2026/07/22, 2026 보안 기초 세미나

암호학과 네트워크 보안

- 13장 디지털 서명 -

이 성 현(seonghyeon@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

- 보충
- 개요
- 구조
- 변형 서명 구조

목 차

- 보충
- 개요
- 구조
- 변형 서명 구조

보충 - 개요

- 디지털 서명 공격
  - 키-단독 공격(Key-only Attack)
    - 공격자가 서명자의 공개키와 공개된 시스템 정보만을 이용하는 공격
      - e.g., 위조된 서명과 공개키를 이용해서 위조된 메시지를 생성함
  - 알려진 메시지 공격(Known-message Attack)
    - 공격자가 서명 대상 메시지 선택에 관여하지 않은 한 개 이상의 공개된 메시지-서명 쌍을 이용하는 공격
      - e.g., 공개된 여러 메시지-서명 쌍을 분석하여 새로운 메시지의 서명을 위조함
  - 선택 메시지 공격(Chosen-message Attack)
    - 공격자가 서명 받을 메시지를 직접 선택하고, 해당 메시지에 대한 서명을 획득하여 이용하는 공격
      - e.g., 공격자가 선택한 메시지들의 서명을 받은 뒤, 이를 이용하여 다른 메시지의 서명을 위조함

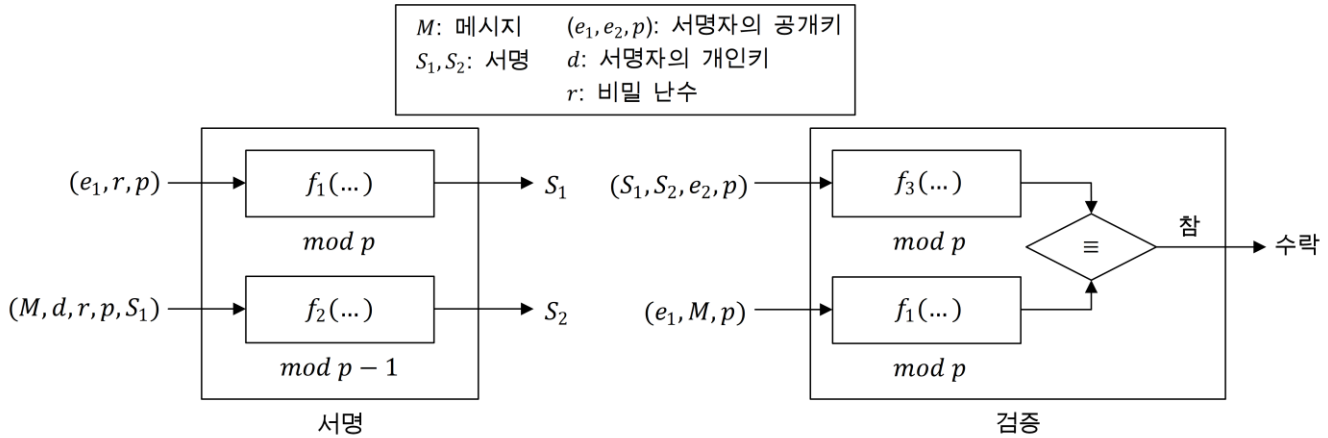
보충 - 구조

- ElGamal Digital Signature Scheme (1/4)
  - 개요
    - ElGamal 암호화 알고리즘을 이용한 디지털 서명 구조로, 서명할 때마다 새로운 비밀 난수를 사용함

군(Group):
하나의 연산에 대해 닫힘성, 결합법칙, 항등원 및 역원 조건을 만족하는 집합

  - 키 생성
    1. 소수 p와 곱셈군 $Z_p{}^*$ 내에서의 원시근 e_1를 선택하고, $p-1$보다 작은 개인키 d를 선택함
    2. 공개값 $e_2 = {e_1}^d$를 계산하여 (e_1, e_2, p)를 공개키로 사용함

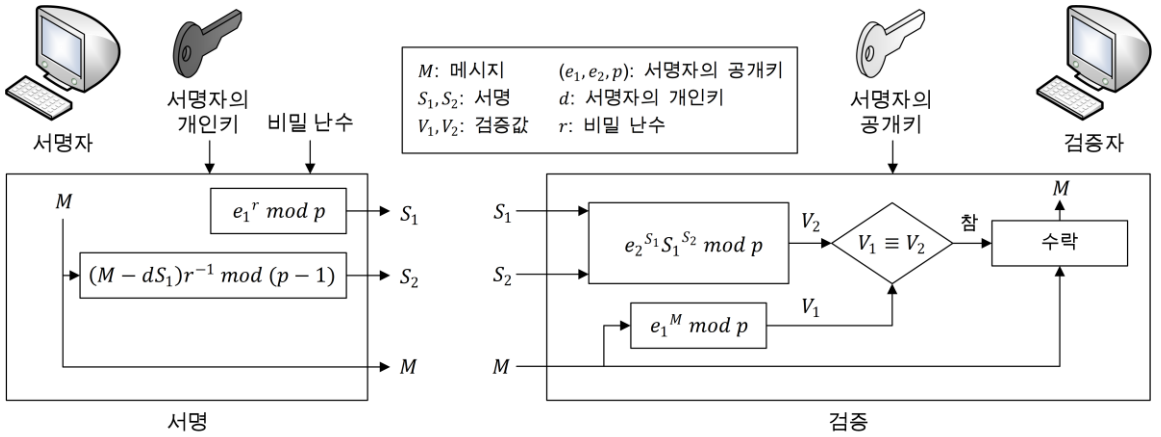
원시근(Primitive Root):
거듭제곱을 통해 곱셈군의 모든 원소를 생성할 수 있는 원소



<그림 9> ElGamal 디지털 서명 구조의 개념도

보충 - 구조

- ElGamal Digital Signature Scheme (2/4)
  - 서명 및 검증
    - 서명자는 매 서명마다 비밀 난수 r을 선택하고 $S_1 = {e_1}^r \; mod\, p, S_2 = (M - d \times S_1) \times r^{-1} \; mod\, (p-1)$를 계산하여 서명을 생성하고 메시지와 서명을 전송함
    - 검증자는 $0 < S_1 < p, 0 < S_2 < p-1$인지 확인하고, $V_1 = {e_1}^M \; mod\, p, V_2 = {e_2}^{S_1} \times {S_1}^{S_2} \; mod\, p$를 계산한 후, $V_1 \equiv V_2$이면 메시지를 수락함



<그림 10> ElGamal 디지털 서명 구조

보충 - 구조

- ElGamal Digital Signature Scheme (3/4)
  - 발생 가능한 공격 (1/2)
    - 키-단독 공격
      - 공격자가 특정 M에 대한 서명을 위조하려는 경우, 임의의 S_1을 선택하고, 그에 대한 S_2를 얻기 위해서는 $S_2 = \log_{S_1}({e_1}^M {e_2}^{-S_1})$를 계산해야 하며, 이는 이산대수 문제로, 복잡한 연산을 필요로 함
      - 공격자가 임의의 x, y를 정하고, $M = xS_2 \ mod\ (p-1), S_1 = -yS_2 \ mod\ (p-1), S_1 = e_1^x e_2^y \ mod\ p$를 만족하는 (M, S_1, S_2)를 계산할 수 있으나, 해당 방식의 위조는 실존적 위조가 됨
    - 알려진 메시지 공격
      - 공격자가 (M, S_1, S_2)를 가로챈 후, S_1, S_2가 유효한 M'를 찾기 위해 ${e_1}^{M'} \equiv {e_1}^M \ mod\ p$를 만족하는 M'을 찾을 수 있으나, 해당 방식의 위조는 실존적 위조가 됨

이산대수 문제(Discrete Logarithm Problem):
g와 y가 주어졌을 때 $g^x = y$를 만족하는 지수 x를 찾는 문제

보충 - 구조

- ElGamal Digital Signature Scheme (4/4)
  - 발생 가능한 공격 (2/2)
    - 선택 메시지 공격
      - 공격자가 선택한 두 메시지 M_1, M_2에 동일한 비밀 난수 r이 사용된 서명 $(S_{1,1}, S_{1,2}), (S_{2,1}, S_{2,2})$을 얻으면, 두 서명식의 차이 $(S_{2,2} - S_{2,1})r \equiv M_1 - M_2 \ (mod\ p-1)$를 이용하여 r을 구할 수 있음
      - 구한 r과 기존 서명을 이용하여 공격자가 원하는 메시지에 대한 유효한 서명을 생성할 수 있으므로 선택적 위조가 가능함
    - 비밀 난수 의존성
      - ElGamal 디지털 서명에 대한 공격의 난이도는 비밀 난수 r의 생성 및 관리 안전성에 의존함
      - 비밀 난수 r이 재사용되거나 예측되면 개인키가 유출되어 선택적 위조가 가능해질 수 있음

보충 - 구조

- ElGamal 계열 디지털 서명 구조의 비교
  - ElGamal 서명 구조는 비밀 난수와 개인키의 결합을 통해 개인키 추출 및 서명 위조를 어렵게 함
  - Schnorr 서명 구조는 ElGamal 서명 구조를 기반으로 메시지와 임시값을 함께 해시하여 임의의 서명값에 맞는 메시지를 생성하는 위조를 어렵게 함
  - DSS는 ElGamal 서명 구조를 $mod\ q$ 연산 중심으로 재구성하여 각 서명을 q 크기로 압축함

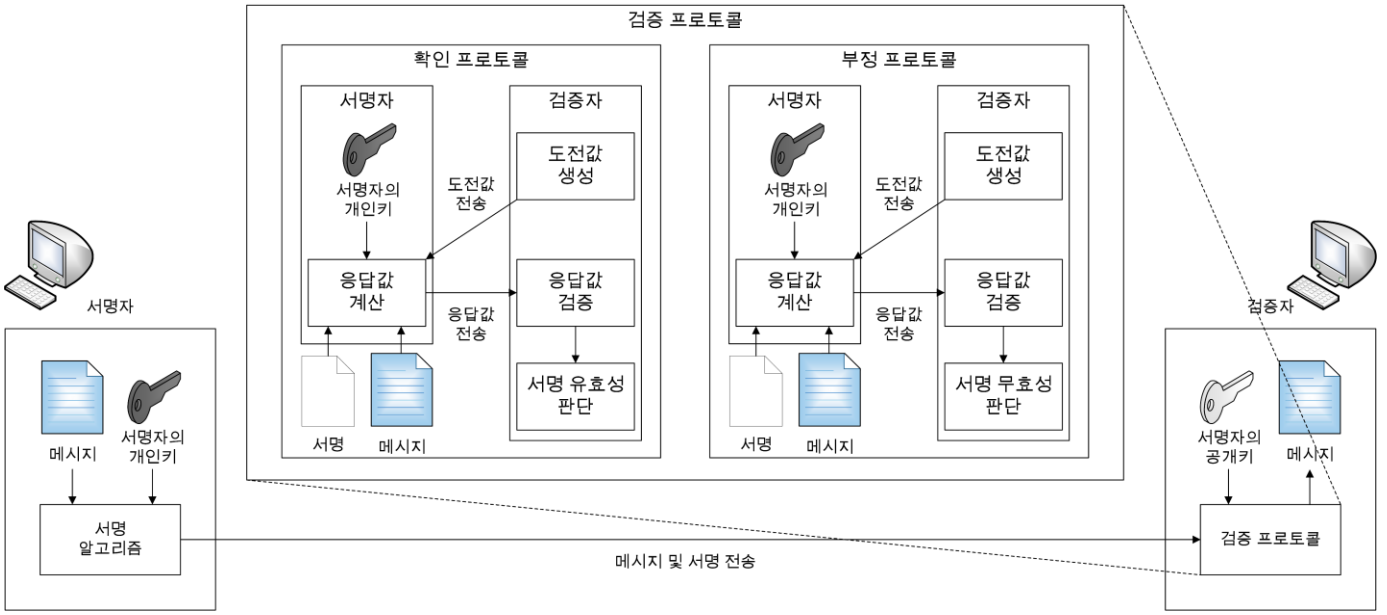
구분	서명 길이	서명 생성 방식	특징
ElGamal	약 $2\|p\|$	메시지, 개인키, 비밀 난수를 결합함	서명값 두 개를 모두 Z_p 기반 연산으로 생성함
Schnorr	약 해시값 길이 $+\|q\|$	메시지와 임시값을 함께 해시하고, 해시값을 개인키 및 비밀 난수와 결합함	해시값 및 q 범위의 서명값으로 서명을 구성함
DSS	약 $2\|q\|$	임시값을 $mod\ q$로 축소하고, 다이제스트와 개인키를 비밀 난수와 결합함	두 서명값 길이를 모두 q 범위로 생성함

