EE817/IS 893

Cryptography Engineering and Cryptocurrency

Yongdae Kim

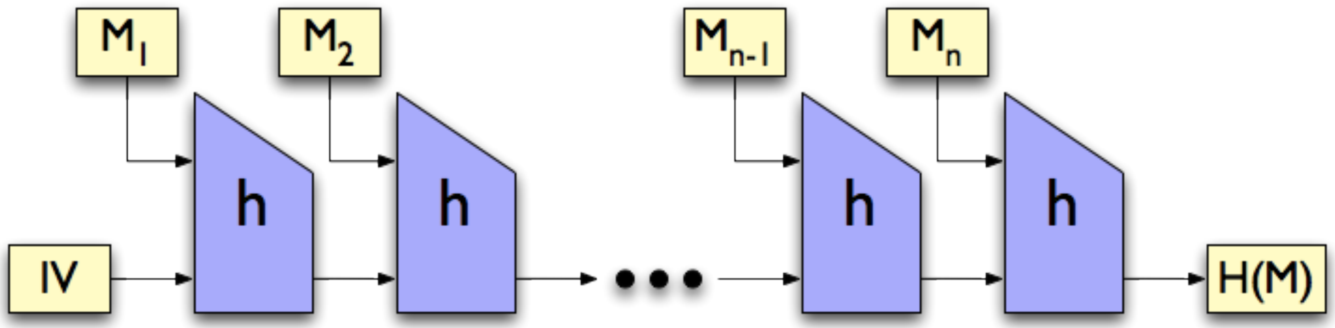
한국과학기술원

Definition

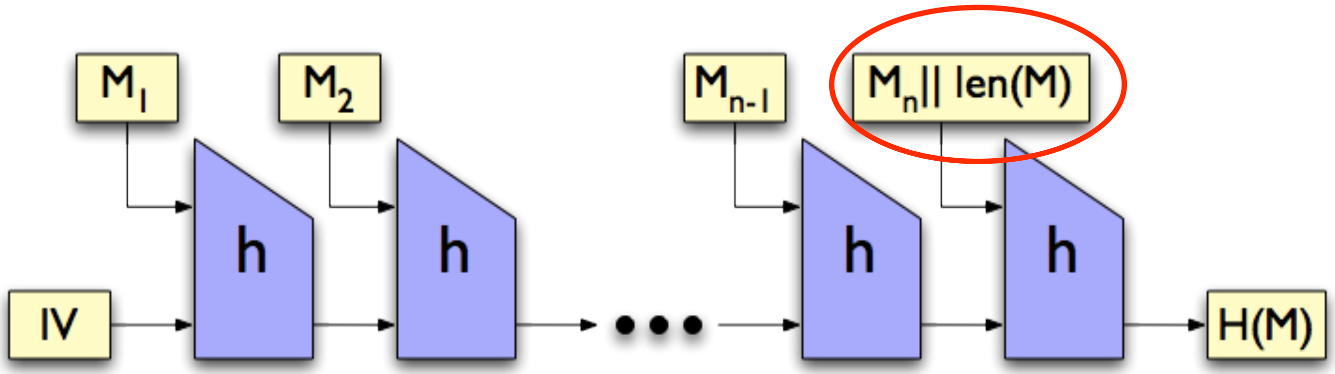
- A *hash function* is a function h
  - *compression* — h maps an input x of arbitrary finite bitlength, to an output h(x) of fixed bitlength n.
  - *ease of computation* — h(x) is easy to compute for given x and h
- *preimage resistance = one-way*
  - it is computationally infeasible to find any input which hashes to that output
- *2nd-preimage resistance = weak collision resistance*
  - it is computationally infeasible to find any second input which has the same output as any specified input
- *collision resistance = strong collision resistance*
  - it is computationally infeasible to find any two distinct inputs x, x' which hash to the same output

Merkle-Damgard Scheme

- The most popular and straightforward method for combining compression functions



