CSC429 – Computer Security

LECTURE 3
MODERN CRYPTOGRAPHY 2

Mohammed H. Almeshekah, PhD
meshekah@ksu.edu.sa

Modern Cryptography

Beyond Confidentiality

More than confidentiality

- We have mainly been discussing the use of cryptography to provide data **confidentiality**.
- Other security services:
  - **Data integrity**
    - The assurance that data has not been altered in an unauthorized (or accidental) manner
  - **Data origin authentication**
    - The assurance that a given entity was the original source of some data (sometimes referred to as message authentication).
  - **Non-repudiation**
    - the assurance that an entity cannot deny any previous commitments or actions.

Hash Functions

Hash Functions

- Hash functions have many important and varied uses:
  - As strong one-way functions.
    - E.g. password storage.
  - To provide a weak notion of data integrity
  - As components to build other cryptographic primitives.
    - E.g. digital signatures.
  - As sources of pseudo-randomness.

What is a Hash Function?

- A hash function is a mathematical function which (generally):
  - does not have a key and is thus publicly computable.
  - has two practical properties.
  - has three security properties

Practical Properties of Hash Functions

1. Condenses arbitrary long inputs into a fixed length output.
   - The hash is a smaller thing that represents a larger thing, it sometimes referred to as a **digest**, and the hash function as a **message digest function**.
   - We refer to an n-bit hash function if the hash is n bits long.

2. Easy to compute.
   - A hash function should run in polynomial time.
   - Hash functions are expected to be faster than symmetric encryption.

