SEOULTECH

암호이론 및 정보보안 실무

Chapter 11. Cryptographic Hash Functions

Seoul National University of Science and Technology

Ubiquitous Computing And Security Laboratory

M.S. Student. Tae Woo Kim

SeoulTech UCS Lab

Ubiquitous Computing & Security Laboratory

INDEX

11.0 해쉬 함수

11.0 해쉬 함수

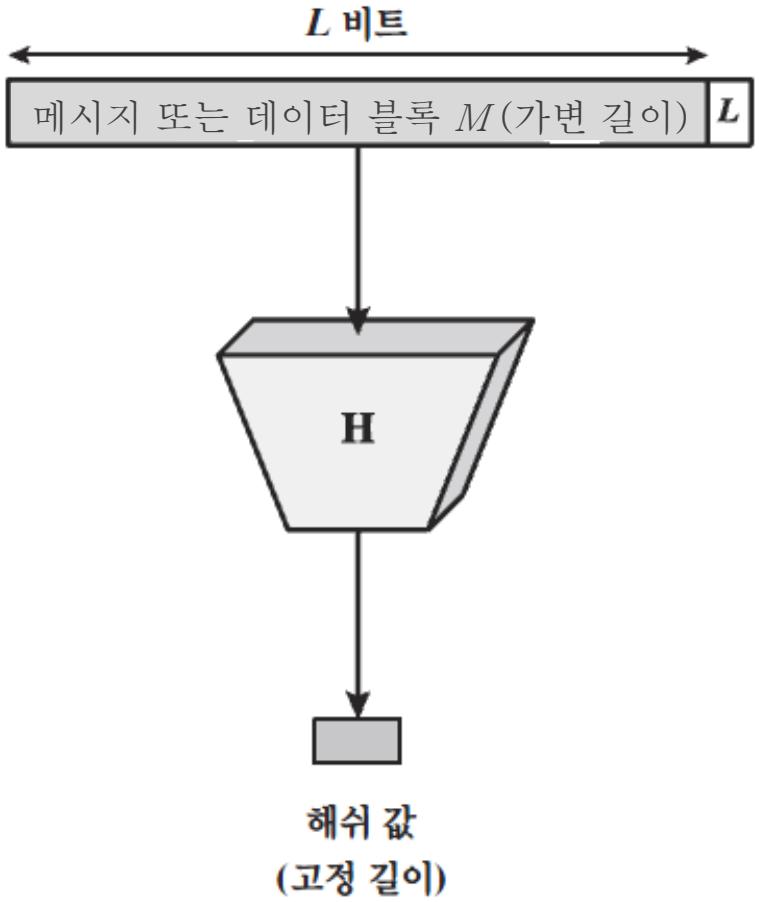
해쉬함수 H

- 입력: 가변길이 블록 데이터 M
- 출력: 고정길이의 해쉬값 h=H(M)

암호학적 해쉬 함수

- 이미 정의된 해쉬값에 대응하는 데이터객체를 찾는 것이 계산적으로 불가능해야 함 (일방향성)
- 동일한 해쉬 값을 갖는 두 개의 데이터 객테를 찾는 것이 계산적으로 불가능해야 함 (충돌회피성)
- 보안 응용에 사용됨

