

제 7 장

하이브리드 암호 시스템



박 종 혁 교수

Tel: 970-6702

Email: jhpark1@seoultech.ac.kr

- 1절 하이브리드 암호 시스템
- 2절 강한 하이브리드 암호 시스템이란
- 3절 암호 기술의 조합
- 4절 기타암호 - 동형암호

제1절 하이브리드 암호 시스템

1.1 대칭 암호와 공개 키 암호

1.2 하이브리드 암호 시스템

1.3 암호화

1.4 복호화

1.1 대칭 암호와 공개 키 암호

- 대칭키 암호방식과 공개키 암호방식 비교

항목	대칭키 암호화 방식	공개키 암호화 방식
키의 상호관계	암호화키 = 복호화키	암호화키 $\neq$ 복호화키
암호화 키	비밀	공개
복호화 키	비밀	비밀
암호알고리즘	비밀/공개	공개
대표적인 예	Vernam/DES	RSA
비밀 키 전송	필요	불필요
키 개수	$n(n-1)/2$	$2n$
안전한 인증	곤란	용이
암호화 속도	고속	저속
경제성	높다	낮다
전자서명	복잡	간단

1.1 대칭 암호와 공개 키 암호

- 대칭 암호

- 기밀성을 유지한 통신이 가능
- 키 배송 문제 해결이 필요

- 공개 키 암호

- 키 배송 문제를 해결할 수 있음

공개 키 암호의 2가지 큰 문제

- 1) 공개 키 암호는 대칭 암호에 비해 처리 속도가 훨씬 느리다
 - 2) 공개 키 암호는 중간자(man-in-the-middle) 공격에 약하다
- 하이브리드 암호 시스템을 이용하면 이 중 (1)의 문제를 해결

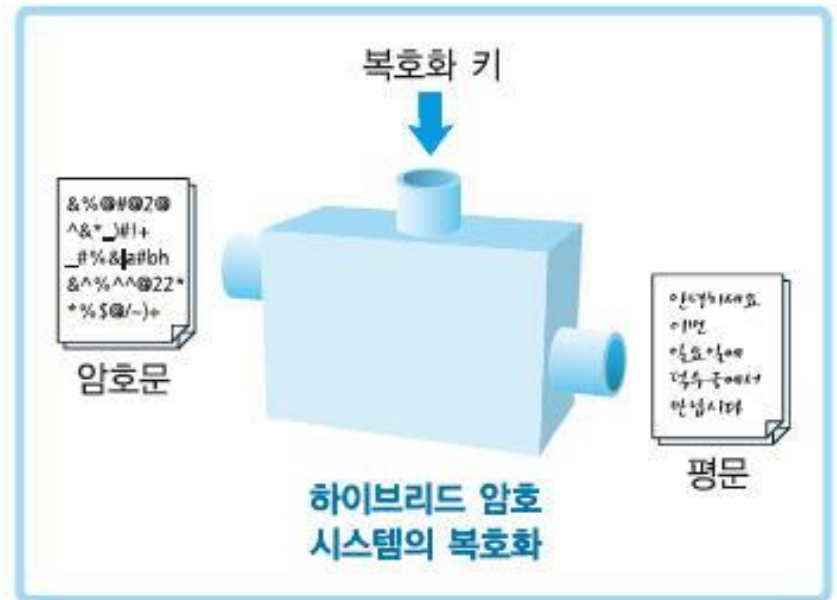
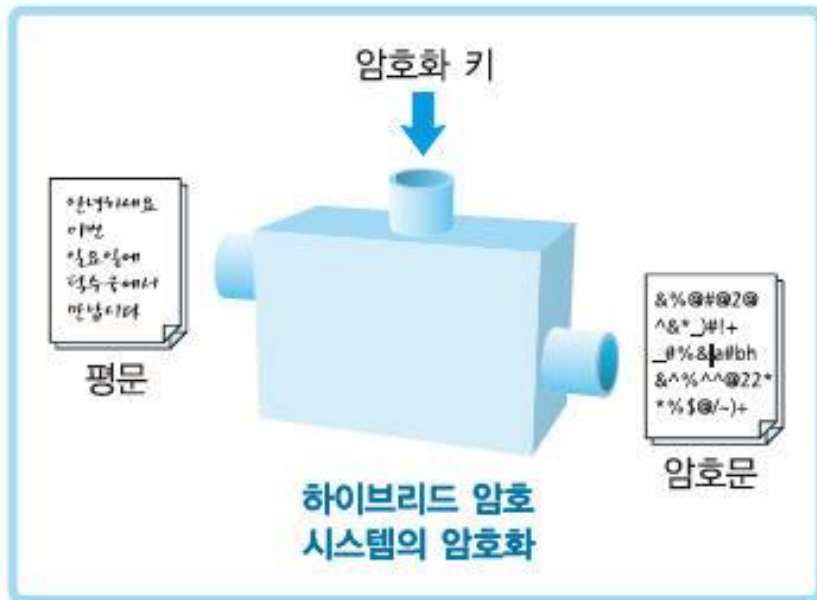
1.2 하이브리드 암호 시스템

