Cryptography Final Exam

Anurag Gulavane

Q1)
(A) What is Kerckhoffs' Principle? Why is it important?

Kerckhoffs' Principle, also known as the principle of the security of the key, is a fundamental principle in cryptography which states that a cryptographic system should remain secure even if everything about the system, except for the key, is known by the attacker. In other words, the security of a cryptographic system should rely solely on the secrecy of the key, not on the obscurity of the algorithm or any other component of the system. This principle is important because it ensures that the security of a cryptographic system can be objectively evaluated and tested, and that the system can be designed and analyzed based on well-established principles and best practices.

(B) Define perfect secrecy and computational security. Explain why popular practical encryption schemes achieve computational security but not perfect secrecy.

Perfect secrecy is a property of an encryption scheme that guarantees that the ciphertext provides no information about the plaintext, even if the attacker has unlimited computational resources. This means that the ciphertext is statistically independent of the plaintext, and the attacker cannot gain any knowledge about the plaintext even if they have access to multiple ciphertexts encrypted using the same key. Computational security, on the other hand, is a weaker property that guarantees that the attacker cannot break the encryption scheme within a reasonable amount of time, given the best-known algorithms and computational resources available. Popular practical encryption schemes, such as AES and RSA, achieve computational security but not perfect secrecy because they are designed based on computational assumptions that are believed to be hard to break, but not impossible.

(C) Explain the main differences between public key and secret key schemes.

The main difference between public (asymmetric) key and secret (symmetric) key schemes is that in public key schemes, there are two separate keys: a public key, which can be openly distributed to anyone, and a private key, which is kept secret by the owner. The encryption and decryption processes use different keys, and the public key is used to encrypt messages, while the private key is used to decrypt them. In secret key schemes, there is only one key, which is shared by the sender and the receiver, and both encryption and decryption processes use the same key.

(D) Explain the similarities and differences between public key encryption schemes and digital signature schemes.

Public key encryption schemes and digital signature schemes are both based on public key cryptography, but they serve different purposes. Public key encryption schemes are used to securely transmit messages between two parties without a prior shared secret, while digital signature schemes are used to provide authenticity, integrity, and non-repudiation of digital messages. The main similarity between the two is that they both rely on the use of asymmetric keys, with one key used for encryption (in the case of public key encryption) or signing (in the case of digital signatures), and the other key used for decryption (in the case of public key encryption) or verification (in the case of digital signatures). The main difference is that in public key encryption, the sender encrypts the message with the receiver's public key, while in digital signatures, the sender signs the message with their own private key, which can be verified by anyone who has access to the sender's public key.

Q2)
(a) Let's analyze each encryption scheme and assess the efficiency of attacking them:
In this scheme, two DES instances are used in sequence, which means the output of the first DES becomes the input of the second DES. If an adversary knows the message-ciphertext pairs (m1, C1) and (m2, c2), they can try all possible keys in a brute-force attack on the first DES instance, and for each key, decrypt C1 using the first DES. Then, they can try all possible keys in a brute-force attack on the second DES instance using the decrypted output of C1 as the input. Once they find a key that successfully decrypts C1 to m1, they can use that key to decrypt c2 using the second DES. The complexity of this attack is the same as a single DES brute-force attack, which is 2^56.
In this scheme, three DES instances are used in sequence. Similar to the previous scheme, the adversary can try all possible keys in a brute-force attack on the first DES instance. Once they find a key that successfully decrypts C1 to m1, they can use that key to decrypt c2 using the second DES. Finally, they can try all possible keys in a brute-force attack on the third DES instance using the decrypted output of C2 as the input. The complexity of this attack is also 2^56, the same as a single DES brute-force attack.
In this scheme, only one DES instance is used. Since the adversary knows the message - ciphertext pairs (m1, C1) and (m2, c2), they can directly try all possible keys in a brute-force attack on the single DES instance. The complexity of this attack is also 2^56.

(b) The effective key length of a scheme is the number of bits an attacker has to guess to break the scheme. In all three schemes, the effective key length is 56 bits, which is the length of the DES key. This is because the attacker can perform a brute-force attack on the DES instances, which requires trying all possible keys. Since DES has a key length of 64 bits, but 8 of those bits are used for parity, the effective key length is reduced to 56 bits.
Explanation:
The effective key length of all three schemes is 56 bits. This means that an attacker would need to guess all 56 bits of the key to successfully break the encryption scheme. DES itself has a key length

of 64 bits, but 8 of those bits are used for parity and do not contribute to the security of the encryption. Therefore, the effective key length, which represents the actual number of bits an attacker needs to guess, is reduced to 56 bits.

