2017/07/29, 2017 보안 기초 세미나

Network Security Essentials

- 3장 공개키 암호와 메시지 인증 (1) -

최창준 (changjun@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

목 차

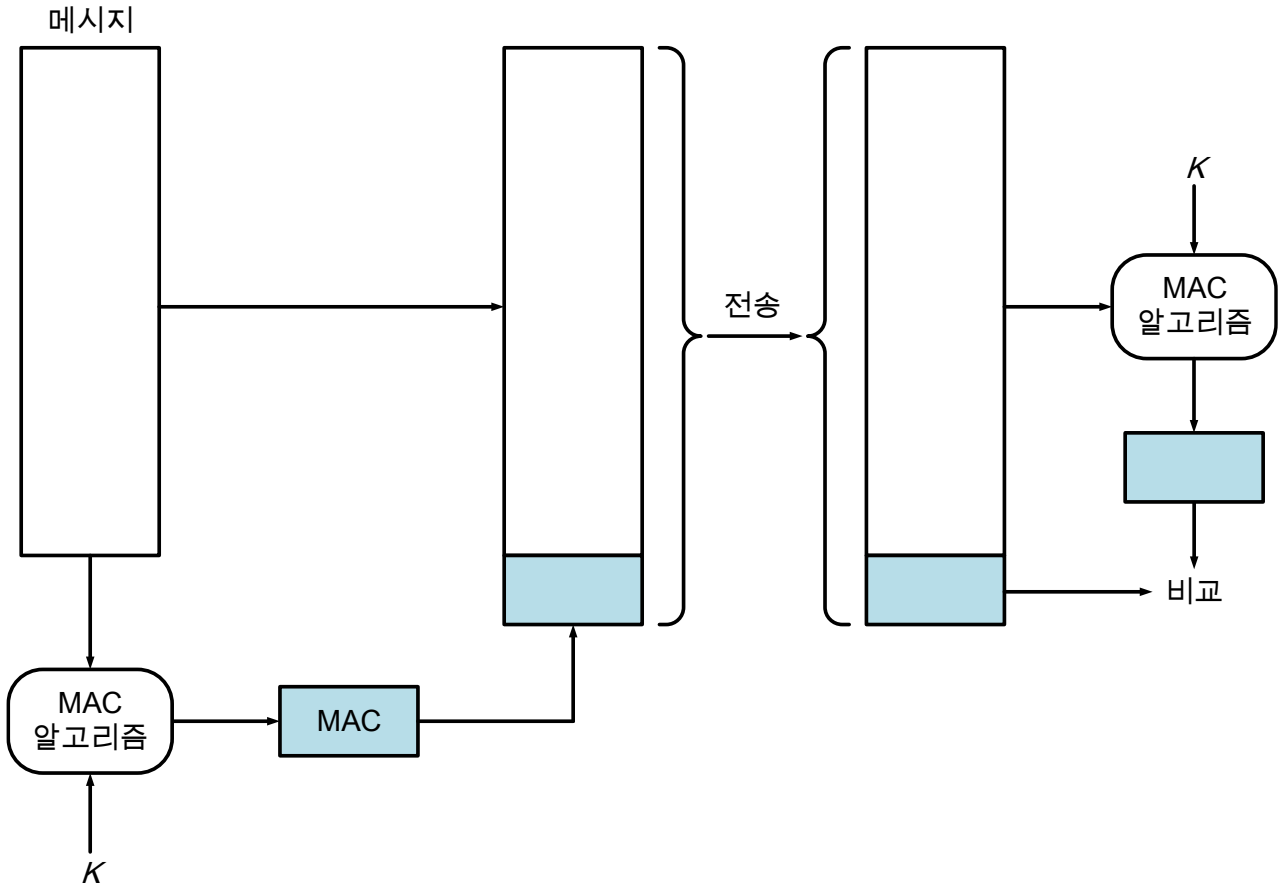
- 메시지 인증
  - 정의 및 인증 방법
- 해시 함수
  - 정의 및 특징
  - 암호학적 해시 함수 기준
  - 안전 해시 알고리즘
- 메시지 인증 코드
  - HMAC
  - 블록 암호기반 MAC

메시지 인증

- 메시지 인증(Message authentication)
  - 정의
    - 메시지 데이터 인증
      - 메시지, 파일, 문서 등 기타 메시지의 내용이 변경되지 않음을 확인
    - 메시지 출처 인증
      - 수신자가 받은 메시지가 진짜 송신자로부터 온 메시지인지를 확인
  - 인증 방법
    - 메시지를 암호화하여 메시지 인증
      - 통신 양측이 공유하는 키로 메시지를 암호화하여 인증
      - 수신자는 메시지의 오류 감지 코드, 순서 번호, 타임스탬프 등을 통해 메시지가 변경 혹은 지연되지 않음을 확인할 수 있음
    - 메시지 암호화 없이 메시지 인증
      - 인증 꼬리표를 생성하여 메시지 인증

메시지 인증

- 메시지 암호화 없이 인증
  - 메시지 인증 코드(MAC, Message Authentication Code)
    - 메시지의 무결성과 출처 확인을 검증할 목적으로 덧붙이는 코드



메시지 인증

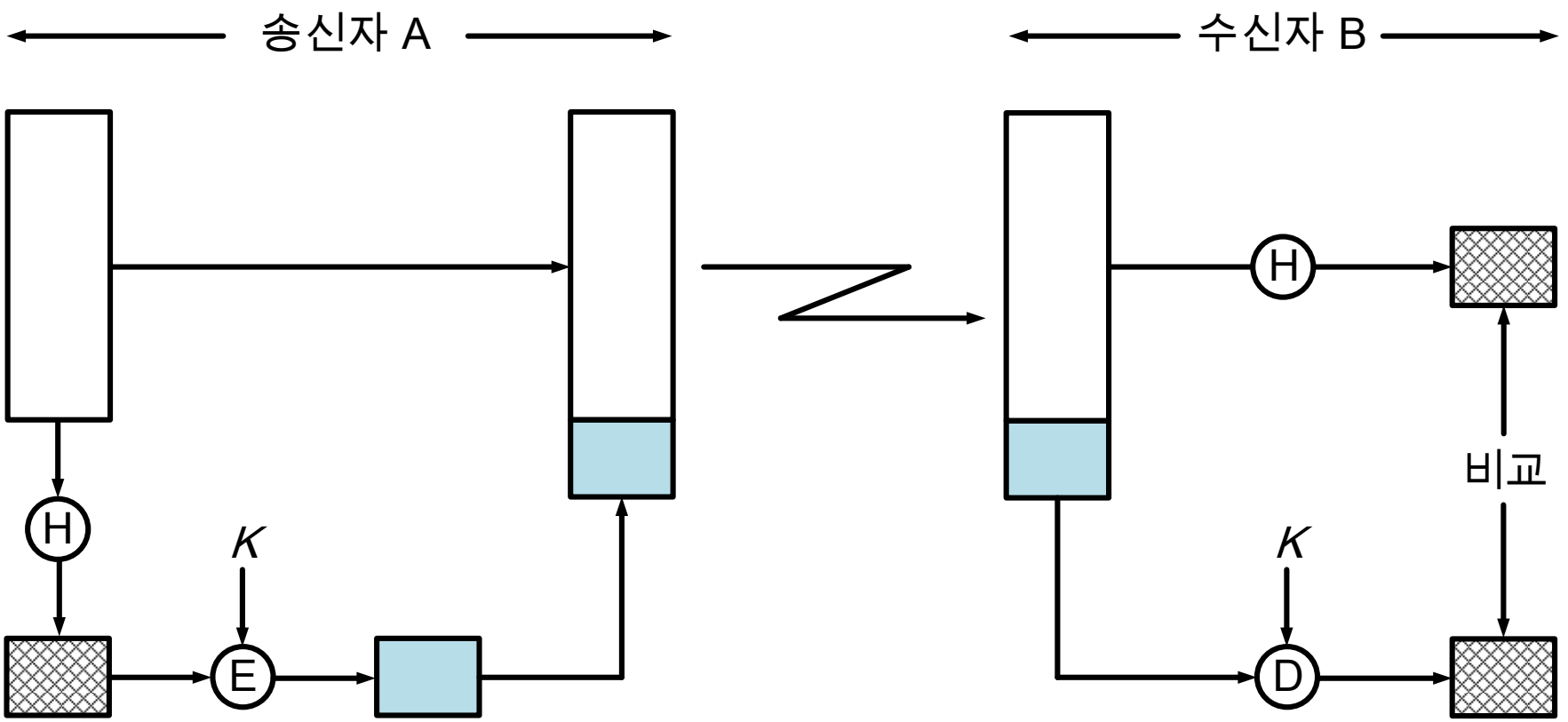
- 메시지 암호화 없이 인증
  - 메시지 인증 코드(MAC, Message Authentication Code)
    - 계산된 코드와 수신된 코드를 비교하여 두 값이 같은 경우
      - 수신자는 메시지가 변경되지 않았다는 것을 확인
      - 수신자는 진짜 송신자로부터 송신되었음을 확인
    - MAC 알고리즘
      - NIST의 FIPS PUB 113은 DES 알고리즘 사용을 권장
      - DES를 이용하여 메시지의 암호문을 생성하고 암호문 끝부분에 있는 여러 비트를 코드로 사용

