정보보호론

해쉬 함수 및 메시지 인증

한림대학교 소프트웨어융합대학 조효진

Contents

- 메시지 무결성
- 암호학적 해쉬 함수(Cryptographic Hash Function)
- 메시지 변조 감지 코드(MDC)
- 메시지 인증코드 (MAC)

메시지 무결성

❒ 메시지 무결성(Message Integrity)

- 원본 메시지의 변조를 방지하기 위한 서비스
- 대칭키 & 공개키 암호시스템은 기밀성을 제공
- 무결성을 위해 Modification Detection Code(MDC) 사용
  - 변조 감지 코드를 생성하기 위한 방법으로 암호학적 해쉬 함수 (Cryptographic Hash Function) 사용

암호학적 해쉬 함수(Cryptographic Hash Function)

- 암호학적 해쉬 함수는 데이터의 무결성을 제공하기 위해 사용됨

Download Kali Linux Images

We generate fresh Kali Linux image files every few months, which we make available for download. This page provides the links to **download Kali Linux** in its latest official release. For a release history, check our Kali Linux Releases page. Please note: You can find unofficial, untested weekly releases at http://cdimage.kali.org/kali-weekly/.

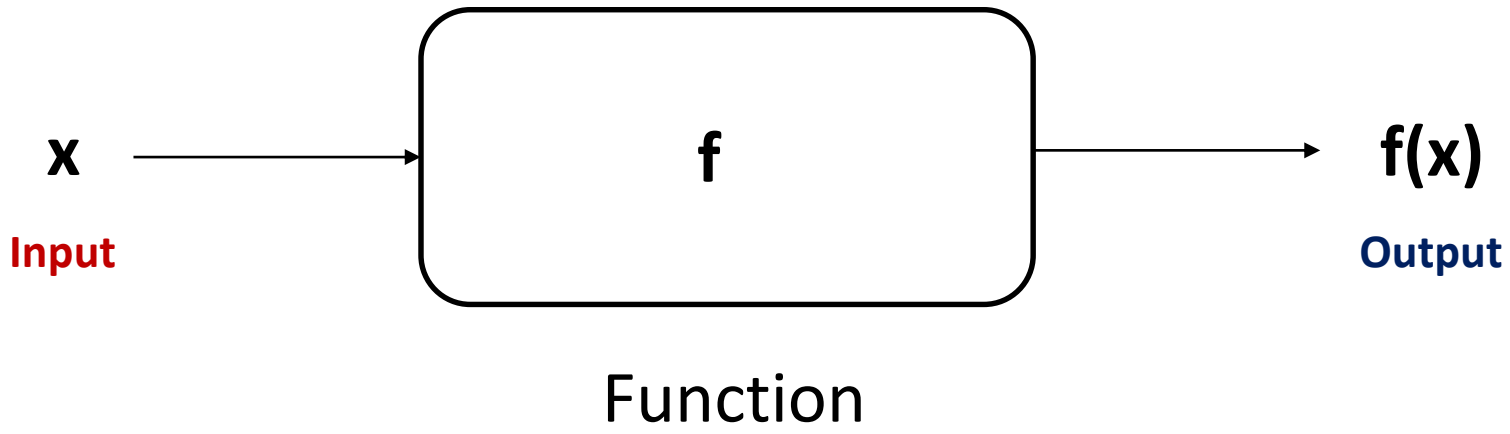
Image Name	Download	Size	Version	sha256sum
Kali Linux 64 Bit	HTTP | Torrent	2.8G	2018.2	56f677e2edfb2efcd0b08662ddde824e254c3d53567ebbbcdbbf5c03efd9bc0f
Kali Linux Light 64 Bit	HTTP | Torrent	865M	2018.2	554f020b0c89d5978928d31b8635a7eeddf0a3900abcacdbc39616f80d247f86
Kali Linux E17 64 Bit	HTTP | Torrent	2.6G	2018.2	be0a858c4a1862eb5d7b8875852e7d38ef852c335c3c23852a8b08807b4c3be8
Kali Linux Lxde 64 Bit	HTTP | Torrent	2.6G	2018.2	449ecca86b0f49a52f95a51acdde94745821020b7fc0bd2129628c56bc2d145d
Kali Linux Xfce 64 Bit	HTTP | Torrent	2.6G	2018.2	0e94035a0a56fccc49961b0da56b9243ed3da6a3f8d696884e6f0b936f74dbfb

암호학적 해쉬 함수(Cryptographic Hash Function)

- 암호학적 해쉬 함수: h = H(M)
  - 임의의 크기의 데이터를 입력 값
  - 고정된 크기의 출력 값
  - 하드웨어 및 소프트웨어의 적용이 쉬워야 하며, 어떤 입력 데이터에 대해서도 출력 값을 계산하기 용이
  - 해쉬 함수는 공개된 함수이며 키가 사용되지 않음
  - 키를 사용한 해쉬 함수: h = H(k,M) → MAC (Message Authentication Code)
  - 암호학적 해쉬 함수는 공개키를 이용한 전자 서명을 생성하기 위하여 사용되기도 함 (전자서명 챕터에서 설명)

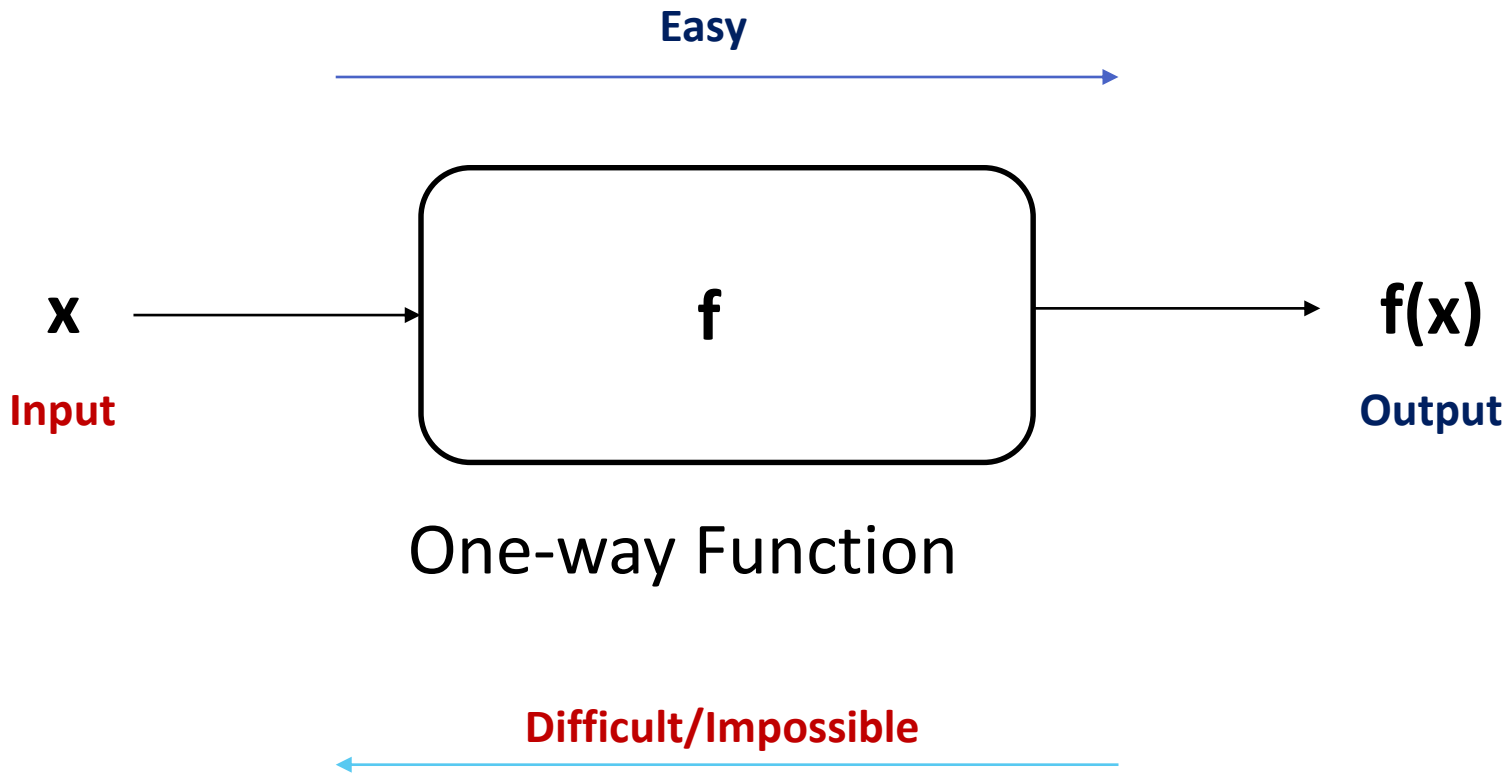
암호학적 해쉬 함수(Cryptographic Hash Function)

- What is a function?



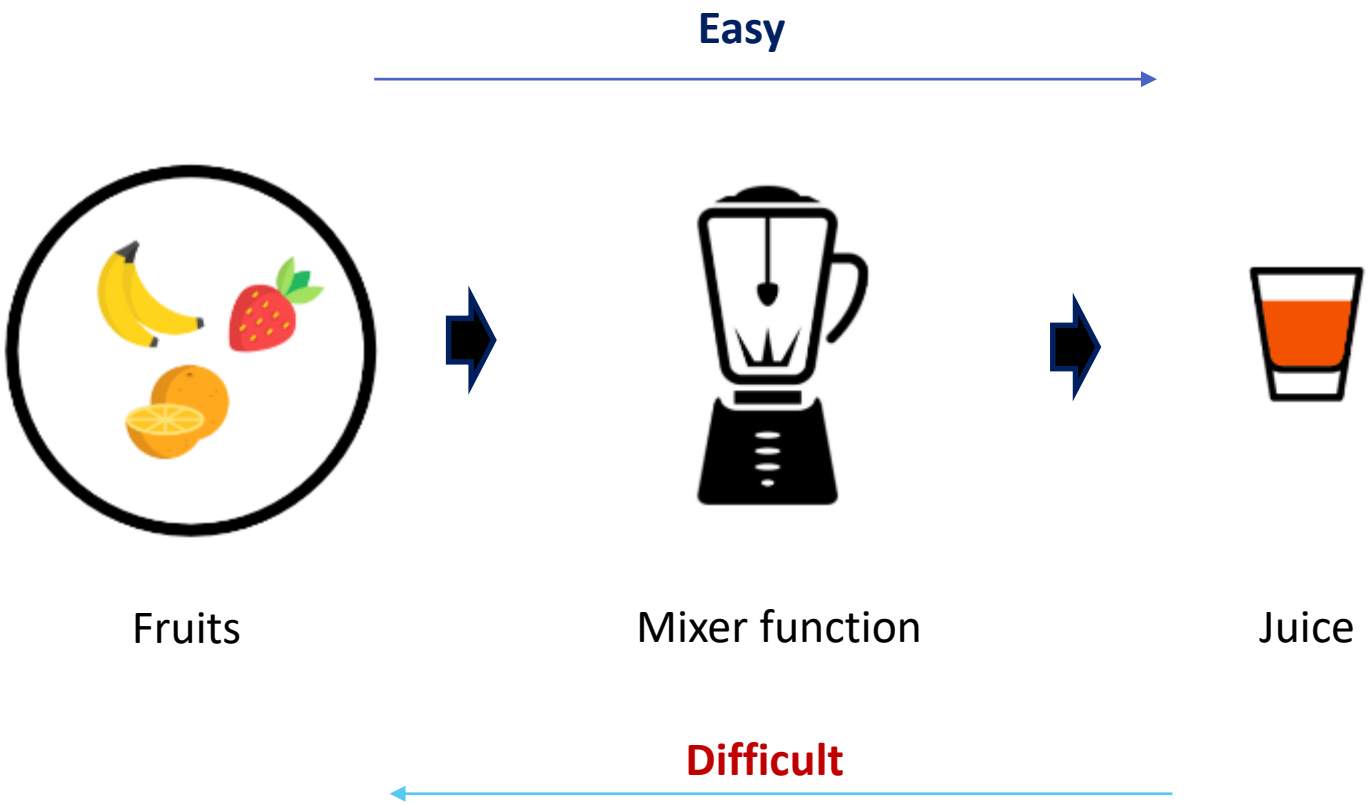
암호학적 해쉬 함수(Cryptographic Hash Function)

- What is a one-way function?



