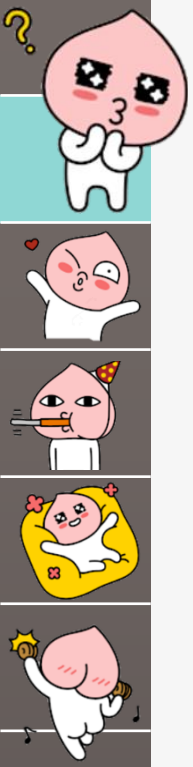
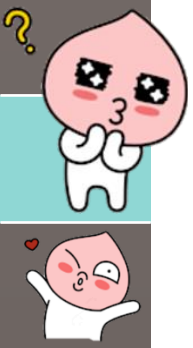
암호학적으로 안전한 카카오톡 E2E기능 제작

이름 비오비 8기 취약점 트랙 교육생 김희연



• 목 차

사용된 암호기법 ▸ 프로토콜 설명 ▸ CSPRNG 설계 ▸ Demo영상

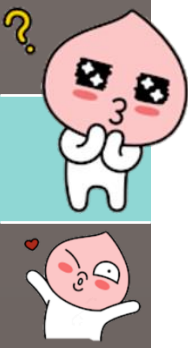


• 사용된 암호 기법

1. RSA로 메시지 암호화

2. ECDSA(Elliptic Curve DSA)로 RSA 공개키에 대한 전자서명

3. CSPRNG의 조건을 만족하는 난수생성기



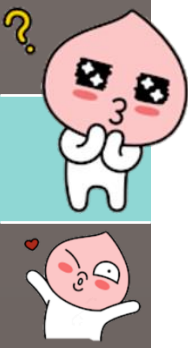
- # RSA로 메시지 암호화: 기밀성

1. RSA: 암호화와 복호화에 다른 키를 사용하는 공개키 암호(비대칭키)

2. 사전에 서로 동일한 키를 공유해야 하는 대칭키 알고리즘의 한계점 극복

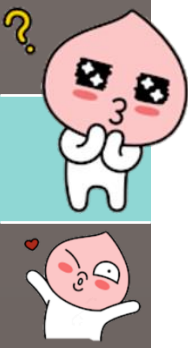
3. RSA에서 공개키(n,e)/개인키(p,q,d) → 키는 통신을 시작할 때마다 새로 생성(일회성)

->두 소수 p와 q는 1024비트를 사용



• ECDSA(Elliptic Curve DSA)로 전자서명: 무결성, 부인방지

1. 타원 곡선을 이용한 공개키 암호시스템 사용 : 224비트 길이의 암호시스템으로도 RSA 2048비트 길이의 암호시스템과 같은 안전성을 갖는다 → 키는 prime256v1으로 생성 (일회성)
2. 해쉬함수(SHA-384)를 함께 사용하여 실제 메시지보다 훨씬 짧은 상태에 대해서 서명을 생성



- **구현상의 문제점**

1. crypto 모듈을 사용하는 과정에서, “randombytes” 부분에서 에러 발생
2. RSA, ECDSA 키 생성하는 과정에서 시간지연으로 카카오톡 터지는 현상

→ 서버를 제작하여 RSA, ECDSA 키 및 randombyte를 생성하여 클라이언트에게 전달

Alice's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

Bob's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

Alice's Server

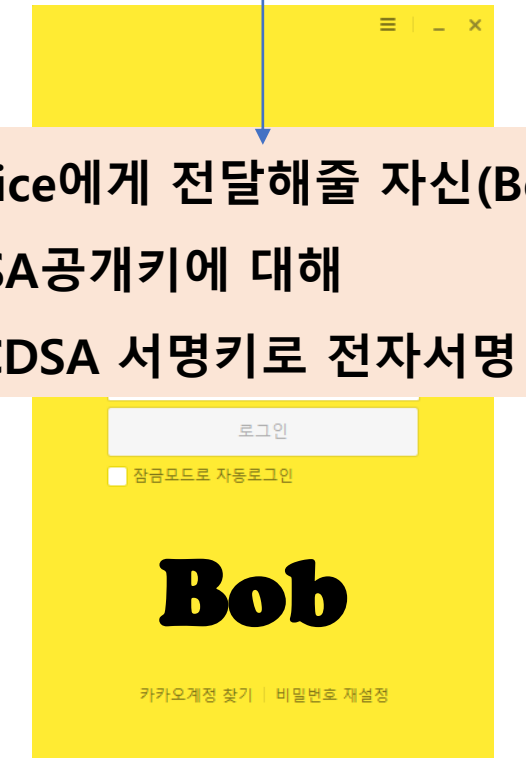
- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