보충 - 변형 서명 구조

- Undeniable Digital Signature Scheme (1/7)
  - 개요
    - 일반 디지털 서명과 달리, 서명자의 협조 없이는 서명의 진위를 확인할 수 없도록 한 방식
  - 목적
    - 서명의 유효성 검증을 서명자가 통제할 수 있도록 함
    - 제3자에 의한 무단 검증을 제한함
  - 과정
    - 검증 알고리즘 대신 확인 프로토콜(Confirmation Protocol)을 사용하여 서명자가 검증자의 도전값(Challenge)에 대한 응답값(Response)를 전송하여 검증을 진행하도록 함
    - 부정 프로토콜(Disavowal Protocol)을 사용하여 유효하지 않은 서명에 대해, 서명자가 서명하지 않았음을 증명할 수 있도록 함

보충 - 변형 서명 구조

- Undeniable Digital Signature Scheme (2/7)
  - 구현
    - 확인 프로토콜은 공개키와 서명 사이의 이산대수 관계가 동일함을 도전-응답 방식으로 증명함
    - 부정 프로토콜은 후보 서명이 해당 관계를 만족하지 않음을 서명값 자체나 개인키를 공개하지 않고 증명함



<그림 19> Undeniable Digital Signature Scheme 개념도

보충 - 변형 서명 구조

- Undeniable Digital Signature Scheme (3/7)
  - 동작 과정 (1/5)
    - 키 생성
      - 소수 p, Z_p^*의 생성원 e_1, 개인키 d를 선택하고 공개키 $e_2 = e_1^d \ (mod\ p)$를 계산하여 공개키를 (e_1, e_2, p)로, 개인키를 d로 정함
    - 서명 알고리즘
      - 서명 $S = M^d \ (mod\ p)$로 하여 메시지-서명 쌍 (M, S)를 검증자에게 전달함
      - 정상적인 서명인 경우, $e_2 = e_1^d \ (mod\ p)$와 $S = M^d \ (mod\ p)$가 모두 성립함

보충 - 변형 서명 구조

- Undeniable Digital Signature Scheme (4/7)
  - 동작 과정 (2/5)
    - 확인 프로토콜
      1. 서명자가 비밀 난수 $r \in_R \{0,1,\dots,p-2\}$을 선택하고 $A = e_1^r \ (mod\ p), B = M^r \ (mod\ p)$를 계산하여 검증자에게 전달함
      2. 검증자가 무작위 도전값 $c \in_R \{0,1,\dots,p-2\}$를 선택하여 서명자에게 전달함
      3. 서명자는 응답값 $T = (r + cd)\ mod\ (p-1)$를 계산하여 검증자에게 전달함
      4. 검증자는 $e_1^T = e_1^{r+cd} = e_1^r\left(e_1^d\right)^c = Ae_2^c \ (mod\ p), M^T = M^{r+cd} = M^r\left(M^d\right)^c = BS^c \ (mod\ p)$를 계산하고, 두 식이 모두 성립하면 서명 S를 유효하다고 판정함

보충 - 변형 서명 구조

- Undeniable Digital Signature Scheme (5/7)
  - 동작 과정 (3/5)
    - 부정 프로토콜
      1. 검증자가 두 비밀값 $a \in_R \{0,1,\dots,k\}, b \in_R \{0,1,\dots,p-2\}$ (단, k는 프로토콜 상에서 서로 사전에 합의한 정수)를 선택함
      2. 검증자는 선택한 a, b에 대해 $C_1 = M^a e_1^b \ (mod\ p), C_2 = S^a e_2^b \ (mod\ p)$를 계산하여 서명자에게 전달함
      3. 서명자는 개인키 d를 이용하여 $R = C_1^d C_2^{-1} (mod\ p)$를 계산하고, 이는 $R = C_1^d C_2^{-1} = \left(M^a e_1^b\right)^d \left(S^a e_2^b\right)^{-1} = M^{ad} e_1^{bd} S^{-a} e_2^{-b} = M^{ad} S^{-a} = \left(M^d S^{-1}\right)^a \ (mod\ p)$로 볼 수 있음

보충 - 변형 서명 구조

- Undeniable Digital Signature Scheme (6/7)
  - 동작 과정 (4/5)
    - 부정 프로토콜

커밋먼트(Commitment):
특정 값을 공개하지 않은 채 미리 고정하고, 이후 해당 값을 공개하여 확인할 수 있도록 하는 암호학적 기법

4. 서명이 유효한 경우, 서명자는 a를 알 수 없음
   - $S = M^d \ (mod\ p)$이므로 $M^d S^{-1} \equiv 1 \ (mod\ p)$이 되어 $R = 1^a = 1 \ (mod\ p)$이 되므로, 서명자는 R만으로는 a를 알 수 없음

5. 서명이 무효한 경우, 서명자는 a를 찾을 수 있음
   - $S \neq M^d \ (mod\ p)$이므로 $M^d S^{-1} \not\equiv 1 \ (mod\ p)$이고, 서명자는 $W = M^d S^{-1} \ (mod\ p)$를 알고 있으므로 $R = W^a \ (mod\ p)$에서 $W^0, W^1, \ldots, W^k$를 계산하는 과정에서 a를 찾을 수 있음

6. 서명자는 임의의 비밀값 ρ를 이용하여 커밋먼트 $D = Com(a, \rho)$를 생성하고 검증자에게 전달함

보충 - 변형 서명 구조

- Undeniable Digital Signature Scheme (7/7)
  - 동작 과정 (5/5)
    - 부정 프로토콜
      7. 검증자가 b를 공개하고 서명자는 알아낸 a와 공개된 b를 통해 $C_1 = M^a e_1^b \ (mod\ p), C_2 = S^a e_2^b \ (mod\ p)$가 모두 성립하는지 확인함
      8. 서명자가 ρ를 공개하고 검증자는 이전에 전달받은 D와 공개된 ρ를 통해 $Open(D, \rho) = a$가 성립하는지 확인하고, 성립할 경우, 서명이 무효하다고 판정함

목 차

- 보충
- 개요
- 구조
- 변형 서명 구조

개요

- 디지털 서명(Digital Signature)
  - 정의
    - 서명자가 개인키를 이용하여 메시지에 대한 서명을 생성하고 검증자가 공개키를 이용하여 서명을 검증하는 기술
  - 목적
    - 메시지가 서명자가 보낸 메시지가 맞는지 확인함
    - 메시지가 중간에 위·변조 되지 않았는지 확인함
  - 특징
    - 개인키를 보유한 서명자만이 유효한 서명을 생성할 수 있음
    - 공개키를 이용하여 불특정 다수가 서명을 검증할 수 있음
    - 메시지나 해당 메시지에 사용한 개인키에 따라 다른 서명이 생성됨

메시지(Message):
계약서, 이메일, 거래 내역, 파일 등 서명자가 작성하거나 승인한 디지털 정보

개요

- 디지털 서명과 전통적 서명의 차이점 (1/3)
  - 서명-문서 결합 형태
    - 전통적 서명은 문서의 일부로서 포함됨
      - 수표 또는 계약서 등의 문서 위에 서명을 작성함
    - 디지털 서명은 문서와 서명이 분리된 상태로 전달함
      - 문서와 서명이 논리적으로 분리되어 있음
  - 검증방법
    - 전통적 서명은 문서의 서명과 서명 견본의 동일 여부를 확인함
      - 검증을 위해서는 검증할 서명에 대한 서명 견본을 가지고 있어야 함
    - 디지털 서명은 검증 기술을 통해 서명의 유효성을 검증함
      - 검증을 위한 서명 파일 저장이 불필요해짐

개요

- 디지털 서명과 전통적 서명의 차이점 (2/3)
  - 서명-문서 관계
    - 전통적 서명은 서명이 문서와 독립적임
      - 문서 내용과 관계 없이 동일한 서명을 사용할 수 있음
    - 디지털 서명은 서명이 문서에 종속적임
      - 문서 내용에 의존하여, 문서에 따라 다른 서명이 생성됨
  - 복제본 식별성
    - 전통적 서명은 서명된 문서의 복사본이 원본 문서와 구별됨
      - 물리적인 서명이므로 복사본이 원본과 정확히 일치하기 어려움
    - 디지털 서명은 복사본과 원본 문서의 구별이 어려움
      - 디지털 정보로 이루어지므로 복사 시, 원본과 동일한 복사본이 생성됨

개요

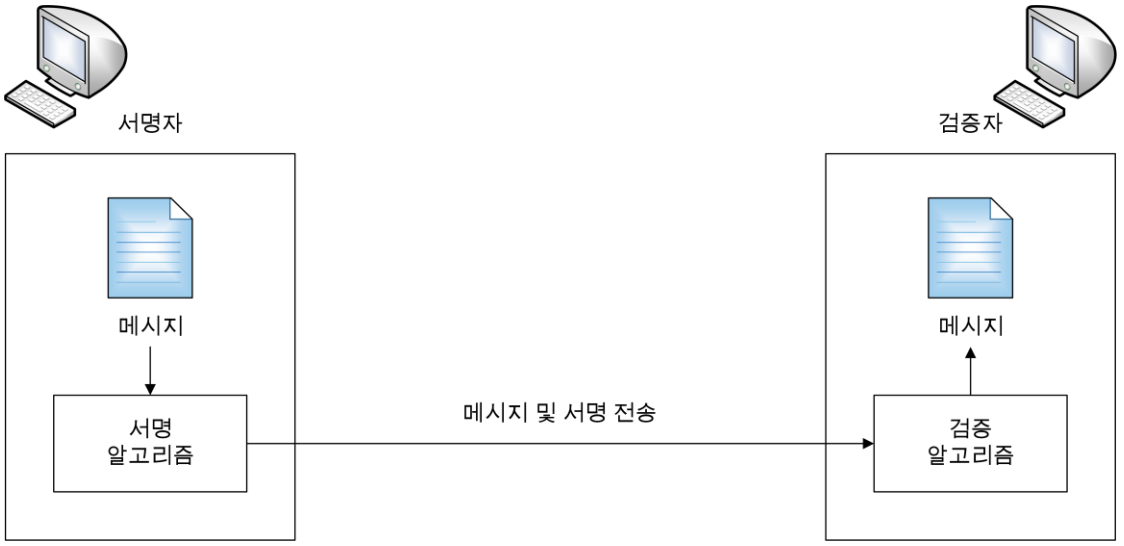
- 디지털 서명과 전통적 서명의 차이점(3/3)
  - 정리
    - 전통적 서명과 디지털 서명은 서명-문서 결합 형태, 검증 방법, 서명-문서 관계, 복제본 식별성에 대해서 다음과 같은 차이가 존재함
    - 전통적 서명은 물리적 특징을 기반으로 검증하지만, 디지털 서명은 문서 내용과 키의 암호학적 관계를 기반으로 검증함

구분	전통적 서명	디지털 서명
서명-문서 결합 형태	서명이 문서에 포함	서명과 문서가 분리
검증방법	서명 견본과 직접 비교	검증 기술을 통해 검증
서명-문서 관계	문서 내용과 독립적	문서 내용에 종속적
복제본 식별성	원본과 구별되는 복제	원본과 동일한 복제

[표 1] 전통적 서명과 디지털 서명의 비교

개요

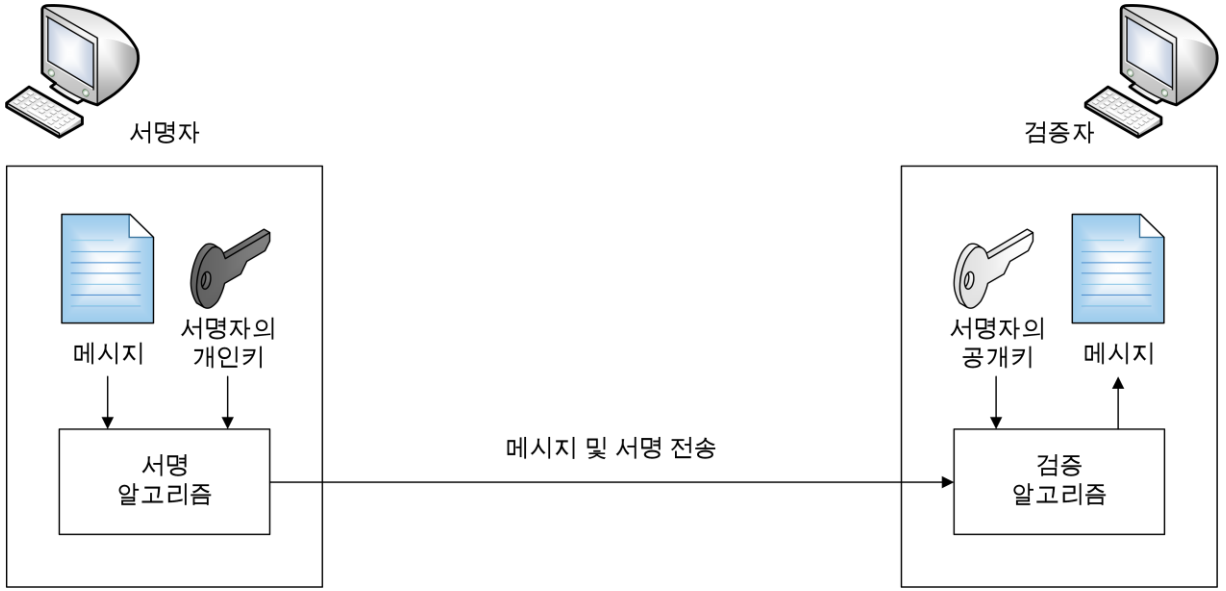
- 서명 및 검증
  - 서명(Signing)
    - 서명자가 서명 알고리즘(Signing Algorithm)을 이용하여 메시지에 대한 서명을 생성함
  - 검증(Verification)
    - 검증자가 검증 알고리즘(Verification Algorithm)을 이용하여 메시지와 서명의 유효성을 확인함