11.0 해쉬 함수



11.1 암호학적 해쉬 함수의 응용

11.1 암호학적 해쉬 함수의 응용

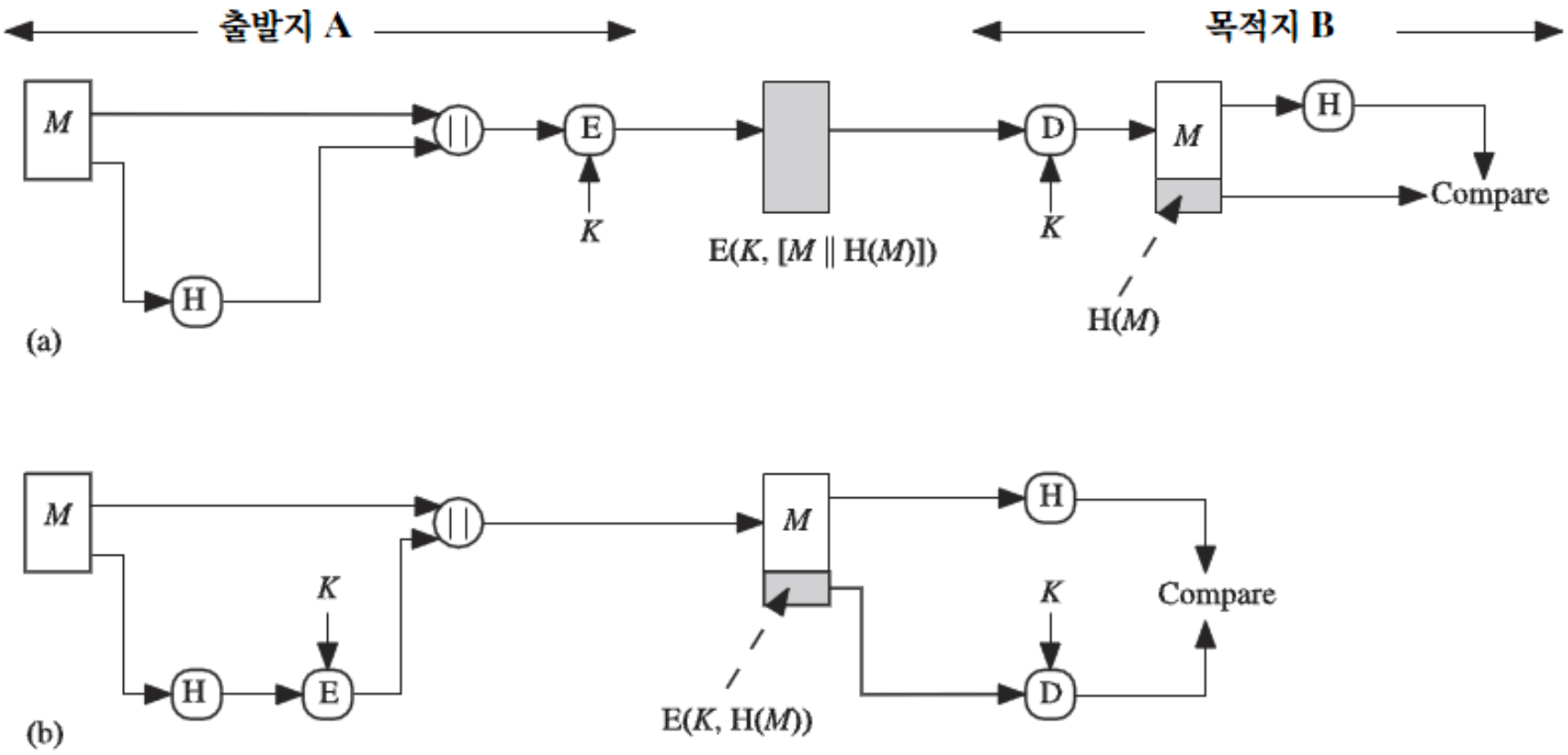
메시지 인증

- 메시지의 무결성을 검증하는데 사용되는 기법이나 서비스
- 메시지 다이제스트 (message digest)
  - 해쉬함수가 메시지 인증을 제공하는데 사용될 때의, 해쉬함수값을 메시지 다이제스트라고 함

- 메시지 인증 방법들
  a) 메시지와 해쉬코드가 대칭키 암호에 의하여 암호화
  b) 해쉬코드만 대칭키 암호를 이용하여 암호화
  c) 메시지와 공통비밀키S로부터 해쉬값 계산, 수신자도 S를 알기 때문에 검증을 위한 해쉬값 계산할 수 있음
  d) c)에 기밀성 추가 (전체 메시지와 해쉬코드를 암호화)

