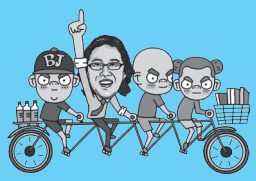
수험생의 마음으로 만든 책! 시나공 시리즈

2020 시나공

시험에 나오는 것만 공부한다!



2020년 2회 정보처리기사 실기 시험 100% 합격전략집

9장 소프트웨어 개발 보안 구축

2020년 2회 정보처리기사 실기 대비용 핵심요약

[핵심126] 보안 요소

- 소프트웨어 개발에 있어 충족시켜야할 요소 및 요건을 의미한다.
- 주요 보안 요소에는 기밀성(Confidentiality), 무결성(Integrity), 가용성(Availability)이 있으며, 그 외에도 인증(Authentication), 부인 방지(NonRepudiation) 등이 있다.

기밀성	시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용된다.
무결성	시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있다.
가용성	인가받은 사용자는 언제라도 사용할 수 있다.
인증	시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위를 말한다.
부인 방지	데이터를 송 · 수신한 자가 송 · 수신 사실을 부인할 수 없도록 송 · 수신 증거를 제공한다.

2020년 1, 2회 기사 필기

1. 다음은 소프트웨어 개발에 있어 충족시켜야 할 보안 요소에 대한 설명이다. 괄호(①~③)에 들어갈 알맞은 보안 요소를 쓰시오.

(①)	인가받은 사용자는 언제라도 사용할 수 있다.
인증	시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위를 말한다.
(②)	시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있다.
부인 방지	데이터를 송 · 수신한 자가 송 · 수신 사실을 부인할 수 없도록 송 · 수신 증거를 제공한다.
(③)	시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용된다.

답
①:
②:
③:

정답 1. ① 가용성 ② 무결성 ③ 기밀성

[핵심127] 보안 점검 – 세션 통제

- 세션은 서버와 클라이언트의 연결을 의미하고, 세션 통제는 세션의 연결과 연결로 인해 발생하는 정보를 관리하는 것을 의미한다.
- 세션 통제는 소프트웨어 개발 과정 중 요구사항 분석 및 설계 단계에서 진단해야 하는 보안 점검 내용이다.
- 세션 통제의 보안 약점

불충분한 세션 관리	• 일정한 규칙이 존재하는 세션ID가 발급되거나 타임아웃이 너무 길게 설정되어 있는 경우 발생할 수 있는 보안 약점이다. • 세션ID(SessionID) : 서버가 클라이언트들을 구분하기 위해 부여하는 키(Key)로, 클라이언트가 서버에 요청을 보낼 때마다 세션ID를 통해 인증이 수행됨
잘못된 세션에 의한 정보 노출	• 다중 스레드(Multi-Thread) 환경에서 멤버 변수에 정보를 저장할 때 발생하는 보안 약점이다. • 멤버 변수보다 지역 변수를 활용하여 변수의 범위를 제한함으로써 방지할 수 있다.

1. 다음은 세션 통제의 보안 약점에 대한 설명이다. 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

다중 스레드(Multi-Thread) 환경에서 ( )에 정보를 저장할 때 정보 노출을 방지하려면 ( )보다 지역 변수를 활용하여 변수의 범위를 제한해야 한다. ( )는 객체와 연결된 변수로, 클래스 내에 선언되어 클래스의 모든 메소드들이 접근 가능한 변수이다.

답 :

정답 1. 멤버 변수

[핵심128] 보안 점검 – 입력 데이터 검증 및 표현

- 입력 데이터 검증 및 표현은 입력 데이터로 인해 발생하는 문제들을 예방하기 위해 구현 단계에서 검증해야 하는 보안 점검 항목들이다.
- 입력 데이터 검증 및 표현의 보안 약점