- 하이브리드 암호 시스템(hybrid cryptosystem)
 - 대칭 암호와 공개 키 암호의 장점을 조합한 방법
 - 메시지의 기밀성: 고속의 대칭 암호
 - 대칭 암호 키의 기밀성: 공개 키 암호

하이브리드 암호 시스템의 구조

- 메시지는 대칭 암호로 암호화
- 대칭 암호의 암호화에서 사용한 세션 키는 의사난수 생성기로 생성
- 세션 키는 공개 키 암호로 암호화
- 공개 키 암호의 암호화에서 사용하는 키는 하이브리드 암호 시스템과 무관한 외부에서 만들어 사용

하이브리드 암호 시스템의 암호화와 복호화



하이브리드 암호 시스템의 암호화와 복호화

1.3 암호화

- 메시지 암호화
- 세션키 암호화
- 결합

- 평문 · 키 · 암호문

- P : 평문
- K_{pub} : 수신자의 공개 키
- C_2 : 공개 키 암호로 암호화된 세션 키
- C_1 : 대칭 암호로 암호화된 메시지
- $C=(C_1, C_2)$: 암호문

메시지 암호화

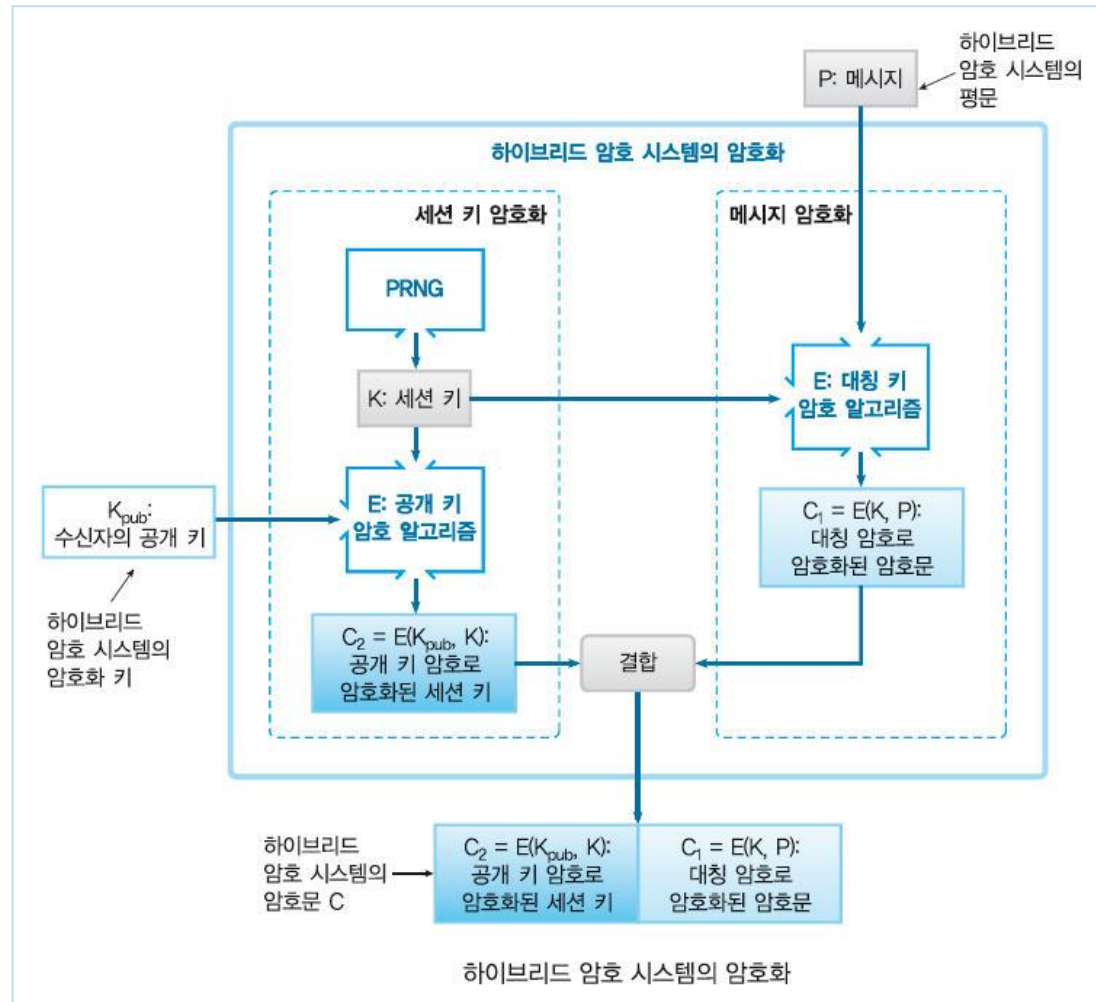
- $C_1 = E(K, P)$
- 대칭 암호를 이용해서 암호화
- 대칭 암호를 이용하면 고속으로 암호화

