정보보호

05 암호개론 (2)

현대 암호 (1)

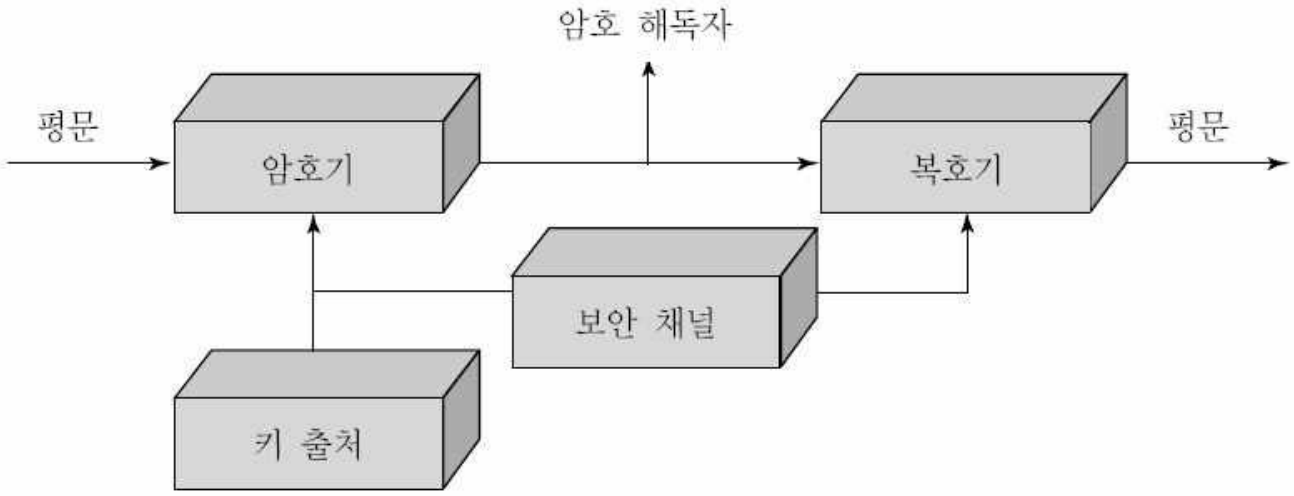
- 근대 암호
  - 기계식 암호
  - SP(Substitution & Permutation)
- 현대 암호
  - 1950년대 이후 컴퓨터를 이용한 암호 방법 개발
  - 수학적 접근 방식에 의해 보다 복잡하고 해독하기 어렵게 만들어짐
  - 구분
    - 대칭키 알고리즘
      - 블록(Block) 암호화
      - 스트림(Stream) 암호화
    - 비대칭키 알고리즘으로 구분

현대 암호 (2)

- 현대 암호 (계속)
  - 대부분의 데이터 암호화에는 속도가 빠른 대칭키 알고리즘 이용
  - 비대칭키 알고리즘의 경우 사용자 인증, 키 생성 등에 이용

대칭키 암호 (1)

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



대칭키 암호 (2)

