Week 6

Lecture 1

Un Keyed Cryptography: Hash Functions

Lecture 2

Message Authentication Codes or Keyed Hash Function

Workshop 3: Workshop based on Lectures in Week5

Quiz 6

Udaya Parampalli

Message Authentication Codes

COMP90043

Lecture 1

Udaya Parampalli

Lecture 2

- 1.1 Message Authentication
  - Issues in Practice
  - Message Encryption-Symmetric and Public key approach

- 1.2 Message Authentication Code
  - Internal and External Error Control
  - MAC in networks
  - Properties and Attacks on MAC

- 1.3 Pseudorandom number generation
  - Using MAC and Hash

Recap:Hash Function Requirements

Requirement	Description
Variable input size	H can be applied to a block of data of any size.
Fixed output size	H produces a fixed-length output.
Efficiency	$H(x)$ is relatively easy to compute for any given x, making both hardware and software implementations practical.
Preimage resistant (one-way property)	For any given hash value h, it is computationally infeasible to find y such that $H(y) = h$.
Second preimage resistant (weak collision resistant)	For any given block x, it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$.
Collision resistant (strong collision resistant)	It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.
Pseudorandomness	Output of H meets standard tests for pseudorandomness

Table 11.1 from the textbook

Hash Function Relationships

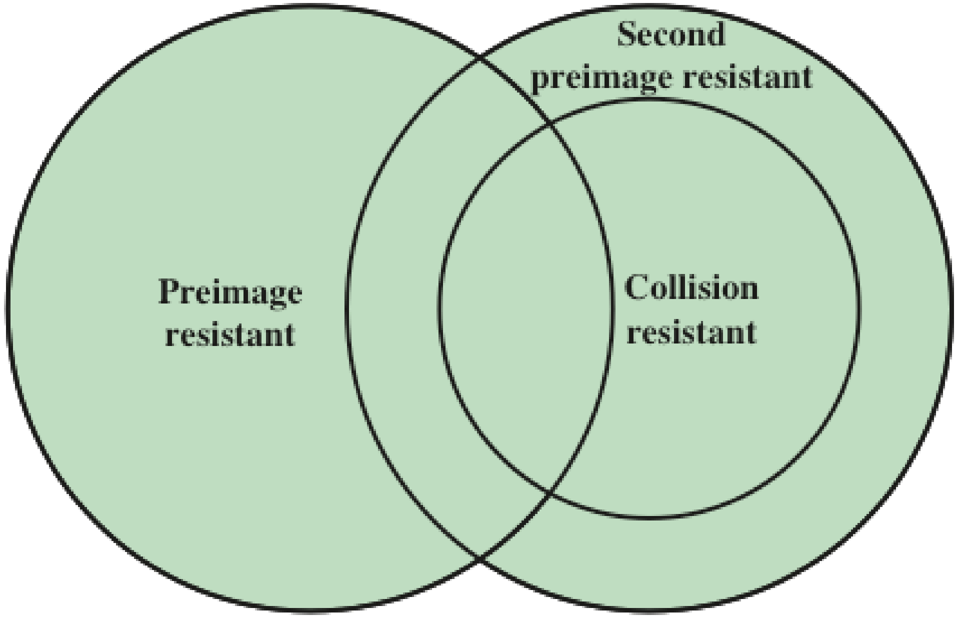


Figure 11.6 Relationship Among Hash Function Properties

Fig 11.6 from the textbook

Lecture 2

- 1.1 Concept of Public Key
  - Limitations of Symmetric key system
  - Notations for Public key
- 1.2 Diffie-Hellman Protocol
  - Motivation
  - The protocol and Implications
  - Man in the Middle Attack
- 1.3 RSA Idea
  - Informal Idea
  - RSA Algorithm
  - Attacks on RSA

Message Authentication

- Let us look at message authentication issue in practice.
- What is it concerned with?
  - To address message authentication
  - A dedicated primitive based on symmetric key cryptography
- Issues for message authentication-
  - Message integrity
  - Validation of originator’s identity
  - Non-repudiation of the message origin
- Three ways of achieving authentication
  - Message Encryption
  - Hash functions (we looked at it in the previous lecture)
  - Message Authentication Code (MAC) (this lecture)

How do we create message authentication

- We need to separate message authentication function and the protocol that helps us to integrate the message authentication in the application.
- At a basic level, we can create a message authentication code using a secret key.
- At a higher level, the keys are carefully managed to obtain higher level guarantees on the exchanged message including source authentication.