11.1 암호학적 해쉬 함수의 응용

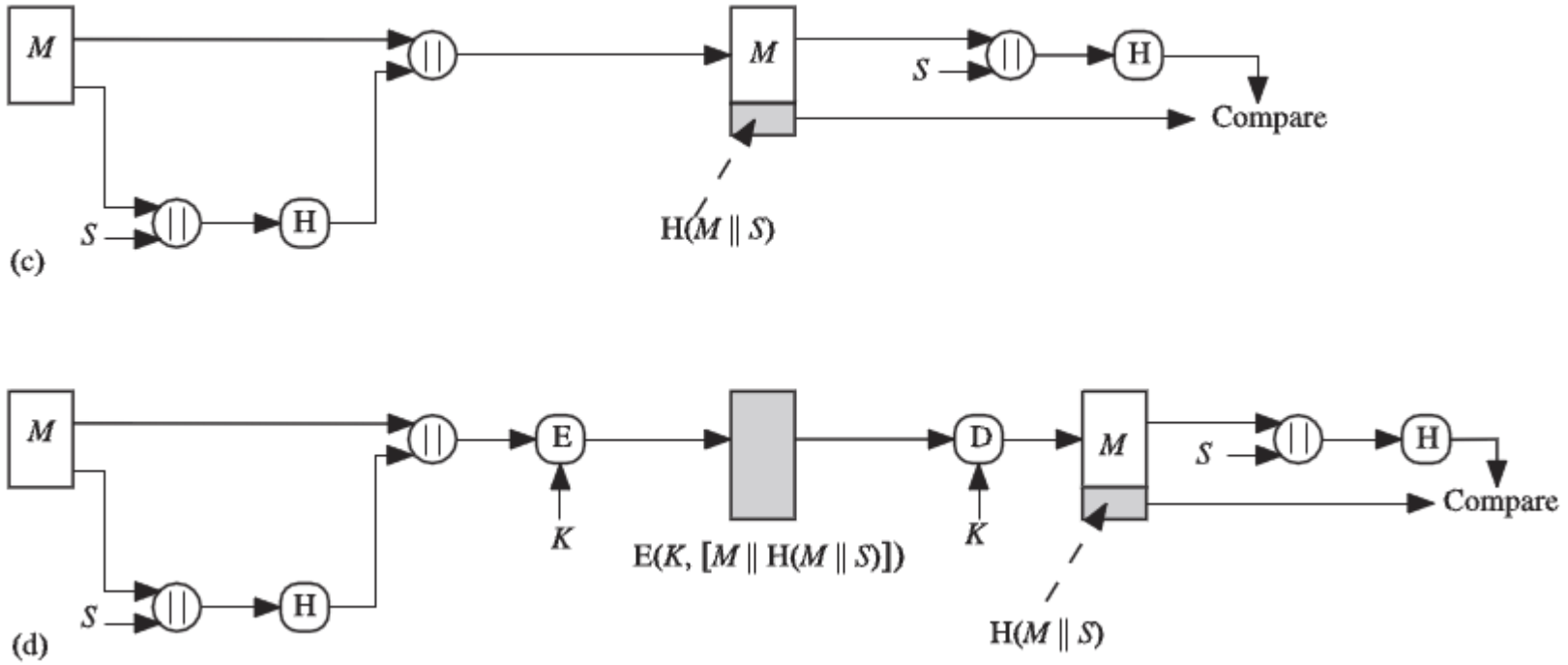
메시지 인증 방법들



a) 메시지와 해쉬코드가 대칭키 암호에 의하여 암호화
b) 해쉬코드만 대칭키 암호를 이용하여 암호화

11.1 암호학적 해쉬 함수의 응용

메시지 인증 방법들



c) 메시지와 공통 비밀키 S로부터 해쉬값 계산, 수신자도 S를 알기 때문에 검증을 위한 해쉬값을 계산할 수 있음

c) 에 기밀성 추가 (전체 메시지와 해쉬코드를 암호화)

11.1 암호학적 해쉬 함수의 응용

전자 서명

- 어떤 데이터가 정말 그 사람 것이 맞는지를 보장해주는 것
- 메시지의 해쉬 값을 사용자의 비밀키로 암호화 됨
- 사용자의 공개키를 아는 자는 누구나 메시지의 무결성을 검증 할 수 있음