SQL 삽입	• 입력란에 SQL을 삽입하여 무단으로 DB를 조회하거나 조작하는 보안 약점이다. • 동적 쿼리에 사용되는 입력 데이터에 예약어 및 특수문자가 입력되지 않게 필터링 되도록 설정하여 방지할 수 있다.
경로 조작 및 자원 삽입	• 데이터 입출력 경로를 조작하여 서버 자원을 수정 · 삭제할 수 있는 보안 약점이다. • 사용자 입력값을 식별자로 사용하는 경우, 경로 순회 공격을 막는 필터를 사용하여 방지할 수 있다.
크로스사이트 스크립팅(XSS)	• 웹페이지에 악의적인 스크립트를 삽입하여 방문자들의 정보를 탈취하거나, 비정상적인 기능 수행을 유발하는 보안 약점이다. • HTML 태그의 사용을 제한하거나 스크립트에 삽입되지 않도록 '<', '>', '&' 등의 문자를 다른 문자로 치환함으로써 방지할 수 있다.
운영체제 명령어 삽입	• 외부 입력값을 통해 시스템 명령어의 실행을 유도함으로써 권한을 탈취하거나 시스템 장애를 유발하는 보안 약점이다. • 웹 인터페이스를 통해 시스템 명령어가 전달되지 않도록 하고, 외부 입력값을 검증 없이 내부 명령어로 사용하지 않음으로써 방지할 수 있다.
위험한 형식 파일 업로드	• 악의적인 명령어가 포함된 스크립트 파일을 업로드함으로써 시스템에 손상을 주거나, 시스템을 제어할 수 있는 보안 약점이다. • 업로드 되는 파일의 확장자 제한, 파일명의 암호화, 웹사이트와 파일 서버의 경로 분리, 실행 속성을 제거하는 등의 방법으로 방지할 수 있다.
신뢰되지 않는 URL 주소로 자동접속 연결	• 입력 값으로 사이트 주소를 받는 경우 이를 조작하여 방문자를 피싱 사이트로 유도하는 보안 약점이다. • 연결되는 외부 사이트의 주소를 화이트 리스트로 관리함으로써 방지할 수 있다.

1. 다음은 입력 데이터로 인해 발생할 수 있는 보안 약점에 대한 설명이다. 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

> 입력란에 SQL을 삽입하여 무단으로 DB를 조회하거나 조작할 경우 발생할 수 있는 문제를 방지하려면 ( )에 사용되는 입력 데이터에 예약어 및 특수문자가 입력되지 않게 필터링 되도록 설정하면 된다. ( )는 질의어 코드를 문자열 변수에 넣어 조건에 따라 질의를 동적으로 변경하여 처리하는 방식이다.

답 :

2. 다음은 입력 데이터로 인해 발생할 수 있는 문제를 예방하기 위한 방법에 대한 설명이다. 괄호에 들어갈 가장 적합한 용어를 쓰시오.

> 입력 값으로 사이트 주소를 받는 경우 이를 조작하여 방문자를 피싱 사이트로 유도할 수 있다. 이를 방지하려면 연결되는 외부 사이트의 주소를 ( )로 관리해야 한다.

답 :

정답 1. 동적 쿼리 2. 화이트 리스트

[핵심129] 보안 점검 – 보안 기능

- 소프트웨어 개발의 구현 단계에서 코딩하는 기능인 인증, 접근제어, 기밀성, 암호화 등을 올바르게 구현하기 위한 보안 점검 항목들이다.
- 보안 기능에 대한 점검을 수행하지 않을 경우 발생할 수 있는 보안 약점

적절한 인증 없이 중요기능 허용	• 보안검사를 우회하여 인증과정 없이 중요한 정보 또는 기능에 접근 및 변경이 가능하다. • 중요정보나 기능을 수행하는 페이지에서는 재인증 기능을 수행하도록 하여 방지할 수 있다.
부적절한 인가	• 접근제어 기능이 없는 실행경로를 통해 정보 또는 권한을 탈취할 수 있다. • 모든 실행경로에 대해 접근제어 검사를 수행하고, 사용자에게는 반드시 필요한 접근 권한만을 부여하여 방지할 수 있다.