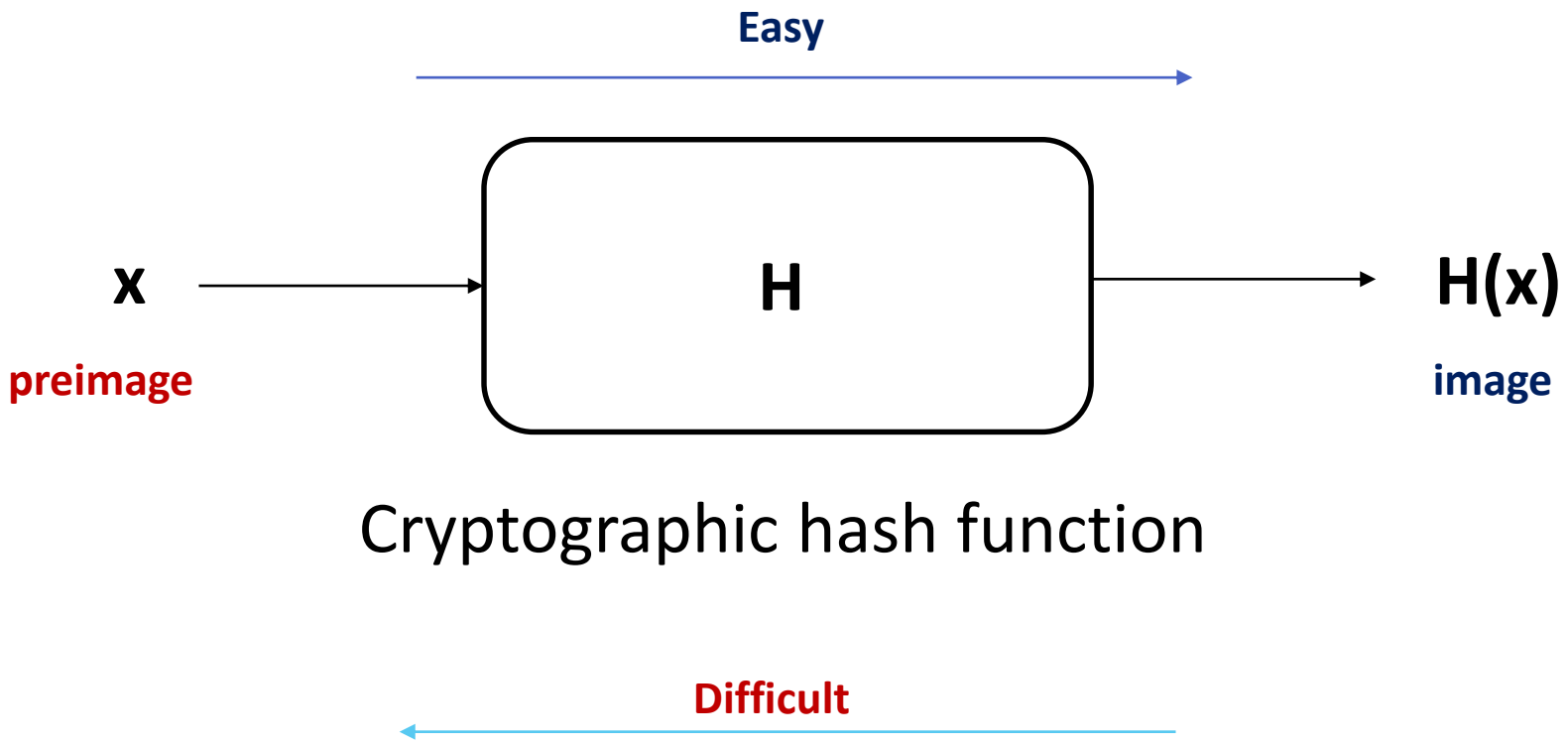
암호학적 해쉬 함수(Cryptographic Hash Function)

- An example of one-way functions

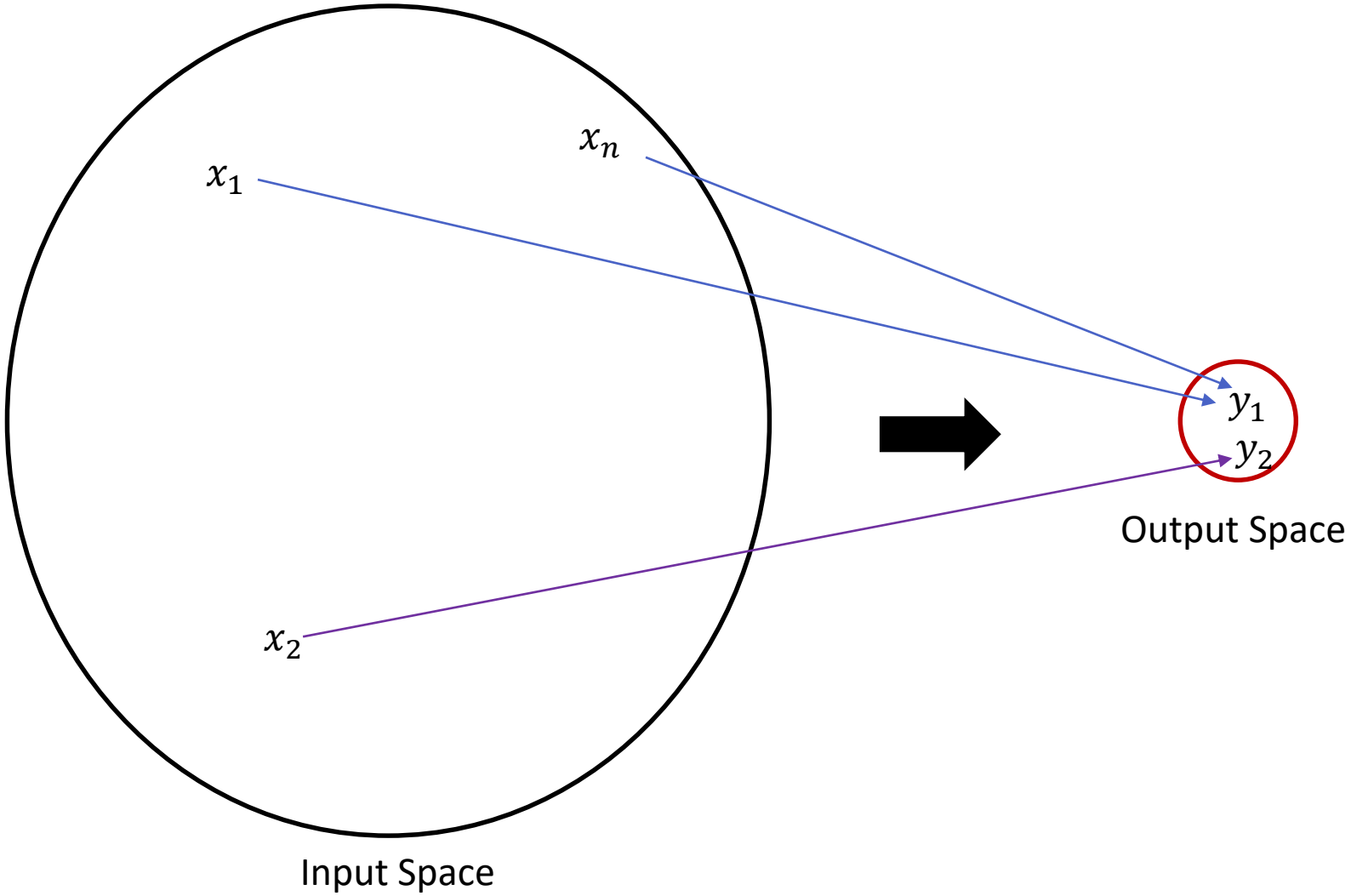


암호학적 해쉬 함수(Cryptographic Hash Function)

- What is a cryptographic hash function?



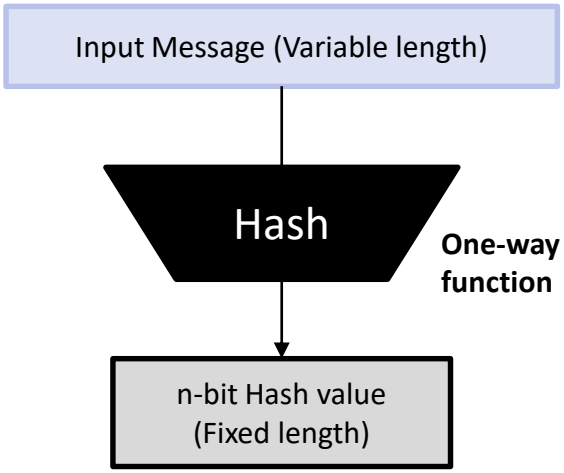
암호학적 해쉬 함수(Cryptographic Hash Function)



암호학적 해쉬 함수(Cryptographic Hash Function)

❒ 암호학적으로 안전한 해쉬 함수의 3가지 성질

- 역상 저항성(Preimage Resistance): Given h, infeasible to find x s.t. H(x)=h
- 제 2 역상 저항성(Second Preimage Resistance): Given x, infeasible to find y s.t. H(y)=H(x)
- 충돌 저항성(Collision Resistance): infeasible to find any (x,y) s.t. H(y)=H(x)



암호학적 해쉬 함수(Cryptographic Hash Function)

❒ 역상 저항성 (Preimage resistance)

- Given H(x), it is computationally difficult to find x

[Source: https://spiritegg.com/wp-content/uploads/2016/03/63180952_fingerprint_types624.jpg]

Fingerprint analogy: Whose fingerprint is this?

암호학적 해쉬 함수(Cryptographic Hash Function)

❒ 제2차 역상 저항성 (Second preimage resistance)

- Given x, it is computationally difficult to find some value x' such that H(x) = H(x')

[Source: https://spiritegg.com/wp-content/uploads/2016/03/63180952_fingerprint_types624.jpg]

Fingerprint analogy:
Can you find someone with same fingerprint as you?

암호학적 해쉬 함수(Cryptographic Hash Function)

❒ 충돌 저항성 (Collision resistance)

- It is computationally difficult to find x and y such that H(x) = H(y)

[Source: https://spiritegg.com/wp-content/uploads/2016/03/63180952_fingerprint_types624.jpg]

Fingerprint analogy:
Can you find two random people with same fingerprint

암호학적 해쉬 함수(Cryptographic Hash Function)

❒ Features of cryptographic hash function

- Preimage resistance

- Second preimage and Collision resistance

X1: Bob sends $5,000 to Alice
X2: Bob transfer 5,000 won to Alice

If H(X1) = H(X2), can we use a cryptographic hash function as a digital fingerprint?