Security Requirements

- Stallings discussed the security issues that can arise in the networked systems and consider following requirements:
  - disclosure
  - traffic analysis
  - masquerade
  - content modification
  - sequence modification
  - timing modification
  - source repudiation
  - destination repudiation

- Please read Section 12.1 for details.

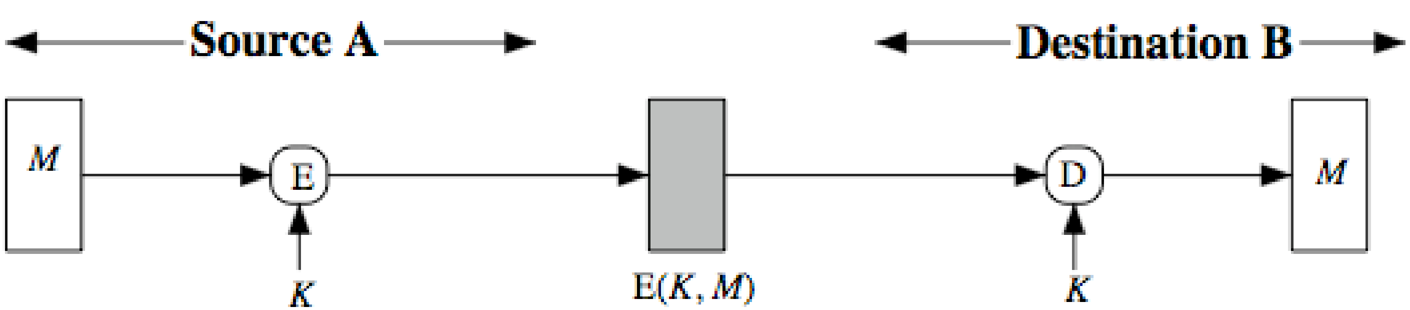
Message Encryption

- First let us understand how message encryption itself provides authentication.
- The issues are different to symmetric and public key methods.
- Note that with public key encryption, anyone could encrypt based on public key of the receiver and if you want source authentication, the sender needs to use signature.
- But symmetric key assumes that sender and receiver share a secret and encryption naturally provides authentication.
- Stallings Section 12.2 gives an account of these discussions.
- Let is first consider Symmetric Encryption.

Symmetric key Encryption

- How authentication is obtained?
  - Since they share the key, receiver is sure that the message was created by the sender.
  - By relying on format and structure of the messages, they can detect any modification,

  Next, we consider other situations:



(a) Symmetric encryption: confidentiality and authentication

Fig 12 (a) from the textbook

Basic Use of Encryption

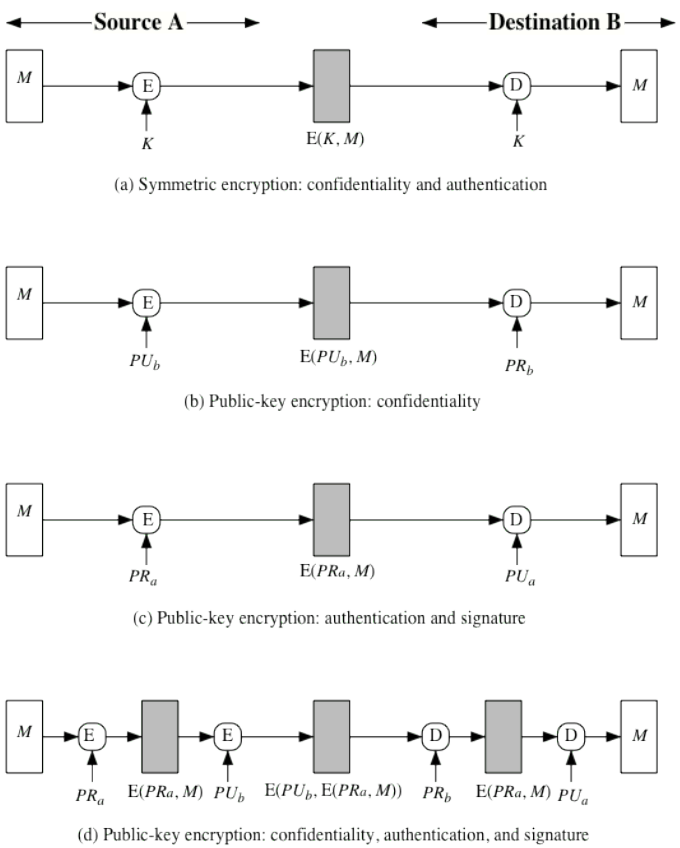


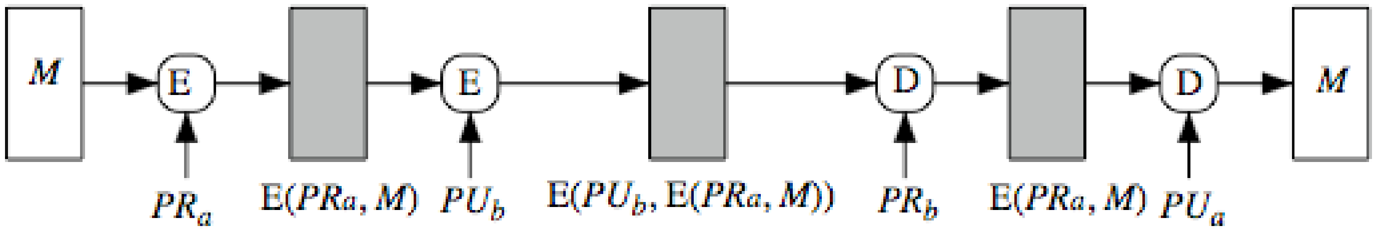
Figure 12.1 Basic Uses of Message Encryption

Read the discussion around Fig. 12.1 in the textbook

Fig 12 from the textbook

Public Key Encryption

- Public key by nature, anyone can use.
- Does not provide any guarantee for the sender.
- To provide authentication, a sender needs to sign as well (use private key) which can be verified by others using the public key.
- How do we decide if the message stream is corrupted or not?
- You need some general formatting rules.



(d) Public-key encryption: confidentiality, authentication, and signature

Fig 12.1from the textbook

Message Authentication Code (MAC)

- A dedicated primitive to address mainly authentication using a key.
- The output of an algorithm can act as a signature.
- Only the receiver with the key can verify the code by running the same algorithm, thus assuring the integrity of the message from the sender.
- There are two ways of using the message authentication code:
  - Internal Error Control
  - External Error Control