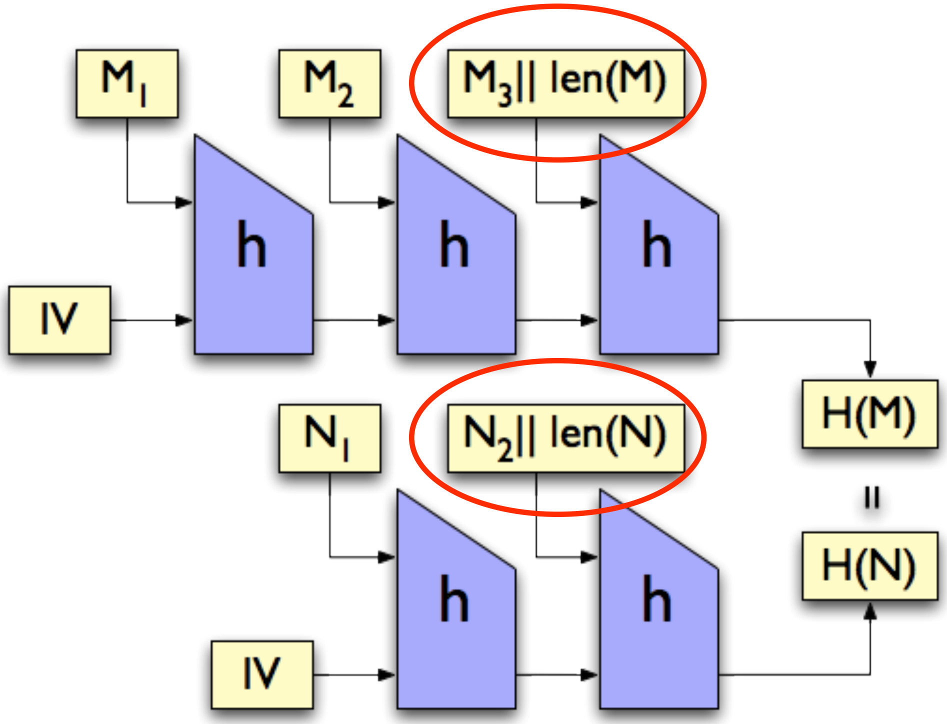
Strengthened Merkle-Damgard



collision resistance

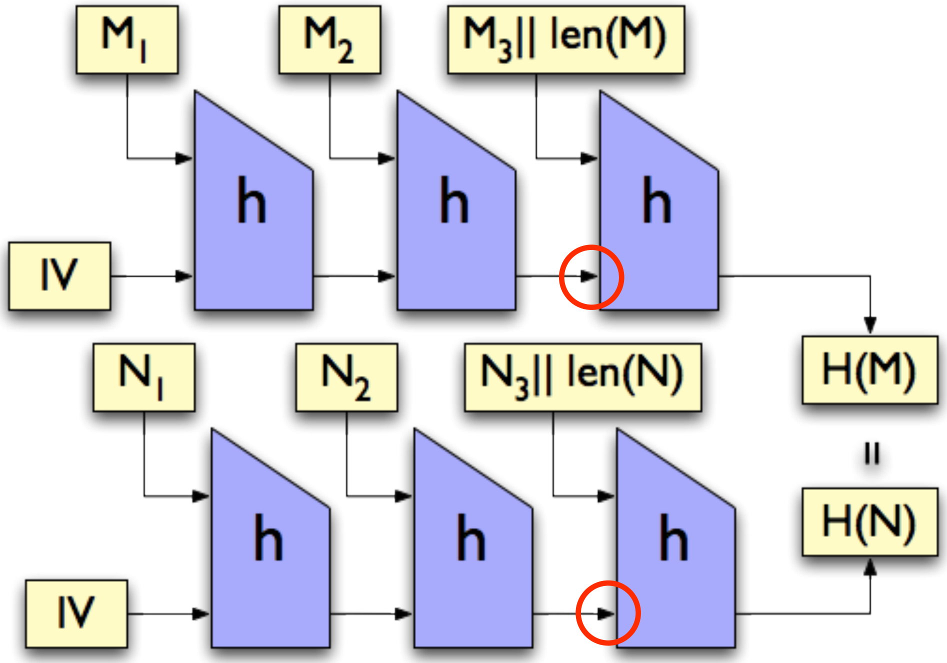
- If the compression function is collision resistant, then strengthened Merkle-Damgard hash function is also collision resistant
- collision of compression function:
  $f(s, x)=f(s', x')$ but $(s, x) \neq (s', x')$