메시지 인증

- 메시지 암호화 없이 인증
  - 일방향 해시 함수(One-way hash function)
    - 임의 크기의 메시지를 입력으로 일정한 크기의 해시 값을 출력하는 함수
    - 특징
      - 메시지 인증 코드(MAC)와는 다르게 입력으로 비밀 키를 사용하지 않음
      - 메시지를 인증하기 위해서 인증된 상태의 메시지 다이제스트를 메시지와 함께 전송

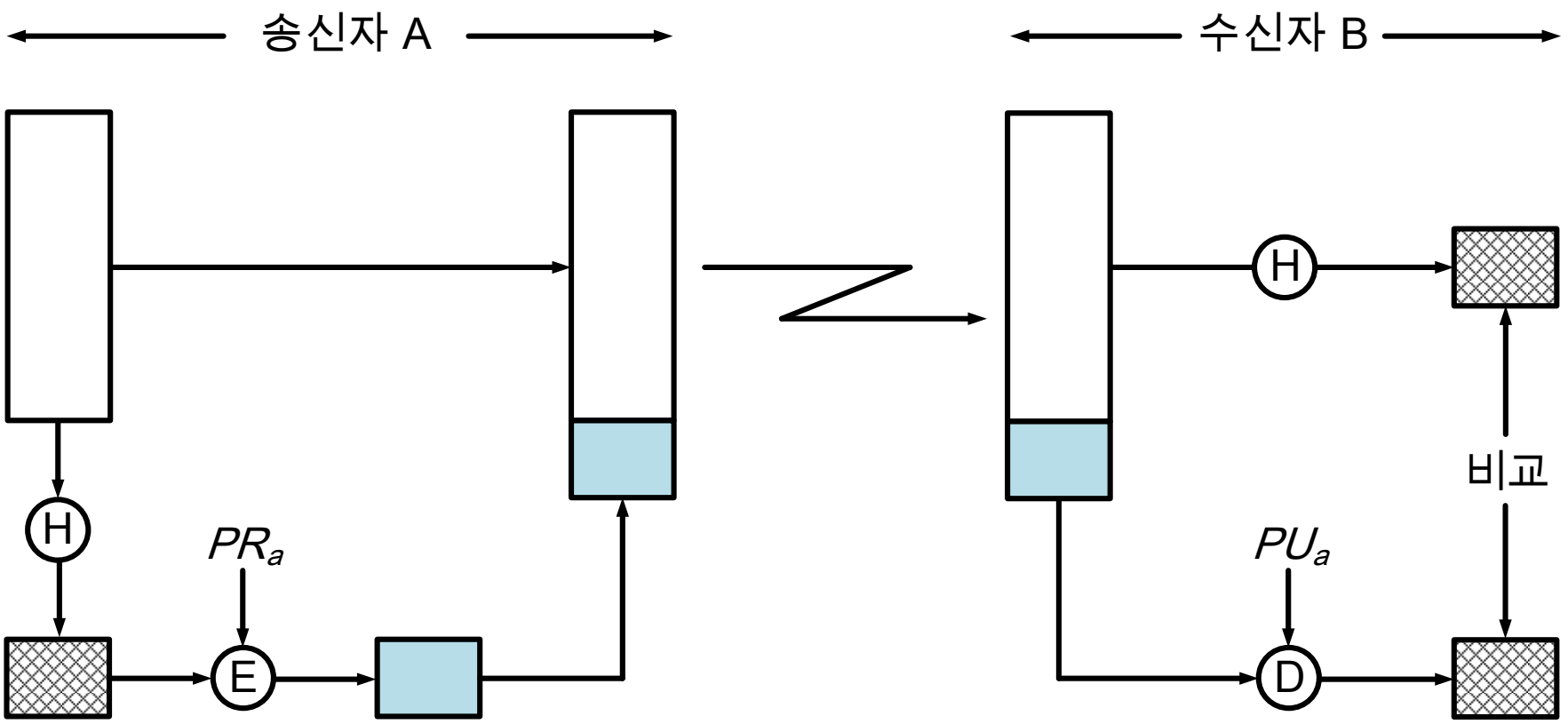
메시지 인증

- 메시지 암호화 없이 인증
  - 일방향 해시 함수
    - 관용 암호를 사용한 메시지 인증



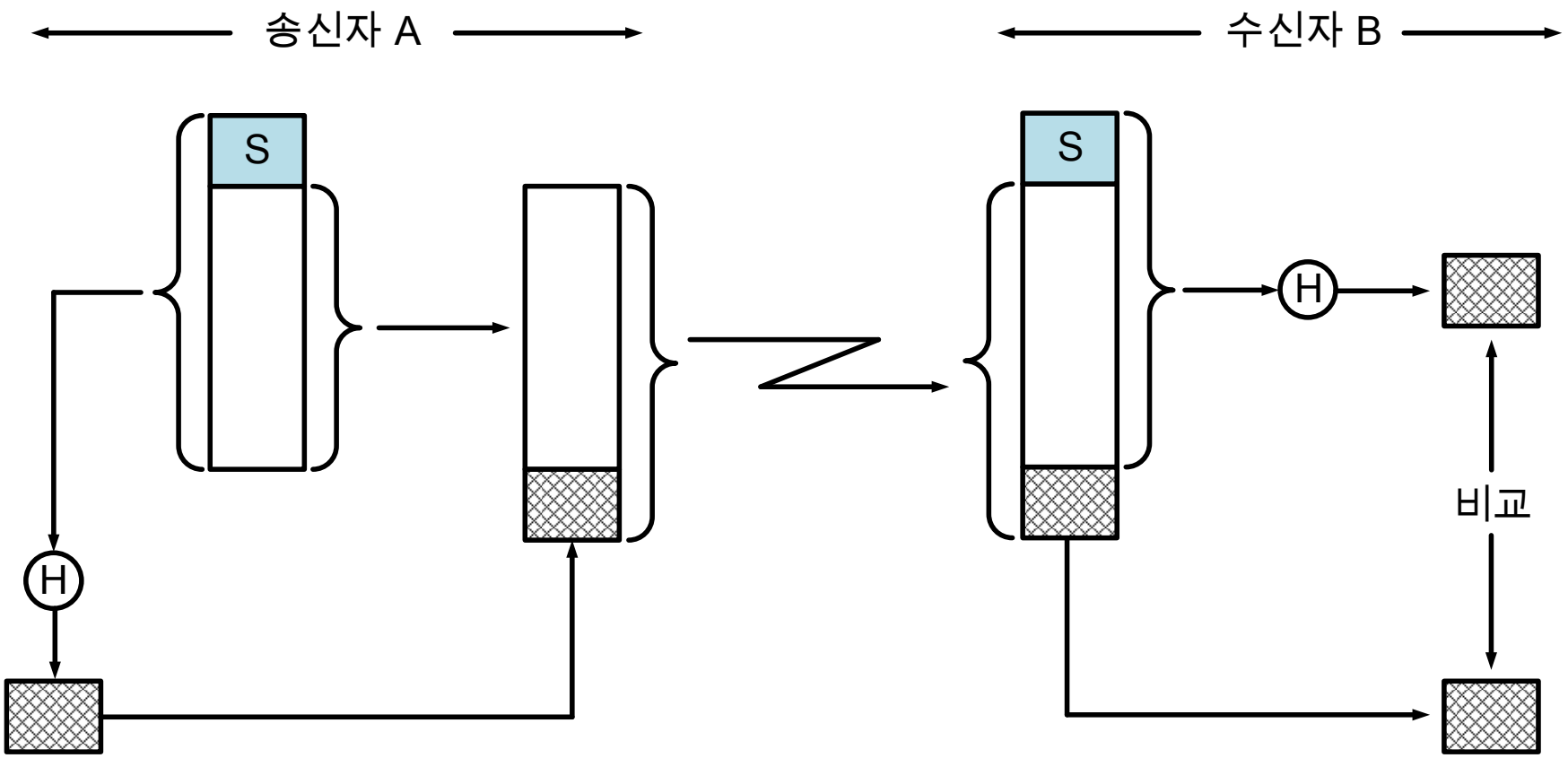
메시지 인증

- 메시지 암호화 없이 인증
  - 일방향 해시 함수
    - 공개키 암호를 사용한 메시지 인증



메시지 인증

- 메시지 암호화 없이 인증
  - 일방향 해시 함수
    - 비밀 값을 사용한 메시지 인증



해시 함수

- 해시 함수(Hash function)
  - 임의 길이의 메시지를 입력해서 고정된 길이의 메시지 다이제스트를 출력하는 함수
  - 특징
    - 메시지 인증에 필요한 파일, 메시지, 데이터 등의 특징을 나타내는 지문(Fingerprint)의 기능을 가지고 있음
    - 메시지 인증, 디지털 서명에 사용됨
      - 디지털 서명 (Digital signature)
        - 공개키 시스템에서 송신자의 신원을 증명하는 방법

