RUTGERS UNIVERSITY

Department of Computer Science

Computer Security

Exam 1

October 7, 2024

Solutions & Discussion

Prepared with tender loving care by Paul Krzyzanowski

100 POINTS – 25 QUESTIONS – 4 POINTS EACH – For each statement, select the *most* appropriate answer.

1. An attack that modifies data without permission violates which part of the *CIA Triad*?
   (a) Authentication.
   (b) Confidentiality.
   (c) Availability.
   (d) Integrity.

The CIA Triad stands for Confidentiality, Integrity, and Availability, which are the core principles of information security.

Confidentiality: Ensures that data is kept secret and only accessible to authorized users.

Integrity: Ensures that the data is accurate, unaltered, and trustworthy. Modifying data without permission violates integrity because it changes the correctness or reliability of the information.

Availability: Ensures that authorized users can access data and resources when needed.

Authentication: While authentication verifies identity, it is not part of the CIA Triad but often works alongside it.

Modifying data without permission compromises its integrity because it affects the accuracy and trustworthiness of the information.

2. What is a *vulnerability* in computer security?
   (a) The intentional release of malicious software.
   (b) A protective measure to secure a system.
   (c) A bug or weakness that can be used by attackers.
   (d) An attack on a computer system that bypasses protection mechanisms.

In computer security, a vulnerability is a flaw or weakness in a system, software, or hardware that could be exploited by an attacker to perform unauthorized actions.

Option (a) describes the release of malicious software, which is an attack, not a vulnerability.
Option (b) refers to security measures, which are meant to protect systems from vulnerabilities.
Option (d) describes an exploit, which is an attack that takes advantage of a vulnerability.

3. Why is *threat modeling* important in designing secure systems?
   (a) It helps identify areas where software bugs may exist.
   (b) It enables the monitoring of network traffic to determine suspicious activities.
   (c) It can help prioritize risks and identify areas to focus on securing.
   (d) It allows security engineers to assess the end-to-end security of a system.

Threat modeling is a proactive approach used during the design and development of secure systems. It involves identifying potential threats, vulnerabilities, and attack vectors that could compromise a system. By understanding potential threats, security engineers can address weaknesses early, prioritize the most severe risks, and implement targeted security measures

Option (a) relates to finding software bugs, but threat modeling focuses on broader risks, not just coding issues.
Option (b) involves monitoring, which happens after the system is deployed, while threat modeling is a design-time activity.
Option (d) addresses system security assessment, but threat modeling specifically helps focus on the most critical risks and prioritize resources effectively.

4. A *trusted computing base* (*TCB*) is:
   (a) The set of all hardware and software components that work together to keep a system secure.
   (b) A separate processor on a system used for security operations, such as storing biometric data.
   (c) An isolated, highly secure computing environment used for security-critical tasks.
   (d) A system specifically engineered to withstand attacks.

The Trusted Computing Base (TCB) refers to the collection of hardware, software, and firmware components within a computer system that are critical to its security. These components enforce the system's security policy and are trusted to operate correctly. If f any part of the TCB is compromised, the system's overall security is at risk. Therefore, the TCB must be kept small and tightly controlled.

Option (b) refers to specific hardware components like a Trusted Platform Module (TPM), but that's not the entire TCB.
Option (c) describes an isolated secure environment, but the TCB is broader and includes more than just an isolated area.
Option (d) refers to a secure system but doesn't encompass the specific set of components that make up the TCB.

5. What does *Kerckhoffs's Principle* state about cryptographic systems?
(a) The algorithm should remain secret for maximum security.
(b) Only the key must remain secret, but the algorithm can be public.
(c) The encryption method should be reversible.
(d) Encryption keys should be updated periodically.

Kerckhoffs's Principle is a fundamental guideline in cryptography, stating that a cryptographic system should remain secure even if everything about the system, except the key, is known to the public. This means that the security of the system should not depend on keeping the algorithm secret but solely on keeping the key confidential. This principle supports the development of widely tested, robust cryptographic algorithms, as their security relies only on the secrecy of the key, not the algorithm.

Option (a) is the opposite of Kerckhoffs's Principle, as it suggests the algorithm must be secret.
Option (c) refers to reversible encryption, which is not relevant to this principle.
Option (d) deals with key management practices but is not part of Kerckhoffs's Principle.

6. Polyalphabetic ciphers are an improvement over monoalphabetic ciphers because:
(a) The same plaintext character may be encoded to different ciphertext characters based on its position.
(b) They are not restricted to supporting a single language.
(c) They are faster for encryption and decryption while providing greater security.
(d) They require shorter keys.

