정보보호론 #4

공개키 암호시스템의 이해

Prof. Byung Il Kwak

Review

- 대칭키 암호화 방식
- Data Encryption Standard (DES)
- Advanced Encryption Standard (AES) 소개

Contents

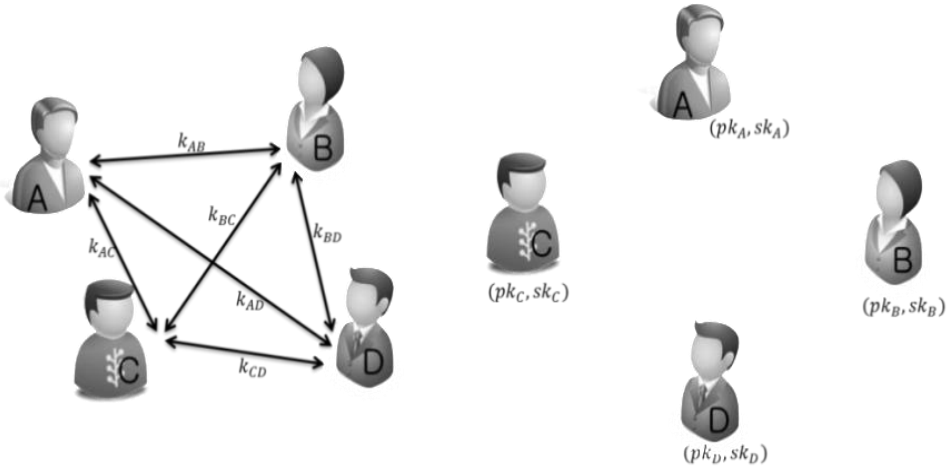
- 공개키 암호화 시스템
- 공개 키 기반 구조 및 활용

CONTENTS

- 공개키 암호화 시스템

공개키 암호 개요

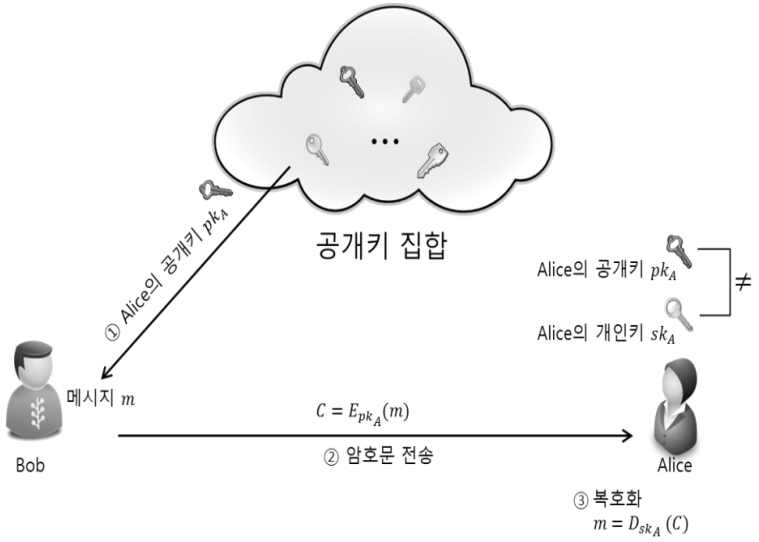
- 대칭키 암호
  - 안전한 채널을 통해서 사용자가 서로 동일한 키를 사전 공유
  - n명이 서로 비밀통신을 하기 위해서는 $(n(n-1))/2$개의 키가 필요
  - 송신자나 수신자의 부인방지를 제공하지 못함.



대칭키 암호시스템 공개키 암호시스템

공개키 암호 개요

- 공개키 (or 비대칭키(Asymmetric) 암호시스템)
  - Diffie와 Hellman은 1976년, 논문 “New Directions in Cryptography” 에서 공개키 암호시스템 소개
  - 각 사람마다 한 쌍의 키(공개키 pk, 개인키 sk)
    - 공개키는 모두에게 공개되고, 개인키는 비밀로 보관
    - 공개키 pk로부터 개인키 sk를 도출하는 것은 계산적으로 불가능 (Computationally Infeasible)



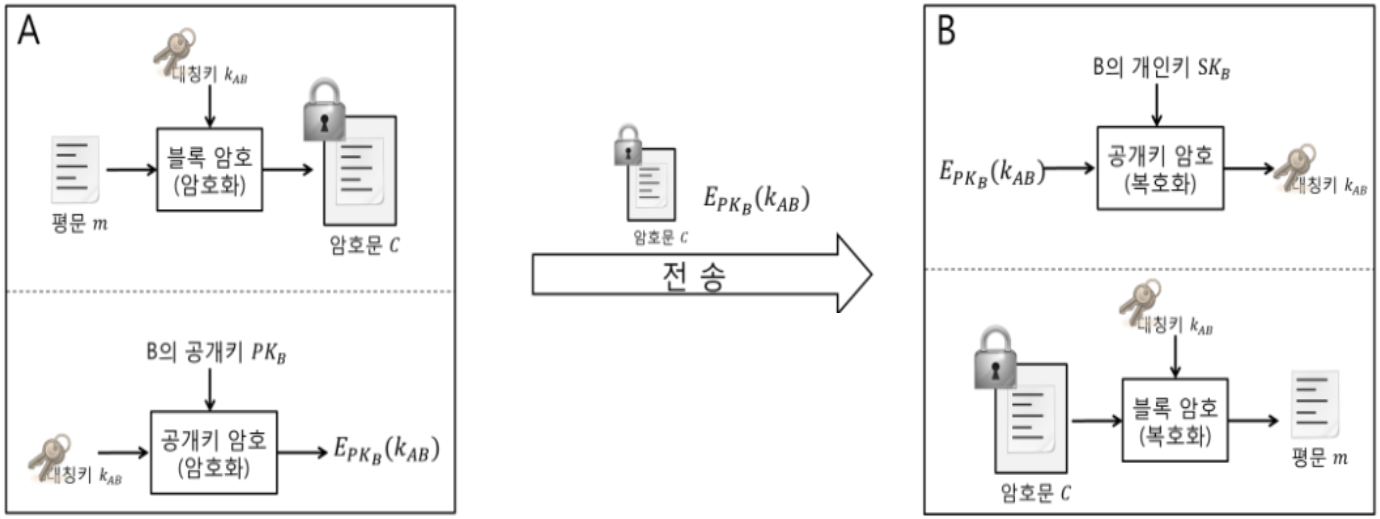
공개키 암호 개요

- 공개키 및 대칭키 암호시스템의 차이점

	대칭키 암호시스템	공개키 암호시스템
비밀키 분배	필요	불필요
보유 비밀키 개수 (n명이 비밀통신 하는 경우)	$(n-1)$개 (상대방별로 키가 필요)	1개 (자신의 비밀키만 보유)
암호화 & 복호화 속도	빠름	느림
대표 예	DES, AES, SEED, ARIA	RSA, ElGamal

공개키 암호 개요

- 하이브리드 암호시스템
  - 대용량의 데이터를 암호화하기 위해서 대칭키 암호 시스템에서 사용되는 비밀키 k를 공개키 암호시스템으로 암호화(E_{pk} (비밀키 k))하여 분배하고, 수신자는 분배된 비밀키를 이용하여 대용량의 데이터를 대칭키 암호시스템으로 암호화

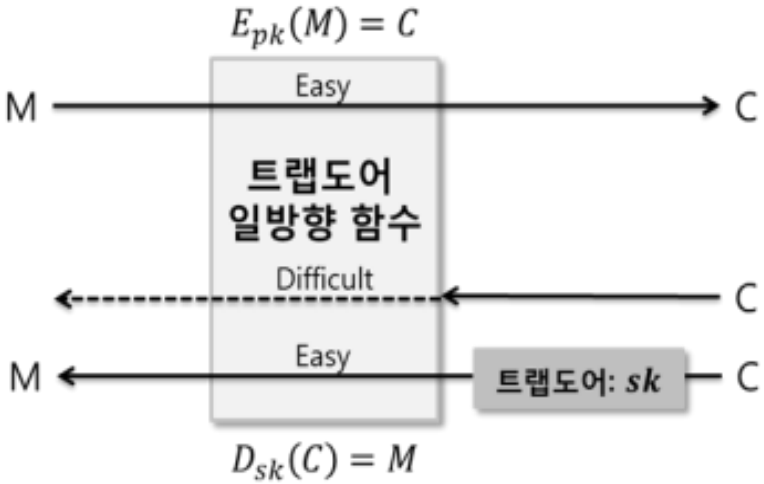


공개키 암호 개요 - 기본 개념

- 소인수분해 문제
  - When n is large, **n = p × q** is a one-way function.
  - Given **p** and **q** , it is always easy to calculate **n**; given **n**, it is **very difficult** to compute **p** and **q**.
  - 최근까지 알려진 결과로는 2009년에 232자리의 십진수를 수 백대의 컴퓨터를 사용하여 2년만에 인수분해에 성공
    - 232자리 십진수는 이진수로 나타내면 768 비트가 필요하며 위의 결과는 768비트 RSA의 경우 동일한 계산능력으로 2년만에 평문이 복호화 될 수 있음을 의미