암호학적 해쉬 함수(Cryptographic Hash Function)

❒ Avalanche effect

- If there is one-bit difference between inputs, the corresponding outputs are totally different!
- Input의 한 비트 차이는 Output의 완전한 변화를 야기함

```
I am Satoshi Nakamoto0 => a80a81401765c8eddee25df36728d732...
I am Satoshi Nakamoto1 => f7bc9a6304a4647bb41241a677b5345f...
I am Satoshi Nakamoto2 => ea758a8134b115298a1583ffb80ae629...
I am Satoshi Nakamoto3 => bfa9779618ff072c903d773de30c99bd...
I am Satoshi Nakamoto4 => bce8564de9a83c18c31944a66bde992f...
I am Satoshi Nakamoto5 => eb362c3cf3479be0a97a20163589038e...
I am Satoshi Nakamoto6 => 4a2fd48e3be420d0d28e202360cfbaba...
I am Satoshi Nakamoto7 => 790b5a1349a5f2b909bf74d0d166b17a...
I am Satoshi Nakamoto8 => 702c45e5b15aa54b625d68dd947f1597...
I am Satoshi Nakamoto9 => 7007cf7dd40f5e933cd89fff5b791ff0...
I am Satoshi Nakamoto10 => c2f38c81992f4614206a21537bd634a...
I am Satoshi Nakamoto11 => 7045da6ed8a914690f087690e1e8d66...
I am Satoshi Nakamoto12 => 60f01db30c1a0d4cbce2b4b22e88b9b...
I am Satoshi Nakamoto13 => 0ebc56d59a34f5082aaef3d66b37a66...
I am Satoshi Nakamoto14 => 27ead1ca85da66981fd9da01a8c6816...
I am Satoshi Nakamoto15 => 394809fb809c5f83ce97ab554a2812c...
I am Satoshi Nakamoto16 => 8fa4992219df33f50834465d3047429...
I am Satoshi Nakamoto17 => dca9b8b4f8d8e1521fa4eaa46f4f0cd...
I am Satoshi Nakamoto18 => 9989a401b2a3a318b01e9ca9a22b0f3...
I am Satoshi Nakamoto19 => cda56022ecb5b67b2bc93a2d764e75f...
```

https://passwordsgenerator.net/sha1-hash-generator/

암호학적 해쉬 함수(Cryptographic Hash Function)

❒ Secure Hash Algorithm (SHA) groups

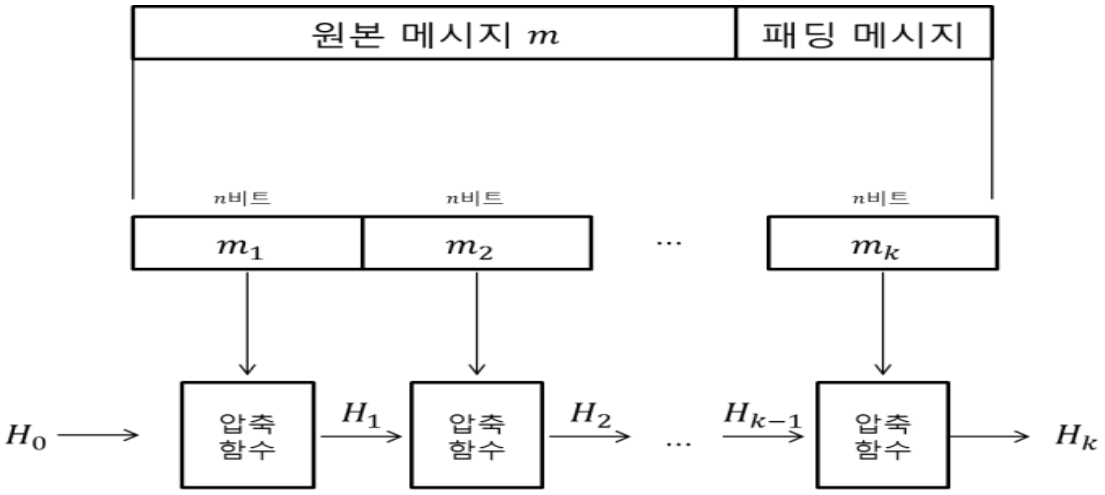
- 1993년 미국 국가 안전 보장국(NSA)과 미국 표준 기술연구소(NIST)가 함께 설계
- SHA-1 : SHA-0의 취약점을 보강, 160 비트 해쉬 값
- 2002년 SHA-2계열 (SHA-256, SHA-384, SHA-512) 설계

Characteristics	*SHA-1*	*SHA-224*	*SHA-256*	*SHA-384*	*SHA-512*
Maximum Message size	$2^{64} - 1$	$2^{64} - 1$	$2^{64} - 1$	$2^{128} - 1$	$2^{128} - 1$
Block size	512	512	512	1024	1024
Message digest size	160	224	256	384	512
Number of rounds	80	64	64	80	80
Word size	32	32	32	64	64

암호학적 해쉬 함수(Cryptographic Hash Function)

❒ 암호학적 해쉬 함수의 설계 원리

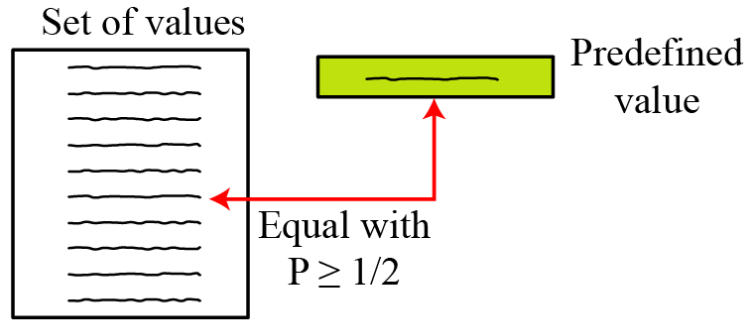
- Merkle-Damgard 구조: MD5, SHA-1, SHA-2 등
  - n비트의 배수가 되도록 (m || 패딩)
  - (m || 패딩)을 한 블록이 n비트가 되도록 분할 $m_1, m_2, \ldots, m_k$
  - $f(H_0, m_1) = H_1$
    - 초기 값 H_0, 압축 함수 $f(\cdot)$
    - k번 수행하여 H_k 를 생성 → $h(m) = H_k$



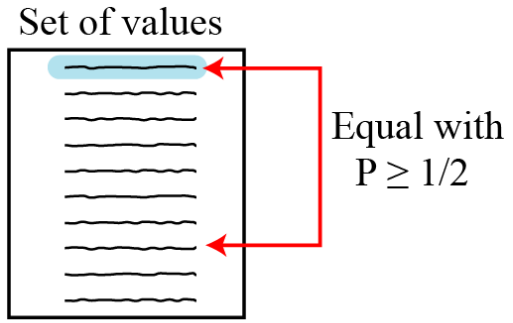
H_0 : 초기값
H_t : 메시지 다이제스트(출력값)

암호학적 해쉬 함수(Cryptographic Hash Function)

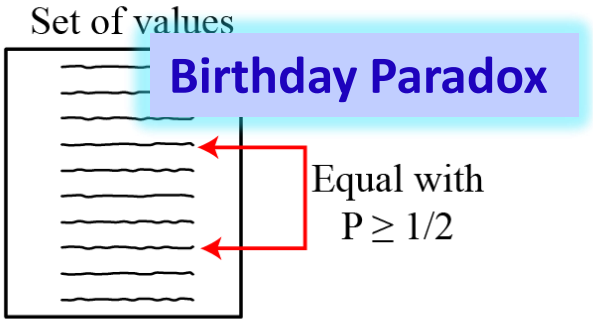
- 암호학적 해쉬 함수는 3가지 성질 중 공격자가 가장 공격하기 쉬운 Collsision Resistance에 대한 공격으로부터 안전해야함
  - 암호학적 해쉬 함수의 안전성은 생일 문제(Birthday Problems)와 연관이 있음



a. First problem



b. Second problem



c. Third problem

암호학적 해쉬 함수(Cryptographic Hash Function)

❒ 암호학적 해쉬 함수의 안전성: 생일 문제(Birthday Problems)

- In a room of just 23 people, there´s a 50% chance of at least two people having the same **birthday**
  - Event A: At least two people in the room have the same birthday
  - Pr(A) = 1 – Pr(A')

$$P(A') = \frac{365}{365} \times \frac{364}{365} \times \frac{363}{365} \times \frac{362}{365} \times \cdots \times \frac{343}{365}$$

$$P(A') = \left(\frac{1}{365}\right)^{23} \times (365 \times 364 \times 363 \times \cdots \times 343) \approx 0.492703$$

$$P(A) \approx 1 - 0.492703 = 0.507297 \ (50.7297\%) \qquad \leftarrow \sqrt{\alpha \times 365} \approx 23$$

  - 모수가 N일때, N이 충분히 크다면 생일 문제 (N개중 동일한 1개를 선택하는 사람이 50%이상의 확률로 2명이상 존재하는 경우)를 만족하는 샘플의 수 (e.g., 날짜를 선택하는 사람의 수)는 $\sqrt{N}$으로 수렴함

- In a room of 75, there´s a 99.9% chance of at least two people matching

암호학적 해쉬 함수(Cryptographic Hash Function)

❒ 암호학적 해쉬 함수의 안전성: 생일 문제(Birthday Problems)

- 만약 h(x)의 결과 값이 N비트라면, Output의 경우의 수는 2^N개임

Like Birthday problem's 365 days

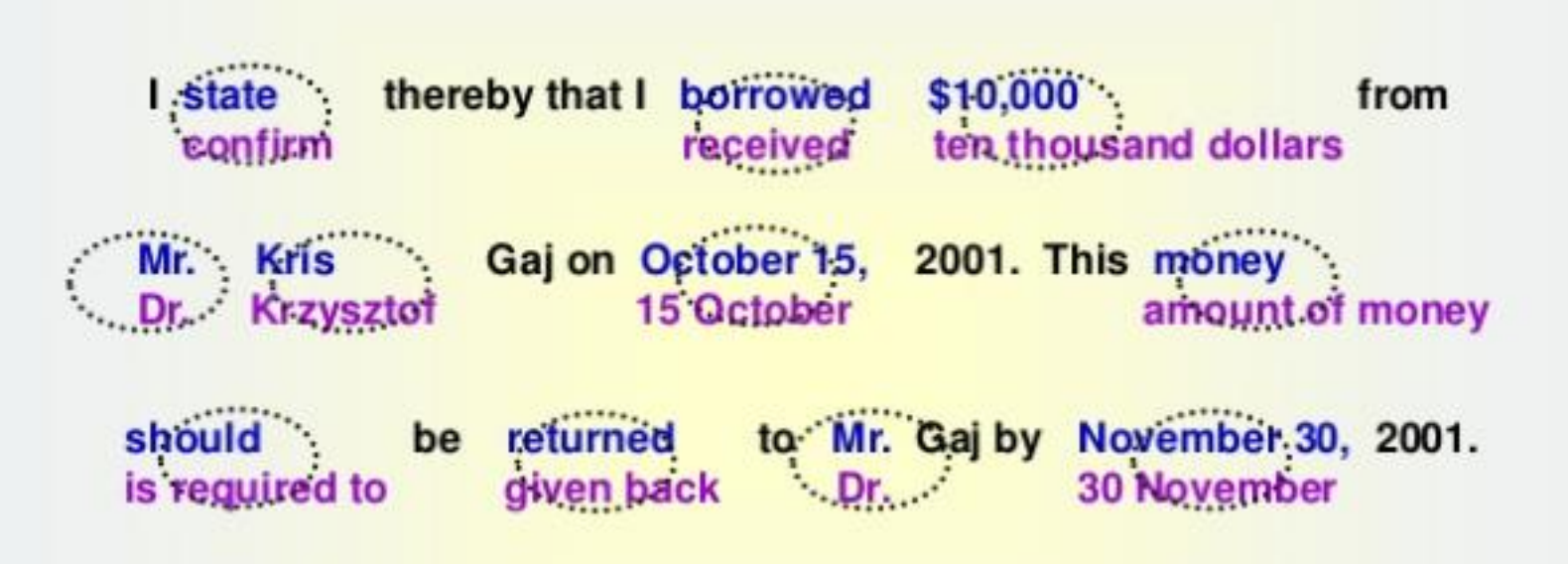
Birthday problem's $\sqrt{\alpha \times 365} \approx 23$ people