Different Error Controls

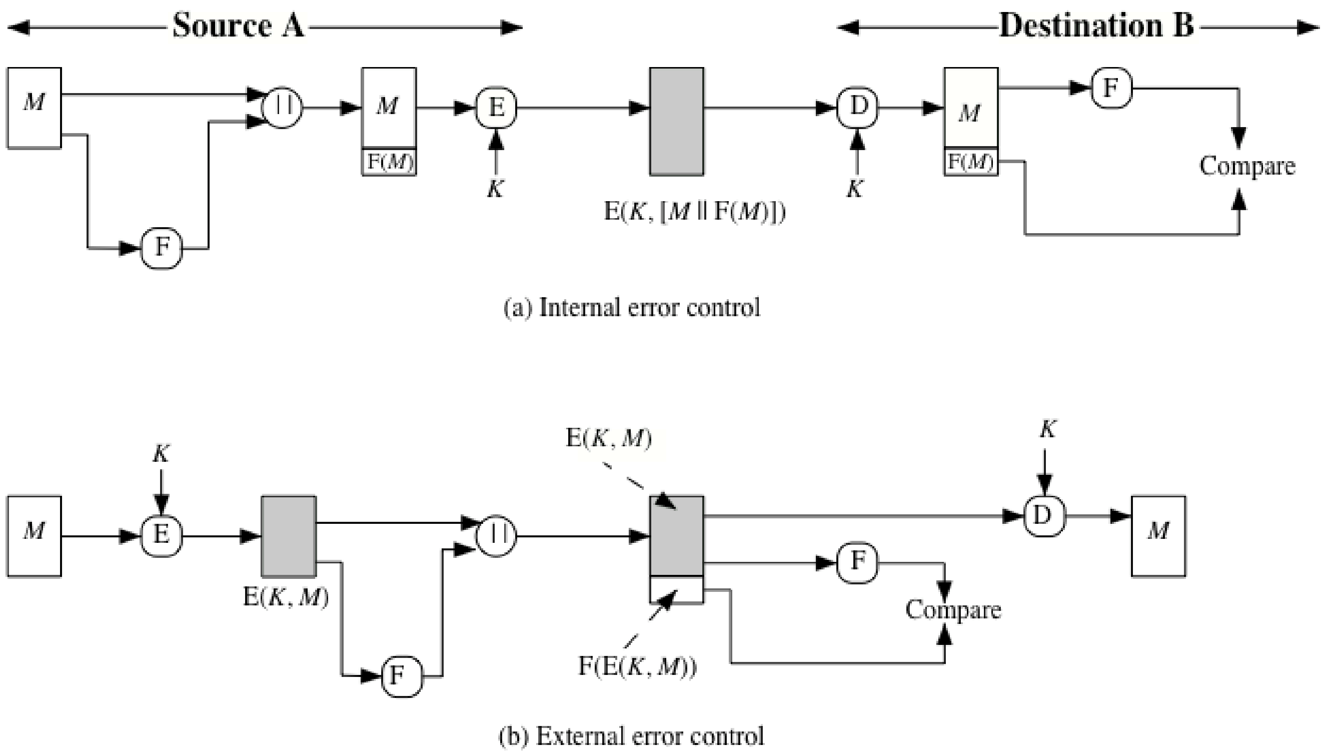


Figure 12.2 Internal and External Error Control

Fig 12.2 from the textbook

MAC use in Practice

- A pair TCP hosts shares a secret key and all exchanges between the hosts use the same key,
- Leads to simple encryptions between hosts-all IP packets between them can be encrypted except the header.

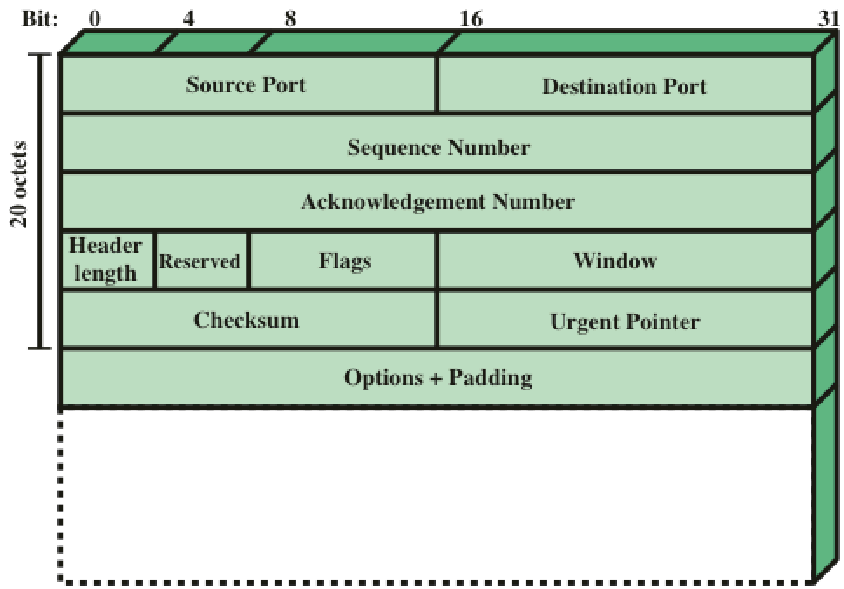


Figure 12.3 TCP Segment

Fig 12.3 from the textbook

Message Authentication Codes

- So formally, MAC is a dedicated symmetric key primitive aimed at providing authentication.
- With encryption it can be easily integerated to provide secrecy also.
- They are useful when in some applications you only need authentication.
- There are many situation where the property of authentication requires longer than confidentiality: authenticated sessions where only at times you may exchange secret information.
- MAC is different to Signatures,