해시 함수

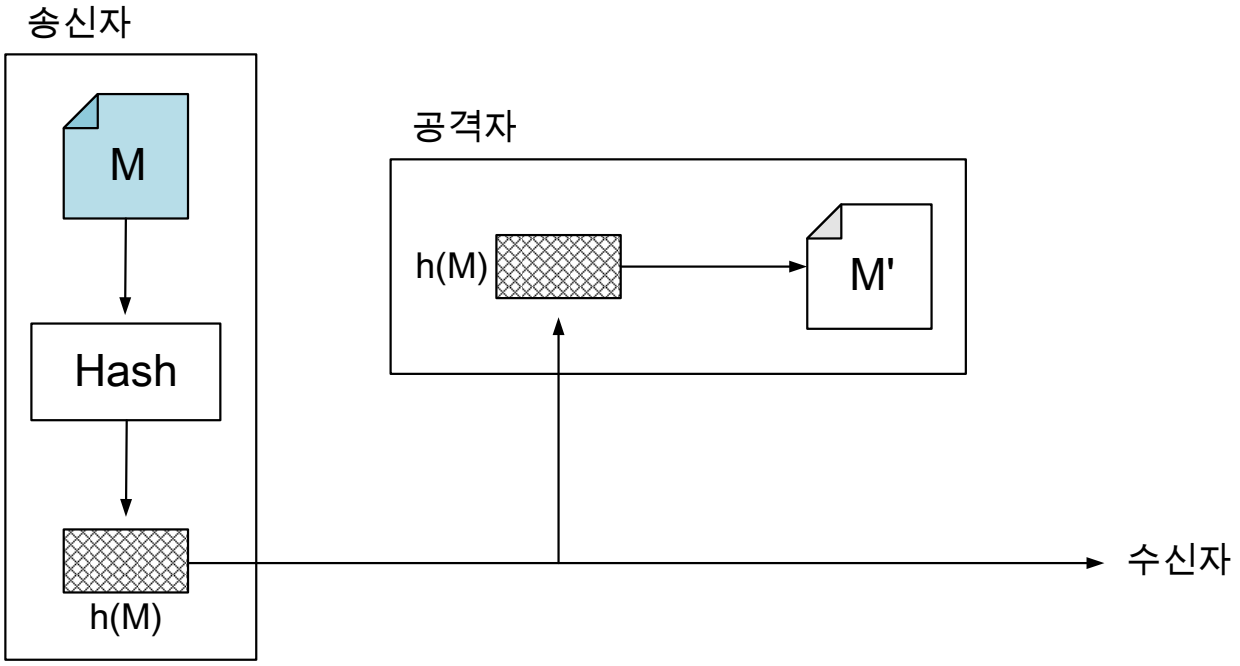
- 안전한 해시 함수의 조건

1. 자유로운 길이의 입력에 대한 처리
2. 일정한 길이의 출력
3. 계산이 쉽고 구현 가능해야 함
4. 일방향 성질
5. 약한 충돌 저항성
6. 강한 충돌 저항성

해시 함수

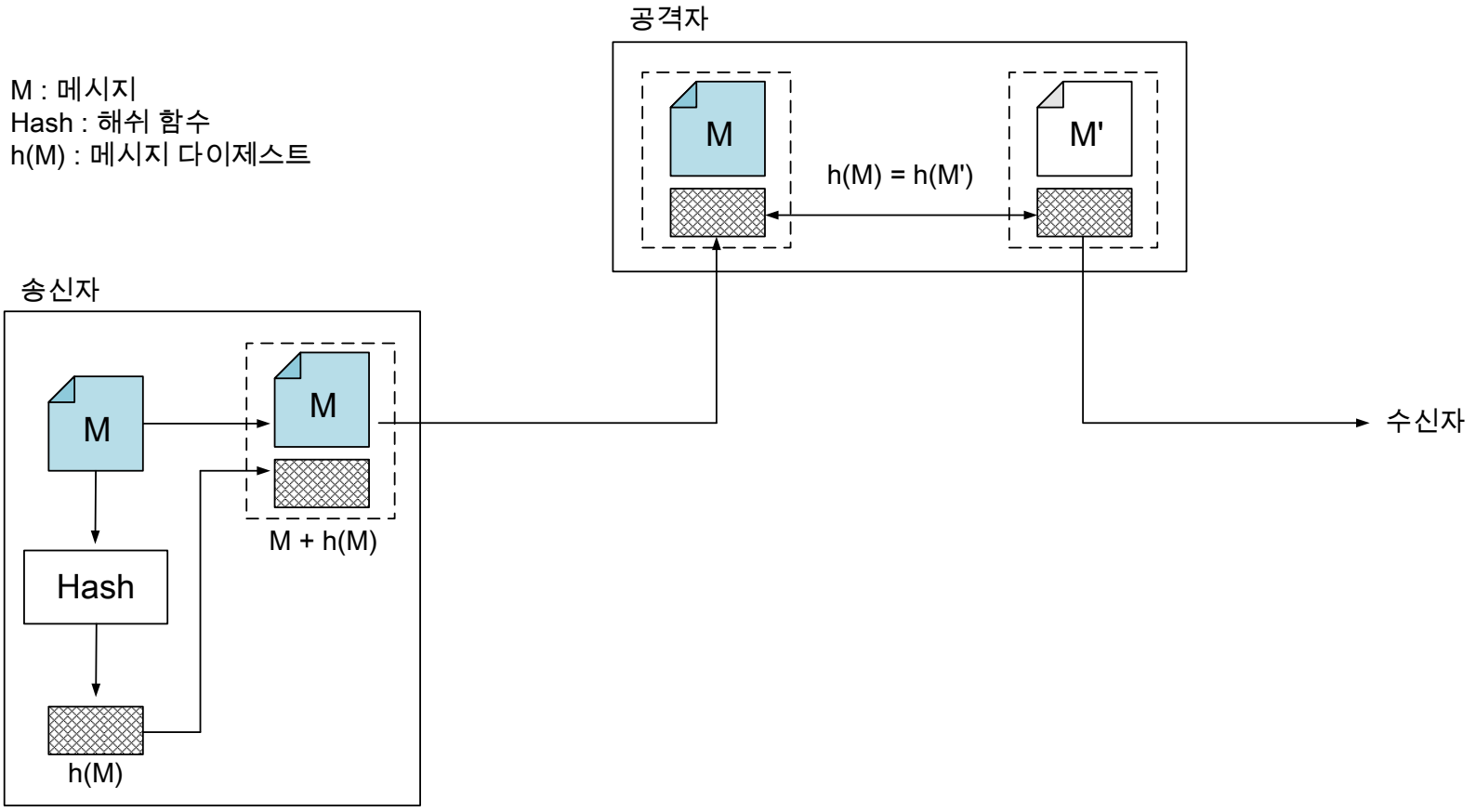
- 일방향 성질(One-way property)
  - 공격자로 하여금 해시 함수에서 출력된 해시 값 h(M)으로부터 다른 메시지 M’을 계산해낼 수 없어야 함

