Network Security

6. Message Authentication and
Hash Functions

Message Authentication

- message authentication is concerned with:
  - protecting the integrity of a message
  - validating identity of originator
  - non-repudiation of origin (dispute resolution)
- will consider the security requirements
- then three alternative functions used:
  - message encryption
  - message authentication code (MAC)
  - hash function

Security Requirements

In communications across a network, the following attacks can be identified.

- disclosure
- traffic analysis

message confidentiality

- masquerade
- content modification
- sequence modification
- timing modification

message authentication

- repudiation

digital signature

Authentication functions

- Any message authentication (or digital signature) mechanism has 2 levels of functionality
  1. At lower level, there must by some sort of function that produces an authenticator
  2. This lower level functionality is used by higher level authentication protocol that enables a receiver to verify the authenticity of a message

Authenticator functions

Functions that may be used to produce an authenticator can be grouped into three classes …

- message encryption
- message authentication code (MAC)
- hash function

Message Encryption

- message encryption by itself also provides a measure of authentication
- if symmetric encryption is used then:
  - receiver knows sender must have created it
  - since only sender and receiver know key used
  - know content cannot be altered
  - if message has suitable structure, redundancy or a checksum to detect any changes

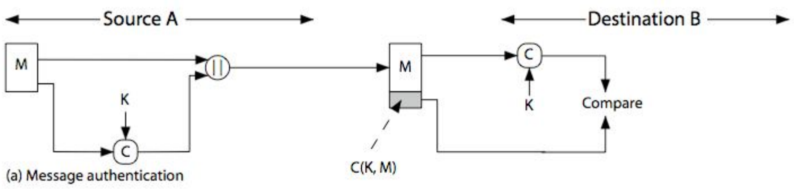
Message Encryption

- if public-key encryption is used:
  - encryption provides no confidence of sender
  - since anyone potentially knows public-key
  - however if
    - sender **signs** message using their private-key
    - then encrypts with recipients public key
    - have both secrecy and authentication
  - again need to recognize corrupted messages
  - but at cost of two public-key uses on message

Message Authentication Code (MAC)

- generated by an algorithm that creates a small fixed-sized block
  - depending on both message and some key
  - like encryption though need not be reversible
- appended to message as a **signature**
- receiver performs same computation on message and checks it matches the MAC
- provides assurance that message is unaltered and comes from sender

Message Authentication Code



Message Authentication Codes

- as shown the MAC provides authentication
- can also use encryption for secrecy
  - generally use separate keys for each
  - can compute MAC either before or after encryption
  - is generally regarded as better done before
- why use a MAC?
  - sometimes only authentication is needed
  - sometimes need authentication to persist longer than the encryption (eg. archival use)
- note that a MAC is not a digital signature

MAC Properties

- a MAC is a cryptographic checksum

  $\mathtt{MAC} = \mathtt{C}_{\mathtt{K}}(\mathtt{M})$

  - condenses a variable-length message M
  - using a secret key K
  - to a fixed-sized authenticator
- is a many-to-one function
  - potentially many messages have same MAC
  - but finding these needs to be very difficult

RQ 11

Requirements for MACs

- taking into account the types of attacks
- need the MAC to satisfy the following:
  1. knowing a message and MAC, is infeasible to find another message with same MAC
  2. MACs should be uniformly distributed
  3. MAC should depend equally on all bits of the message

RQ 12

Using Symmetric Ciphers for MACs

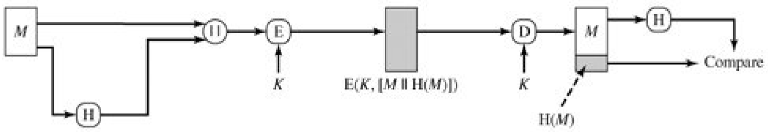
- can use any block cipher chaining mode and use final block as a MAC
- **Data Authentication Algorithm (DAA)** is a widely used MAC based on DES-CBC
  - using IV=0 and zero-pad of final block
  - encrypt message using DES in CBC mode
  - and send just the final block as the MAC
    - or the leftmost M bits ($16 \le M \le 64$) of final block
- but final MAC is now too small for security