중요한 자원에 대한 잘못된 권한 설정	• 권한 설정이 잘못된 자원에 접근하여 해당 자원을 임의로 사용할 수 있다. • 소프트웨어 관리자만 자원들을 읽고 쓸 수 있도록 설정하고, 인가되지 않은 사용자의 중요 자원에 대한 접근 여부를 검사함으로써 방지할 수 있다.
취약한 암호화 알고리즘 사용	• 암호화된 환경설정 파일을 해독하여 비밀번호 등의 중요정보를 탈취할 수 있다. • 안전한 암호화 알고리즘을 이용하고, 업무 관련 내용이나 개인정보 등에 대해서는 IT보안인증사무국이 안정성을 확인한 암호모듈을 이용함으로써 방지할 수 있다.
중요 정보 평문 저장 및 전송	• 암호화되지 않은 평문 데이터를 탈취하여 중요한 정보를 획득할 수 있다. • 중요한 정보를 저장하거나 전송할 때는 반드시 암호화 과정을 거치도록 하고, HTTPS 또는 SSL과 같은 보안 채널을 이용함으로써 방지할 수 있다.
하드코드된 비밀번호	• 소스코드 유출 시 내부에 하드코드된 패스워드를 이용하여 관리자 권한을 탈취할 수 있다. • 패스워드는 암호화하여 별도의 파일에 저장하고, 디폴트 패스워드나 디폴드 키의 사용을 피함으로써 방지할 수 있다.

1. 다음은 보안 기능에 대한 점검을 수행하지 않을 경우 발생할 수 있는 보안 약점에 대한 설명이다. 괄호(①~②)에 들어갈 알맞은 용어를 쓰시오.

> 개인정보나 인증정보와 같은 중요한 정보를 저장하거나 전송할 때는 반드시 암호화 과정을 거치도록 하고, (①) 또는 (②)과 같은 보안 채널을 이용함으로써 방지할 수 있다.
> (①)는 웹브라우저와 서버 간의 안전한 통신을 위해 암호통신규약을 결합한 것이다. (②)는 데이터를 송 · 수신하는 두 컴퓨터 사이에 위치하여 인증, 암호화, 무결성을 보장하는 업계 표준 프로토콜이다.

답
① :
② :

정답 1. ① HTTPS(Hypertext Transfer Protocol Secure)
② SSL(Secure Sockets Layer)

[핵심130] 보안 점검 – 시간 및 상태

- 동시 수행을 지원하는 병렬 처리 시스템이나 다수의 프로세스가 동작하는 환경에서 시간과 실행 상태를 관리하여 시스템이 원활하게 동작되도록 하기 위한 보안 검증 항목들이다.
- 시간 및 상태를 점검하지 않은 코딩이 유발하는 보안 약점

TOCTOU 경쟁 조건	• 검사 시점(Time Of Check)과 사용 시점(Time Of Use)을 고려하지 않고 코딩하는 경우 발생하는 보안 약점이다. • 코드 내에 동기화 구문을 사용하여 해당 자원에는 한 번에 하나의 프로세스만 접근 가능하도록 구성함으로써 방지할 수 있다.
종료되지 않는 반복문 또는 재귀함수	• 반복문이나 재귀함수에서 종료 조건을 정의하지 않았거나 논리 구조상 종료될 수 없는 경우 발생하는 보안 약점이다. • 모든 반복문이나 재귀함수의 수행 횟수를 제한하는 설정을 추가하거나, 종료 조건을 점검하여 반복 또는 호출의 종료 여부를 확인함으로써 방지할 수 있다.

1. 다음은 시간 및 상태를 점검하지 않은 코딩이 유발하는 보안 약점에 대한 설명이다. 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

> 검사 시점(Time Of Check)과 사용 시점(Time Of Use)을 고려하지 않고 코딩하는 경우 프로세스가 가진 자원 정보와 실제 자원 상태가 일치하지 않는 오류가 발생할 수 있다. 이를 방지하려면 코드 내에 ( ) 구문을 사용하여 해당 자원에는 한 번에 하나의 프로세스만 접근 가능하도록 구성해야 한다.
> ( ) 구문은 공유 자원에 대해 둘 이상의 프로세스가 접근하는 것을 막는 구문이다.

