정보보호

(5111041)

2장

암호화 기법

대칭키 암호화 방식

- 송신된 데이터나 저장된 데이터에 기밀성을 제공하기 위한 일반적인 기법
- 관용 암호화 방식 또는 단일-키 암호화 방식이라고도 불림
- 대칭키 암호화 방식 사용의 요구사항 2가지:
  - 강력한 암호 알고리즘이 필요함
  - 송신자와 수신자는 비밀키(secret key)의 복제 본을 안전한 방식으로 획득하여 이를 안전하게 잘 보관해야 함

그림 2.1

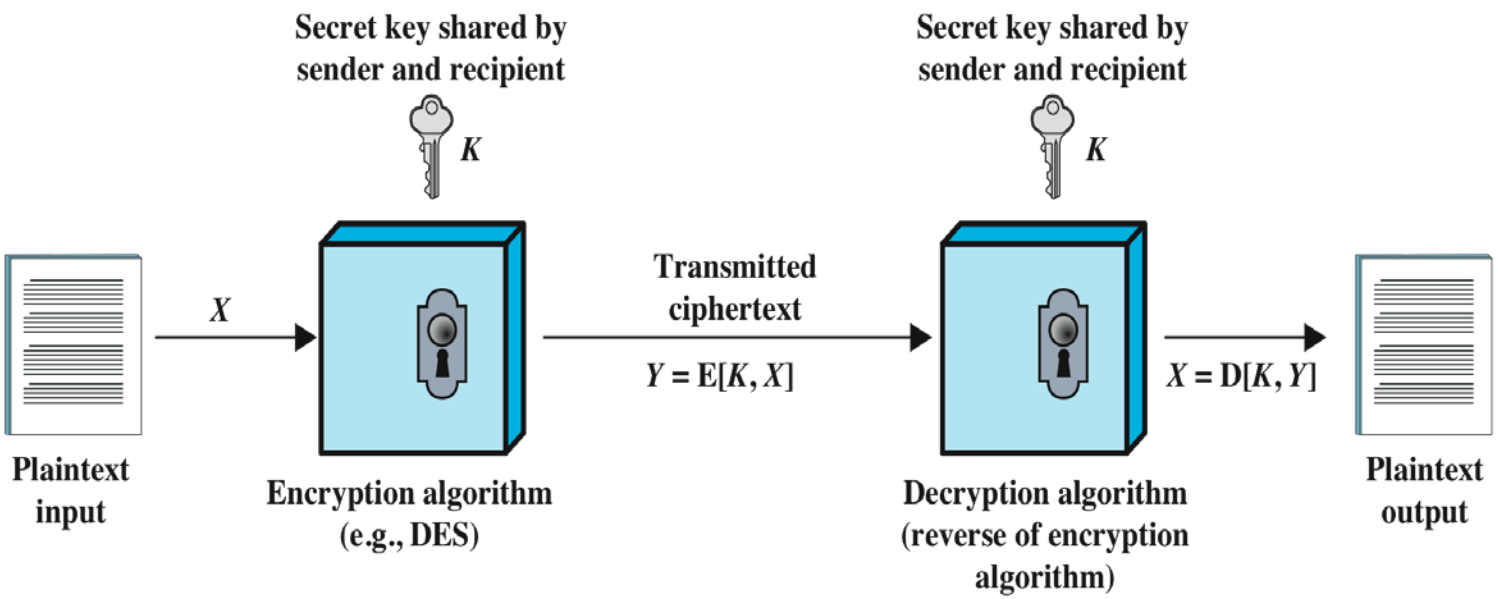


Figure 2.1 Simplified Model of Symmetric Encryption

대칭키 암호화 공격

암호 해독 공격

- 활용:
  - 알고리즘의 본질
  - 특정 평문의 일반적 특성
  - 몇 개의 평문-암호문 쌍에 대한 지식
- 특정 평문 또는 사용된 키를 추론하기 위해 알고리즘의 특성 활용
  - 추론을 성공할 경우 해독된 키로 암호화된 메시지는 안전하지 못하게 됨
  - Cryptanalysis
    - **에니그마의 해독**(Cryptanalysis of the Enigma)

무차별 대입 공격

- 하나의 암호문에 대하여 이해할 수 있는 평문으로 전환될 때까지 가능한 모든 키를 대입함
  - x 개의 키 옵션이 있을 경우, 평균 x/2 번의 시도 후 실제 키를 찾아냄
- Brute-force Attack

키 크기 (bits)	Number of Alternative Keys	1 Decryption/μs 에 요구되는 시간	10^6 Decryptions/μs 에 요구되는 시간
32	2^{32} = 4.3 □ 10^9	2^{31} μs = 35.8 minutes	2.15 milliseconds
56	2^{56} = 7.2 □ 10^{16}	2^{55} μs = 1142 years	10.01 hours
128	2^{128}= 3.4 □ 10^{38}	2^{127} μs = 5.4 □ 10^{24} years	5.4 □ 10^{18} years
168	2^{168} = 3.7 □10^{50}	2^{167} μs = 5.9 □ 10^{36} years	5.9 □ 10^{30} years
26 characters (permutation)	26! = 4 □ 10^{26}	2 □ 10^{26} μs = 6.4 10^{12} years	6.4 □ 10^6 years

키를 탐색하는 데 요구되는 평균 시간

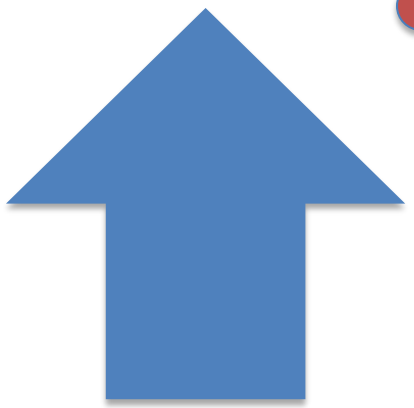
표 2.1

	DES	Triple DES	AES
평문 블록 크기 (bits)	64	64	128
암호문 블록 크기 (bits)	64	64	128
키 크기 (bits)	56	112 or 168	128, 192, or 256