세션키 암호화

- $C_2 = E(K_{\text{pub}}, K)$
- 수신자의 공개 키로 암호화된다
- 세션키는 짧다
- 공개 키 암호가 아무리 느려도 세션 키 암호화에 그다지 시간이 걸리지 않음
- 세션 키는 대칭 암호에 있어서는 키이지만, 공개 키 암호의 입장에서 보면 하나의 평문

- 대칭 키(K)로 암호화된 암호문
($C_1 = E(K, P)$)
- 수신자의 공개 키(K_{pub})로 암호화된
세션 키($C_2 = E(K_{\text{pub}}, K)$)
- 암호문: $C = C_2 \parallel C_1 = E(K_{\text{pub}}, K) \parallel E(K, P)$

하이브리드 암호 시스템의 암호화



1.4 복호화

- 분할
- 세션 키 복호화
- 메시지 복호화

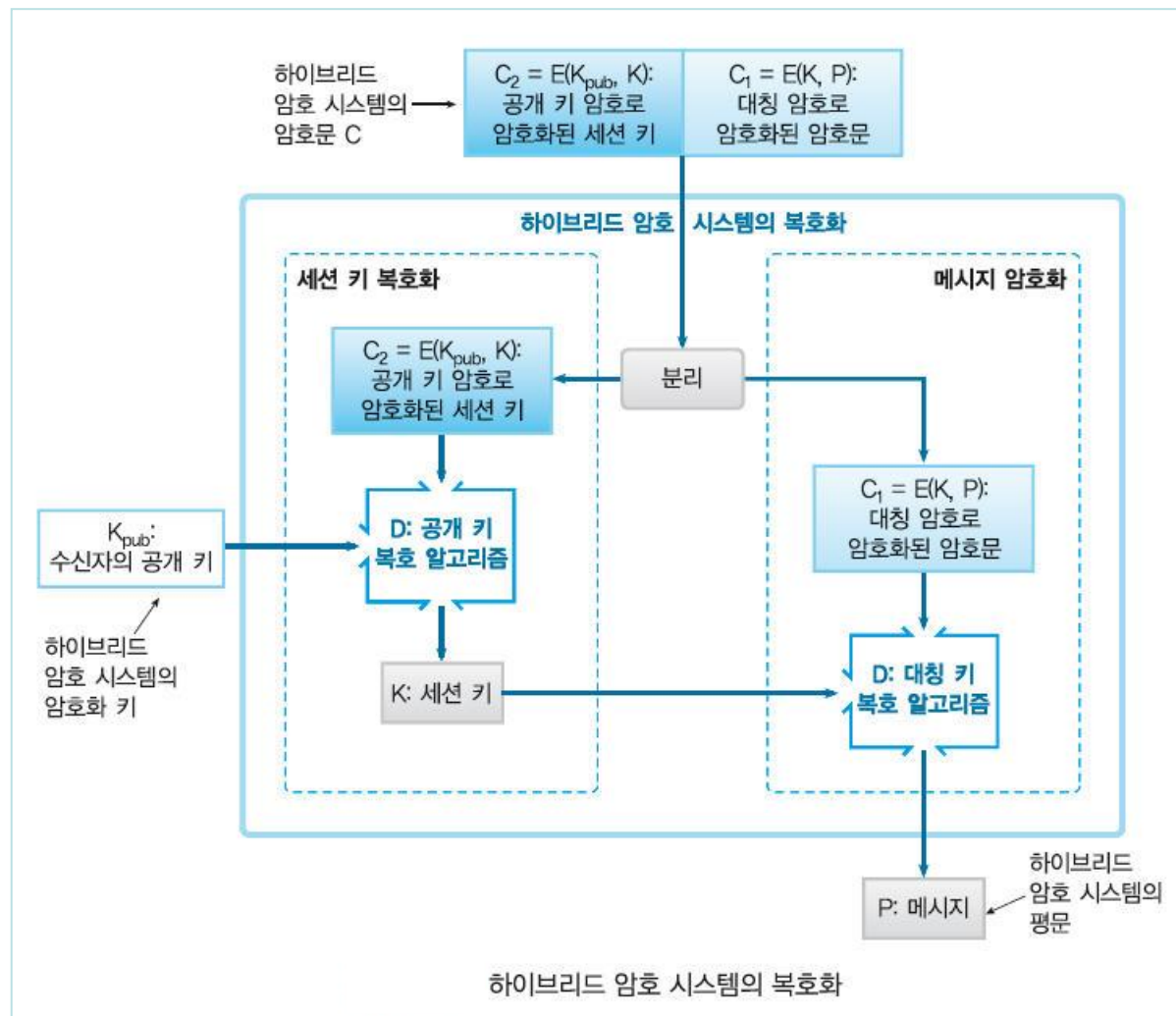
- 암호문: $C = C_2 \parallel C_1 = E(K_{\text{pub}}, K) \parallel E(K, P)$ 을 분할
 - $C_1 = E(K, P)$: 대칭 키(K)로 암호화된 암호문
 - $C_2 = E(K_{\text{pub}}, K)$: 수신자의 공개 키(K_{pub})로 암호화된 세션 키

세션 키 복호화

- $C_2 = E(K_{\text{pub}}, K)$ 복호화
- 수신자의 개인 키(K_{pri})가 필요
 - 개인 키를 가지고 있는 사람이 아니면 세션 키를 복호화 할 수 없음
- $K = D(K_{\text{pri}}, C_2)$: 수신자의 개인키로 복호화된 세션 키는 메시지 복호화 키로 이용

- $P = D(K, C_1)$

하이브리드 암호 시스템의 복호화



하이브리드 암호 시스템의 구체 예

- PGP
 - 하이브리드 암호 시스템
 - 디지털 서명이나 디지털 서명의 검증
 - 개인 키 관리
- SSL/TLS
 - 하이브리드 암호 시스템
 - Web의 암호 통신에서 사용

