

# Cryptography CS 555

## Lecture 9



Department of Computer Sciences  
Purdue University

# About the Homework

- For problem 1, assume the following:
  - if one bit in one 64-bit block of ciphertext is corrupted, then every bit in the decrypted 64-bit plaintext block *may* be wrong

# Review of Hash Functions

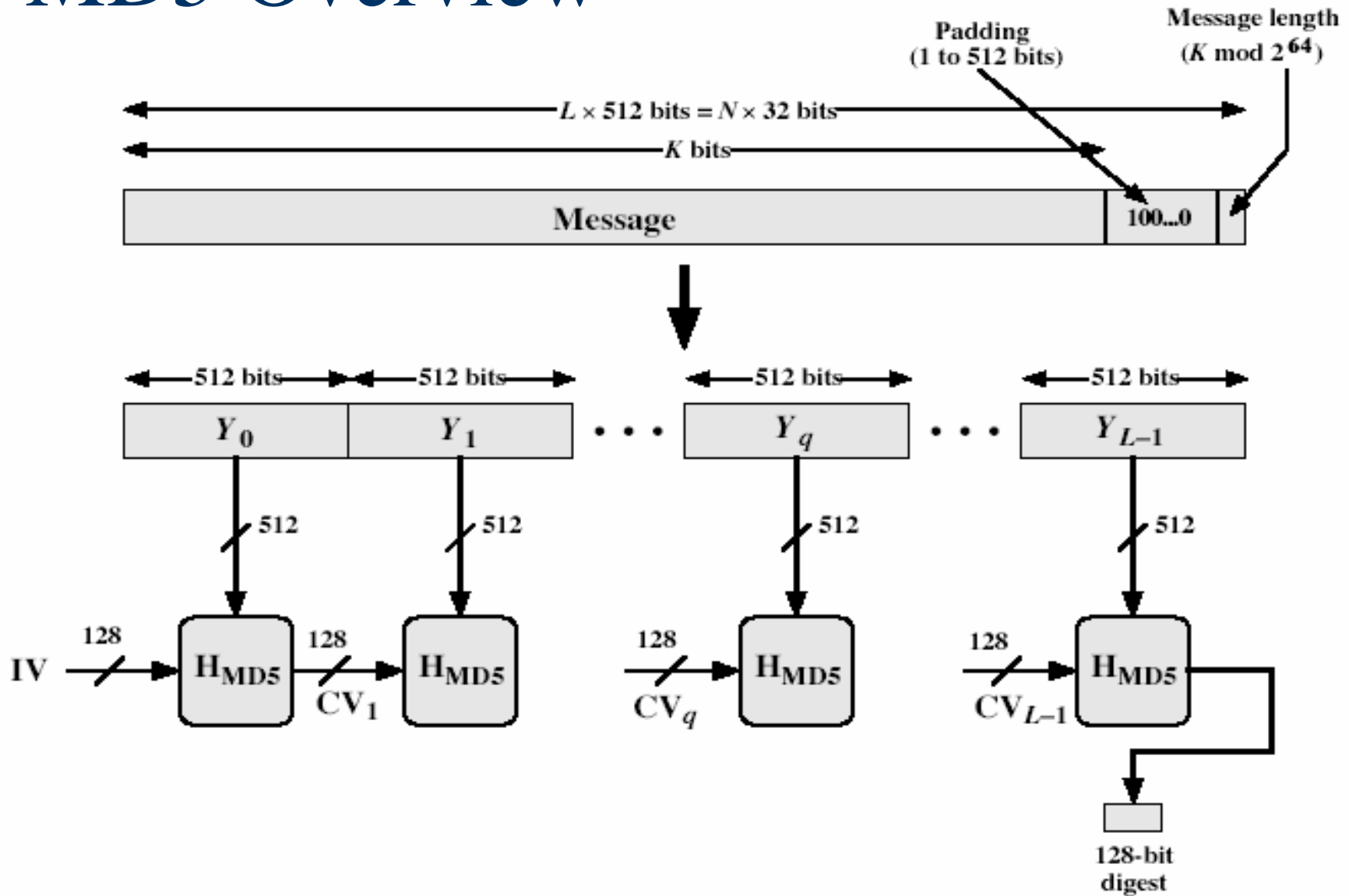
- A hash function  $h: \{0,1\}^* \rightarrow \{0,1\}^m$
- Constructs a short “fingerprint” of an arbitrarily long message
- Given a message  $M$ ,  $h(M)$  is often referred to as a message digest

