# Introduction to Cryptography and Information Security
# UEE4611, Spring Semester 2020

Ching-Yi Lai
Institute of Communications Engineering
National Chiao Tung University

Chapter 12: Message Authentication Codes

- Message Authentication Requirements
- Message Authentication Functions
- Requirements for Message Authentication Codes
- Security of MACs
- MACs Based on Hash Functions: HMAC
- MACs Based on Block Ciphers: ~~DAA~~ and CMAC
- ~~Authenticated Encryption: CCM and GCM~~
- Key Wrapping
- Pseudorandom Number Generation Using Hash Functions and MACs

# Message Authentication

① Hash function

② Message Encryption

Ex. $\{0,1\}^n$ $H$: $r \times n$ matrix.

$M = \{ m \in \{0,1\}^n : Hm = 0 \}$ Error-correcting codes

③ Message Authentication code (MAC)

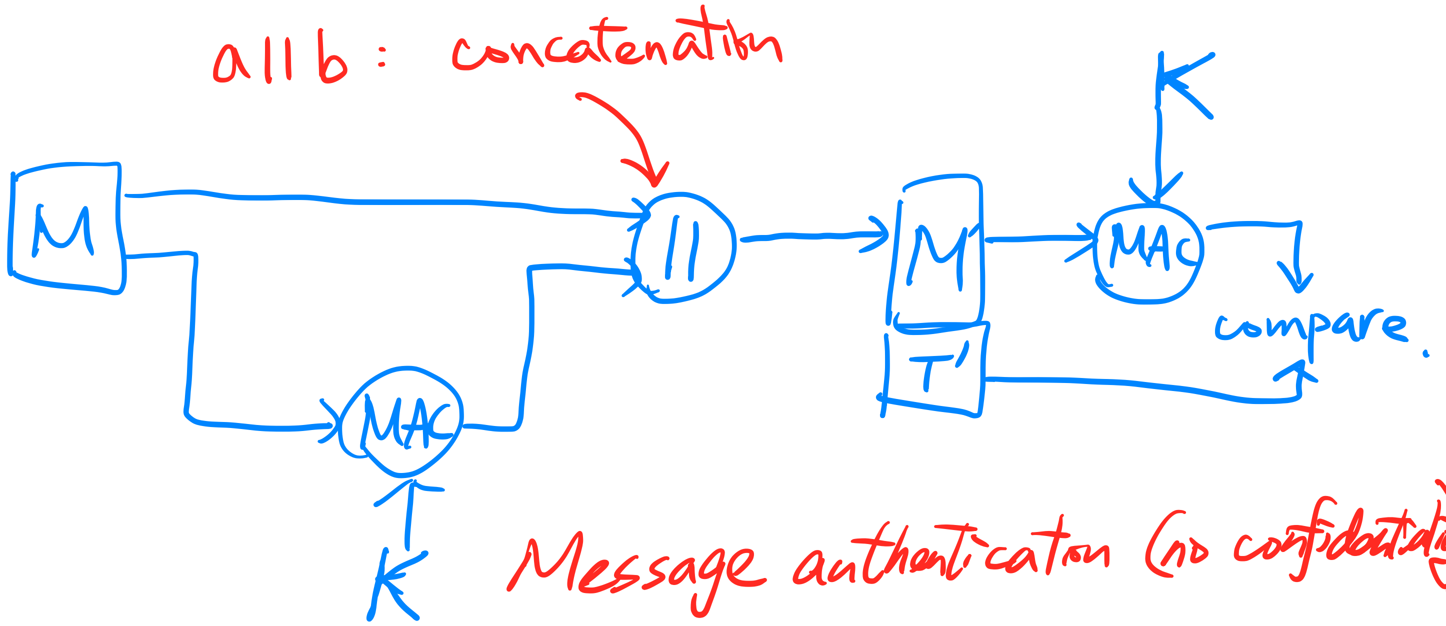
Assume that two parties A & B share a common secret key $k$.

$$T = MAC(k, M)$$

where MAC: many-to-one function

$M$ is message (variable length)

$T$: tag. (fixed length)




Message authentication (no confidentiality)

---

Symmetric Encryption

k-bit key

$\Rightarrow$ $2^k$ possibilities

MAC

k-bit key

n-bit tag (MAC)

($k > n$).