제2절 강한 하이브리드 암호 시스템이란

2.1 의사난수 생성기

2.2 대칭 암호

2.3 공개키 암호

2.4 키 길이의 밸런스

강한 하이브리드 암호 시스템이란

- 하이브리드 암호 시스템의 구성 요소
 - 의사난수 생성기
 - 대칭 암호
 - 공개 키 암호
- 각각의 기술 요소의 강도
- 강도의 밸런스

2.1 의사난수 생성기

- 세션 키 생성에 사용
- 품질이 나쁘면 만들어지는 세션 키를 공격자가 추측하게 될 위험성
- 세션 키 중 일부 비트라도 추측되지 않도록 주의

2.2 대칭암호

- 메시지 암호화에 사용
- 강한 대칭 암호 알고리즘을 사용
- 충분히 길이가 긴 키 사용
- 적절한 블록 암호 모드 사용

2.3 공개키 암호

- 세션 키 암호화에 사용
- 강한 공개 키 암호 알고리즘 사용
- 충분히 길이가 긴 키 사용

2.4 키 길이의 밸런스

- 어느 쪽인가 한 쪽의 키 길이가 극단적으로 짧으면, 공격이 그 쪽으로 집중될 가능성이 있음
- 대칭 암호와 공개 키 암호의 키 길이는 양쪽이 같은 정도의 강도가 되도록 길이의 균형을 맞추
- 장기간의 운용을 고려한다면 대칭 암호보다도 공개 키 암호 쪽을 강하게

제3절 암호 기술의 조합

하이브리드 암호 시스템

대칭 암호와 공개 키 암호를 조합해서 양쪽의 장점을 살리는 시스템을 구축

블록 암호 모드

고정 키 길이밖에 암호화할 수 없는 블록 암호를 조합해서 보다 긴 평문을 암호화

트리플 DES

DES를 3개 조합해서 DES보다도 긴 키 길이를 갖는 대칭 암호

암호 기술의 조합

- 디지털 서명
 - 일방향 해시 함수와 공개 키 암호를 조합
- 인증서
 - 공개 키와 디지털 서명을 조합
- 메시지 인증 코드
 - 일방향 해시 함수와 키를 조합
 - 대칭 암호로부터 생성
- 의사난수 생성기
 - 대칭 암호
 - 일방향 해시 함수
 - 공개 키 암호

기타 암호기술의 조합

- 전자 투표
- 디지털 캐시
- 블라인드 서명
 - 내용을 모르고 서명
- 영지식 증명
 - 상대방에게 정보를 건네지 않고 자신이 그 정보를 가지고 있다는 사실만을 증명해 보이는 방법

제4절 기타암호 - 동형암호

4.1 동형암호 개념

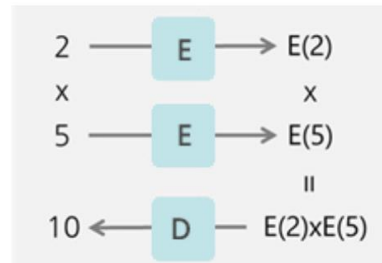
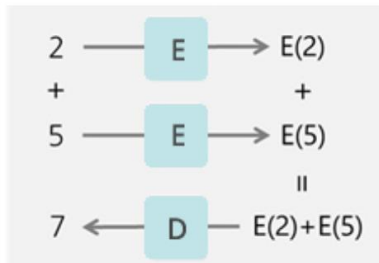
4.2 동형암호 응용

4.1 동형암호 개념

- 동형암호(Homomorphic Encryption)
 - 암호화된 상태에서도 덧셈, 곱셈 등을 보존하여 머신러닝 및 딥러닝이 가능한 공개키 암호기술

• 동형암호의 성질

- AES와 같은 암호 알고리즘에 암호문을 두 개 더하여 복호화할 때, 결과는 두 개 더한 값이 나오지 않음
- RSA나 ECC 등 동형암호는 곱셈 혹은 덧셈 두 가지 연산 중 하나를 보존해주는 Partial Homomorphic Encryption의 성질을 가짐



동형암호 성질

RSA 암호 알고리즘

e, n : 공개키 d : 비밀키
 (e, n, d 는 일련의 수학적 성질을 만족)