Hash Functions

- condenses arbitrary message to fixed sized **hash code** / **message digest** / **hash value**
  `h = H(M)`
- usually assume that the hash function is public and not keyed
  - Remember MAC, which is keyed
- hash used to detect changes to message
- can use in various ways with message
- most often to create a digital signature

Uses of Hash Function - 1

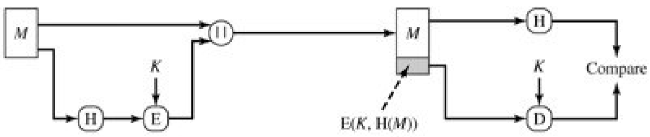
- The message plus concatenated hash code is encrypted using symmetric encryption.



(a)

Uses of Hash Function - 2

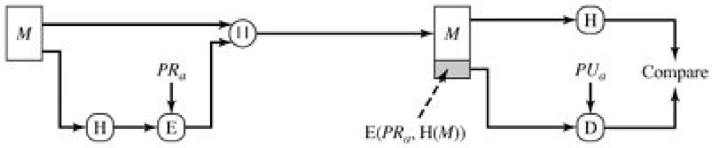
- Only the hash code is encrypted, using symmetric encryption.
- This reduces the processing burden for those applications that do not require confidentiality.



(b)

Uses of Hash Function - 3

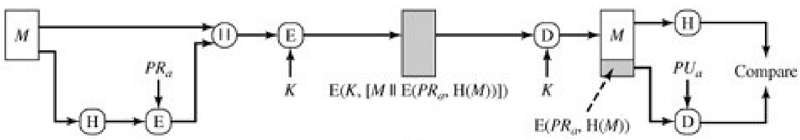
- Only the hash code is encrypted, using public-key encryption and using the sender's private key.
- This provides authentication.
- It also provides a digital signature, because only the sender could have produced the encrypted hash code.
  - In fact, this is the essence of the digital signature technique.



(c)

Uses of Hash Function - 4

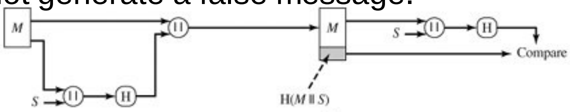
- If confidentiality as well as a digital signature is desired, then the message plus the private-key-encrypted hash code can be encrypted using a symmetric secret key.



(d)

Uses of Hash Function - 5

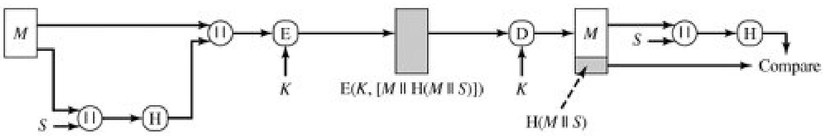
- It is possible to use a hash function but no encryption for message authentication.
- The two parties A & B share a common secret value S
- A computes the hash value over the concatenation of M and S and appends the resulting hash value to M .
- Because B possesses S , it can recompute the hash value to verify.
- Because the secret value itself is not sent, an opponent cannot modify an intercepted message and cannot generate a false message.



(e)

Uses of Hash Function - 6

- Confidentiality can be added to the previous approach by encrypting the entire message plus the hash code.



(f)

Requirements for Hash Functions

1. can be applied to any sized message `M`
2. produces fixed-length output `h`
3. is easy to compute `h=H(M)` for any message `M`
4. given `h` is infeasible to find `x` s.t. `H(x)=h`
   - one-way property
5. given `x` is infeasible to find `y` s.t. `H(y)=H(x)`
   - weak collision resistance
6. is infeasible to find any `x,y` s.t. `H(y)=H(x)`

Simple Hash Functions

- are several proposals for simple functions
- based on XOR of message blocks
- not secure since can manipulate any message and either not change hash or change hash also
- need a stronger cryptographic function

Summary

- have considered:
  - message authentication using
    - message encryption
    - MACs
    - hash functions