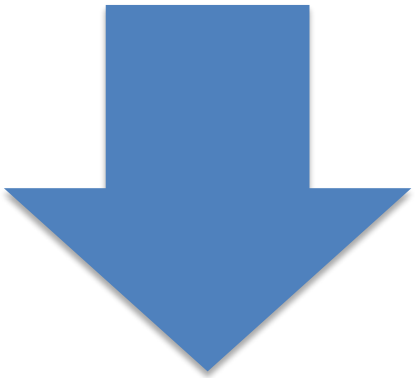
DES = Data Encryption Standard
AES = Advanced Encryption Standard

3가지 주요 대칭키 암호화 방식들의 비교

데이터 암호화 표준
(Data Encryption Standard: DES)



- **가장 널리 사용되고 있는 암호 알고리즘**
  - FIPS PUB 46
  - 데이터 암호화 알고리즘 이라고 불림 (Data Encryption Algorithm :DEA)
  - 64비트 평문 블록과 56비트 키를 이용하여 64비트 암호문 블록생성



- **염려사항:**
  - 알고리즘 자체에 관한 염려
    - DES 는 현존하는 암호 알고리즘 중 가장 많이 연구된 알고리즘으로서 몇가지 특징들로 해독이 용이함
  - 56비트 짧은 키의 사용
    - 전자프런티어재단(EFF)은 1998년 DES 암호화를 폐지할 것을 공표

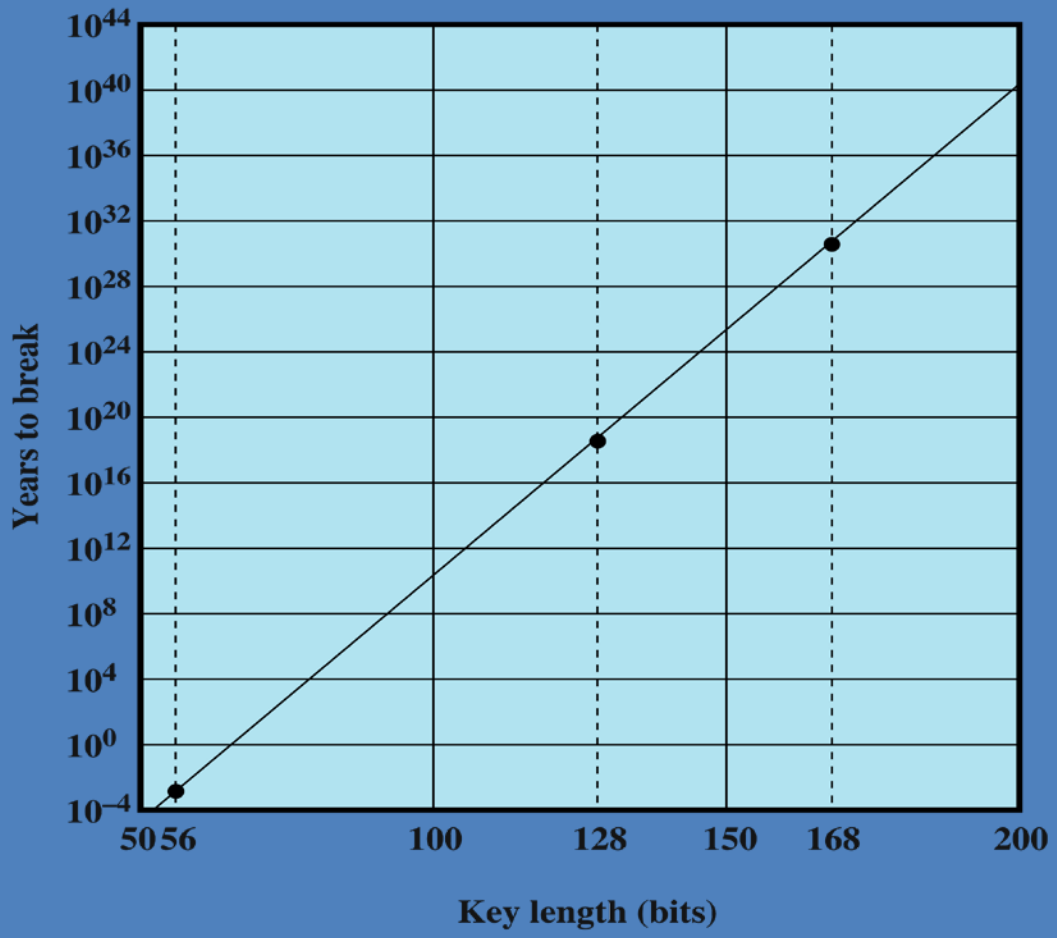


Figure 2.2 Time to Break a Code (assuming 106 decryptions/ms) The graph assumes that a symmetric encryption algorithm is attacked using a brute-force approach of trying all possible keys

트리플 DES (3DES)

- 2개 또는 3개의 고유 키를 이용하여 DES를 3번 반복하여 암호화 함
- 1985년에 ANSI X9.17에서 표준으로 지정하여 금융 기관에서 많이 사용되고 있음
- 특징:
  - 168비트의 키 길이로 DES의 무차별 대입 공격의 취약점 극복
  - 근본적인 암호화 알고리즘은 DES와 같음
- 문제점:
  - 소프트웨어에서 알고리즘의 성능이 떨어짐
  - 64비트 크기의 블록 사용

고급 암호화 표준

(Advanced Encryption Standard: AES)

3DES를 대체할 기술이 필요해짐

3DES는 오랜 기간 동안 사용하기에 비적합 함

NIST는 1997년에 AES를 제안

- 3DES와 동등하거나 더 뛰어난 보안 길이의 암호 기법이 필요 됨
- 효율성 면에서 높은 향상
- 대칭 블록 암호화
- 128 비트 데이터와 128/192/256 비트 키

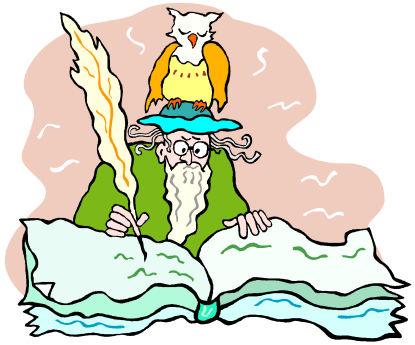
2001년 11월에 라인달(Rijndael)이 2011년 11월에 AES표준으로 선별

