컴퓨터 네트워크

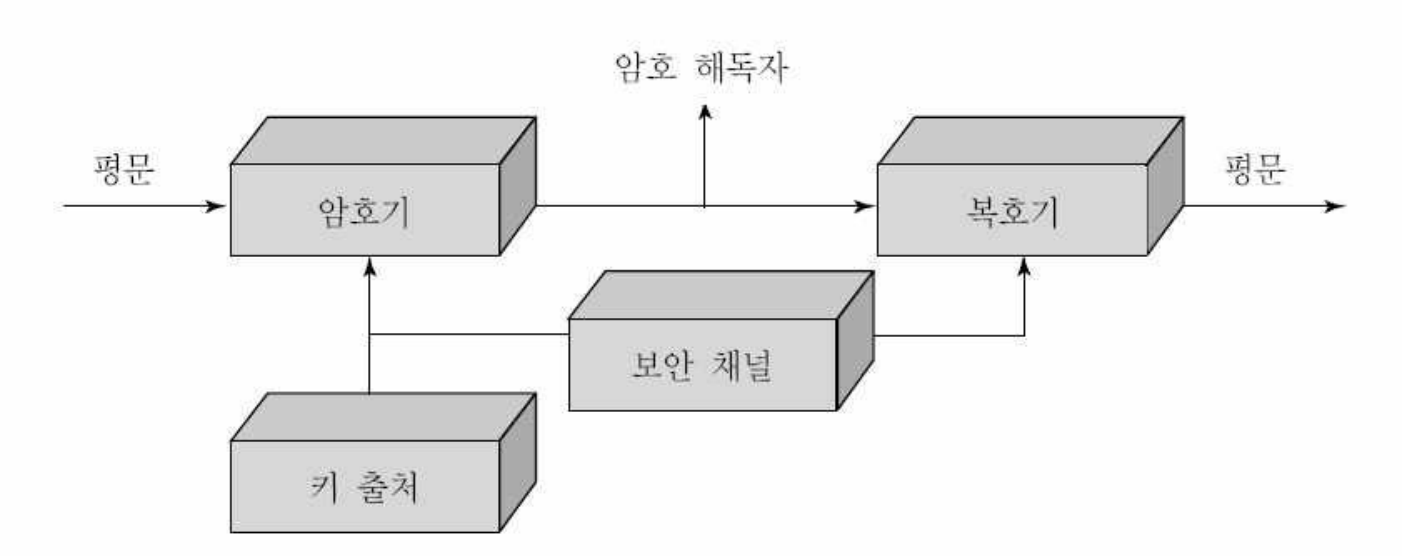
13장. 네트워크 보안 (2)
- 암호화 시스템

이번 시간의 학습 목표

- 암호화 알고리즘인 DES, RSA의 구조 이해
- 전자 서명의 필요성과 방법 이해

대칭키 암호 방식 (1)

- 암호화와 복호화에 하나의 키를 이용
- 공통키 또는 대칭키 암호방식이라고 지칭
- 이때의 키를 비밀키(secret key)라고 지칭



대칭키 암호 방식 (2)

