Armaan Shamsaasef

Hasing and Digital Signatures

Hashing and Digital Signatures - SY0-701 1.4

Part 1: Hashing

Definition: A one-way mathematical process that takes an input (of any size) and produces a fixed-size, unique string of characters called a hash value, checksum, or digest.

Key Properties of a Cryptographic Hash:

1. Deterministic: The same input will *always* produce the exact same hash output.
2. Fixed Length: The hash is always the same size, regardless of the input size (e.g., a single word and an entire book both produce a 64-character SHA-256 hash).
3. Irreversible (One-Way): It is computationally infeasible to re-create the original input data from the hash. You cannot "decrypt" a hash.
4. Avalanche Effect: A tiny change in the input (even a single character) produces a dramatically different hash output.
5. Collision Resistant: It is extremely difficult to find two different inputs that produce the same hash output.

Common Hashing Algorithms:

- MD5 (Message Digest 5): Produces a 128-bit hash. Now considered cryptographically broken due to collision vulnerabilities. Should not be used for security.
- SHA-1 (Secure Hash Algorithm 1): Produces a 160-bit hash. Also now considered insecure due to collision attacks.
- SHA-2: A family of stronger hashes, including SHA-256 and SHA-512. This is the current standard for security.
- SHA-3: The latest generation of secure hashing algorithms, designed differently from SHA-2.

Primary Use Cases for Hashing:

- Verifying Integrity: Ensuring a file has not been tampered with.
    - Example: A software developer provides a file for download and posts its official SHA-256 hash. After you download the file, you hash it yourself. If your calculated hash matches the developer's, the file is intact and authentic.
- Password Storage: Websites store a hash of your password, not the password itself. During login, they hash your input and compare it to the stored hash.
- Digital Signatures & Certificates: The foundation for proving authenticity and integrity.

Part 2: Digital Signatures

Purpose: To provide Authentication, Non-Repudiation, and Integrity for a digital message or document.

How it Works (Combining Hashing & Asymmetric Encryption):

1. Signing (Sender's Side):
    - You create a message.
    - You generate a hash of the message. This is the unique fingerprint.
    - You encrypt that hash with your Private Key. This encrypted hash *is* the digital signature.
    - You send the original message and the digital signature to the recipient.
2. Verification (Recipient's Side):
    - The recipient receives the message and the digital signature.
    - They decrypt the digital signature using the sender's Public Key. This gives them the original hash that the sender created.
    - They independently generate a new hash from the received message.
    - They compare the two hashes:
        - If they match, it proves three things:
            1. Integrity: The message was not altered.
            2. Authentication: It truly came from the claimed sender (only they have the private key that created the signature).
            3. Non-Repudiation: The sender cannot deny having sent it.

Summary

- Hashing: A one-way function to create a unique fingerprint for data. Used for verifying integrity.
- Digital Signatures: Use a combination of hashing and asymmetric encryption (with a private/public key pair) to provide integrity, authentication, and non-repudiation.