collision resistance

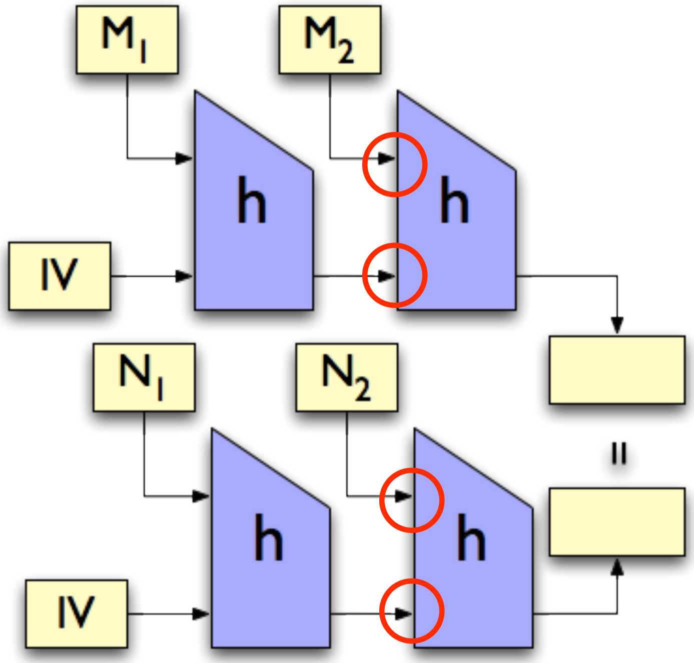


- If h(,) is collision resistant, and if H(M)=H(N), then len(M) should be len(N), and the last blocks should coincide

collision resistance

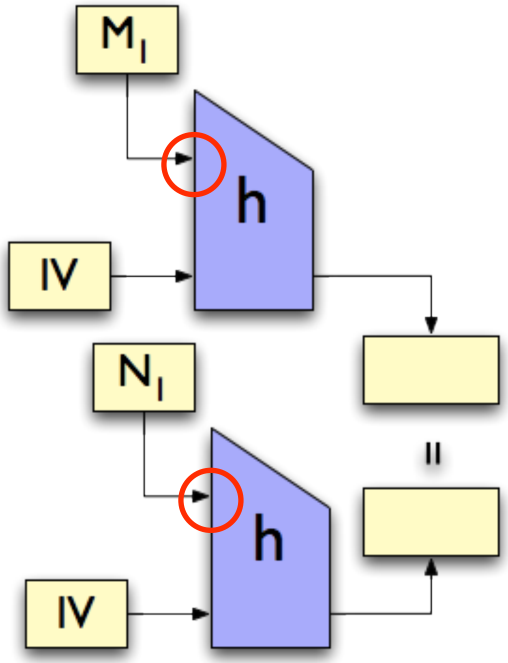


collision resistance



- 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-Damgard hash function,

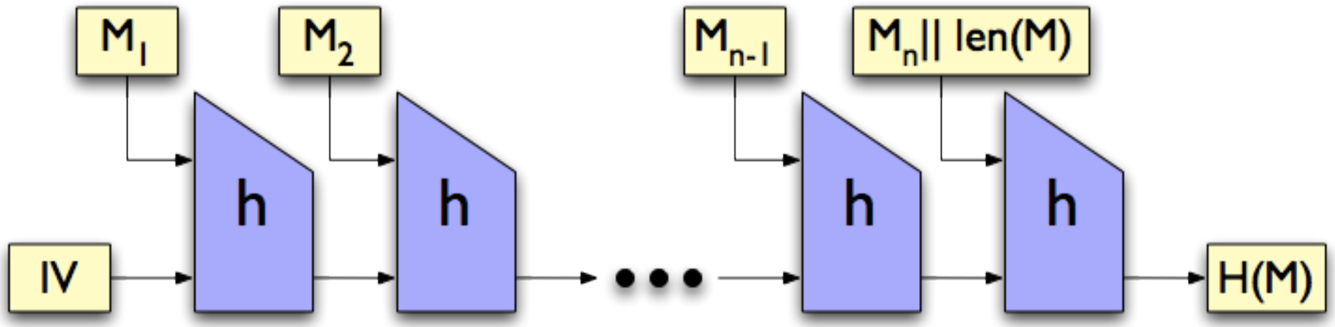
  $H(x, y) = h(H(x), y)$

  - 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?