- 따라서, $2^{\frac{N}{2}}$개의 해쉬함수 입력값, 결과 값 쌍을 가지고 있으면, 50%이상의 확률로 충돌쌍을 찾을 수 있음

암호학적 해쉬 함수(Cryptographic Hash Function)

❒ 암호학적 해쉬 함수의 안전성: 생일 문제(Birthday Problems)

- How to generate hash output values



(Source: https://www.slideshare.net/TonyNguyen197/hashfunction)

11 different positions of similar expressions

2^{11} different messages of the same meaning

암호학적 해쉬 함수(Cryptographic Hash Function)

- 암호학적 해쉬 함수에 대한 충돌쌍 공격
  - 과거에는 64비트 Ouput을 가지는 암호학적 해쉬 함수가 안전하다고 생각됨

- 그러나 컴퓨터의 발달로 64비트 암호학적 해쉬 함수는 쉽게 충돌쌍을 찾을 수 있게 됨
  - $2^{64}/2 = 2^{32}$ tests to launch an attack with probability 1/2 or more.
  - With 2^{20} (one million) tests/sec, an attack in $2^{32}/2^{20} = 2^{12}$ seconds (almost an hour)

X1: Bob sends $5,000 to Alice
X2: Bob transfer 5,000 won to Alice

If H(X1) = H(X2), can we use a cryptographic hash function as a digital fingerprint?

암호학적 해쉬 함수(Cryptographic Hash Function)

- SHA-1은 160비트의 Output을 가지고 있음
  - To launch a collision attack, test $2^{160}/2 = 2^{80}$ tests. With 2^{30} (more than one billion) tests/sec, it takes 2^{50} seconds (more than 1000-yrs)

- In 2005, security flaws were identified in SHA-1, a possible weakness might exist, indicating a stronger hash function would be desirable.
  - Merkle-Damgard 구조로 인한 취약점
  - SHA-2도 SHA-1과 동일한 구조로 설계되었으므로 취약점이 존재했음

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Announcing the first SHA1 collision

February 23, 2017

암호학적 해쉬 함수(Cryptographic Hash Function)

❒ SHA-3 (Secure Hash Algorithm 3) 공모

- 2008년 : 총 51개의 해쉬 함수가 1차 후보로 발표
- 2009년 : 2차 라운드를 통해 14개 후보로 압축
- 2010년 12월 : 5개의 해쉬 함수가 최종 후보로 발표
- 2012년 10월 : Guido Bertoni, Joan Daemen, Gilles Van Assche (STMicroelectronics), , Michaël Peeters (NXP Semiconductors)가 제안한 Keccak (pronounced "catch-ack") 알고리즘 최종 선정
- 160, 224, 256, 384, 512비트 해쉬 값

NIST hash function competition

https://en.wikipedia.org/wiki/Secure_Hash_Algorithms

Appendix #1: KISA 가이드라인

- https://www.kisa.or.kr/public/laws/laws3_View.jsp?mode=view&p_No=259&b_No=259&d_No=82&ST=total&SV=

〈표 4〉 보안강도에 따른 단순해쉬/전자서명용 해쉬함수 분류

보안강도	NIST(미국)	CRYPTREC(일본)	ECRYPT(유럽)	국내
80 비트 이상	SHA-1 SHA-224/256/384/512	SHA-1 SHA-256/384/512 RIPEMD-160	SHA-1 SHA-224/256/384/512 RIPEMD-160 Whirlpool	HAS-160[2] SHA-1[3] SHA-224/256/384/512
112 비트 이상	SHA-224/256/384/512	SHA-256/384/512	SHA-224/256/384/512 Whirlpool	SHA-224/256/384/512
128 비트 이상	SHA-256/384/512	SHA-256/384/512	SHA-256/384/512 Whirlpool	SHA-256/384/512
192 비트 이상	SHA-384/512	SHA-384/512	SHA-384/512 Whirlpool	SHA-384/512
256 비트 이상	SHA-512	SHA-512	SHA-512	SHA-512

NIST
(2030년 이상 사용가능)

SHA-1 : 충돌저항성(안전성)이 80비트 보안강도 이하를 제공하여(Crypto'05), 새로운 어플리케이션에 적용하는 것을 권장하지 않지만 현재 광범위하게 사용되므로 해쉬함수 보안강도 표에 추가하였음

Appendix #2: 암호학적 해쉬 함수 활용

❐ 데이터/파일에 대한 무결성

Download Kali Linux Images

We generate fresh Kali Linux image files every few months, which we make available for download. This page provides the links to **download Kali Linux** in its latest official release. For a release history, check our Kali Linux Releases page. Please note: You can find unofficial, untested weekly releases at http://cdimage.kali.org/kali-weekly/.

Image Name	Download	Size	Version	sha256sum
Kali Linux 64 Bit	HTTP | Torrent	2.8G	2018.2	56f677e2edfb2efcd0b08662ddde824e254c3d53567ebbbcdbbf5c03efd9bc0f
Kali Linux Light 64 Bit	HTTP | Torrent	865M	2018.2	554f020b0c89d5978928d31b8635a7eeddf0a3900abcacdbc39616f80d247f86
Kali Linux E17 64 Bit	HTTP | Torrent	2.6G	2018.2	be0a858c4a1862eb5d7b8875852e7d38ef852c335c3c23852a8b08807b4c3be8
Kali Linux Lxde 64 Bit	HTTP | Torrent	2.6G	2018.2	449ecca86b0f49a52f95a51acdde94745821020b7fc0bd2129628c56bc2d145d
Kali Linux Xfce 64 Bit	HTTP | Torrent	2.6G	2018.2	0e94035a0a56fccc49961b0da56b9243ed3da6a3f8d696884e6f0b936f74dbfb

Appendix #2: 암호학적 해쉬 함수 활용

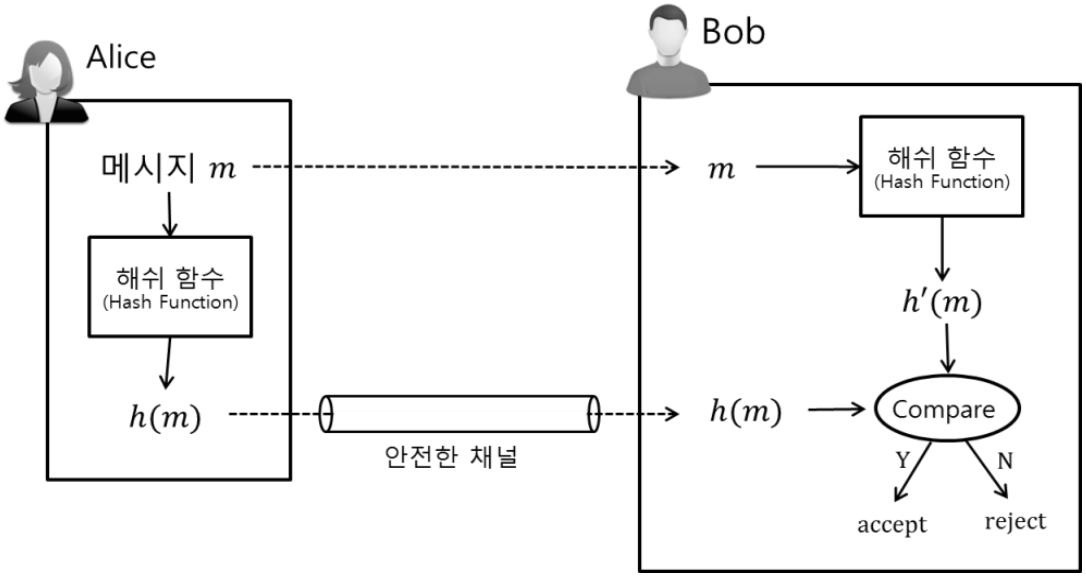
- 패스워드 관리
  - 리눅스 패스워드 관리를 위한 shadow파일

- 그 밖에도 전자서명, 블록체인 등 다양한 암호 기반 기술에 사용됨

메시지 변조 감지 코드(MDC)

❒ MDC (Message Modification Code)

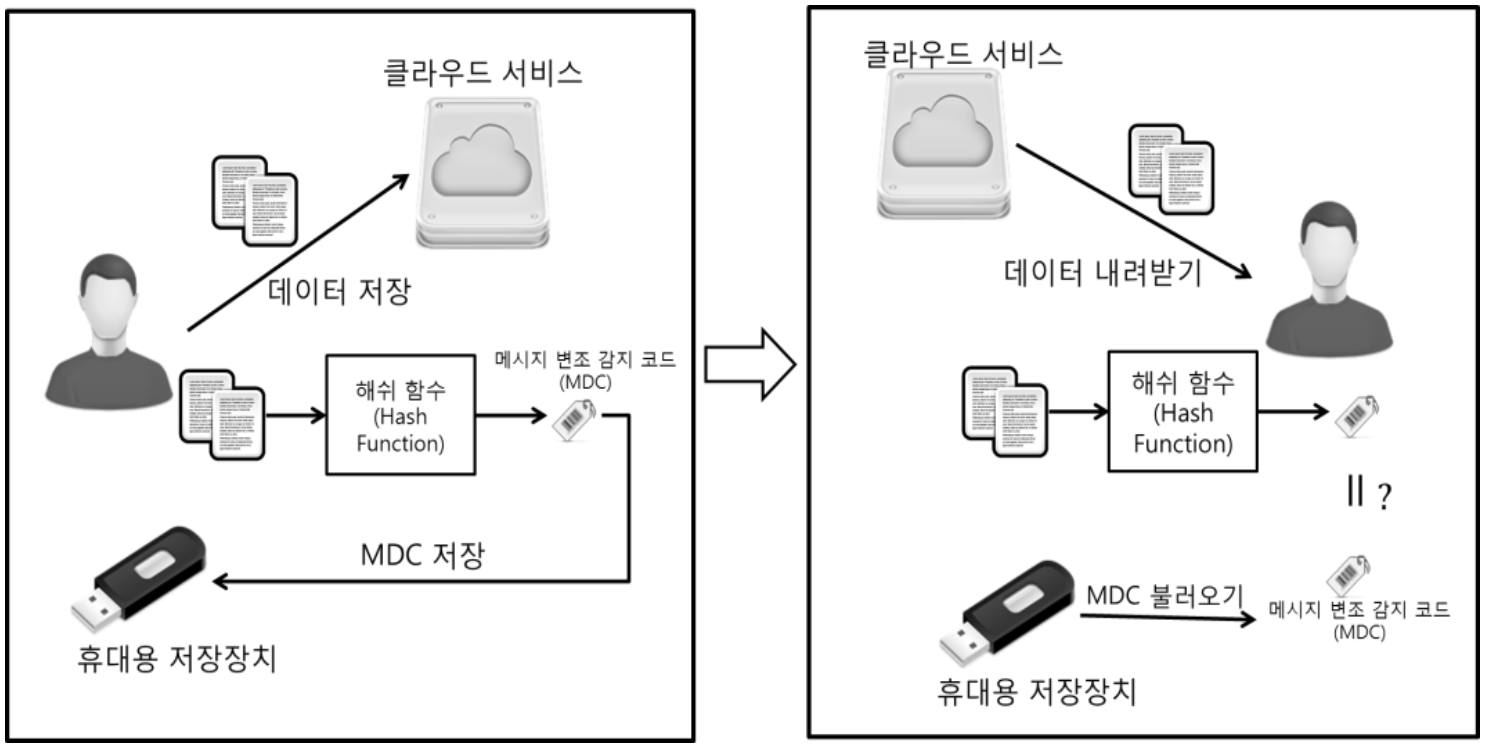
- 메시지 m과 MDC h(m)은 분리될 수 있으며, h(m)은 변조되지 않도록 하는 것이 중요!
  - Alice는 메시지 m에 대한 MDC h(m) 생성
  - Alice → Bob : m, h(m), 단 h(m)은 안전한 채널로 전송