Polyalphabetic ciphers improve upon monoalphabetic ciphers by using multiple alphabets for encryption, meaning the same plaintext character can be encrypted to different ciphertext characters depending on its position in the text. By varying the substitution alphabet, polyalphabetic ciphers provide stronger encryption than monoalphabetic ciphers, which use the same substitution throughout the entire message. This variation makes frequency analysis attacks more difficult because the ciphertext no longer maintains a one-to-one relationship with the plaintext characters.

Option (b) refers to language support, which is unrelated to the type of cipher.
Option (c) suggests performance advantages, which are not the reason for improved security.
Option (d) refers to key length, but polyalphabetic ciphers often require more complex key structures rather than shorter keys.

7. The property of *confusion* in a cipher refers to:
(a) Making the relationship between the key and the ciphertext as complex as possible.
(b) An attacker not being able to identify what encryption algorithm was used.
(c) Mixing false data into the message so the attacker cannot tell which parts are valid.
(d) Applying a transposition to the generated ciphertext to break up digraphs and trigraphs.

Confusion is a property of a cipher that aims to make the relationship between the encryption key and the resulting ciphertext as complex as possible. This ensures that changing even a small part of the key results in a completely different ciphertext, making it difficult for an attacker to deduce the key from the ciphertext. The goal of confusion is to prevent attackers from understanding how changes in the key affect the ciphertext, thereby making it harder to reverse-engineer the encryption.

Option (b) refers to the identification of the encryption algorithm, which is unrelated to confusion.
Option (c) involves adding false data, which is more related to obfuscation, not confusion.
Option (d) describes transposition techniques, which are related to another concept called diffusion, not confusion.

8. How does an SP-Network achieve both confusion and diffusion in encryption?
   (a) By using S-boxes for substitution and key expansion for diffusion.
   (b) Via private keys to create confusion and public keys for diffusion.
   (c) By applying multiple rounds of substitution and permutation.
   (d) By applying a cipher mode such as counter (CTR) mode.

> An SP-Network (Substitution-Permutation Network) achieves confusion and diffusion through the combined use of substitution (S-boxes) and permutation (P-boxes) applied in multiple rounds.
>
> Confusion is introduced by the substitution step (S-boxes), where the relationship between the plaintext and the ciphertext is obscured by substituting parts of the data with others based on a complex rule set.
>
> Diffusion is achieved through the permutation step (P-boxes), which spreads out the influence of individual bits of the plaintext across the ciphertext, ensuring that changes in the input affect many parts of the output.
>
> SP-Networks, such as those used in AES, combine substitution and permutation over several rounds to provide strong encryption through confusion and diffusion.
>
> Option (a) refers to S-boxes for substitution, but key expansion is more of a key management process, not diffusion.
>
> Option (b) describes asymmetric encryption, not relevant to SP-Networks.
>
> Option (d) refers to a cipher mode, which is used in symmetric encryption but is not part of the core structure of an SP-Network.

9. Why is AES considered secure against brute-force attacks?
   (a) It uses a secure key exchange algorithm.
   (b) It uses large key sizes, making brute-force infeasible.
   (c) It operates with unbreakable keys.
   (d) It is a symmetric algorithm.

> Tutorial Explanation: AES (Advanced Encryption Standard) is considered secure against brute-force attacks primarily because of its large key sizes—128-bit, 192-bit, or 256-bit. These key lengths make it computationally infeasible for attackers to try all possible key combinations within a reasonable amount of time. The huge number of possible key combinations with large key sizes in AES makes brute-forcing impractical with current technology. Even with a huge network of powerful computers, it would take an astronomically long time to try all potential keys.
>
> Option (a) refers to key exchange, which is not the focus of AES, as it's a symmetric algorithm, meaning both parties use the same key.
> Option (c) implies that keys are unbreakable, which isn't accurate; security relies on large key space rather than unbreakability.
> Option (d) describes the type of encryption (symmetric), but this alone does not explain AES's security against brute-force attacks.

10. How does CBC mode ensure that identical plaintext blocks do not produce identical ciphertext blocks?
   (a) By XORing the previous ciphertext block with the current plaintext block.
   (b) By changing the encryption key for each block.
   (c) By encrypting a different counter value for each block of plaintext.
   (d) By rearranging the plaintext blocks based on an initialization vector (IV) before encryption.

> In Cipher Block Chaining (CBC) mode, each plaintext block is XORed with the previous ciphertext block before being encrypted. This ensures that even if identical plaintext blocks appear in the message, they will produce different ciphertext blocks due to the influence of the previous block's ciphertext. The use of an initialization vector (IV) in CBC mode for the first block ensures that identical plaintext messages encrypted with the same key will still produce different ciphertexts by introducing randomness at the start
>
> Option (b) suggests changing the key for each block, but CBC mode keeps the key constant.
> Option (c) describes Counter (CTR) mode, not CBC mode.
> Option (d) incorrectly describes block rearrangement, whereas CBC uses XORing with previous ciphertext blocks.