# Requirements for Cryptographic Hash Functions

Given a function  $h: X \rightarrow Y$ , then we say that  $h$  has:

- **preimage resistance (one-way):**  
if given  $y \in Y$  it is computationally infeasible to find a value  $x \in X$  s.t.  $h(x) = y$
- **2-nd preimage resistance (weak collision resistance):**  
if given  $x \in X$  it is computationally infeasible to find a value  $x' \in X$ ,  $x' \neq x$  s.t.  $h(x') = h(x)$
- **collision resistance (strong collision resistance):**  
if it is computationally infeasible to find any two distinct values  $x', x \in X$ , s.t.  $h(x') = h(x)$

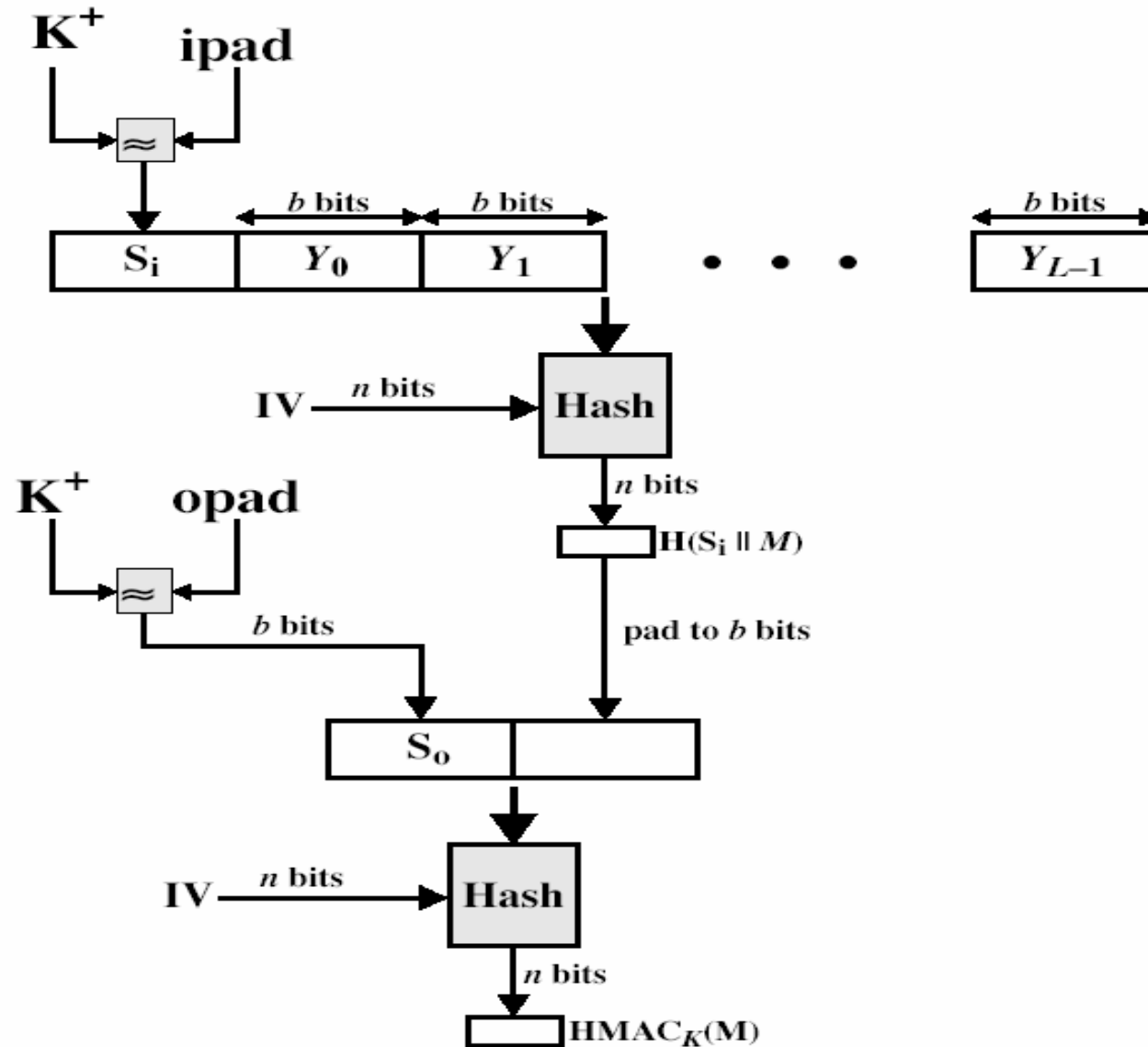
# MD5 Overview



# Review of MAC

- $MAC = C_k(M)$ 
  - M: input message
  - C: MAC function
  - k: shared secret key between sender and receiver
  - MAC: message authentication code