메시지 변조 감지 코드(MDC)

❒ MDC를 이용한 메시지 변조 감지 코드 활용 예

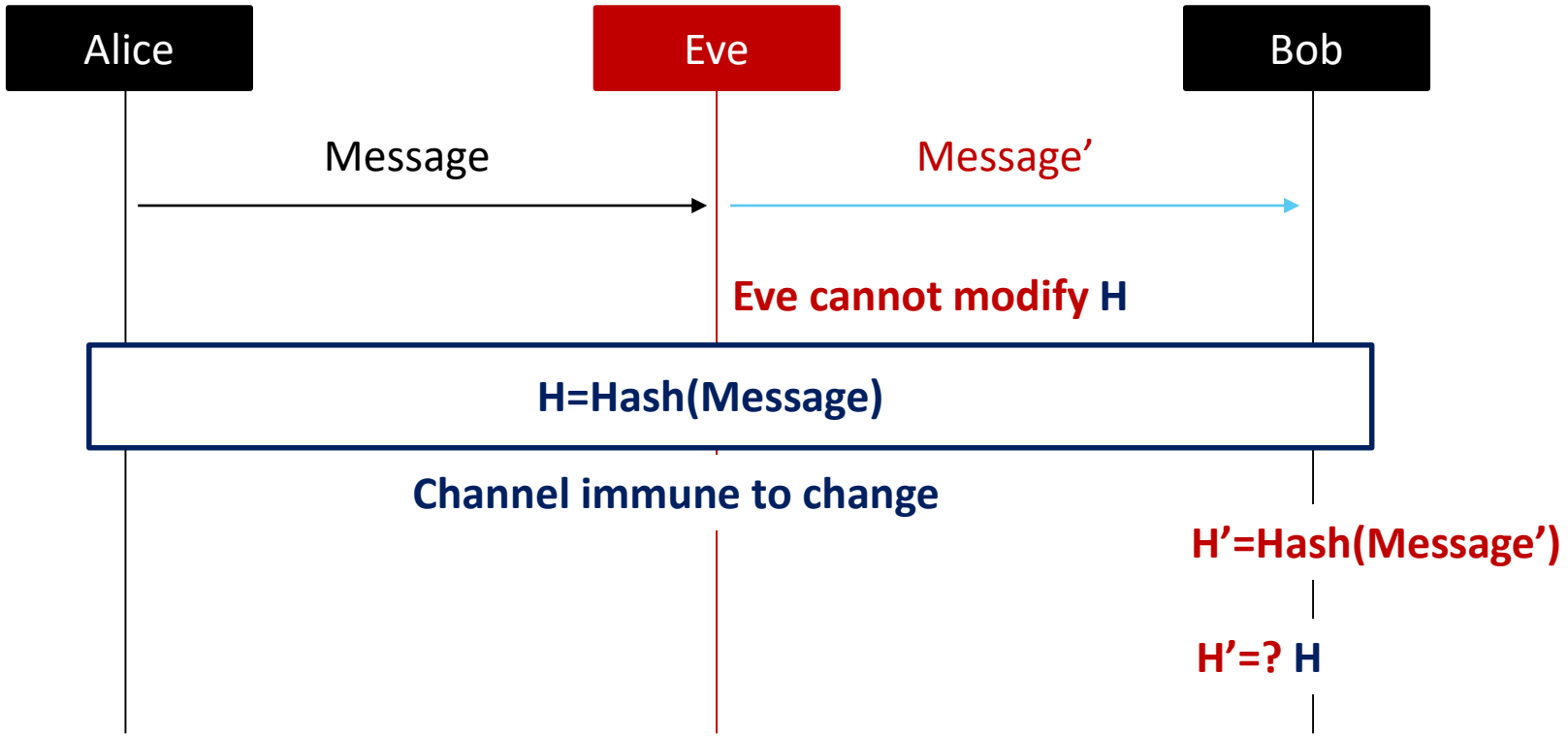
- 클라우드 서비스에서 원본 메시지에 대한 무결성을 제공



메시지 변조 감지 코드(MDC)

❒ MDC

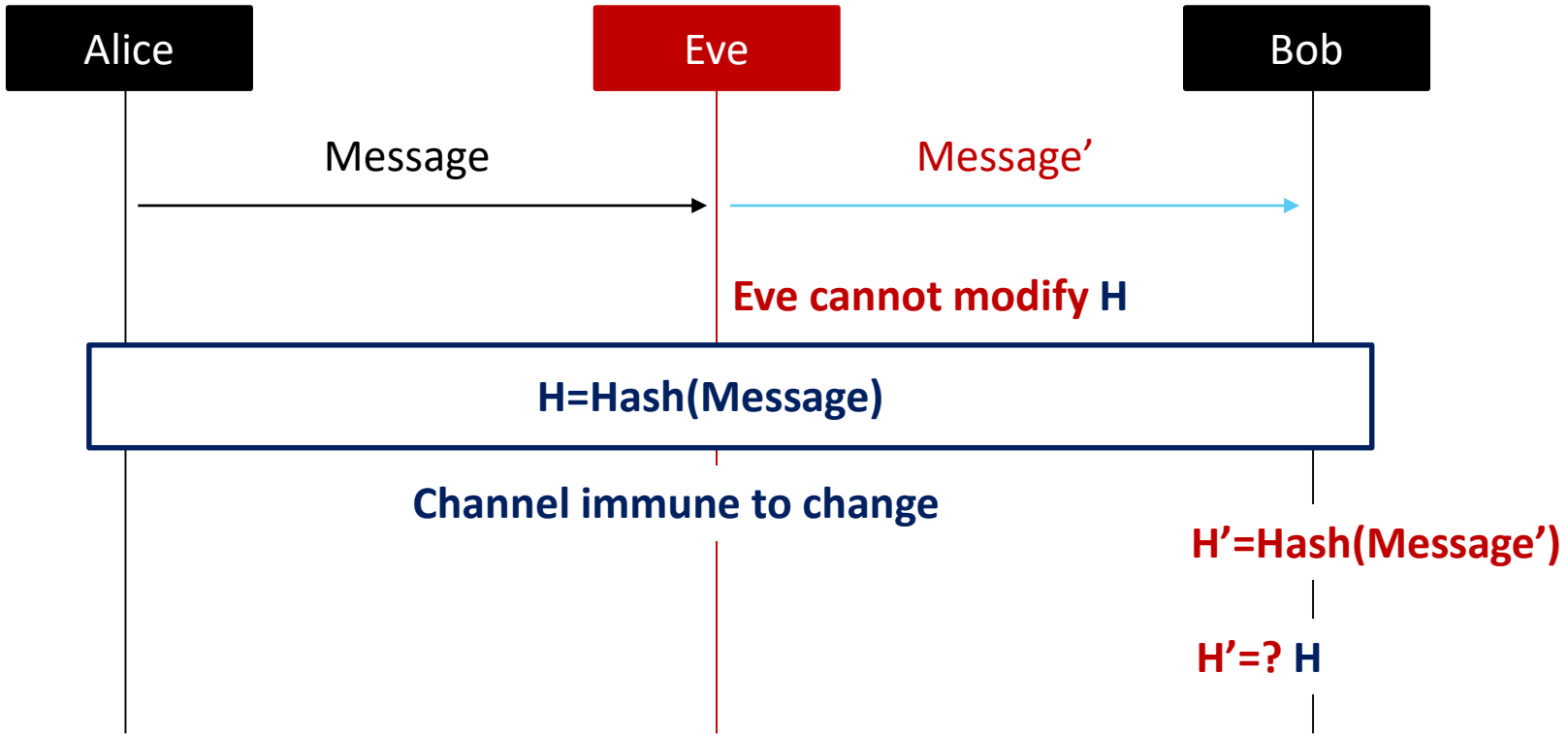
- There is a channel immune to message change between Alice and Bob



메시지 변조 감지 코드(MDC)

❒ MDC

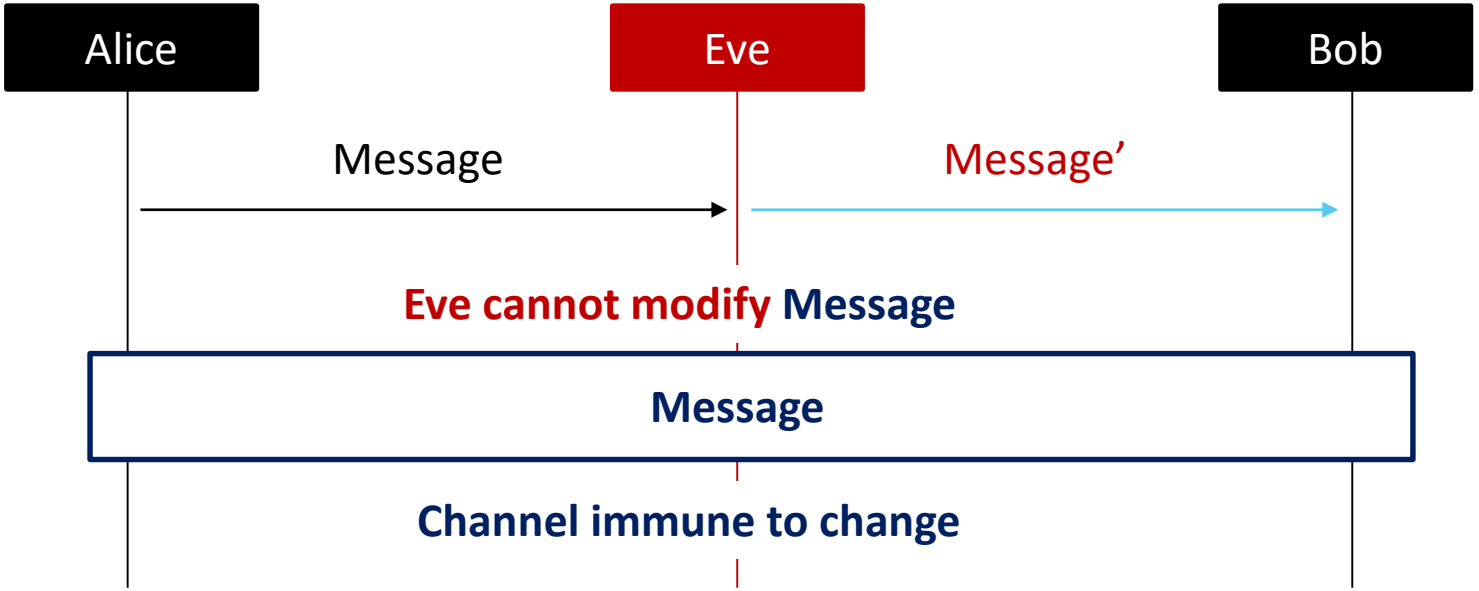
- There is a channel immune to message change between Alice and Bob