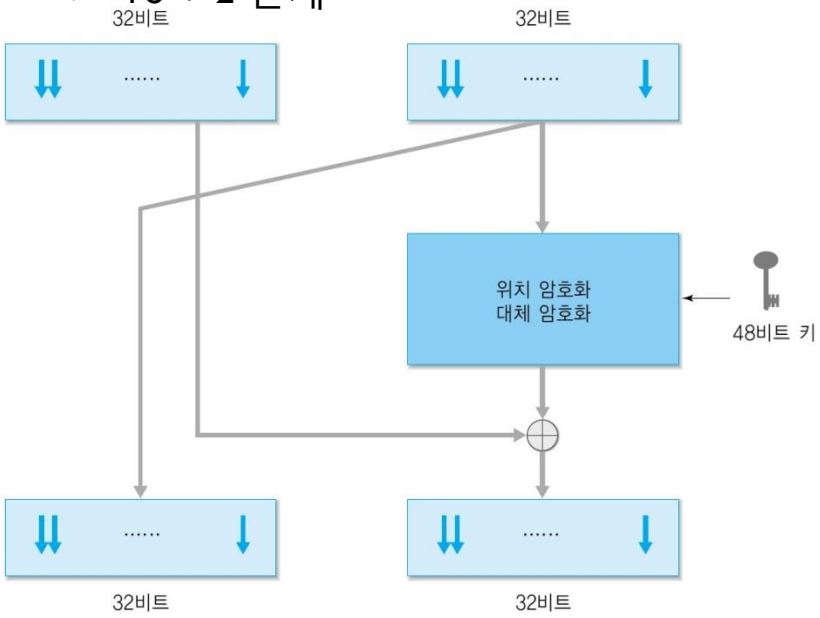
- 암호화 복호화를 수행하는 두 사용자가 동일한 키를 가지고 있어야 함
  - Pre-shared key
  - 온라인 상에서 구두 또는 메일, 전화로 교환 : 수동키
  - 블록 암호와 스트림 암호로 분류
  - 대표적 알고리즘 : DES, 3DES, SEED, RC2, RC5, AES(Rijndael)

대칭키 암호 (3)

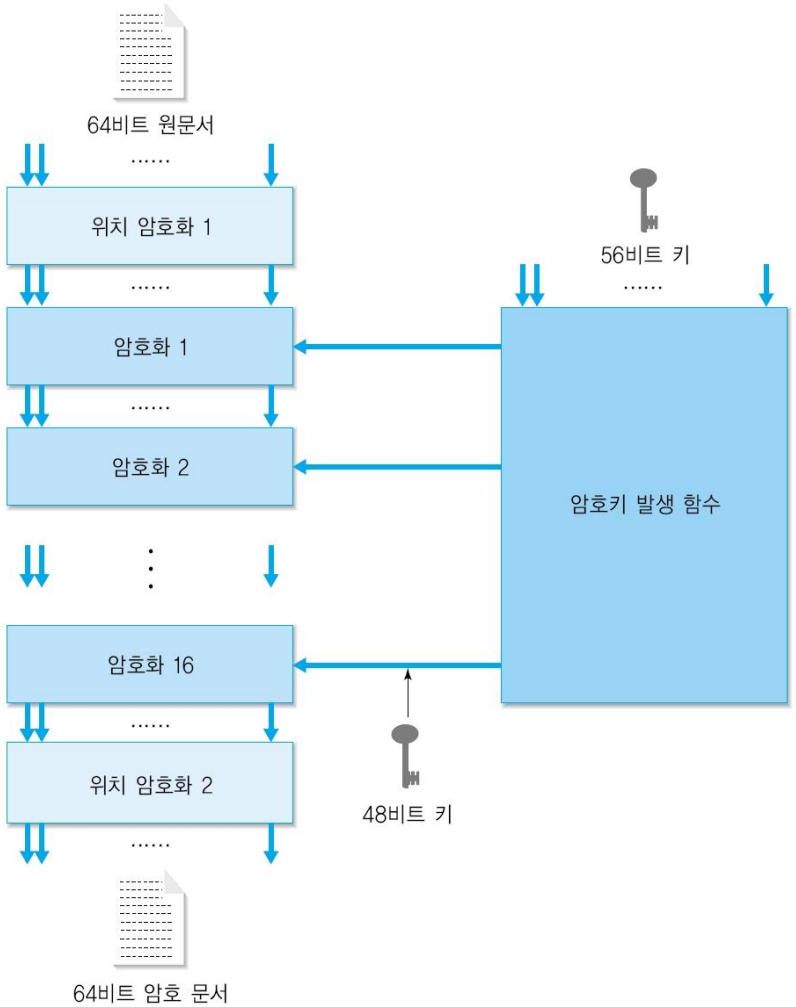
- 블록 암호
  - 특정 블록 크기로 암호화/복호화를 수행하여 스트림 암호에 비해 속도가 빠름
  - 블록 간의 연관성 때문에 오류 발생시 전체 데이터에 영향을 미침
- 스트림 암호
  - 1970년대 초 유럽에서 연구
  - 비밀키를 상호 공유하고, 사용한 비밀키는 재사용되지 않는 특징
  - 비트열에 오류가 발생해도 오류 확산이 없다는 장점
  - 1비트씩 연산을 하므로 수행속도가 느리다는 것과 비밀키를 안전하게 전송해야 하는 단점