# HMAC Overview



# Lecture Outline

- More precise security definitions for security properties of hash
- Merkle-Damgard construction
- MAC Security
- Commitment schemes



# More Precise Definitions

- A hash function  $h$  is  $(t, \varepsilon)$  **one-way** if there exists no  $t$ -time probabilistic algorithm  $A$   
 $h(A(y)) = y$  with probability  $> \varepsilon$ 
  - probability taken over random  $y$  and internal random
- A hash function  $h$  is  $(t, \varepsilon)$  **weak collision** resistant if there exists no  $t$ -time probabilistic algorithm  $A$  such that when given  $x$ , with probability  $> \varepsilon$ , it outputs  $x'$  such that  $x' \neq x$  and  $h(x') = h(x)$

# More Precise Definitions

- A hash function  $h$  is  **$(t, \epsilon)$  collision resistant** if there exists no  $t$ -time probabilistic algorithm that outputs two messages  $x_1$  and  $x_2$  such that  $h(x_1) = h(x_2)$  with probability  $> \epsilon$

# Reduction among the security properties

- $(t+c, \varepsilon)$  collision resistant implies  $(t, \varepsilon)$  weak-collision resistant, where  $c$  is a small constant
- Proof idea:
  - Suppose that  $h$  is not  $(t, \varepsilon)$  weak-collision resistant, then there exists algorithm  $A$ , when given  $x$ , outputs  $x'=A(x)$  such that  $h(x')=y$ .
  - Construct  $B$  as follows,  $B$  picks a random  $x$ , feeds it to  $A$ , and then outputs  $(A(x), x)$ .

# Reduction among the security properties

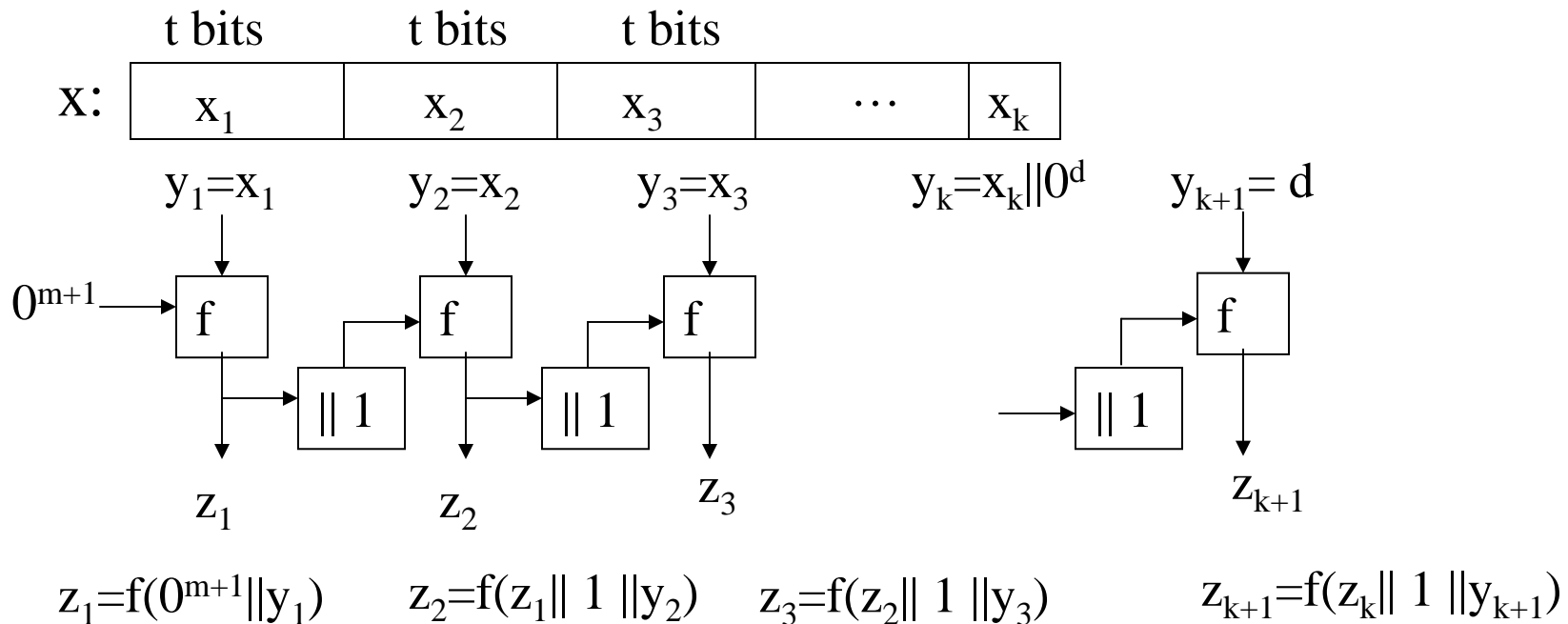
- Roughly speaking, collision-resistant implies one-way
  - proof & precise parameters in Stinson (4.2.3)