Eve cannot find Message' such that Hash(Message') = **Hash(Message)**

메시지 변조 감지 코드(MDC)

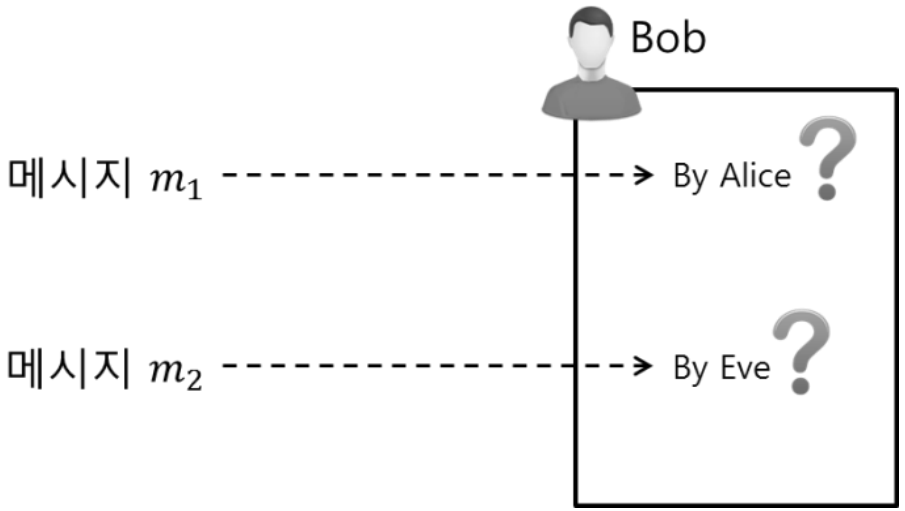
- But, MDC is impractical
  - Alice can just send a message using the channel immune to change
  - Setup of the channel is very difficult and not cost-effective



메시지 인증코드 (MAC)

❒ Message Authentication Code(MAC)

- 메시지 근원 인증 (Message origin authentication)을 제공
- 해쉬 함수를 이용한 MDC는 무결성만을 제공 → 메시지의 출처를 확인할 수 없음
- MAC은 무결성도 함께 제공

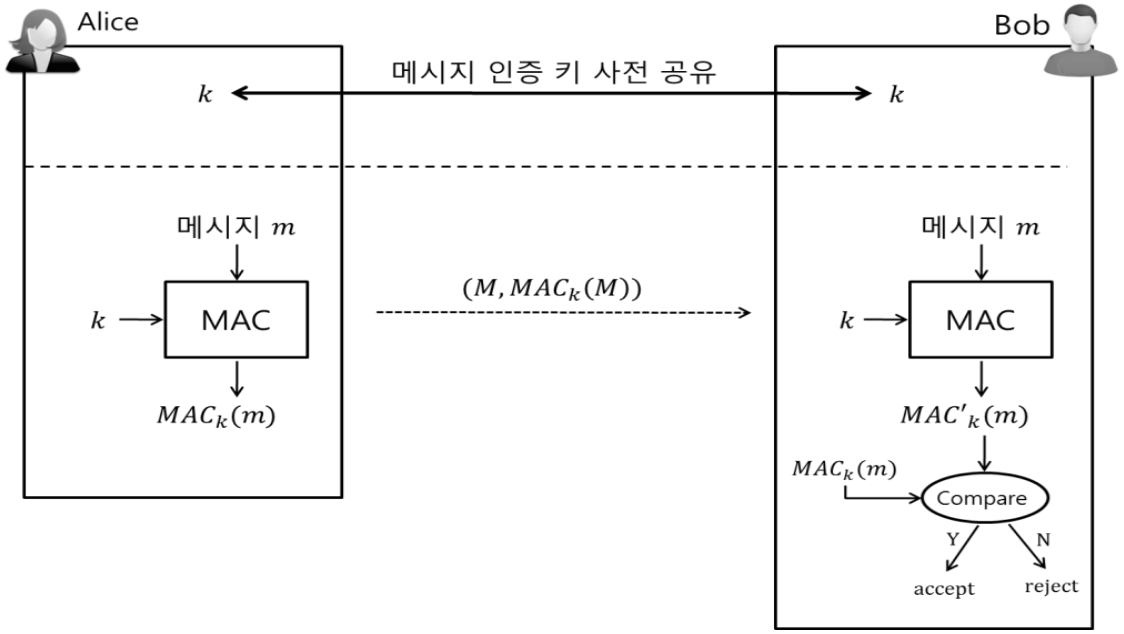


메시지 인증코드 (MAC)

❒ Message Authentication Code(MAC) :

- $\mathrm{MAC}_k(\mathrm{m}) = \mathrm{h}(k, \mathrm{m})$

1. Alice와 Bob은 MAC 키 k를 사전공유
2. Alice는 메시지 m에 대한 $\mathrm{MAC}_k(\mathrm{m})$ 생성
3. Alice → Bob : m, $\mathrm{MAC}_k(\mathrm{m})$
4. Bob은 수신 메시지 m' 에 대한 $\mathrm{MAC}_k(\mathrm{m}')$ 생성
5. $\mathrm{MAC}_k(\mathrm{m})$ =? $\mathrm{MAC}_k(\mathrm{m}')$



메시지 인증코드 (MAC)

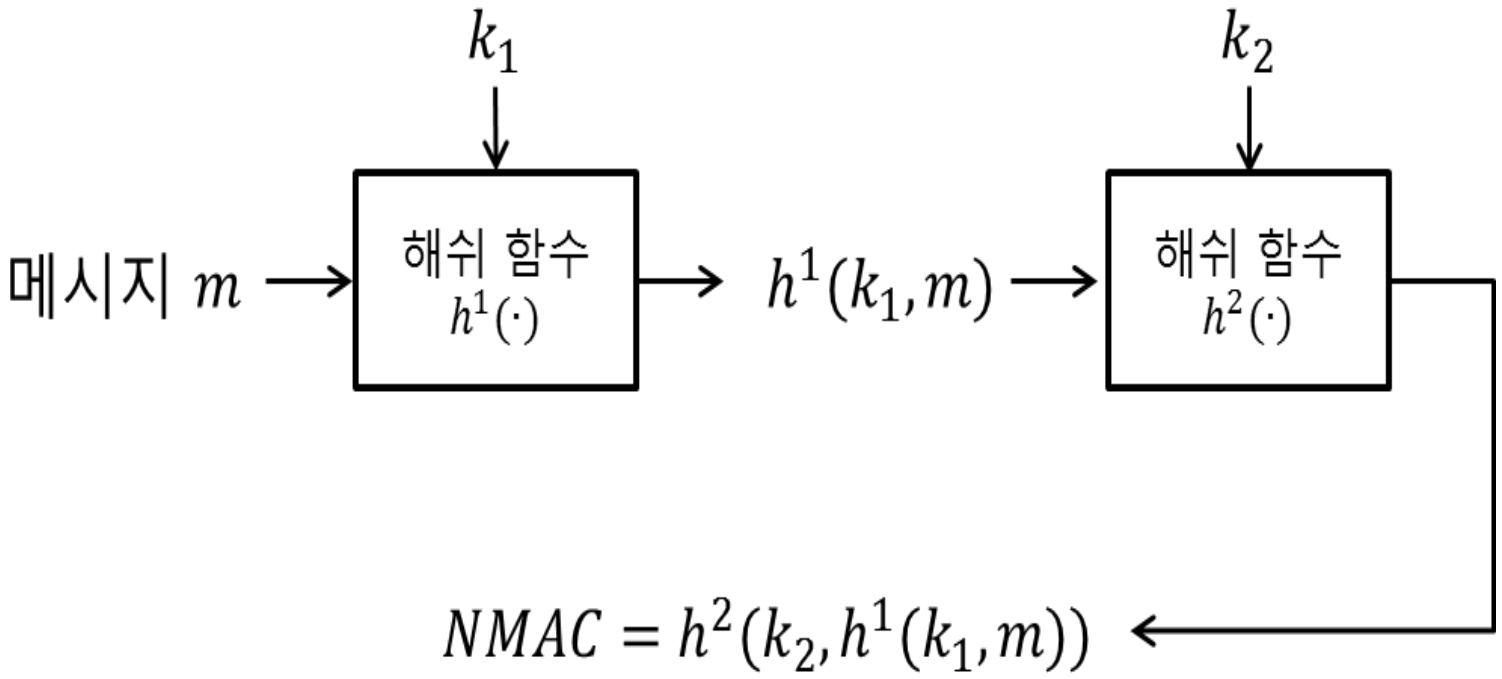
❒ MAC 생성 방법

해쉬 함수 기반 메시지 인증 코드	Nested MAC
	HMAC
블록 암호 알고리즘 기반 메시지 인증 코드	CBC-MAC
	CMAC

메시지 인증코드 (MAC)

❒ Nested MAC

- $NMAC_{k_1,k_2}(m) = h^2(k_2, h^1(k_1, m))$

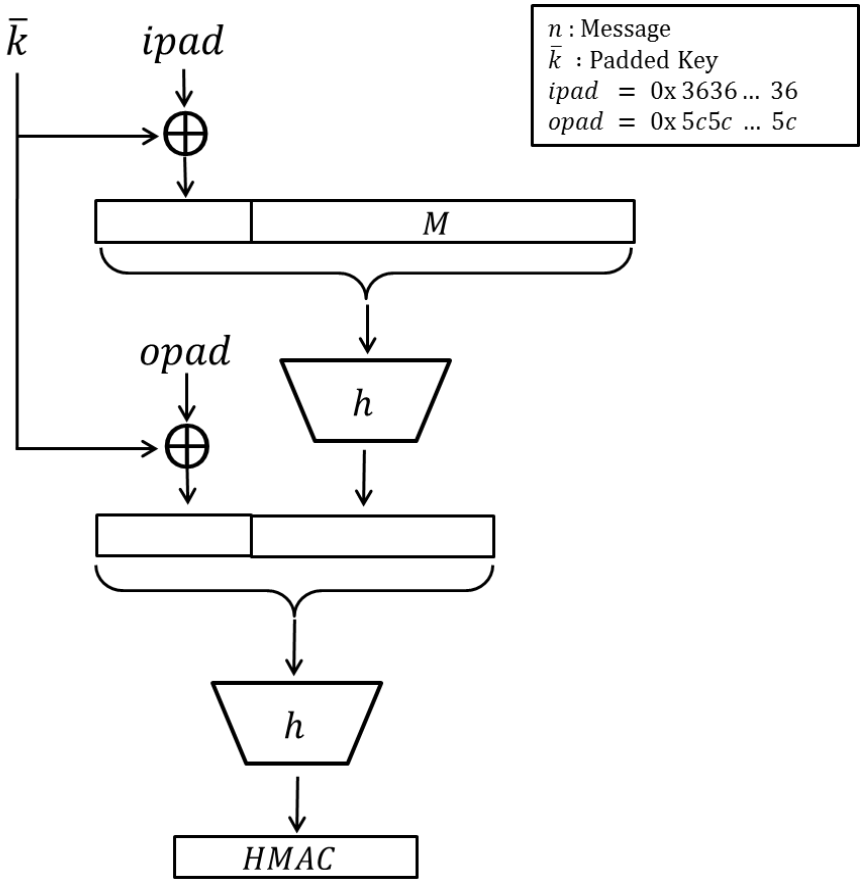


메시지 인증코드 (MAC)