DES 알고리즘

- 대칭키 알고리즘
- 동작 방식
  - 암호키: 56 비트
  - 64 비트 단위로 암호화
  - 16 단계의 암호화 과정을 수행
    - 16 + 2 단계



[그림 13-4] [그림 13-3]의 16단계 암호화 알고리즘



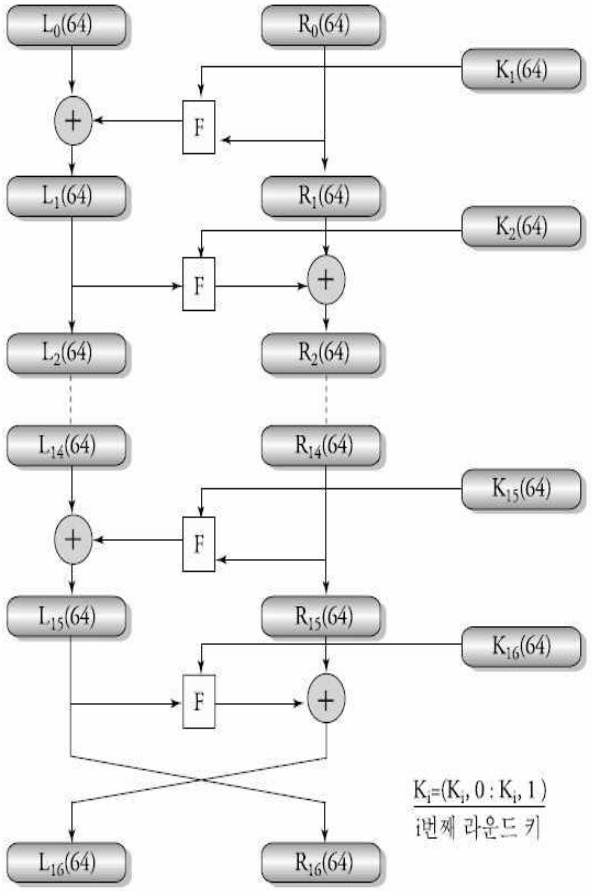
[그림 13-3] DES 알고리즘 동작 과정

DES의 안전성

- 키가 56비트이므로 2^{56}개 키 존재
- 1977년 Diffe-Hellman에 의해 1,000,000대의 병렬 컴퓨터로, 1usec에 1번 encryption이 가능하다면 10시간 이내 찾을 수 있다고 제안
- Wiener에 의해 Known Plain-text Attack으로 정확히 분석
- 1997년 DES 키를 찾는 프로젝트에서 96일만에 키를 찾아냄
- 3DES로 키 길이와 라운드 수를 3배로 증가시킴

SEED

- 국내 대표적인 암호화 알고리즘
  - DES와 같은 Feistel 구조
  - 128비트 키
  - 128비트 고정 길이 입출력
  - Known Attack에 강한 라운드 기능
  - 4개의 8x8 S-Box
  - XOR과 Modular의 혼합된 연산
  - 16 라운드 수행