M : 메시지
Hash : 해쉬 함수
h(M) : 메시지 다이제스트



해시 함수

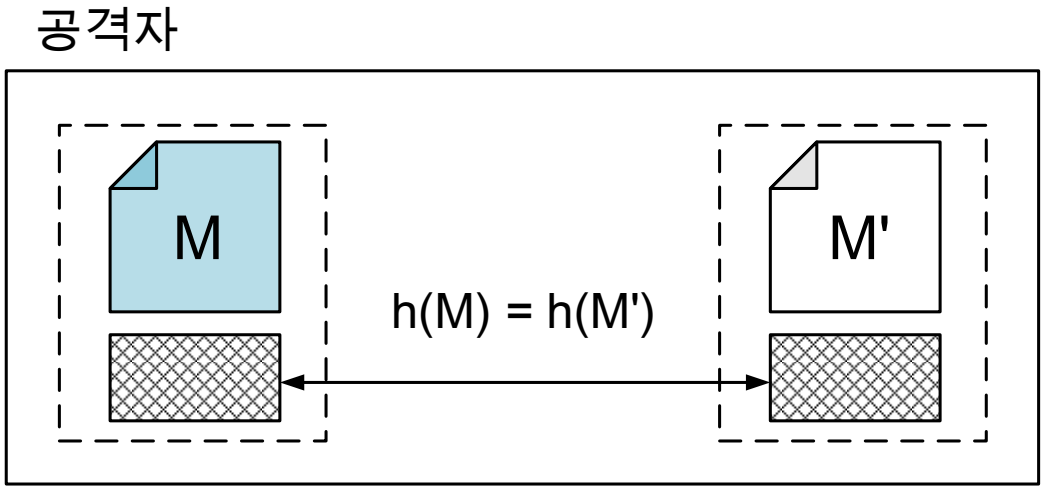
- 약한 충돌 저항성 (Weak collision resistance)
  - 동일한 해시 값 h(M)을 가지는 다른 메시지 M’을 계산해낼 수 없어야 함



해시 함수

- 강한 충돌 저항성 (Strong collision resistance)
  - 동일한 해시 값 h(M)을 가지는 서로 다른 2개의 메시지 쌍 M, M'을 계산해낼 수 없어야 함

M : 메시지
h(M) : 메시지 다이제스트



해시 함수

- 해시 함수의 보안
  - 전수조사 공격에 대한 해시 함수의 강도는 해시 코드의 길이에 달려있음
  - 길이가 n-비트인 해시 코드에 대한 공격 난이도

해시 함수 요건	공격 복잡도
일방향 성질	2^n
약한 충돌 저항성	2^n
강한 충돌 저항성	$2^{n/2}$

해시 함수

- 단순 해시 함수(Simple hash function)
  - 전체 입력 데이터를 연속적인 n-비트 블록으로 간주
    - 세로 덧붙임 검사(Longitudinal redundancy check)
      - 각 비트별로 패리티를 계산하는 방법
      - 임의의 데이터에 대한 무결성 검사에 효과적
      - $C_i = b_{i1} \oplus b_{i2} \oplus \bullet\bullet\bullet \oplus b_{im}$

	비트 1	비트 2	● ● ●	비트 n
블록 1	b_{11}	b_{21}		b_{n1}
블록 2	b_{12}	b_{22}		b_{n2}
●	●	●	●	●
●	●	●	●	●
●	●	●	●	●
블록 m	b_{1m}	B_{2m}		b_{nm}
해시 코드	C_1	C_2	● ● ●	C_n

해시 함수

- 안전 해시 알고리즘(SHA, Secure Hash Algorithm)
  - 1993년 디지털 서명을 위해 NIST가 개발한 알고리즘
  - SHA-0
    - 1993년 NIST에 의해 안전한 해시 표준(Secure Hash Standard, FIPS PUB 180)으로 출판함
  - SHA-1
    - 1995년 개정된 알고리즘(FIP PUB 180-1)을 새로 출판함
  - SHA-2
    - 2002년 해시 값의 길이가 더 긴 네 개의 변형(SHA-224, SHA-256, SHA-384, SHA-512)을 FIP PUB 180-2으로 발표함

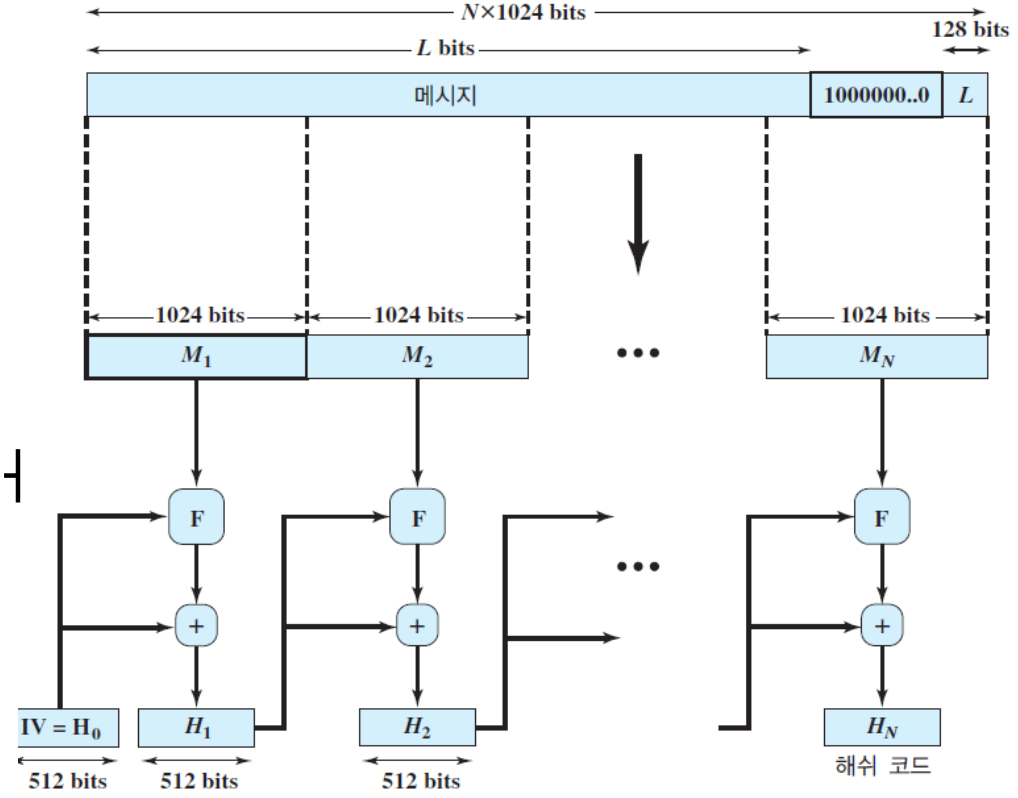
해시 함수

- 안전 해시 알고리즘
  - SHA 매개변수 비교 표
    - 모든 길이의 단위는 비트

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
메시지 다이제스트 길이	160	224	256	384	512
최대 메시지 길이	2^{64}	2^{64}	2^{64}	2^{128}	2^{128}
블록 길이	512	512	512	1024	1024
단어 길이	32	32	32	64	64
라운드 수	80	64	64	80	80
보안	80	112	128	192	256

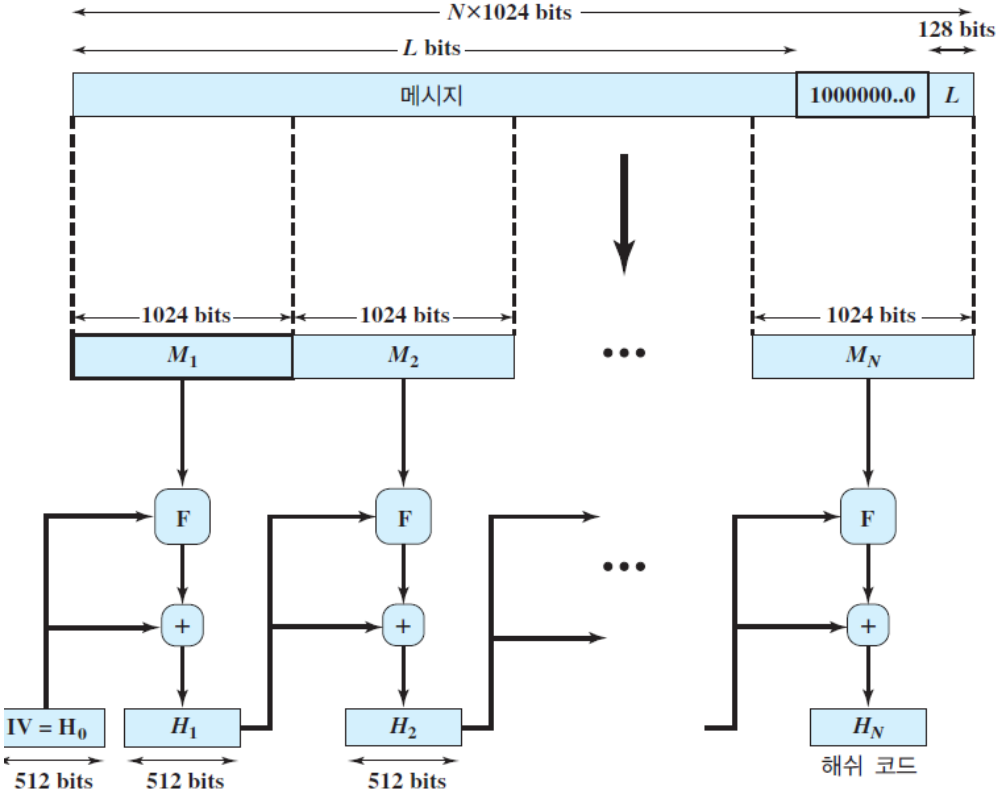
해시 함수

- 안전 해시 알고리즘
  - SHA-512 구조
    - 과정 (1/2)
      - 최대 길이가 2^{128} 비트 이하인 메시지를 입력으로 사용
      - 길이가 512 비트인 메시지 다이제스트를 출력
      - 입력 데이터는 길이가 1024 비트인 블록으로 나누어 처리됨