FIPS 197에 발행됨

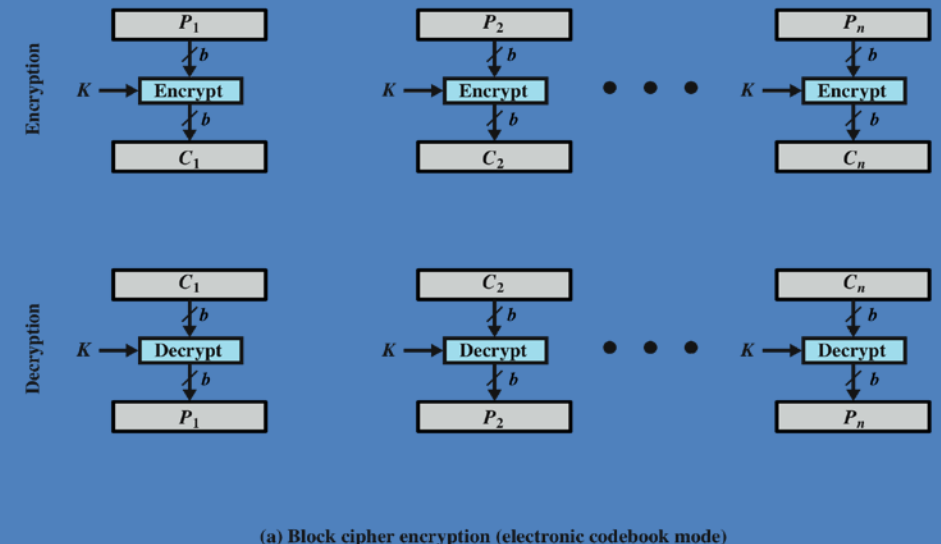
유용한 보안 이슈

- 전형적으로 대칭키 암호화 방식은 단일 64비트 또는 128비트 이상의 데이터 단위에 적용됨
- ECB(electronic codebook)모드는 블록 암호 방식의 가장 간단한 방식
  - 각각의 평문은 같은 키를 사용하여 암호화 됨
  - 암호 해독자는 평문에 있는 규칙들을 쉽게 이용할 수 있음
- 운용모드
  - 대형 시퀀스에 대한 대칭키 블록 암호화의 보안을 향상시키기 위한 대안 기법으로 개발됨
  - ECB모드의 취약점 극복

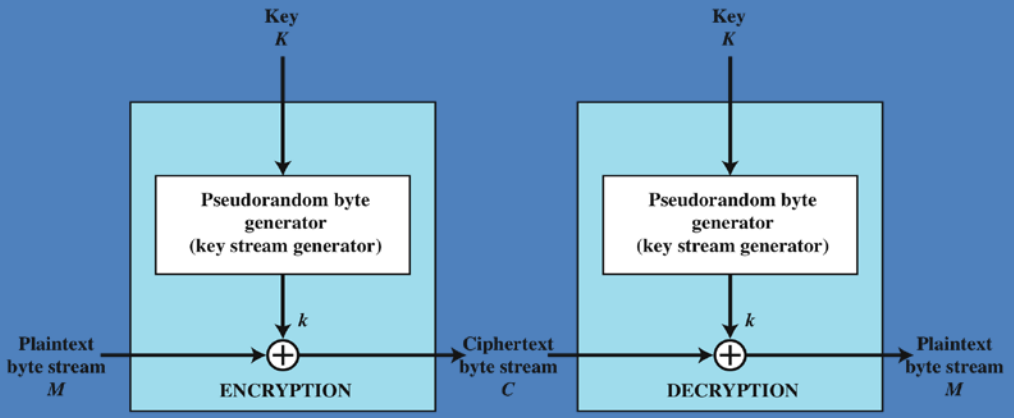
블록 암호화 방식



스트림 암호화



(a) Block cipher encryption (electronic codebook mode)



(b) Stream encryption

Figure 2.3 Types of Symmetric Encryption

블록 & 스트림 암호

블록암호

- 요소들을 한번에 하나의 입력 블록으로 처리
- 각 입력 블록에 대하 출력 블록 생성
- 키를 재사용 할 수 있음
- 보편적으로 사용되는 방식

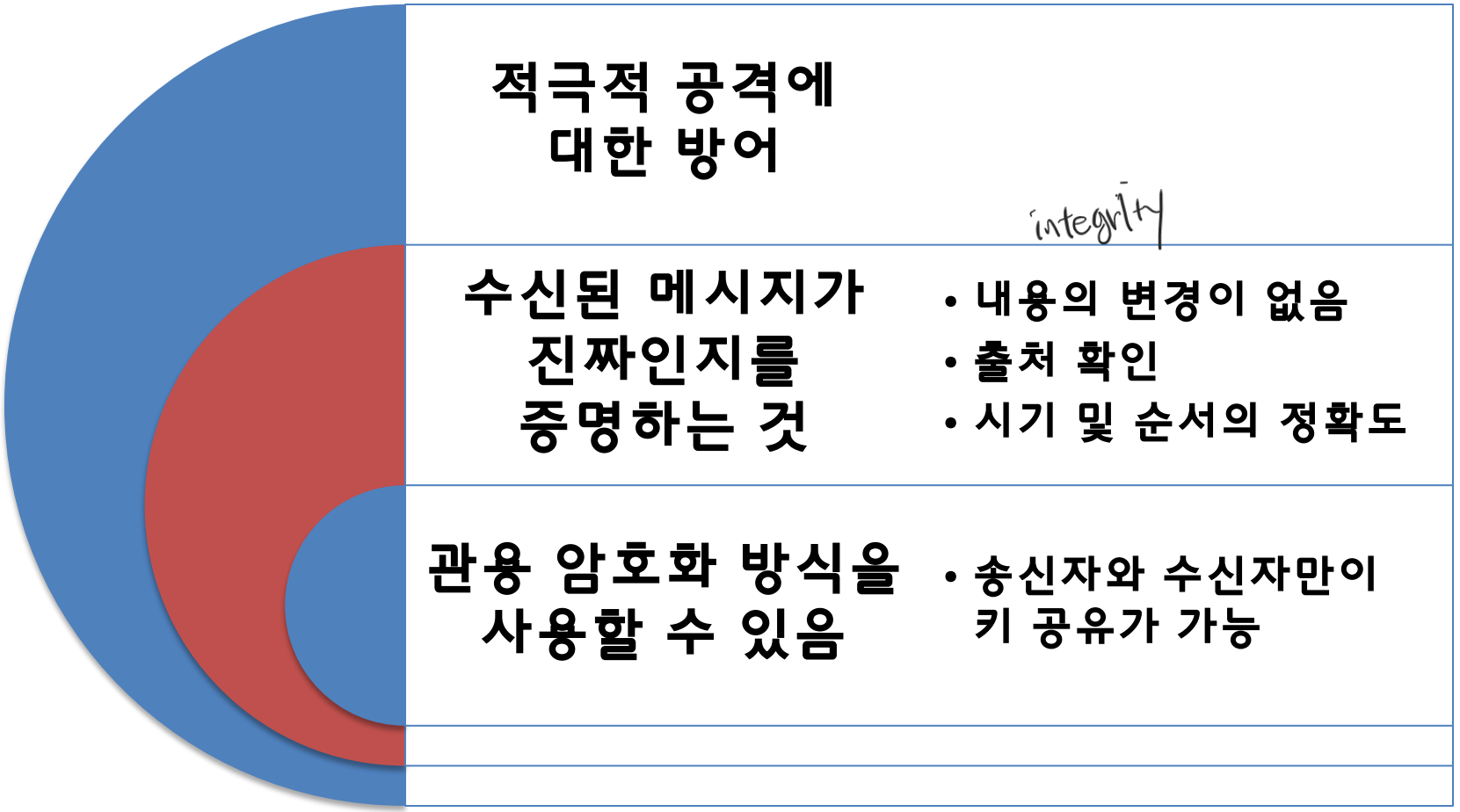
1. 하나 → 하나 출력
2. 재사용, 보편적.