<그림 1> 디지털 서명 과정

개요

- 비대칭키 사용 (1/2)
  - 서명자는 서명자의 개인키를 이용하여 메시지에 대한 서명을 진행함
  - 검증자는 서명자의 공개키를 이용하여 서명에 대한 검증을 진행함



<그림 2> 비대칭키를 사용한 디지털 서명 과정

개요

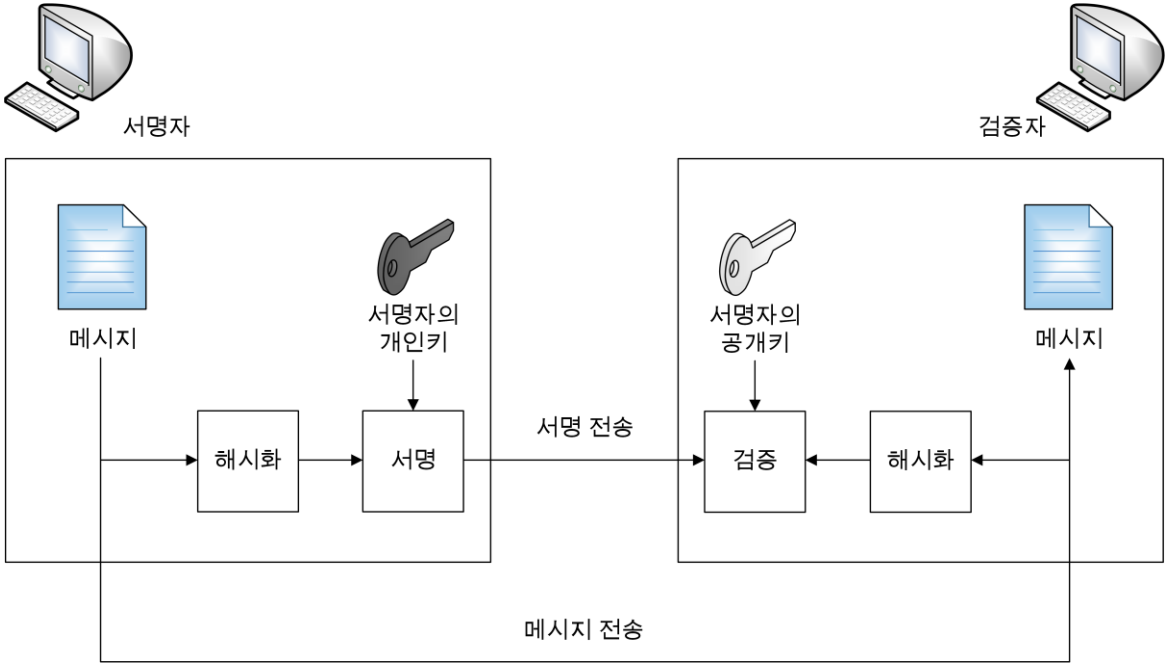
- 비대칭키 사용 (2/2)
  - 대칭키를 사용할 경우, 다음과 같은 문제가 발생할 수 있음
    - 키 관리 부담
      - 비밀키는 서명자와 검증자만 알고 있어야 하므로, 서명자는 검증자 수만큼의 비밀키를 생성하고 관리해야 함
    - 키 교환의 순환 문제
      - 대칭키를 교환하기 위해 디지털 서명을 사용할 경우, 해당 디지털 서명을 생성하기 위해 대칭키가 필요한 악순환이 발생할 수 있음
    - 검증자에 의한 서명 위조
      - 서명자와 검증자가 동일한 비밀키를 사용하고 있으므로, 검증자가 비밀키를 이용하여 서명을 하면 서명자가 보낸 서명인 것처럼 위장할 수 있음

개요

- 다이제스트 서명

메시지 다이제스트(Message Digest):
메시지에 해시 함수를 적용시켜 생성한 해시값

  - 비대칭키 암호시스템은 큰 연산 비용으로 인해 긴 메시지를 처리하기에 적합하지 않음
  - 해당 문제를 해결하기 위해 메시지 다이제스트에 서명하고 검증하는 방법을 이용하여 해결할 수 있음



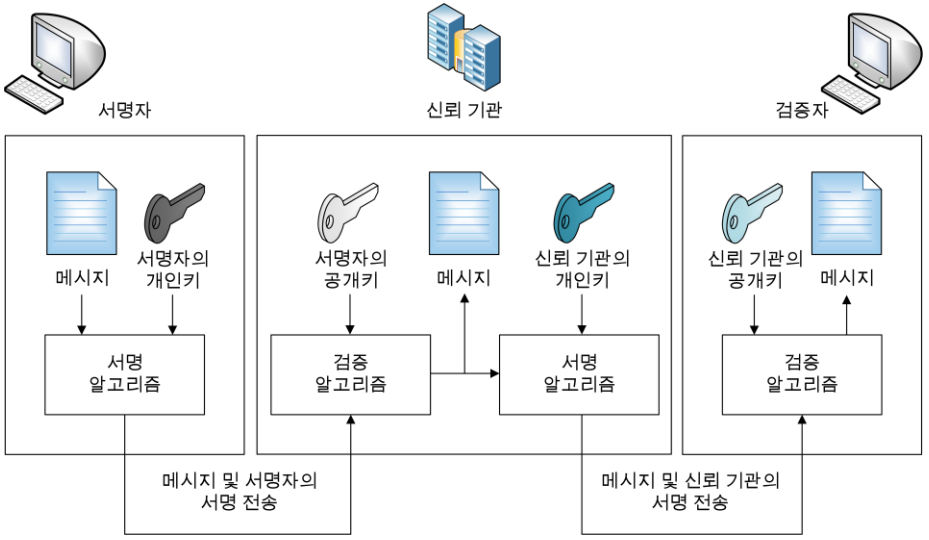
<그림 3> 다이제스트를 사용한 디지털 서명 과정

개요

- 디지털 서명의 보안 특성 (1/2)
  - 개요
    - 비대칭키 기반 구조의 특징으로 인해, 메시지 인증, 메시지 무결성, 부인봉쇄를 제공함
  - 메시지 인증
    - 서명자의 개인키를 이용하여 서명이 생성되고 서명자의 공개키를 이용해서 검증을 진행하므로, 검증자는 받은 메시지가 서명자에게서 온 메시지임을 신뢰할 수 있음
  - 메시지 무결성
    - 같은 개인키를 사용하더라도 메시지가 변경되면 생성되는 서명도 변경되므로, 검증자는 받은 메시지가 서명 이후 변경되지 않았음을 신뢰할 수 있음

개요

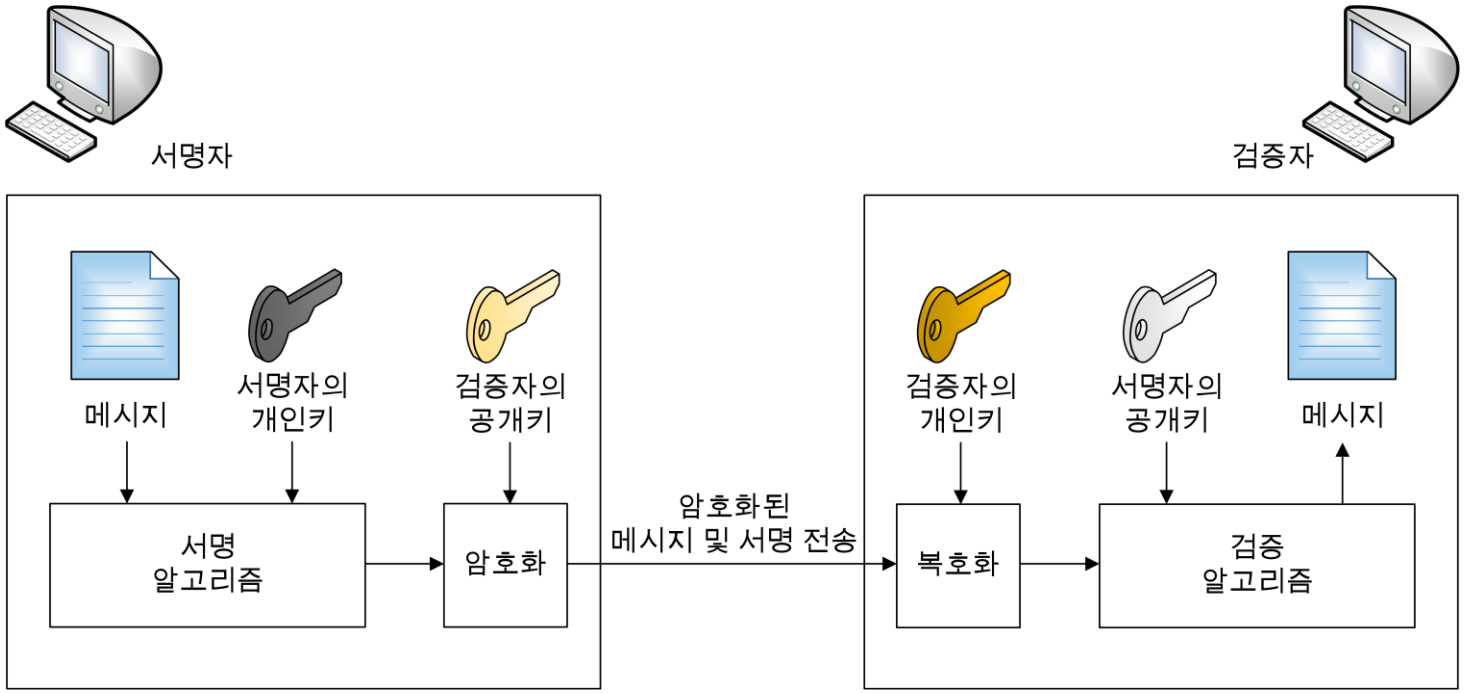
- 디지털 서명의 보안 특성 (2/2)
  - 부인봉쇄
    - 개인키로 생성된 서명은 대응되는 공개키로 검증 가능하므로, 검증에 성공한 서명에 대해 서명자는 서명 사실을 부인하기 어려움
    - 키 변경 및 탈취 등을 이유로 부인하는 경우, 타임스탬프, 신뢰 기관 등을 통해 부인봉쇄를 강화할 수 있음



<그림 4> 부인봉쇄를 위한 신뢰 기관 활용

개요

- 디지털 서명과 기밀성
  - 디지털 서명 자체가 기밀성을 보장해주지는 않음
  - 기밀성이 필요하면 메시지와 서명을 암호화하여 사용할 수 있음



<그림 5> 암호화를 통한 디지털 서명 구조의 기밀성 보장

개요

- 디지털 서명 공격
  - 키-단독 공격(Key-only Attack)
    - 공격자가 서명자의 공개키와 공개된 시스템 정보만을 이용하여 하는 공격
      - e.g., 위조된 서명과 공개키를 이용해서 위조된 메시지를 생성함
  - 알려진 메시지 공격(Known-message Attack)
    - 공격자가 서명 대상 메시지 선택에 관여하지 않은 한 개 이상의 공개된 메시지-서명 쌍을 이용하여 하는 공격
      - e.g., 공개된 여러 메시지-서명 쌍을 분석하여 새로운 메시지의 서명을 위조함
  - 선택 메시지 공격(Chosen-message Attack)
    - 공격자가 서명 받을 메시지를 직접 선택하고, 해당 메시지에 대한 서명을 획득하여 이용하는 공격
      - e.g., 공격자가 선택한 메시지들의 서명을 받은 뒤, 이를 이용하여 다른 메시지의 서명을 위조함

