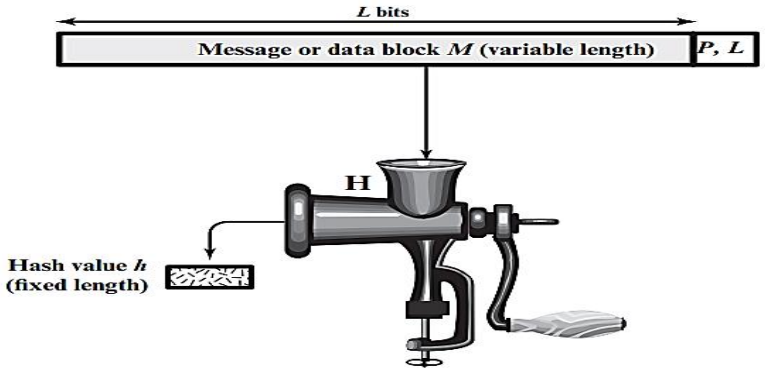
UNIT 04:
CRYPTOGRAPHIC DATA INTEGRITY ALGORITHMS **AFTAB-1405**

01. WHAT IS HASH FUNCTION? WITH NEAT DIAGRAM EXPLAIN BASIC USES OF HASH FUNCTION.



P, L = padding plus length field

Figure 11.1 Cryptographic Hash Function; $h = H(M)$

Hash Function Explained

A hash function is a ***mathematical function*** that takes an input of any size and ***converts it into a fixed-size output,*** called a ***hash value or digest.*** This ***hash value acts like a fingerprint for the input data.***

Key Properties of a Good Hash Function:

- **Even distribution:** When applied to a large dataset, the ***hash function should produce a random and evenly distributed set of hash values.***
- **Collision resistance:** It should be very difficult, to find ***two different inputs that produce the same hash value (collisions).***
- **Sensitivity:** Even a ***minor change in the input data*** should ***result in a significantly different hash value.***

Benefits of Hash Functions:

- **Data Integrity:** Hash functions are used to ***verify the integrity of data.*** If the ***data is tampered with during transmission or storage,*** the ***hash value will change, indicating a modification.***

Basic Uses of Hash Functions:

- **Checksums:** Hash functions are often ***used to create checksums for files.*** The ***hash value of a file is calculated before transmission or storage.*** After receiving or retrieving the file, the hash value is calculated again and compared to the original hash. If the values match, it's likely the data hasn't been altered.
- **Digital Signatures:** Hash functions are a crucial part of digital signatures. A ***sender signs a message with their private key and the hash of the message.*** The ***receiver can verify the signature using the sender's public key and recalculating the hash*** of the received message. If the ***signatures match, it ensures the message authenticity and integrity.***

- **Password Storage:** Hash functions are ***used to securely store passwords.*** Instead of storing passwords directly, systems typically store hash values of the passwords. When a user enters a password, the system hashes it and compares it to the stored hash. ***This approach protects passwords from being stolen in a data breach.***
- **Data Deduplication:** Hash functions are ***used to identify duplicate data.*** By comparing hash values of files, systems can efficiently determine if the data already exists, ***saving storage space.***

In conclusion, ***hash functions are a versatile cryptographic tool*** with various applications that ensure ***data integrity, security, and efficiency.***

02. STATE MESSAGE AUTHENTICATION REQUIREMENTS. EXPLAIN BASIC USES OF MESSAGE ENCRYPTION. (2 TIMES)

Message authentication is a ***cryptographic procedure used to verify the authenticity and integrity of a message,*** ensuring that ***it has not been altered*** and that ***it indeed originated from the purported sender.*** This process helps in safeguarding against various types of attacks, including ***masquerade, content modification, sequence modification, and timing modification,*** among others.

