1. a) Security in a message provides some mandatory basic services termed as CIA that means confidentiality, integrity, and availability. Explain each service briefly with a suitable example. What are the techniques that can be applied to ensure each of the above-mentioned service respectively? 6

Ans: The CIA Triad—Confidentiality, Integrity, and Availability—is a guiding model in information security. The CIA triad guides the information security in a broad sense and is also useful for managing the products and data of research.

Confidentiality

Confidentiality refers to protecting information from unauthorized access.

Integrity

Integrity means data are trustworthy, complete, and have not been accidentally altered or modified by an unauthorized user.

Availability

Availability means means that theauthorized users should be able to access data whenever required.

A comprehensive information security strategy includes policies and security controls that minimize threats to these three crucial components.

**CIA Triad Examples**

To have a better understanding of how the CIA triad works in practice, consider an ATM that allows users to access bank balances and other information. An ATM incorporates measures to cover the principles of the triad:

* The two-factor authentication (debit card with the PIN code) provides **confidentiality** before authorizing access to sensitive data.
* The ATM and bank software ensure data **integrity**by maintaining all transfer and withdrawal records made via the ATM in the user’s bank accounting.
* The ATM provides **availability** as it is for public use and is accessible at all times.

To fight against confidentiality breaches, you can classify and label restricted data, enable access control policies, encrypt data, and use multi-factor authentication (MFA) systems.

* Categorize data and assets being handled based on their privacy requirements.
* Require data encryption and two-factor authentication to be basic security hygiene.
* Ensure that access control lists, file permissions and white lists are monitored and updated regularly.
* Ensure organization have the training and knowledge they need to recognize the dangers and avoid them

Ensuring data integrity involves protecting the data at all times, including when it is being used, transmitted, or stored. This includes implementing measures to prevent unauthorized access, data corruption, or tampering during these various stages. Countermeasures like encryption, **digital signatures**, hashing, and digital certificates can help maintain data integrity. Aside from these, intrusion detection systems, strong authentication mechanisms, version control, auditing, and access controls can ensure integrity.

Availability can be ensured through network, server, application, and service redundancy. Hardware fault tolerance in servers and storage is another good countermeasure to avoid violation of availability. DoS protection solutions, system upgrades, regular software patching, comprehensive disaster recovery plans, backups, etc. are all ways to ensure availability.

b) Briefly explain symmetric cipher and asymmetric cipher with example. Discuss the advantages of symmetric cipher over asymmetric cipher.4+1

Ans: Symmetric cipher is **a cryptographic algorithm that uses a single key to encrypt (encode) and decrypt (decode) data.** Ex-AES, DES

A symmetric encryption scheme has five ingredients:

• **Plaintext:** This is the original message or data that is fed into the algorithm as input.

• **Encryption algorithm:** The encryption algorithm performs various substitutions and transformations on the plaintext.

• **Secret key:** The secret key is also input to the encryption algorithm. The exact substitutions and transformations performed by the algorithm depend on the key.

• **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts.

• **Decryption algorithm:** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

Asymmetric cryptography, also known as *public key cryptography*, is a process that uses a pair of related [keys](https://www.techtarget.com/searchsecurity/definition/key) -- one public key and one private key -- to [encrypt](https://www.techtarget.com/searchsecurity/definition/encryption) and decrypt a message and protect it from unauthorized access or use.

A [public key](https://www.techtarget.com/searchsecurity/definition/public-key) is a cryptographic key a person can use to encrypt a message so it can only be decrypted by the intended recipient with their private key. A [private key](https://www.techtarget.com/searchsecurity/definition/private-key) -- also known as a *secret key* -- is shared only with the key's initiator. Examples of asymmetric encryption algorithms include RSA, Diffie-Hellman, and Elliptic Curve Cryptography (ECC).

Disadvantages of asymmetric cryptography include the following:

* It's a slow process compared to symmetric cryptography. It's, therefore, not appropriate for decrypting bulk messages.
* If an individual loses their private key, they can't decrypt the messages they receive.
* Because public keys aren't authenticated, no one can ensure a public key belongs to the person specified. Consequently, users must verify their public keys belong to them.

c) Explain the term vulnerability, threat and attack with a suitable example.4

Ans: Threat : A potential for violation of security, which exists when there is a circumstance, capability, action, or event, that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.