개요

- 디지털 서명 위조
  - 실존적 위조(Existential Forgery)
    - 공격자가 메시지를 원하는 대로 통제하지 못하더라도, 새로운 유효한 메시지-서명 쌍 하나를 생성하는 위조
    - 추가적인 공격에 활용되거나 우연히 악용 가능한 메시지를 획득할 수 있음
  - 선택적 위조(Selective Forgery)
    - 공격자가 특정 메시지를 선택하고, 해당 메시지에 대해 유효한 메시지-서명 쌍을 생성하는 위조
    - 해당 위조의 성공이 공격자가 개인키를 알아냈거나 모든 메시지를 위조 가능함을 의미하지는 않음

목 차

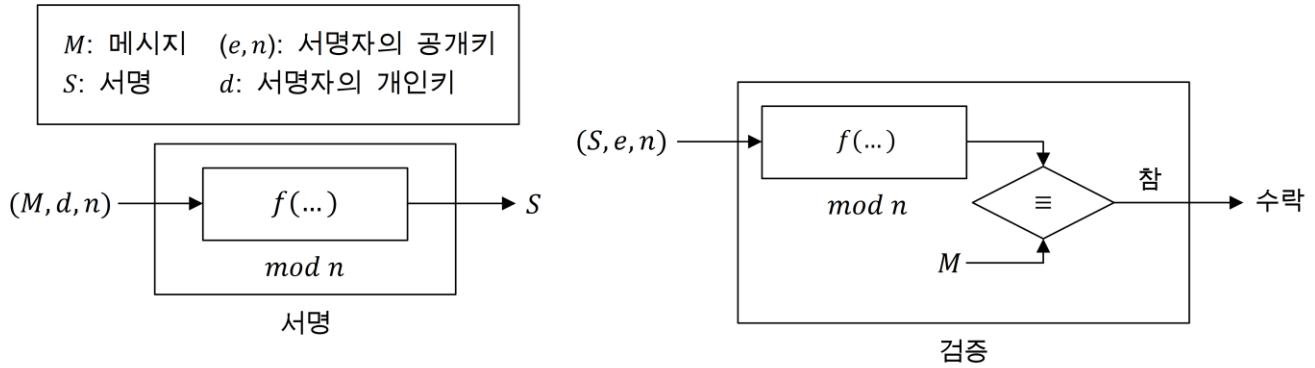
- 보충
- 개요
- 구조
- 변형 서명 구조

구조

- 개요
  - 디지털 서명 구조에 따라 사용하는 공개키 및 서명, 검증 알고리즘이 다름
- 종류
  - RSA Digital Signature Scheme
  - ElGamal Digital Signature Scheme
  - Schnorr Digital Signature Scheme
  - DSS (Digital Signature Standard)
  - ECDSS (Elliptic Curve Digital Signature Scheme)

구조

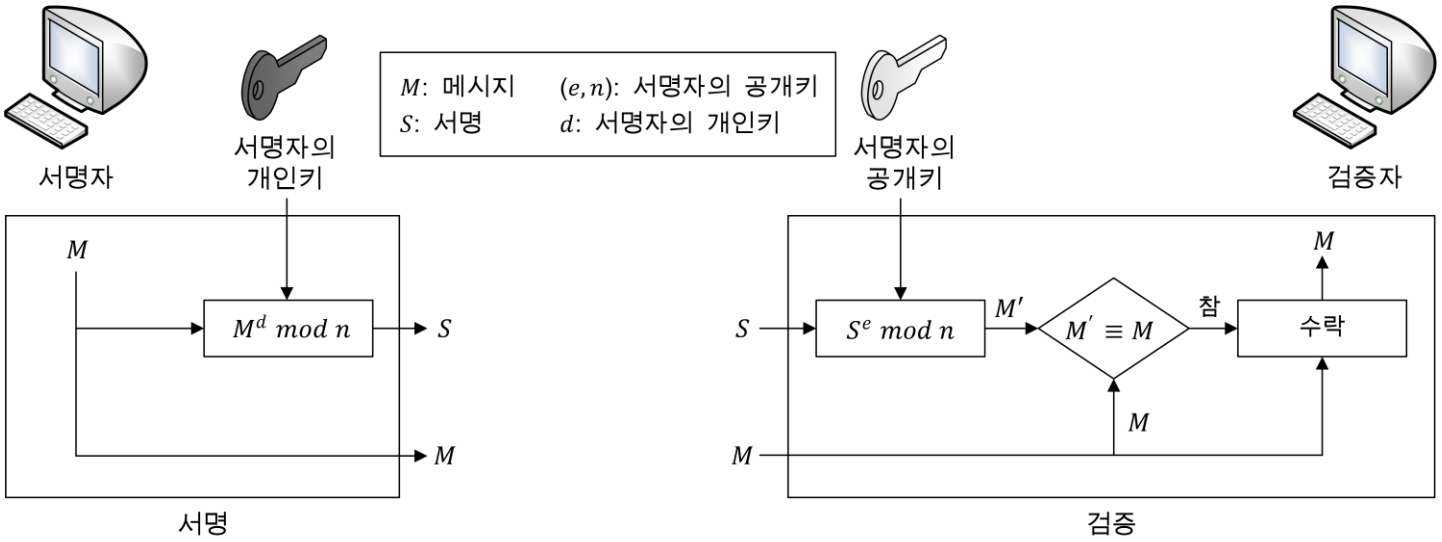
- RSA Digital Signature Scheme (1/8)
  - 개요
    - RSA 공개키 암호화를 이용하여 구현한 디지털 서명 구조로, 정수 인수분해 문제의 어려움을 기반으로 함
  - 키 생성
    1. 서명자는 두 소수 p, q를 선택하여 $n = p \times q, \phi(n) = (p-1)(q-1)$를 계산함
    2. 공개값 e를 설정하고 $e \times d = 1 \, mod \, \phi(n)$을 만족하는 d를 계산하여 개인키로 사용함



<그림 6> RSA 디지털 서명 구조의 개념도

구조

- RSA Digital Signature Scheme (2/8)
  - 서명 및 검증 (1/2)
    - 서명자는 자신의 개인키 d로 계산한 $S = M^d \ mod \ n$을 이용하여 메시지에 서명을 생성하고 이를 메시지와 함께 전송함
    - 검증자는 메시지와 서명을 수신하여 $M' = S^e \ mod \ n$을 계산하고, $M' \equiv M(mod \ n)$이면 메시지를 수락함



<그림 7> RSA 디지털 서명 구조

- RSA Digital Signature Scheme (3/8)
  - 서명 및 검증 (2/2)
    - 예제 13.1

서명자가 $p = 823, q = 953, e = 313$을 선택하고, 이를 통해 $n = 784319, \phi = 782544, d = 160009$를 계산하였다. 서명자가 $M = 19070$인 메시지를 송신하려고 할 때, 서명 및 검증 과정을 계산하시오.

- 서명
  - $S = (19070^{160009})\ mod\ 784319 = 210625\ mod\ 784319$

  $\therefore M = 19070, S = 210625$를 검증자에게 전송함

- 검증
  - $M' = 210625^{313}\ mod\ 784319 = 19070\ mod\ 784319$

  $\therefore M' = 19070, M = 19070$으로 $M' \equiv M$이므로, 검증자는 메시지를 수락함

구조

- RSA Digital Signature Scheme (4/8)
  - 발생 가능한 공격 (1/2)
    - 키-단독 공격
      - 공격자가 임의의 S를 선택한 후, $M = S^e \ mod \ n$과 공개키 (e, n)을 이용하여 S가 유효한 메시지 M을 생성할 수 있음
      - 공격자가 메시지를 선택하여 생성하는 것이 아니므로 실존적 위조에 그치게 됨

    - 알려진 메시지 공격
      - 공격자가 동일한 개인키로 서명된 $(M_1, S_1), (M_2, S_2)$를 가로챈 경우, 만약 $M = (M_1 \times M_2) \ mod \ n$이라면 RSA의 곱셈 성질(Multiplicative Property)에 의해 $S = (S_1 \times S_2) \ mod \ n$이 됨
      - 공격자가 (M, S)를 생성하여 서명자의 메시지와 서명으로 위장할 수 있으나, 해당 방식의 위조는 실존적 위조에 그치고, 원하는 메시지를 선택하여 생성하기 어려움

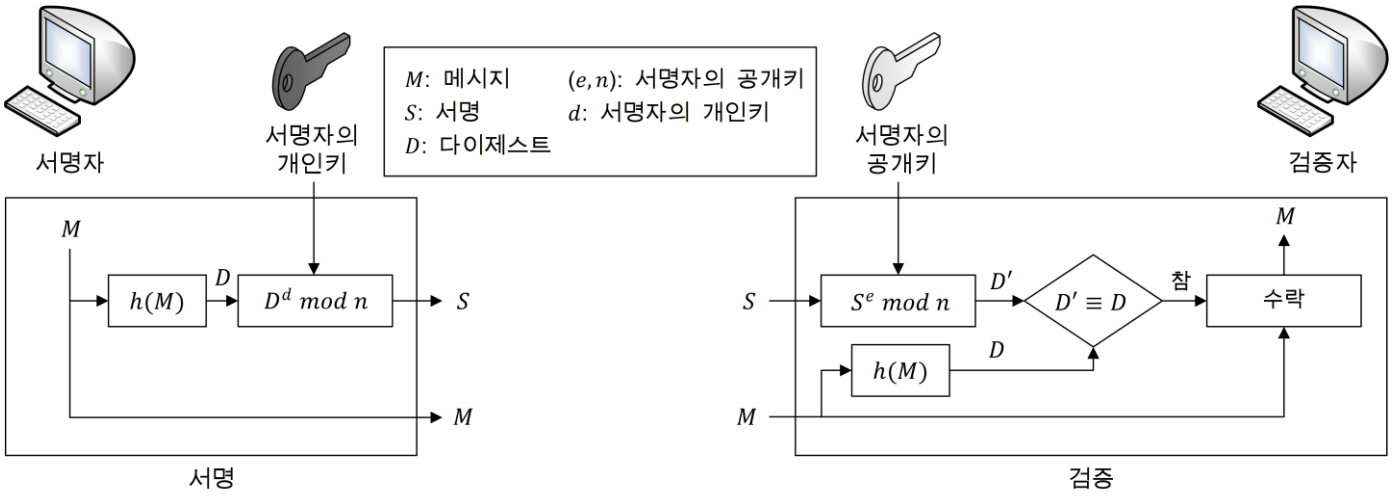
곱셈 성질(Multiplicative Property):
두 입력의 곱에 대한 연산 결과가 각 입력의 연산 결과를 곱한 값과 일치하는 성질

구조

- RSA Digital Signature Scheme (5/8)
  - 발생 가능한 공격 (2/2)
    - 선택 메시지 공격
      - 공격자가 서명자에게 M_1, M_2에 대한 서명을 요청한 후, 곱셈 성질을 이용하여 $M = M_1 \times M_2$를 생성하고 서명자가 M에 서명한 것이라고 주장할 수 있음
      - 해당 공격의 경우, 임의의 M_1을 선택하고 $M_2 = M \times M_1^{-1} \; mod \; n$으로 하면 $M_1 \times M_2 = M \; mod \; n$이므로, 공격자가 원하는 메시지를 선택하여 서명을 생성할 수 있게 됨
      - 이러한 문제를 해결하기 위해 메시지 다이제스트에 서명을 진행할 수 있음

구조

- RSA Digital Signature Scheme (6/8)
  - 메시지 다이제스트 서명 (1/3)
    - 서명자가 사전에 합의된 해시 함수를 이용하여 다이제스트 $D = h(M)$을 생성하고, 해당 다이제스트에 기존과 동일한 서명 알고리즘으로 서명을 진행할 수 있음
    - 검증자는 기존과 동일한 검증 알고리즘으로 메시지 대신 다이제스트를 검증하여 서명을 검증할 수 있음



<그림 8> 메시지 다이제스트에 대한 RSA 디지털 서명

- RSA Digital Signature Scheme (7/8)
  - 메시지 다이제스트 서명 (2/3)
    - 발생 가능한 공격
      - 키-단독 공격
        - 공격자가 임의의 서명 S가 유효한 다이제스트 D를 생성하고 $D = h(M)$을 만족하는 M을 찾아내어 (M, S)를 얻어낼 수 있음
          - 사용하는 해시 함수가 강한 역상 저항성을 가진다면 공격이 어려움
      - 알려진 메시지 공격
        - 공격자가 동일한 개인키로 생성된 $(M_1, S_1), (M_2, S_2)$를 가지고 있을 때, $S \equiv S_1 \times S_2, h(M) \equiv h(M_1) \times h(M_2)$를 통해 M을 찾아내기 위한 시도를 할 수 있음
          - 사용하는 해시 함수가 강한 역상 저항성을 가진다면 공격이 어려움
        - 공격자가 (M, S)를 가로챈 후, $h(M) = h(M')$를 만족하는 M'을 생성하기 위한 시도를 할 수 있음
          - 사용하는 해시 함수가 강한 제 2 역상 저항성을 가진다면 공격이 어려움

