SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

AN AUTONOMOUS INSTITUTION

DEPARTMENT OF CSE(IOT & CS Including BCT)

Message Authentication Algorithms (MACs) and Hash Functions

Message Authentication Algorithms (MACs) and Hash Functions are both crucial concepts in cryptography used to ensure data integrity and authenticity. However, they serve different purposes and operate in distinct ways. Let's explore them in detail.

1. Message Authentication Algorithms (MACs)

A **Message Authentication Code (MAC)** is a short piece of information derived from a message and a secret key. It is used to verify both the integrity of the message (whether it has been altered) and the authenticity (whether it came from a legitimate source). MACs are commonly used in secure communication protocols and data storage.

- **Purpose**: To authenticate the message's sender and ensure the message has not been altered in transit.
- **How it works**: A MAC is generated by applying a cryptographic algorithm to the message and a secret key. The resulting value is appended to the message. When the recipient receives the message, they can recompute the MAC using the same key and verify whether the MAC matches the one sent with the message.

Types of MACs:

1. **HMAC (Hash-based MAC)**: Uses a cryptographic hash function (e.g., SHA-256) in combination with a secret key.

   - **HMAC Example**: HMAC_SHA256(message, key)
   - **Security**: HMAC provides high security and resistance to attacks like collision or preimage attacks because of the properties of the hash function.
2. **CMAC (Cipher-based MAC)**: Uses a block cipher (e.g., AES) to generate a MAC.
   - **CMAC Example**: CMAC_AES(message, key)
   - **Security**: CMAC is generally considered secure if the block cipher is strong and the key is kept secret.

2. Hash Functions

A **Hash Function** is a mathematical function that takes an input (or "message") and returns a fixed-size string of bytes. The output, called a hash value or digest, represents the original data. Cryptographic hash functions are designed to be computationally hard to reverse (preimage resistance) and to produce unique outputs for distinct inputs (collision resistance).

- **Purpose**: To produce a unique and fixed-size output for any input data. Hash functions are used in various applications like data integrity checks, digital signatures, and password storage.

Properties of Cryptographic Hash Functions:

1. **Deterministic**: The same input always produces the same output.
2. **Fixed-size output**: No matter the size of the input, the output will always be a fixed size (e.g., SHA-256 always produces a 256-bit hash).
3. **Preimage Resistance**: It is hard to reverse the hash function (given a hash, it is computationally infeasible to find the original input).
4. **Collision Resistance**: It is difficult to find two distinct inputs that produce the same hash value.
5. **Avalanche Effect**: A small change in input drastically changes the output.

Common Hash Functions:

- **MD5** (Message Digest Algorithm 5): A widely used hash function that produces a 128-bit hash value. However, it is considered broken due to vulnerabilities to collisions.
- **SHA-1** (Secure Hash Algorithm 1): Produces a 160-bit hash value, but also has known vulnerabilities.
- **SHA-256** and **SHA-3**: Part of the SHA-2 family and newer SHA-3 family, these are widely considered secure and produce 256-bit and other sizes of output.

Differences Between MACs and Hash Functions

1. **Purpose**:
    - **MACs**: Used for verifying both the integrity and authenticity of a message using a secret key.
    - **Hash Functions**: Used to produce a fixed-size output (hash) that represents data, primarily for integrity verification.
2. **Key Usage**:
    - **MACs**: Require a secret key for generation and verification.
    - **Hash Functions**: Do not use any secret key, just the input data.
3. **Security Goals**:
    - **MACs**: Ensures authenticity and integrity (i.e., who sent it and whether it was modified).
    - **Hash Functions**: Ensures integrity but does not authenticate the sender.

Applications

- **MACs**: Often used in secure communications protocols like TLS (Transport Layer Security), IPSec, and in digital signatures.
- **Hash Functions**: Used in applications like file integrity checks (e.g., checksums), digital signatures, password storage, and blockchain technology.