❒ HMAC(Hashed MAC)

- **NIST 표준 FIPS198**

1. 키 k 앞에 0으로 패딩 하여 b비트로 맞추어 $\bar{k}$ 생성
2. $HMAC_K = h[(\bar{k} \oplus opad) \,||\, Hash[(\bar{k} \oplus ipad)||M)]]$



$$HMAC = h(\bar{k} \oplus opad \,||\, h(\bar{k} \oplus ipad \,||\, m))$$

메시지 인증코드 (MAC)

HMAC(Hashed MAC)의 안전성

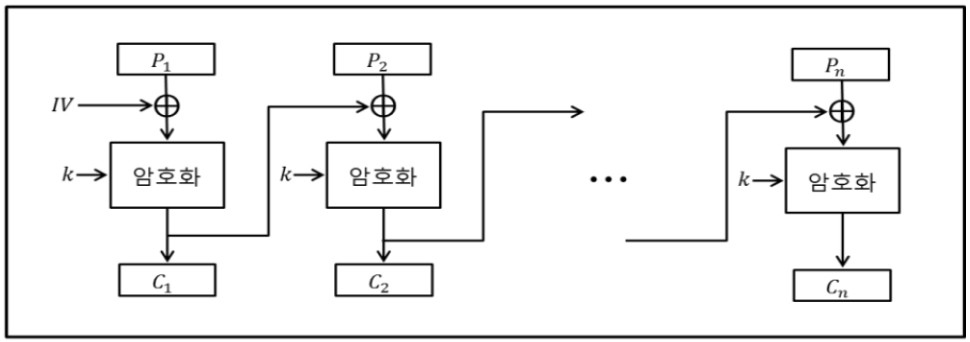
- 사용된 해쉬 함수의 안전성에 기반
- HMAC에 대한 공격
  - 사용된 MAC 키 전수 공격
  - 생일 공격
    - 키가 사용되었기 때문에 동일한 키로 생성된 $2^{n/2}$ 개의 HMAC값을 수집
    - 해쉬 함수 H는 공개되었기 때문에 MDC는 자유로이 생성할 수 있지만 MAC은 키가 사용되기 때문에 수동적으로 관찰해서 수집해야 함!

메시지 인증코드 (MAC)

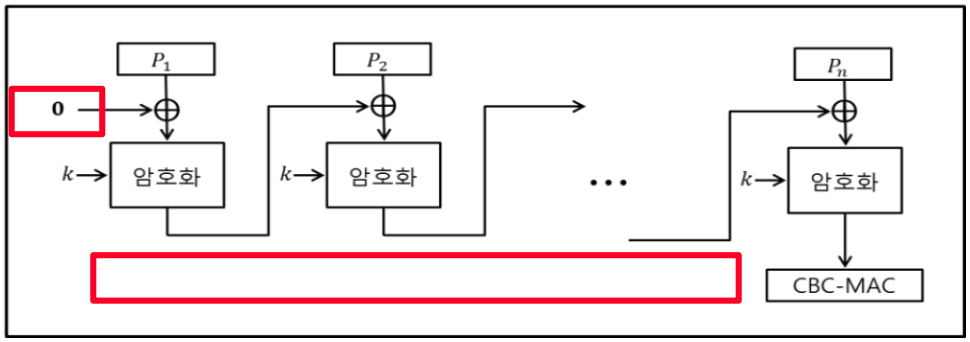
❒ CBC-MAC(Cipher Block Chaining MAC)

- 블록 암호의 운영모드 중 CBC모드를 사용하여 메시지 인증 코드를 생성 → 실제 환경에서 편리
  - 고정된 초기벡터(IV=0)를 사용
  - 고정된 메시지에 대한 MAC

CBC모드 암호화



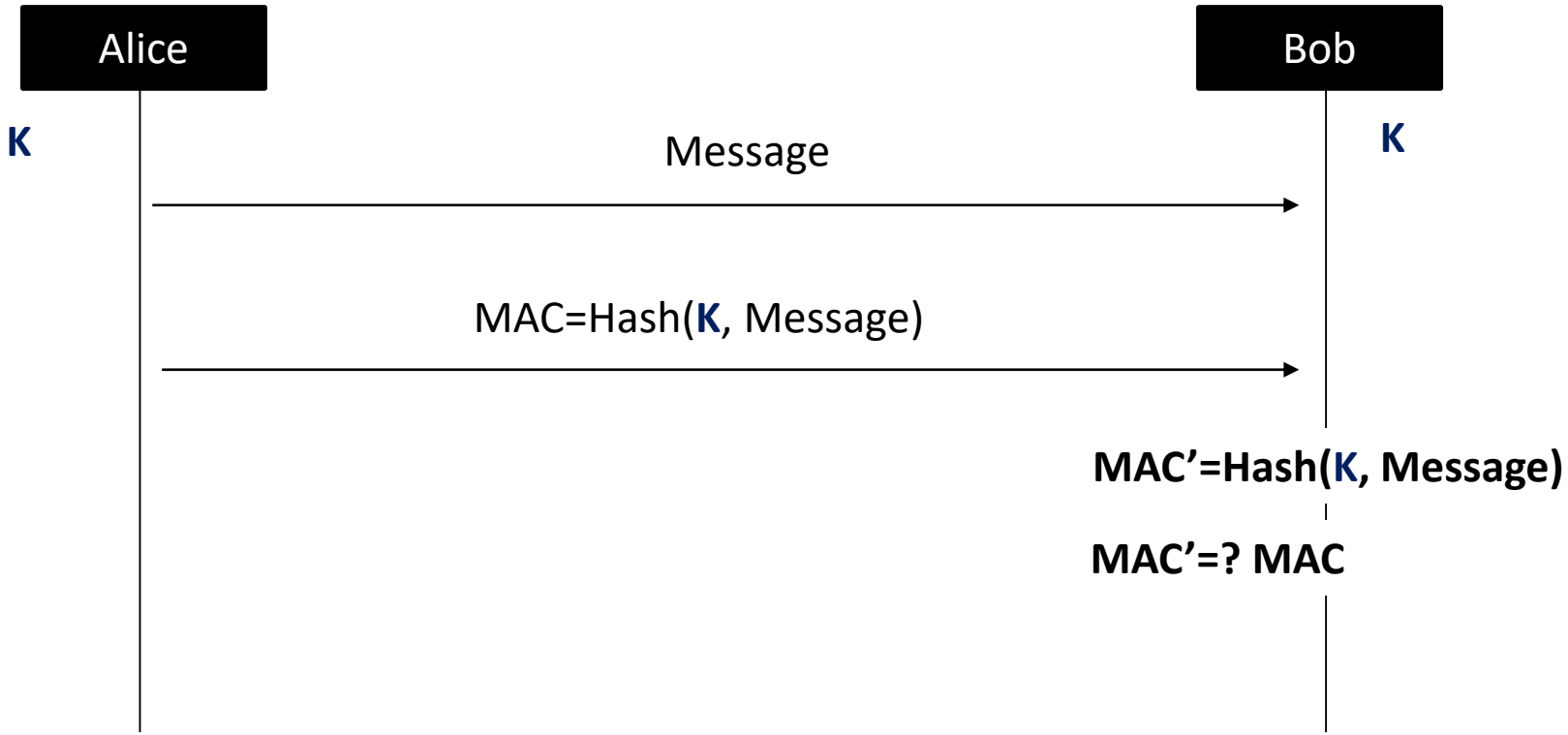
CBC-MAC



메시지 인증코드 (MAC)

❒ MAC

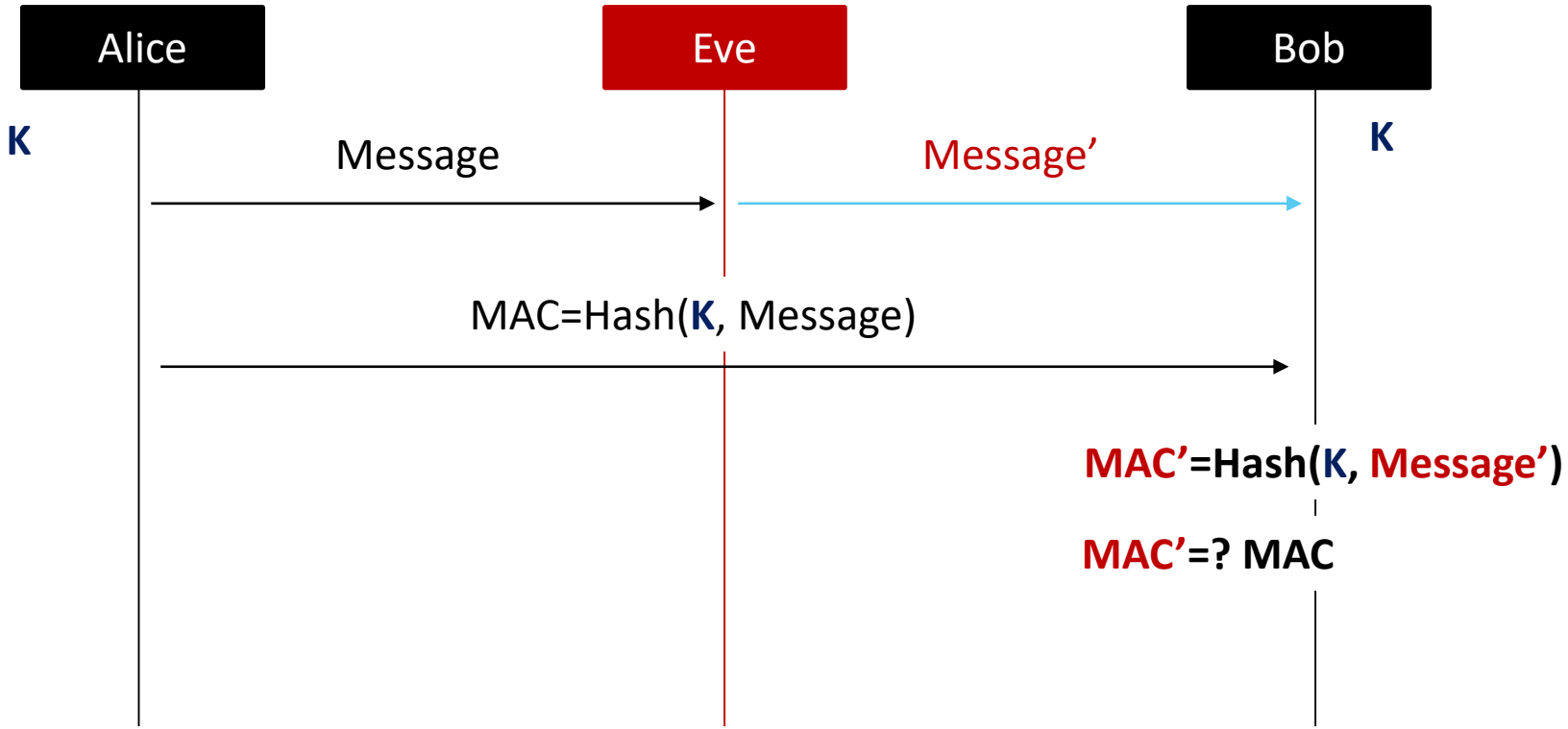
- Alice and Bob share a key for message authentication



