: w0 = 0100 1010 w1 = 1111 0101 The first sub-key, Key0, is in fact just the input key: Key0 = w0w1 = K The other sub-keys are generated as follows:

w2 = w0 XOR 10000000 XOR SubNib(RotNib(w1)) (Note: RotNib() is “rotate the nibbles”, which is equivalent to swapping the nibbles) = 0100 1010 XOR 10000000 XOR SubNib(0101 1111) (Note: SubNib() is “apply S-Box substitution on nibbles using encryption S-Box”) = 1100 1010 XOR SubNib(0101 1111) = 1100 1010 XOR 0001 0111 = 1101 1101

w3 = w2 XOR w1 = 1101 1101 XOR 1111 0101 = 0010 1000 2

w4 = w2 XOR 0011 0000 XOR SubNib(RotNib(w3)) = 1101 1101 XOR 0011 0000 XOR SubNib(1000 0010) = 1110 1101 XOR 0110 1010 = 1000 0111 w5 = w4 XOR w3 = 1000 0111 XOR 0010 1000 = 1010 1111 3

Now the sub-keys are: Key0 = w0w1 = 0100 1010 1111 0101 Key1 = w2w3 = 1101 1101 0010 1000 Key2 = w4w5 = 1000 0111 1010 1111 1.2 Encryption Now let's do the encryption. There is an initial operation (Add Round Key), followed by the main Round, followed by the final Round.

(Note, the main difference in the real DES is that the main Round is repeated many times). Remember, the output of each operation is used as the input to the next operation, always operating on 16-bits. The 16-bits can be viewed as a state matrix of nibbles. 1.2.1 Add Round 0 Key Plaintext XOR Key1 = 1101 0111 0010 1000 XOR 0100 1010 1111 0101 = 1001 1101 1101 1101 1.2.2

Round 1 Nibble Substitution (S-boxes). Each nibble in the input is used in the Encryption S-Box to generate an output nibble. Input = 1001 1101 1101 1101 Output = 0010 1110 1110 1110 Shift Row. Swap 2nd nibble and 4th nibble (note, in this example, its not so easy to see since 2nd and 4 th nibbles are the same!) = 0010 1110 1110 1110 Mix Columns.

Apply the matrix multiplication with the constant matrix, Me, using GF(24). For GF(24), the addition operation is simply an XOR, and for the multiplication operation you can use a lookup table.

Me = 1 4 4 1 S = 0010 1110 = S00' S01' 1110 1110 S10' S11' 4 S' = Me x S S00' = = = = = 0010 XOR (4 x 1110) 0010 XOR (4 x E) 0010 XOR D 0010 XOR 1101 1111 S10' = = = (4 x 0010) XOR 1110 1000 XOR 1110 0110 S01' = = = = 1110 XOR (4 x 1110) 1110 XOR (4 x E) 1110 XOR 1101 0011 S11' = = = (4 x 1110) XOR 1110 1101 XOR 1110 0011 Output = = S00' S10' S01' S11' 1111 0110 0011 0011 Add Round 1 Key. = 1111 0110 0011 0011 XOR 1101 1101 0010 1000 = 0010 1011 0001 1011 1.2.3

Final Round Nibble Substitution (S-boxes) = 1010 0011 0100 0011 Shift Row (2nd and 4 th) = 1010 0011 0100 0011 Add Round 2 Key 1010 0011 0100 0011 XOR 1000 0111 1010 1111 = 0010 0100 1110 1100 Now we have the final ciphertext. Ciphertext = 0010 0100 1110 1100 5 1.3

Decryption Now lets decrypt. Note that we use the same keys generated during the encryption (that is, the decryptor would generate the round sub-keys using the input key K, using the encryption S-Box).

Add Round 2 Key 0010 0100 1110 1100 XOR 1000 0111 1010 1111 = 1010 0011 0100 0011 Inverse Shift Row (same as normal) = 1010 0011 0100 0011

Inverse Nibble Sub (use the inverse or decryption S-box) = 0010 1011 0001 1011

Add Round 1 Key = 0010 1011 0001 1011 XOR 1101 1101 0010 1000 = 1111 0110 0011 0011 Inverse Mix Columns S = S00 S10 S01 S11 = 1111 0011 0110 0011 S' = S00' S10' S01' S11' = 9 x S00 XOR 2 x S10 2 x S00 XOR 9 x S10 9 x S01 XOR 2 x S11 2 x S01 XOR 9 x S11 S00' = = = = = (9 x 1111) XOR (2 x 0110) 9 x F XOR 2 x 6 E XOR C 1110 XOR 1100 0010 S10' = = = = = 2 x 1111 XOR 9 x 0110 2 x F XOR 9 x 6 D XOR 3 1101 XOR 0011 1110 S01' = = = = = 9 x 0011 XOR 2 x 0011 9 x 3 XOR 2 x 3 8 XOR 6 1000 XOR 0110 1110 6 S11' = 2 x 0011 XOR 9 x 0011 = 1110 Output = 0010 1110 1110 1110

Inverse Shift Row = 0010 1110 1110 1110 Inverse Nibble Sub = 1001 1101 1101 1101

Add Round 0 Key = 1001 1101 1101 1101 XOR 0100 1010 1111 0101 = 1101 0111 0010 1000 Plaintext = 1101 0111 0010 1000 Original = 1101 0111 0010 1000

The decryption worked!