(c) None of the schemes show a significantly improved security compared to a single encryption. In all three schemes, the effective key length is still 56 bits, which is the same as a single DES encryption. The additional DES instances in the schemes do not increase the effective key length or provide significant security enhancements against brute-force attacks

Q3)
(a) Protocol (a) provides the security service of integrity. This is because the hash function H(m) produces a fixed-size output (hash) that is derived from the message m. If the message m is altered in any way during transmission, the hash value will also be changed. Therefore, the recipient can compare the received hash value with a computed hash value of the received message, and if they match, it can be concluded that the message has not been altered.

However, protocol (a) does not provide any other security services such as secrecy, authenticity, or non-repudiation.
Explanation:

(b) Protocol (b) provides the security services of integrity and authenticity. The message authentication code (MAC) provides integrity by generating a tag that is appended to the message. The recipient can compute the MAC of the received message and compare it with the received MAC. If they match, it can be concluded that the message has not been altered.

The MAC also provides authenticity by using a secret key shared between the sender and the receiver. This ensures that the sender of the message is authentic since only the sender and the receiver know the shared secret key.

However, protocol (b) does not provide secrecy or non-repudiation. authenticity, and non-repudiation.

(c) Protocol (c) provides the security service of secrecy. The message m is encrypted using a stream cipher (Encs), and the hash function H(m) is used to encrypt the key for the stream cipher. This ensures that an adversary cannot read the message or derive the key for the stream cipher without knowing the secret key used by the sender and the receiver.

However, protocol (c) does not provide any other security services such as integrity, authenticity, or non-repudiation.

Explanation:
(d) Protocol (d) provides the security service of secrecy and authenticity. The message m is encrypted using a block cipher (Enca), and the hash function H(m) is used to encrypt the key for the block cipher. This ensures that an adversary cannot read the message or derive the key for the block cipher without knowing the secret key used by the sender and the receiver.

The use of the hash function H(m) also provides authenticity by ensuring that the encrypted key for the block cipher matches the message m. This means that the recipient can decrypt the message and compute the hash value of the decrypted message. If the computed hash value matches the received hash value, it can be concluded that the message is authentic.

However, protocol (d) does not provide integrity or non-repudiation.

Q4)
The core inquiry in the Triple RSA question revolves around whether employing RSA encryption iteratively with distinct keys but a shared modulus contributes to enhanced security.

To comprehensively address this query, a grasp of RSA encryption mechanisms is imperative. RSA encryption for a message m with a public key (e, N) is executed as ($c = m^e \bmod N$), where c signifies the ciphertext, e denotes the public exponent, and N stands for the modulus (a product of two primes).

Triple RSA involves the following steps:
1. Encrypt m with (e_1, N) to yield $c_1 = m^{e1} \bmod N$.
2. Encrypt c_1 with (e_2, N) to obtain $c_2 = c_1$ ^e2mod N = $(m^{e1})^{e2} \bmod N$.
3. Encrypt c_2 with (e_3, N) resulting in $c_3 = c_2$ ^e3 mod N = $(m^{e1})^{e2})^{e3}$ mod N .

This can be expressed as $c_3 = m^{(e1 \cdot e2 \cdot e3)} \bmod N$.

The fundamental facet of RSA security lies in the challenge of factoring the modulus N into its prime components. The security landscape does not necessarily experience an augmentation with multiple encryptions utilizing the same modulus since the inherent complexity of breaking RSA—factoring N —remains unaltered. However, if the intermediate ciphertexts are undisclosed, and the exponents (e_1, e_2, e_3) are chosen such that their greatest common divisor (gcd) with $\phi(N)$ (where ϕ is Euler's totient function) is 1, it theoretically increases the workload for potential attackers due to the larger exponent $e_1 * e_2 * e_3$.

To substantiate this mathematically, consideration must be given to the RSA assumption and its connection to the factoring problem:

- *RSA Assumption*: Given (e, N) and c, deriving m such that $c = m^e \bmod N$ is deemed challenging.
- *Factoring Problem*: Determining the prime factors p and q of N is considered a formidable task.

While the act of encrypting multiple times with RSA technically results in an expanded exponent, it does not alter the foundational assumption that factoring N poses a formidable challenge. If an adversary can successfully factor N , they can breach the encryption irrespective of the number of applications.

In practical terms, recurrent encryption with distinct public keys does not markedly elevate the security of RSA due to:
- The uniformity of the modulus N, implying that its factorization once compromises the entire layered encryption.
- The amalgamated exponent $e_1 * e_2 * e_3$ may introduce mathematical characteristics susceptible to exploitation.

In conclusion, while Triple RSA might escalate the computational demands for decryption without the private key owing to the enlarged exponent, it does not confer a substantial enhancement in cryptographic resilience akin to Triple DES. The crux of RSA's resilience lies in the complexity of factoring N, not in overcoming exponentiation challenges.