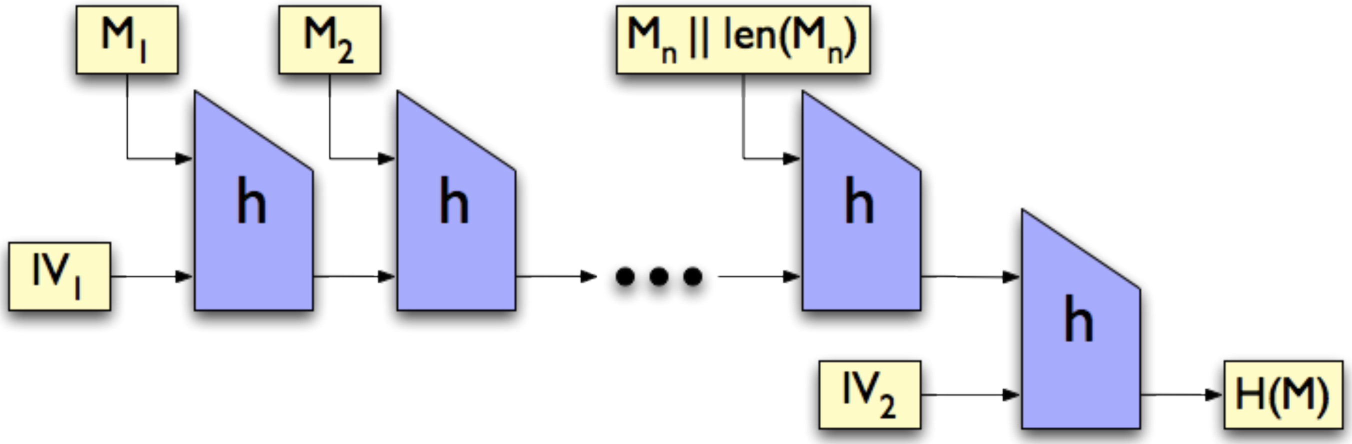
Fixing Merkle-Damgard

- Merkle-Damgard: 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

SMD

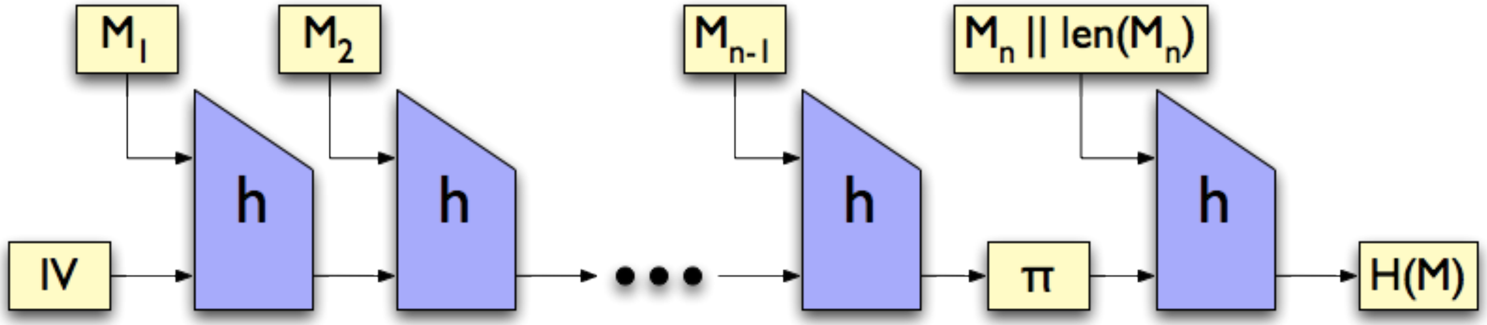


EMD



- $IV_1 \neq IV_2$

MDP



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

Hash chain

- h: cryptographically strong hash function
- $H_0 = x$
- $H_n = h(H_{n-1}) = h(h(h(\cdots h(x))))$