7. Random number generators (RNGs) play a crucial role in both cryptography and cybersecurity. The properties of RNGs are of paramount importance in these fields, as they impact the security and effectiveness of various cryptographic algorithms and cybersecurity measures. Here are some key properties of RNGs in the context of cryptanalysis and cyber defense:

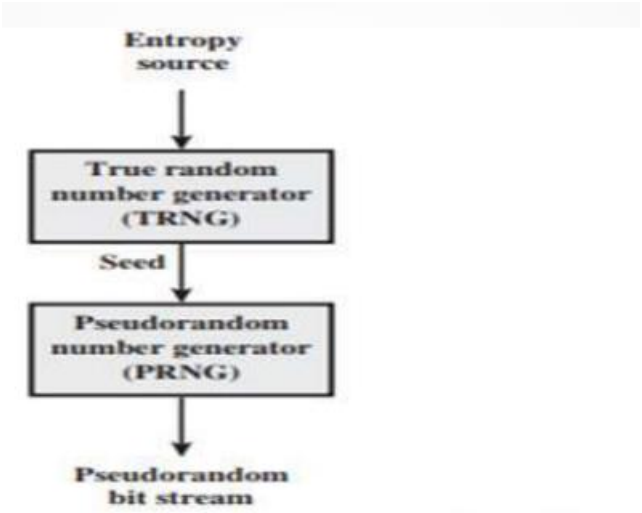
1. **Uniformity**: Random numbers generated by an RNG should be uniformly distributed. This means that each possible output value has an equal probability of occurring. Non-uniform RNGs can introduce biases and vulnerabilities, making cryptographic systems easier to attack.
2. **Independence**: Random numbers should be independent of each other. This property ensures that knowledge of one generated number does not provide any information about the next number in the sequence. Lack of independence can lead to patterns that attackers can exploit.
3. **Determinism**: In many cryptographic applications, it's essential that RNGs produce the same sequence of numbers when given the same initial state or seed. This determinism allows for reproducibility, which is necessary for security protocols to work consistently.
4. **Periodicity**: RNGs should have a long and unpredictable period. The period is the number of values the RNG can generate before repeating. Short periods can lead to the repetition of sequences, which is a significant security weakness.

8. Pseudorandom Number Generation

Random Numbers A random number is a value that is generated unpredictably and lacks any discernible pattern or relationship to previously generated values. Random numbers play a crucial role in various fields, including cryptography, simulations, statistical analysis, gaming, and more.

True Random Numbers (TRNG): • True random numbers are generated from unpredictable physical processes, such as atmospheric noise,radioactive decay, or thermal noise. • TRNGs provide genuinely random and unbiased numbers, as they are derived from natural phenomena. • However, generating true random numbers can be challenging and often requires specialized hardware.

Pseudorandom Numbers (PRNG): • Pseudorandom numbers are generated by deterministic algorithms that use mathematical formulas and a seed value to produce a sequence of numbers that appears random. • PRNGs are repeatable, meaning that with the same seed value, they will produce the same sequence of numbers. • However, the generated numbers are not truly random, but rather exhibit statistical randomness and pass various tests for randomness.



Generation of Seed Input to PRNG

12.

The Advanced Encryption Standard (AES),- is a block cipher adopted as an encryption standard by the U.S. government for military and government use.

ECB (Electronic Codebook) - is essentially the first generation of the AES. It is the most basic form of block cipher encryption.

CBC (Cipher Blocker Chaining) - is an advanced form of block cipher encryption. With CBC mode encryption, each ciphertext block is dependent on all plaintext blocks processed up to that point. This adds an extra level of complexity to the encrypted data.

Cryptography methods such as Electronic Code Book (ECB) and Cipher Block Chaining (CBC) are widely used.

ECB is a simple method of encrypting plaintext by dividing it into fixed-size blocks and encrypting each block independently using the same secret key. In other words, if the same plaintext block appears more than once in the message, it will be encrypted into the same ciphertext block (aka will look the same). The ECB encryption method is relatively easy to implement; however, it can be vulnerable to certain types of attacks, such as pattern recognition.

By contrast, CBC is a more secure encryption method that addresses the weaknesses of ECB. CBC encrypts plaintext blocks using the same key and combines them with the previous ciphertext blocks through an operation called an XOR. Thus, even if the same plaintext block appears multiple times in the message, it will be encrypted to a different ciphertext block each time.

The major difference between ECB and CBC is that ECB encrypts each block independently, whereas CBC encrypts each block with the previous block. CBC is

therefore considered more secure and resistant to pattern recognition attacks than ECB.

Implementation of CBC mode requires an initialization vector (IV), which is a random value added to the first plaintext block before encryption. An IV is sent along with an encrypted message, so the receiver can use it to decrypt it.

ECB and CBC are symmetric-key encryption methods, meaning that the same key is used for encryption and decryption. As computing power increases, it becomes increasingly important to use more secure encryption methods, such as AES-GCM or RSA-OAEP.