AES

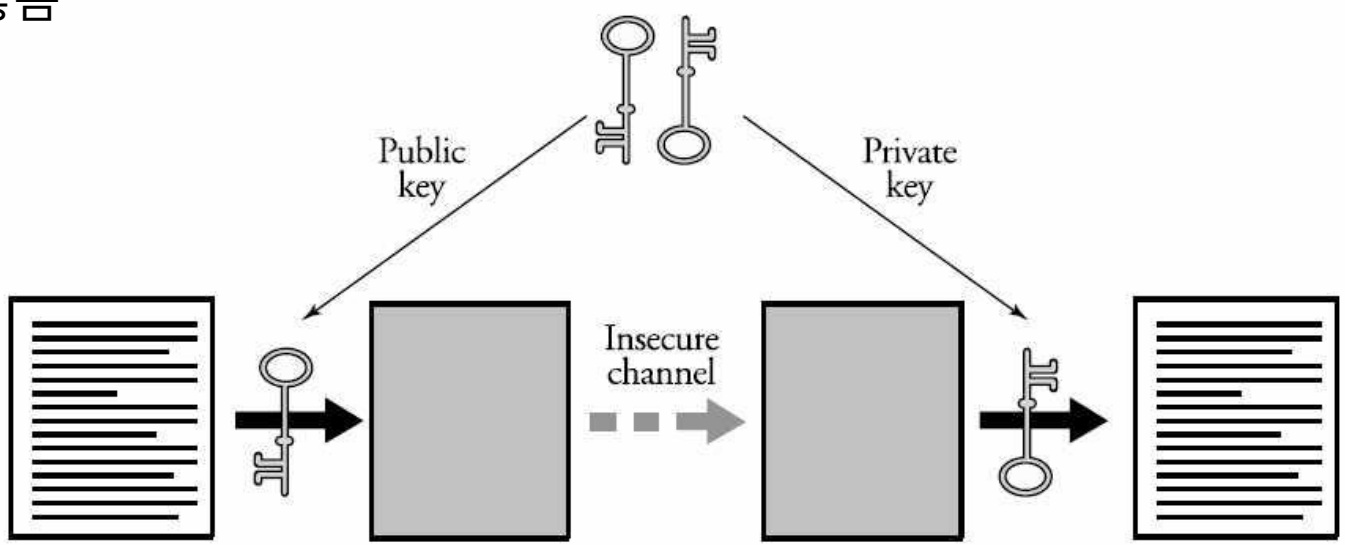
- 1998년 사용기한이 만료된 DES를 대체할 알고리즘으로 공모
- 벨기에에서 개발한 'Rijndael'이 선정되어 2000년 10월 표준으로 선정
- 특징
  - 가변 블록길이(128, 192, 256) 지원
  - 키도 128, 192, 256비트 사용
  - 키 길이에 의해 라운드 결정
  - Feistel 구조가 아닌 레이어(layer)로 구성
    - 선형 혼합(Linear mixing) : 라운드
    - 비선형(Non-linear) : S-Box
    - 키 추가(Key addition) : 라운드 키의 XOR

대칭키 알고리즘 비교

	DES	SEED	AES
개발	미국	한국	벨기에
구조	Feistel 구조	Feistel 구조	Layer
크기	64	128	variable
키 길이	56(DES)/168(3DES)	128	128, 192, 256, …
키 취약성	yes	no	no
암호/복호화 방식	Block	Block	Block