- Random mapping statistics

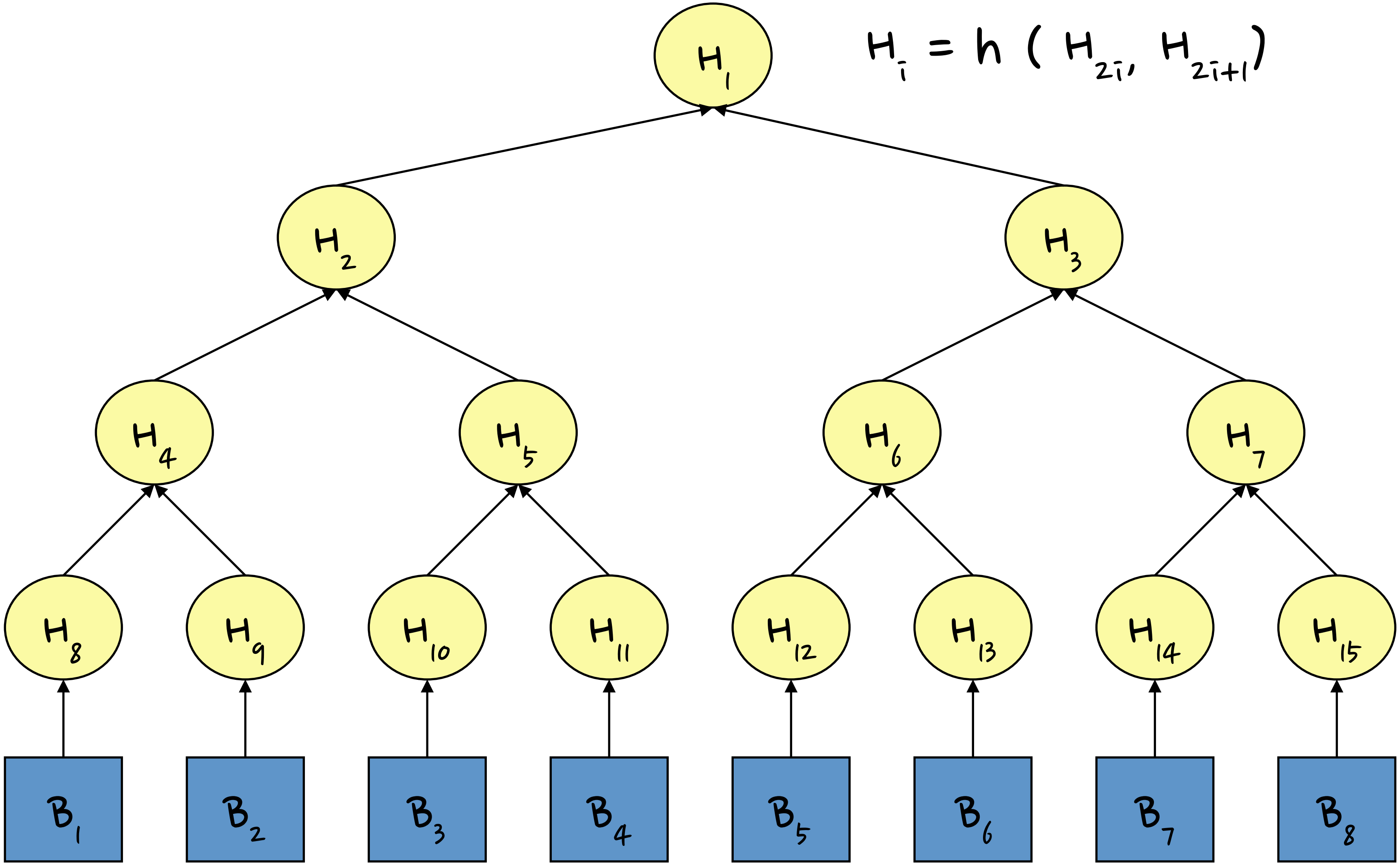
One time password

- Setup
  - User generates $H_0, H_1, \cdots H_n$.
  - User → Server: H_n
  - Server stores H_n as the user's public password.

- Authentication
  - At time 0: User →Server: H_{n-1}
  - Server verifies $h(H_{n-1}) = H_n$
  - Server stored H_{n-1} as the user's public password.
  - At time 1: User → Server: H_{n-2}
  - …

Hash Tree

$H_i = h(H_{2i}, H_{2i+1})$



MAc & AE

MAC

- Message Authentication code
- 'keyed hash function' $H_k(x)$
  - k: secret key, x: message of any length, $H_k(x)$: fixed length (say, 128 bits)
  - deterministic
- Purpose: to 'prove' to someone who has the secret key k, that x is written by someone who also has the secret key k

How to use?

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

Attack scenario

- E may eavesdrop many communications (x, M) between A & B
- E then tries (possibly many times) to 'forge' (x', M') so that B accepts: $M' = H_k(x')$
- Question: what if E 'replays' old transmission (x, M)? Is this a successful forgery?

capabilities of attackers

- known-text attack
  - Simple eavesdropping
- chosen-text attack
  - Attacker influences Alice's messages
- Adaptive chosen-text attack
  - Attacker adaptively influences Alice

Types of forgery

- Universal forgery: attacker can forge a MAC for any message
- Selective forgery: attacker can forge a MAC for a message chosen before the attack
- Existential forgery: attacker can forge some message x but in general cannot choose x as he wishes

Security of MAc

- Should be secure against adaptively chosen-message existential forger
  - Attacker may watch many pairs (x, H_k(x))
  - May even try x of his choice
  - May try many verification attempts (x, M)
  - Still shouldn't be able to forge a new message at all

Two easy attacks

- Exhaustive key search
  - Given one pair (x, M), try different keys until M=Hk(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?