# Understanding the security of iterative hash functions

- Compression function:  
 $f : \{0,1\}^{m+t} \rightarrow \{0,1\}^m$
- Takes a fixed-length input string and output a shorter string
- The following properties can be defined similarly as hash functions
  - preimage resistance (one-way):
  - 2-nd preimage resistance (weak collision resistance):
  - collision resistance (strong collision resistance):

# The Merkle-Damgard Construction of Hash Functions

- Goal: construct a hash function  $h: \{0,1\}^* \rightarrow \{0,1\}^m$  from a compression function  $f: \{0,1\}^{m+t+1} \rightarrow \{0,1\}^m$
- Given message  $x$  of arbitrary length



# Example:

- Compression function:  $f: \{0,1\}^{128+512+1} \rightarrow \{0,1\}^{128}$
- Message  $x$  has 1000 bits:
  - $y_1$  is first 512 bits of  $x$
  - $y_2$  is last 488 bits of  $x \parallel 0^{24}$
  - $y_3$  is 32-bit binary representation of 24  $\parallel 0^{480}$
  - $z_1 = f(0^{129} \parallel y_1)$        $z_1$  has 128 bits
  - $z_2 = f(z_1 \parallel 1 \parallel y_2)$
  - $z_3 = f(z_2 \parallel 1 \parallel y_3)$        $z_3$  is the message digest  $h(x)$

# Example:

- Suppose that message  $x'$  has 488 bits and  $h(x)=h(x')$ :
  - $y_1'$  is  $x' \parallel 0^{24}$
  - $y_2'$  is 32-bit binary representation of  $24 \parallel 0^{480}$
  - $z_1' = f(0^{129} \parallel y_1')$        $z_1$  has 128 bits
  - $z_2' = f(z_1' \parallel 1 \parallel y_2')$        $z_2'$  is  $h(x)$
- Then  $f(z_1' \parallel 1 \parallel y_2') = f(z_2 \parallel 1 \parallel y_3)$  and  $y_3=y_2'$ 
  - if  $z_1' \neq z_2$  then a collision is found for  $f$
  - if  $z_1' = z_2$  then  $f(0^{129} \parallel y_1') = f(z_1 \parallel 1 \parallel y_2)$ , there is also a collision for  $f$

# Security of the Merkle-Damgard Construction

- If  $f: \{0,1\}^{m+t+1} \rightarrow \{0,1\}^m$  is collision resistant, then the Merkle-Damgard construction  $h: \{0,1\}^* \rightarrow \{0,1\}^m$  is collision resistant.
- Proof:
  - suppose that we can find  $x \neq x'$  such that  $h(x) = h(x')$ , we show that we can find collision on  $f$
  - let  $y(x) = y_1 \parallel y_2 \parallel \dots \parallel y_{k+1}$
  - let  $z_1, z_2, \dots, z_{k+1}$  be the intermediate results of  $h(x)$ , then  $h(x) = z_{k+1} = f(z_k \parallel 1 \parallel y_{k+1})$
  - let  $y(x') = y_1' \parallel y_2' \parallel \dots \parallel y_{n+1}'$  and  $z_1', z_2', \dots, z_{n+1}'$  be the intermediate results of  $h(x')$ , then
$$f(z_k \parallel 1 \parallel y_{k+1}) = h(x') = z_{n+1}' = f(z_n' \parallel 1 \parallel y_{n+1}')$$

# Security of the Merkle-Damgard Construction (Proof continued)

$$f(z_k \parallel 1 \parallel y_{k+1}) = f(z_n' \parallel 1 \parallel y_{n+1}')$$