공개키 암호 개요 - 기본 개념

- Trapdoor One-Way Function (TOWF)
  1. f is easy to compute.
  2. f^{-1} is difficult to compute.
     - 메시지 m에 대하여 $c = f(m)$의 계산은 용이하고, c만 주어진 경우 $m = f^{-1}(c)$를 구하는 것은 어렵지만 어떤 정보 k가 주어지면 쉽게 역함수를 계산하여 m을 얻을 수 있다면, **함수 f를 트랩도어 일방향 함수(Trapdoor One-way Function)**라고 하며 **k를 트랩도어 정보(Trapdoor Information)**
  - 공개키 설계의 기본 개념



공개키 암호 개요 - 기본 개념

- Trapdoor One-Way Function (TOWF)
  - 공개키 암호시스템은 트랩도어 일방향 함수를 기반으로 설계
    - 공개키로 생성된 암호문을 공격자가 복호화하는 작업은 트랩도어 정보 없이 일방향 함수의 역함수를 계산해야 하는 것이므로 어려운 문제임
      - 암호문 c를 획득한 공격자가 c에서 m을 알아내는 것은 계산적으로 불가능
      - $c = f(m) = E_{pk}(m)$이며 $E_{pk}(\cdot)$가 바로 앞에서 정의한 트랩도어 일방향 함수가 됨
      - 수신자는 자신의 개인키 sk를 사용하여 암호문 c에서 메시지 m을 쉽게 얻어낼 수 있다. 즉, $m = f^{-1}(c) = D_{sk}(E_{pk}(m))$을 계산할 수 있게 됨
      - 따라서, 개인키 sk가 트랩도어 정보가 됨

공개키 암호 알고리즘

- RSA
- RABIN
- ElGamal
- …

공개키 암호 알고리즘

- RSA 암호
  - 가장 많이 사용되고 있는 공개키 암호시스템
    - **R**ivest, **S**hamir, and **A**dleman 의 이름에서 RSA
  - RSA 암호 알고리즘의 안전성은 아주 큰 소수로 이루어진 합성수(현재 컴퓨터의 계산 능력을 고려하면 2048비트)를 인수분해 하는 것이 어려움을 기반함
    - 합성수의 소인수분해를 효율적으로 할 수 없게끔 각 변수를 생성 해야함

공개키 암호 알고리즘

- RSA 알고리즘
  - 키 생성
    - 1. 서로 다른 두 소수 p와 q 선택 (크기가 동일한 1024비트 이상의 수로 선택);
    - 2. $n = p \times q$값 계산, p와 q를 곱하여 n 계산
    - 3. $\varphi(n) = (p-1)(q-1)$ 계산
    - 4. $1 < e < \varphi(n) - 1$의 범위에서 $\gcd(e, \varphi(n)) = 1$인 e, 즉 $\varphi(n)$과 서로소인 e 선택
    - 5. $d \equiv e^{-1} \ mod \ \varphi(n)$
      - (e, n) : public-key
      - (d) : private-key

공개키 암호 알고리즘

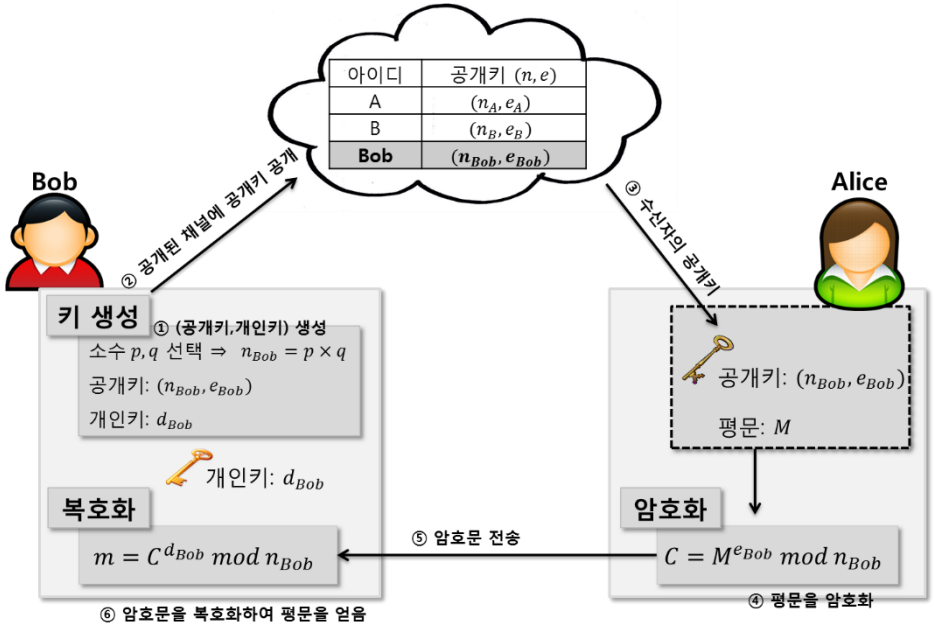
- RSA 알고리즘
  - 키 생성
    - 개인키를 알고 있는 사용자는 (d, n) 값을 이용해 복호화 연산을 쉽게 수행할 수 있음
    - 하지만, 개인키를 모르는 공격자는 d 값을 알아내기 위해 $d \equiv e^{-1} \ mod \ \varphi(n)$ 을 풀어내야 함
    - 즉, $\varphi(n)$ 값을 구해내기 위해 큰 정수값 n 을 소인수분해해야 함 (n 값이 크게 되면 수학적으로 구해내기 어려움)
      - 공격자가 공개키 n을 통해 p와 q를 찾는 것은 매우 어려움. 또한, n을 인수분해 하지 않고 $\varphi(n)$을 구하는 것 역시 매우 어려움.

공개키 암호 알고리즘

- RSA 알고리즘
  - 키 생성
    - 일반적으로 RSA 암호 알고리즘의 경우, 두 소수 p와 q는 1024비트 이상의 수로 선택하며, e와 d 값은 보통 2048비트보다 크게 됨
      - 공개키 e는 작은 수가 선택되기도 하지만 d는 항상 큰수가 됨
      - 즉, 작은 공개키를 사용하더라도, 개인키 d는 값이 작아지지 않으므로 RSA 암호 알고리즘의 경우 안전하다고 할 수 있음

공개키 암호 알고리즘

- RSA 알고리즘
  - 암 · 복호화
    - Encryption : $c = m^e \bmod n$
      - Note $m < n$ (for uniqueness)
    - Decryption : $m = c^d \bmod n$



공개키 암호 알고리즘

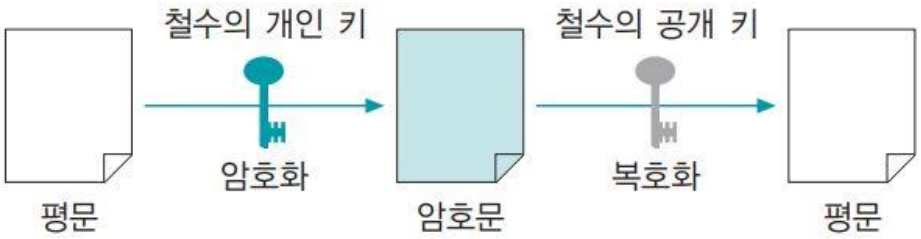
- RSA 알고리즘
  - 암 · 복호화
    - Encryption : $c = m^e \ mod \ n$
      - Note $m < n$ (for uniqueness)
    - Decryption : $m = c^d \ mod \ n$

- 예) 개인키 [d = 3, n = 33], 공개키 [e = 7, n = 33] 일 때,
  - 평문 m = 9
    - 암호화
      - $c = 9^7 \ mod \ 33 =$ **15**
    - 복호화
      - $\mathrm{m} = 15^3 \ mod \ 33 =$ **9**