스트림 암호

- 입력 요소들을 연속적으로 처리
- 한번에 하나의 출력 요소 생성
- 적은 코드의 사용과 어떠한 블록 암호보다도 빠르다는 것이 장점
- 한번에 평문을 한 바이트로 암호화함
- 입력 키의 정보가 주어지지 않을 경우 무작위 추출된 스트림은 예측이 불가능 함

1. 연속 → 하나
2. fast
3. 평문 → 한바이트 암호
4. 랜덤 → 예측 X.

메시지 인증


적극적 공격에
대한 방어
integrity
수신된 메시지가
진짜인지를
증명하는 것
• 내용의 변경이 없음
• 출처 확인
• 시기 및 순서의 정확도
관용 암호화 방식을
사용할 수 있음
• 송신자와 수신자만이
키 공유가 가능

메시지 인증 코드

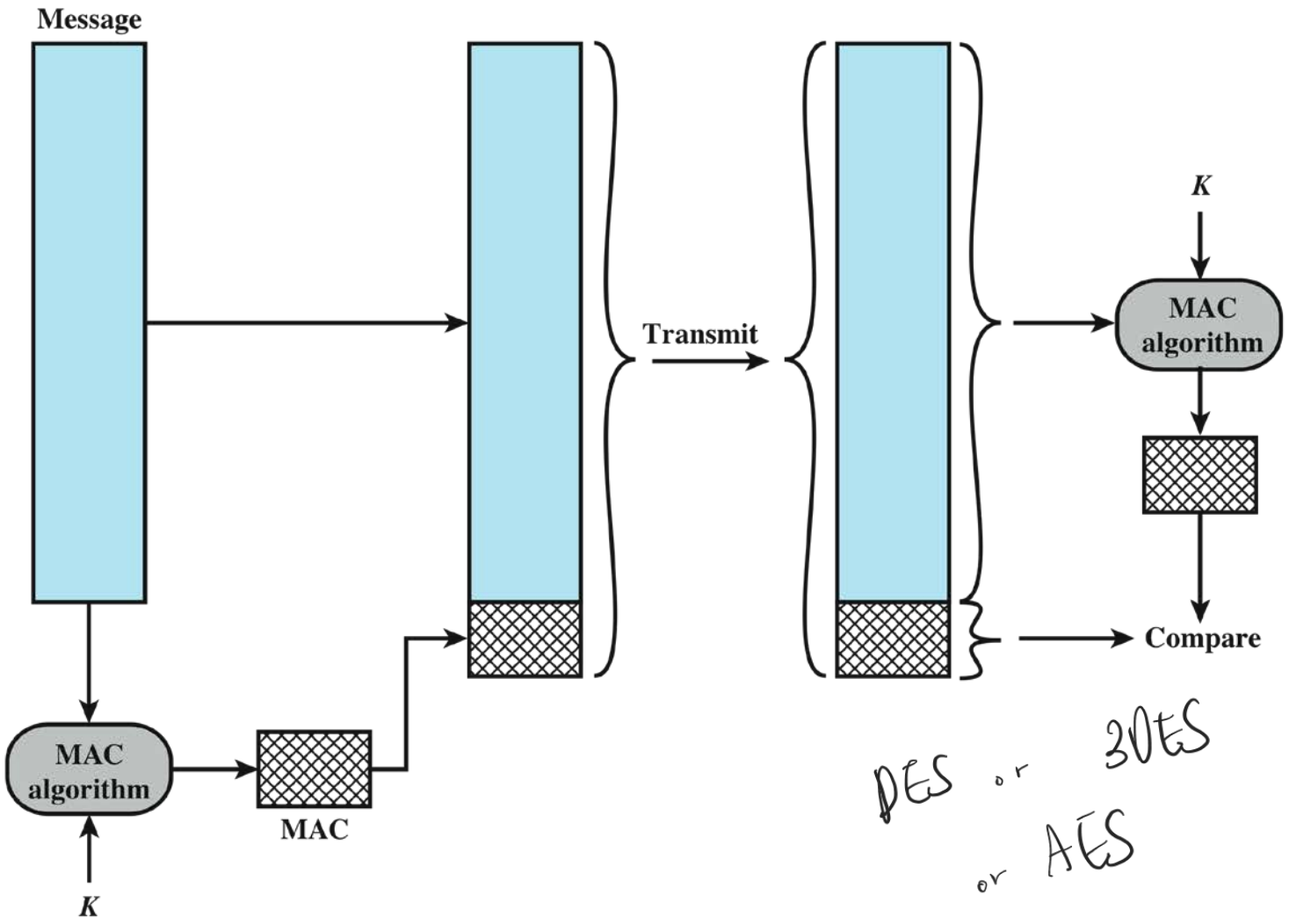


Figure 2.4 Message Authentication Using a Message Authentication Code (MAC). The MAC is a function of an input message and a secret key.

