Midterm

1. Which of the following activities might be considered as a possible breach of security to a company's network? (12 points)

(a) The daily carrier personnel who drop off and picks up packages.

(b) Former employees who left the company because of downsizing.

(c) An employee traveling on company business to another city.

(d) The building management company where an organization has its offices has decided to install a fire sprinkler system.

Ans:

(a) The daily carrier personnel responsible for package delivery and pickup pose no security threat to the company's network unless there is clear evidence of malicious intent. Their role primarily pertains to physical security and package management.

(b) Former employees who left the company because of downsizing: This situation can be a significant breach of security if proper measures are not taken. When employees leave an organization, especially in cases of downsizing, it's crucial to ensure that their access to the company's network, systems, and sensitive information is promptly revoked or controlled. If this process is not managed effectively, former employees might retain access or knowledge of security practices, potentially leading to unauthorized access, data breaches, or insider threats.

(c) An employee traveling on company business to another city is a common and legitimate activity. It doesn't necessarily indicate a breach of security unless there are specific concerns related to the employee's behavior or the security of their devices while traveling.

(d) The building management company installing a fire sprinkler system: This activity is primarily building-related and doesn't directly impact network security. Installing a fire sprinkler system is a safety measure, not a threat to network security. However, it's essential to ensure that building-related activities do not compromise physical security measures, like server rooms or network infrastructure housed within the building. While the installation itself may not pose a threat, it's still essential to consider how it might affect the overall security of the company's assets.

So, my answer is (b).

2. Consider a sensor X that periodically sends a 64-octet measurement to a receiver Y. One day the administrator decides that X should encrypt the measurement data using DES in CBC mode. How many octets does X now send for each measurement?

Explain your answer. (8 points)

Ans:

Initially, DES processes an 8-octet (64-bit) plaintext block, resulting in an 8-octet cipherblock.

Additionally, to meet the requirements of CBC mode, an 8-octet initialization vector (IV) must be included with the cipherblocks.

Therefore, X now transmits a total of 72 octets, consisting of 64 octets of cipherblocks and 8 octets of IV.

3. DES is insecure because of its short key length (56 bits). A modified DES algorithm has key length of 120 bits, $k = (k_1, k_2)$, where k_1 is 56 bits and k_2 is 64 bits. The new encryption algorithm is as follows.

$$DES'(k, m) = k_2 \oplus DES\big(k_1, (k_2 \oplus m)\big)$$

Explain how decryption is done. (10 points)

Ans:

Begin with the ciphertext C and the 120-bit key $k = (k_1, k_2)$. Calculate the intermediate result, which is equivalent to the ciphertext being XORed with the second part of the key k_2:

$$Intermediate\ result = C \oplus k_2$$

Apply the standard DES decryption using k_1 and the intermediate result:

$$Plaintext = DES\ Decryption\ (k_1, Intermediate\ result)$$

The DES Decryption function is the standard DES decryption process that uses k_1 as the key. This process reverses the encryption that was applied using k_1 during encryption.

By performing these steps, you effectively decrypt the ciphertext using the two components of the key (k_1, k_2), as described in the modified DES algorithm. This approach enhances security by increasing the key length while still allowing for backward compatibility with the original DES algorithm for decryption.

4. Briefly explain the Shift Rows and Byte Substitution layers in AES

algorithm. What happens if we change their order in the AES algorithm (Shift Rows and then Byte Substitution)? (10 points)

Ans:

Shift Rows:

In this step, the rows of the state (a 4x4 matrix representing the data) are shifted to the left. The first row remains unchanged, the second row is shifted one position to the left, the third row is shifted two positions to the left, and the fourth row is shifted three positions to the left. This step provides diffusion in the data and ensures that different bytes interact with each other.

Byte Substitution:

In this step, each byte of the input data is substituted with a corresponding byte from a fixed substitution table called the S-box. This substitution adds non-linearity to the encryption process and helps to provide confusion in the data.

The order of these layers is critical to the security and correctness of the AES algorithm. If you were to change their order and perform Shift Rows before Byte Substitution, you would significantly alter the algorithm's behavior. This change would result in a less secure and different encryption scheme:

Performing Shift Rows before Byte Substitution would lead to the bytes being shifted before substitution, causing a different pattern of byte interactions. This could result in a less effective diffusion of data and a less secure encryption algorithm.

The original order of operations in AES (Byte Substitution followed by Shift Rows) is designed to provide strong security properties and has been extensively analyzed and tested. Changing the order of these layers would likely weaken the encryption algorithm and may introduce vulnerabilities. It's crucial to adhere to the standard AES specification for secure encryption.