$\sim 2^k/2^n$ keys will generate a match from a given message/tag pair.

$\Rightarrow$ have to iterate the attack on several known message/tag pairs.

A & B share K.

A sends $x_1$, MAC($k$, $x_1$) to B

⋮

$x_n$, MAC($k$, $x_n$)

$\boxed{\text{Eve}}$ wants to construct $x \neq x_i$, $i = 1, \ldots, n$.

(doesn't know K) & $t = \text{MAC}(k, x)$.

(Attack)

Round 1.

Given $x_1$, $t_1 = MAC(k, x_1)$

Compare $t_1$ with $MAC(k', x_1)$ for all $k' \in \{0,1\}^k$

$\approx 2^{k-n}$ matches

Round 2.

Given $x_2$, $t_2 = MAC(k, x_2)$

Compare $t_2$ with $MAC(k'', x_2)$ for all $k''$ left in the previous round.

$\approx 2^{k-2n}$ matches

repeat $\approx \lceil k/n \rceil$ rounds

Ex. $k=80$, $n=32$. roughly three rounds are required.

Requirements.

1. Given $x_i$ & $MAC(k, x_i)$, It is computationally infeasible to find $x \neq x_i \ni MAC(k, x) = MAC(k, x_i)$
2. $MAC(k, x)$ should be uniformly distributed in $x$. That is $Pr(MAC(k,x) = MAC(k,x')) = 2^{-n}$

# HMAC : MAC based on hash functions

1. to use available hash functions without modification
2. easy replaceability of the embedded hash function
3. preserve the original performances of hash functions
4. well-understood cryptanalysis