Random function as MAC

- Suppose A and B share a random function R(x), which assigns random 128-bit value to its input x
- Even if E sees many messages of form (x, R(x)), for a new y, R(y) can be any of 2^{128} strings
- Successful forgery prob. $\leq 2^{-128}$

Random function as MAC

- It is a perfect MAC, but the 'key size' is too large: how many functions of form $R: \{0,1\}^m \longrightarrow \{0,1\}^n$? Answer: 2^(n 2^m)
- But there are keyed functions which are 'indistinguishable' from random functions: called PRFs (PseudoRandom Functions)
- Designing a secure PRF is a good way to design a secure MAC

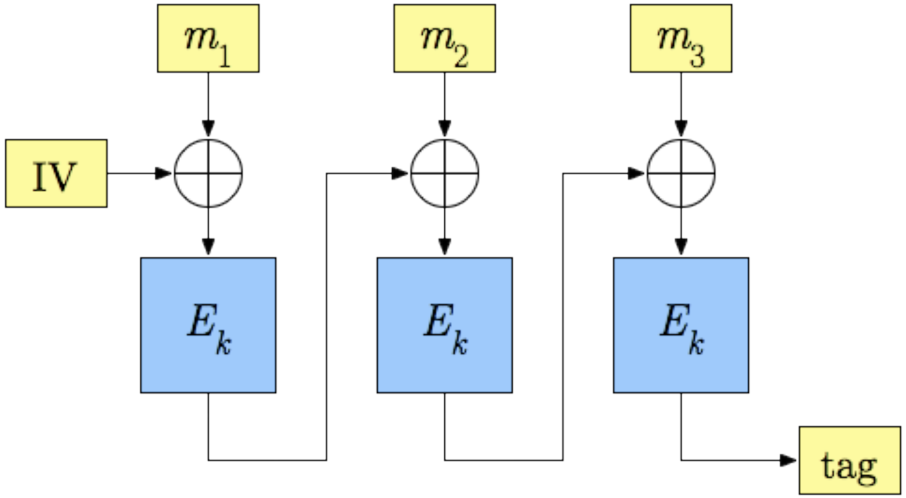
Truncation of MAC

- ❑ $H_k(x)$ is a secure MAC with 256-bit output
- ❑ $H'_k(x)$ = the first 128 bits of $H_k(x)$
- ❑ Question: is $H'_k(x)$ a secure MAC?
  - Answer: not in general, but secure if $H_k(x)$ is a secure PRF

Practical constructions

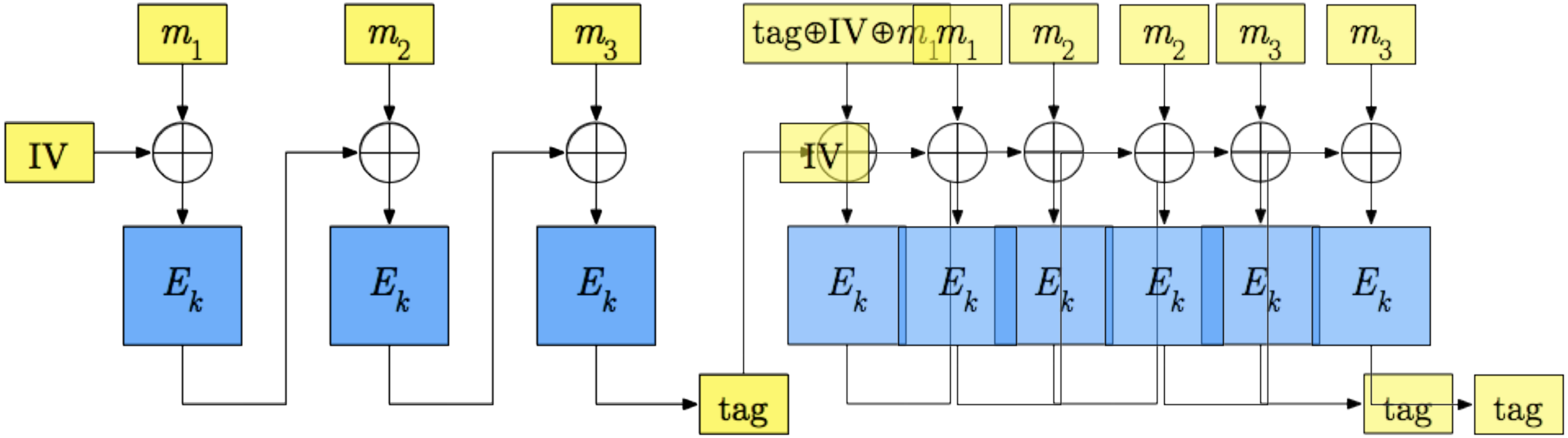
- Blockcipher based MACs
  - CBC-MAC
  - CMAC
- Hash function based MACs
  - secret prefix, secret suffix, envelop
  - HMAC

