IS511 Introduction to Information Security

Lecture 3 Cryptography 2

Yongdae Kim

KAIST

Recap

- http://syssec.kaist.ac.kr/~yongdaek/courses/is511/
- E-mail policy
  - Include [is511]
  - Profs + TA: IS511_prof@gsis.kaist.ac.kr
  - Profs + TA + Students: IS511_student@gsis.kaist.ac.kr
- Text only posting, email!
- Preproposal
- Proposal: English only

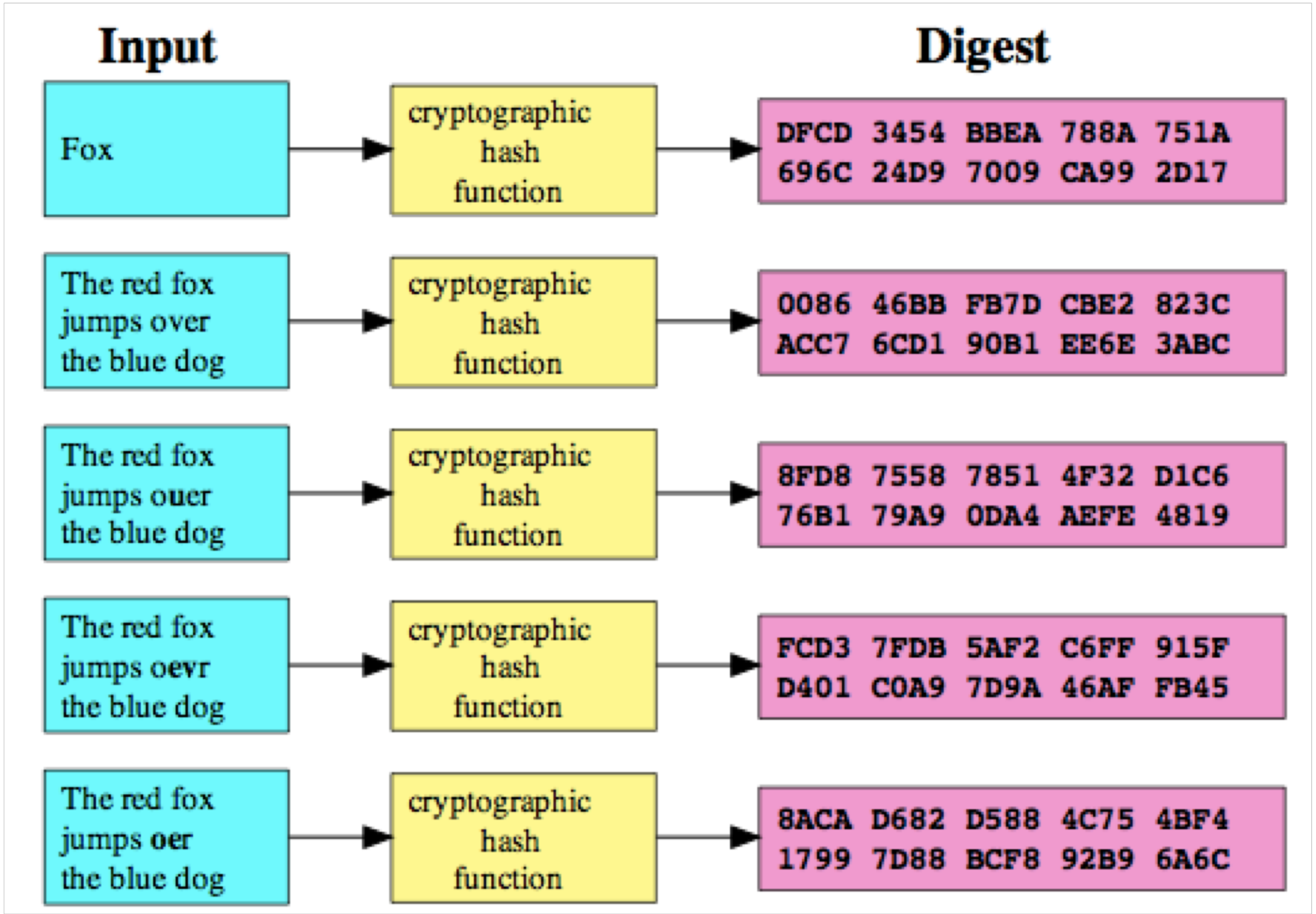
Hash function and MAC

- A hash function is a function h
  - compression
  - ease of computation
  - Properties
    - one-way: for a given y, find x' such that h(x') = y
    - collision resistance: find x and x' such that h(x) = h(x')
  - Examples: SHA-1, MD-5

- MAC (message authentication codes)
  - both authentication and integrity
  - MAC is a family of functions h_k
    - ease of computation (if k is known !!)
    - compression, x is of arbitrary length, $h_k(x)$ has fixed length
    - computation resistance
  - Example: HMAC

KAIST

How Random is the Hash function?



KAIST

Applications of Hash Function

- File integrity

Instructions

The Windows SDK is available as a DVD ISO image file so that you can bu
that you are downloading the correct ISO file, please refer to the table bel
to validate that the file you've downloaded is the correct file.