- Case 1:  $|x| \neq |x'| \pmod t$  (the number of padding bits are different), then  $y_{k+1} \neq y_{n+1}'$ , a collision has been found
- Case 2a:  $|x| = |x'|$ , then  $k=n$ , either  $z_k \neq z_k'$ , in which case a collision has been found, or  $z_k = z_k'$ , in which case
$$f(z_{k-1} \parallel 1 \parallel y_k) = z_k = z_k' = f(z_{k-1}' \parallel 1 \parallel y_k')$$
if  $y_k \neq y_k'$ , then a collision has been found; otherwise consider  $z_{k-1}$  and  $z_{k-1}'$ , if they are different, a collision has been found, otherwise go backwards. There must exist an number  $j$  such that  $y_j \neq y_j'$ .

# Security of the Merkle-Damgard Construction (Proof continued)

- Case 2b:  $|x| \neq |x'|$ . Similar to case (2a), except that we may go all the way back to the beginning of one of the strings and have

$$f(0^{m+1} \parallel y_1) = f(z_j' \parallel 1 \parallel y_{j+1}')$$

A collision has been found.

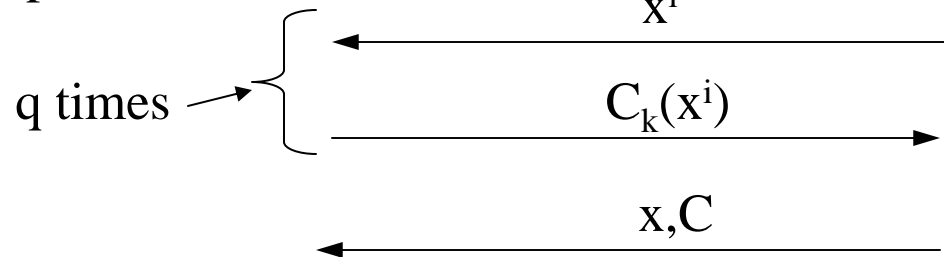
# The MAC Security Game

- Let  $C$  be a MAC function  $C_k(M)$  is the MAC for  $M$  under  $k$

Challenger

Attacker

1. picks random  $k$



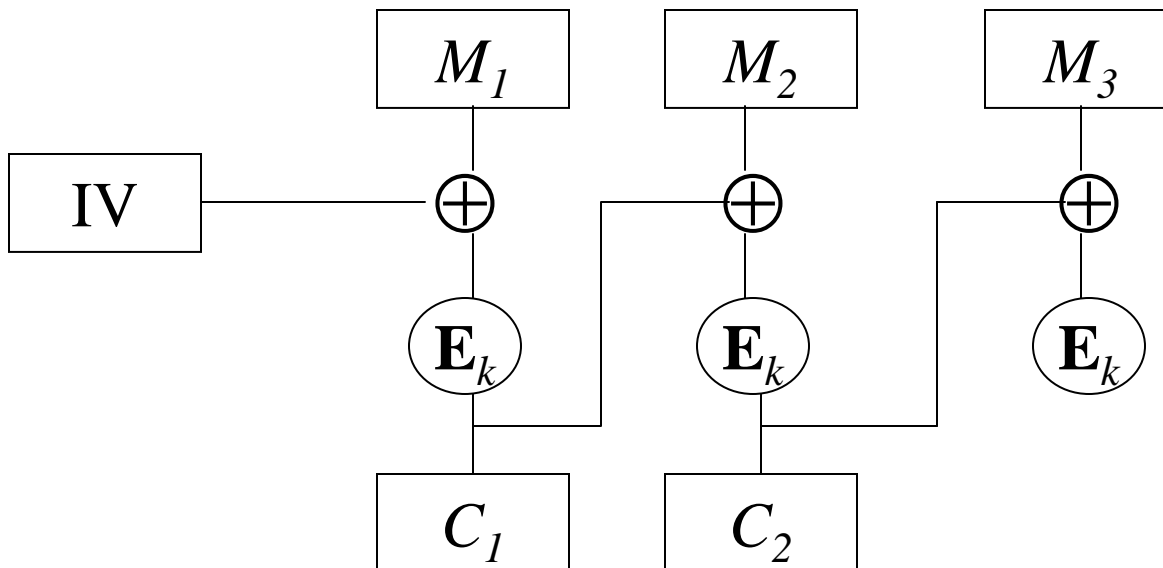
Attacker wins game if  $C_k(x)=C$

# Incorrect MAC scheme based on Collision-resistant Hash function

- MAC:  $C_k(x) = h(k || x)$
- Suppose that the hash function is constructed in the iterated fashion, further suppose that no padding is added, i.e.,  $h(x||x')=f(h(x)||x')$ .
- Then given  $x$  whose length is a multiple of the block size and  $C_k(x)$ , then one can construct  $y=x||x'$  and compute its  $C_k(y)=f(C_k(x),x')$ .