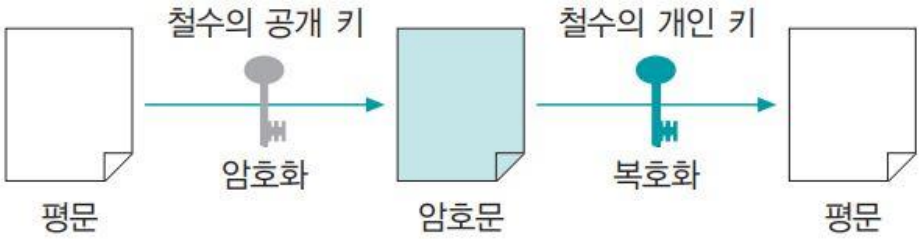
공개키 암호 알고리즘

- 공개키 암호화 알고리즘 방식
  - 대칭 암호화 알고리즘과 달리 메시지의 암호화와 복호화가 같은 키로 이루어지지 않음
  - 언제나 한 쌍의 개인 키와 공개 키로 암호화와 복호화가 이루어짐

(1)



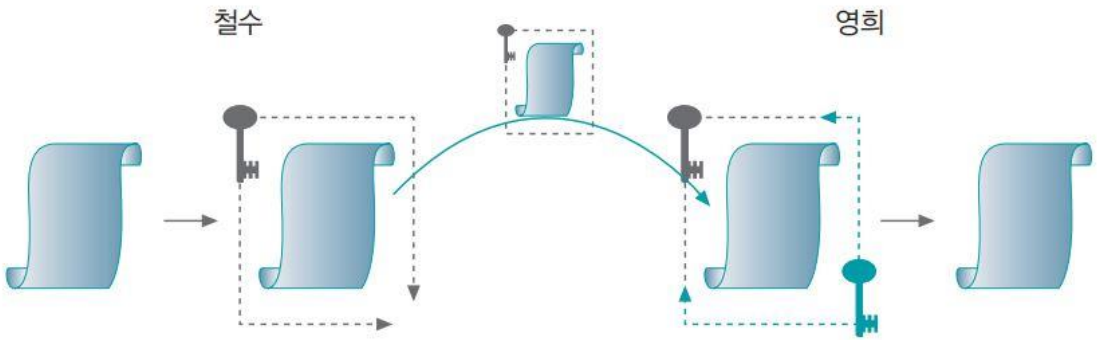
(2)



언제 이렇게 사용할까?

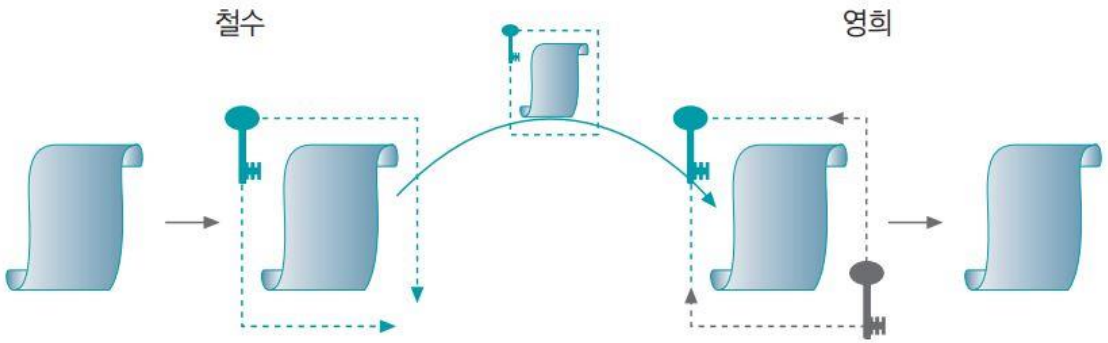
공개키 암호 알고리즘

- 공개키 암호화 알고리즘의 방식
  - 비대칭 암호화 알고리즘은 대칭 암호화 알고리즘보다 더 엄밀한 기밀성을 제공
    - 철수는 공개 키를 구함
    - 편지(평문)를 공개키를 이용하여 암호화한 후 영희에게 전송
    - 영희는 개인 키로 철수의 편지(암호문)를 복호화 후 내용 확인
    - 민수가 중간에서 편지(암호문)를 가로채더라도 공개 키로 암호화한 편지를 민수의 개인 키로는 복호화 불가
    - Q) 편지(평문)을 암호화한 공개키는 누구의 공개키?



공개키 암호 알고리즘

- 공개키 암호화 알고리즘의 방식
  - 비대칭 암호화 알고리즘은 부인방지 기능을 제공
    - 철수의 개인 키로 암호화된 편지는 철수의 공개 키로만 열 수 있음
    - 영희는 수신한 편지가 철수의 공개 키로 풀려야만 철수가 보낸 편지라고 확신할 수 있음
    - Q) 편지(평문)을 암호화한 공개키는 누구의 공개키?



CONTENTS

- 공개 키 기반 구조 및 활용

공개 키 기반 구조 및 활용

- 전자 상거래
  - 컴퓨터 등을 이용해 인터넷과 같은 네트워크 상에서 이뤄지는 전자적 매체(시스템)을 이용한 가상 공간에서의 제품/용역을 사고파는 거래 행위
    - 광고, 마케팅, 고객 지원, 배송, 지불 등과 같은 활동

공개 키 기반 구조 및 활용

- 전자 상거래의 보안 요건
  - 신분 확인 수단 제공
    - 원격의 거래 상대를 신뢰할 수 없기 때문에 네트워크에서 상대방이나 자신에 대한 신분 확인 수단이 필요
  - 제삼자의 중재
    - 거래 사실(거래 내역)을 공증할 수 있는 신뢰할 만한 제삼자의 중재 필요
  - 지불 방식의 안전성
    - 전자지불 방식(과정)의 안전성 보장 방법이 확보되어야 함
  - 블록체인을 활용하는 비트코인과 같은 거래 체계가 활성화된다면 전자 상거래의 세 가지 보안 요건 중 '제삼자의 중재'는 앞으로 완전히 사라질 수 있음

공개 키 기반 구조 및 활용

- 공개 키 기반 구조의 개념
  - 공개 키 기반 구조 (PKI)
    - 메시지의 암호화 및 전자 서명을 제공하는 복합적인 보안 시스템 환경
    - 공개 키 기반 구조는 '인터넷에서 신분증 검증'의 역할
    - 가장 가까운 관청인 주민센터가 있고 그 위에 구청, 시청이 있으며 맨 위에 정부가 있는 것과 마찬가지
    - 공개 키 기반 구조에 속하는 사람은 어디서든지 자신의 인터넷상 신분을 인증 기관(CA)에서 공인인증서로 증명 가능

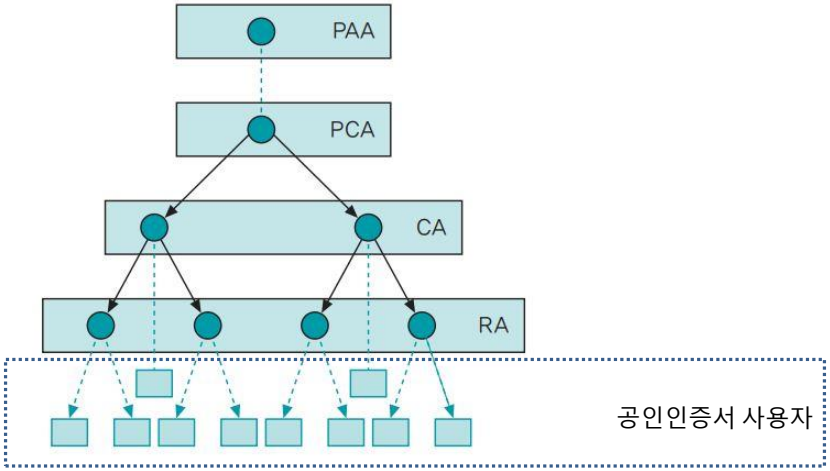
공개 키 기반 구조 및 활용

공개 키 기반 구조의 개념

트리형 공개 키 기반 구조

- 공개 키 기반 구조가 되려면 인증 정보를 일원화하여 호환성을 갖춤으로써 개인이 쉽게 접근할 수 있어야 함
- 순수 계층 구조: 트리형으로 구성된 공개 키 기반 구조