==

File Name: GRMSDK_EN_DVD.iso

Chip: X86

CRC#: 0xCA4FE79D

SHA1: 0x8695F5E6810D84153181695DA78850988A923F4E

- Digital signature

  Sign = $S_{SK}(h(m))$

- Password verification

stored hash = h(password)

- File identifier
- Hash table
- Generating random numbers

Hash function and MAC

- A hash function is a function h
  - compression
  - ease of computation
  - Properties
    - one-way: for a given y, find x' such that h(x') = y
    - collision resistance: find x and x' such that h(x) = h(x')
  - Examples: SHA-1, MD-5
- MAC (message authentication codes)
  - both authentication and integrity
  - MAC is a family of functions h_k
    - ease of computation (if k is known !!)
    - compression, x is of arbitrary length, $h_k(x)$ has fixed length
    - computation resistance
  - Example: HMAC

MAC construction from Hash

- Prefix
  - M=h(k||x)
  - appending y and deducing h(k||x||y) form h(k||x) without knowing k
- Suffix
  - M=h(x||k)
  - possible a birthday attack, an adversary that can choose x can construct x' for which h(x)=h(x') in $O(2^{n/2})$
- STATE OF THE ART: HMAC (RFC 2104)
  - HMAC(x)=h(k||p_1||h(k|| p_2||x)), p1 and p2 are padding
  - The outer hash operates on an input of two blocks
  - Provably secure

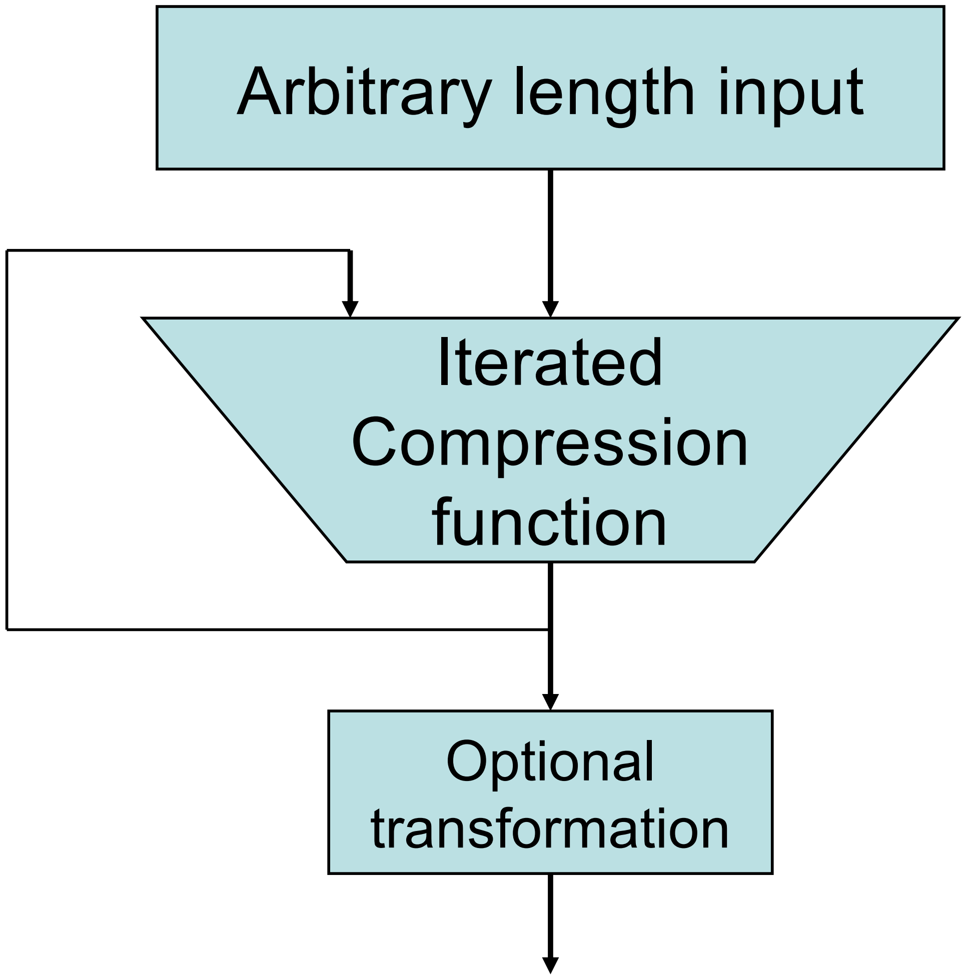
How to use MAC?

- A & B share a secret key k
- A sends the message x and the MAC $M \leftarrow H_k(x)$
- B receives x and M from A
- B computes $H_k(x)$ with received M
- B checks if $M = H_k(x)$

How to design a hash function

- Phase 1: Design a 'compression function'
  - Which compresses only a single block of fixed size to a previous state variable
- Phase 2: 'Combine' the action of the compression function to process messages of arbitrary lengths
- Similar to the case of encryption schemes

General Model



MDC h with compression function f:

H_0=IV, H_i=f(H_{i-1}, x_i), h(x)= H_t

Basic properties

- *preimage resistance = one-way*
  - it is computationally infeasible to find any input which hashes to that output
  - for a given y, find x' such that h(x') = y
- *2nd-preimage resistance = weak collision resistance*
  - it is computationally infeasible to find any second input which has the same output as any specified input
  - for a given x, find x' such that h(x') = h(x)
- *collision resistance = strong collision resistance*
  - it is computationally infeasible to find any two distinct inputs x, x' which hash to the same output
  - find x and x' such that h(x) = h(x').

KAIST

Relation between properties

- Collision resistance $\Rightarrow$ Weak collision resistance ?
  - Yes! Why?