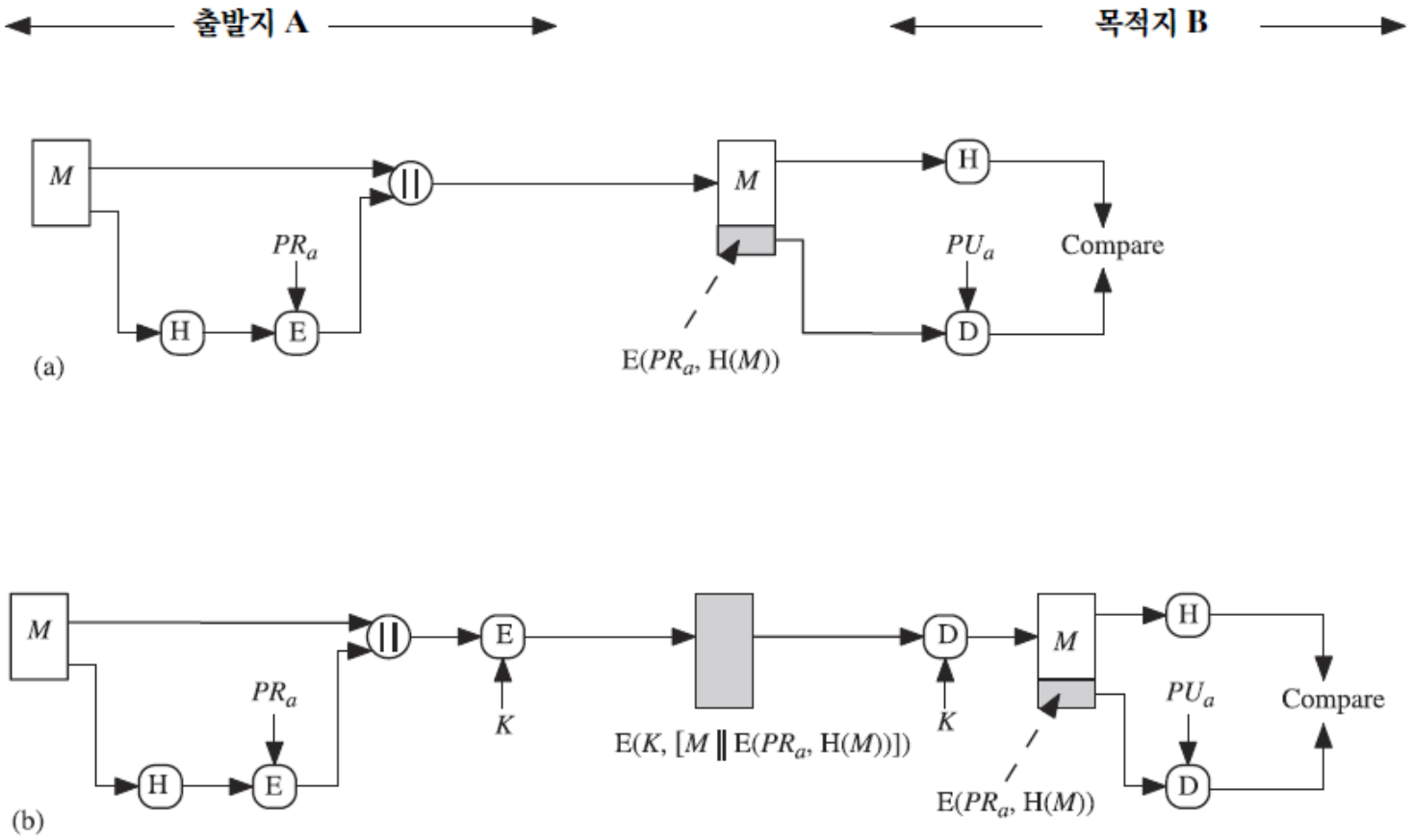
- 전자서명으로 이용하는 방법들

  (a) 해쉬 코드가 송신자의 비밀키를 이용하여 공개 키 암호화가 됨

  (b) '메시지'와 '비밀키로 암호화된 해쉬 코드'가 대칭키로 암호화됨

11.1 암호학적 해쉬 함수의 응용

전자서명 방법들



a) 메시지와 해쉬코드가 대칭키 암호에 의하여 암호화
b) 해쉬코드만 대칭키 암호를 이용하여 암호화

11.2 두 개의 간단한 해쉬 함수

11.2 두 개의 간단한 해쉬 함수

모든 블록을 비트 단위로 exclusive-OR (XOR)

$C_i = b_{i1} \oplus b_{i2} \oplus \cdots \oplus b_{im}$

C_i = 해쉬 코드의 i번째 비트, $1 \le i \le n$

m = 입력에서 n-비트 블록의 수

b_{ij} = j번째 블록의 i번째 비트

$\oplus$ = XOR 동작

- 세로 중복 검사 (longitudinal redundancy check)
- 데이터 무결성 검사 기능

각 블록들이 처리된 후에 해쉬 값에 대해서 1비트 순환 이동(one-bit circular shift) 또는 교대(rotation)

- 좀더 나은 데이터 무결성 검사 기능 지원
- 보안에는 취약

11.3 요구사항과 보안

11.3 요구사항과 보안

암호학적 해쉬 함수의 요구사항

1) 선이미지 회피성 => 약 일방향성

- 주어진 임의 출력값 y에 대하여 y=h(x)를 만족하는 x를 발견하는 것이 계산적으로 불가능해야 한다.

2) 2차 선이미지 회피성 ==> 약한 충돌 내성

- 주어진 입력값 x에 대하여 h(x) = h(y)인 x≠y 발견하는 것이 계산적으로 불가능해야한다.

3) 충돌 저항성 => 강한 충돌 내성

- h(x) = h(x')를 만족하는 x, x'를 발견하는 것이 계산적으로 불가능해야한다.

11.3 요구사항과 보안

암호학적 해쉬 함수의 요구사항

요구사항	설명
다양한 입력 길이	H는 어떤 크기의 데이터 블록에도 적용될 수 있다.
고정된 출력 길이	H는 고정된 크기의 출력을 만든다.
효율성	$H(x)$는 실질적으로 하드웨어 및 소프트웨어에 적용하기 용이하여야 하며, 어떠한 x에 대해서도 계산이 비교적 수월해야 한다.
선이미지 회피성 (일 방향성)	어떠한 코드 h에 대해서도, $H(x) = h$인 x를 찾는 것은 계산적으로 실행 불가능하다.
2차 선이미지 회피성 (약한 충돌 회피성)	어떠한 블록 x에 대해서도, $H(y) = H(x)$인 $y \neq x$인 것을 찾는 것이 계산적으로 실행 불가능하다.
충돌 회피성 (강한 충돌 회피성)	$H(x) = H(y)$인 어떤 (x, y)짝을 찾는 것이 계산적으로 실행 불가능하다.
의사 난수성	H의 출력이 의사난수에 대한 표준 시험을 만족한다.