Vulnerability :A flaw or weakness in a system’s design, implementation, or operation and management that could be exploited to violate the system’s security policy.

Attack : An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

2. a) Comments on the key size, plaintext block size, number of rounds, round key size, and expanded key size in standard AES (128-bit AES) algorithm. What basic functions are used in AES? 4+2

Key Size -128 bits

Block size-128 bit

Round-10

Round key size-128 bits(4 words)

Expanded key size-44/176

b) What is the purpose of the State array in AES? Briefly explain the operation of the substitute byte transformation in AES.1+2

e) Briefly explain the mutual authentication and challenge-response authentication scheme. What are the drawbacks of password-based authentication and how it could be addressed?4+2

**Mutual authentication** is an authentication mechanism in which both parties in a transaction authenticate each other. These parties are typically software-based. In the standard, one-way authentication process, the client authenticates to the server. In mutual authentication, not only does the client authenticate to the server, but the server authenticates to the client. Mutual authentication often relies on digital certificates.

Challenge-response authentication refers to a set of protocols that helps validate actions to protect digital assets and services from unauthorized access. This protocol usually has two components – a question and a response – where a verifier presents a challenge to a user, who must provide a correct answer for authentication. Challenge-response protocols can be as simple as a password or a dynamically generated request.

A challenge-response authentication mechanism, or CRAM, provides businesses with an easy-to-use tool that they can use to control access to sensitive information and identify bad actors.

**Drawbacks of password-based authentication**

If you construct a password that uses lowercase letters only and is eight characters long, you can use a password-cracking utility to crack it quickly.

Don’t write your password down and post it under your keyboard or on your monitor; doing so completely defeats the purpose of having a password in the first place

Another common problem is the manual synchronization of passwords—in short, using the same password everywhere. If you use the same password for your email, for your login at work, and for your online knitting discussion forum, you’re putting the security of all the accounts in the hands of those system owners. If any one of them is compromised, all of your accounts become vulnerable; all an attacker needs to do to access the others is look up your account name on the internet to find your other accounts and log in using your default password. By the time the attacker gets into your email account, the game is over because an attacker can generally use it reset account credentials for any other accounts you have

**AES**

Some of the pertinent facts of AES are as follows.

• The block size is 128 bits.

• Three key lengths are available: 128, 192, or 256 bits.

• The number of rounds varies from 10 to 14, depending on the key length.

• Tt is not a Feistel structure. Recall that in the classic Feistel structure, half of the data block is used to modify the other half of the data block, and then the halves are swapped. AES does not use a Feistel structure but processes the entire data block in parallel during each round using substitutions and permutation.

• Each round consists of four functions, in three layers—the functions are listed below, with the layer in parentheses.

*—* ByteSub (nonlinear layer)

— ShiftRow (linear mixing layer) , A simple permutation

— MixColumn (nonlinear layer)

— AddRoundKey (key addition layer)

The input to the encryption and decryption algorithms is a single 128-bit block, this block is depicted as a square matrix of bytes. This block is copied into the **State** array, which is modified at each stage of encryption or decryption. After the final stage, **State** is copied to an output matrix.

Similarly, the 128-bit key is depicted as a square matrix of bytes. This key is then expanded into an array of key schedule words; each word is 4 bytes and the total key schedule is 44 words for the 128-bit key. The ordering of bytes within a matrix is by column.

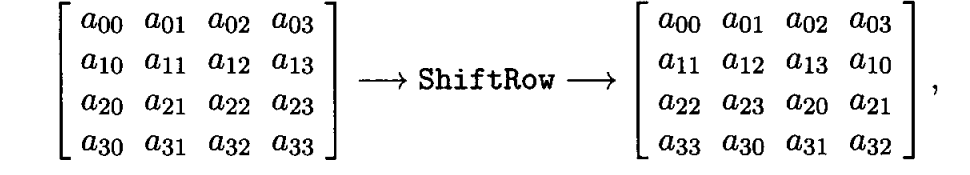
ByteSub, which is roughly the AES equivalent of the DES S-boxes, can be viewed as a nonlinear—but invertible—composition of two mathematical functions.