트리형 공개 키 기반 구조



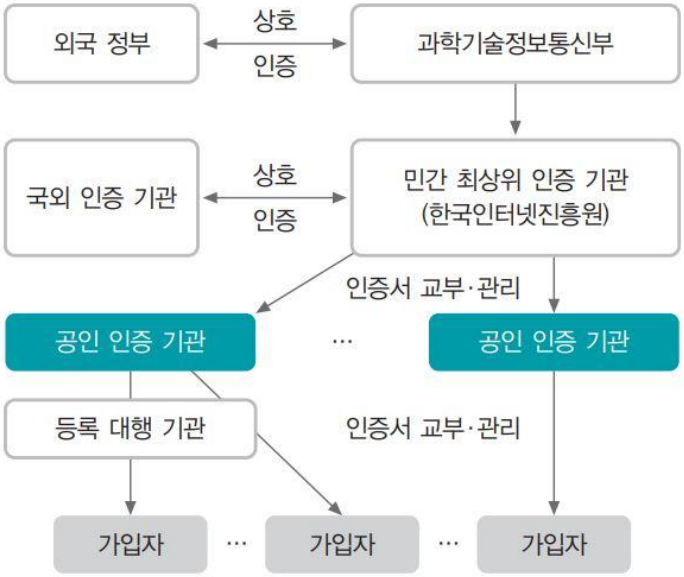
- PAA: 정책 승인기관으로 공인인증서 정책 결정/하위기관 정책을 승인
- PCA: 정책 인증기관으로 Root CA 인증서 발급/기본 정책 수립
- CA: PCA의 하위기관인 인증기관으로 인증서 발급/취소 등의 업무 담당
- RA: 등록 기관으로 공인인증서 인증 요청을 확인하고 CA 간 인터페이스 제공

공개 키 기반 구조 및 활용

공개 키 기반 구조의 개념

상호 인증으로 연결된 공개 키 기반 구조

- 인증 기관이 상호 인증을 통해 연결되어 있는 모델도 존재
- 두 인증 기관이 상대방의 공개 키를 서로 인증하는 인증서, 즉 상호 인증서를 발급하여 사용
- 일반적으로 공개 키 기반 구조는 트리형과 네트워크 구조가 혼합



국가와 기관이 상호 인증으로 연결된 공개 키 기반 구조

공개 키 기반 구조 및 활용

- 공인 인증서
  - 공인인증서의 개념
    - 공개 키와 공개 키의 소유자를 연결해주는 전자문서.
    - 오늘날 사용하는 대부분의 공인인증서는 X.509 인증서(버전 3)를 표준으로 따름
    - SPK I인증서, PGP 인증서가 있음
  - 공인인증서의 특성
    - 누구나 사용자의 공인 인증서와 공개 키를 획득할 수 있음
    - 인증 기관 이외에는 공인 인증서를 수정 및 발급할 수 없음
    - 같은 인증 구조 내의 사용자 간에는 상호 인증의 신뢰 가능

공개 키 기반 구조 및 활용

공인 인증서

공인인증서의 구성

① 버전: 공인 인증서의 형식을 구분
(우리가 사용하는 대부분의 공인 인증서는 버전 3)

② 일련 번호: 공인 인증서를 발급한 인증 기관 내의 인증서 일련번

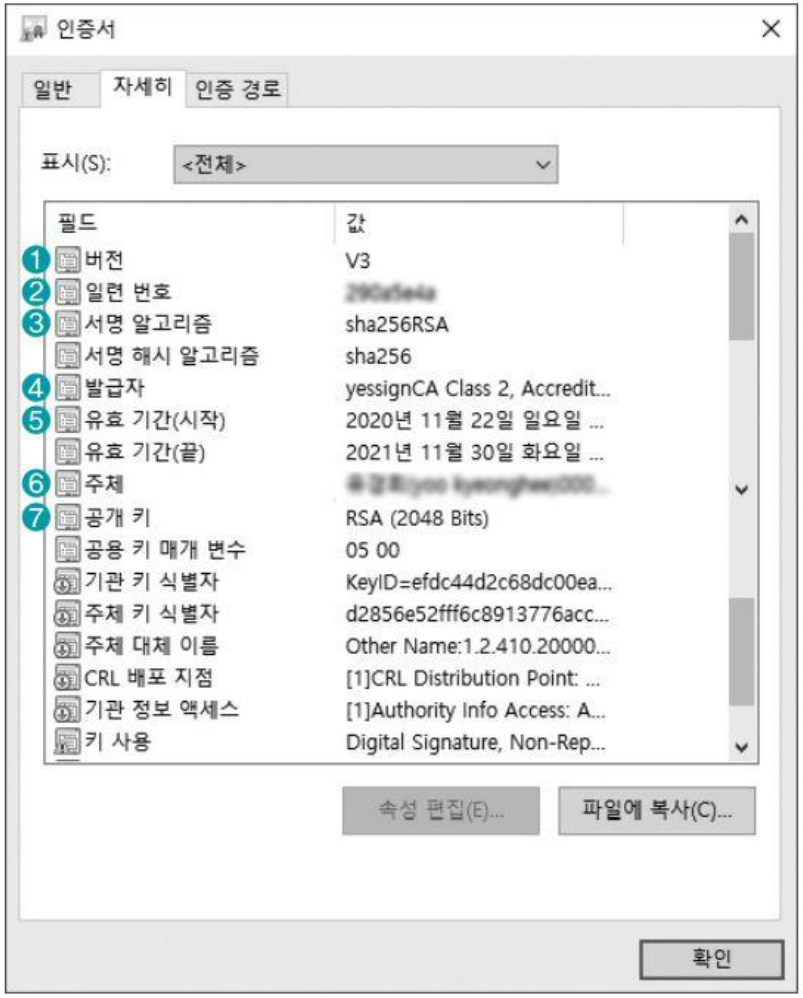
③ 서명 알고리즘: 공인 인증서를 발급할 때 사용한 알고리즘

④ 발급자: 공인 인증서를 발급한 인증 기관의 DN, DN은
X.500 표준에 따라 명명된 이름

⑤ 유효 기간: 공인 인증서를 사용할 수 있는 시작일과 만료일로
초 단위까지 표기

⑥ 주체: 공인 인증서 소유자의 DN

⑦ 공개 키: 공인 인증서의 모든 영역을 해시하여 인증 기관의
개인 키로 서명한 값



공개 키 기반 구조 및 활용

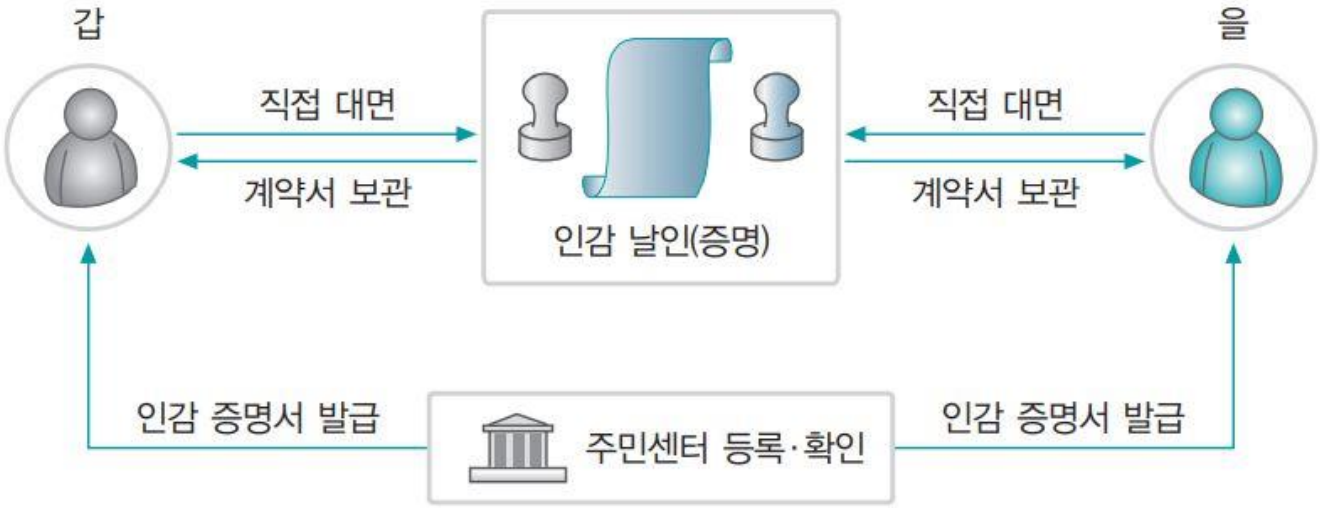
- 공인 인증서
  - 공인인증서의 폐기
    - 공인인증서를 시기 적절하게 폐기하여 피해를 줄이는 것이 폐기의 목적.
    - 인증 기관에서 인증서 폐기 목록(CRL)을 주기적으로 발급
    - 이 폐기 목록도 인증 기관이 전자서명을 하여 발급함