역상 저항성(Preimage Resistance):
해시값으로부터 원래 메시지를 찾기 어려운 성질

제 2 역상 저항성(Second-Preimage Resistance):
주어진 메시지와 동일한 해시값을 갖는 다른 메시지를 찾기 어려운 성질

- RSA Digital Signature Scheme (8/8)
  - 메시지 다이제스트 서명 (3/3)
    - 발생 가능한 공격
      - 선택 메시지 공격
        - 공격자가 서명자에게 요청하여 M_1, M_2에 대한 S_1, S_2를 얻어낸 후, $S \equiv S_1 \times S_2$, $h(M) \equiv h(M_1) \times h(M_2)$를 통해 M을 찾아내기 위한 시도를 할 수 있음
          - 사용하는 해시 함수가 강한 역상 저항성을 가진다면 공격이 어려움
        - 공격자가 $h(M) = h(M')$를 만족하는 M, M'을 만들고, 서명자에게 M에 서명하게 하여 S를 얻어낸 후, (M', S)로 검증을 통과할 수 있음
          - 사용하는 해시 함수가 강한 충돌 저항성을 가진다면 공격이 어려움
    - 해시 함수 강도 의존성
      - 메시지 다이제스트에 서명할 경우, 해당 서명에 대한 공격의 난이도는 사용하는 해시 함수의 강도에 의존함

충돌 저항성(Collision Resistance):
동일한 해시값을 갖는 서로 다른 두 메시지를 찾기 어려운 성질

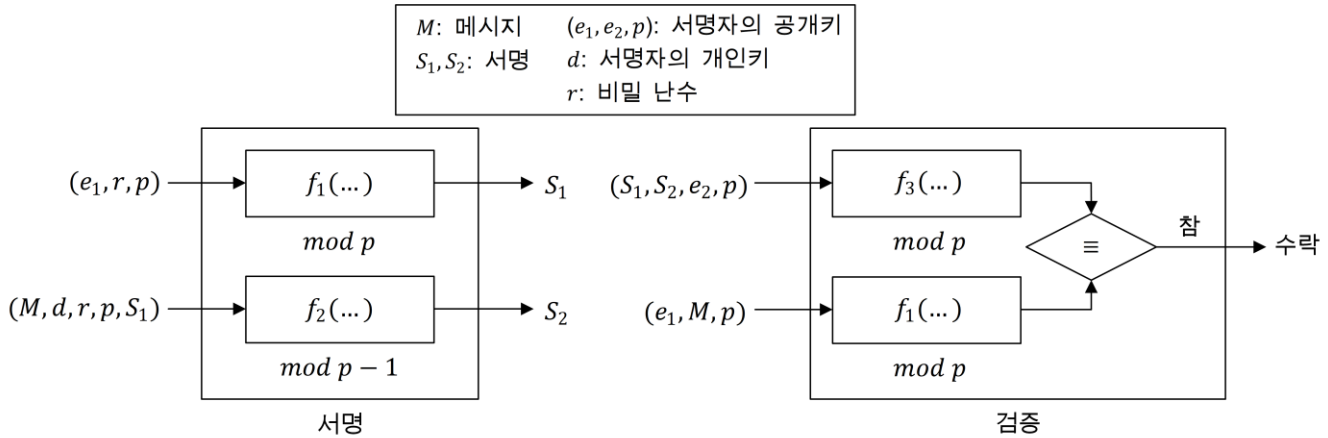
구조

- ElGamal Digital Signature Scheme (1/6)
  - 개요
    - ElGamal 암호화 알고리즘을 이용한 디지털 서명 구조로, 서명할 때마다 새로운 비밀 난수를 사용함

군(Group):
하나의 연산에 대해 닫힘성, 결합법칙, 항등원 및 역원 조건을 만족하는 집합

  - 키 생성
    1. 소수 p와 곱셈군 $Z_p{}^*$ 내에서의 원시근 e_1를 선택하고, $p-1$보다 작은 개인키 d를 선택함
    2. 공개값 $e_2 = {e_1}^d$를 계산하여 (e_1, e_2, p)를 공개키로 사용함

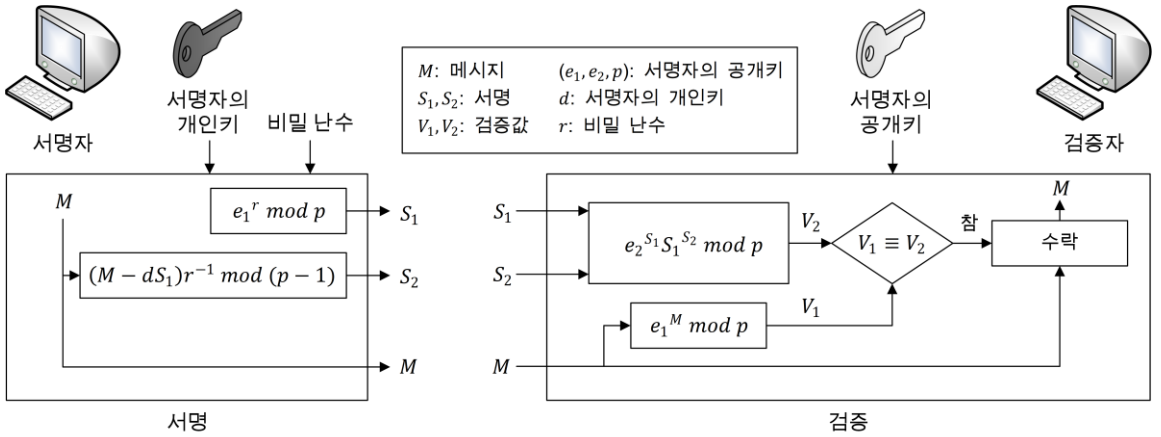
원시근(Primitive Root):
거듭제곱을 통해 곱셈군의 모든 원소를 생성할 수 있는 원소



<그림 9> ElGamal 디지털 서명 구조의 개념도

구조

- ElGamal Digital Signature Scheme (2/6)
  - 서명 및 검증 (1/3)
    - 서명자는 매 서명마다 비밀 난수 r을 선택하고 $S_1 = {e_1}^r \ mod\ p, S_2 = (M - d \times S_1) \times r^{-1}\ mod\ (p-1)$를 계산하여 서명을 생성하고 메시지와 서명을 전송함
    - 검증자는 $0 < S_1 < p, 0 < S_2 < p-1$인지 확인하고, $V_1 = {e_1}^M\ mod\ p, V_2 = {e_2}^{S_1} \times {S_1}^{S_2}\ mod\ p$를 계산한 후, $V_1 \equiv V_2$이면 메시지를 수락함



<그림 10> ElGamal 디지털 서명 구조

- ElGamal Digital Signature Scheme (3/6)
  - 서명 및 검증 (2/3)
    - 예제 13.2

서명자가 $p = 3119, e_1 = 2, d = 127, r = 307$을 선택하고, 이를 통해 $e_2 = 2^{127} \, mod \, 3119 = 1702$를 계산하였다. 서명자가 $M = 320$인 메시지를 송신하려고 할 때, 서명 및 검증 과정을 계산하시오.

- 서명
  - $S_1 = {e_1}^r = 2^{307} = 2083 \, mod \, 3119$
  - $S_2 = (M - d \times S_1) \times r^{-1} = (320 - 127 \times 2083) \times 307^{-1} = 2105 \, mod \, 3118$

  $\therefore M = 320, S_1 = 2083, S_2 = 2105$를 검증자에게 전송함

- 검증
  - $V_1 = {e_1}^M = 2^{320} = 3006 \, mod \, 3119$
  - $V_2 = {e_2}^{S_1} \times {S_1}^{S_2} = 1702^{2803} \times 2083^{2105} = 3006 \, mod \, 3119$

  $\therefore V_1 = 3006, V_2 = 3006$으로 $V_1 \equiv V_2$이므로, 검증자는 메시지를 수락함

- ElGamal Digital Signature Scheme (4/6)
  - 서명 및 검증 (3/3)
    - 예제 13.3

서명자가 $p = 3119, e_1 = 2, d = 127, r = 107$을 선택하고, 이를 통해 $e_2 = 2^{127} \, mod \, 3119 = 1702$를 계산하였다. 서명자가 $M = 3000$인 메시지를 송신하려고 할 때, 서명 및 검증 과정을 계산하시오.

- 서명
  - $S_1 = {e_1}^r = 2^{107} = 2732 \, mod \, 3119$
  - $S_2 = (M - d \times S_1) \times r^{-1} = (3000 - 127 \times 2732) \times 107^{-1} = 2526 \, mod \, 3118$

  $\therefore M = 3000, S_1 = 2732, S_2 = 2526$을 검증자에게 전송함

- 검증
  - $V_1 = {e_1}^M = 2^{3000} = 704 \, mod \, 3119$
  - $V_2 = {e_2}^{S_1} \times {S_1}^{S_2} = 1702^{2732} \times 2732^{2526} = 704 \, mod \, 3119$

  $\therefore V_1 = 704, V_2 = 704$으로 $V_1 \equiv V_2$이므로, 검증자는 메시지를 수락함

구조

- ElGamal Digital Signature Scheme (5/6)
  - 발생 가능한 공격 (1/2)
    - 키-단독 공격
      - 공격자가 특정 M에 대한 서명을 위조하려는 경우, 임의의 S_1을 선택하고, 그에 대한 S_2를 얻기 위해서는 $S_2 = \log_{S_1}({e_1}^M {e_2}^{-S_1})$를 계산해야 하며, 이는 이산대수 문제로, 복잡한 연산을 필요로 함
      - 공격자가 임의의 x, y를 정하고, $M = xS_2 \ mod\ (p-1), S_1 = -yS_2 \ mod\ (p-1), S_1 = e_1^x e_2^y \ mod\ p$를 만족하는 (M, S_1, S_2)를 계산할 수 있으나, 해당 방식의 위조는 실존적 위조가 됨
    - 알려진 메시지 공격
      - 공격자가 (M, S_1, S_2)를 가로챈 후, S_1, S_2가 유효한 M'를 찾기 위해 ${e_1}^{M'} \equiv {e_1}^M \ mod\ p$를 만족하는 M'을 찾을 수 있으나, 해당 방식의 위조는 실존적 위조가 됨

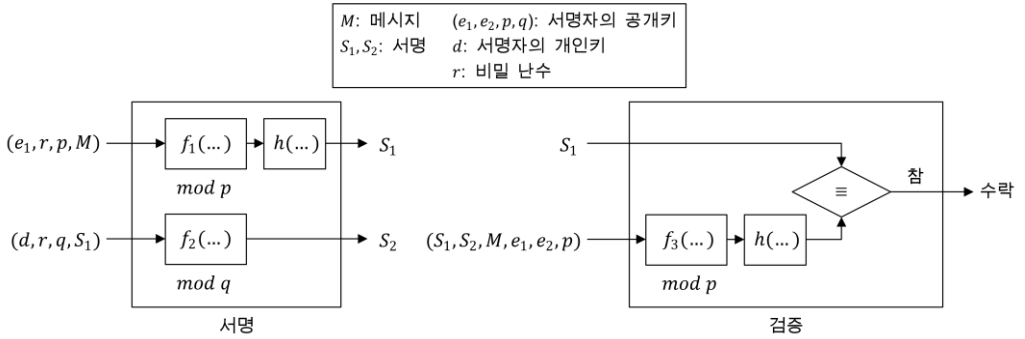
이산대수 문제(Discrete Logarithm Problem):
g와 y가 주어졌을 때 $g^x = y$를 만족하는 지수 x를 찾는 문제

구조

- ElGamal Digital Signature Scheme (6/6)
  - 발생 가능한 공격 (2/2)
    - 선택 메시지 공격
      - 공격자가 선택한 두 메시지 M_1, M_2에 동일한 비밀 난수 r이 사용된 서명 $(S_{1,1}, S_{1,2}), (S_{2,1}, S_{2,2})$을 얻으면, 두 서명식의 차이 $(S_{2,2} - S_{2,1})r \equiv M_1 - M_2 \ (mod\ p-1)$를 이용하여 r을 구할 수 있음
      - 구한 r과 기존 서명을 이용하여 공격자가 원하는 메시지에 대한 유효한 서명을 생성할 수 있으므로 선택적 위조가 가능함
    - 비밀 난수 의존성
      - ElGamal 디지털 서명에 대한 공격의 난이도는 비밀 난수 r의 생성 및 관리 안전성에 의존함
      - 비밀 난수 r이 재사용되거나 예측되면 개인키가 유출되어 선택적 위조가 가능해질 수 있음