답 :

정답 1. 동기화

[핵심131] 보안 점검 – 에러처리

- 소프트웨어 실행 중 발생할 수 있는 오류(Error)들을 사전에 정의하여 오류로 인해 발생할 수 있는 문제들을 예방하기 위한 보안 점검 항목들이다.
- 에러처리의 미비로 인한 코딩이 유발하는 보안 약점

오류 메시지를 통한 정보노출	• 오류 발생으로 실행 환경, 사용자 정보, 디버깅 정보 등의 중요 정보를 소프트웨어가 메시지로 외부에 노출하는 보안 약점이다. • 예외처리 구문에 예외의 이름이나 스택 트레이스를 출력하도록 코딩한 경우 해커는 소프트웨어의 내부구조를 쉽게 파악할 수 있다. • 오류 발생 시 가능한 한 내부에서만 처리되도록 하거나 메시지를 출력할 경우 최소한의 정보 또는 사전에 준비된 메시지만 출력되도록 함으로써 방지할 수 있다.
오류 상황 대응 부재	• 소프트웨어 개발 중 예외처리를 하지 않았거나 미비로 인해 발생하는 보안 약점이다. • 오류가 발생할 수 있는 부분에 예외처리 구문을 작성하고, 제어문을 활용하여 오류가 악용되지 않도록 코딩함으로써 방지할 수 있다.
부적절한 예외처리	• 함수의 반환값 또는 오류들을 세분화하여 처리하지 않고 광범위하게 묶어 한 번에 처리하거나, 누락된 예외가 존재할 때 발생하는 보안 약점이다. • 모든 함수의 반환값이 의도대로 출력되는지 확인하고, 세분화된 예외처리를 수행함으로써 방지할 수 있다.

1. 다음은 에러처리의 미비로 인한 코딩이 유발하는 보안 약점에 대한 설명이다. 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

( )는 오류가 발생한 위치를 추적하기 위해 소프트웨어가 실행 중에 호출한 메소드의 리스트를 기록한 것으로 예외처리 구문에 ( )를 출력하도록 코딩한 경우 해커는 소프트웨어의 내부구조를 쉽게 파악할 수 있다. 이를 방지하려면 오류 발생 시 가능한 한 내부에서만 처리되도록 하거나 메시지를 출력할 경우 최소한의 정보 또는 사전에 준비된 메시지만 출력되도록 해야 한다.

답 :

정답 1. 스택 트레이스(Stack Trace)

[핵심132] 보안 점검 – 코드 오류

- 소프트웨어 구현 단계에서 개발자들이 코딩 중 실수하기 쉬운 형(Type) 변환, 자원 반환 등의 오류를 예방하기 위한 보안 점검 항목들이다.
- 코드 오류로 발생할 수 있는 보안 약점

널 포인터(Null Pointer) 역참조	• 널 포인터가 가리키는 메모리에 어떠한 값을 저장할 때 발생하는 보안 약점이다. • 널 포인터(Null Pointer) : 널(Null)은 값이 없음을 의미하며, 포인터(Pointer)는 메모리의 위치를 가리키는 요소이다. 널 포인터(Null Pointer)는 포인터에 널(Null)이 저장되어 어떠한 곳도 가리키지 못하는 상태의 요소를 말한다. • 널(Null)이 될 수 있는 포인터를 이용하기 전에 널 값을 갖고 있는지 검사함으로써 방지할 수 있다.
부적절한 자원 해제	• 자원을 반환하는 코드를 누락하거나 프로그램 오류로 할당된 자원을 반환하지 못했을 때 발생하는 보안 약점이다. • 프로그램 내에 자원 반환 코드가 누락되었는지 확인하고, 오류로 인해 함수가 중간에 종료되었을 때 예외처리에 관계없이 자원이 반환되도록 코딩함으로써 방지할 수 있다.
해제된 자원 사용	• 이미 사용이 종료되어 반환된 메모리를 참조하는 경우 발생하는 보안 약점이다. • 반환된 메모리에 접근할 수 없도록 주소를 저장하고 있는 포인터를 초기화함으로써 방지할 수 있다.
초기화되지 않은 변수 사용	• 변수 선언 후 값이 부여되지 않은 변수를 사용할 때 발생하는 보안 약점이다. • 변수 선언 시 할당된 메모리를 초기화함으로써 방지할 수 있다.