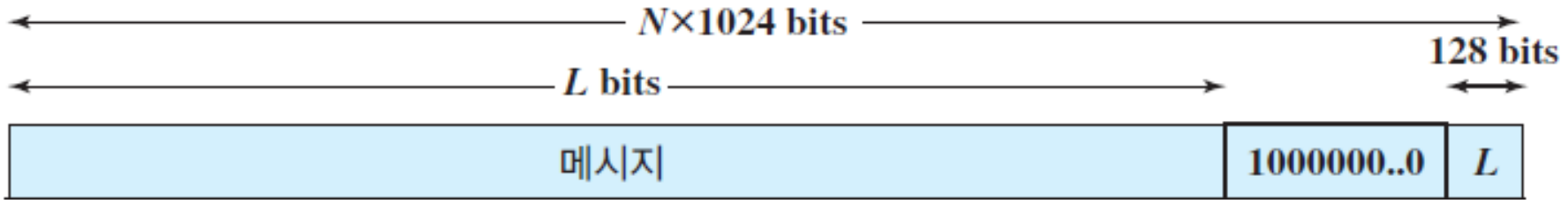
해시 함수

- 안전 해시 알고리즘
  - SHA-512 구조
    - 과정 (2/2)
      - 출력된 메시지 다이제스트는 다음 블록과 섞여 중간 단계 메시지 다이제스트를 생성
      - 마지막 블록과 섞여 처리된 메시지 다이제스트가 전체 메시지에 대한 메시지 다이제스트가 됨



해시 함수

- 안전 해시 알고리즘
  - SHA-512 구조
    1. 패딩 비트 붙이기(Appending padding bits)
       - 메시지에 패딩을 추가하여 총 길이를 896비트(mod 1024)가 되도록 만듦
       - 메시지의 길이가 이미 원하는 길이여도 패딩은 항상 추가함
       - 패딩 시 첫 비트는 1, 나머지 비트는 0으로 채움
    2. 길이 붙이기(Append length)
       - 부호 없는 128 비트 정수로 패딩을 추가하기 전 메시지 길이를 포함



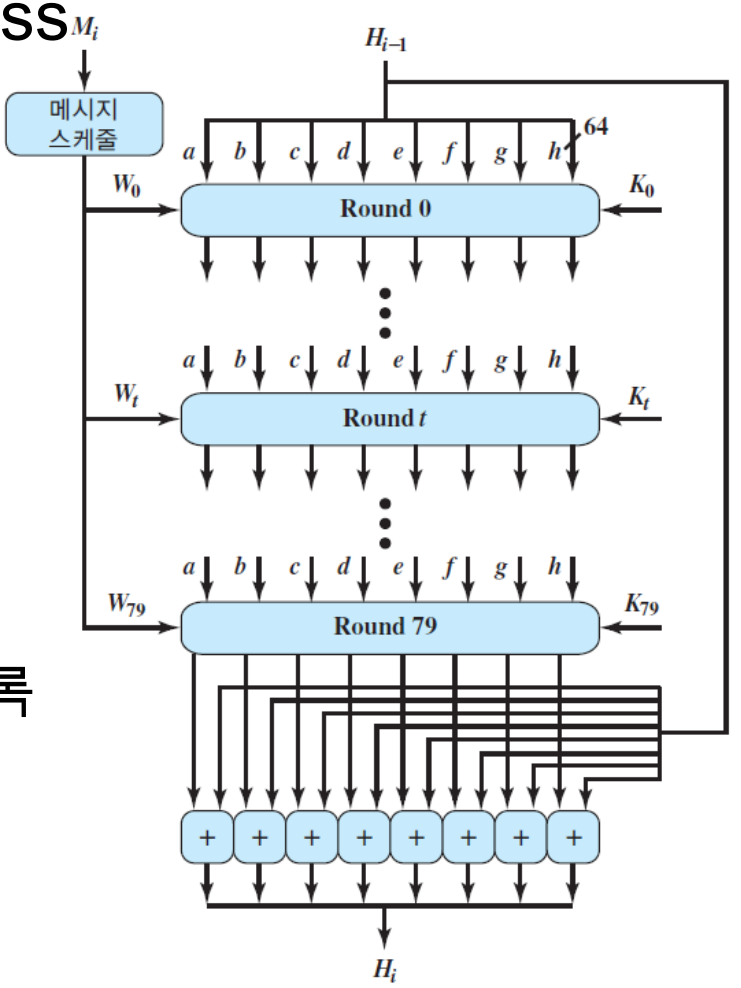
해시 함수

- 안전 해시 알고리즘
  - SHA-512 구조
    3. MD 버퍼 초기화(Initialize MD buffer)
       - 512 비트 버퍼를 해시 함수의 중간 값과 최종 값을 저장하기 위해 사용함
       - 버퍼를 8개의 64 비트 레지스터(a, b, c, d, e, f, g, h)로 나타냄
         - 레지스터의 초기값은 16 진수로 초기화됨

a = 6A09E667F3BCC908	e = 510E527FADE682D1
b = BB67AE8584CAA73B	f = 9B05688CEB3E6C1F
c = 3C6EF372FE94F82B	g = 1F83D9ABFB41BD6B
d = A54FF53A5F1D36F1	h = 5BE0CDI9137E2179

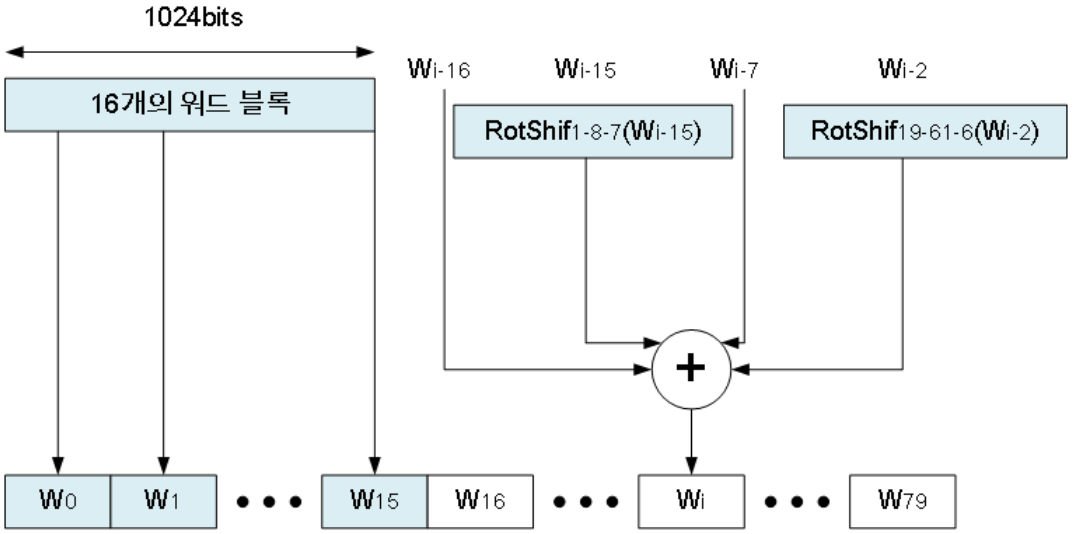
해시 함수

- 안전 해시 알고리즘
  - SHA-512 구조
    4. 1024 비트 블록 메시지 처리(Process message in 1024 bit blocks) (1/3)
       - 각 블록 처리는 80 라운드로 이루어짐
       - 직전 8개의 64 비트 버퍼 값과 64 비트 덧셈 상수, 스케줄된 메시지 64 비트 워드를 라운드 함수에 입력으로 사용하여 버퍼의 내용을 매 라운드마다 갱신
       - 각 블록의 길이가 1024 비트인 다중 블록 메시지를 압축하여 512 비트(8개의 64 비트 워드) 메시지 다이제스트를 생성