11. A *trapdoor* function differs from a one-way function in that:
    (a) It is created through the application of symmetric key cryptography.
    (b) It is easy to compute both the function and its inverse.
    (c) Its output is not always a fixed size.
    [✓ (d) Its inverse can be computed only with knowledge of a secret parameter.]

A trapdoor function is a special type of one-way function. While both are easy to compute in one direction, a trapdoor function has the additional property that its inverse can only be computed if a secret parameter (the "trapdoor") is known. Trapdoor functions are a key concept in public-key cryptography, where a public key allows for easy encryption (one-way function), but only someone with the private key (trapdoor) can decrypt the message efficiently.

Option (a) refers to symmetric key cryptography, which is unrelated to trapdoor functions.
Option (b) is incorrect because, without the secret, the inverse is hard to compute.
Option (c) refers to output size, which is not relevant to the definition of trapdoor functions.

12. For Alice to send data securely to Bob, Alice will encrypt the data with:
    (a) Alice's private key.
    (b) Alice's public key.
    (c) Bob's private key.
    [✓ (d) Bob's public key.]

In public-key cryptography, when Alice wants to send data securely to Bob, she encrypts the data using Bob's public key. Only Bob, who has the corresponding private key, can decrypt the data. This ensures confidentiality because even if someone intercepts the encrypted message, they cannot decrypt it without Bob's private key. The use of Bob's public key ensures that only Bob can decrypt the message, as he alone has the corresponding private key.

13. How is *forward secrecy* typically achieved in secure communication protocols?
    [✓ (a) By using ephemeral session keys, often generated through protocols like Diffie-Hellman.]
    (b) By encrypting the data with a symmetric key and using an X.509 certificate to encrypt the key.
    (c) By using pre-shared keys and a symmetric algorithm to encrypt all communication data.
    (d) By requiring data to be encrypted only in one direction during the communication session.

With cryptographic hash functions, a small change in the input (even a single bit) results in a significant and unpredictable change in the output hash. On average, changing one byte of the input should change about half of the bits in the output hash.

Option (a) and Option (b) suggest very few bits changing, which doesn't align with the diffusion expected in hash functions.
Option (d) suggests all bits will change, which is also not expected.

Since the hash output is 256 bits, about half of them—approximately 128 bits—should change if you modify even one byte of the input message. This property helps ensure that the hash output does not reveal any patterns about the input.

14. Approximately how many bits do you expect will change in the 256-bit hash of a 256-byte message if you modify the last byte of the message?
    (a) 1
    (b) 8
    [✓ (c) 128]
    (d) 256

With cryptographic hash functions, a small change in the input (even a single bit) results in a significant and unpredictable change in the output hash. On average, changing one byte of the input should change about half of the bits in the output hash. Since the hash output is 256 bits, about half of them—approximately 128 bits—should change if you modify even one byte of the input message. This property helps ensure that the hash output does not reveal any patterns about the input.

Option (a) and Option (b) suggest very few bits changing, which doesn't align with the diffusion expected in hash functions.
Option (d) suggests all bits will change, which is also a completely non-random result.

15. How does a MAC (Message Authentication Code) differ from a hash?
    (a) A MAC uses a secret key in combination with the message.
    (b) A MAC generates a larger output than a hash function.
    (c) A hash function uses a MAC to provide integrity.
    (d) A MAC is a hash of an encrypted message.

> A Message Authentication Code (MAC) is like a hash function but with a critical difference: it uses a secret key along with the message to produce the MAC output. This makes it unique to the key-message pair, providing both integrity (ensuring the message hasn't been tampered with) and authenticity (confirming it came from a trusted source who knows the key). In contrast, a hash function takes only the message as input and is used solely for data integrity, not authenticity.

16. Why are hash functions important in digital signatures?
    (a) They provide message confidentiality.
    (b) They convert large messages into fixed-size digests before signing.
    (c) They generate the encryption key used for signing.
    (d) They enable authentication without using keys.

> In digital signatures, a hash function is used to create a fixed-size digest of the message, regardless of the message's original size. This digest is then signed instead of the full message, making the signing process more efficient and manageable. The hash function ensures that any change to the original message would result in a different hash, allowing verification of the message's integrity when the signature is checked.

17. What is the primary purpose of an X.509 certificate?
    (a) To bind a public key to an identity and provide trust in that identity.
    (b) To encrypt messages between two parties.
    (c) To generate session keys for symmetric encryption
    (d) To sign digital documents.