1. 다음은 코드 오류로 발생할 수 있는 보안 약점에 대한 설명이다. 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

( )는 포인터에 널(Null)이 저장되어 어떠한 곳도 가리키지 못하는 상태의 요소를 말하는 것으로 ( )가 가리키는 메모리에 어떠한 값을 저장할 경우 ( ) 역참조가 발생한다.

답 :

정답 1. 널 포인터(Null Pointer)

[핵심 133] 보안 점검 – 캡슐화

- 정보 은닉이 필요한 중요한 데이터와 기능을 불충분하게 캡슐화하거나 잘못 사용함으로써 발생할 수 있는 문제를 예방하기 위한 보안 점검 항목들이다.
- 캡슐화로 인해 발생할 수 있는 보안 약점

잘못된 세션에 의한 정보 노출	• 다중 스레드(Multi-Thread) 환경에서 멤버 변수에 정보를 저장할 때 발생하는 보안 약점이다. • 멤버 변수보다 지역 변수를 활용하여 변수의 범위를 제한함으로써 방지할 수 있다.
제거되지 않고 남은 디버그 코드	• 개발 중에 버그 수정이나 결과값 확인을 위해 남겨둔 코드들로 인해 발생하는 보안 약점이다. • 소프트웨어를 배포하기 전에 코드 검사를 통해 남아있는 디버그 코드를 삭제함으로써 방지할 수 있다.
시스템 데이터 정보 노출	• 시스템의 내부 정보를 시스템 메시지 등을 통해 외부로 출력하도록 코딩했을 때 발생하는 보안 약점이다. • 시스템 메시지를 통해 노출되는 메시지는 최소한의 정보만을 제공함으로써 방지할 수 있다.
Public 메소드로부터 반환된 Private 배열	• 선언된 클래스 내에서만 접근이 가능한 Private 배열을 모든 클래스에서 접근이 가능한 Public 메소드에서 반환할 때 발생하는 보안 약점이다. • Private 배열을 별도의 메소드를 통해 조작하거나, 동일한 형태의 복제본으로 반환받은 후 값을 전달하는 방식으로 방지할 수 있다.
Private 배열에 Public 데이터 할당	• Private 배열에 Public으로 선언된 데이터 또는 메소드의 파라미터를 저장할 때 발생하는 보안 약점이다. • Public으로 선언된 데이터를 Private 배열에 저장할 때, 레퍼런스가 아닌 값을 직접 저장함으로써 방지할 수 있다.

잠깐만요 ❗ Java의 접근 제어자(제한자)

Java에서 특정 개체를 선언할 때 외부로부터의 접근을 제한하기 위해 사용되는 예약어로 Public, Private, Protected, Default가 있습니다.

2020년 1, 2회 기사 필기

1. Java에서 특정 개체를 선언할 때 외부로부터의 접근을 제한하기 위해 사용되는 접근 제한자의 종류 4가지를 쓰시오.

답 :

정답 1. Public, Private, Protected, Default

[핵심 134] 보안 점검 – API 오용

- 소프트웨어 구현 단계에서 API를 잘못 사용하거나 보안에 취약한 API를 사용하지 않도록 하기 위한 보안 검증 항목들이다.
- API 오용으로 발생할 수 있는 보안 약점

DNS Lookup에 의존한 보안 결정	• 도메인명에 의존하여 인증이나 접근 통제 등의 보안 결정을 내리는 경우 발생하는 보안 약점이다. • DNS 검색을 통해 도메인 이름을 비교하지 않고 IP 주소를 직접 입력하여 접근함으로써 방지할 수 있다.
취약한 API 사용	• 보안 문제로 사용이 금지된 API를 사용하거나, 잘못된 방식으로 API를 사용했을 때 발생하는 보안 약점이다. • 보안 문제로 금지된 대표적인 API에는 C언어의 문자열 함수 strcat( ), strcpy( ), sprintf( ) 등이 있다. • 보안 문제로 금지된 함수는 안전한 함수로 대체하고, API의 매뉴얼을 참고하여 보안이 보장되는 인터페이스를 사용함으로써 방지할 수 있다.

1. 다음은 API 오용으로 발생할 수 있는 보안 약점에 대한 설명이다. 괄호에 들어갈 가장 적합한 용어를 쓰시오.