해시 함수

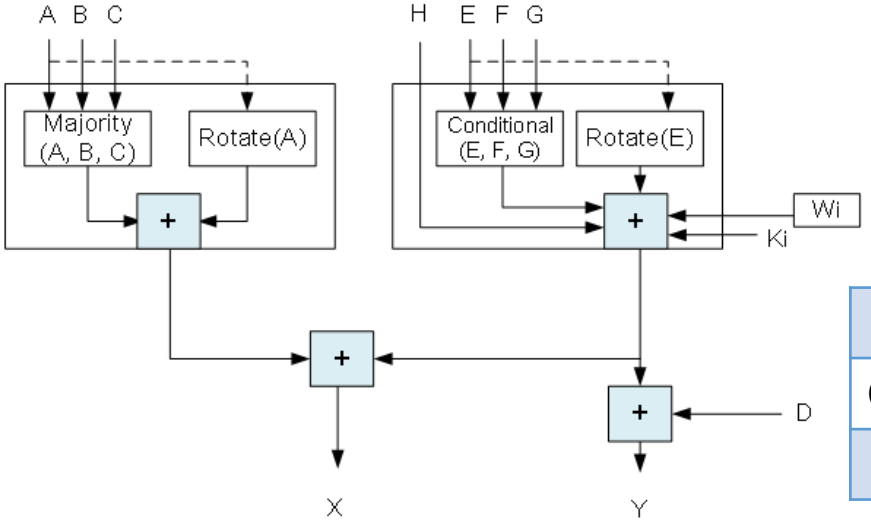
- 안전 해시 알고리즘
  - SHA-512 구조
    4. 1024 비트 블록 메시지 처리 (2/3)
       - 메시지 스케줄 구조
         - 16개의 워드 블록을 80개로 확장

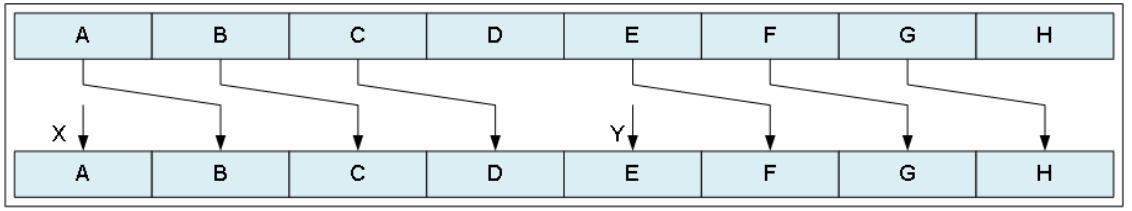


$RotShift_{l-m-n}(x)$	$RotR_l(x) \oplus RotR_m(x) \oplus ShL_n(x)$
$RotR_i(x)$	x를 i만큼 오른쪽 쉬프트
$ShL_i(x)$	x를 i비트 씩 왼쪽 쉬프트 하고 0을 왼쪽으로 채움

해시 함수

- 안전 해시 알고리즘
  - SHA-512 구조
    4. 1024 비트 블록 메시지 처리 (3/3)
       - 라운드 함수 구조
         - 각 라운드에 해당되는 워드 및 덧셈 상수를 입력으로 사용
         - Majority 함수 : 3개의 버퍼(A, B, C)에서 대응되는 3개의 비트를 계산
         - Conditional 함수 : 3개의 버퍼(E, F, G)에서 대응되는 3개의 비트를 계산
         - Rotate 함수 : 동일한 버퍼(A 또는 E)의 값을 오른쪽으로 순환

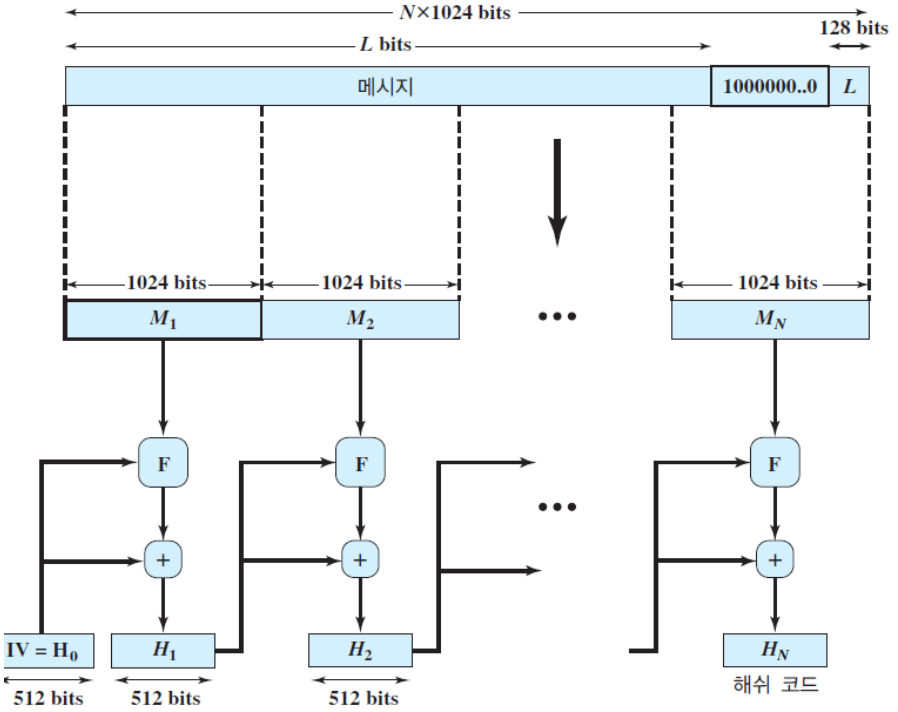




Majority (x, y, z)	(x AND y) ⊕ (y AND z) ⊕ (z AND x)
Conditional (x, y, z)	(x AND y) ⊕ (NOT x AND z)
Rotate (x)	$RotR_{28}$ (x) ⊕ $RotR_{34}$ (x) ⊕ $RotR_{39}$ (x)

해시 함수

- 안전 해시 알고리즘
  - SHA-512 구조
    5. 출력(Output)
       - n개의 1024 비트 블록 모두가 처리된 뒤에 n번째 단계에서 512 비트 메시지 다이제스트를 출력



메시지 인증 코드

- 메시지 인증 코드(Message Authentication Code)
  - HMAC(Hashed MAC)
    - 해싱 기법을 적용하여 메시지의 위·변조를 방지하는 기법
    - 응용
      - IP 보안, 전송 계층 보안(TLS, Transport Layer Security), 안전한 전자 결제(SET, Secure Electronic Transaction)와 같은 인터넷 프로토콜에서 사용됨
    - 설계 목표
      - 수정하지 않고 쓸 수 있는 해시 함수를 만듦
      - 더 빠르고 안전한 해시 함수가 있다면 기존 해시 함수를 교체할 수 있도록 함
      - 기능저하 없이 해시 함수의 원래 기능을 유지하도록 함
      - 키를 보다 쉽게 다루고자 함
      - 내장된 해시 함수에서 인증 메커니즘의 강도에 대해 암호 해독을 확실히 파악할 수 있도록 함