- Collision resistance $\Rightarrow$ One-way ?
  - No! Why?
  - Let g collision resistant hash function, $g: \{0,1\}^* \rightarrow \{0,1\}^n$
  - Consider the function h defined as

  $h(x) = 1 \| x$ if x has bit length n

  $= 0 \| g(x)$ otherwise

  $h: \{0,1\}^* \rightarrow \{0,1\}^{n+1}$
  - h(x) : collision and pre-image resistant (unique), but not one-way

Birthday Paradox (I)

- What is the probability that a student in this room has the same birthday as Yongdae?
  - 1/365. Why?
- What is the minimum value of k such that the probability is greater than 0.5 that at least 2 students in a group of k people have the same birthday?
  - $1\,(1 - 1/n)(1 - 2/n)\ldots(1 - (k-1)/n)$

$\leq e^{-1/n}\, e^{-2/n} \ldots e^{-(k-1)/n}$ $\Leftarrow 1 + x \leq e^{x}$ Taylor series

$= e^{-\Sigma\, i/n} = e^{-k(k-1)/2n}$

$\leq 1/2$

  - $-\,k(k-1)/2n \leq \ln(1/2) \Rightarrow k \geq (1 + (1 + (8 \ln 2)\, n)^{1/2})\,/\,2$
  - For $n = 365$, $k \geq 23$

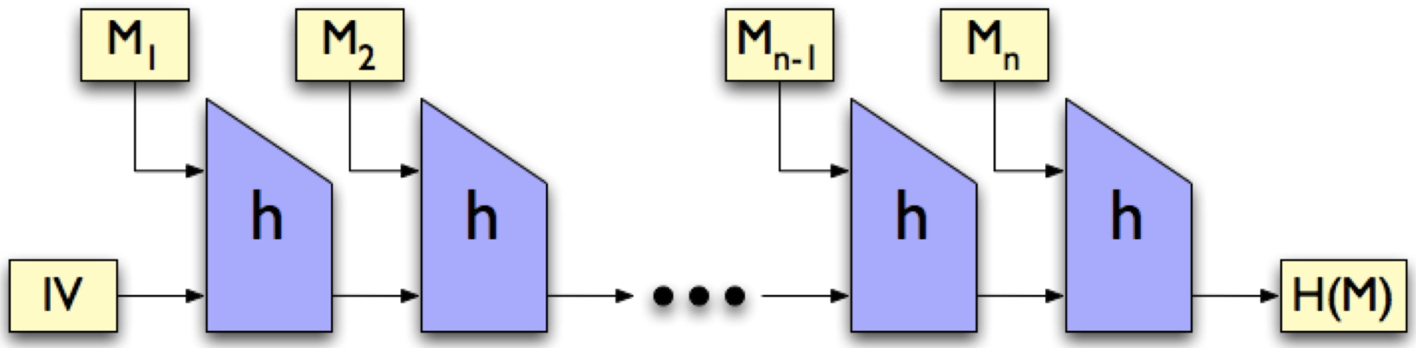
KAIST

Birthday Paradox (II)

- Relation to Hash Function?
  - When n-bit hash function has uniformly random output
  - One-wayness: Pr[y = h(x)] ?
  - Weak collision resistance: Pr[h(x) = h(x') for given x] ?
  - Collision resistance: Pr[h(x) = h(x')] ?

Merkle-Damgård scheme

- The most popular and straightforward method for combining compression functions



Merkle-Damgård scheme

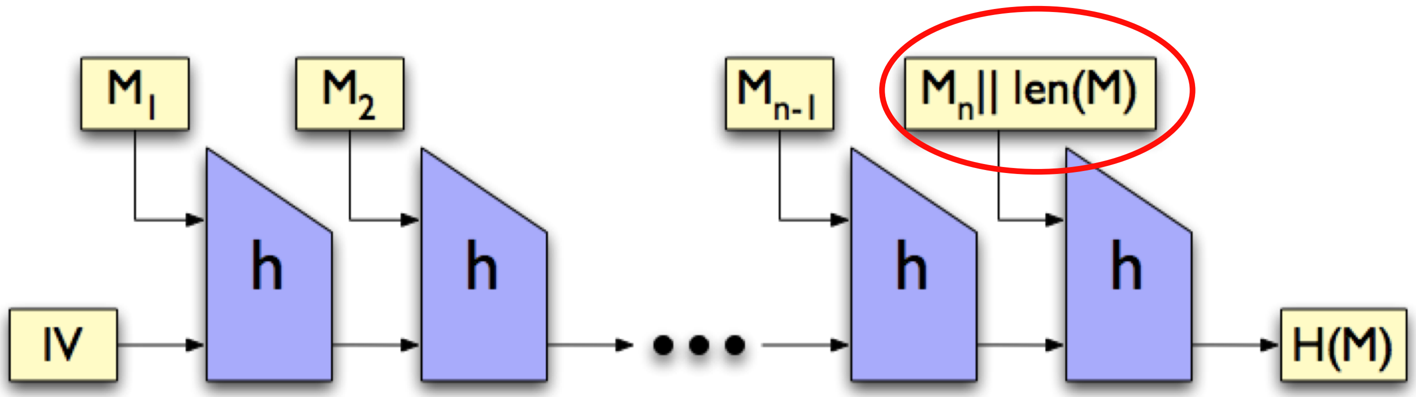
- $h(s, x)$: the compression function
  - s: 'state' variable in $\{0,1\}^n$
  - x: 'message block' variable in $\{0,1\}^m$
- $s_0=IV$, $s_i=h(s_{i-1}, x_i)$
- $H(x_1||x_2||\ldots||x_n)=h(h(\ldots h(IV,x_1),x_2)\ldots,x_n)=s_n$

KAIST

Merkle-Damgård strengthening

- In the previous version, messages should be of length divisible by m, the block size
  - a padding scheme is needed: x||p for some string p so that m | len(x||p)
- Merkle-Damgård strengthening:
  - encode the message length len(x) into the padding string p