구조

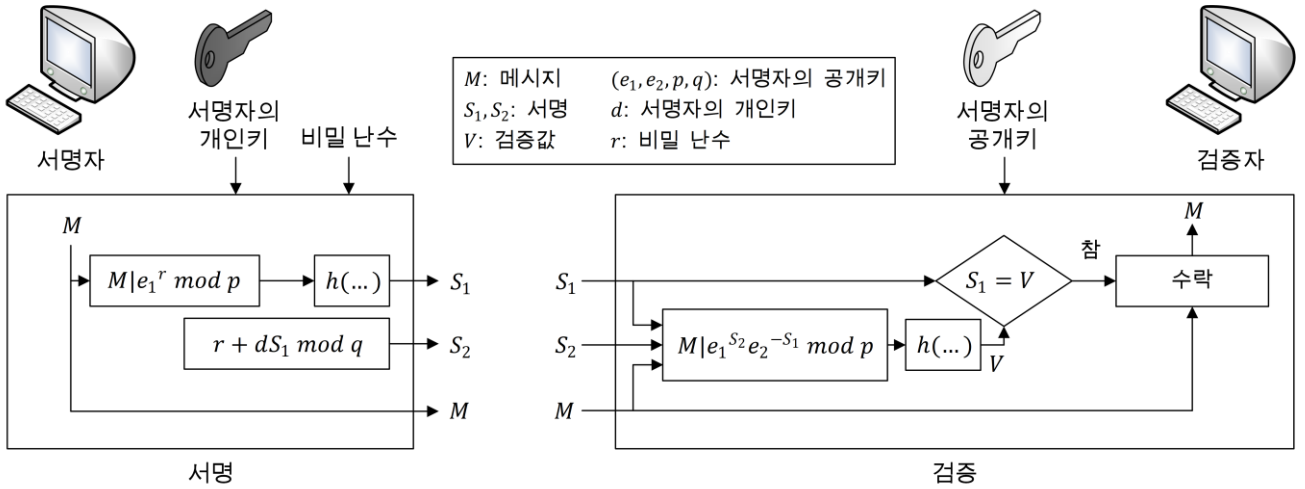
- Schnorr Digital Signature Scheme (1/4)
  - 개요
    - Schnorr 프로토콜에 기반한 디지털 서명 구조로, ElGamal 계열의 서명을 단순화하고 서명 크기를 줄이도록 설계됨
  - 키 생성
    1. 소수 p와 $p-1$의 약수인 소수 q를 선택한 후, $Z_p{}^*$의 원시근 e_0를 통해 $e_1 = e_0{}^{(p-1)/q} \; mod \; p$를 구함
    2. 개인키 $d \not\equiv 0 \; mod \; q$를 선택하고 $e_2 = e_1{}^d \; mod \; p$를 계산하여 (e_1, e_2, p, q)를 공개키로 사용함



<그림 11> Schnorr 디지털 서명 구조의 개념도

구조

- Schnorr Digital Signature Scheme (2/4)
  - 서명 및 검증 (1/2)
    - 서명자는 매 서명마다 비밀 난수 r을 선택하고 $S_1 = h(M|{e_1}^r \ mod\ p), S_2 = r + d \times S_1 \ mod\ q$를 계산하여 서명을 생성하고 메시지와 서명을 전송함
    - 검증자는 $V = h(M|{e_1}^{S_2} {e_2}^{-S_1} \ mod\ p)$를 계산하고 $S_1 = V$이면 메시지를 수락함



<그림 12> Schnorr 디지털 서명 구조

- Schnorr Digital Signature Scheme (3/4)
  - 서명 및 검증 (2/2)
    - 예제 13.4

서명자가 $p = 2267, q = 103, e_0 = 2, d = 30$을 선택하고, 이를 통해 $e_1 = 2^{22} \; mod \; 2267 = 354, e_2 = 354^{30} \; mod \; 2267 = 1206$을 계산하였다. 서명자가 $M = 1000$인 메시지를 송신하려고 하며, $r = 11, h(1000630) = 200$이라고 할 때, 서명 및 검증 과정을 계산하시오.

- 서명
  - ${e_1}^r = 354^{11} = 630 \; mod \; 2267, S_1 = h(M|{e_1}^r \; mod \; p) = h(1000630) = 200$
  - $S_2 = r + d \times S_1 \; mod \; q = (11 + 30 \times 200) \; mod \; 103 = 6011 \; mod \; 103$
    $= 37 \; mod \; 103$

  $\therefore M = 1000, S_1 = 200, S_2 = 37$을 검증자에게 전송함

- 검증
  - $V = h(M|{e_1}^{S_2} {e_2}^{-S_1} \; mod \; p) = h(M|354^{37} \times 1206^{-200} \; mod \; 2267)$
    $= h(M|630 \; mod \; 2267) = h(1000|630) = h(1000630) = 200$

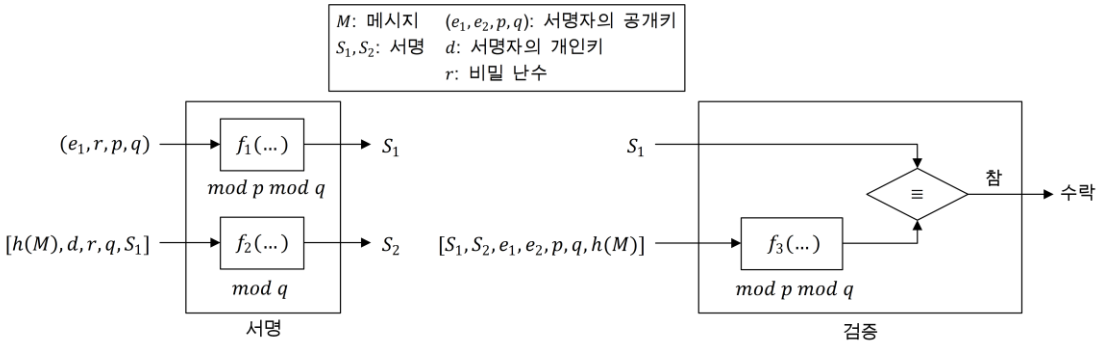
  $\therefore S_1 = 200, V = 200$으로 $S_1 = V$이므로 검증자는 메시지를 수락함

구조

- Schnorr Digital Signature Scheme (4/4)
  - ElGamal 디지털 서명과의 비교
    - ElGamal 디지털 서명 구조와 같이 이산대수 문제에 기반하지만, 메시지와 비밀 난수로 만든 값을 함께 해시하여 위조가 더 어려움
      - $V = h(M|{e_1}^{S_2} {e_2}^{-S_1} \ mod \ p)$ 형태로 인하여, 공격자가 임의의 S_1, S_2에 대한 M을 찾으려고 해도 사용하는 해시 함수가 강한 역상 저항성을 가지면 공격이 어려움
      - $S_1 = h(M|{e_1}^{r} \ mod \ p)$ 형태로 인하여, 공격자가 M를 찾기 위해서는 비밀 난수 r까지 고려해야 하므로, 위조가 어려움

구조

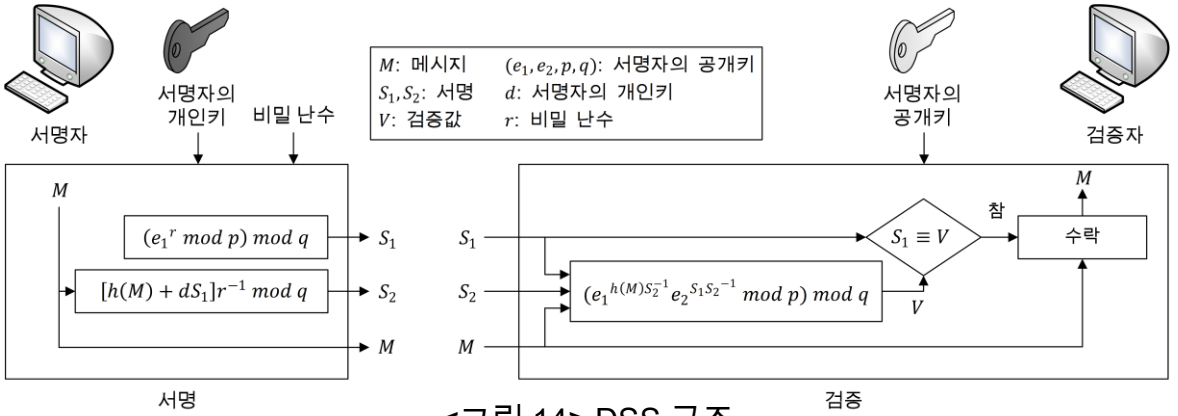
- DSS (1/5)
  - 개요
    - ElGamal 서명을 표준화한 디지털 서명 구조로, DSA(Digital Signature Algorithm)를 이용함
  - 키 생성
    1. 소수 p와 $p-1$의 약수인 소수 q를 선택한 후, $Z_p{}^*$의 원시근 e_0를 통해 $e_1 = e_0{}^{(p-1)/q} \; mod \; p$를 구함
    2. 개인키 $d \not\equiv 0 \; mod \; q$를 선택하고 $e_2 = e_1{}^d \; mod \; p$를 계산하여 (e_1, e_2, p, q)를 공개키로 사용함



<그림 13> DSS의 개념도

구조

- DSS (2/5)
  - 서명 및 검증 (1/2)
    - 서명자는 매 서명마다 비밀 난수 r을 선택하고 $S_1 = (e_1{}^r \bmod p) \bmod q, S_2 = [h(M) + dS_1]r^{-1} \bmod q$를 계산하여 서명을 생성하고 메시지와 서명을 전송함
    - 검증자는 $0 < S_1 < q, 0 < S_2 < q$인지 확인하고, $V = \left[\left(e_1{}^{h(M)S_2{}^{-1}} e_2{}^{S_1 S_2{}^{-1}}\right) \bmod p\right] \bmod q$를 계산한 후, $S_1 \equiv V$이면 메시지를 수락함



<그림 14> DSS 구조

- DSS (3/5)
  - 서명 및 검증 (2/2)
    - 예제 13.5

서명자가 $p = 8081, q = 101, e_0 = 3, d = 61$을 선택하고, 이를 통해 $e_1 = 3^{80} \ mod \ 8081 = 6968, e_2 = 6968^{61} \ mod \ 8081 = 2038$을 계산하였다. $r = 61, h(M) = 5000$이라고 할 때, 서명 및 검증 과정을 계산하시오.

- 서명
  - $S_1 = ({e_1}^r \ mod \ p) \ mod \ q = (6968^{61} \ mod \ 8081) \ mod \ 101 = 18$
  - $S_2 = [h(M) + dS_1]r^{-1} \ mod \ q = (5000 + 61 \times 18) \times 61^{-1} \ mod \ 101$
    $= 2014 \ mod \ 101 = 95$

  $\therefore M, S_1 = 18, S_2 = 95$를 검증자에게 전송함

- 검증
  - ${S_2}^{-1} = 84 \ mod \ 101 (\because 95 \times 84 \ mod \ 101 = 7980 \ mod \ 101 = 1 \ mod \ 101)$
  - $V = [(6968^{5000 \times 84} \times 2038^{18 \times 84}) \ mod \ 8081] \ mod \ 101 = 18$
  - $\therefore S_1 = 18, V = 18$로 $S_1 \equiv V$이므로, 검증자는 메시지를 수락함

구조

- DSS (4/5)
  - RSA 디지털 서명과의 비교
    - 서명에 큰 개인키 d로 지수 연산을 하는 RSA 디지털 서명 구조와 다르게, 서명에 비교적 작은 r을 이용하여 동일한 모듈러스를 기준으로 서명 계산이 더 빠름
    - 다만, RSA 디지털 서명 구조는 e를 비교적 작게 설정할 수 있기 때문에, 해당 상황에서는 DSS의 검증 계산이 더 느릴 수 있음
  - ElGamal 디지털 서명과의 비교
    - 모듈러스로 $p, p-1$을 사용하여 $2|p|$에 가까운 서명 길이를 가지는 ElGamal 디지털 서명 구조와 다르게, $p-1$의 약수로 설정하는 q를 모듈러스로 사용하여 더 짧은 서명 길이를 가짐