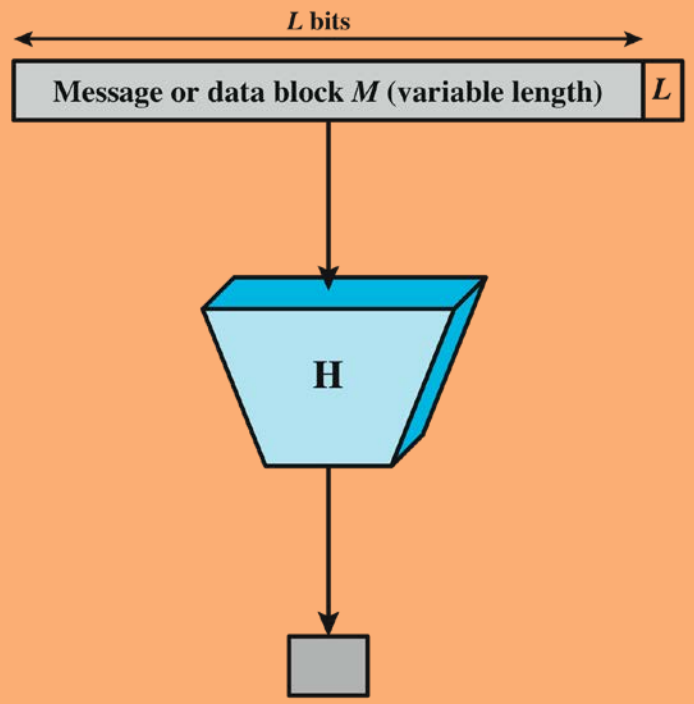


Figure 2.5 Block Diagram of Secure Hash Function; $h = H(M)$

input → length ∞ (variable)

output → fixed length.

안전한 해시 함수

그림 2.5

i → 0 easy
0 → i hard

단 방향 해시 함수를 이용한 메시지 인증

Keyed hash MAC (message authentication code) →

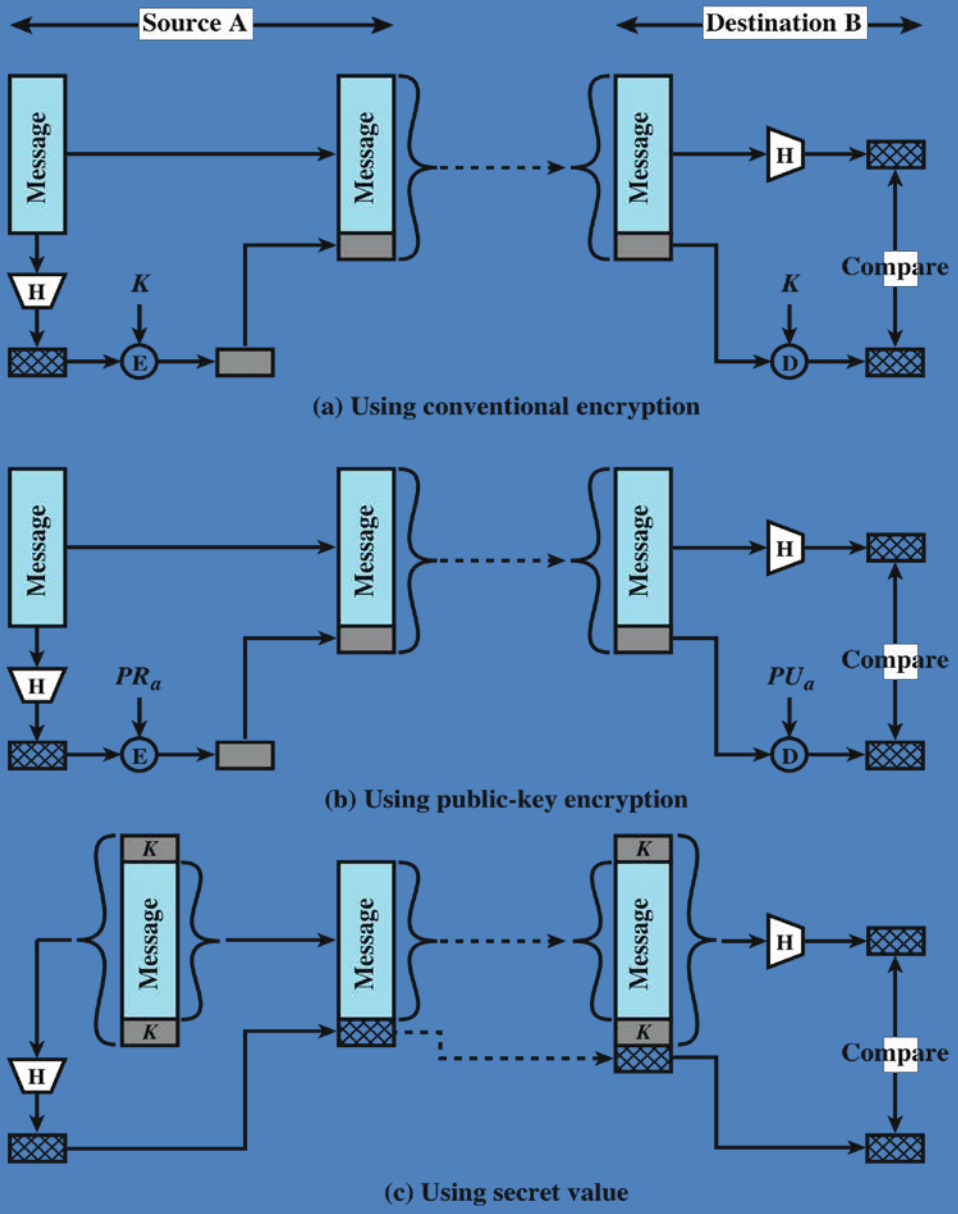


Figure 2.6 Message Authentication Using a One-Way Hash Function. The hash function maps a message into a relatively small, fixed-size block.