Strengthened Merkle-Damgård

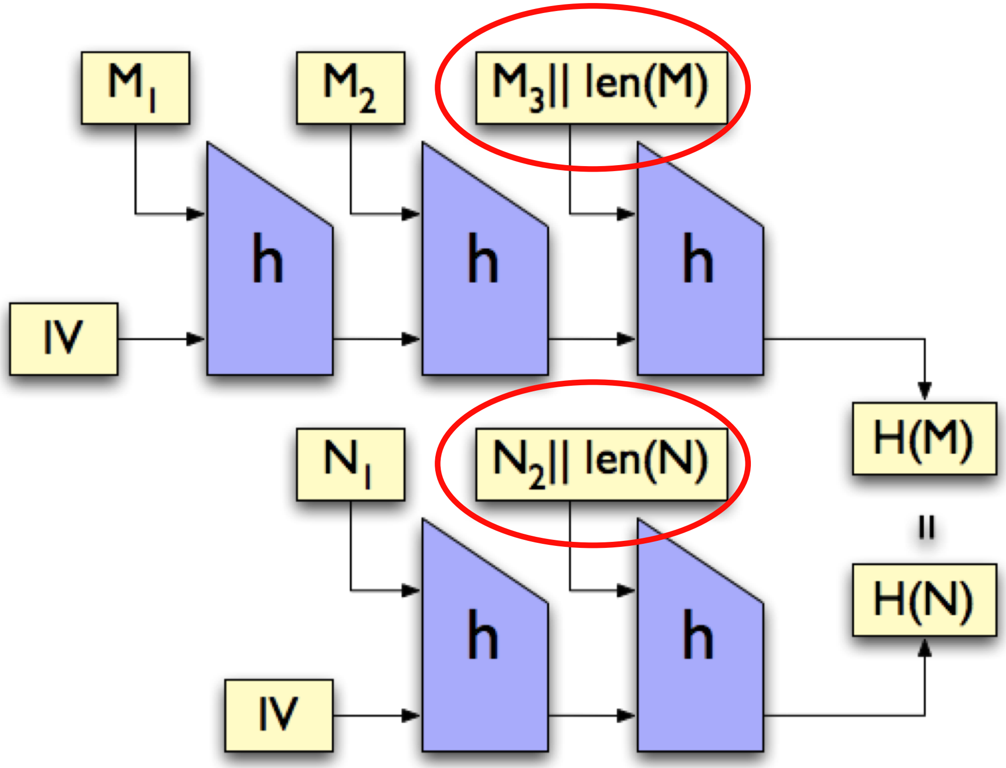


Collision resistance

- If the compression function is collision resistant, then strengthened Merkle-Damgård hash function is also collision resistant
- Collision of compression function:
  $f(s, x)=f(s', x')$ but $(s, x)\neq(s', x')$

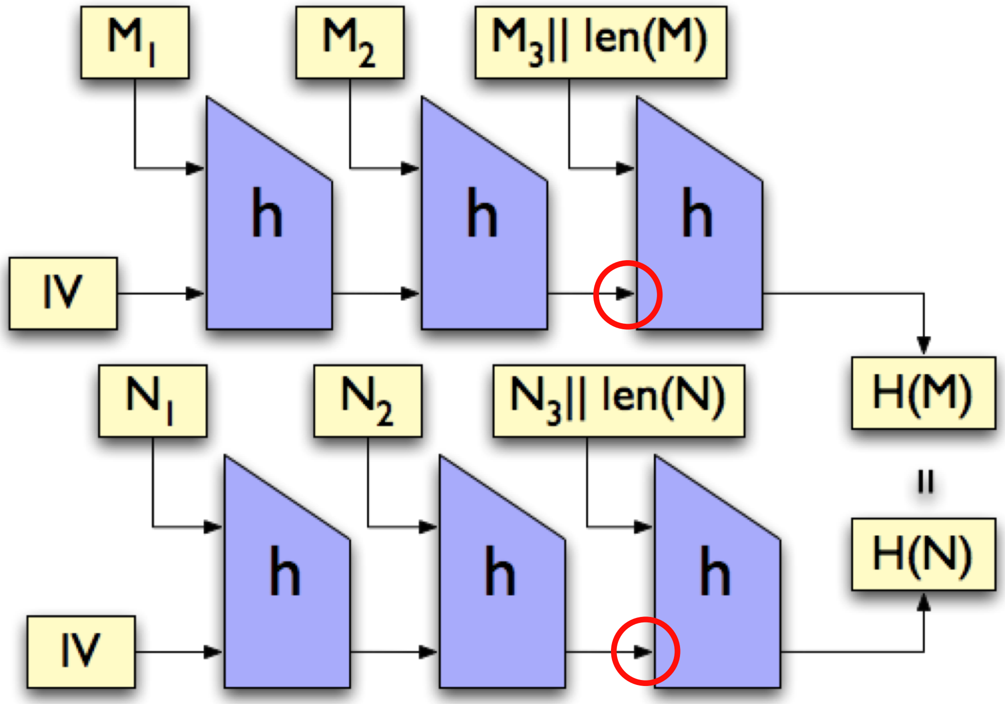
KAIST

Collision resistance

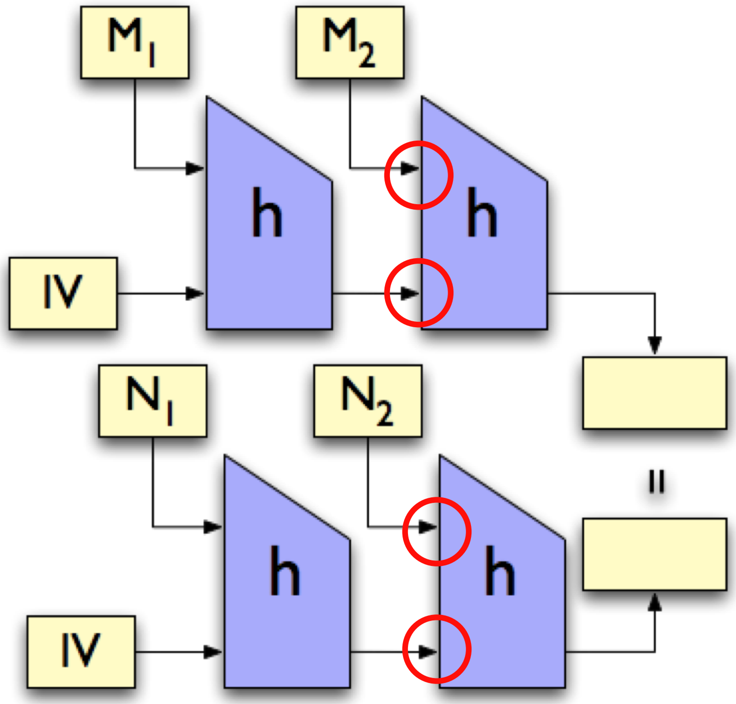


- If h(,) is collision resistant, and if H(M)=H(N), then len(M) should be len(N), and the last blocks should coincide

