CPE 542: CRYPTOGRAPHY & NETWORK SECURITY

Chapter 11 – Message Authentication and Hash Functions

Dr. Lo'ai Tawalbeh
Computer Engineering Department
Jordan University of Science and Technology
Jordan

Message Authentication

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

Security Requirements

- traffic analysis
- content modification
- sequence modification
- timing modification
- source repudiation
- destination repudiation

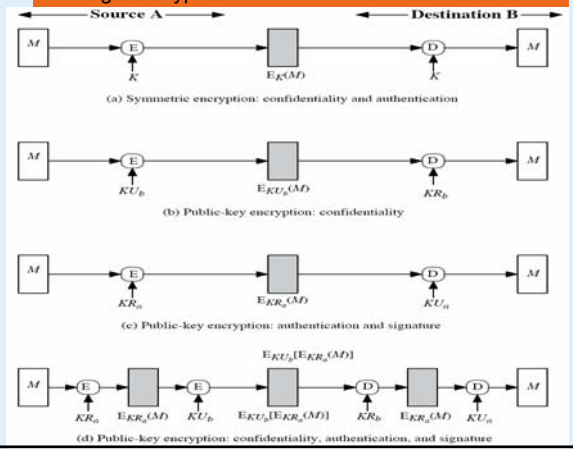
Message Encryption

- message encryption by it self 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 the key used

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 confidentiality and authentication
  - but at cost of two public-key use on a message

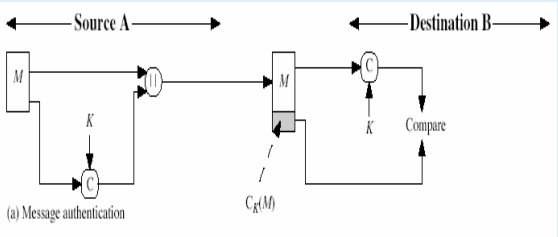
Message Encryption- see Table 11.1 in the book



Message Authentication Code (MAC)

- generated by an algorithm that creates a small fixed-size 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 MAC provides Authentication
- can also use encryption for Confidentiality
  - generally use separate keys for each
  - can compute MAC either before or after encryption
- 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- the same key is shared between the two parties.

Message Authentication Codes

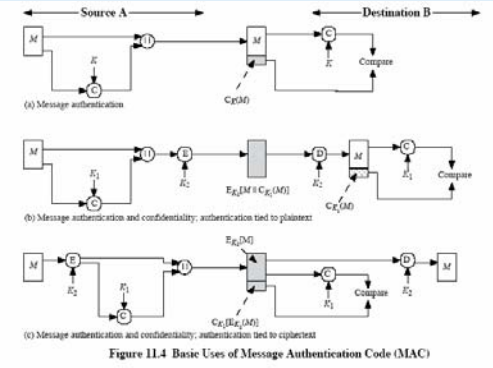


Figure 11.4 Basic Uses of Message Authentication Code (MAC)

MAC Properties

- a MAC is a cryptographic checksum

  $MAC = C_K(M)$

  - 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

Requirements for MACs

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

Using Symmetric Ciphers for MACs

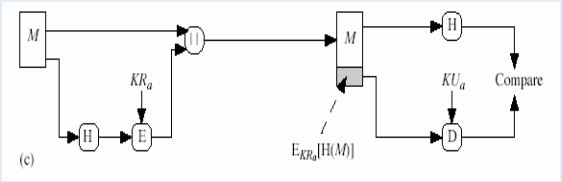
- can use any block cipher chaining mode and use final block as a MAC
- **Data Authentication Algorithm (DAA)** is a widely used MAC based on DES-CBC
  - using IV=0 and zero-pad of final block
  - encrypt message using DES in CBC mode
  - and send just the final block as the MAC
    - or the leftmost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

Hash Functions

- condenses arbitrary message to fixed size
- usually assume that the hash function is public and not keyed
  - different than MAC which is keyed
- hash used to detect changes to message
- can be used in various ways with message, mostly to create a digital signature
- a Hash Function produces a fingerprint of some file/message/data

```
h = H(M)
```


Hash Functions & Digital Signatures



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. is infeasible to find any `x,y` s.t. `H(y)=H(x)`
   - strong collision resistance

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 (next chapter)

Block Ciphers as Hash Functions

- can use block ciphers as hash functions
  - using H_0=0 and zero-pad of final block
  - compute: $H_i = E_{M_i} [H_{i-1}]$
  - and use final block as the hash value
  - similar to CBC but without a key
- resulting hash is too small (64-bit)-

Hash Example: Secure Hash Algorithm-SHA

1. pad message so its length is congruent to 448 mod 512 (first bit 1, then followed by zeros)
1. append a 64-bit integer value to the msg (cantinas the original msg length).
2. initialise 5-word (160-bit) buffer (A,B,C,D,E) to (67452301,efcdab89,98badcfe,10325476,c3d2e1f0)
3. process message in 16-word (512-bit) chunks:
   - expand 16 words into 80 words by mixing & shifting
   - use 4 rounds of 20 bit operations on message block & buffer
   - add output to input to form new buffer value
4. output hash value is the final buffer value

Hash Example: Secure Hash Algorithm-SHA

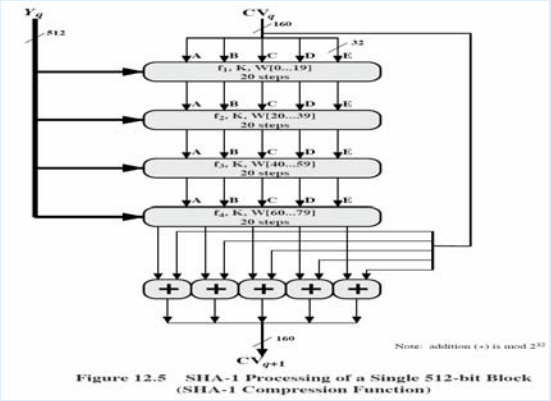


Figure 12.5 SHA-1 Processing of a Single 512-bit Block (SHA-1 Compression Function)

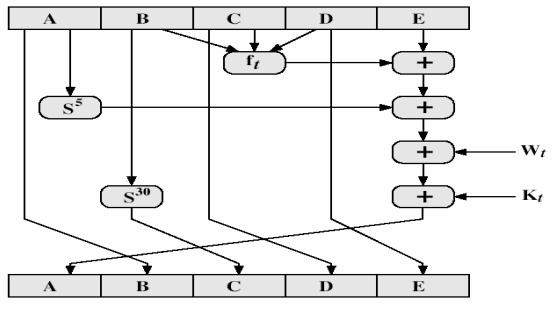
SHA-1 Compression Function

- each round has 20 steps which replaces the 5 buffer words thus:

  $(A,B,C,D,E) \leftarrow (E+f(t,B,C,D)+(A<<5)+W_t+K_t),A,(B<<30),C,D)$

- a,b,c,d refer to the 4 words of the buffer
- t is the step number
- $f(t,B,C,D)$ is nonlinear function for round
- W_t is derived from the message block
- K_t is a constant value derived from sin

SHA-1 Compression Function



Revised Secure Hash Standard

- NIST have issued a revision FIPS 180-2
- adds 3 additional hash algorithms
- SHA-256, SHA-384, SHA-512
- designed for compatibility with increased security provided by the AES cipher
- structure & detail is similar to SHA-1
- hence analysis should be similar