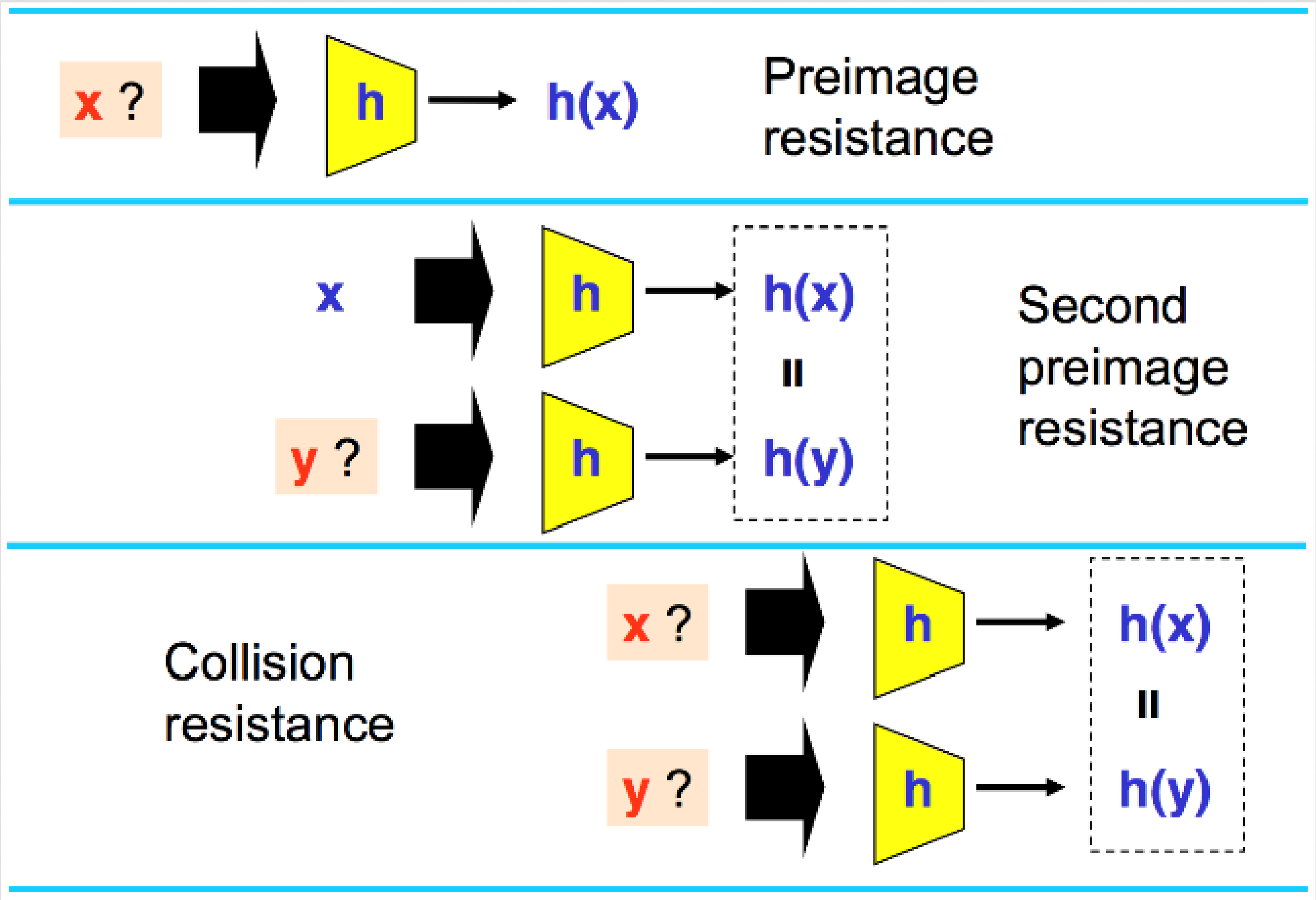
Security Properties of Hash Functions

1. pre-image resistant:
   - The hash function should be a one-way function:
     - Given x it is easy to compute h(x)
     - Given h(x) it is hard to determine x.