> 도메인명에 의존하여 인증이나 접근 통제 등의 보안 결정을 내리는 경우 공격자는 ( )을 악용하여 인증이나 접근 통제를 우회하는 수법으로 권한을 탈취한다. 이를 방지하려면 IP 주소를 직접 입력하여 접근해야 한다.

답 :

정답 1. DNS Lookup

[핵심135] 암호 알고리즘

- 패스워드, 주민번호, 은행계좌와 같은 중요정보를 보호하기 위해 평문을 암호화된 문장으로 만드는 절차 또는 방법을 의미한다.
- 암호 알고리즘은 해시(Hash)를 사용하는 단방향 암호화 방식과, 개인키 및 공개키로 분류되는 양방향 암호화 방식이 있다.

개인키 암호화(Private Key Encryption) 기법	• 동일한 키로 데이터를 암호화하고 복호화한다. • 개인키 암호화 기법은 대칭 암호 기법 또는 단일키 암호화 기법이라고도 한다.
공개키 암호화(Public Key Encryption) 기법	• 데이터를 암호화할 때 사용하는 공개키(Public Key)는 데이터베이스 사용자에게 공개하고, 복호화할 때의 비밀키(Secret Key)는 관리자가 비밀리에 관리한다. • 공개키 암호화 기법은 비대칭 암호 기법이라고도 함
해시(Hash)	• 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키로 변환하는 것을 의미한다. • 해시 함수의 종류 : SHA 시리즈, MD5, SNEFRU 등 – SHA 시리즈 : 1993년에 미국 NSA가 제작하고 미국 국립표준기술연구소(NIST)에서 표준으로 채택한 암호화 알고리즘으로 가장 많이 사용됨 – MD5 : 1991년 R.rivest가 MD4를 개선한 암호화 알고리즘으로 각각의 512 비트 짜리 입력 메시지 블록에 대해 차례로 동작함 – SNEFRU : 1990년에 R.C. Merkle에 의해 제안된 128, 254비트 암호화 알고리즘

2020년 1회 기사 실기

1. 다음 설명에 가장 적합한 알고리즘을 쓰시오.

> 1991년 R.rivest가 MD4를 개선한 암호화 알고리즘으로 각각의 512 비트짜리 입력 메시지 블록에 대해 차례로 동작한다. 각 512 비트 입력 메시지 블록을 처리하고 나면 128 비트 스테이트(State)의 값이 변하게 된다.

답 :

정답 1. MD5

[핵심136] 개인키 암호화 방식과 공개키 암호화 방식에서 사용되는 주요 암호화 알고리즘

SEED	• 1999년 한국인터넷진흥원(KISA)에서 개발한 블록 암호화 알고리즘 • 블록 크기는 128비트이며, 키 길이에 따라 128, 256으로 분류
ARIA(Academy, Research Institute, Agency)	• 2004년 국가정보원과 산학연협회가 개발한 블록 암호화 알고리즘 • 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류
DES(Data Encryption Standard)	• 1975년 미국 NBS에서 발표한 개인키 암호화 알고리즘 • 블록 크기는 64비트이며, 키 길이는 56비트
AES(Advanced Encryption Standard)	• 2001년 미국 표준 기술 연구소(NIST)에서 발표한 개인키 암호화 알고리즘 • 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류
RSA(Rivest Shamir Adleman)	• 1978년 MIT의 라이베스트(Rivest), 샤미르(Shamir), 애들먼(Adelman)에 의해 제안된 공개키 암호화 알고리즘 • 소인수 분해 문제를 이용한 공개키 암호화 기법에 널리 사용되는 암호화 알고리즘

2020년 1, 2회 기사 필기

1. 소인수 분해 문제를 이용한 공개키 암호화 기법에 널리 사용되는 암호화 알고리즘이 무엇인지 쓰시오.

답 :

2. 다음 설명에 가장 적합한 알고리즘을 쓰시오.

> 2001년 미국 표준 기술 연구소(NIST)에서 발표한 개인키 암호화 알고리즘으로 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류된다.

답 :

정답 1. RSA(Rivest Shamir Adleman)
2. AES(Advanced Encryption Standard)