공개 키 기반 구조 및 활용

전자 서명과 전자 봉투

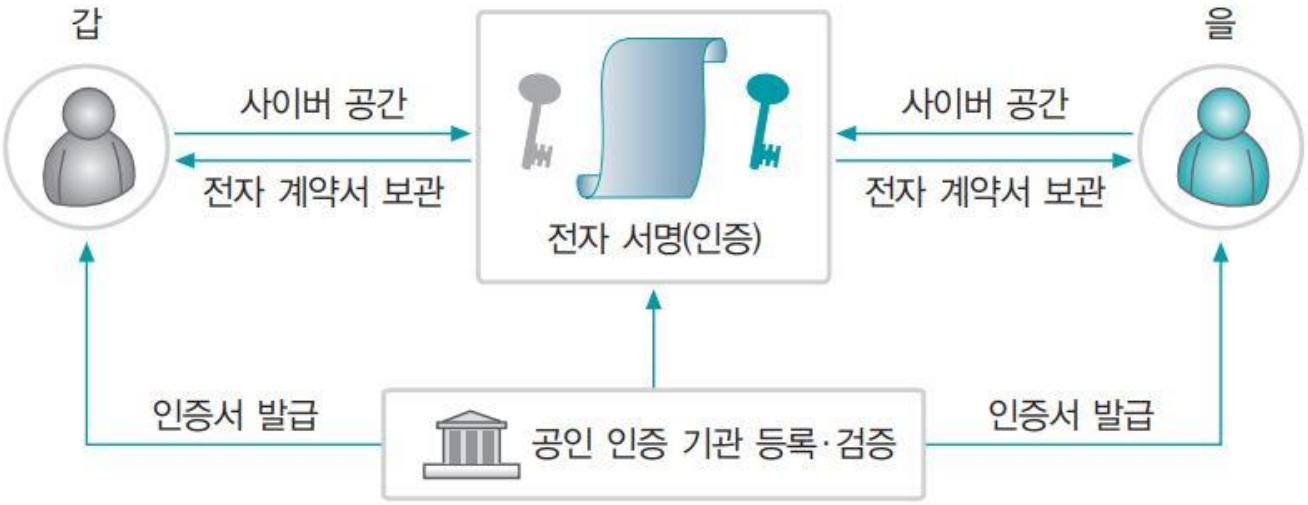
전자 서명이란

- 전자서명법: 서명자가 해당 전자 문서에 서명하였음을 나타내기 위해 전자 문서에 첨부되거나 논리적으로 결합된 전자적 형태의 정보
- 인감도장처럼 전자서명도 공인된 인증 기관에 등록 및 검증하여 사용 가능



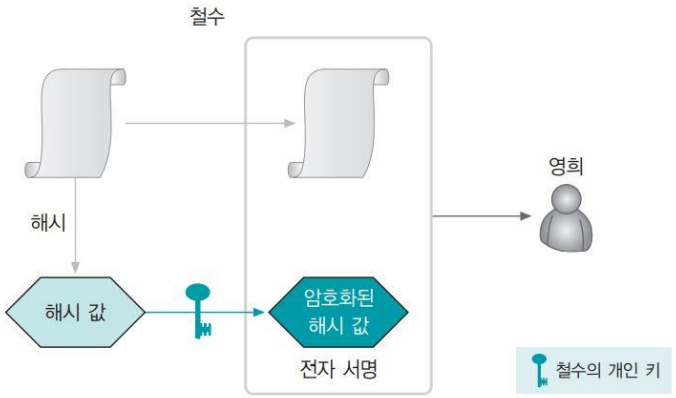
공개 키 기반 구조 및 활용

- 전자 서명과 전자 봉투
  - 전자 서명 (Digital Signature)
    - 전자서명법: 서명자가 해당 전자 문서에 서명하였음을 나타내기 위해 전자 문서에 첨부되거나 논리적으로 결합된 전자적 형태의 정보
    - 인감도장처럼 전자서명도 공인된 인증 기관에 등록 및 검증하여 사용 가능

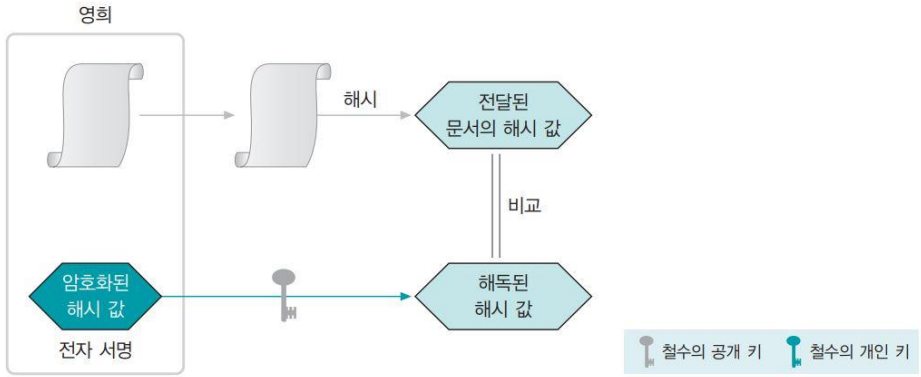


공개 키 기반 구조 및 활용

- 전자 서명과 전자 봉투
  - 전자서명 구현 원리
    - 전자서명은 원본의 해시 값을 구한 뒤 부인 방지 기능을 부여하기 위해 공개 키 방법을 사용
    - 복호화한 해시 값과 편지에서 구한 해시 값이 일치하면 위조 되지 않았다고 확신할 수 있음



전자 서명 생성



전자 서명이 된 전송 문서 확인

공개 키 기반 구조 및 활용

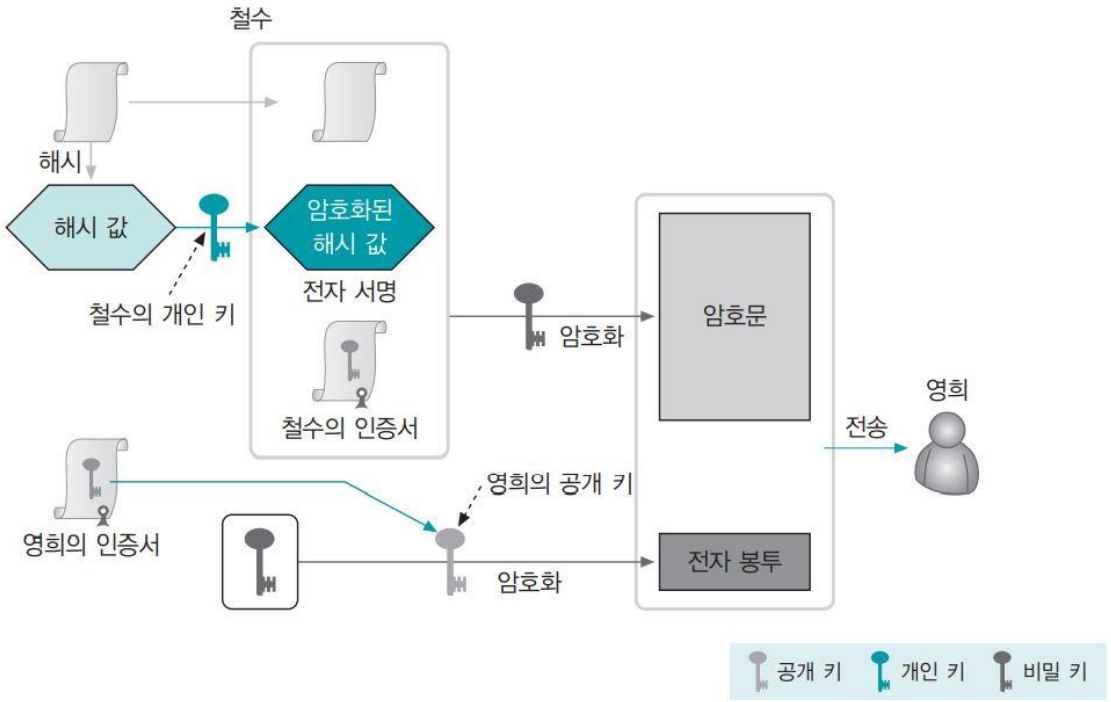
- 전자 서명과 전자 봉투
  - 전자 서명이 제공하는 조건 5가지
    - 위조 불가 (Unforgeable)
      - 서명자만이 서명문을 생성
    - 인증 (User authentication)
      - 서명문의 서명자를 확인
    - 재사용 불가 (Not reusable)
      - 서명문의 해시 값을 전자서명에 이용하므로 한 번 생성된 서명을 다른 문서의 서명으로 사용할 수 없음
    - 변경 불가 (Unalterable)
      - 서명된 문서는 내용을 변경할 수 없기 때문에 데이터가 변조되지 않았음을 보장하는 무결성을 만족
    - 부인 방지 (Non-repudiation)
      - 서명자가 서명한 사실을 나중에 부인할 수 없음
  - 전자 서명의 대표적인 표준
    - 1994년 미국에서 만들어진 DSS, 우리나라는 1996년 개발된 KCDSA가 있음
    - 우리나라의 전자서명법에 따르면 인터넷을 통해 전자 문서를 교환할 때 전자 서명은 일반 문서에 쓰이는 인감도장과 법적으로 똑같은 효력