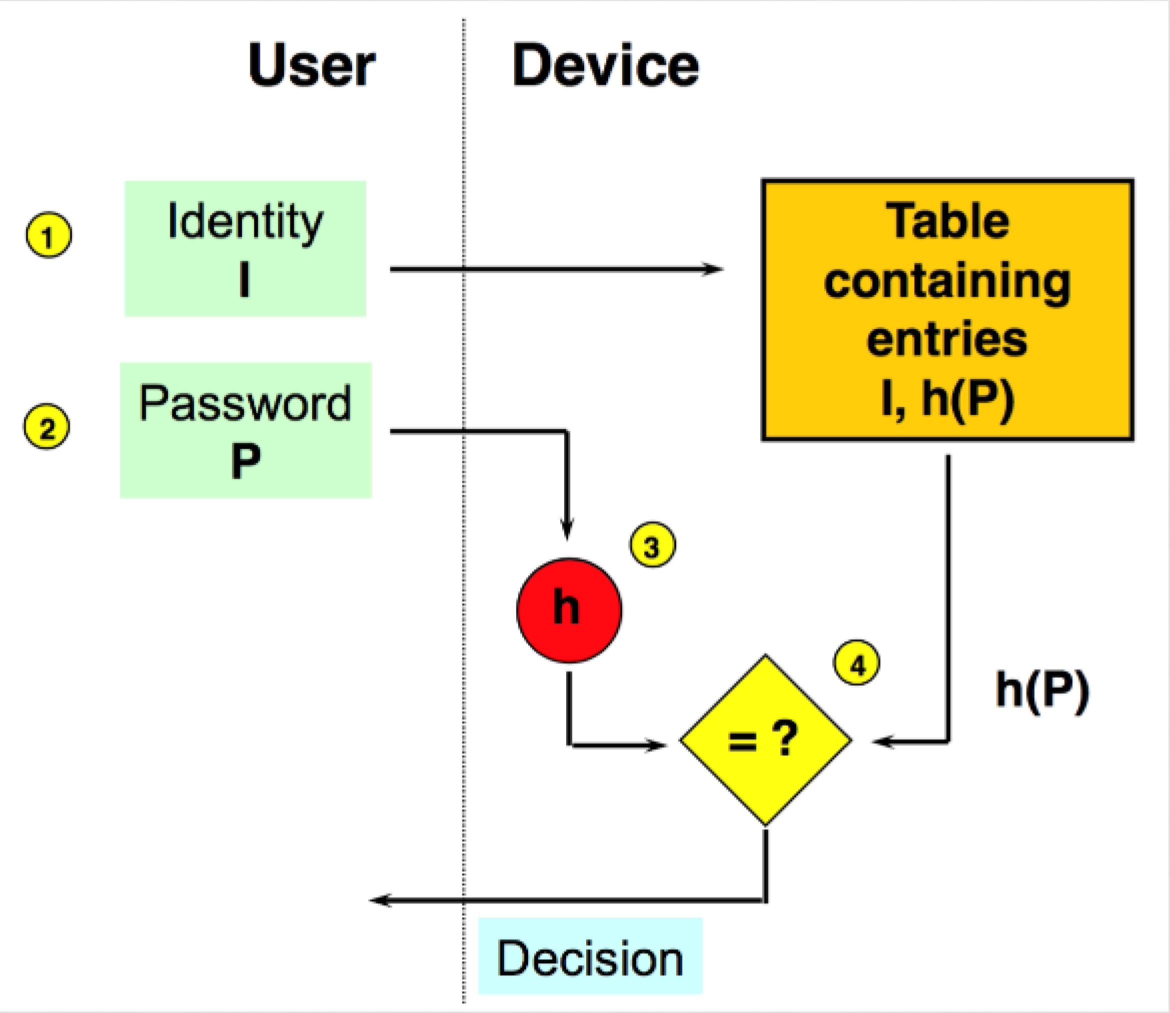
2. second pre-image resistance:
   - Given a message and its hash, it is hard to find a different message with that same hash.
   - Given x and h(x) it is hard to find y (different from x) such that h(x)=h(y)