---

H: embedded hash function

$M$ = message input

$= Y_0 \| Y_1 \| \cdots \| Y_{L-1}$

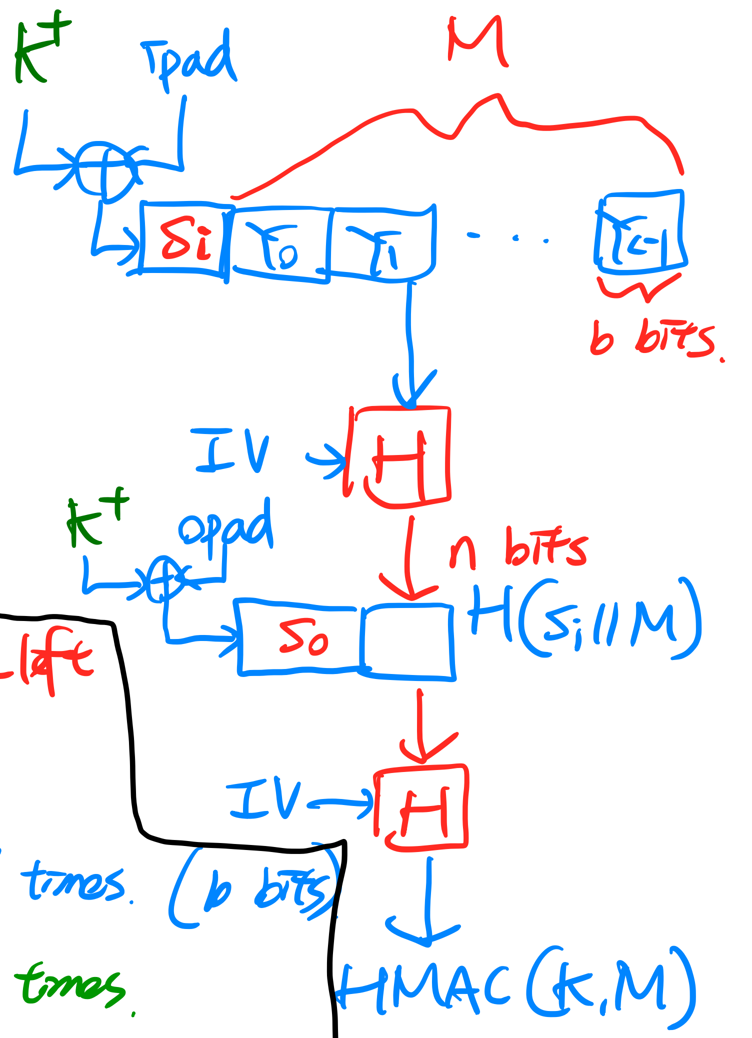
where $Y_i$ of $b$ bits

K = secret key of length $\geq n$

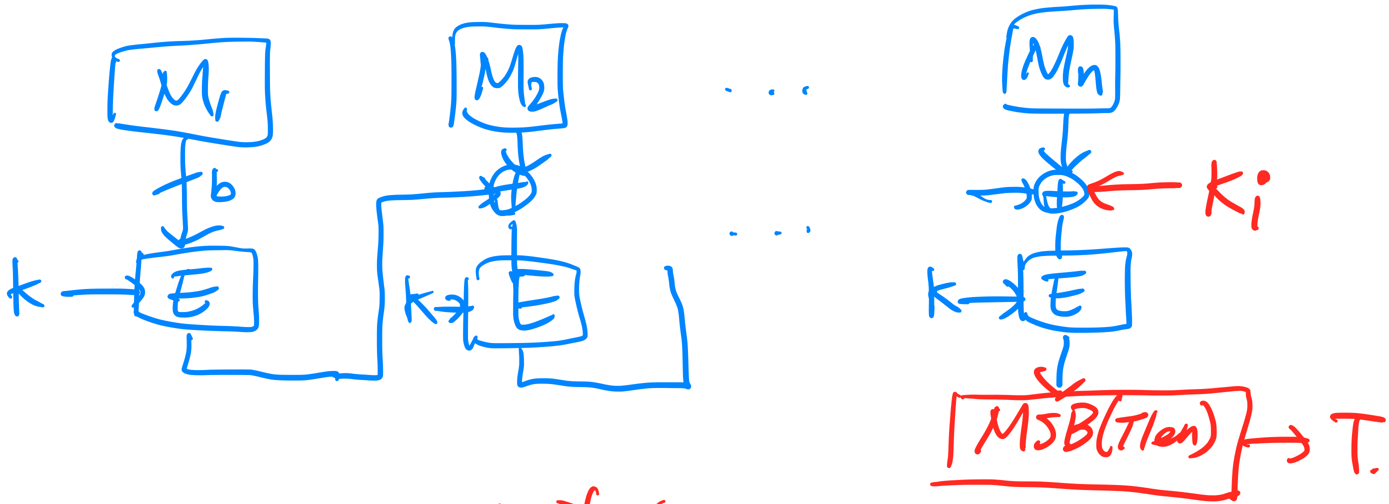
$K^+$: K padded with zeros on the left to b-bit long.

ipad = $\underbrace{0011}_{3}\underbrace{0110}_{6}$ repeated $b/8$ times. ($b$ bits)

opad = 0101 1100 repeated $b/8$ times.



# Cipher-based MAC (CMAC)




Tlen : length of tag

$$L = E(K, \underbrace{0^b}_{00\cdots0 \text{ (b times)}})$$

$$K_i = \begin{cases} L \cdot \underset{00\cdots010}{x}, & \text{if Message length is integer multiple of } b. \\ L \cdot \underset{00\cdots0100}{x^2}, & \text{otherwise, with } M_n \text{ zero-padded.} \end{cases}$$

multiplication in $GF(2^b)$

# PRNG based on Hash functions & MACs.