# Encryption Modes: CBC

- **Cipher Block Chaining (CBC):** next input depends of previous output
  - Plaintext is  $M_1, M_2, M_3, M_4,$
  - Ciphertext is:  $C_1 = IV \oplus \mathbf{E}_k(M_1)$        $C_2 = C_1 \oplus \mathbf{E}_k(M_2)$   
 $C_3 = C_2 \oplus \mathbf{E}_k(M_3)$        $C_4 = C_3 \oplus \mathbf{E}_k(M_4)$



# CBC-MAC

- Given a block cipher  $\mathbf{E}$  with block size  $m$
- Given message  $M = M_1 || M_2 || \dots || M_n$
- MAC of  $M$  is  $\mathbf{E}_k(M)$
- $z_0 = 0^m$
- $z_i = \mathbf{E}_k(z_{i-1} \oplus M_i)$  for  $1 \leq i \leq n$
- $\text{MAC} = z_n$
- Provably secure assuming that  $\mathbf{E}$  is Pseudorandom Permutation
- Subject to birthday attack

# Commitment schemes

- An electronic way to temporarily hide a value that cannot be changed
  - Stage 1 (Commit)
    - Sender locks a message in a box and sends the locked box to another party called the Receiver
  - State 2 (Reveal)
    - the Sender proves to the Receiver that the message in the box is a certain message

# Types of commitment

- Bit commitment
  - the committed value is one bit
- String commitment
  - the committed value is a string

# Security properties of commitment schemes

- Hiding
  - at the end of Stage 1, no adversary receiver learns information about the committed value
- Binding
  - at the end of State 1, no adversary sender can successfully convince reveal two different values in Stage 2

# A broken commitment scheme

- Using encryption
  - Stage 1 (Commit)
    - the Sender generates a key  $k$  and sends  $E_k[M]$  to the Receiver
  - State 2 (Reveal)
    - the Sender sends  $k$  to the Receiver, the Receiver can decrypt the message
- What is wrong using the above as a commitment scheme?

# Formalizing Security Properties of Commitment schemes

- Two kinds of adversaries
  - those with infinite computation power and those with limited computation power
- Unconditional hiding
  - the commitment phase does not leak any information about the committed message, in the information theoretical sense (similar to perfect secrecy)
- Computational hiding
  - an adversary with limited computation power cannot learn anything about the committed message (similar to semantic security)

# Formalizing Security Properties of Commitment schemes

- Unconditional binding
  - after the commitment phase, an infinite powerful adversary sender cannot reveal two different values
- Computational binding
  - after the commitment phase, an adversary with limited computation power cannot reveal two different values
- No commitment scheme can be both unconditional hiding and unconditional binding

# Another (also broken) commitment scheme

- Using a collision-resistant hash function  $h$ 
  - Stage 1 (Commit)
    - the Sender sends  $c=h(M)$  to the Receiver
  - State 2 (Reveal)
    - the Sender sends  $M$  to the Receiver, the Receiver verifies that  $c=h(M)$
- What is wrong using this as a commitment scheme?

# A third scheme

- Using a collision-resistant hash function  $h$ 
  - Stage 1 (Commit)
    - the Sender sends  $(r_1, c=h(r_1, r_2, M))$  to the Receiver
  - State 2 (Reveal)
    - the Sender sends  $r_2, M$  to the Receiver, the Receiver verifies that  $c=h(r_1, r_2, M)$
- Security not provable based on collision-resistant hash functions, may be okay in practice

# Review of important concepts for Hash Functions and MAC

- Different security properties of cryptographic hash functions and their relationship
- Merkle-Damgard construction, and its security proof
- Parameters and basic structures of MD5 and SHA-1
- Birthday attacks & choosing length of hash functions
- Concept and security definitions of MAC
- Weakness of certain MAC constructions from collision-resistant Hash functions
- HMAC structure and security properties (no proof required)

# Next Lectures..

- Number theory
- Readings:
  - Stallings: Chapter 8
  - Stinson: 5.1, 5.2