1. collision-resistant:
   - It is hard to find any two messages with the same hash.
   - It is hard to find x and y (y different from x) such that h(x)=h(y).

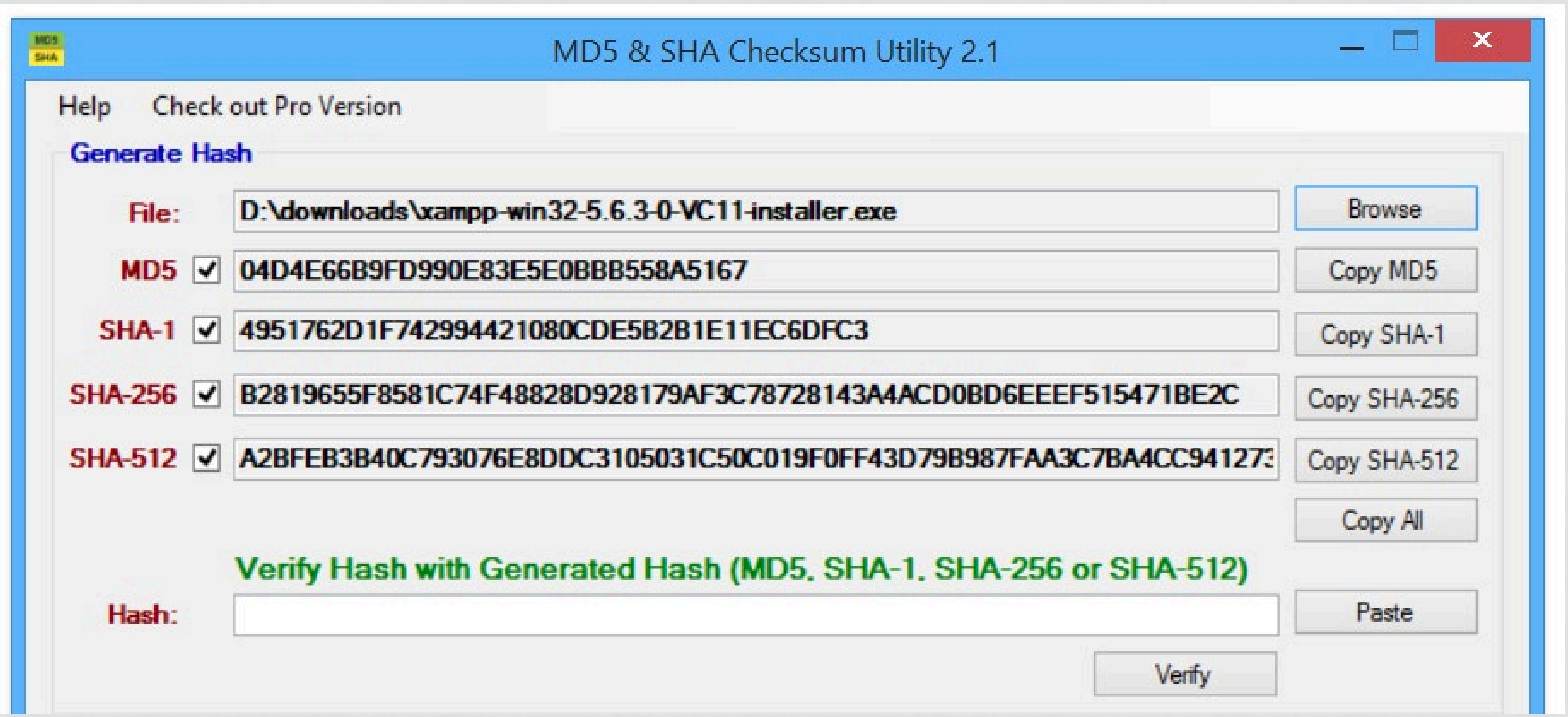
Summary of Security Properties



Pre-Image Resistance Example

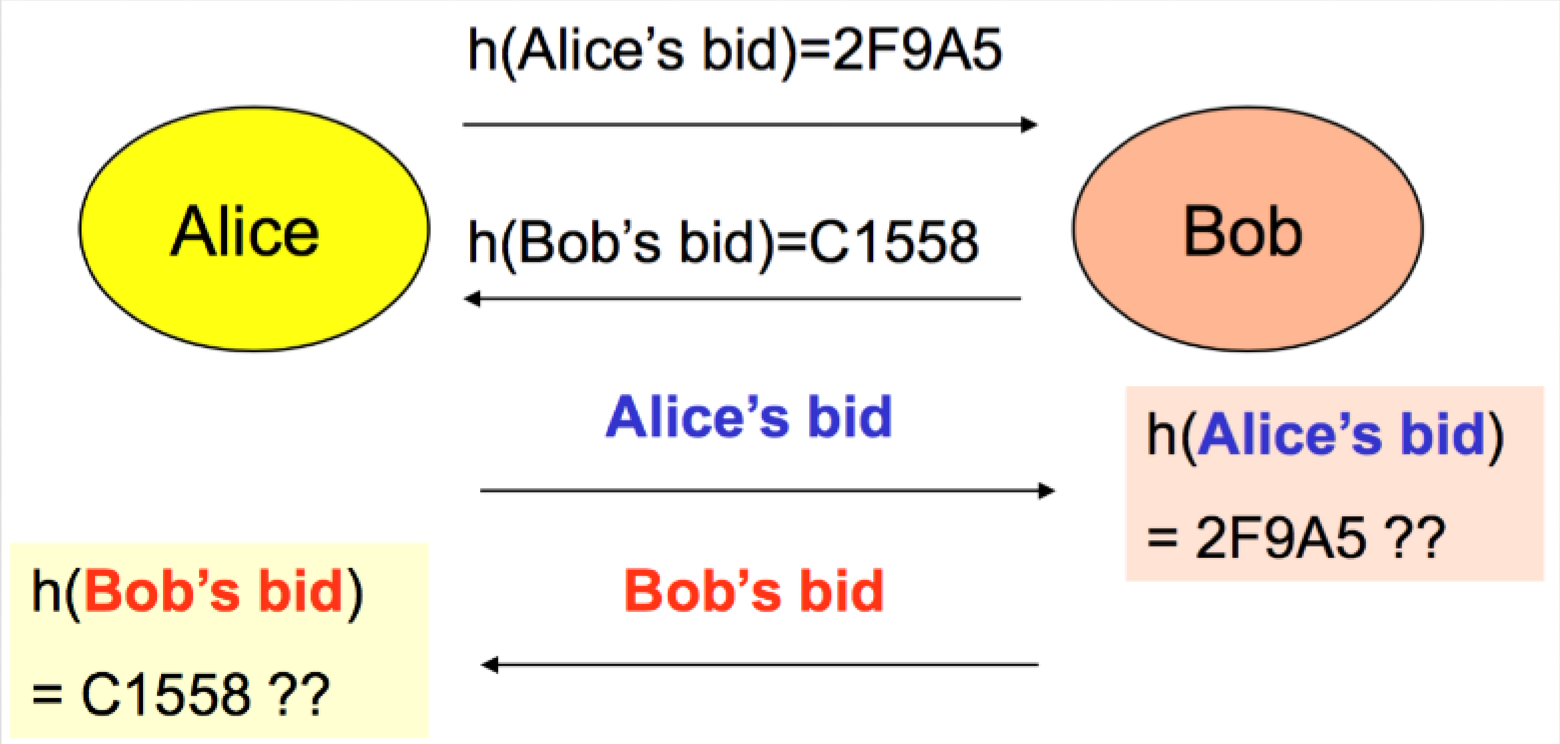


2nd Pre-Image Resistance Example



Collision Resistance Example

- Bidding for contracts openly.



Collisions

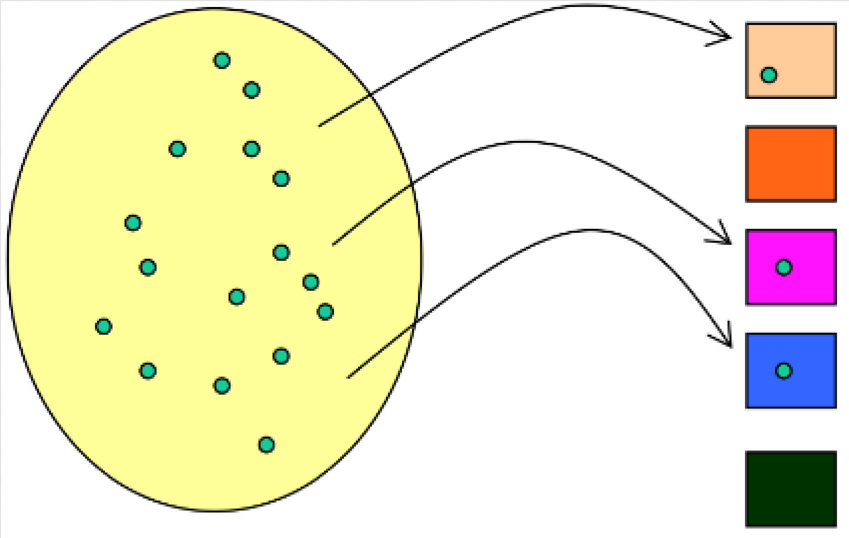
- Suppose we use a hash that is 10-bits output long.

Ali owes Me SAR 1000 Ali needs to pay me SAR 1000 Ali owes me SAR 1000 Ali owes me SAR 1,000 …..	Ali owes Me SAR 1,000,000 Ali should pay me 1,000,000 Ali owes Mohammed 1,000,000 Ali owes me 1000000 …..

- Can we cheat the system?

Birthday Attack on Hash Functions

- Consider an experiment where we take Q balls and start throwing them into M bins (where M is a smaller number than Q).



- After how many throws there is a greater than half chance that one bin contains two balls?

Birthday Attack

- A hash function with a 128-bit output will require **the square root of the length** = 2^{64} hashes to conduct a birthday attack.

- Because of the birthday attack, the length of hash outputs in general should double the key length of block ciphers
  - SHA-256, SHA-384, SHA-512 to match the new key lengths (128,192,256) in AES.

Practical Hash Functions

- MD5
  - 128 bit hash.
  - RFC1321, used for file integrity checking.
- SHA-1
  - 160-bit hash
  - Used in TLS/SSL, PGP, SSH, S/MIME, IPSec
- SHA-2:
  - (SHA224, SHA256, SHA384, SHA512).
- SHA-3:
  - Same sizes as SHA-2 family.