Authentication Requirements:

1. **Disclosure**: ***Preventing unauthorized access*** to message contents.
2. **Traffic Analysis**: Ensuring ***confidentiality*** of the traffic pattern.
3. **Masquerade**: ***Detecting*** and ***preventing messages*** from ***fraudulent sources***.
4. **Content Modification**: ***Detecting any changes made*** to the ***message contents***.
5. **Sequence Modification**: Detecting any ***reordering*** or manipulation of message sequences.
6. **Timing Modification**: Preventing ***delays or replays*** of messages.
7. **Source Repudiation**: Ensuring the sender cannot deny sending the message.
8. **Destination Repudiation**: Ensuring the receiver cannot deny receiving the message.

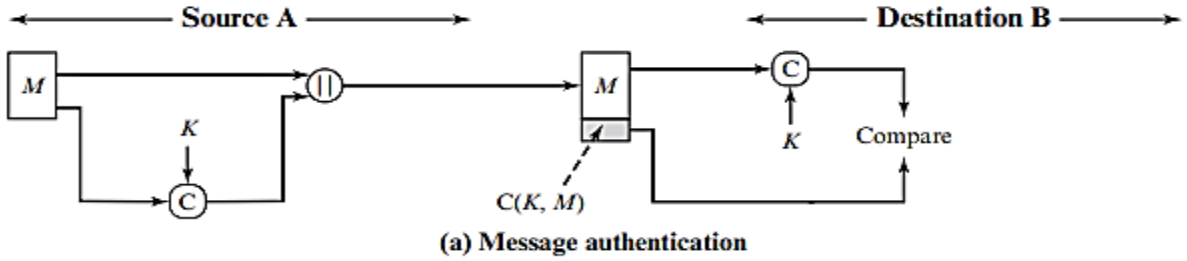
Basic Uses of Message Authentication:

1. **Message Integrity**: By applying cryptographic techniques such as ***hashing or message authentication codes (MACs),*** the ***integrity of the message can be verified***. Any alteration to the message during transmission will result in a ***different hash value or MAC, indicating tampering.***
2. **Message Origin Authentication**: ***Digital signatures*** are commonly ***used to authenticate the origin of a message.*** The sender generates a digital signature using their private key, which can be verified by the receiver using the sender's public key. This ensures that the message indeed originated from the claimed sender.

3. **Sequencing and Timeliness Verification**: Authentication mechanisms can also include measures to verify the ***sequencing and timeliness of messages.*** For example, ***sequence numbers or timestamps can be included in messages*** to ensure their correct order and timely delivery.

In summary, ***message authentication plays a crucial role in ensuring the trustworthiness of communication*** in various applications by verifying the ***authenticity, integrity, sequencing, and timeliness of messages*** exchanged between parties.

03. WHAT IS MESSAGE AUTHENTICATION CODES (MAC)? WITH NEAT DIAGRAM EXPLAIN THE BASIC USES OF MAC.



(a) Message authentication

Message Authentication Codes (MACs) are cryptographic techniques used to ***authenticate messages, ensuring their integrity and origin. MACs are generated using a secret key shared between the sender and the receiver.*** The process involves calculating a ***fixed-length authenticator*** based on the message and the secret key. This ***MAC is then appended to the message*** and transmitted to the receiver.

Here's how MAC works in the authentication function:

1. **Message and Key**: Assume two parties, A and B, share a secret key, K. When A wants to send a message, M, to B, it calculates the MAC using a ***MAC function (denoted as C)*** and the shared secret key: ***MAC = C(K, M).***
2. **Transmission**: ***'A'*** transmits the message ***'M'*** along with the calculated ***'MAC'*** to ***'B'.***
3. **Verification**: Upon receiving the message and MAC, B performs the same calculation using the received message and the shared secret key: ***MAC' = C(K, M').*** If ***MAC'*** matches the received MAC, authentication is successful.

MAC provides the following security assurances:

- **Message Integrity**: If an attacker alters the message during transmission, the MAC calculated by the receiver will differ from the received MAC, indicating tampering.
- **Message Origin Authentication**: Since only the sender and receiver know the secret key, only the sender can generate a valid MAC. Therefore, the receiver can trust that the message originated from the alleged sender.
- **Sequence Verification**: If the message includes a sequence number, the receiver can verify the proper sequence of messages, ensuring that no segments are delayed, reordered, or deleted.