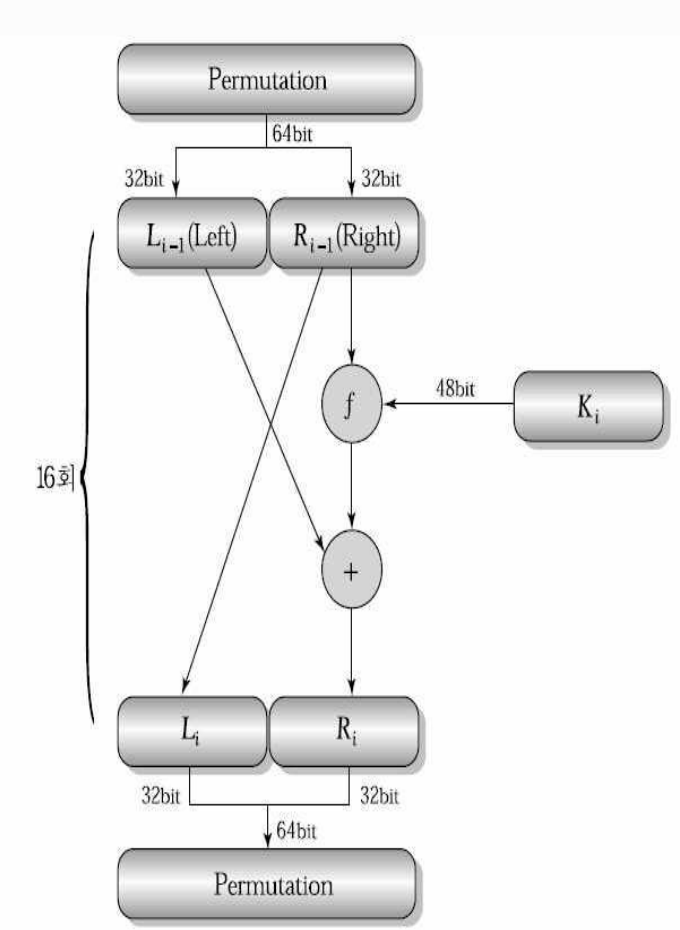
- 암호화 복호화를 수행하는 두 사용자가 동일한 키를 가지고 있어야 함
  - Pre-shared key
  - 온라인 상에서 구두 또는 메일, 전화로 교환 : 수동키
  - 블록 암호와 스트림 암호로 분류
  - 대표적 알고리즘 : DES, 3DES, SEED, RC2, RC5, AES(Rijndael)

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

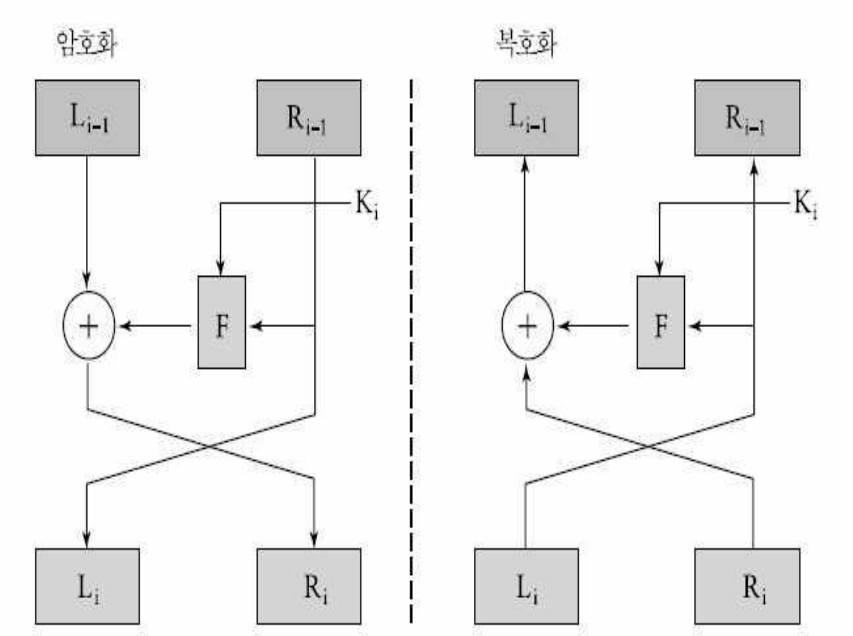
DES (1)

- Data Encryption Standard
  - IBM에서 기존의 LUCIFER를 변형하여 개발
  - 1977년 NIST에 의해 표준(FIPS PUB-46)으로 채택
  - 최초 128비트의 키로 64비트 블록상에서 실현
  - 하나의 칩으로 구현하기 위해 56비트 키로 줄임
  - 1994년 2차로 5년간 사용 연장



DES (2)

- Feistel 구조
  - 2t비트의 평문 블록을 각각 t비트로 나누고, L_{i-1}과 R_{i-1} 자리를 교환하는 과정을 라운드(i >= 1)라고 하고, 암호문(L_i, R_i)으로 변환되는 반복 구조
  - 평문 블록이 암호화 과정 중에 라운드 수만큼 반복 수행
  - 이 구조는 비교적 암호화 복호화 속도가 빠르고, H/W, S/W 구현이 용이하며, 아직까지는 구조상의 문제를 발견하지 못함



