Computer Security

Lecture 11: Message Authentication, Hash Function & Digital Signatures

Prof. Dr. Sadeeq Jan

Department of Computer Systems Engineering

University of Engineering and Technology Peshawar

Lecture Outline

- Message Authentication
- Hash Function
- Digital Signatures

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

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

Message Encryption

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

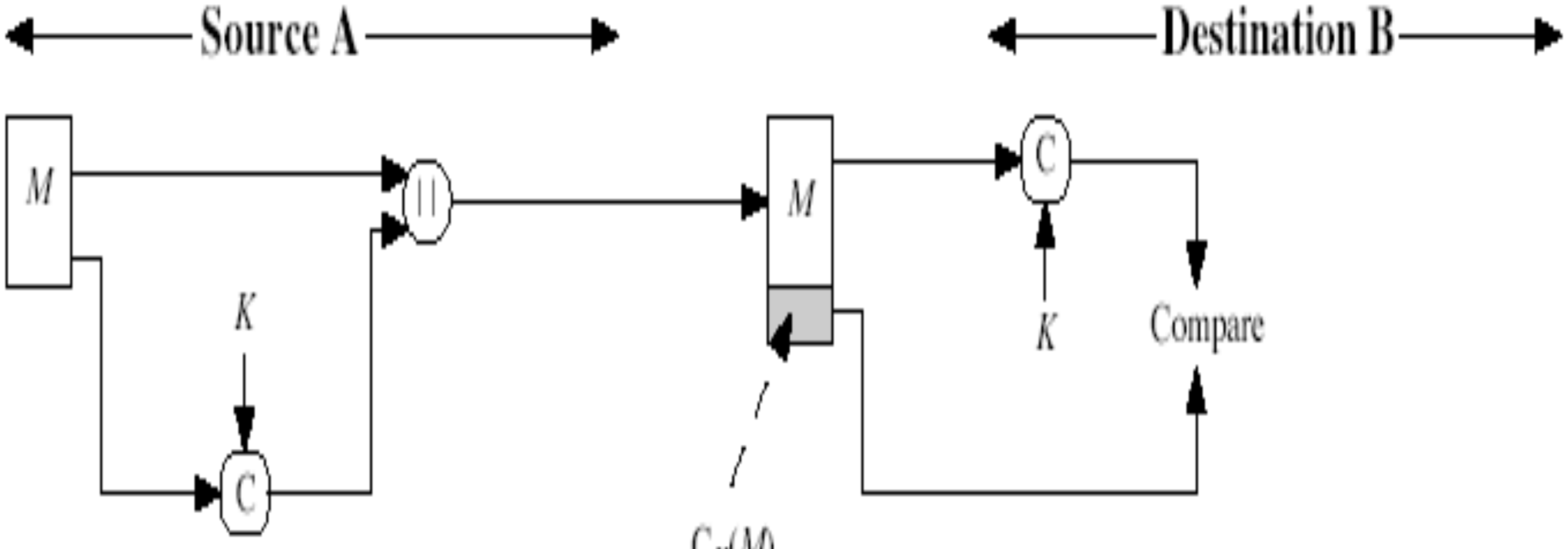
Message Encryption

- 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 the cost of two public-key uses on the 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



(a) Message authentication

Message Authentication Codes

- as shown the MAC provides message 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
- note that a MAC is not a digital signature

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, 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