Collision resistance

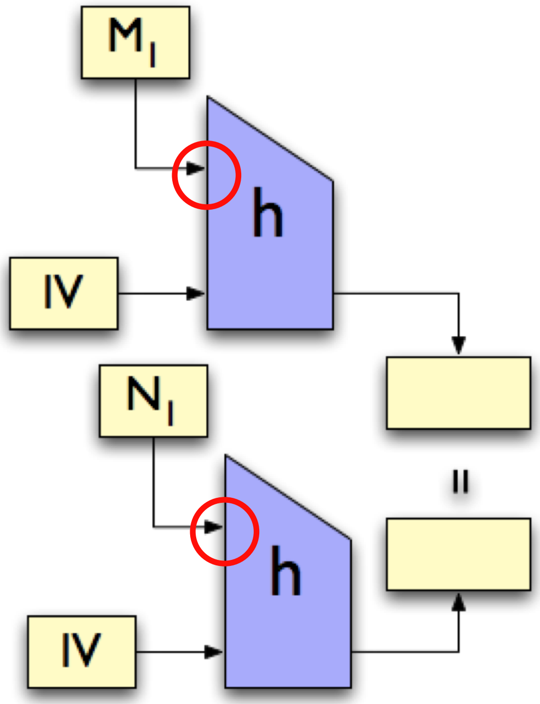


Collision resistance



- And the penultimate blocks should agree, and,

Collision resistance



- And the ones before the penultimate, too…
- So in fact M=N

Extension property

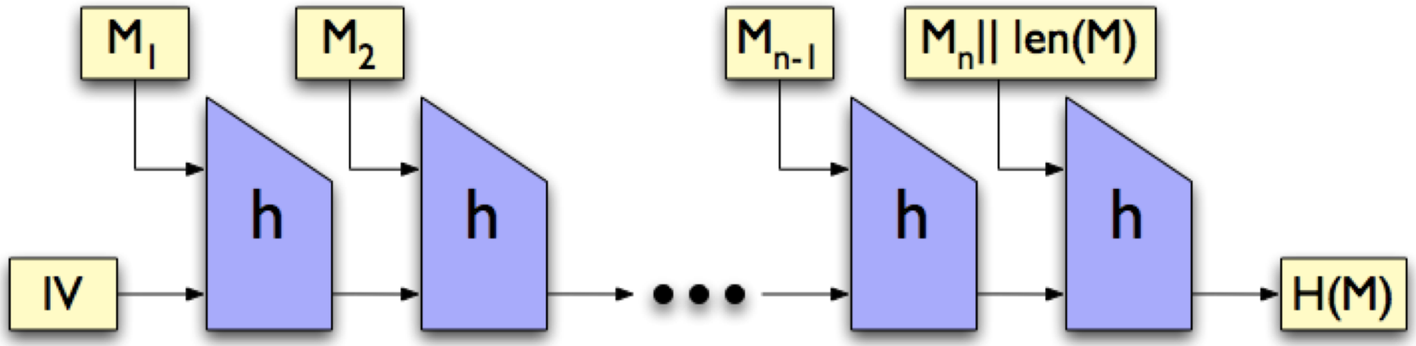
- For a Merkle-Damgård hash function, H(x, y) = h(H(x),y)
  - Even if you don't know x, if you know H(x), you can compute H(x, y)
  - H(x, y) and H(x) are *related* by the formula
  - Would this be possible if H() was a random function?

Fixing Merkle-Dåmgard

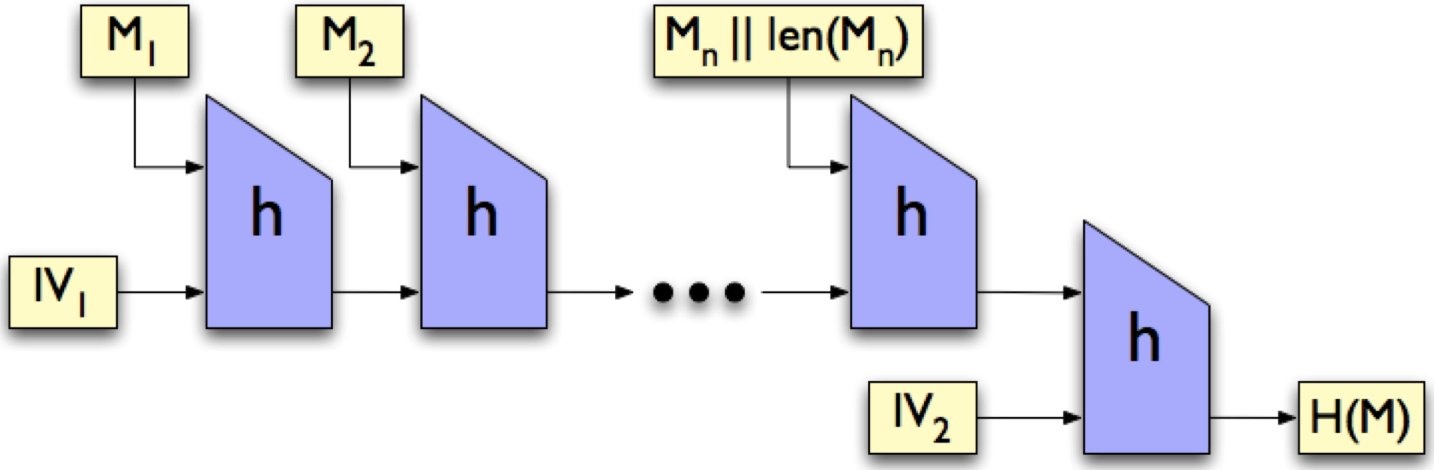
- Merkle-Dåmgard: historically important, still relevant, but likely will not be used in the future (like in SHA-3)
- Clearly distinguishable from a random oracle
- How to fix it? Simple: do something completely different in the end