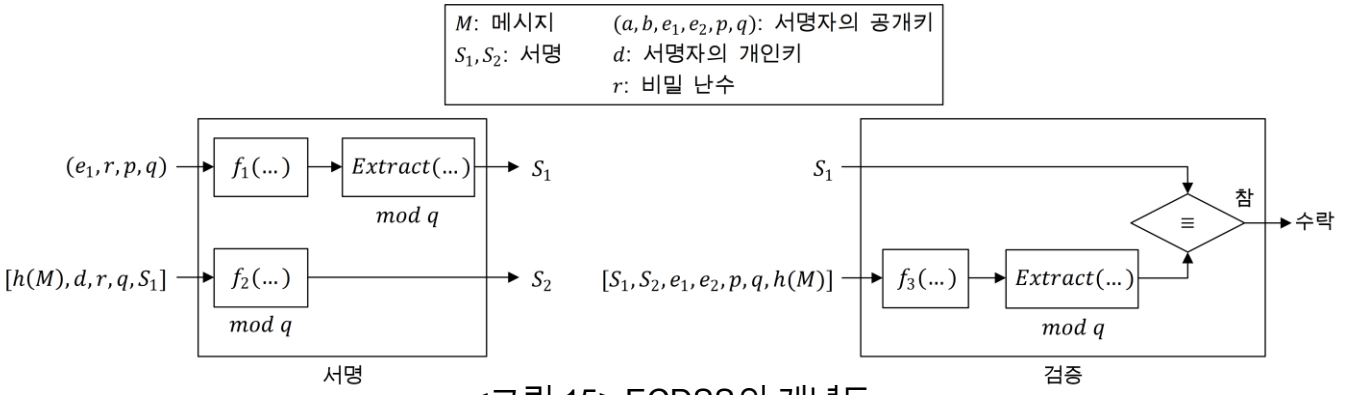
구조

- DSS (5/5)
  - ElGamal 계열 디지털 서명 구조의 비교
    - ElGamal 서명 구조는 비밀 난수와 개인키의 결합을 통해 개인키 추출 및 서명 위조를 어렵게 함
    - Schnorr 서명 구조는 ElGamal 서명 구조를 기반으로 메시지와 임시값을 함께 해시하여 임의의 서명값에 맞는 메시지를 생성하는 위조를 어렵게 함
    - DSS는 ElGamal 서명 구조를 $mod\ q$ 연산 중심으로 재구성하여 각 서명을 q 크기로 압축함

구분	서명 길이	서명 생성 방식	특징
ElGamal	약 $2\|p\|$	메시지, 개인키, 비밀 난수를 결합함	서명값 두 개를 모두 Z_p 기반 연산으로 생성함
Schnorr	약 해시값 길이 $+\|q\|$	메시지와 임시값을 함께 해시하고, 해시값을 개인키 및 비밀 난수와 결합함	해시값 및 q 범위의 서명값으로 서명을 구성함
DSS	약 $2\|q\|$	임시값을 $mod\ q$로 축소하고, 다이제스트와 개인키를 비밀 난수와 결합함	두 서명값 길이를 모두 q 범위로 생성함

구조

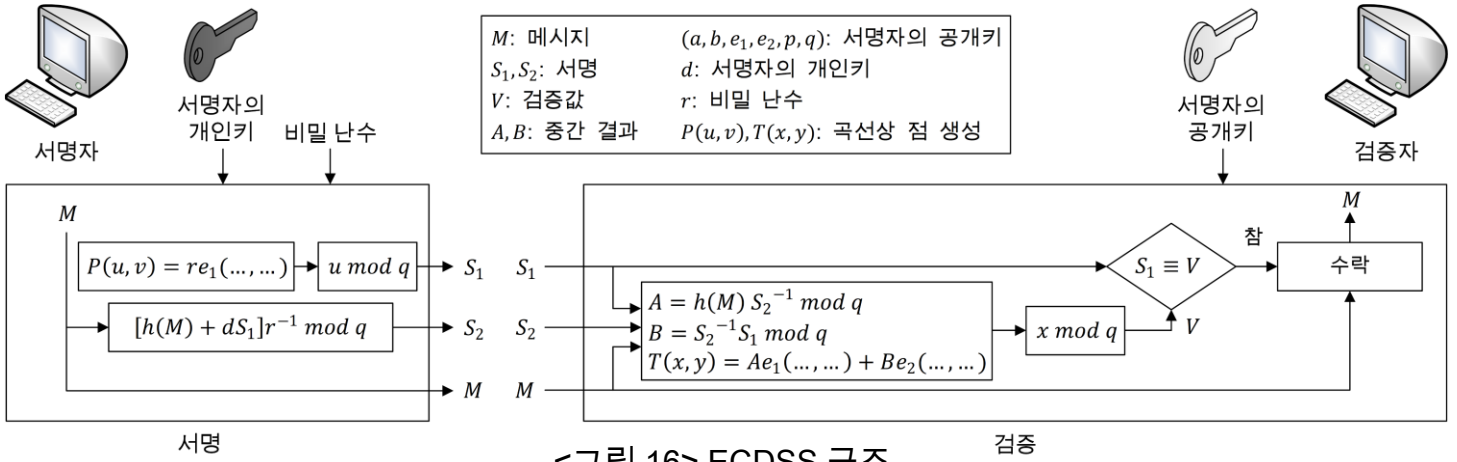
- ECDSS (1/3)
  - 개요
    - 타원곡선 암호를 이용한 디지털 서명 구조로, DSA를 타원 곡선 위에서 구현한 ECDSA(Elliptic Curve DSA)가 사용됨
  - 키 생성
    1. 소수 p에 대해 타원곡선 $E_p(a,b) = y^2 \equiv x^3 + ax + b$(단, $4a^3 + 27b^2 \not\equiv 0 \; mod \; p$)와 소수 q, 개인키 d를 선택함
    2. 곡선 상의 한 점 $e_1(\dots,\dots)$에 대해 $e_2 = de_1(\dots,\dots)$를 계산하고, (a,b,p,q,e_1,e_2)를 공개키로 사용함



<그림 15> ECDSS의 개념도

구조

- ECDSS (2/3)
  - 서명 및 검증 (1/2)
    - 서명자는 매 서명마다 비밀 난수 r을 선택하여 곡선 상의 점 $P(u,v) = r \times e_1(\dots,\dots)$를 계산한 후, $S_1 = u \bmod q, S_2 = [h(M) + dS_1]r^{-1} \bmod q$를 통해 서명을 생성함
    - 검증자는 $A = h(M){S_2}^{-1} \bmod q, B = {S_2}^{-1}S_1 \bmod q$를 계산하여 $T(x,y) = A \times e_1(\dots,\dots) + B \times e_2(\dots,\dots)$를 재구성하고, $V = x \bmod q$에 대해 $S_1 \equiv V$이면 메시지를 수락함



<그림 16> ECDSS 구조

구조

- ECDSS (3/3)
  - 서명 및 검증 (2/2)
    - 예제 13.6

서명자가 $a = 2, b = 2, p = 17, q = 19, e_1 = (5,1), d = 7$을 선택하고, 이를 통해 $E_p(2,2) = y^2 \equiv x^3 + 2x + 2\ (mod\ 17), e_2 = (0,6)$을 계산하였다. $r = 3, h(M) = 11$이라고 할 때, 서명 및 검증 과정을 계산하시오.

- 서명
  - $P(u, v) = 3 \times e_1(5,1) = (10,6), S_1 = u\ mod\ 19 = 10\ mod\ 19 = 10$
  - $S_2 = [h(M) + dS_1]r^{-1}\ mod\ q = (11 + 7 \times 10) \times 13\ mod\ 19 = 8$
  - $\therefore M, S_1 = 10, S_2 = 8$을 검증자에게 전송함
- 검증
  - $A = h(M){S_2}^{-1}\ mod\ q = 11 \times 12\ mod\ 19 = 18,$
    $B = {S_2}^{-1}S_1\ mod\ q = 12 \times 10\ mod\ 19 = 6$
  - $T(x, y) = Ae_1(\ldots, \ldots) + Be_2(\ldots, \ldots) = 18e_1 + 6e_2 = 18e_1 + 6(7e_1) = 60e_1$
    $= (10,6), V = x\ mod\ q = 10\ mod\ 19 = 10$
  - $\therefore S_1 = 10, V = 10$으로 $S_1 \equiv V$이므로, 검증자는 메시지를 수락함

구조

- 정리

체(Field):
덧셈, 뺄셈, 곱셈이 가능하고, 0이 아닌 원소에 대해 나눗셈이 가능한 대수적 구조

  - RSA, ElGamal, Schnorr, DSS, ECDSS는 기반이 되는 수학적 문제와 서명 및 검증 방식에서 차이가 있음
  - RSA는 정수 인수분해 문제를 기반으로 하며, ElGamal, Schnorr, DSS는 유한체의 이산 대수 문제, ECDSS는 타원곡선의 이산 대수 문제를 기반으로 함

서명 구조	기반 문제	특징
RSA	정수 인수분해 문제	개인키의 지수 연산으로 서명하고 공개키의 지수 연산으로 검증함
ElGamal	이산대수 문제	비밀 난수를 이용한 연산으로 서명하고 서명값의 지수 연산으로 검증함
Schnorr	이산대수 문제	ElGamal 계열의 서명 구조에서 메시지와의 결합과 해시 함수를 이용하여 서명과 검증을 진행함
DSS	이산대수 문제	ElGamal 계열의 서명 구조에서 두 서명값을 q 범위로 축소함
ECDSS	타원곡선 이산대수 문제	DSS 계열의 서명 구조를 타원곡선 점 연산으로 구현함

[표 2] 디지털 서명 구조 비교

목 차

- 보충
- 개요
- 구조
- 변형 서명 구조

변형 서명 구조

- 개요
  - 상황에 따라 기존의 디지털 서명 구조를 변형하거나 새로운 요소를 추가하여 사용할 수 있음
- 종류
  - Timestamped Digital Signature Scheme
  - Blind Digital Signature Scheme
  - Undeniable Digital Signature Scheme

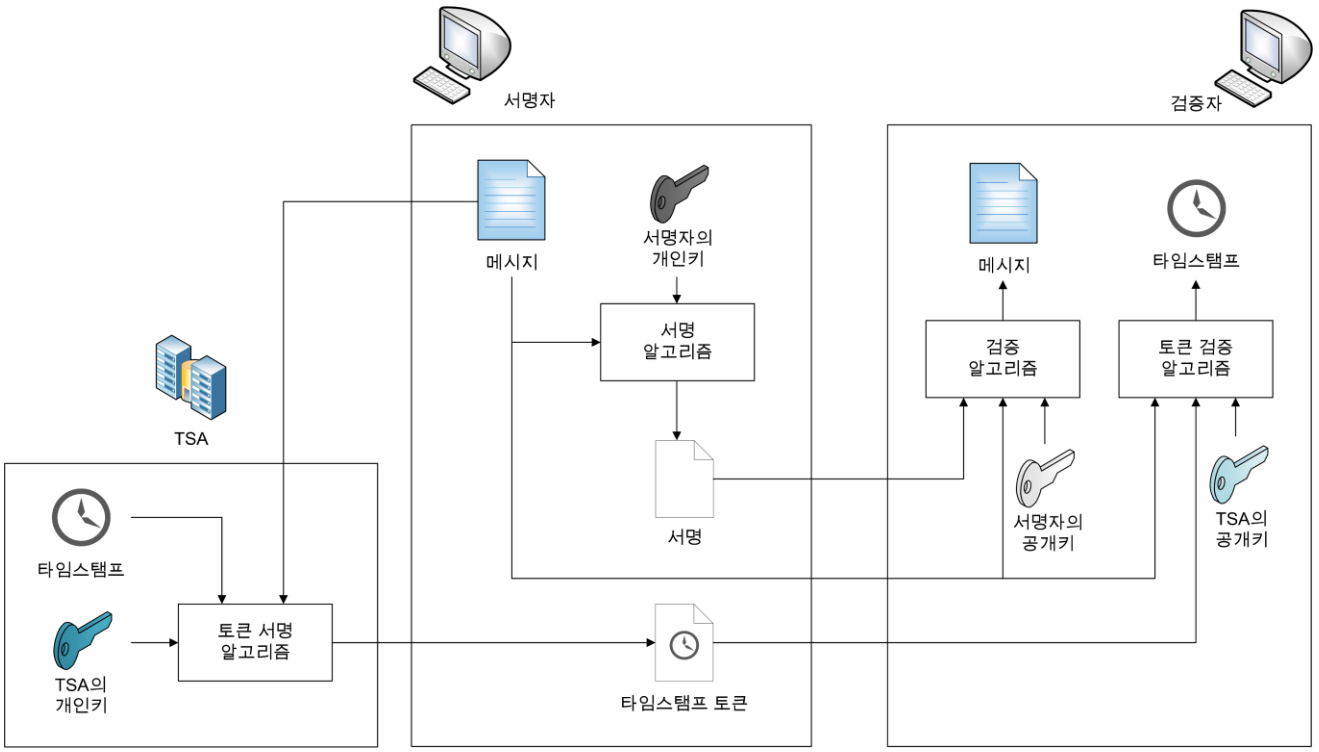
변형 서명 구조

- Timestamped Digital Signature Scheme (1/2)
  - 개요
    - 서명 또는 메시지가 특정 시점에 존재했음을 증명하기 위해 시간 정보를 서명에 결합하는 방식
  - 목적
    - 서명이 특정 시점에 존재했음을 증명함
    - 메시지가 특정 시점에 존재했음을 증명함
  - 과정
    - 서명자가 메시지에 타임스탬프를 추가하여 서명을 진행하고 메시지, 서명, 타임스탬프를 함께 전송함
    - 검증자는 메시지에 타임스탬프를 추가한 값에 대해 서명을 검증하여 타임스탬프의 유효성을 확인함