Chewing functions

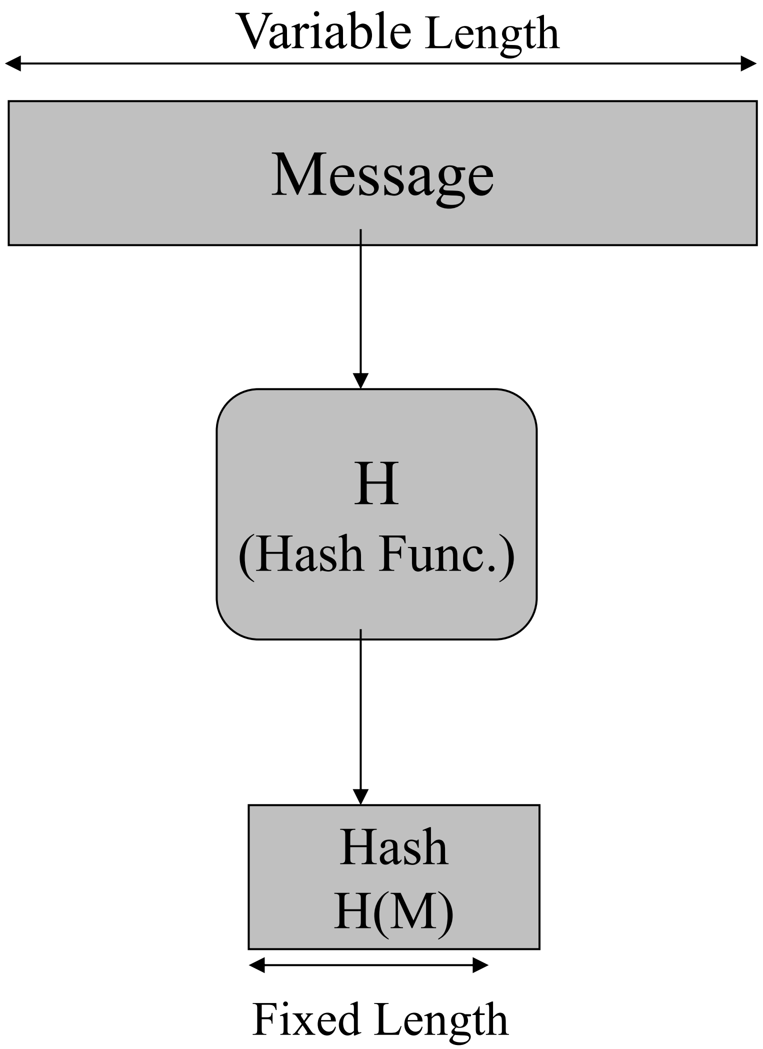
Hashing function as “chewing” or “digest” function

Hash Functions

- condenses arbitrary message to fixed size
- usually assume that the hash function is public and not keyed
  - cf. 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

Hash Functions

- are used to generate fixed-length fingerprints of arbitrarily large messages
- denoted as H(M)
  - M is a variable length message
  - H is the hash function
  - H(M) is of fixed length
  - H(M) calculations should be easy and fast
    - indeed they are even faster than symmetric ciphers



Hash Function Properties

- a Hash Function produces a fingerprint of some file/message/data

  `h = H(M)`

  - condenses a variable-length message M
  - to a fixed-sized fingerprint
- assumed to be public

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)`
    - strong collision resistance