KAIST

SMD

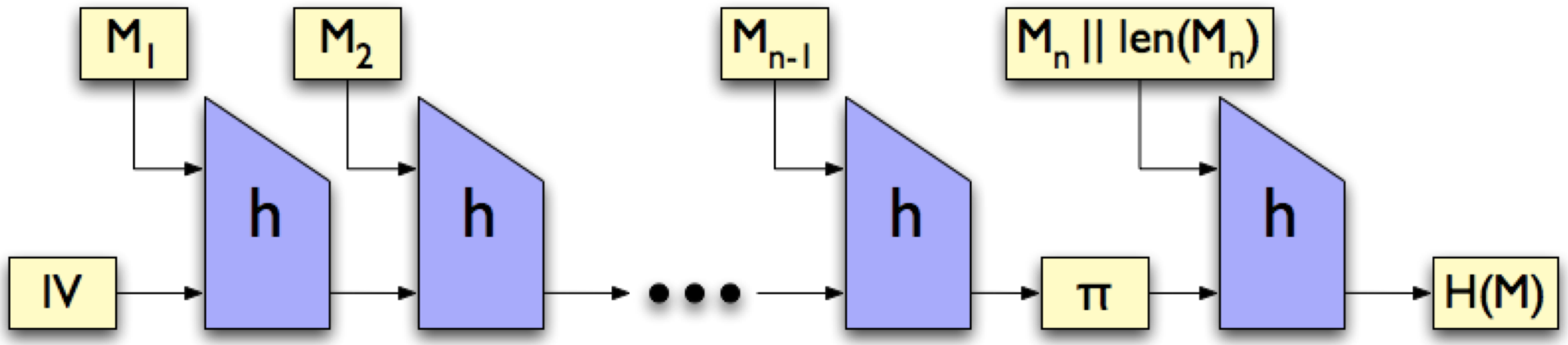


EMD



- $IV_1 \neq IV_2$

MDP



- π: a permutation with few fixed points
  - For example, $\pi(x)=x\oplus C$ for some $C\neq 0$

MAC & AE

Two easy attacks

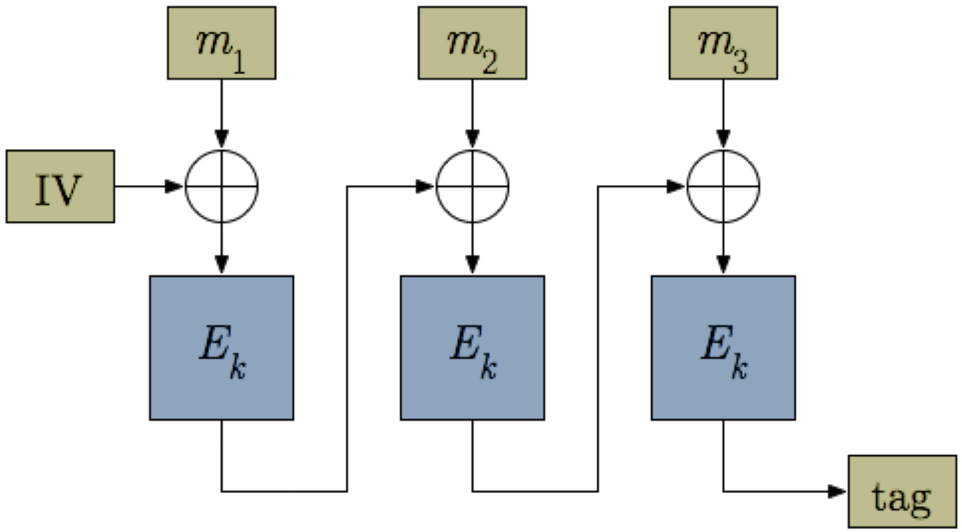
- Exhaustive key search
  - Given one pair (x, M), try different keys until $M=H_k(x)$
  - Lesson: key size should be large enough
- Pure guessing: try many different M with a fixed message x
  - Lesson: MAC length should be also large

- Question: which one is more serious?