DES (3)

- 초기 치환 과정
  - 64비트 스트링으로된 평문 X가 IP(Initial Permutation)에 의해 초기 치환과정을 거침

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

DES (4)

- 역치환과정
  - 16번의 치환 과정을 거친 후 IP^{-1}을 적용하여 암호문을 얻음

IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

DES (5)

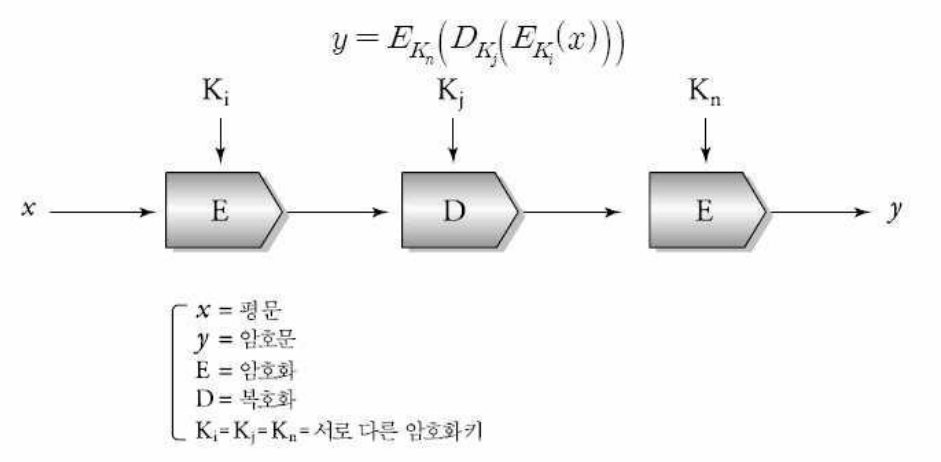
- 암호화 과정
  - 치환과정
  - F함수 과정
    - S-Box
    - P-Box
  - 키생성과정