Digital Signatures

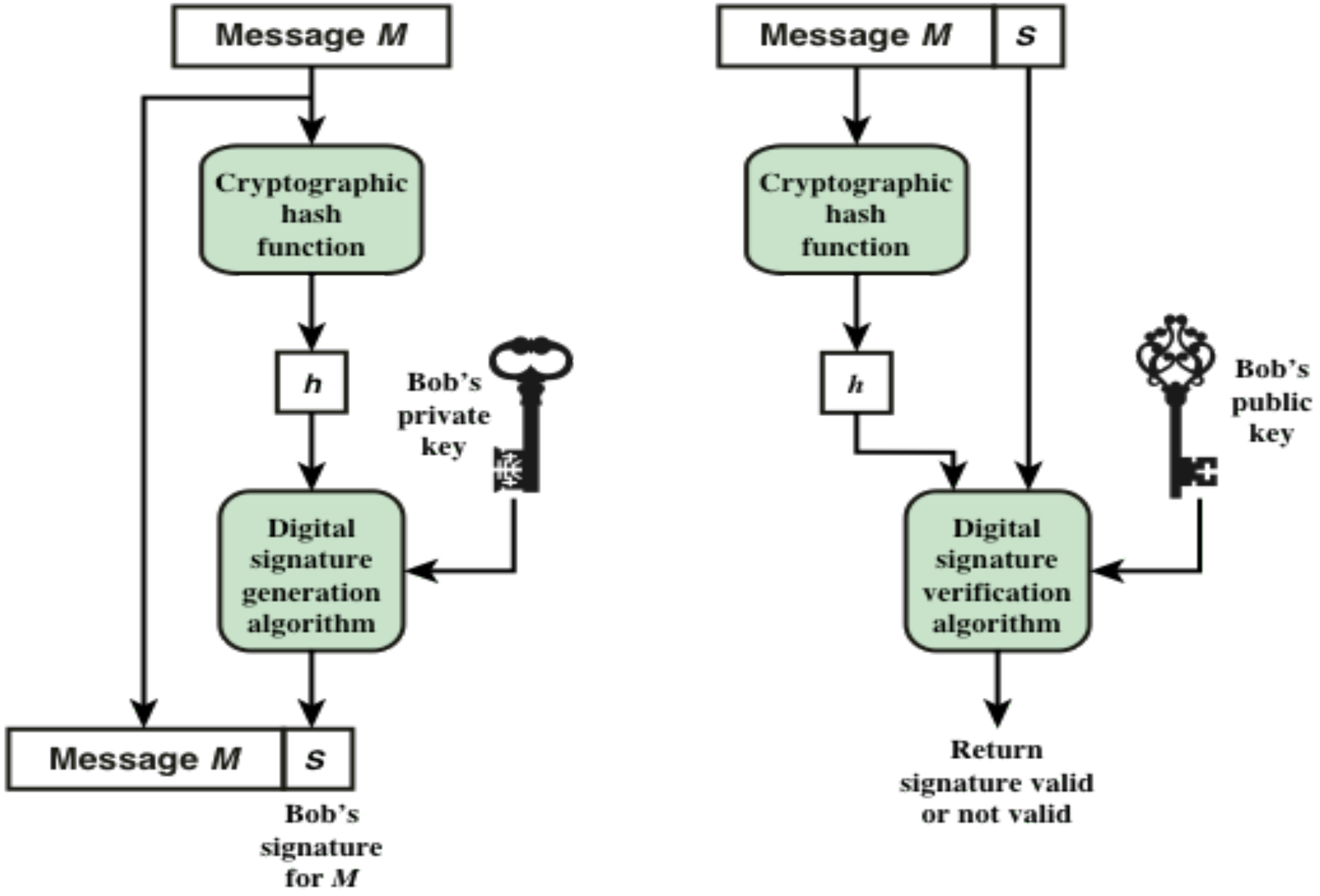
Digital Signatures

- have looked at message authentication
  - but does not address issues of lack of trust
- digital signatures provide the ability to:
  - verify author, date & time of signature
  - authenticate message contents
  - be verified by third parties to resolve disputes
- hence include authentication function with additional capabilities

Digital Signatures

- Mechanism for non-repudiation
- Basic idea
  - use private key on the message to generate a piece of information that can be generated only by yourself
    - because you are the only person who knows your private key
  - public key can be used to verify the signature
    - so everybody can verify
- Generally, signatures are created and verified over the hash of the message
  - Why?

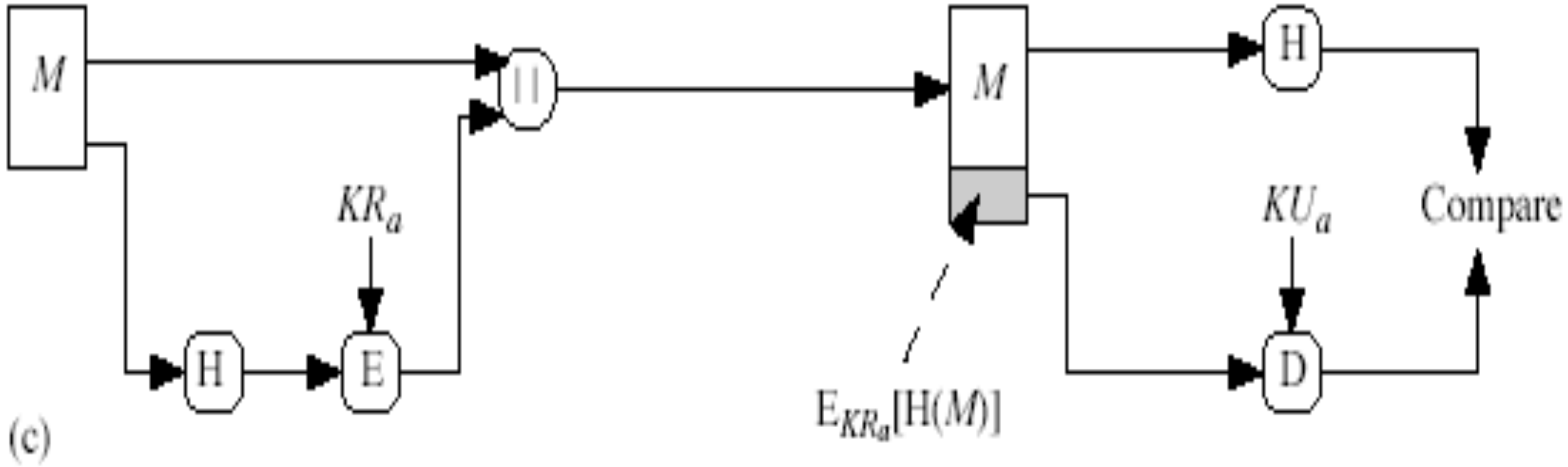
Generic Digital Signature Model



(a) Bob signs a message

(b) Alice verifies the signature

Hash Functions & Digital Signatures



END