해시함수 요구사항

- 어떤 크기의 데이터 블록에도 적용이 가능함
- 고정된 길이의 출력 값 생성
- 주어진 어떤 *x*에 대해 H(x) 는 상대적으로 산출이 용이
- 단방향(one-way) 또는 역상저항성(pre-image resistant)
  - 주어진 h값에 대해 H(x) = h를 만족하는 x값을 산출하는 것은 불가능 해야 함
- 제2 역상저항성 또는 약한 충돌 방지
  - For given x, H(y) = H(x)인 y ≠ x 를 찾기 불가능 해야 함
- 충돌 저항성 또는 강한 충돌 방지
  - H(x) = H(y)인 어떠한 (x, y)쌍을 찾는 것은 불가능 해야 함

해시 함수의 보안성

- 안전한 해시함수를 공격하는 두 가지 접근:
  - 암호 해독 공격
    - 알고리즘의 논리적 약점 이용
  - 무차별 대입 공격
    - 해시 함수는 알고리즘에 의해 생성된 해시 코드의 길이에만 의존 한다는 점을 이용
- SHA 가장 널리 사용되는 해시 알고리즘
- 안전한 기타 해시 어플리케이션:
  - 패스워드
    - 해시 패스워드는 운영체제에 저장됨
  - 침입 탐지
    - 각 파일에 대한 H(F)를 시스템에 저장하여 해시 값이 안전하게 보안됨