CBC-MAC



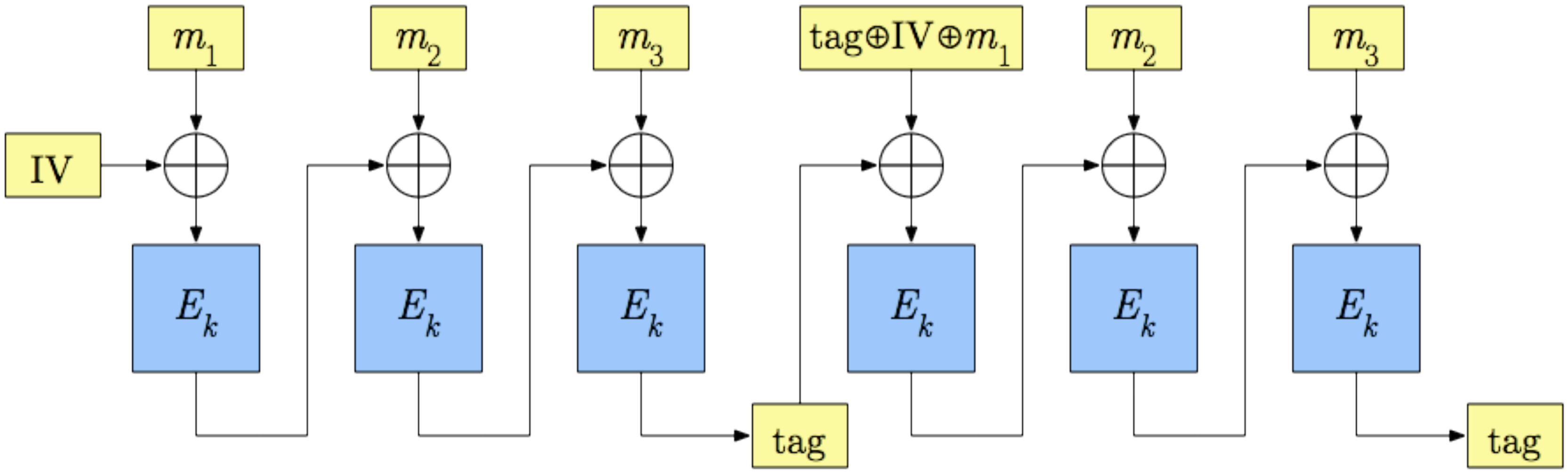
- CBC, with some fixed IV. Last 'ciphertext' is the MAC
- Block ciphers are already PRFs. CBC-MAC is just a way to combine them
- Secure as PRF, if message length is fixed

CBC-MAC



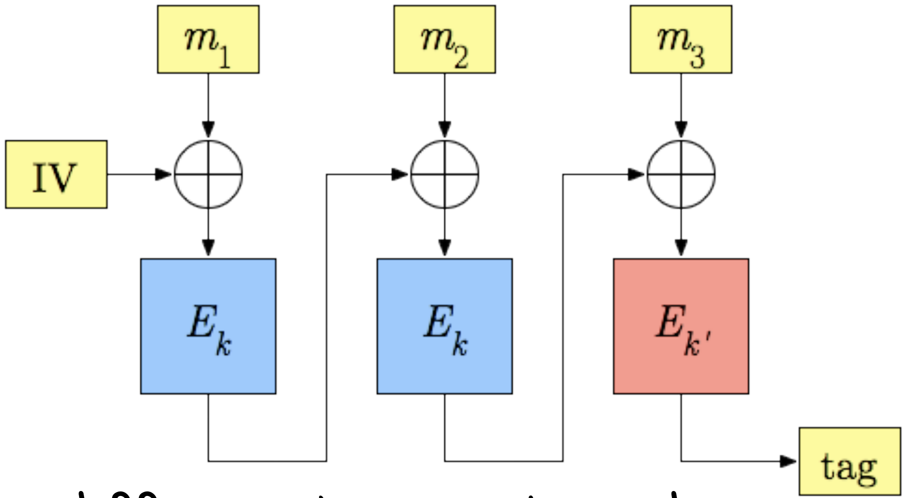
- Secure as PRF, if message length is fixed
- completely insecure if the length is variable!!!