5. In Counter mode (CTR) mode of operation:

(a) Describe the encryption and decryption process. (10 points)

Ans:

Encryption Process:

1. The plaintext data is divided into blocks, typically the same block size as the encryption algorithm (e.g., 128 bits or 16 bytes for AES).

2. A unique counter value (often called the "nonce") and an initialization vector (IV) are combined to create the input to the block cipher encryption function (e.g., AES encryption). The counter value is incremented for each subsequent block.
3. The encryption function is applied to this input, producing a keystream for that block.
4. The keystream is then XORed (bitwise exclusive-OR) with the plaintext block to produce the corresponding ciphertext block.
5. Steps 2-4 are repeated for each block of plaintext to encrypt the entire message.

Decryption process:

1. The same unique counter value and IV, as used during encryption, are combined to create the input to the block cipher encryption function.
2. The encryption function is applied to this input, producing the same keystream used during encryption.
3. The keystream is XORed with the ciphertext block to recover the original plaintext block.
4. Steps 1-3 are repeated for each block of ciphertext to decrypt the entire message.

CTR mode turns a block cipher into a stream cipher, where each block of plaintext is encrypted independently.

(b) Suppose that there is a transmission bit error in c_i . Show how many plaintext blocks are affected due to the transmission bit error in c_i . (10 points)

Ans:

If there is a transmission bit error in one of the ciphertext blocks, it will affect exactly one corresponding block of plaintext. This is because the error in the ciphertext block will cause an error in the XOR operation when decrypting, which will directly affect the corresponding plaintext block.

In CTR mode, there is a one-to-one relationship between the ciphertext blocks and the plaintext blocks. Therefore, a transmission bit error in a single ciphertext block will lead to an error in only one plaintext block during decryption, with no propagation of errors to adjacent blocks.

6. We consider a banking system, where message m of the form fromAccount, toAccount, and amount are sent within the bank

network, with the meaning that amount dollars should be transferred from fromAccount, to toAccount. Each message consists of three blocks, with each block holding one of the three parameters. Messages are encrypted with AES in Counter Mode as follows:

$$K_j = E(k, T_j), C_j = M_j \oplus K_j$$

Each of the three parts of a message is sixteen characters, i.e., one block, so messages consist of three blocks.

(a) The adversary has an account in the bank and can intercept and changes messages. Imagine now that he know the toAccount for a particular message $m = C_1C_2C_3$. Explain how he can modify the message so that the amount is transferred to his own account. (10 points)

Ans:

1. Intercept the original message m, which consists of three blocks: fromAccount, toAccount, and amount.
2. The adversary knows the toAccount and wishes to change it to their own account. Let's denote the adversary's account as advAccount.
3. Compute a new message, let's call it m', where:
   - fromAccount remains the same.
   - toAccount is set to the adversary's account, i.e., advAccount.
   - amount remains the same.
4. Encrypt each block of the new message m' using the same counter mode encryption process:
   - Calculate $K_1 = E(k, T_1)$, where T_1 is the counter for the first block.
   - Calculate $K_2 = E(k, T_2)$, where T_2 is the counter for the second block.
   - Calculate $K_3 = E(k, T_3)$, where T_3 is the counter for the third block.
5. XOR each corresponding block of the new message m' with the generated keystreams to obtain the modified ciphertext:
   - Calculate $C_1 = M_1 \oplus K_1$ for the first block.
   - Calculate $C_2 = M_2 \oplus K_2$ for the second block.
   - Calculate $C_3 = M_3 \oplus K_3$ for the third block.
6. Send the modified ciphertext $C_1C_2C_3$ to the bank, and the bank will decrypt it as if it's a legitimate transfer to the adversary's

account.

(b) Explain how the use of MAC would prevent this attack? (10 points)

Ans:

The use of a Message Authentication Code (MAC) can prevent this attack by ensuring the integrity and authenticity of the message. Here's how it works:

Along with encrypting the message using AES in Counter Mode, a MAC is calculated for the message. The MAC is typically computed over the entire message, including all three blocks.

The MAC is computed using a secret key known only to the bank and appended to the message. When the bank receives the message, it first decrypts it using the counter mode encryption process as described in part (a).

After decryption, the bank verifies the MAC using the same secret key. If the MAC doesn't match the computed MAC for the decrypted message, the bank will reject the message as it has been tampered with. In this way, the use of a MAC ensures that the message's integrity has not been compromised, and it came from a legitimate source. It prevents adversaries from modifying the message without the bank detecting the alteration.