SHA: Secure Hash Algorithm

공개키 암호화 구조

1976년에 Diffie와 Hellman이 제안

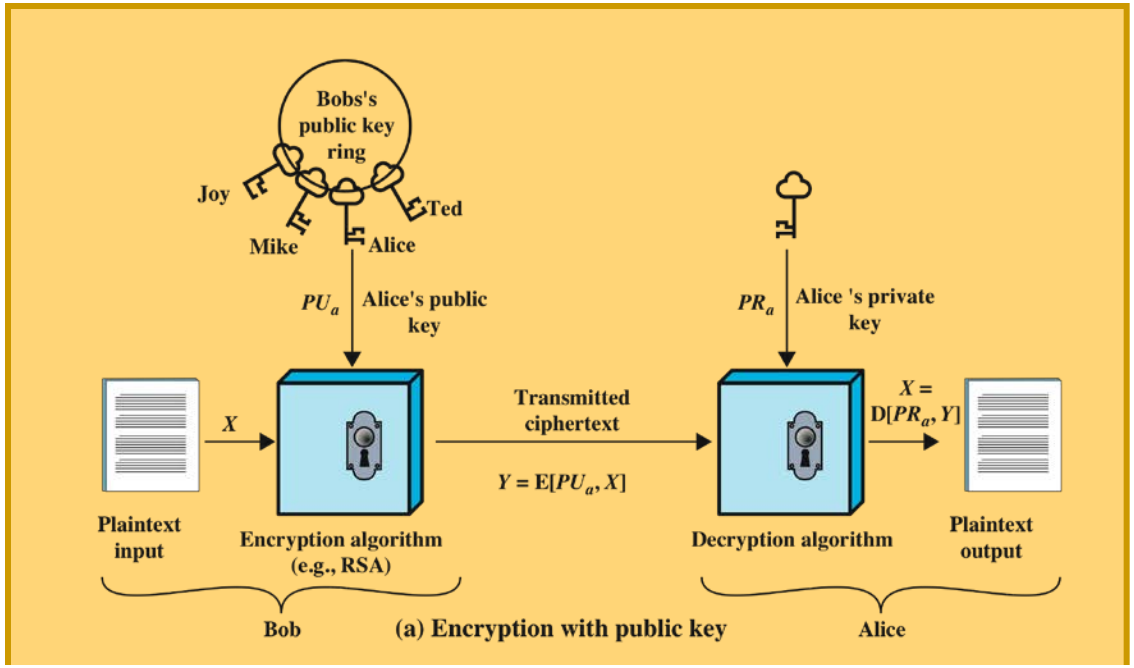
수학적 함수에 기반함

비대칭적 구조

- 두 개의 개별적 키 사용
- 공개키, 개인키
- 공개키는 모든 사람이 알 수 있도록 공개 됨

분산을 위해 특정 형식의 프로토콜이 필요함

그림 2.7a
공개키 암호화

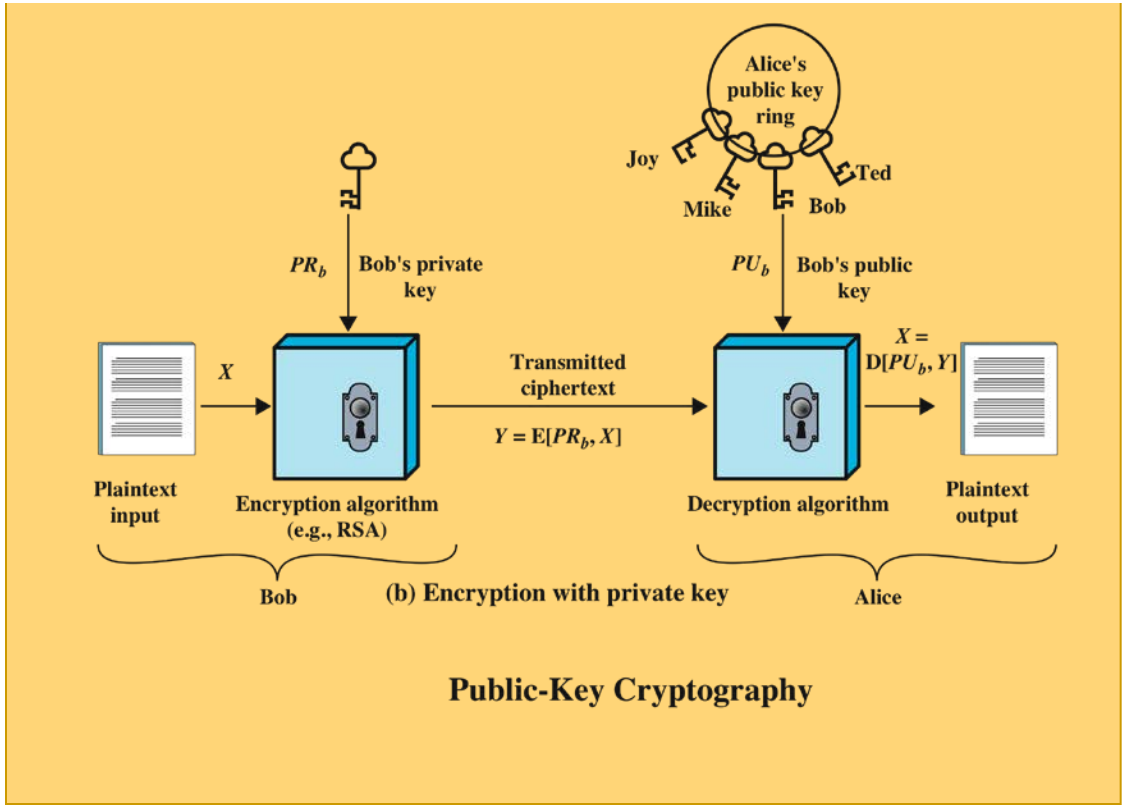


***directed toward providing confidentiality

- 평문
  - 알고리즘에 입력으로 제공되는 해독 가능한 메시지 또는 데이터
- 암호화 알고리즘
  - 평문을 암호문으로 변환하는 일 수행
- 공개 키, 개인 키
  - 암호화 또는 복호화를 위한 키 쌍
- 암호문
  - 출력 값으로 해독 불가능한 형태의 메시지를 출력
- 복호화 키
  - 본래의 평문 생성

RSA (Rivest–Shamir–Adleman)

그림 2.7b
개인 키 암호화



- 사용자는 본인의 개인키를 이용하여 데이터를 암호화 함
- 공개키를 아는 사람은 누구든지 메시지를 해독 할 수 있음

***directed toward providing authentication

공개키 암호화 방식에 대한 요구사항

키 쌍의 생성이 용이

키가 각 역할에 유용하게 사용되어야 함

공개키를 아는 송신자는 메시지 암호화 하는 것이 용이

공개키와 암호문으로 부터 메시지의 복구가 어려워야 함

개인키를 아는 수신자는 암호문 해독이 용이

공개키로 부터 개인키를 결정하는 것은 어려워야 함

비대칭 암호화 알고리즘

RSA (Rivest, Shamir, Adleman)	1977년에 개발됨	공개키 암호화에 가장 널리 사용되고 구현되는 방식	평문과 암호문이 어떤 수 n에 대하여 0과 n-1의 사이의 정수로 이루어진 블록 암호
Diffie-Hellman의 키 교환 알고리즘	차후 대칭 암호화 메시지에 비밀키로 사용될 비밀키를 두 사용자가 비밀리에 공유하기 위한 방법	키 교환에 제한이 있음	
전자 서명 표준안(DSS)	SHA-1을 사용한 전자서명만을 제공	암호화 또는 키 교환에는 사용될 수 없음	
타원 곡선 암호(ECC)	더 짧은 키 사이즈로 RSA와 대등한 안정도를 가짐		

전자 서명

- 자원과 데이터의 무결성 인증에 사용됨
- 개인키를 가진 해시 코드로 생성
- 기밀성을 제공 하지는 않음
  - 완벽한 암호화가 되어 있는 경우라 할지라도 메시지 변경에 대한 안전은 보장되지만 도청에 대한 안전은 보장 되지 않음