RSA 암호화

$$\frac{m}{\text{평문}} \cdot e \bmod n = \frac{c}{\text{암호문}}$$

RSA 복호화

$$c \cdot d \bmod n = m \cdot e \cdot d \bmod n = m$$

Homomorphic Property of RSA

$$\begin{aligned} & (C_1 C_2)^d \bmod n \stackrel{?}{=} m_1 m_2 \\ & (C_1 C_2)^d \bmod n \\ &= (m_1^e m_2^e)^d \bmod n \\ &= (m_1 m_2)^{ed} \bmod n \\ &= m_1 m_2 \end{aligned}$$

RSA 암호 알고리즘의 곱셈에 대한 Partial Homomorphic Property

- 동형암호(Fully Homomorphic Encryption)

- 동형암호로 데이터를 암호화하면 곱셈과 덧셈 연산 모두를 보존
- 즉, 동형암호로 두 개의 데이터 암호화하면, 두 암호문에 대해 곱셈 또는 덧셈하는 것이 원문에 해당 연산을 하는 것과 같은 효과를 가짐
- 머신러닝과 딥러닝의 연산은 곱셈과 덧셈으로 표현 가능하므로 동형암호를 이용할 경우 데이터가 암호화된 상태에서도 머신러닝, 딥러닝 데이터 분석 가능

4.2 동형암호 응용

- 장점

- 컴퓨터에서 데이터의 모든 계산은 AND, OR, NOT의 논리 게이트로 연산
- 암호화된 상태로 AND/OR/NOT연산 → 컴퓨터로 하는 모든 연산이 가능
 - 암호화 이후 검색/통계처리/Machine Learning
- 해커의 데이터 유출 원천봉쇄

- 단점

- 암호문 확장: 10-100 K배 → 0.1-1 K배(대칭키방식)
- 암호복호화 속도: 수십 ms (AES 1us, RSA 1ms)
- 암호문 연산: 곱셈 수백ms
- 응용연산 종류에 따른 속도의 차이가 큼 → 개별적 최적화

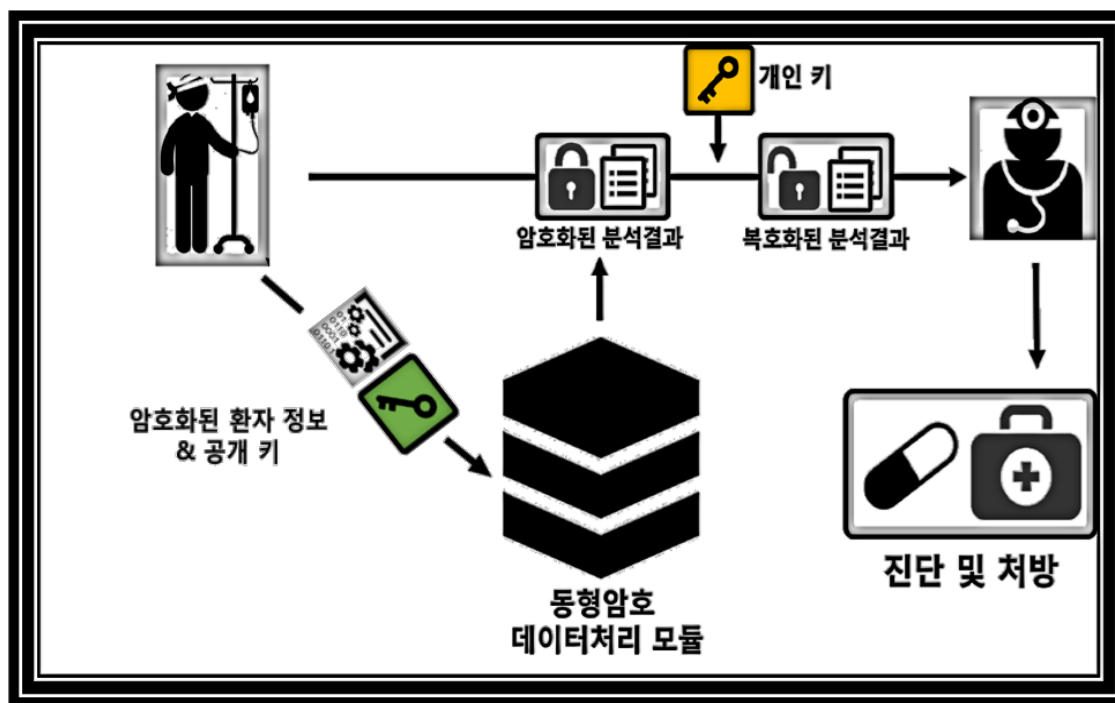
예제 2. 암호화된 헬스케어



건강진단



생체정보
(심박/활동량)



하이브리드 암호 시스템

연습문제 풀이

1. 다음 중 이 장에서 설명한 하이브리드 암호를 만드는 데 사용되는 기술의 장점을 설명한 것으로 적합한 것은?