***Substitute Bytes Transformation***

The **forward substitute byte transformation**, called SubBytes, is a simple table lookup. AES defines a 16·16 matrix of byte values, called an S-box (Table 20.2a), that contains a permutation of all possible 256 8-bit values. Each individual byte of **State** is mapped into a new byte in the following way: 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. For example, the hexadecimal value4 {95} references row 9, column 5 of the S-box, which contains the value {2A}. Accordingly, the value {95} is mapped into the value {2A}.

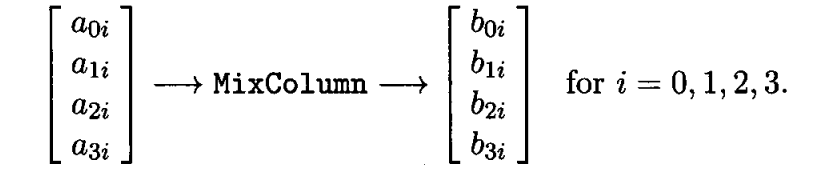
The **inverse substitute byte transformation**, called InvSubBytes, makes use of the inverse S-box . The S-box is designed to be resistant to known cryptanalytic attacks. Specifically, the AES developers sought a design that has a low correlation between input bits and output bits and the property that the output cannot be described as a simple mathematical function of the input.

The ShiftRow operation is a cyclic shift of the bytes in each row of the 4x 4byte array. That is, the first row of **State** doesn't shift, the second-row circular left-shifts by one byte, the third row left-shifts by two bytes, and the last row left-shifts three bytes. Note that ShiftRow is inverted by simply shifting in the opposite direction.



The **inverse shift row transformation**, called InvShiftRows, performs the circular shifts in the opposite direction for each of the last three rows, with a 1-byte circular right shift for the second row, and so on.

the MixColumn operation is applied to each column of the 4x 4 byte array as indicated below:



***Mix Column Transformation*** The **forward mix column transformation**, called MixColumns, operates on each column individually. Each byte of a column is mapped into a new value that is a function of all 4 bytes in the column. The mapping makes use of equations over finite fields. The mapping is designed to provide a good mixing among the bytes of each column. The mix column transformation combined with the shift row transformation ensures that after a few rounds, all output bits depend on all input bits.

**Add Round key:** A simple bitwise XOR of the current block with a portion of the expanded key. Only the Add Round Key stage makes use of the key. For this reason, the cipher begins and ends with an Add Round Key stage. Any other stage, applied at the beginning or end, is reversible without knowledge of the key and so would add no security.

For the Add Round Key stage, the inverse is achieved by XORing the same round key to the block, using the result that A ⊕ A ⊕ B = B.

In the **forward add round key transformation**, called AddRoundKey, the 128 bits of **State** are bitwise XORed with the 128 bits of the round key. The operation is viewed as a column-wise operation between the four bytes of a **State** column and one word of the round key; it can also be viewed as a byte-level operation.

The **inverse add round key transformation** is identical to the forward add round key transformation.

***AES Key Expansion*** The AES key expansion algorithm takes as input a 4-word (16-byte) key and produces a linear array of 44 words (156 bytes). This is sufficient to provide a 4-word round key for the initial Add Round Key stage and each of the 10 rounds of the cipher.

The key is copied into the first four words of the expanded key. The remainder of the expanded key is filled in four words at a time. Each added word **w**[i] depends on the immediately preceding word, **w**[i - 1], and the word four positions back, **w**[i - 4]. A complex finite-field algorithm is used in generating the expanded key.

A **block cipher** processes the input one block of elements at a time, producing an output block for each input block. A **stream cipher** processes the input elements continuously, producing output one element at a time, as it goes along. Although block ciphers are far more common, there are certain applications in which a stream cipher is more appropriate.