메시지 인증코드 (MAC)

❒ MAC

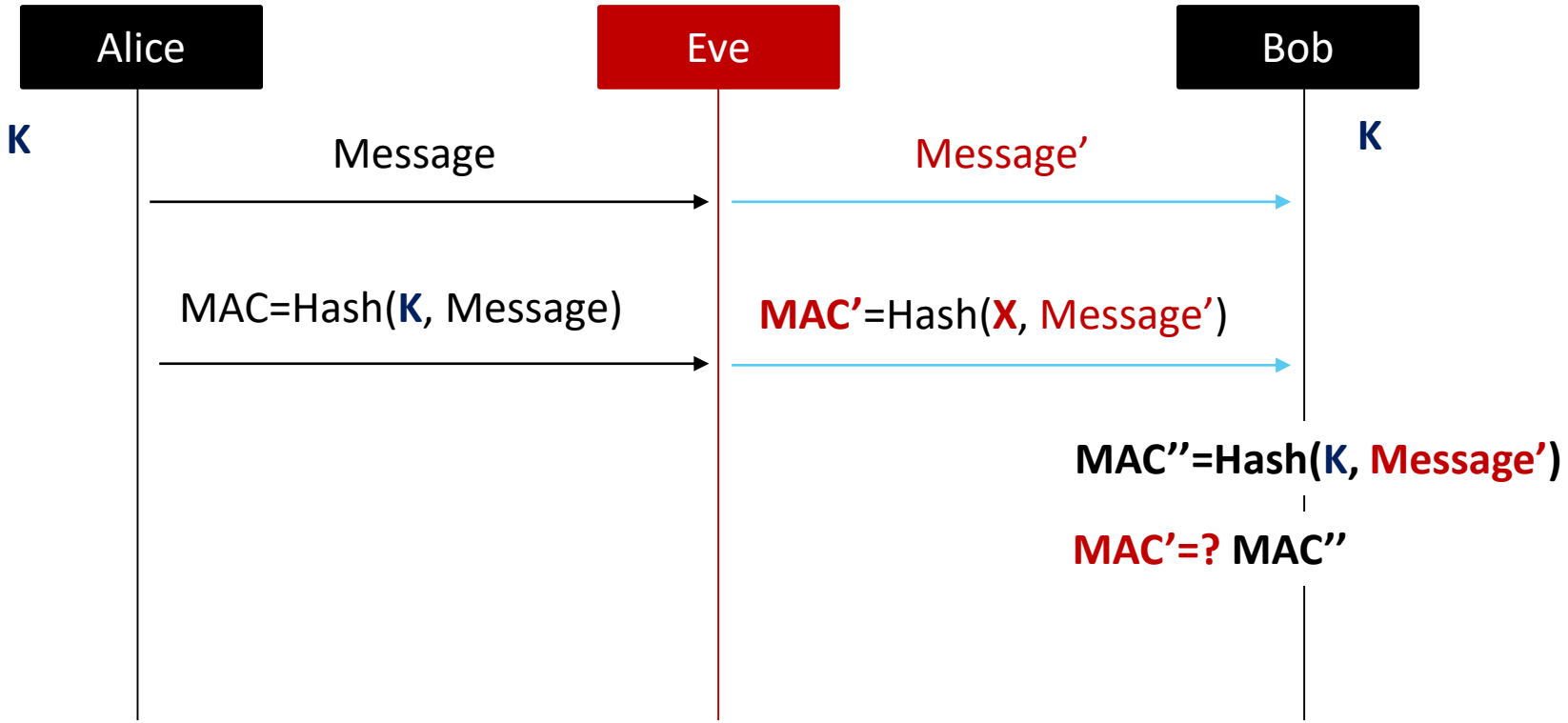
- Alice and Bob share a key for message authentication



메시지 인증코드 (MAC)

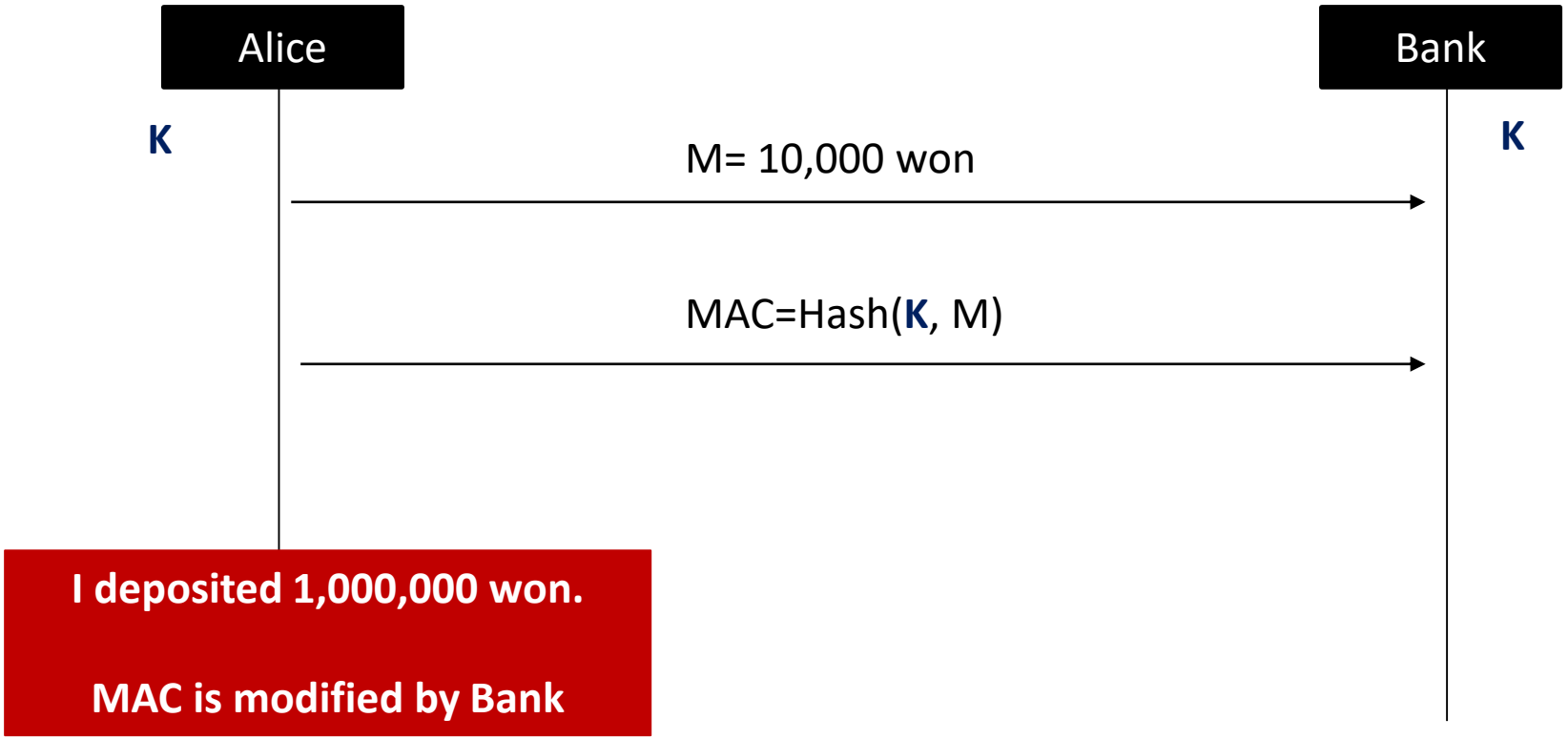
❒ MAC

- Alice and Bob share a key for message authentication



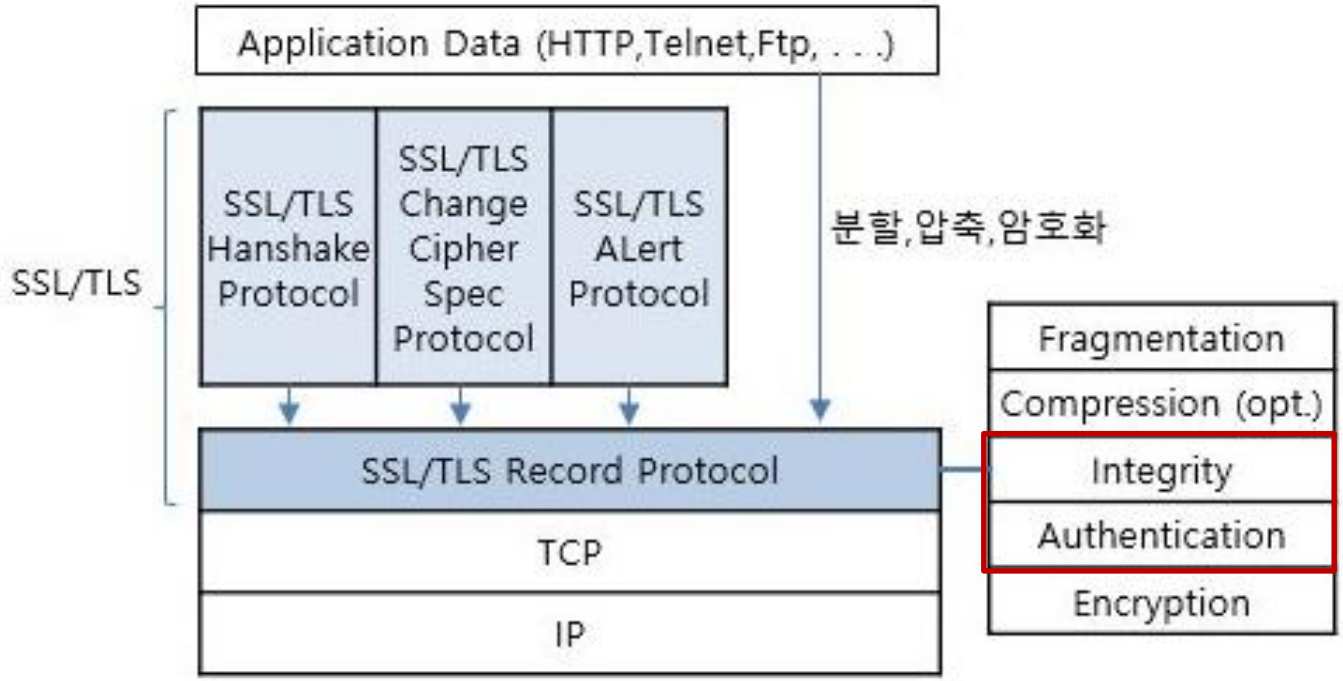
메시지 인증코드 (MAC)

- But, MAC does not provide non-repudiation (부인방지)
  - Because MAC values can be generated by entities who shares a key



Appendix #3: MAC의 활용

- HTTPS 패킷 무결성 및 인증에 MAC함수가 사용됨



http://www.ktword.co.kr/word/abbr_view.php?m_temp1=2552

HW #5

- 암호학적 해쉬 함수의 3가지 성질에 대해서 예를들어 설명하시오.

- 암호학적 해쉬 함수를 이용하는 MDC와 MAC의 차이점을 기술적인 관점에서 논하시오. (차이점 2개이상 언급)

- 4페이지 이내 (A4, 10 pt)
  - ★5월 10일 정오 12시까지★
  - 늦은 제출 시, 감점
  - 과제 copy 시, 관련된 과제들 모두 0점 처리
  - 제출양식: hwp or pdf

Thank you