변형 서명 구조

- Timestamped Digital Signature Scheme (2/2)
  - 구현
    - 신뢰성을 강화하기 위하여 TSA (Time Stamping Authority) 같은 신뢰 기관이나 비표(Nonce)를 사용할 수 있음

비표(Nonce):
재사용을 방지하기 위해 한 번만 사용하는 임의의 값



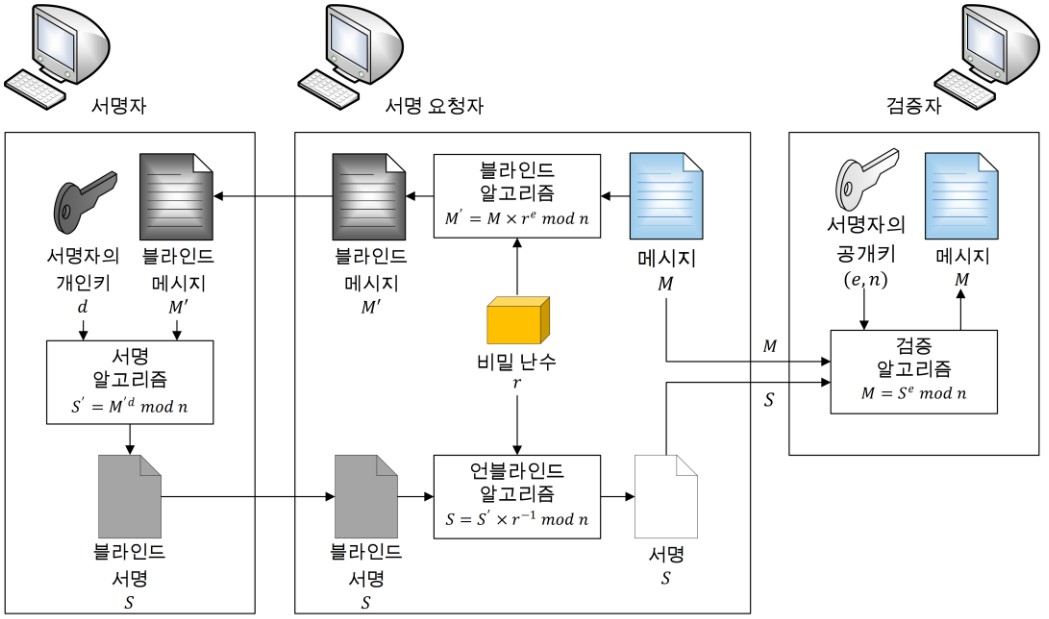
<그림 17> Timestamped Digital Signature Scheme 개념도

변형 서명 구조

- Blind Digital Signature Scheme (1/2)
  - 개요
    - 메시지 내용을 서명자에게 노출하지 않은 상태에서 서명을 받도록 하는 방식
  - 목적
    - 서명 과정에서 메시지 내용의 기밀성 보장
    - 서명 요청자와 서명 결과 사이의 연결성 추적 방지
  - 과정
    - 서명 요청자는 메시지 M에 블라인드(Blind) 알고리즘을 적용하여 블라인드된 메시지 M'를 생성함
    - 서명자에게 M'에 대한 서명 S'를 요청하고, 해당 서명에 언블라인드(Unblind) 알고리즘을 적용하여 M에 대해 유효한 서명 S로 변환함

변형 서명 구조

- Blind Digital Signature Scheme (2/2)
  - 구현
    - RSA 구조를 변형하고 비밀 난수 r을 이용하여 구현할 수 있음
      - $M' = M \times r^e \ mod \ n$로 하면 $S' = M'^d \ mod \ n$이 됨
      - $S' \times r^{-1} \equiv M'^d r^{-1} \equiv (M \times r^e)^d r^{-1} \equiv M^d r^{ed} r^{-1} \equiv M^d r^1 r^{-1} \equiv M^d (mod \ n)$이므로, 서명 요청자는 S', r을 이용하여 S를 얻을 수 있음



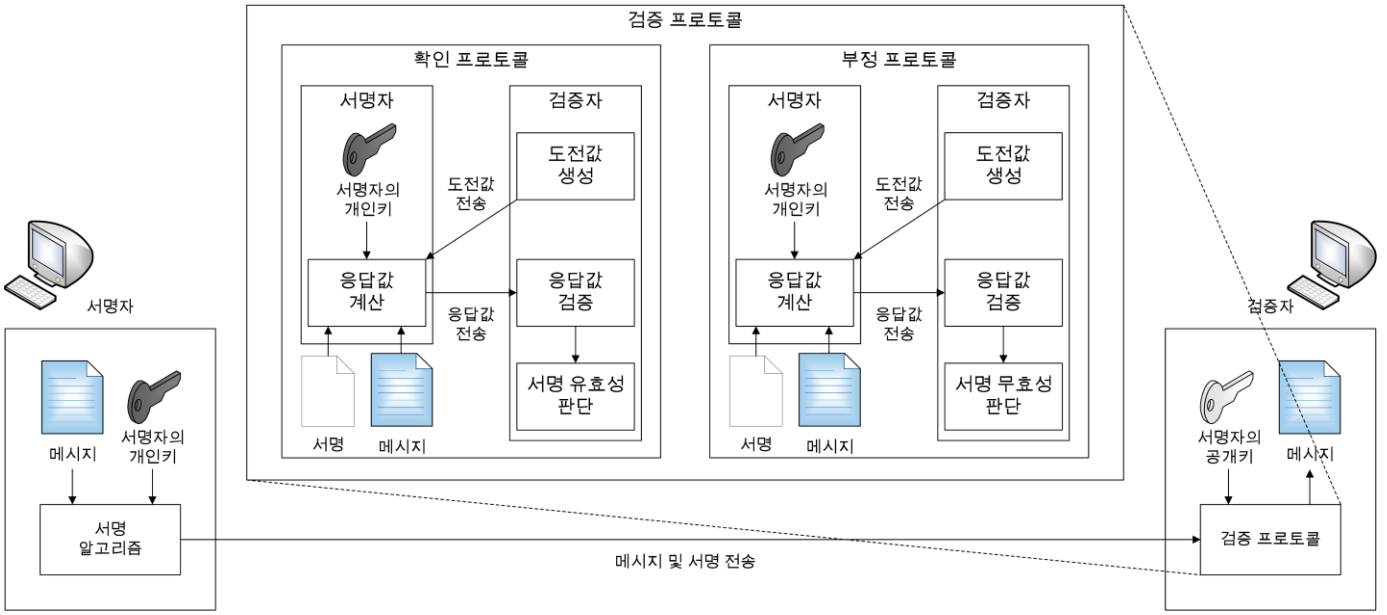
<그림 18> Blind Digital Signature Scheme 개념도

변형 서명 구조

- Undeniable Digital Signature Scheme (1/2)
  - 개요
    - 일반 디지털 서명과 달리, 서명자의 협조 없이는 서명의 진위를 확인할 수 없도록 한 방식
  - 목적
    - 서명의 유효성 검증을 서명자가 통제할 수 있도록 함
    - 제3자에 의한 무단 검증을 제한함
  - 과정
    - 검증 알고리즘 대신 확인 프로토콜(Confirmation Protocol)을 사용하여 서명자가 검증자의 도전값(Challenge)에 대한 응답값(Response)를 전송하여 검증을 진행하도록 함
    - 부정 프로토콜(Disavowal Protocol)을 사용하여 유효하지 않은 서명에 대해, 서명자가 서명하지 않았음을 증명할 수 있도록 함

변형 서명 구조

- Undeniable Digital Signature Scheme (2/2)
  - 구현
    - 확인 프로토콜은 공개키와 서명 사이의 이산대수 관계가 동일함을 도전-응답 방식으로 증명함
    - 부정 프로토콜은 후보 서명이 해당 관계를 만족하지 않음을 서명값 자체나 개인키를 공개하지 않고 증명함



<그림 19> Undeniable Digital Signature Scheme 개념도

Thanks!

이 성 현(seonghyeon@pel.sejong.ac.kr)

부록 #1

- Undeniable Digital Signature Scheme 동작 과정 (1/5)
  - 키 생성
    - 소수 p, Z_p^*의 생성원 e_1, 개인키 d를 선택하고 공개키 $e_2 = e_1^d \ (mod\ p)$를 계산하여 공개키를 (e_1, e_2, p)로, 개인키를 d로 정함
  - 서명 알고리즘
    - 서명 $S = M^d \ (mod\ p)$로 하여 메시지-서명 쌍 (M, S)를 검증자에게 전달함
    - 정상적인 서명인 경우, $e_2 = e_1^d \ (mod\ p)$와 $S = M^d \ (mod\ p)$가 모두 성립함

부록 #1

- Undeniable Digital Signature Scheme 동작 과정 (2/5)
  - 확인 프로토콜
    1. 서명자가 비밀 난수 $r \in_R \{0,1,\dots,p-2\}$을 선택하고 $A = e_1^r \ (mod\ p), B = M^r \ (mod\ p)$를 계산하여 검증자에게 전달함
    2. 검증자가 무작위 도전값 $c \in_R \{0,1,\dots,p-2\}$를 선택하여 서명자에게 전달함
    3. 서명자는 응답값 $T = (r + cd)\ mod\ (p-1)$를 계산하여 검증자에게 전달함
    4. 검증자는 $e_1^T = e_1^{r+cd} = e_1^r\left(e_1^d\right)^c = Ae_2^c \ (mod\ p), M^T = M^{r+cd} = M^r\left(M^d\right)^c = BS^c \ (mod\ p)$를 계산하고, 두 식이 모두 성립하면 서명 S를 유효하다고 판정함

부록 #1

- Undeniable Digital Signature Scheme 동작 과정 (3/5)
  - 부정 프로토콜 (1/3)
    1. 검증자가 두 비밀값 $a \in_R \{0,1,\dots,k\}, b \in_R \{0,1,\dots,p-2\}$ (단, k는 프로토콜 상에서 서로 사전에 합의한 정수)를 선택함
    2. 검증자는 선택한 a, b에 대해 $C_1 = M^a e_1^b \ (mod\ p), C_2 = S^a e_2^b \ (mod\ p)$를 계산하여 서명자에게 전달함
    3. 서명자는 개인키 d를 이용하여 $R = C_1^d C_2^{-1} (mod\ p)$를 계산하고, 이는 $R = C_1^d C_2^{-1} = \left(M^a e_1^b\right)^d \left(S^a e_2^b\right)^{-1} = M^{ad} e_1^{bd} S^{-a} e_2^{-b} = M^{ad} S^{-a} = \left(M^d S^{-1}\right)^a \ (mod\ p)$로 볼 수 있음

부록 #1

- Undeniable Digital Signature Scheme 동작 과정 (4/5)
  - 부정 프로토콜 (2/3)

커밋먼트(Commitment):
특정 값을 공개하지 않은 채 미리 고정하고, 이후 해당 값을 공개하여 확인할 수 있도록 하는 암호학적 기법

4. 서명이 유효한 경우, 서명자는 a를 알 수 없음
   - $S = M^d \ (mod\ p)$이므로 $M^d S^{-1} \equiv 1 \ (mod\ p)$이 되어 $R = 1^a = 1 \ (mod\ p)$이 되므로, 서명자는 R만으로는 a를 알 수 없음
5. 서명이 무효한 경우, 서명자는 a를 찾을 수 있음
   - $S \neq M^d \ (mod\ p)$이므로 $M^d S^{-1} \not\equiv 1 \ (mod\ p)$이고, 서명자는 $W = M^d S^{-1} \ (mod\ p)$를 알고 있으므로 $R = W^a \ (mod\ p)$에서 $W^0, W^1, \ldots, W^k$를 계산하는 과정에서 a를 찾을 수 있음
6. 서명자는 임의의 비밀값 ρ를 이용하여 커밋먼트 $D = Com(a, \rho)$를 생성하고 검증자에게 전달함

부록 #1

- Undeniable Digital Signature Scheme 동작 과정 (5/5)
  - 부정 프로토콜 (3/3)
    7. 검증자가 b를 공개하고 서명자는 알아낸 a와 공개된 b를 통해 $C_1 = M^a e_1^b \ (mod\ p), C_2 = S^a e_2^b \ (mod\ p)$가 모두 성립하는지 확인함
    8. 서명자가 ρ를 공개하고 검증자는 이전에 전달받은 D와 공개된 ρ를 통해 $Open(D, \rho) = a$가 성립하는지 확인하고, 성립할 경우, 서명이 무효하다고 판정함