공개 키 기반 구조 및 활용

전자 서명과 전자 봉투

전자봉투(Digital Envelope)

- 전달하려는 메시지를 암호화하여 한 사람을 통해 보내고 암호화 키는 다른 사람이 가져가도록 암호학적으로 구현



전자 봉투를 이용한 암호 전송

1. 철수는 전자 봉투를 사용하기 위해 먼저 전자 서명을 생성

2. 전자 서명과 원문, 자신의 공개 키가 들어 있는 인증서를 비밀 키(DES 알고리즘 등에 사용되는 대칭 키)로 암호화

3. 전자 서명 세트와 인증서, 암호화한 비밀 키가 영희의 공개 키로 암호화

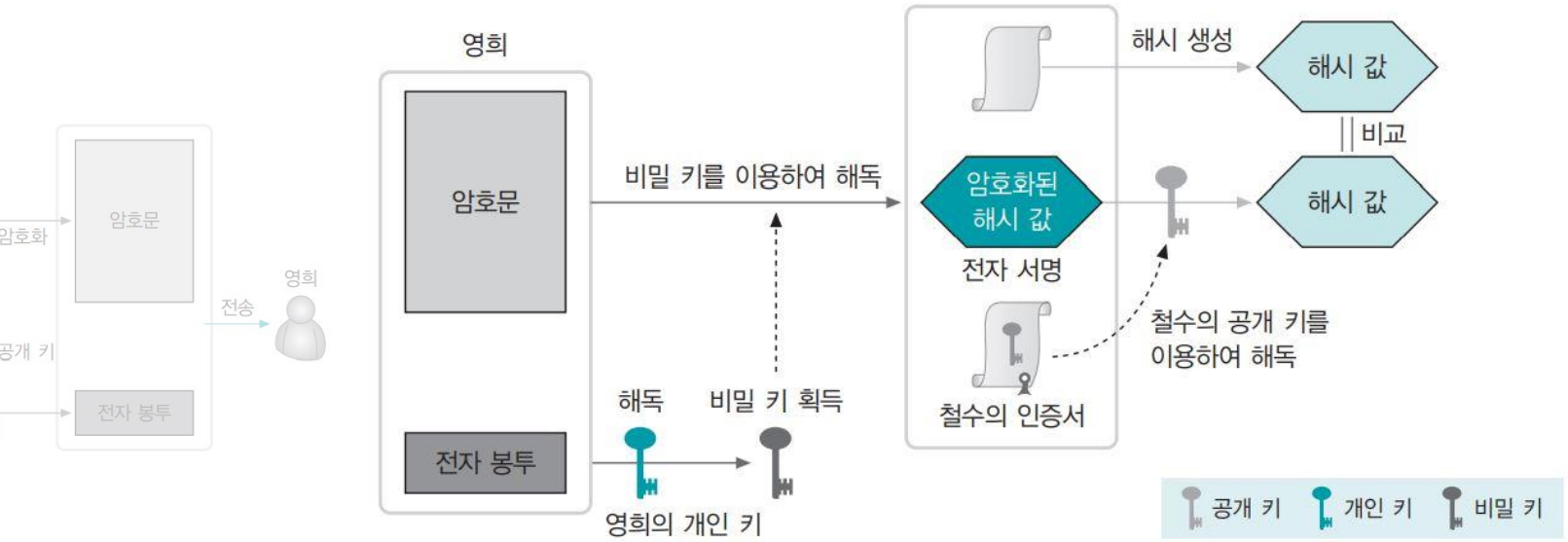
4. 최종적으로 철수는 비밀 키로 암호화한 결과와 비밀 키가 암호화된 전자 봉투를 영희에게 전송

공개 키 기반 구조 및 활용

전자 서명과 전자 봉투

전자봉투(Digital Envelope)

- 전자봉투는 기밀성, 무결성, 부인 방지를 모두 지원
  - 전달받은 영희는 전자 봉투를 자신의 개인 키로 복호화하여 비밀 키를 획득
  - 비밀 키를 이용하여 전자 서명과 편지, 철수의 인증서를 복호화(해독)
  - 복호화한 인증서에서 철수의 공개 키를 얻어 전자 서명을 복호화하고 이를 편지의 해시 결과와 비교



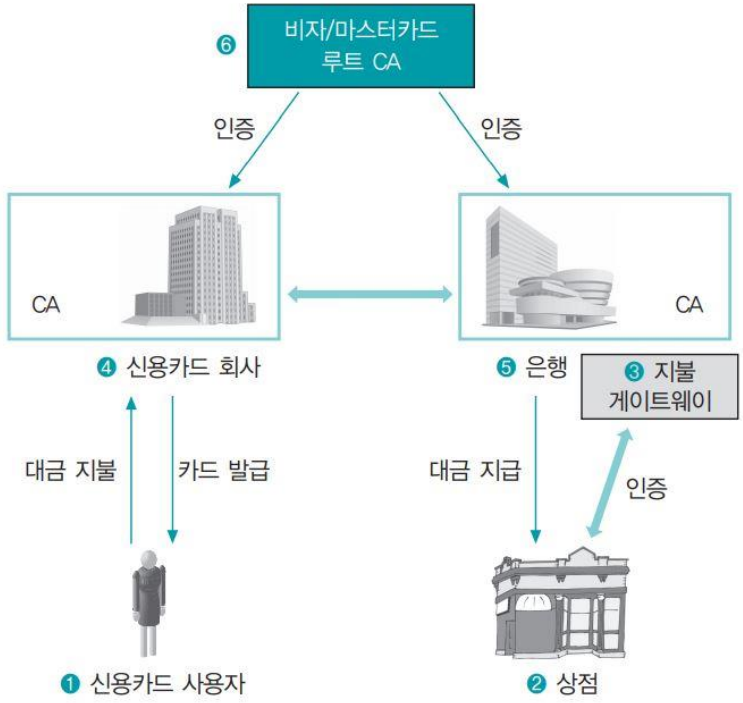
전자 봉투 복호화

공개 키 기반 구조 및 활용

전자결제

SET (Secure Electronic Transaction)

- 1996년 비자와 마스터카드의 합의로 만들어진 프로토콜
- 신용카드 거래에서 사실상의 표준



SET 구성요소

① 신용카드 사용자: 신용카드를 소지한 사람
SET에 이용하는 공인 인증서를 소유

② 상점: 인터넷 쇼핑몰을 운영하며 SET를 이용하여 상품을 판매

③ 지불 게이트웨이(PG): 기존의 신용카드 지불 방식으로 은행과 거래 내역을 주고받음

④ 신용카드 회사: 사용자에게 신용카드를 발급하고, CA를 운영하여 사용자에게 공인 인증서를 발급

⑤ 은행: 상점의 계좌가 있는 곳으로 지불 게이트웨이를 운영하고, CA를 운영하여 상점에 공인 인증서를 발급

⑥ 인증 기관: SET에 참여하는 모든 구성원의 정당성을 보장하는 루트 CA

** Certificate Authority (CA): 디지털 인증서를 발급하는 인증기관

공개 키 기반 구조 및 활용

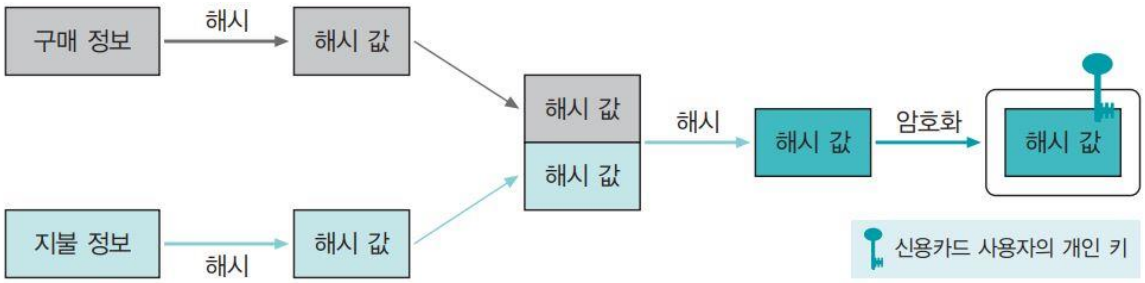
- 전자결제
  - SET (Secure Electronic Transaction)
    - SET 지불 과정
      - 신용카드 사용자가 SET를 이용하여 상점에 결제 의뢰
      - 주문서를 받은 판매자는 고객의 신용카드 회사에서 신용카드의 유효성 여부 확인
      - 신용카드가 정상임을 확인하면 주문 확인 메시지를 고객에게 전송하며, 고객은 자신의 신용카드 정보를 판매자에게 전송
      - 판매자는 고객에게 받은 정보를 신용카드 결제에 다시 이용, 이때 SET는 **전자봉투**와 **이중 서명**을 사용

