III. Hash Functions and Message Authentication

1. What is Hashing and its objective?

- Hashing is a cryptographic process that converts input data into a fixed-lenght string of characters, known as a hash value or digest.
- Hash Functions are one-way, meaning it is computationally infeasible to reverse the process and obtain the original input.

- Objective of Hashing:
  - → Data Integrity: Ensures data has not been altered during transmission or storage.
  - → Authentication: Verifies the identity and some info. of (A party) other party.
  - → Digital Signitures: Used in secure online transactions
  - → Efficient Data Retreival: Used in hash tables for fast lookups in databases.

2. Hash Functions

- Used to ensure data integrity and authentication.

a) SHA (Secure Hash Algorithm)

- A Family of cryptographic hash functions developed by the NSA (National Security Agency) and standardized by NIST (National Institute of Standards and Technology)

Hash func. don't authenticate the Sender (no Secret Key)

. What are SHA-1, SHA-256, and SHA-512?

→ SHA-1: . Produces a 160-bit hash output.
. Input is processed in 512-bit blocks.
. Was widely used but is now considered insecure due to collision attacks.

→ SHA-256: . Produces a 256-bit hash.
. Input is processed in 512-bit blocks
. Used in blockchains, digital signatures, authentication.

→ SHA-512: . Produces a 512-bit hash
. Input is processed in 1024-bit blocks
. Offers stronger security.
. Used for high-security applications.

. Characteristics (For All)

→ Fixed-Length Output: Converts any input into a hash of fixed-size.

→ Deterministic: The same input always produces the same hash

→ Avalanche Effect: A small change in input results in a completely different hash.

→ Collision Resistance: The difficulty of finding two different inputs with the same hash
. SHA-1: Weak (Collisions have been found)
. SHA-256: Strong
. SHA-512: Stronger than SHA-256.

→ Preimage Resistance: The difficulty of finding the original input from a hash.
  . SHA_1: Moderate (not fully secured)
  . SHA_256 & SHA_512: Strong
→ Computation Speed and Efficiency:
  . SHA_1 & SHA_256: Faster
  . SHA_512: Slower but more secure

. Steps (For All):
Each SHA algo. follows a similar process but differs in block size, hash size, nb. of rounds and message expansion.

1. Padding the Message:
  . The message is padded so that its length is a multiple of the required block size.
  . Padding: Single '1' bit followed by '0' bits
  . Padding is applied even if the message is already of the required length.
2. Append Message Length:
  . A block representing the original message length is appended
  . This value is treated as an unsigned integer representing the length before padding
    → SHA_1 & SHA_256: Append a 64-bit block
    → SHA_512: Append a 128-bit block
3. Initialize the Buffer: A buffer is initialized with predefined hash values and holds intermediate and final results:
→ SHA_1: Uses a 160-bit buffer (5 five 32-bit registers)
→ SHA-256: 256-bit buffer (eight 32-bit registers)
→ SHA_512: 512-bit buffer (eight 64-bit registers)

4. Process Blocks: Each block undergoes multiple rounds of logical and arithmetic operations
   - → SHA_1: Each 512_bit block is processed in 4 rounds of 20 steps (80 steps)
   - → SHA_256: Each 512_bit block is processed in 64 rounds
   - → SHA_518: Each 1024_bit block is processed in 80 rounds.
5. Output the Hash: After processing all blocks the final hash value is computed.
   - . SHA_1 : 160_bit hash
   - . SHA_256 : 256_bit hash
   - . SHA_512 : 512_bit hash

. Applications:
- → SHA_1: Used in older SSL/TLS certificates, version control systems (Git)
- → SHA_256: Used in blockchain, digital signature, authentication, SSL/TLS
- → SHA_512: Used in password hashing (bcrypt, HMAC_SHA_512), high security cryptographic applications

MAC alone may not be strong enough for security

b) MAC (Message Authentication Code)
- Small piece of data generated from a message and a secret key using a MAC function.
- Process:
  1) Sender generates a MAC using MAC = F(K, M)
  2) Sender sends the message and MAC to the receiver
  3) Receiver recomputes the MAC (using the same key) and verifies it against the received MAC

HMAC provide both message integrity and authentication

c) HMAC (Hash-based Message Authentication Code)
- Combines a cryptographic hash function (e.g SHA) with a secret key to provide both message integrity and authentication
- Formula:

$$HMAC(K, M) = H((K^+ \oplus opad) \| H((K^+ \oplus ipad) \| M))$$

3. Properties of a Good Hash Function
- → Fixed output size
- → Fast Compilation (easy to compute H(M))
- → One way property (impossible to find original message)
- → Weak Collision Resistance (Infeasible to find two messages with the same hash H(x) = H(y))
- → Strong Collision Resistance (Impossible to find H(x) = H(y))