UNIT 04:
CRYPTOGRAPHIC DATA INTEGRITY ALGORITHMS **AFTAB-1405**

MACs are ***similar to encryption, but they are not reversible like encryption algorithms.*** The MAC algorithm is typically a ***many-to-one function,*** where multiple messages can produce the same MAC. However, the ***probability of generating the same MAC for different messages should be sufficiently low*** to prevent unauthorized authentication.

Applications of MACs:

- **Network Security**: Used in various network protocols to ensure ***data integrity and authentication.***
- **Banking and Financial Transactions**: Ensures the ***integrity of the transactions*** and ***authentication of messages*** exchanged between banks and clients.
- **Software Distribution**: Authenticates the source of software updates and verifies that the update has not been tampered with before installation.

In summary, ***MACs provide a robust method for authenticating messages*** in communication systems, ***ensuring both integrity and origin verification*** using a shared secret key between the communicating parties.

DIGITAL SIGNATURE:

A digital signature is a cryptographic technique used to ensure the ***authenticity, integrity, and non-repudiation of digital messages or documents.*** It functions similarly to a ***handwritten signature or a stamped seal*** but in the digital realm. Here's a detailed explanation of digital signatures:

1. Purpose of Digital Signatures:

- **Authenticity:** Digital signatures ***verify the identity of the signer.***
- **Integrity:** They ensure that the content of the message or document has not been altered.
- **Non-repudiation:** The signer cannot deny signing the document once the signature is applied.
- **Timestamping:** Optionally, ***digital signatures can include a timestamp to provide evidence of when the document was signed.***

2. Components of Digital Signature:

- **Message or Document:** The content to be signed.
- **Private Key:** A secret key held only by the signer used for generating the digital signature.
- **Public Key:** A publicly available key used for verifying the digital signature.
- **Hash Function:** A cryptographic hash function used to ***create a unique digest or hash value of the message.***

UNIT 04:
CRYPTOGRAPHIC DATA INTEGRITY ALGORITHMS **AFTAB-1405**

Digital signatures play a crucial role in various applications, including ***electronic transactions, secure communication, and document authentication,*** by providing a reliable method for verifying the authenticity and integrity of digital content.

4. EXPLAIN DIRECT DIGITAL SIGNATURE AND ARBITED DIGITAL SIGNATURE.

Direct Digital Signature (DDS)

In a Direct Digital Signature, ***only the communicating parties (sender and receiver) are involved.*** When Alice sends a message to Bob with her digital signature, she uses her private key to sign her message. Upon receiving the message and its signature, Bob uses Alice's public key to verify the legitimacy of both the message and Alice's identity.

Process

For example, if Alice wants to send a secure message to Bob:

1. **Signing:**
   - Alice creates a hash of the message.
   - She encrypts this hash with her private key, forming the digital signature.
   - She sends the original message with the digital signature to Bob.
2. **Verification:**
   - Bob decrypts the digital signature using Alice's public key to retrieve the hash.
   - He then hashes the received message himself.
   - If both hash values match, Bob is confident that the message is authentic and unchanged.

Advantages:

- Simple and efficient for direct communication between two parties.
- ***No third-party involvement,*** making it ***faster and potentially cheaper.***

Disadvantages:

- **Limited scope:** Only suitable for two-party communication.

Arbitrated Digital Signature (ADS)

Explanation

An Arbitrated Digital Signature ***involves a third-party arbitrator*** in addition to the sender and the receiver. The ***arbitrator checks and confirms the signatures*** before they reach the receiver. This extra step ensures an ***additional layer of security*** where the ***authenticity of the identity and the integrity of the message are guaranteed by an impartial third party.***

Process

If Alice wants to send Bob a message with an arbitrator involved:

1. **Signing**:
    - Alice signs her message using her private key.
    - She sends the message and her signature to the arbitrator.
2. **Arbitration**:
    - The arbitrator verifies Alice's signature using her public key.
    - If valid, the arbitrator then signs the message with his private key and sends it to Bob.
3. **Verification**:
    - Bob first uses the arbitrator's public key to verify the arbitrator's signature.
    - If that is valid, he then uses Alice's public key to ensure her signature on the original message is authentic.