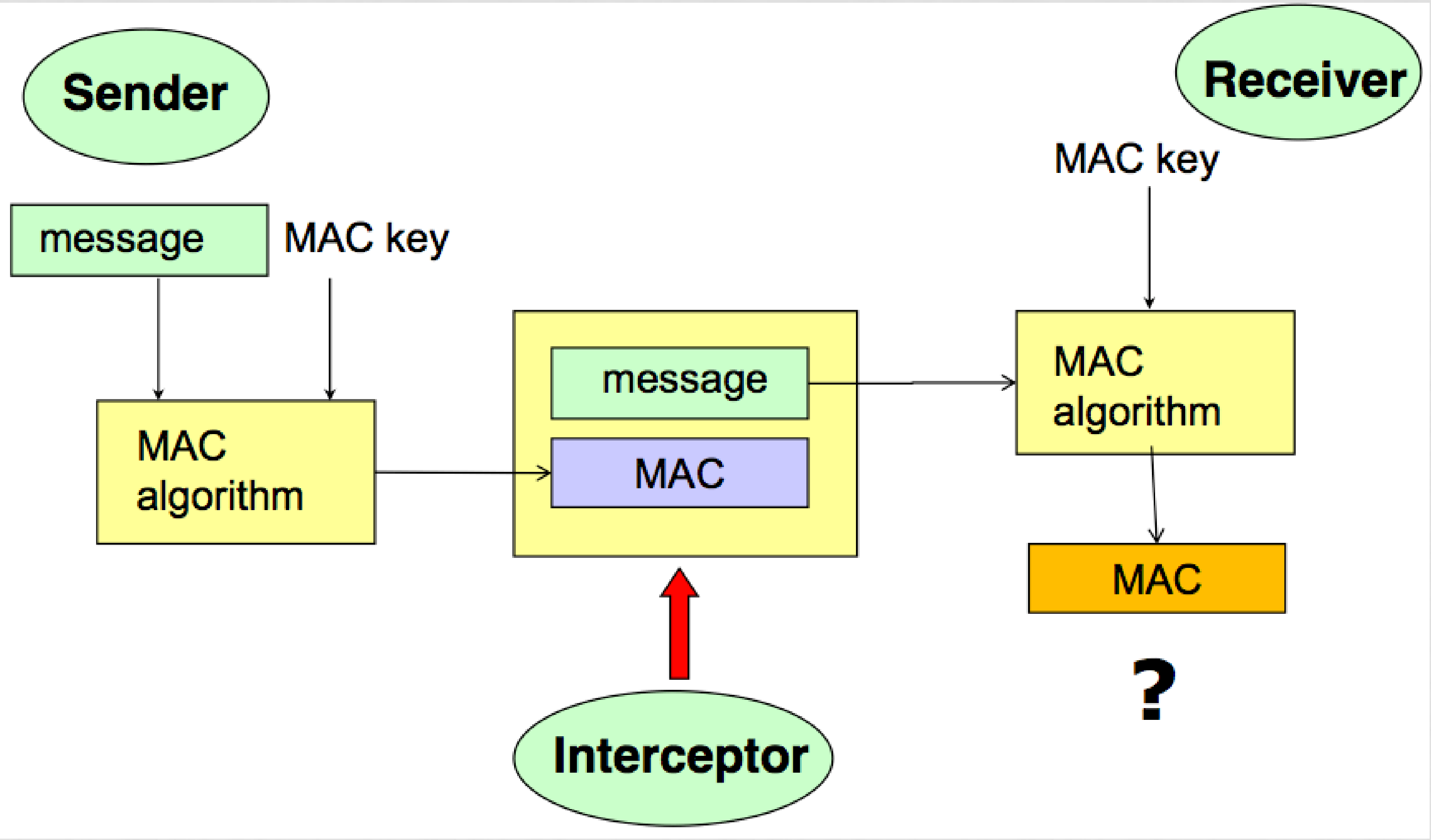
Message Authentication Code (MAC)

Integrity Protection using Hash Functions

- Are hash functions enough to protect the integrity of a message?
  - anyone can compute the hash value of a message, as the hash function is public

- Two solutions:
  - Message Authentication Codes (MAC).
  - Digital Signatures (discussed later).

Basic Model of a MAC



HMAC

- RFC 2104 describes how to convert a hash function into a MAC
- HMAC(message) = h(K || h(K || message))
  - K is a cryptographic key.

Next Lecture

- Asymmetric Cryptography
  - Public-Key Encryption.
  - Digital Signatures.
- Cryptographic Keys Establishment and Management.

- Readings for next lecture:
  - Anderson's book – (5.3.4), (5.3.5), (5.7.1), (5.7.2), (3.7) and (21.4.5.7)