11.3 요구사항과 보안

해쉬 함수에 대한 공격 (1) : 전사적 공격

- 선 이미지(preimage)와 2차 선 이미지 공격
  - 공격자는 주어진 해쉬 값 h와 같은 값을 갖는 H(y)에 대응하는 y를 찾고자 한다.
  - 길이가 m인 코드에 대한 공격 수준 (2^m 에 비례)
- 충돌 회피 공격(collision resistant attack)
  - 공격자는 같은 해쉬 값 H(x)=H(y)를 만족하는 2개의 메시지/데이터 블록인 x와 y를 찾고자 한다
  - 공격 복잡도는 생일 역설(birthday paradox)로 알려진 수학적인 분석으로 설명됨
  - 길이가 m인 코드에 대한 공격 수준 ($2^{m/2}$ 에 비례)
- 해쉬코드 길이에 따른 안정성
  - 128비트: 안전이 취약함
  - 160비트: 사용가능하지만 탐색기술이 발전하는 중인 만큼 안전한 것만은 아님.

11.3 요구사항과 보안

해쉬 함수에 대한 공격 (2) : 암호해독

- 암호해독 (Cryptanalysis)
  - 암호해독 방법은 해쉬함수 알고리즘의 취약성을 분석해 내는 것
  - 이상적인 해쉬 알고리즘에 대한 공격의 경우, 암호 분석적 노력이 전사적 공격에서 요구되는 노력보다 크거나 같기를 요구

11.4 암호블록연결(CBC)에 기반한 해쉬 함수

11.4 암호블록연결(CBC)에 기반한 해쉬 함수

블록 암호 알고리즘을 해쉬함수로 이용

- 메시지 M을 고정된 크기의 블록 $M_1, M_2, \cdots, M_N$으로 나누고, 다음과 같이 해쉬 코드 G를 계산하기 위해서 대칭 암호 시스템을 이용
  - H_0 = 초기 값
  - $H_i = E(M_i, H_{i-1})$
  - $G = H_N$

- 대칭암호시스템으로 DES가 사용되면 해쉬코드 길이가 64비트이므로 취약함.

- 여러 변종 방법들이 제안되고 있으나, 여전히 다음 공격에 의해 취약함
  - 생일역설공격
  - 중간자 공격 (man-in-the-middle attack)

11.5 SHA (Secure Hash Algorithm)

11.5 SHA (Secure Hash Algorithm)

- 1993년, 미국 국립표준기술연구소 (NIST)에서 연방정보처리표준규격 (FIPS PUB 180)으로 공포 (후에 SHA-0으로 알려짐)

- 1995년, FIPS 180-1 으로 개정 (후에 SHA-1으로 알려짐)
  - 160 비트 해쉬 생성

- 2002년, FIPS 180-2 개정 (SHA-2 명칭 공포)
  - SHA-2 (SHA-256, SHA-384, SHA-512) 공포
  - SHA-1과 동일한 기반 구조, 모듈라 연산, 논리 이진연산 사용
  - 암호분석방법은 SHA-2가 SHA-1과 동일하나, 좀더 높은 안전성을 갖고 있음

11.5 SHA (Secure Hash Algorithm)

SHA 인자 크기 비교

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
메시지 다이제스트 크기	160	224	256	384	512
메시지 크기	$<2^{64}$	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
블록 크기	512	512	512	1024	1024
워드 크기	32	32	32	64	64
단계의 수	80	64	80	80	80

주의: 모든 크기의 단위는 비트이다.