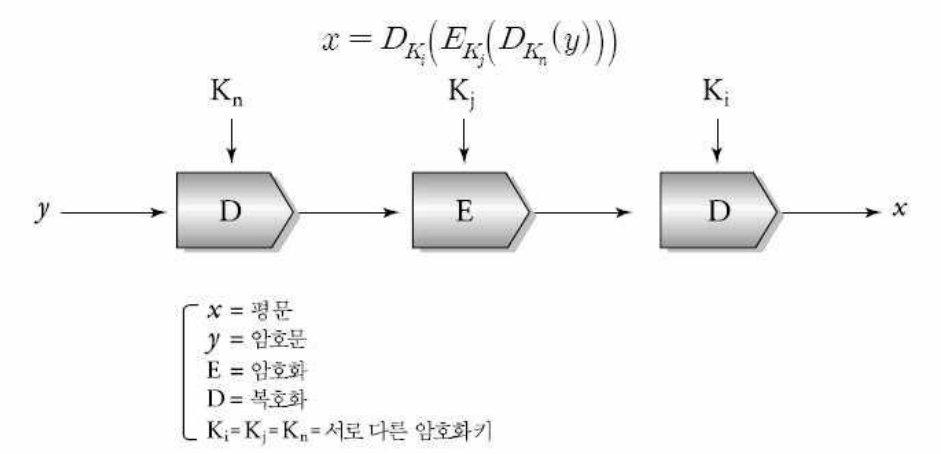
DES의 안전성

- 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

3DES

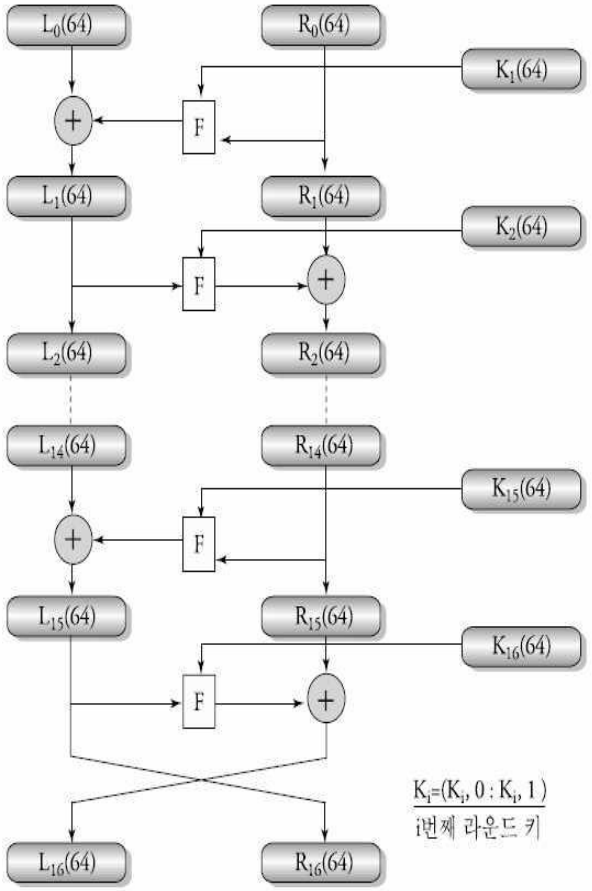
- Triple DES
  - DES의 암호화/복호화 수행과정을 3회 반복
  - 158비트의 암호화키 사용





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

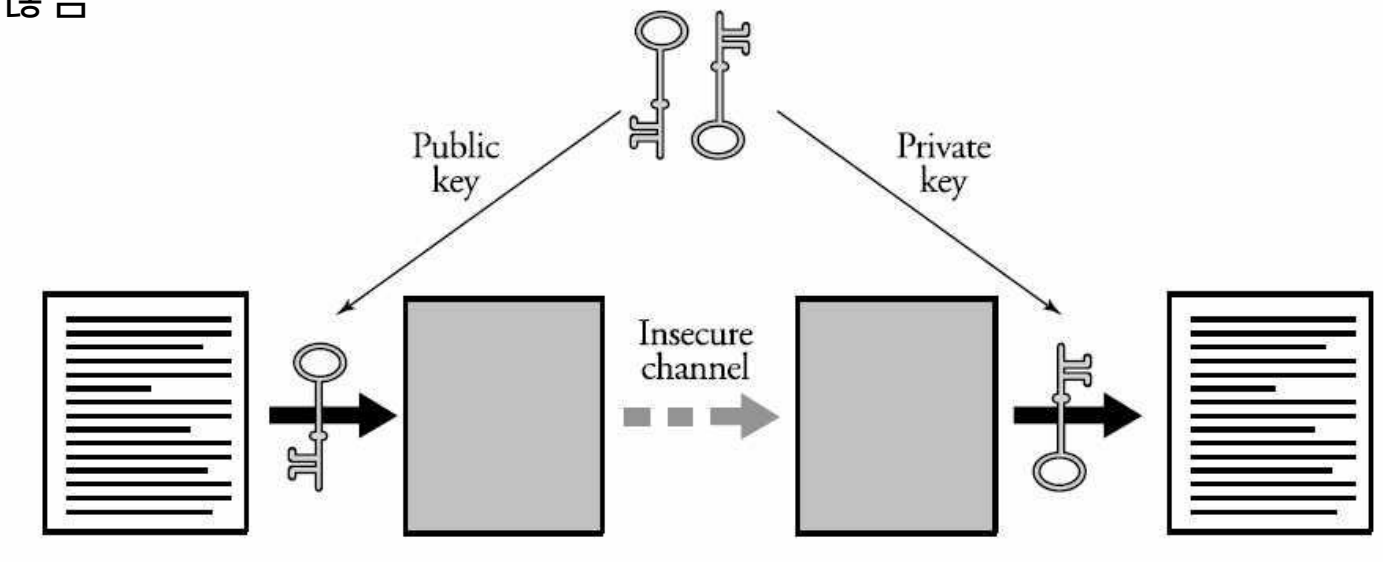
- 1998년 사용기한이 만료된 DES를 대체할 알고리즘으로 공모
- 벨기에에서 개발한 'Rijndael'이 선정되어 2000년 10월 표준으로 선정
- 특징
  - 가변 블록길이(128, 192, 256) 지원
  - 키도 128, 192, 256비트 사용
  - 키 길이에 의해 라운드 결정
  - Feistel 구조가 아닌 레이어(layer)로 구성
    - 선형 혼합(Linear mixing) : 라운드
    - 비선형(Non-linear) : S-Box
    - 키 추가(Key addition) : 라운드 키의 XOR