Properties of MAC

- MAC has many properties similar to Hash.
- mac:= MAC(Key, message).
- You can treat it as a cryptographic checksum/digest: It takes a arbitrary length message as input and outputs a fixed length authenticator using a key.
- Like hash functions, it is many-to-one function with Preimage resistance (PR).
- For every key, it satisfies hash function properties.
- So sometimes, MAC is referred to as a family of Hash functions.

Attacks on MAC

- Brute-force attack: Here the objective is to find a collision.

- For cryptanalysis, there are two approaches:

- Attacker may first determine the key, then he can produce MAC value for any message.
- Sometimes, he may just try to determine a valid tag for a given message.

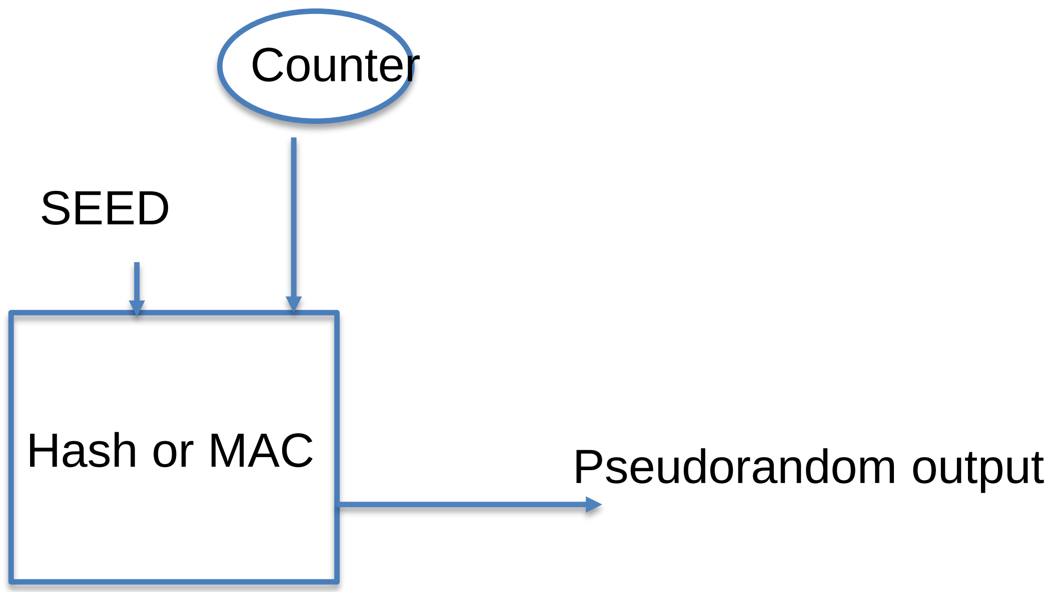
- Similar to Hash functions, you realize that MAC has to have a certain
- length to defeat brute-force attacks.
- In general you try to create new MAC functions using existing Hash functions.

MACs Based on Hash Functions: HMAC

- We do not study constructions of MAC in detail.
- It is sufficient to think of it as a keyed hash function.
- MAC based on Hash functions are popular in practice.
- A simple proposal:
- KeyedHash = Hash(Key ||Message)
- Some weaknesses were discovered using the simple proposal which led to development of HMAC.
- HMAC is thoroughly studied in literature. The textbook explains the concept with some detail. Please go through the discussion in Section 12.5 of the textbook.

Pseudorandom Generation

- As opposed to random numbers, pseudo random number generator takes a seed value as input and generates a sequence of digits.
- Like hash, for the same seed value it generates the same sequence.
- We briefly look at the topic and consider some Pseudorandom proposals based on hash and mac.



Week 6

Lecture 1

Un Keyed Cryptography: Hash Functions

Lecture 2

Message Authentication Codes or Keyed Hash Function

Workshop 3: Workshop based on Lectures in Week5

Quiz 6