Practical constructions

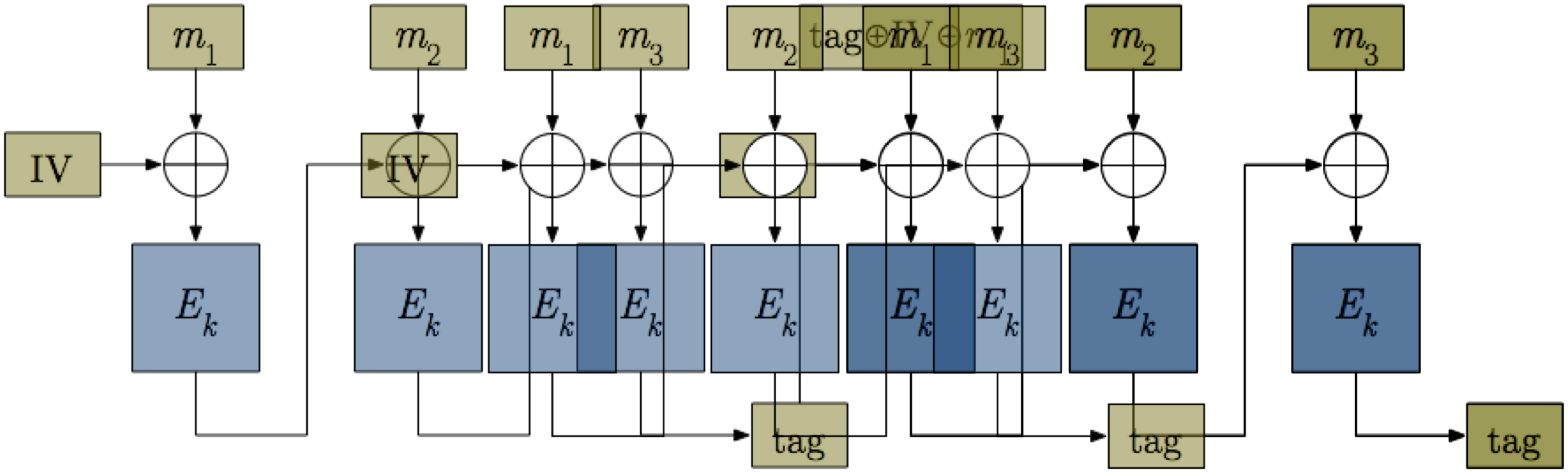
- Blockcipher based MACs
  - CBC-MAC
  - CMAC
- Hash function based MACs
  - secret prefix, secret suffix, envelop
  - HMAC

CBC-MAC



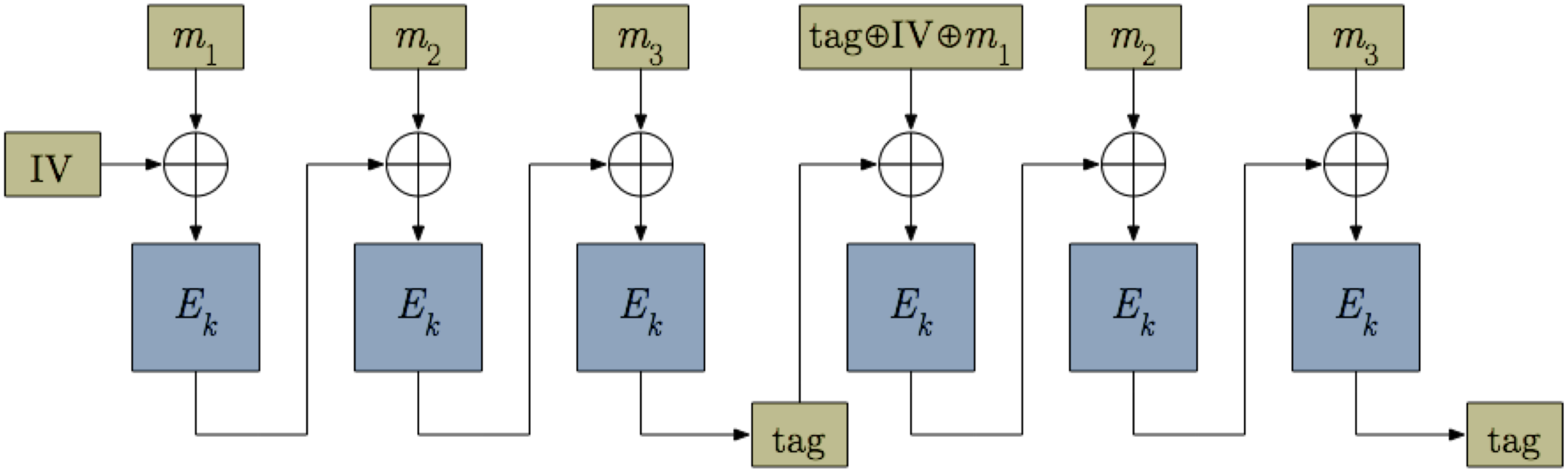
- CBC, with some fixed IV. Last 'ciphertext' is the MAC
- Block ciphers are already PRFs. CBC-MAC is just a way to combine them
- Secure as PRF, if message length is fixed

CBC-MAC



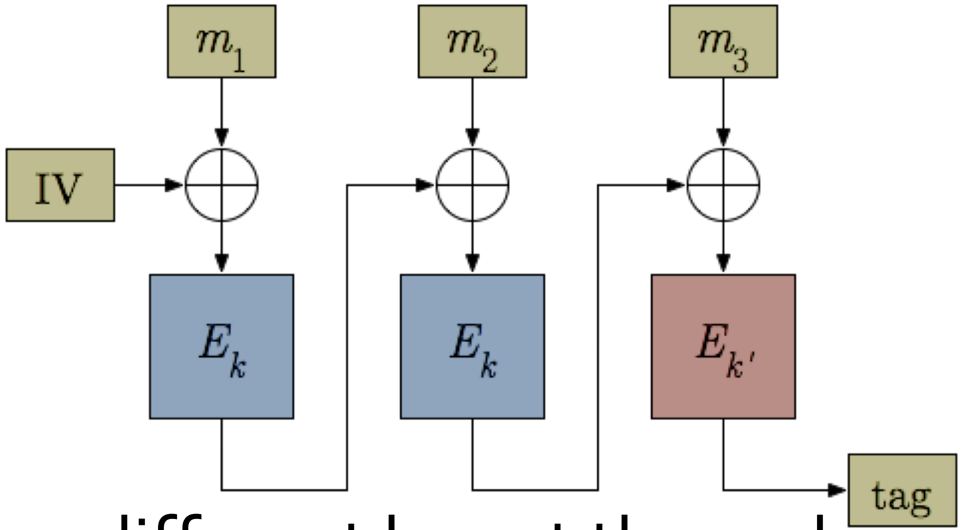
- Secure as PRF, if message length is fixed
- Completely insecure if the length is variable!!!