- ① 대칭 암호의 빠른 처리 속도와 공개 키 암호의 키 배송의 편리성
- ② 대칭 암호의 빠른 처리 속도와 공개 키 암호의 기밀성
- ③ 공개 키 암호의 빠른 처리 속도와 대칭 암호의 키 배송의 편리성
- ④ 공개 키 암호의 빠른 처리 속도와 대칭 암호의 기밀성
- ⑤ 공개 키 암호의 키 배송의 편리성과 대칭 암호의 무결성

2. 공개 키 암호의 단점으로 적합한 것은?

- ① 느린 처리 속도와 기밀성에 대한 취약성
- ② 무결성의 취약성과 재전송 공격에 대한 취약성
- ③ 부인방지에 대한 취약성과 개인 키 관리의 취약성
- ④ 느린 처리 속도와 중간자 공격에 대한 취약성
- ⑤ 키 배송의 불편함과 인증에 대한 취약성

3. 이 장에서 설명한 하이브리드 암호에 대한 설명으로 적합하지 않은 것은?

- ① 세션 키는 의사난수 생성기로 만든다.
- ② 세션 키를 암호화 하는 데 시간이 걸리는 단점이 있다.
- ③ 세션 키 암호화에 공개 키 암호를 사용한다.
- ④ 무결성을 보장하지는 않는다.
- ⑤ 메시지를 암호화할 때 대칭 암호를 사용한다.

4. 이 장에서 설명한 하이브리드 암호에서 사용하는 공개 키 암호와 대칭 암호의 키 길이 밸런스가 중요한 이유는?

- ① 세션 키의 길이는 공개 키의 길이와 같아야 처리 속도가 빨라지기 때문이다.
- ② 공개 키가 노출되어 있어서 세션 키의 길이도 노출되기 때문이다.
- ③ 공격자가 사용된 키 중 취약한 암호 시스템의 키를 공격할 가능성이 있기 때문이다.
- ④ 난수 생성기에 서 생성되는 세션 키의 안전성을 보장하기 때문이다.
- ⑤ 공개 키 암호에 대한 다양한 공격이 가능하기 때문이다.

5. 암호 기술을 조합하게 되면 여러 가지 장점을 갖는 하이브리드 암호 시스템을 만들 수 있다. 이런 조합의 방법에 속하지 않는 것은?

- ① 메시지 인증 코드
- ② 트리플 DES
- ③ 디지털 서명
- ④ 인증서
- ⑤ 시저 암호

6. 이 장에서 설명한 하이브리드 암호를 이용하여 암호화된 암호문을 수신한 수신자가 복호화를 하기 위해 제일 먼저 해야 하는 일은 무엇인가?

- ① 공개 키로 암호화된 세션 키와 세션 키로 암호화된 메시지를 분리한다.
- ② 의사난수 생성기를 이용하여 의사난수를 생성한다.
- ③ 수신자의 개인 키로 세션 키를 복호화 한다.
- ④ 세션 키로 암호문을 복호화 한다.
- ⑤ 따로 전송된 세션 키와 암호화된 메시지를 결합한다.

7. 하이브리드 암호 시스템에서는 세션 키를 만들어내는 데 _____을(를) 사용한다. 만약 _____의 품질이 나쁘면 만들어진 세션 키를 공격자가 추측하게 될 위험성이 있다.

- ① 의사난수 생성기
- ② 암호 알고리즘
- ③ 초기화 벡터
- ④ 복호 알고리즘
- ⑤ 난수 생성기

Q & A

Thanks!