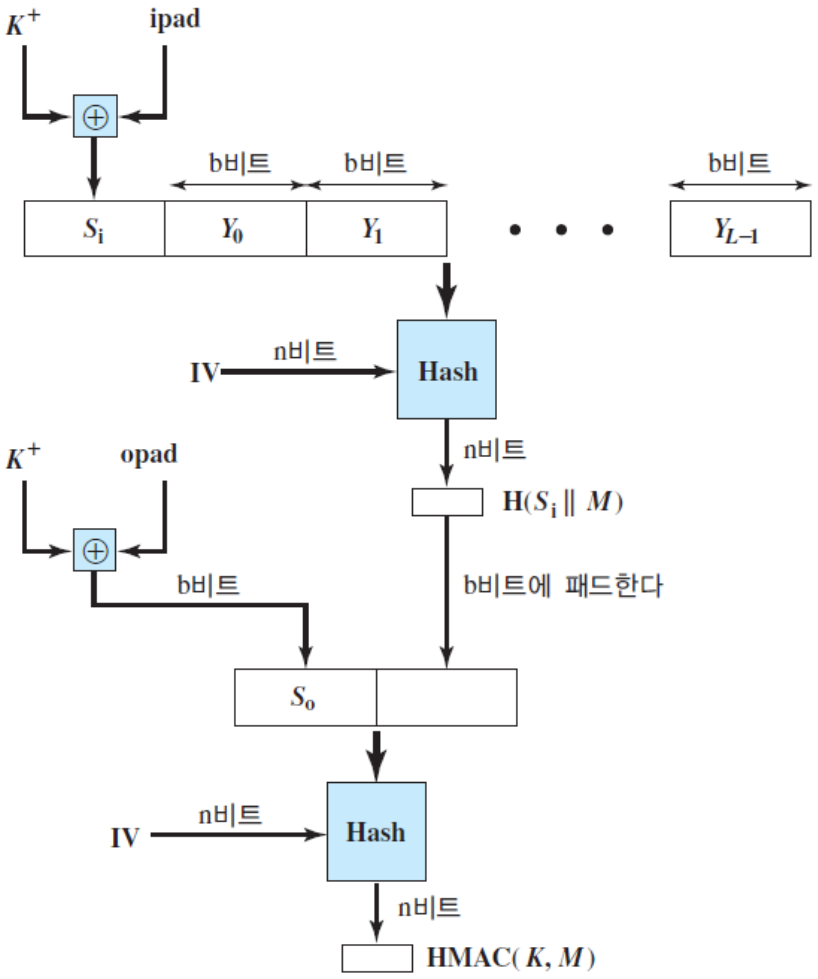
메시지 인증 코드

- HMAC
  - 용어

용어	정의
H	해시 함수
M	HMAC의 입력 메시지
Y_i	M의 i번째 블록
L	M의 블록 수
b	블록의 비트 수
n	내장된 해시 함수에 의해 생성된 해시 코드의 길이
K	비밀키, 키의 길이가 b보다 길면 n 비트 키를 생성하는 해시 함수에 입력으로 사용
K^+	K의 왼쪽에 0을 붙여서 길이가 b 비트가 되도록 한 것
ipad	00110110(16 진수 36)을 b/8번 반복한 2진 수열
opad	01011100(16 진수 5C)을 b/8번 반복한 2진 수열

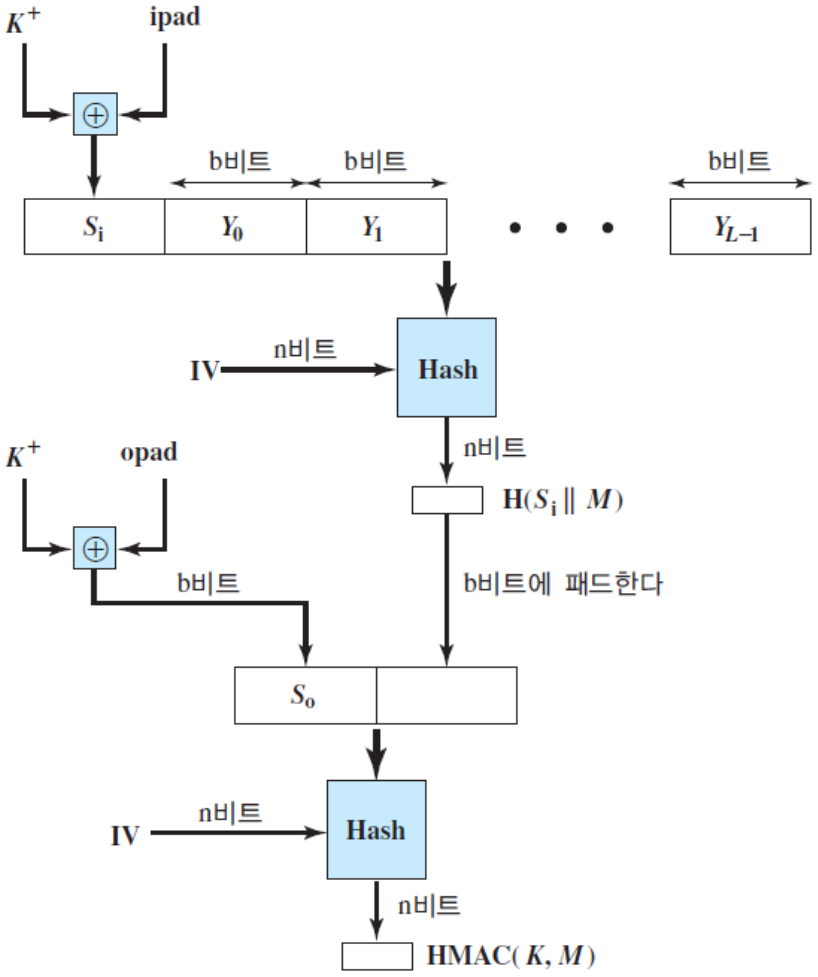
메시지 인증 코드

- HMAC
  - 구조 (1/2)
    1. 메시지를 길이가 b 비트인 블록으로 분리
    2. 비밀 키 K의 왼쪽에 0으로 된 열을 패딩하여 b 비트의 K^+ 생성
    3. K^+와 상수 ipad(input pad)를 XOR 연산하여 b 비트 S_i 블록 생성
    4. S_i를 메시지 맨 앞에 붙이고 n 비트 IV와 해시 함수에 입력
    5. 출력 값으로 생성된 n 비트 다이제스트를 중간 HMAC이라고 부름



메시지 인증 코드

- HMAC
  - 구조 (2/2)
    6. n 비트로 된 중간 HMAC의 왼쪽에 0으로 이루어진 열을 패딩하여 b 비트 블록 생성
    7. 단계 2와 단계 3을 다른 상수인 opad(output pad)로 반복하여 b 비트 S_o 블록 생성
    8. 단계 7에서 얻은 결과를 단계 6에서 얻은 블록 앞에 붙임
    9. 단계 8의 결과에 동일한 해시 함수를 적용하여 최종 n 비트 HMAC을 생성



메시지 인증 코드

- 블록 암호 기반 MAC
  - 암호 기반 메시지 인증 코드(CMAC, Cipher-based Message Authentication Code)
    - 대칭키 암호를 N번 사용하여 N개의 평문 블록으로부터 하나의 MAC을 생성
    - 운용 모드는 AES와 3DES를 사용
      - AES
        - 암호 블록 길이 b : 128 비트
        - 키 길이 k : 128, 192 또는 256 비트
      - 3DES
        - 암호 블록 길이 b : 64 비트
        - 키 길이 k : 112 또는 168 비트
    - 메시지는 n개 블록으로 나뉨
    - n 비트 K_1을 사용

메시지 인증 코드

- 블록 암호 기반 MAC
  - 암호 기반 메시지 인증 코드
    - 계산식과 용어

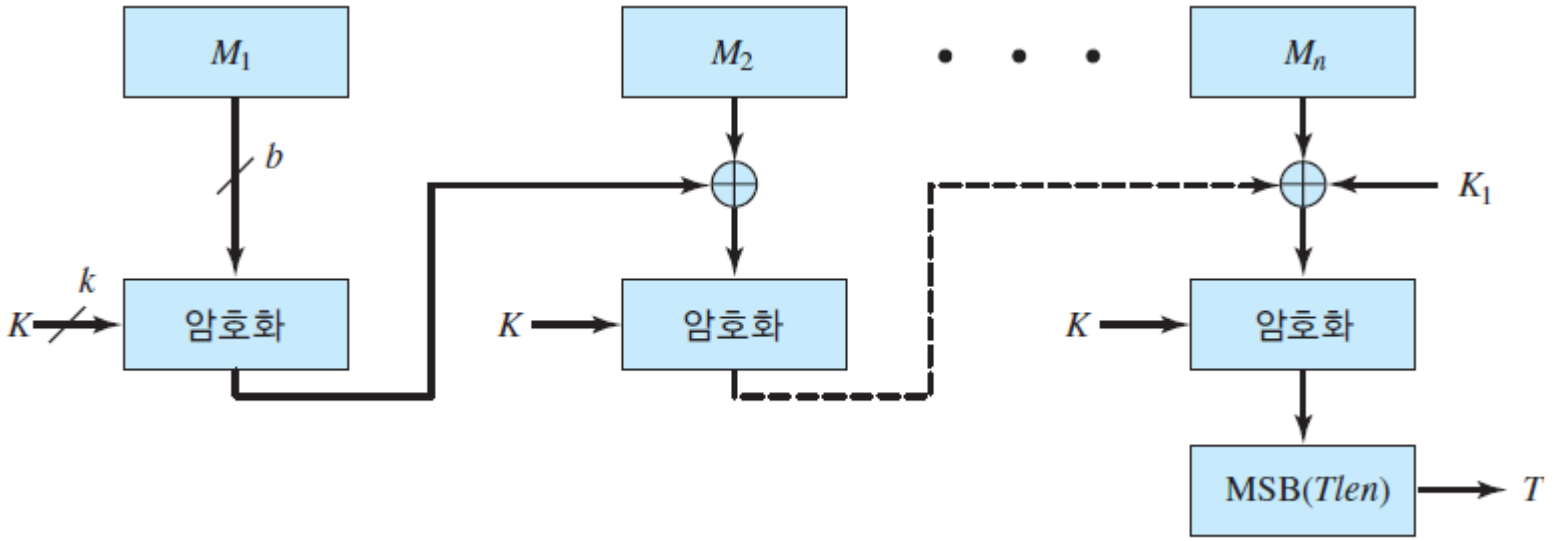
계산식
$C_1 = E(K, M_1)$
$C_2 = E(K, [M_2 \oplus C_1])$
$C_3 = E(K, [M_3 \oplus C_2])$
• • •
$C_n = E(K, [M_n \oplus C_{n-1} \oplus K_1])$
$T = MSB_{Tlen}(C_n)$