공개 키 기반 구조 및 활용

전자결제

SET (Secure Electronic Transaction)

- 이중 서명
  - 신용카드 사용자의 **구매 정보**와 **지불 정보**를 각각 해시한 후 두 값을 합하여 다시 해시함
  - 최종 해시 값을 신용카드 사용자의 개인 키로 암호화(서명)하면 이중 서명 값 생성
  - **이중 서명**은 상점에 대금을 지불하는 은행은 신용카드 사용자가 구입한 물건을 모르지만, **상점이 요구한 결제 대금이 정확한지 확인을 위한 목적**



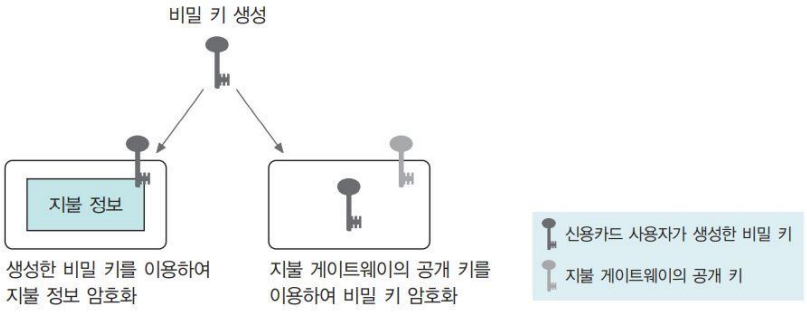
이중 서명의 기본 동작

공개 키 기반 구조 및 활용

전자결제

SET (Secure Electronic Transaction)

- 이중 서명 원리 - 1
  - **신용카드 사용자**는 하나의 **비밀 키(대칭 키)를 생성**
  - **비밀 키**를 사용하여 **지불 정보를 암호화**
  - **비밀 키**는 은행이 운영하는 지불 게이트웨이(PG)의 **공개 키**로 암호화
  - 신용카드 사용자는 결제를 위한 데이터를 모두 생성하여 상점에 전송



비밀키 생성

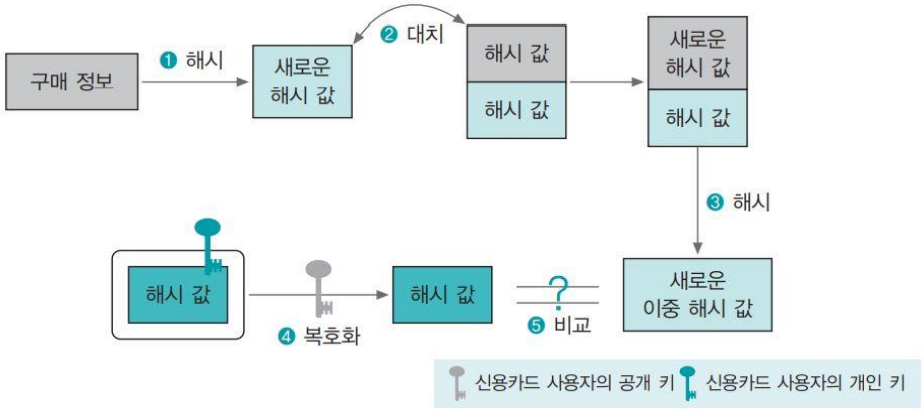
결제 시 신용카드 사용자가 상점에 전송하는 데이터

공개 키 기반 구조 및 활용

전자결제

SET (Secure Electronic Transaction)

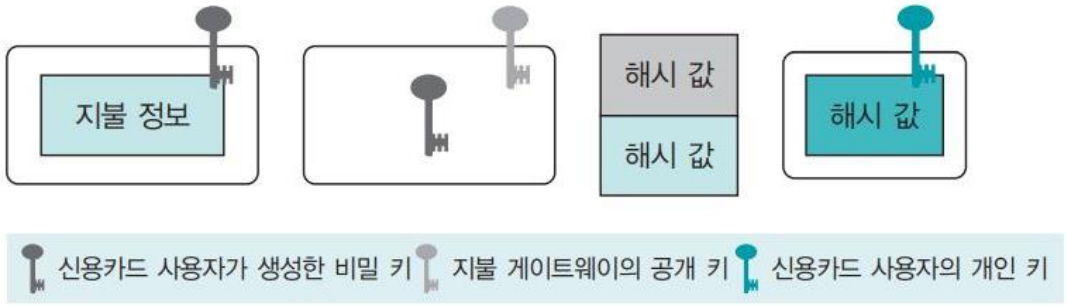
- 이중 서명 원리 - 2
  - 상점은 구매 정보를 확인
    - 신용카드 사용자가 구매한 물건에 대한 **구매 정보**의 해시를 구함
    - 신용카드 사용자가 보내온 한 쌍의 해시 값을 새로 구한 해시로 대치
    - 새로운 이중 해시를 구함
    - 신용카드 사용자의 개인 키로 암호화된 해시 값을 복호화하여 이를 새로 구한 이중 해시 값과 비교



이중 해시 값을 이용한 구매 정보 확인

공개 키 기반 구조 및 활용

- 전자결제
  - SET (Secure Electronic Transaction)
    - 이중 서명 원리 - 3
      - 구매 정보를 확인한 상점은 다시 데이터 세트를 만들어 **지불 게이트웨이로 전송**
      - 상점이 지불 게이트웨이로 보내는 데이터는 구매 정보만 빼면 신용카드 사용자가 처음 상점에 전송한 데이터와 같음



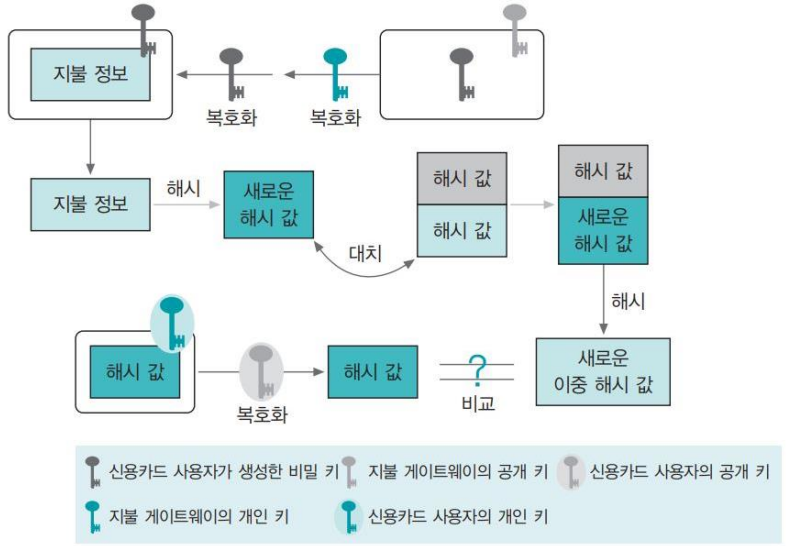
상점이 지불 게이트웨이로 보내는 데이터

공개 키 기반 구조 및 활용

전자결제

SET (Secure Electronic Transaction)

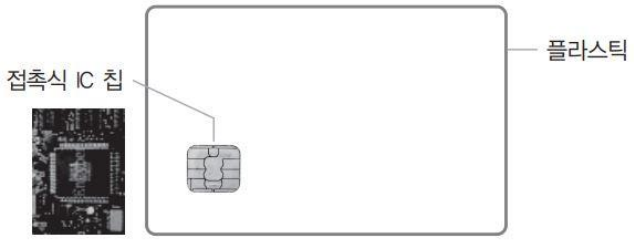
- 이중 서명 원리 - 4
  - 데이터를 상점으로부터 받은 지불 게이트웨이(PG)는 자신의 개인 키로 비밀 키를 복호화하여 지불 정보를 확인
  - 상점이 한 것처럼 지불 정보를 해시한 값으로 한 쌍의 해시 값을 대치하여 이중 해시 값을 비교
  - 지불 정보의 변조 여부를 확인한 뒤 상점에 대금을 지불