Block vs Stream Cipher

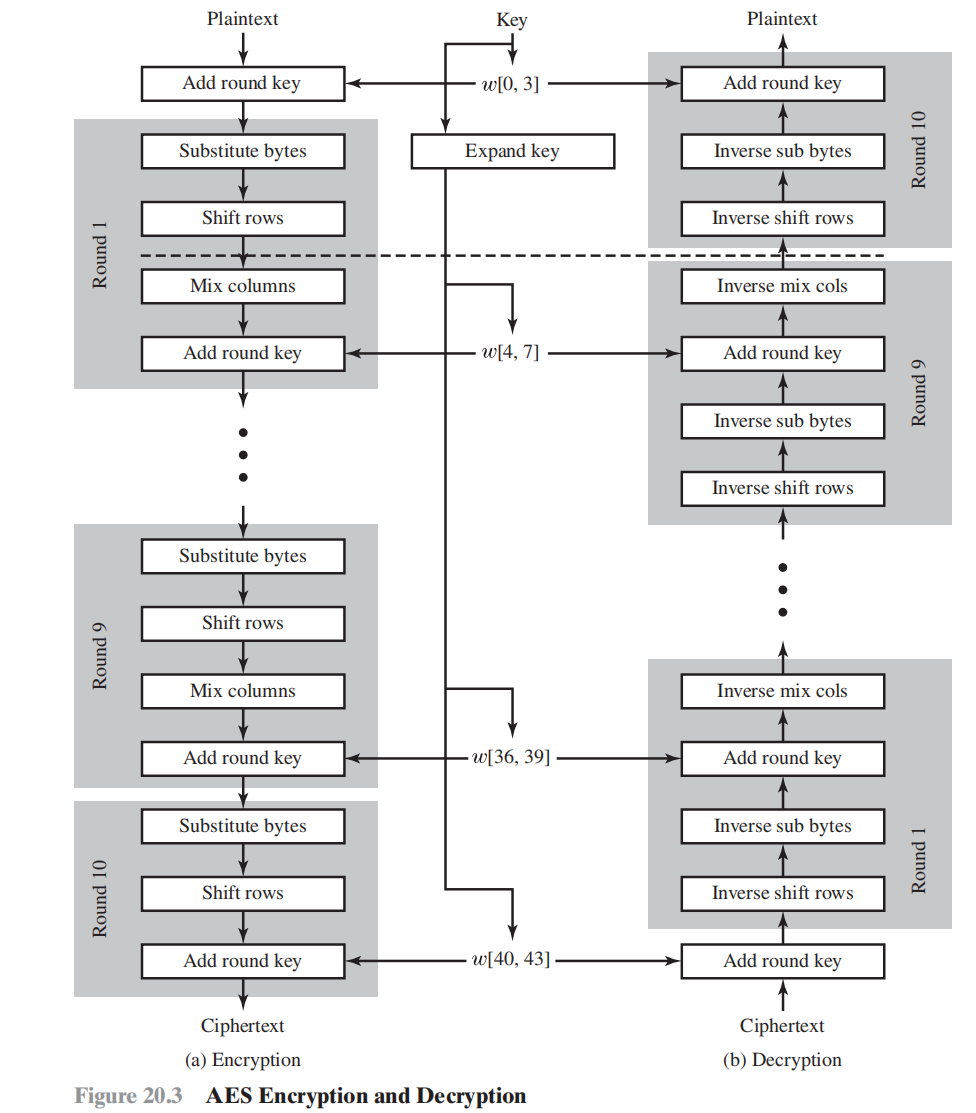
The majority of the encryption algorithms currently in use are block ciphers. Although block ciphers are often slower than stream ciphers,

Block cipher tend to be more versatile.

Since block ciphers operate on larger blocks of the message at a time, they’re usually more resource intensive and more complex to implement.

They’re also more prone to errors in the encryption process.

For example, an error in block cipher encryption would render a large segment of data unusable, whereas in a stream cipher, an error would corrupt only a single bit.



The term *S-box*, or substitution box, is commonly used in the description of symmetric ciphers to refer to a table used for a table-lookup type of substitution mechanism.

There are two general approaches to attacking a symmetric encryption scheme. The first attack is known as **cryptanalysis**. Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext-ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used. If the attack succeeds in deducing the key, the effect is catastrophic: All future and past messages encrypted with that key are compromised.

The second method, known as the **brute-force attack**, It is important to note that there is more to a brute-force attack than simply running through all possible keys. Unless known plaintext is provided, the analyst must be able to recognize plaintext as plaintext. If the message is just plain text in English, then the result pops out easily, although the task of recognizing English would have to be automated. If the text message has been compressed before encryption, then recognition is more difficult. And if the message is some more general type of data, such as a numerical file, and this has been compressed, the problem becomes even more difficult to automate. Thus, to supplement the brute-force approach, some degree of knowledge about the expected plaintext is needed, and some means of automatically distinguishing plaintext from garble is also needed.

*Confidentiality* deals with preventing unauthorized reading of information.

*Integrity* deals with preventing, or at least detecting, unauthorized changes to data.