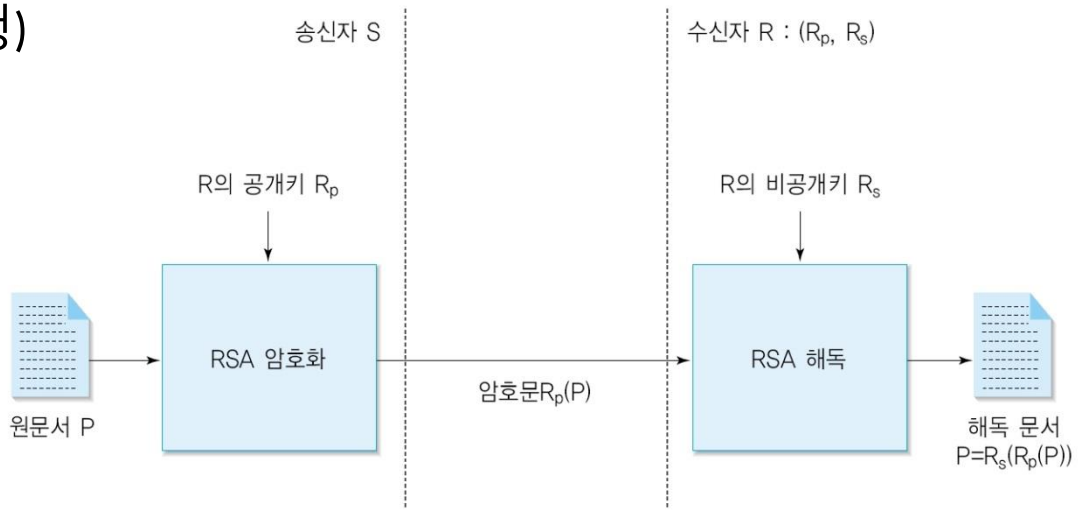
비대칭키 암호

- 1976년 Diffie와 Hellman에 의해 키 분배 방식알고리즘 발표 이후 많은 알고리즘이 제안됨
- 두 키가 서로 다르므로 '비대칭'이라고 부르며, 두 키가 공개키와 비밀키로 명명되어 '공개키 암호'라고 부름
- 비밀키 보관에 따라 안전도가 좌우되고, 통신 상대의 확인에 디지털 서명 사용이 가능하고, 키 관리에 뛰어남
- 상대적으로 암호화 속도가 느려 직접데이터를 암호화하는 데에는 사용되지 않음



RSA 알고리즘 (1)

- RSA(Rivest, Shamir, Adelman)
  - 1978년 MIT의 Rivest, Shamir, Adelman에 의해 제안
    - 비대칭키의 공개키 알고리즘
      - 공개키: 원문서를 암호화하는 용도로 사용 (모든 사람이 암호화 과정 수행)
      - 비공개키: 암호문을 해독하는 용도로 사용 (특정인만 해독 과정 수행)



[그림 13-5] RSA 알고리즘

RSA 알고리즘 (2)

- 암호화 과정
  - 소인수 분해의 복잡성을 이용하여 구현
  - 가입자는 두 개의 소수 p, q 선택하여 n = p q 계산
  - p, q를 알고 있는 사용자는 n을 계산하기 쉽지만, n만 가지고는 p, q를 유추하기 어려움

Key 생성	
$p,\ q$	prime number $(p \neq q)$
$n = p \times q$	
$\phi(n) = (p-1)(q-1)$	
정수 e 선택	$\gcd(\phi(n_B), e_B) = 1; 1 \leq e \leq \phi(n)$
d 계산	$d \equiv e^{-1} (\mathrm{mod} \phi(n))$
공개키	$PU = \{e, n\}$
개인키	$PR = \{d, n\}$

RSA 알고리즘 (3)

- 암호화

암호화	
평문	$M < n$
암호문	$C = M^e \bmod n$

- 복호화

복호화	
암호문	C
평문	$M = C^d \bmod n$

RSA 알고리즘 (4)

- RSA 암호의 안전성
  - 소수 p와 n에 달려있음
  - 공개키 e와 n으로 비밀키 d를 찾을 수 있으면 쉽게 해독됨
  - n으로부터 p,q를 찾을 수 있으면 n의 소인수 분해가 가능하고, 오일러 함수를 찾게 되어 e로부터 d를 찾아낼 수 있음
  - 부가 조건
    - p와 q는 거의 같은 크기의 소수
    - p - 1과 q – 1은 큰 소수를 인수로 가져야 함
    - p – 1과 q – 1의 최대공약수는 작아야 함
  - 현재까지 p, q의 크기가 100자리이고, n이 200자리인 합성수의 경우 n의 소인수분해가 거의 불가능한 것으로 알려짐
  - e와 d의 크기가 너무 작아도 안되지만, 지나치게 크면 연산 양이 많아져서 속도가 저하됨
  - 상용장비의 경우 512비트의 n, 약 155자리 수
  - 연산 부하 증가로 상용화에 어려움이 있음