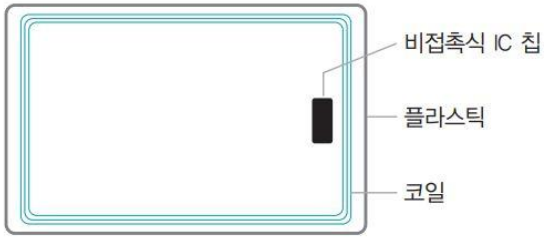
지불 정보 확인

공개 키 기반 구조 및 활용

- 스마트카드 인증
  - 스마트카드
    - 접촉식
      - 스마트카드 리더기와 스마트카드 접촉부CHIP 사이의 물리적 접촉으로 작동하는 스마트카드
    - 비접촉식
      - 안테나 겸 전기 코일로 이용되는 구리선을 통해 무선 주파수 파장을 전력으로 전환하는 방식으로 구동하여 스마트카드 리더기와 통신하는 카드



접촉식 카드의 구조



비접촉식 카드의 구조

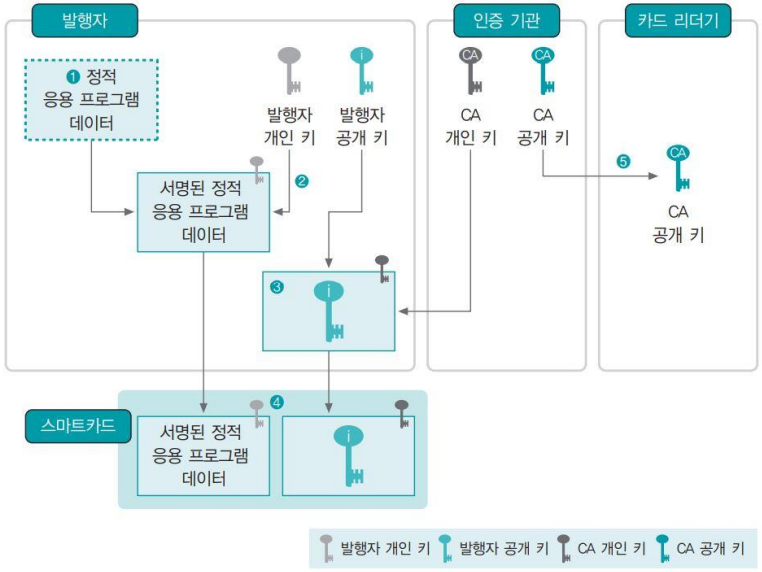
  - 정적 데이터 인증
    - 인증할 때마다 같은 데이터를 사용하는 방식
    - 정적 데이터 인증을 이용한 스마트카드의 대표적인 예는 출입 카드

공개 키 기반 구조 및 활용

스마트카드 인증

정적 데이터 인증 스마트카드의 발행 구조

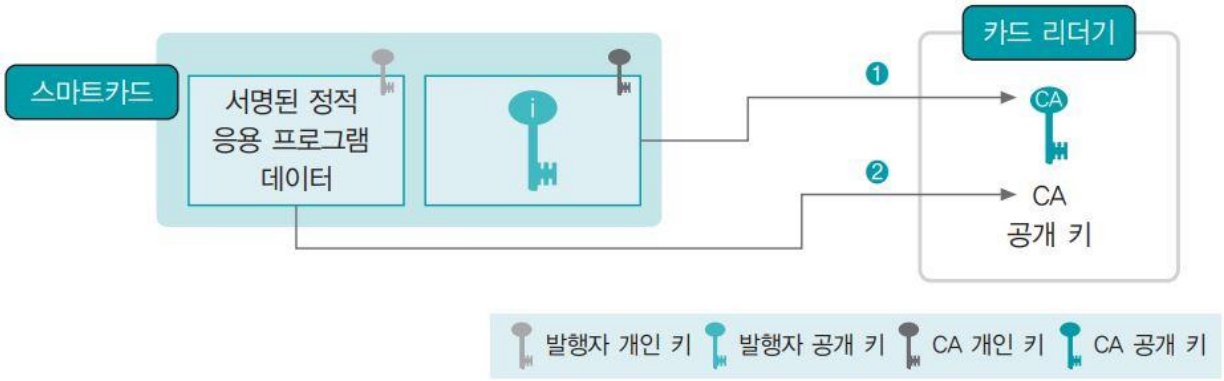
① 정적 응용 프로그램 데이터
② 정적 응용 프로그램 데이터 암호화
③ 인증 기관(CA)의 개인 키로 발행자의 공개 키를 암호화
④ 인증 데이터 저장
⑤ 인증 기관의 공개 키를 스마트카드 리더기에 배포



정적 데이터 인증 스마트카드의 발행 구조

공개 키 기반 구조 및 활용

- 스마트카드 인증
  - 정적 데이터 인증 스마트카드의 인증 과정
    - 1. 인증 기관의 개인 키로 암호화된 발행 기관의 공개 키가 전달, 전달된 발행 기관의 공개 키는 카드 리더기에 저장된 인증 기관의 공개 키로 복호화
    - 2. 복호화된 인증 기관의 공개 키로 스마트카드에 저장된 '서명된 정적 응용 프로그램 데이터'를 복호화하여 카드 리더기가 정적 응용 프로그램 데이터를 확인

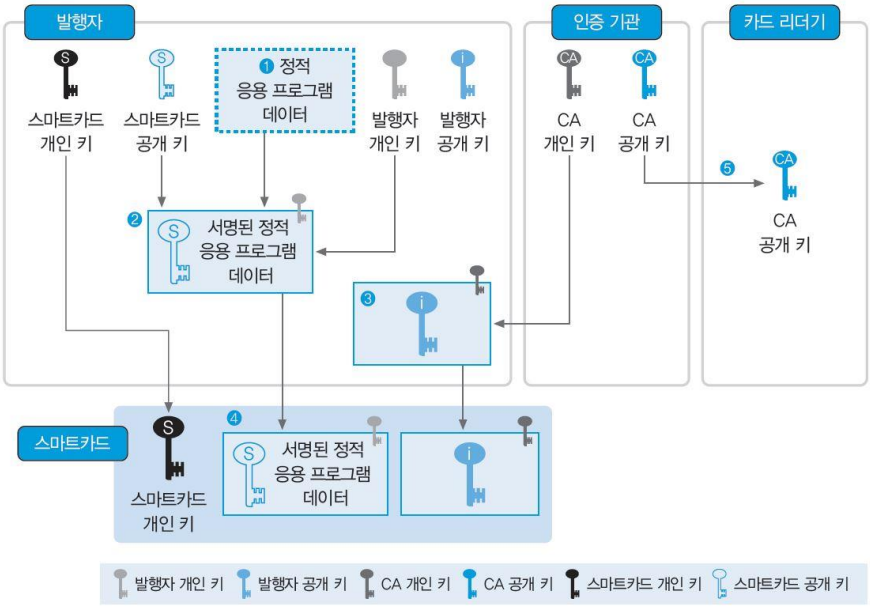


정적 데이터 인증 스마트카드의 인증 과정

공개 키 기반 구조 및 활용

스마트카드 인증

동적 데이터 인증 스마트카드의 인증 과정



동적 데이터 인증 스마트카드의 인증 과정

① 카드 리더기는 스마트카드에 임의의 난수와 기타 동적 데이터를 생성하여 스마트카드로 전송

② 스마트카드는 자신의 스마트카드 개인 키를 사용하여 전달받은 데이터와 기타 응용 프로그램 데이터를 암호화(서명)하여 카드 리더기에 전달

③ 인증 기관CA의 개인 키로 암호화된 발행 기관의 공개 키가 전달
전달된 발행 기관의 공개 키는 리더기에 저장된 인증 기관의 공개 키로 복호화

④ 복호화된 인증 기관의 공개 키로 스마트카드에 저장된 '서명된 정적 응용 프로그램 데이터'를 복호화하고 카드 리더기는 정적 응용 프로그램 데이터와 스마트카드의 공개 키를 확인

⑤ ④에서 얻은 스마트카드의 공개 키를 이용하여 ②에서 스마트카드로부터 전달받은 데이터를 복호화하고 스마트카드에서 전송한 데이터의 진위 여부를 확인하여 처리

Summary

- 공개키 암호화 시스템
- 공개 키 기반 구조 및 활용

참고문헌

- 정보 보안 개론 - 한권으로 배우는 핵심 보안 이론, 양대일, 한빛 아카데미
- 현대 암호학 개론, 이동훈, 이룬 출판

Q & A