7. When one signs an electronic document using digital signature, one often performs the signature operation on a message digest produced by passing the document through a cryptographically strong hash function $h = H(m)$. Why it is important that it is difficult to find two documents with the same message digest? (10 points)

Ans:

It is important that it is difficult to find two documents with the same message digest because this property is fundamental for the security and reliability of digital signatures and various other cryptographic applications.

Uniqueness and Data Integrity:

When you sign a document using a digital signature, you want to ensure the integrity of the document, meaning that it has not been tampered with or modified in transit. If two different documents had the same message digest, an attacker could forge a different document with the same digest value and claim it was signed by you. This would compromise data integrity.

Collision Resistance:

Cryptographic hash functions are designed to be collision-resistant, meaning it should be computationally infeasible to find two different inputs (documents in this case) that produce the same hash value. If collisions were easily found, it would be possible to create fraudulent documents with the same digest as legitimate ones, undermining the trust in digital signatures.

Non-repudiation:

Digital signatures are often used to establish non-repudiation, meaning that the signer cannot deny their involvement. If collisions were common, a signer could claim that a document with a particular hash value wasn't theirs because there could be another document with the same hash. This would weaken the legal and evidentiary value of digital signatures.

Security and Trust:

The uniqueness of message digests is a core security property of cryptographic hash functions. It is essential for the security of various cryptographic protocols and applications, not just digital signatures. When cryptographic hash functions have a well-defined output size, such as 256 bits, the number of possible hash values is vast, making it highly improbable to find two different inputs with the same hash.

In summary, ensuring the uniqueness of message digests is vital for preserving data integrity, security, non-repudiation, and trust in digital signatures and other cryptographic applications. A well-designed cryptographic hash function minimizes the risk of collision, making it computationally infeasible to find two different documents that produce the same hash value.

8. In RSA algorithm, is it possible for more than one d to work with a given e, p, and q?

Hint (Is it possible to have d and u $s.t. ed = 1 \bmod \phi(n)$ and $eu = 1\, mod\, \phi(n)$?) (10 points)

Ans:

To be a valid RSA decryption exponent, d must satisfy the equation:

$$d \equiv e^{-1} mod\ (p-1)(q-1)$$

In other words, d is the modular multiplicative inverse of e within

the context of Euler's totient function $(\boldsymbol{p}-\mathbf{1})(\boldsymbol{q}-\mathbf{1})$.

For $\boldsymbol{e}$ and $\boldsymbol{d}$ to be inverses modulo$(\boldsymbol{p}-\mathbf{1})(\boldsymbol{q}-\mathbf{1})$, it is essential that $\boldsymbol{e}$ is relatively prime to$(\boldsymbol{p}-\mathbf{1})(\boldsymbol{q}-\mathbf{1})$. In practical terms, this means that $\boldsymbol{e}$ and $(\boldsymbol{p}-\mathbf{1})(\boldsymbol{q}-\mathbf{1})$ share no common factors other than 1.

Since $\boldsymbol{e}$ is relatively prime to $(\boldsymbol{p}-\mathbf{1})(\boldsymbol{q}-\mathbf{1})$, every element in the set $\boldsymbol{Z}(\boldsymbol{p}-\mathbf{1})(\boldsymbol{q}-\mathbf{1})$ multiplied by $\boldsymbol{e}$ results in a unique element in the same set. In other words, multiplying elements of $\boldsymbol{Z}(\boldsymbol{p}-\mathbf{1})(\boldsymbol{q}-\mathbf{1})$ by $\boldsymbol{e}$ generates a permutation of the set $\boldsymbol{Z}(\boldsymbol{p}-\mathbf{1})(\boldsymbol{q}-\mathbf{1})$.

Therefore, there is only one unique element in $\boldsymbol{Z}(\boldsymbol{p}-\mathbf{1})(\boldsymbol{q}-\mathbf{1})$ that yields 1 when multiplied by $\boldsymbol{e}$. This unique element corresponds to the modular multiplicative inverse $\boldsymbol{d}$ modulo $(\boldsymbol{p}-\mathbf{1})(\boldsymbol{q}-\mathbf{1})$.
In summary, the uniqueness of $\boldsymbol{d}$ modulo $(\boldsymbol{p}-\mathbf{1})(\boldsymbol{q}-\mathbf{1})$ is a consequence of the relatively prime property between $\boldsymbol{e}$ and $(\boldsymbol{p}-\mathbf{1})(\boldsymbol{q}-\mathbf{1})$, ensuring that $\boldsymbol{d}$ is a single, unique modular multiplicative inverse within the context of RSA encryption.