대칭키 암호 알고리즘 비교

	DES	SEED	AES
Country	America	Korea	Belgium
Structure	Feistel	Feistel	Layer
Size	64	128	Variable
Key Length	56(DES)/168(3DES)	128	128, 192, 256
Weak key	yes	no	no
Encrypt/Dcrypt	Block Algorithm	Block Algorithm	Block Algorithm
Function	Non-linear F	Non-linear F, G	Non-linear layer

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 (1)

- RSA(Rivest, Shamir, Adelman)
  - 1978년 MIT의 Rivest, Shamir, Adelman에 의해 제안
  - 인터넷 뱅킹과 같은 인증서를 통한 인증체제에서 주로 활용

RSA (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 (3)

- 암호화

암호화	
평문	$M < n$
암호문	$C = M^e \bmod n$

- 복호화

복호화	
암호문	C
평문	$M = C^d \bmod n$

RSA (4)

[예] 소수 $p = 17$, $q = 11$로 RSA 공개키 암호의 공개키와 개인키를 구하고, 암·복호화의 값을 알아보기로 한다.

$n = p \cdot q = 187$, $\phi(n) = (p-1)(q-1) = 160$ $\gcd(e, 2668) = 1$, $e = 7$로 선택,

$e \cdot d \equiv 1 \bmod 160$을 만족하는 $d = 23$

d 계산 과정

$$\begin{aligned} 161 &= 23 \times 7 \\ &= 10 \times 160 + 1 \\ &\equiv 1 \bmod 160 \end{aligned}$$

공개키 $PU = \{7, 187\}$와 개인키 $PR = \{23, 187\}$이 생성되며, $M = 88$일 경우에 암호화 값과 복호화 값을 구하면 다음과 같다.

RSA (5)

암호화 $C = 88^7 \bmod 187$

$88^7 \bmod 187 = [(88^4 \bmod 187) \times (88^2 \bmod 187) \times (88^1 \bmod 187)] \bmod 187$

$88^1 \bmod 187 = 88$

$88^2 \bmod 187 = 7744 \bmod 187 = 77$

$88^4 \bmod 187 = 59{,}969{,}536 \bmod 187 = 132$

$88^7 \bmod 187 = (88 \times 77 \times 132) \bmod 187 = 894{,}432 \bmod 187 = 11$

복호화 $M = 11^{23} \bmod 187$

$$11^{23} \bmod 187 = \left[\begin{array}{l} (11^1 \bmod 187) \times (11^2 \bmod 187) \times (11^4 \bmod 187) \\ \times (11^8 \bmod 187) \times (11^8 \bmod 187) \end{array} \right]$$

$11^1 \bmod 187 = 11$

$11^2 \bmod 187 = 121$

$11^4 \bmod 187 = 14{,}641 \bmod 187 = 55$

$11^{23} \bmod 187 = (11 \times 121 \times 55 \times 33 \times 33) \bmod 187 = 79{,}720{,}245 \bmod 187 = 88$

RSA (6)

[예1] $p=7$, $q=11$, $n=7\times11=77$, $\phi(n)=(7-1)(11-1)$, $e=13$이고, $d=37$일 때 평문=5라면, 암호화 값과 복호화 값을 알아본다.

Plaintext : 5

$$C=5^{13}=26 \bmod 77$$
$$=26$$

Ciphertext : 26

$$P=26^{37}=5 \bmod 77$$
$$=5$$

RSA (7)

- 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 공개키 알고리즘

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

알고리즘 비교 (1)

	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

알고리즘 비교 (2)

대칭키	공개키
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