> An X.509 certificate binds a public key to a specific identity, such as a person, organization, or device. This binding is authenticated by a trusted Certificate Authority (CA), which digitally signs the certificate. By verifying the certificate, others can trust the identity associated with the public key, enabling secure communication and authentication in digital environments.

18. What is one way that *Kerberos* improves the security of the Needham-Schroeder protocol?
    (a) Kerberos eliminates the need for a trusted third party.
    (b) Kerberos uses timestamps to protect against replay attacks.
    (c) Kerberos uses public key cryptography.
    (d) Kerberos does not need session keys for secure communication.

> **Kerberos** improves on the Needham-Schroeder protocol by incorporating timestamps into its authentication process. This addition helps protect against replay attacks, where an attacker could intercept and reuse valid authentication messages. By using timestamps, Kerberos ensures that authentication messages have a limited validity period, making it much harder for an attacker to use intercepted messages at a later time.

19. Why is a *salt* used in combination with password hashes?
    (a) To reduce the size of the password hash.
    (b) To protect against attacks via precomputed hashes.
    (c) To serve as a unique encryption key for the password.
    (d) To add a message authentication code to a stored password.

> A salt is a random string added to a password before hashing to ensure that each user's password hash is unique, even if they have the same password. This makes precomputed attacks ineffective, as an attacker would need to generate a table for each unique salt value or create a table with an exhaustive list of all possible passwords that incorporate every salt string. Either approach would be not feasible.

20. How does a *passkey* system authenticate a user?
    (a) By generating a random password for each login attempt
    (b) By using public key cryptography without the need for a user's password.
    (c) By encrypting a traditional password with a unique key for that service.
    (d) By storing a hash of the password on the server.

A passkey system leverages public key cryptography to authenticate users without requiring them to enter a password. Instead, a passkey (a public-private key pair) is generated for the user, where the private key is securely stored on the user's device, and the public key is registered with the service. During authentication, the user's device uses the private key to respond to a challenge from the service, proving their identity without revealing any secret that could be stolen or intercepted, enhancing security and user convenience.

21. In *HOTP*, (Hash-based One-Time Passwords), what happens each time a new password is generated?
    (a) The shared secret key is updated.
    (b) A counter is incremented.
    (c) The server accepts it as valid for a short time, typically 30 seconds.
    (d) The server sends a different challenge value the next time the user logs in.

In HOTP, each time a new password is generated, a counter value is incremented. This counter, combined with a shared secret key, is used to generate a unique one-time password (OTP) through a hashing process. Because the one-time password depends on this counter, it changes predictably each time it's requested, ensuring the password is unique for each authentication attempt without relying on a specific time interval, unlike time-based one-time passwords (TOTP).

22. Why is *MFA* (Multi-Factor Authentication) considered more secure than using just a password?
    (a) MFA uses stronger encryption algorithms.
    (b) MFA reduces the complexity of passwords, making them easier to remember.
    (c) MFA uses time-based passwords that change every 30 seconds.
    (d) MFA adds additional layers of authentication.

Multi-Factor Authentication (MFA) enhances security by requiring users to provide more than one form of verification, such as something they know (a password), something they have (a device or token), or something they are (biometrics). This layered approach makes it much harder for attackers to gain unauthorized access, as compromising one factor (like a password) alone is insufficient to bypass the security.

23. Why is *number matching* (NMA) considered more secure than a standard push notification?
    (a) It uses symmetric encryption for all communication, avoiding eavesdropping attacks.
    (b) It allows the user to avoid two-factor authentication.
    (c) It doesn't require an internet connection.
    (d) It avoids unintentional approvals by requiring the user to actively transcribe data.

Number Matching Authentication (NMA) improves security over standard push notifications by requiring the user to actively confirm a specific number shown on their device by entering it manually on the authentication prompt. This step helps prevent accidental approvals, such as mistakenly approving a push notification without careful review.

24. How does a *ROC curve* assist in selecting the threshold for a biometric system?
    (a) It helps determine the trade-off between false acceptances and false rejections.
    (b) It optimizes the system for faster processing.
    (c) It helps prioritize authentication performance over user enrollment.
    (d) It makes it possible to normalize biometric data for efficient matching.

A Receiver Operating Characteristic (ROC) curve is used in biometric systems to visualize and select the optimal threshold by illustrating the trade-off between the false acceptance rate (FAR) and the false rejection rate (FRR). By analyzing the ROC curve, system administrators can choose a threshold that balances security (minimizing false acceptances) and user convenience (minimizing false rejections), depending on the security requirements of the application.

The end.