11.5 SHA (Secure Hash Algorithm)

SHA-512 소개

입력: 최대길이 2^{128} 비트 미만인 메시지

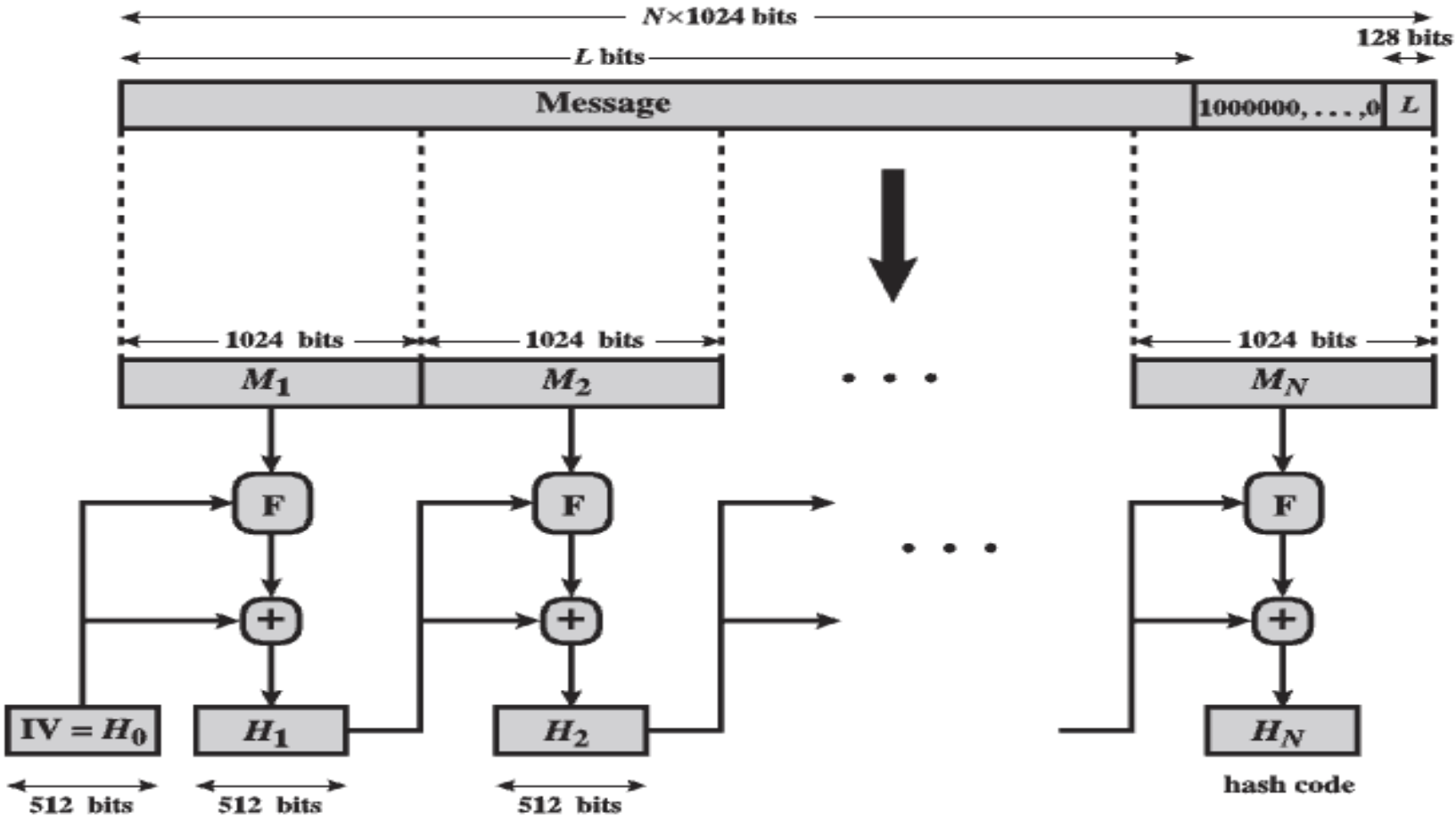
출력: 512 비트 메시지 다이제스트

처리 단계

- 패딩 비트 붙이기
- 길이 붙이기
- MD 버퍼 초기화
- 1024-비트(128-워드)블록 메시지 처리
- 출력

11.5 SHA (Secure Hash Algorithm)

SHA-512 논리



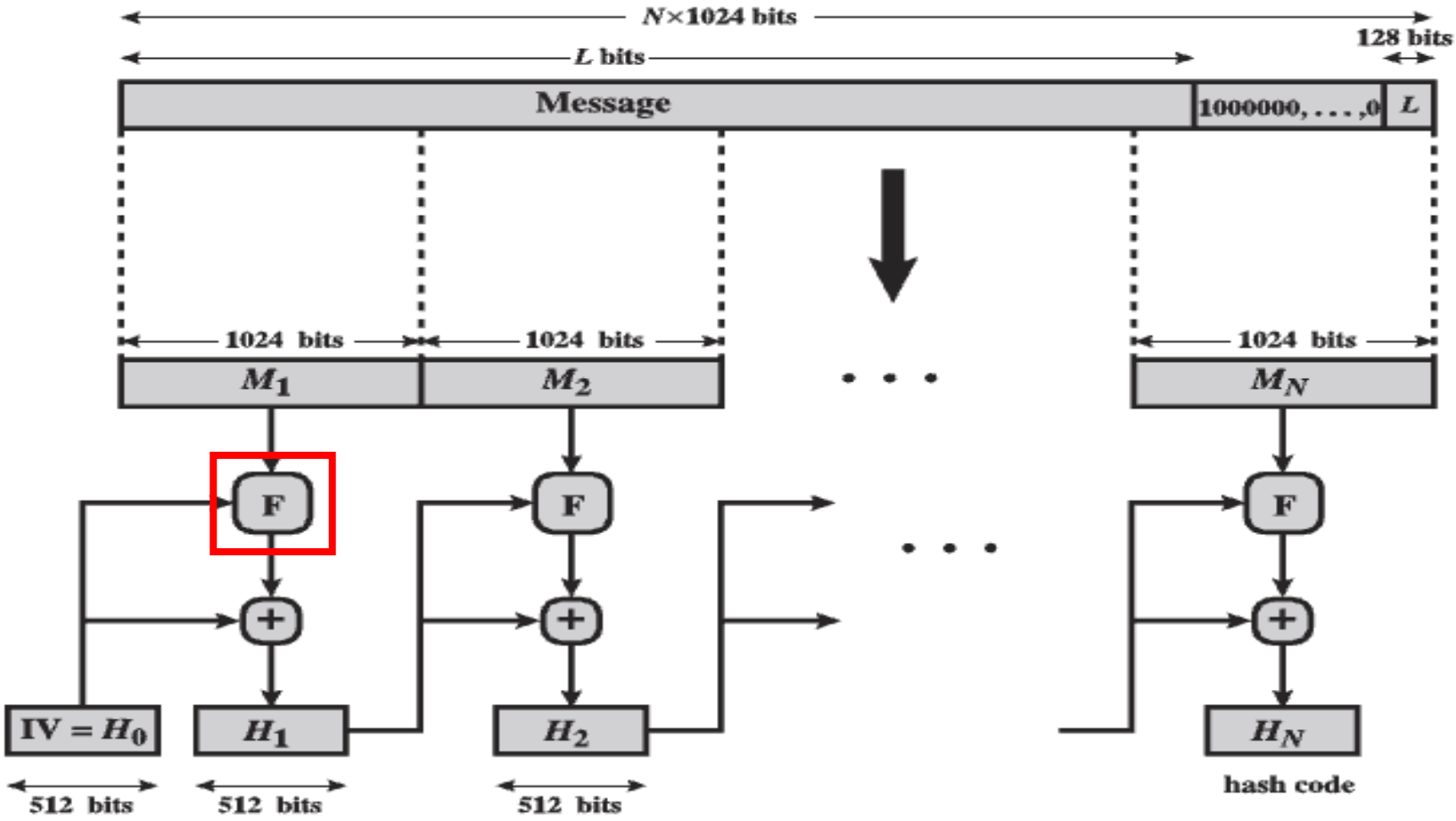
11.5 SHA (Secure Hash Algorithm)