공개키 인증서

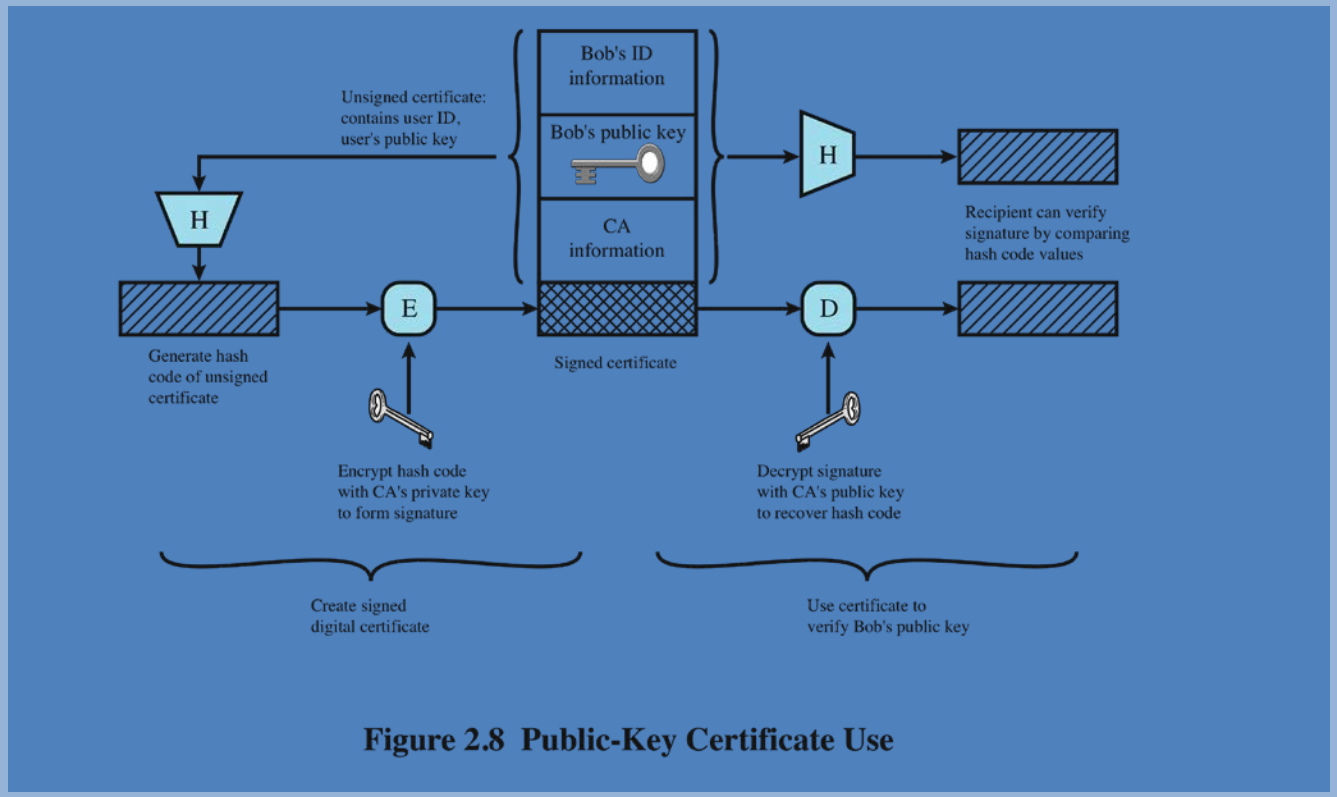


Figure 2.8 Public-Key Certificate Use

전자 봉투

- 송신자와 수신자가 동일한 비밀키를 공유하지 않고서도 메시지를 보호하는 데 이용됨

***무기명 편지를 포함한 봉인된 봉투로 간주

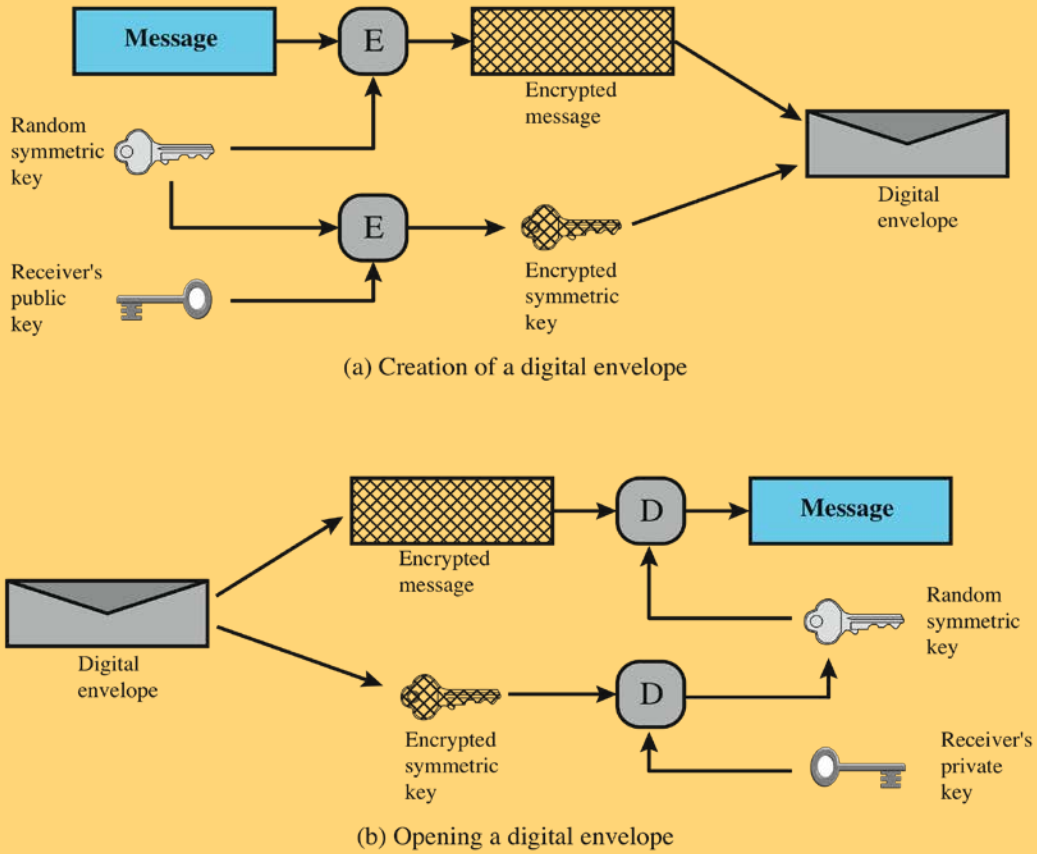


Figure 2.9 Digital Envelopes

난수

활용:

- 공개키 암호화 알고리즘 키
- 대칭 스트림 암호의 스트림 키
- 세션키나 전자 봉투를 만드는데 사용되는 대칭 키
- 재사용공격(reply attacks)을 예방하기 위한 핸드셰이킹
- 세션키

난수 요구사항

임의성

- 기준:
  - 균등 분포
    - 숫자들이 발생하는 빈도수는 대략적으로 같아야 함
  - 독립성
    - 순서상 어떤 값도 다른 값으로부터 추론 불가능

비예측성

- 통계적으로 각 숫자는 연속하는 다른 숫자에 대하여 독립적
- 초기 요소에 근거하여 미래 요소들을 예측할 수 없음

랜덤 VS 의사랜덤

- 일반적으로 암호 어플리케이션들은 난수 생성 알고리즘 기법을 활용함
  - 알고리즘들은 결정성이 있으므로 통계적으로 랜덤하지 않은 일련의 숫자들을 생성
- 의사 난수:
  - 통계적인 임의성 테스트를 만족시키기 위해 생성된 일련의 수
  - 예측 가능
- 난수 발생기(True Random Number Generator: TRNG)
  - 임의성을 추출하는데 비 결정성의 소스 활용
  - 대부분 예측 불가능한 자연적 현상에서 측정됨
    - e.g. 이온화 방사선 추이의 펄스 탐지기, 가스 방출 튜브, 누전 축전기
  - 점차 현대 프로세서에 제공되고 있음

요약

- 대칭키 암호화
  - 단일키를 사용하고 관용적으로 쓰이는 암호화 방식
  - 5가지 요소: 평문, 암호화 알고리즘, 비밀키, 암호문, 복호화 알고리즘
  - 공격 유형: 암호 해독 공격, 무차별 대입 공격
  - 블록 암호기법이 일반적으로 사용됨 (DES, 트리플 DES, AES)
- 해시 함수
  - 메시지 인증
  - 전자 서명 생성
- 공개키 암호화
  - 수학적 함수에 기반함
  - 비대칭 적
  - 6가지 요소: 평문, 암호화 알고리즘, 공개키와 개인키, 암호문, 복호화 알고리즘
- **전자서명**
  - **개인키로 암호화 되는 해시코드 사용**
- **전자 봉투**
  - **송신자와 수신자가 동일키를 공유할 필요 없이 메시지를 보호할 수 있음**
- **난수**
  - **요구사항: 임의성, 비예측성**
  - **검증: 균등 분포, 독립성**
  - **의사 난수**