Advantages:

- Provides higher security and legal enforceability due to the involvement of a trusted third party.
- Can resolve disputes about the authenticity of the signature.
- Suitable for situations requiring non-repudiation and high levels of trust.

Disadvantages:

- More complex and involves additional cost due to the TTP's involvement.
- May introduce delays due to the extra step of involving the arbiter.

In summary:

- **Direct digital signatures** are simpler and faster for direct communication, but lack the legal weight and dispute resolution capabilities of arbiter signatures.
- **Arbiter digital signatures** offer greater security and legal enforceability, but involve a trusted third party and additional complexity.

5. STATE REQUIREMENTS OF DIGITAL SIGNATURE. EXPLAIN IN DETAIL ARBITRATED DIGITAL SIGNATURE.

Requirements of Digital Signature

A digital signature is a mathematical scheme for verifying the authenticity and integrity of digital messages or documents. Digital signatures have several requirements to ensure they serve their purpose effectively:

1. **Authentication**:
    - Digital signatures must verify the identity of the signatory and ascertain that the signature corresponds to the entity claiming to have written the message.
2. **Integrity**:
    - It must be impossible to alter a message without affecting its associated signature. Therefore, a digital signature should be able to show any alteration made to a document after it was signed.

3. **Non-repudiation**:
    - A digital signature should prevent the sender from denying the authorship or sending of the message. This is crucial for legal and contractual purposes.
4. **Inalterability**:
    - ***Once a document is signed, its content should not be modifiable.*** The ***digital signature should seal the document,*** making unauthorized changes detectable.

Arbitrated Digital Signature

The Arbitrated Digital Signature involves a third party to validate each signed message. This model is essential in scenarios where enhanced security measures are necessary, serving as an assurance that the received messages are genuinely from a verifiable source and have not been tampered with during transit.

Detailed Explanation

In an arbitrated digital signature, a trusted third party known as an arbitrator is involved in the signing process. The arbitrator is responsible for certifying the authenticity of the signer's digital signature before it reaches the recipient. This approach adds an extra layer of security by validating each signed digital message through the arbitrator.

Process of Arbitrated Digital Signatures:

1. **Message Generation**:
    - The sender, say Alice, creates a message that she wants to send to the receiver, say Bob. Alice then prepares the message for signing.
2. **Signing**:
    - Alice signs the message using her private key. This signed message, however, does not go directly to Bob.
3. **Arbitration**:
    - The signed message is sent to the arbitrator. The arbitrator uses Alice's public key to validate the signature. If the validation succeeds, the arbitrator knows the message is genuinely from Alice and has not been altered.
    - After validating, the arbitrator re-signs the message using the arbitrator's private key, adding an additional layer of authentication and security.
4. **Sending to Receiver**:
    - The re-signed message is then sent to Bob. Upon receiving the message, Bob uses the public keys of Alice and the arbitrator to verify both signatures.
5. **Verification**:
    - If both signatures are verified successfully, Bob can be confident in the message's origin and integrity.

Advantages of Arbitrated Digital Signatures:

- **Enhanced Security**: Involvement of a neutral arbitrator adds a layer of trust and validation not present in typical digital signature schemes.

- **Prevention of Fraud**: The arbitrator checks both the origin and integrity of the message, significantly reducing the risk of fraud.
- **Legal Compliance**: For certain sensitive or high-value transactions, arbitrated signatures may meet higher legal or compliance standards than regular digital signatures.

In summary, arbitrated digital signatures provide a robust mechanism for securing digital communications, adding a level of validation that is vital in many high-security areas. They meet the stringent requirements of authentication, integrity, and non-repudiation while offering advantages such as fraud prevention and compliance with high legal standards.