SHA-512 압축 함수

- SHA-512의 중심 함수
- 1024-비트 블록의 메시지를 다룸
- 80개의 라운드로 구성됨
  - 512 비트 버퍼를 갱신함
  - 현재의 메시지 블록으로부터 얻은 64-비트 값 Wt와
  - 덧셈 상수 Kt를 사용함

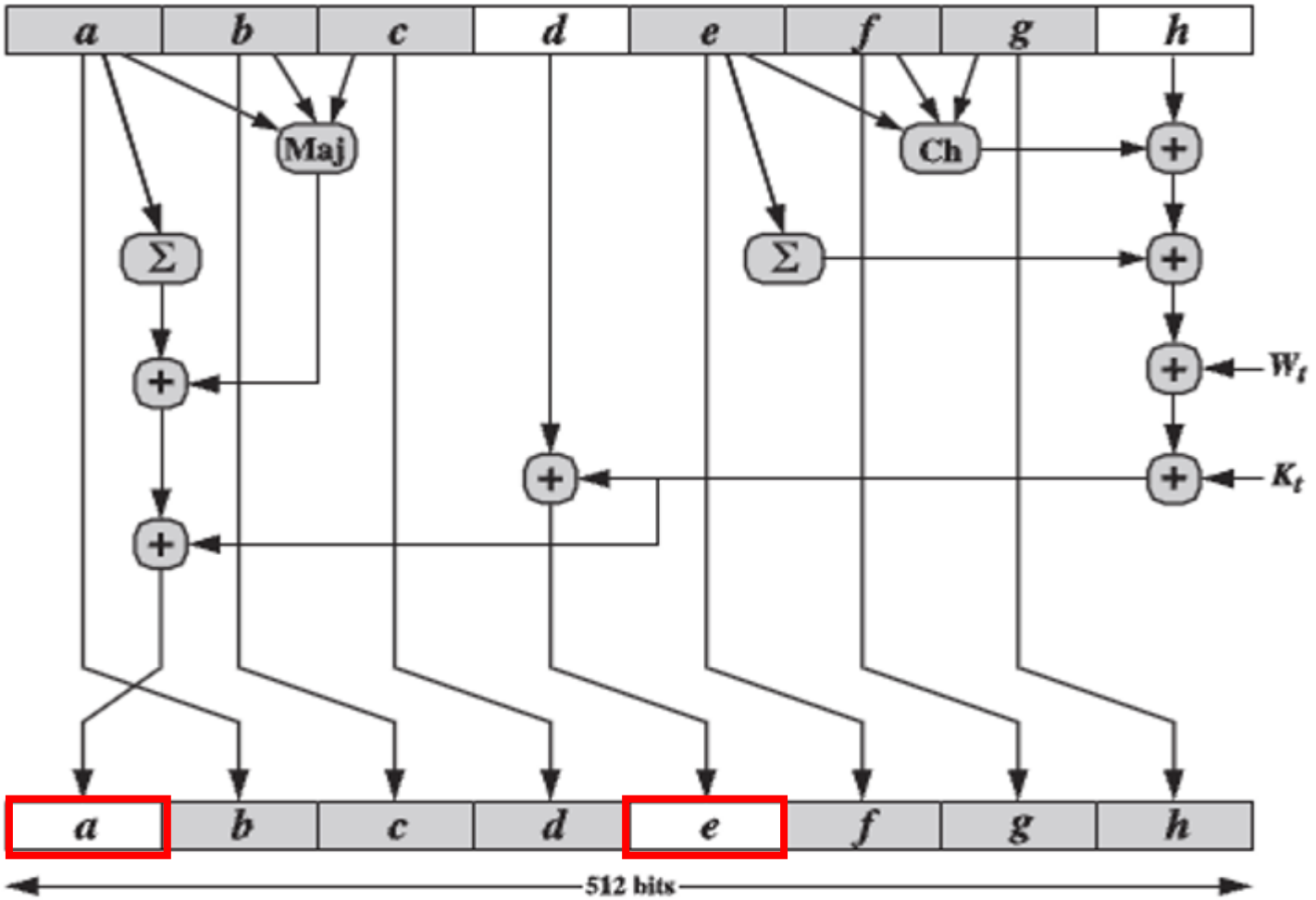
11.5 SHA (Secure Hash Algorithm)