CBC-MAC



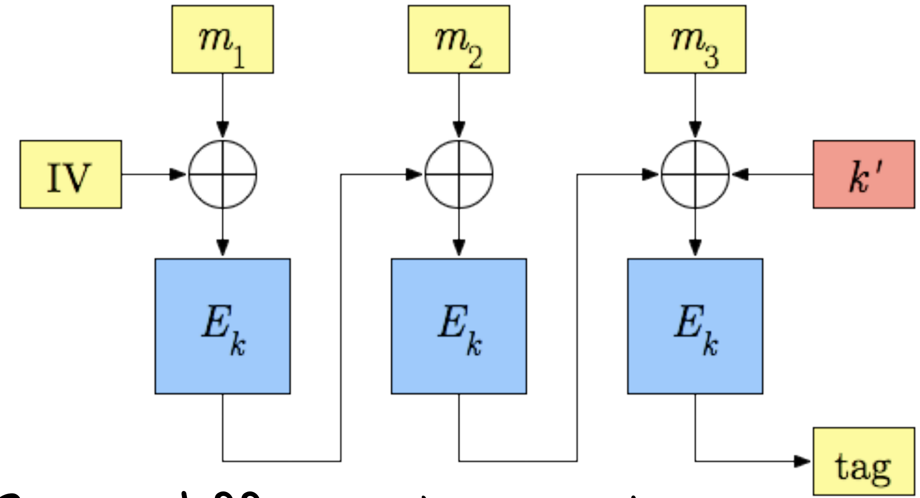
- ‘Extension property’ once more!
- How to fix it?
  - ▷ Again, do something different at the end to break the chain

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-Damgard 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-Damgard
  - An attack of complexity $2^{n/2}$ exists:
    - Assume that H is Merkle-Damgard
    - 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:

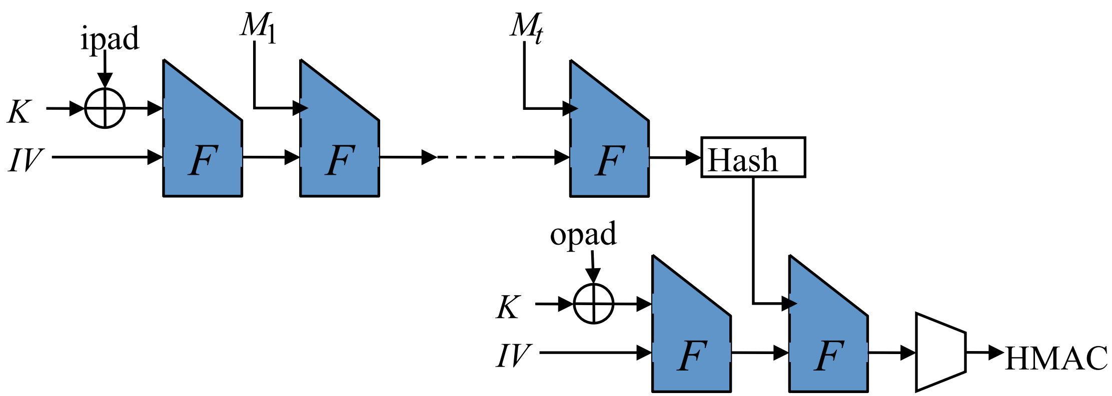
  $H_k(x)=H(k, p, x, k)$

  - 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



MAc vs Signature

- secret key vs. public key
- private verification vs. public verification
- MAc doesn't provide non-repudiation
  - Bob claims that Alice sends (x, M), showing that $M=H_k(x)$. Who else can write this message?

confidentiality & integrity

- Two symmetric key primitives
  - Encryption scheme: protects confidentiality
  - MAC: protects integrity
- Usually, what we want is to protect both

Encryption not enough?

- 'It's encrypted so nobody can alter it!'
- $C=E_k(P)$
- If any string is a valid ciphertext (e.g., a blockcipher), modifying C to C' will alter your P (to P', perhaps a garbage)
  - Question: is this a problem?

Giving redundancy

- Solution: not all strings are valid ciphertext
  - Format plaintext with some redundancy
  - Only correctly formatted plaintext is to be accepted
  - Example, $C=E_k(P \| P)$, or $C=E_k(P \| H(P))$
  - Be careful: what if $E_k()$ is a stream cipher?

Generic composition

- Instead of using an ad-hoc method,
- combine a secure encryption scheme (say, CBC, CTR) and a secure MAC (say, CMAC, HMAC)
  - Two keys are needed
  - How to combine two?
  - 'Generic' here means 'black-box'

Generic composition

- MAc-and-Encrypt: $E_{ke}(P) \| M_{km}(P)$
- MAc-then-Encrypt: $E_{ke}(P \| M_{km}(P))$
- Encrypt-then-MAc: $E_{ke}(P) \| M_{km}(E_{ke}(P))$

Questions?

- Yongdae Kim
  - email: yongdaek@kaist.ac.kr
  - Home: http://syssec.kaist.ac.kr/~yongdaek
  - Facebook: https://www.facebook.com/y0ngdaek
  - Twitter: https://twitter.com/yongdaek
  - Google "Yongdae Kim"