그외 비대칭키 알고리즘

- ElGamal
  - 이산대수 문제를 근간으로 만들어진 공개키 기반 암호 알고리즘
- ECC
  - ElGamal의 이산대수 문제 대신 타원곡선 이산대수 문제를 응용한 것

비대칭키 알고리즘 비교

	RSA	ElGamal	ECC
수학적 문제	소인수 분해	이산대수	타원곡선 이산대수
키 크기	크다	크다	작다
속도	비교적 느리다	비교적 느리다	빠르다
암호문 크기	-	평문의 두배	-
메모리	ElGamal에 비해 적음	가장 많이 차지	가장 적게 차지
비용	많이 소요	많이 소요	적게 소요
통신	유선	유선	무선

대칭키와 비대칭키 알고리즘 비교

대칭키	비대칭키
암호화/복호화에 동일한 키 사용	암호화/복호화에 각기 다른 키 사용
수신자와 송신자의 키 교환 필요	수신자와 송신자는 연관된 쌍 중 하나를 알아야 함
공유한 키를 비밀로 유지	키 쌍 중 하나(개인키)를 비밀로 유지
디지털 서명 불가능	공개키를 이용한 디지털 서명 가능
속도가 빠름	속도가 느림

전자서명 (1)

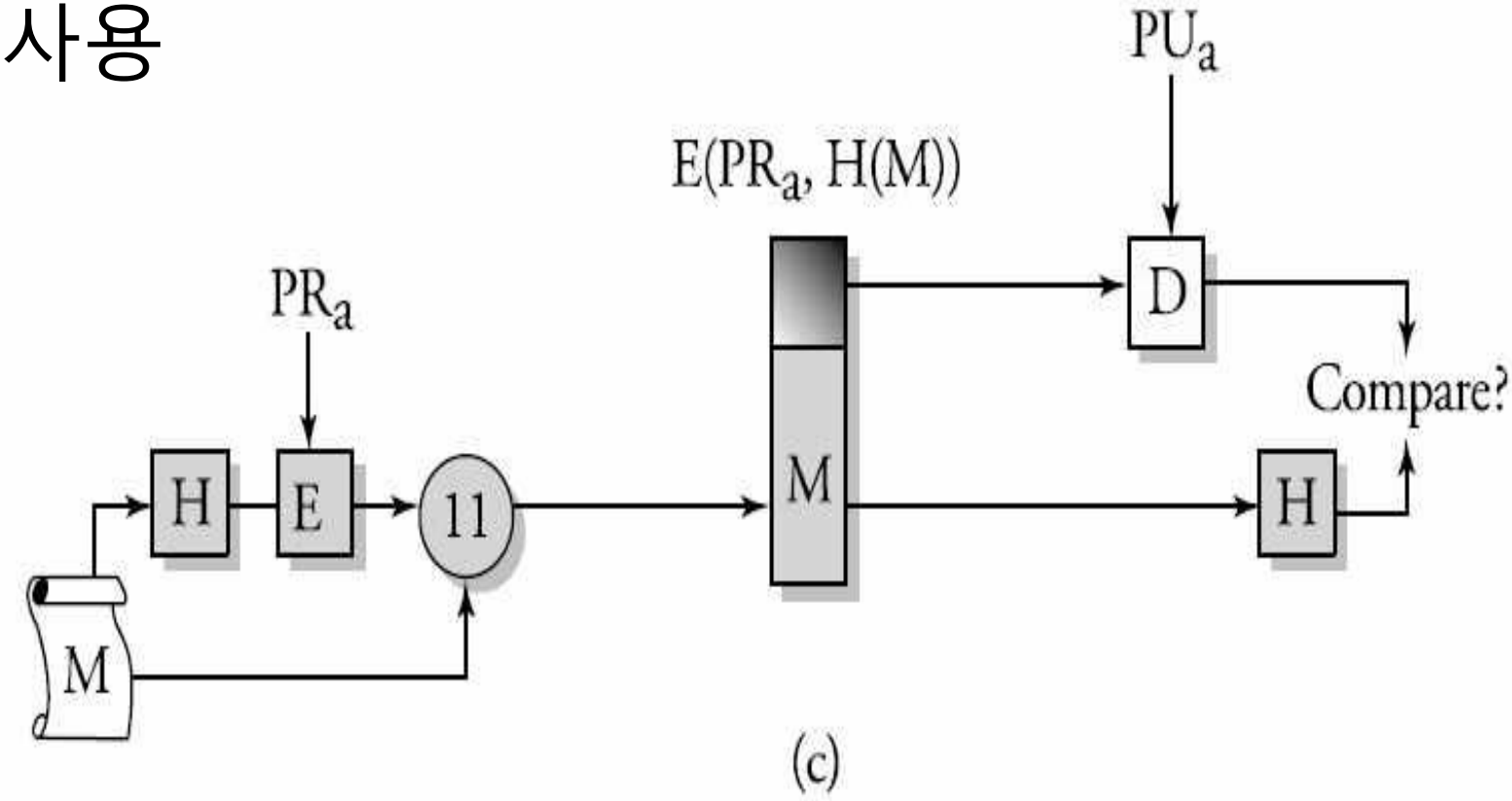
- 전자서명의 조건
  - 위조 불가
    - 서명자만이 서명 생성 가능
  - 서명자 인증
    - 서명자의 신분 확인 가능
  - 재사용 불가
    - 다른 문서의 서명으로 사용 불가능
  - 변경 불가
    - 서명된 문서 내용 변경 불가
  - 부인 불가
    - 서명한 사실 부인 불가