SHA-512 논리

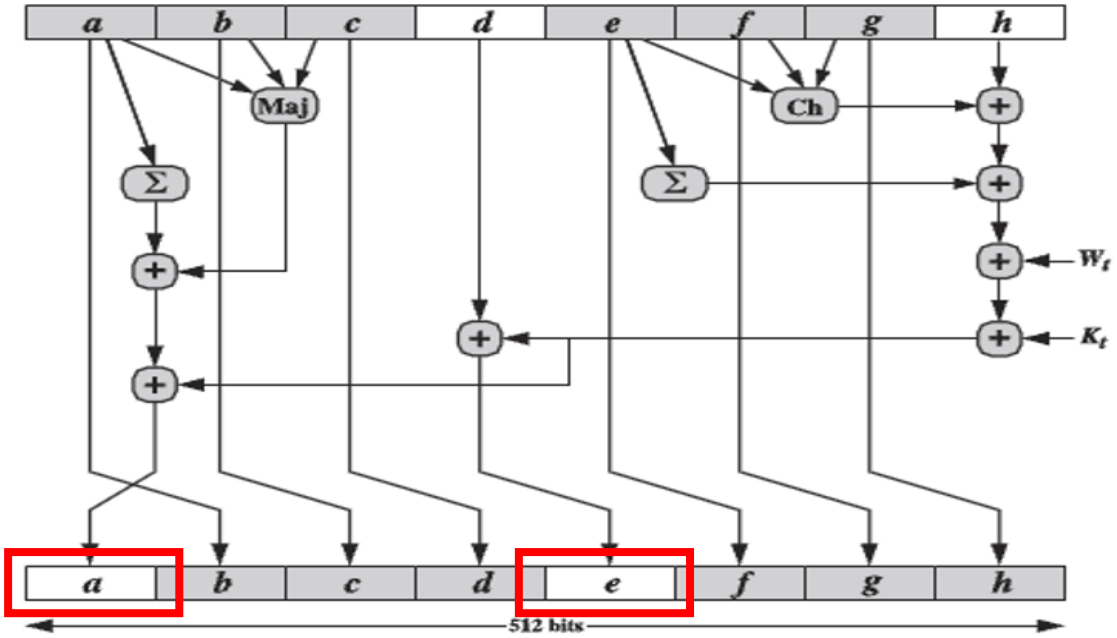


11.6 SHA-3

SHA-512 기본 동작 (80단계 중 한 단계)



11.6 SHA-3



11.6 SHA-3

SHA-512 기본 동작 (80단계 중 한 단계)

11.6 SHA-3

- 미국국립표준기술연구소가 2015년 8월에 발표한 암호화 해시 함수로, SHA-1과 SHA-2를 대체하기 위해 기획됨
- 공개적인 방식을 통해 후보를 모집한 다음 함수 안전성을 분석하여 몇 차례에 걸쳐 후보를 걸러내는 방식으로 진행

- NIST가 제안한 평가기준
  - 보안: SHA-3의 보안강도는 요구되는 여러 해쉬크기에 대하여 그리고 선이미지 회피성 및 충돌 회피성 모두에 대하여 이론적인 최대치에 가까워야.
  - 비용: SHA-3은 다양한 하드웨어 플랫폼 상에서 기간과 메모리 모든 면에서 효율적이어야.
  - 알고리즘과 구현 특성: 유연성(보안 성능 조절을 위하여 조정이 가능한 인자 등)와 간결성(알고리즘의 보안 특성 분석을 수월하게 해줌)을 가져야.

- 2012년 10월 1일에 귀도 베르토니조앤 데먼, 질 반 아쉐, 마이클 피터스가 설계한 Keccak이 SHA-3의 해시 알고리즘으로 선정됨.

Thank you