용어	정의
T	메시지 인증 코드, “태그(Tag)”
Tlen	T의 비트 길이
MSB_s (X)	비트열 X의 왼쪽부터 S개 비트
K_1	결과로 나온 암호문을 한 비트 왼쪽으로 이동시킨 값
K_2	K_1을 한 비트 왼쪽으로 이동시킨 값

- MSB(Most Significant Bit) : 최상위 비트

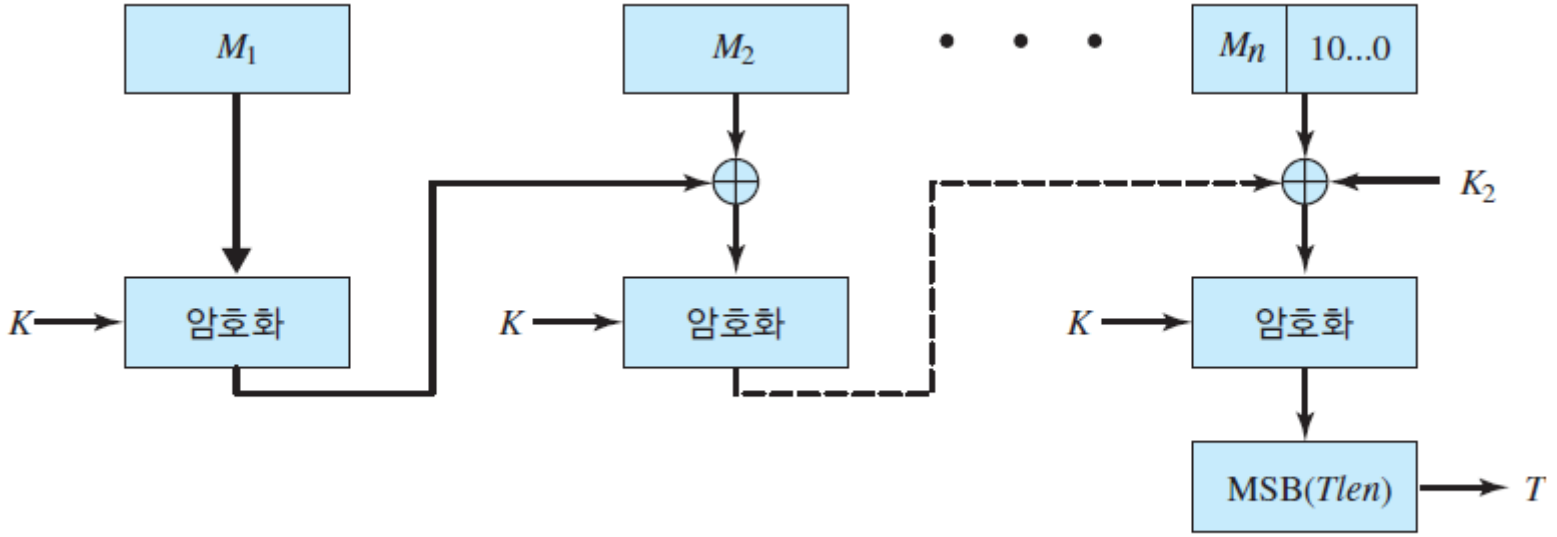
메시지 인증 코드

- 블록 암호 기반 MAC
  - 암호 기반 메시지 인증 코드
    - 메시지 길이가 블록 길이의 정수배일 때
      - k 비트 키와 b 비트 서브키 K_1을 사용



메시지 인증 코드

- 블록 암호 기반 MAC
  - 암호 기반 메시지 인증 코드
    - 메시지 길이가 블록 길이의 정수배가 아닐 때
      - 마지막 블록에 패딩을 붙여 블록의 길이가 b 비트가 되게 함
      - k 비트 키와 b 비트 서브키 K_2를 사용

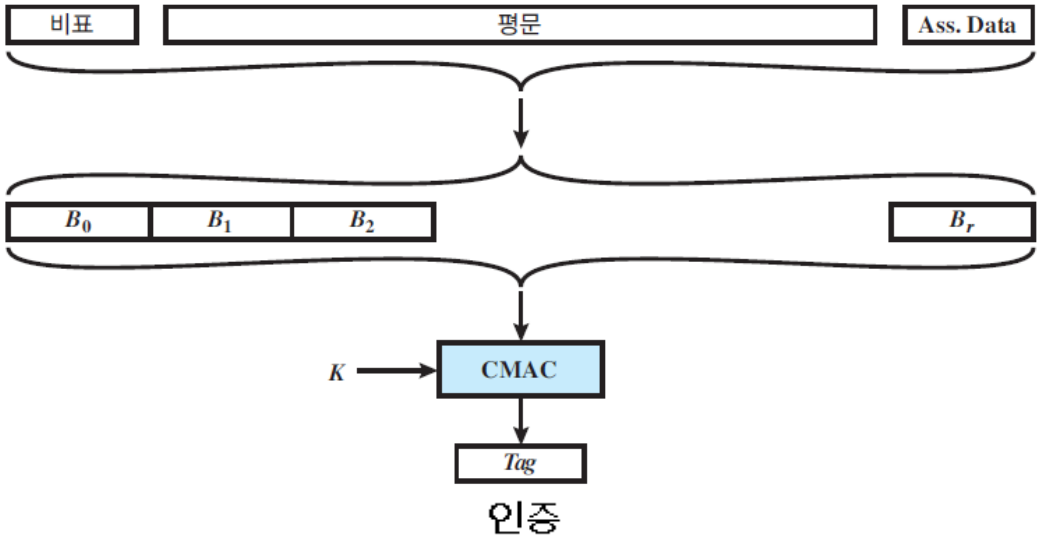


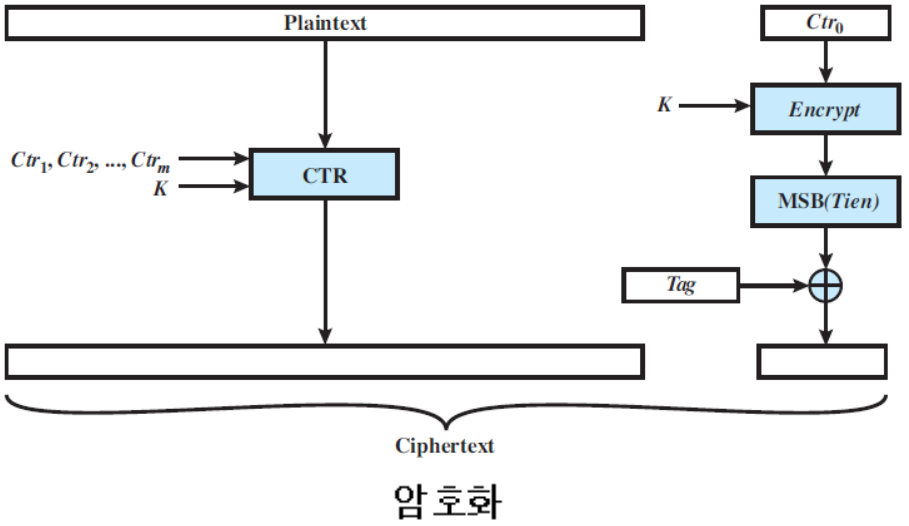
메시지 인증 코드

- 블록 암호 기반 MAC
  - 암호 블록 체인 카운터 MAC(CCM, Counter with Cipher block chaining-Message authentication code)
    - 인증된 암호화(Authentication encryption)모드라고도 함
      - 통신상 기밀성과 인증(무결성)을 동시에 보호하는 암호 시스템을 설명할 때 사용되는 용어
    - 핵심 알고리즘 요소
      - AES 암호 알고리즘
      - CTR 운용 모드
      - CMAC 인증 알고리즘
    - 암호화와 인증에 동일한 키를 사용
    - 암호화 과정에 입력되는 내용
      - 인증하고 암호화할 데이터 (평문 메시지 데이터 블록 P)
      - 인증은 하지만 암호화는 하지 않는 데이터 (유관 데이터 A)
      - 프로토콜 연관이 있는 동안 모든 순간에 달라지는 유일 값 (유관 데이터에 할당되는 비표 N)

메시지 인증 코드

- 블록 암호 기반 MAC
  - 암호 블록 체인 카운터
    - 구조





감사합니다!