CBC-MAC



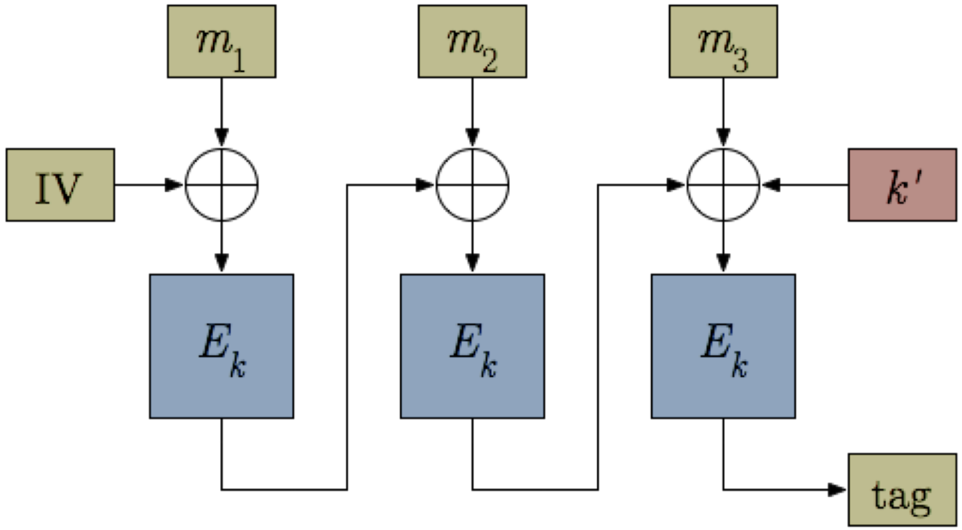
- ‘Extension property’ once more!
- How to fix it?
  - Again, do something different at the end to break the chain

Modification 1



- Use a different key at the end
- Good: this solves the problem
- Bad: switching block cipher key is bad

Modification 2



- XORing a different key at the input is indistinguishable from switching the block cipher key

CMAC

- NIST standard (2005)
- Solves two shortcomings of CBC-MAC
  - variable length support
  - message length doesn't have to be multiple of the blockcipher size

Some Hash-based MACs

- Secret prefix method: $H_k(x)=H(k, x)$
- Secret suffix method: $H_k(x)=H(x, k)$
- Envelope method with padding:
  $H_k(x)=H(k, p, x, k)$

Secret prefix method

- Secret prefix method: $H_k(x)=H(k, x)$
  - Secure if H is a random function
  - Insecure if H is a Merkle-Damgård hash function
    - $H_k(x, y)=h(H(k, x), y)=h(H_k(x), y)$

Secret suffix method

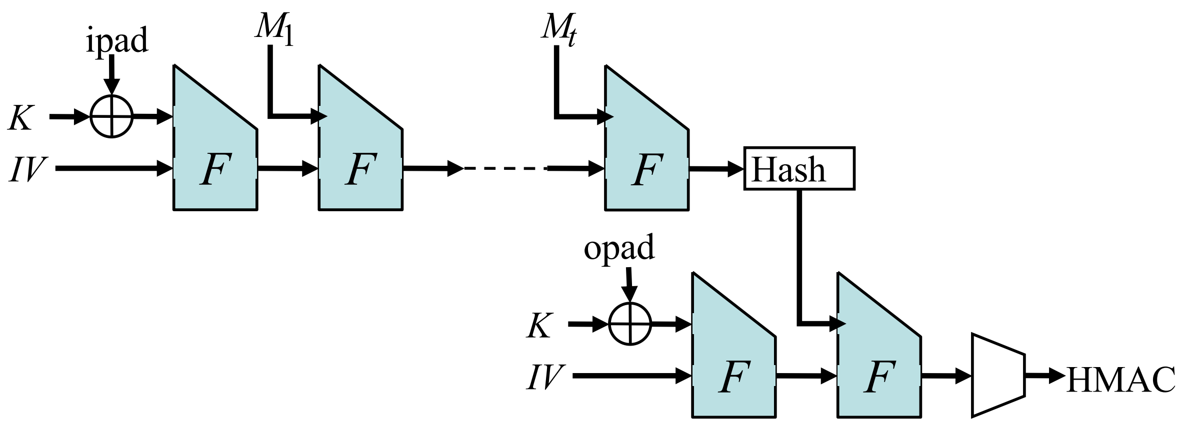
- Secret suffix method: $H_k(x)=H(x, k)$
  - Much securer than secret prefix, even if H is Merkle-Damgård
  - An attack of complexity $2^{n/2}$ exists:
    - Assume that H is Merkle-Damgård
    - Find hash collision $H(x)=H(y)$
    - $H_k(x) = h(H(x), k) = h(H(y), k) = H_k(y)$
    - off-line!

Envelope method

- Envelope method with padding:
  $H_k(x)=H(k, p, x, k)$
  - For some padding p to make k||p at least one block
- Prevents both attacks

HMAC

- NIST standard (2002)
- $HMAC_k(x)=H(K \oplus opad \,||\, H(K \oplus ipad \,||\, x))$
- Proven secure as PRF, if the compression function h of H satisfies some properties



Encryption and Authentication

- $E_K(M)$

- Redundancy-then-Encrypt: $E_K(M, R(M))$
- Hash-then-Encrypt: $E_K(M, h(M))$
- Hash and Encrypt: $E_K(M), h(M)$
- MAC and Encrypt: $E_{h1(K)}(M), HMAC_{h2(K)}(M)$
- MAC-then-Encrypt: $E_{h1(K)}(M, HMAC_{h2(K)}(M))$