전자서명 (2)

- 전자서명 알고리즘
  - 공개키 암호방식을 이용한 서명 방식
  - 서명자가 비밀키로 서명을 생성하고, 검증자가 공개키로 확인하는 시스템
  - 직접 서명 방식
    - 송신자와 수신자 간에 직접 서명 및 검증
  - 중계 서명 방식
    - 중재자를 통해 확인
    - 통신 전에 정보 공유가 필요 없고, 외부로부터 공격에 강하며, 시간 확인까지 가능

전자서명 (3)

- 해쉬함수와 비대칭키 알고리즘 결합하여 사용



해쉬함수 (1)

- MD5 (Message Digest Version 5)
  - 512비트 입력 128비트 출력
  - 충돌회피성에 대한 문제로 인해 기존 응용과 호환으로만 사용 제한
- MD4 (Message Digest Version 4)
  - 1990년 Rivest가 개발
  - 메시지를 128비트로 압축
  - MD5보다 약간 빠르고, 안전성 측면에서는 다소 떨어짐
- SHA (Secure Hash Algorithm)
  - NIST에 의해 1993년 FIPS PUB 180으로 표준화
  - MD4와 유사하게 설계
  - 512비트 단위로 메시지를 입력하여 160비트 해쉬값 출력 (입력 전 메시지 길이를 512 비트 정수배로 조정)

해쉬함수 (2)

- 일반적으로 MD5가 많이 사용되고 있음
  - 취약성이 발견되어 제한적 사용 권고
- SHA-1은 디지털 서명에 사용하도록 제안됨
- AES의 128, 192, 256비트에 적용하도록 SHA256, SHA382, SHA512로 확장
- RIPE-MD-128, RIPE-MD-160, RIPE-MD-256, RIPE-MD-320은 MD5를 대신할 수 있도록 제안
  - RIPE-MD-128은 충돌저항성 문제가 있음
  - RIPE-MD-160은 효율성은 낮지만 높은 안전성으로 널리 사용 중

질의 / 응답