Bob에게 전달해줄 자신(Alice)의
RSA공개키에 대해
ECDSA 서명키로 전자서명 생성

Bob's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

Alice에게 전달해줄 자신(Bob)의
RSA공개키에 대해
ECDSA 서명키로 전자서명 생성



Alice's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

RSA공개키, ECDSA 검증키,
RSA공개키에 대한 **전자서명**

RSA공개키, ECDSA 검증키,
RSA공개키에 대한 **전자서명**

Bob's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

Alice's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

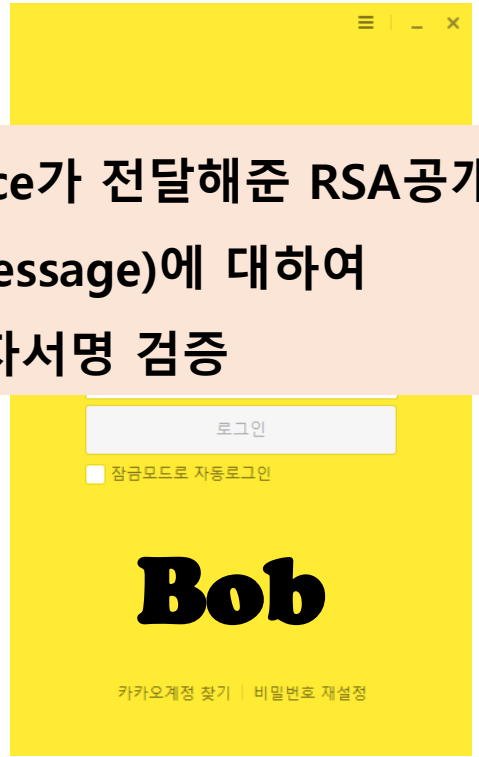
RSA공개키, ECDSA 검증키,
RSA공개키에 대한 **전자서명**

RSA공개키, ECDSA 검증키,
RSA공개키에 대한 **전자서명**

Bob's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

Bob이 전달해준 RSA공개키
(Message)에 대하여
전자서명 검증



Alice가 전달해준 RSA공개키
(Message)에 대하여
전자서명 검증

Alice's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

Bob의 RSA공개키로 암호화한
카카오톡 메시지 전달

Bob's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

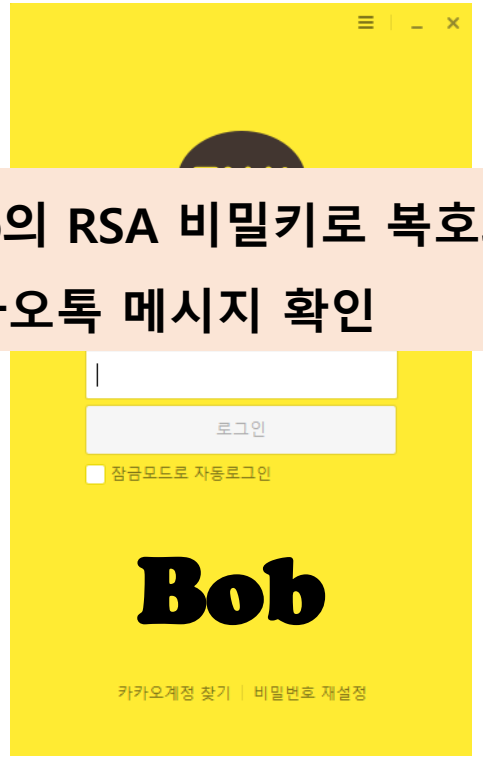
Alice's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

Bob의 RSA공개키로 암호화한
카카오톡 메시지 전달

Bob's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키



Bob의 RSA 비밀키로 복호화하여
카카오톡 메시지 확인

Alice's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

Alice의 RSA공개키로 암호화한
카카오톡 메시지 전달

Bob's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

Alice's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

Alice의 RSA 비밀키로 복호화하여
카카오톡 메시지 확인

Alice의 RSA공개키로 암호화한
카카오톡 메시지 전달

Bob's Server

- 난수
- RSA 공개키+비밀키
- ECDSA 검증키+서명키

- **CSPRNG(Cryptographically Secure Pseudo Random Number Generator)**

1. CSPRNG가 생성하는 키 스트림 s_{i+1}, s_{i+2}, ... s_{i+k}가 주어졌을 때, 다음 비트 s_{i+k+1}을 50%이상의 확률로 예측하는 것이 계산적으로 불가능해야 한다.

2. 주어진 키 스트림의 이전 비트인 s_i, s_{i-1}, ... 중 한 비트를 50% 이상의 확률로 예측하는 것이 계산적으로 불가능해야 한다.

주어진 난수열에서 이전 난수 또는 앞으로 생성될 난수를 예측하기 어려워야 한다!

- **CSPRNG(Cryptographically Secure Pseudo Random Number Generator)**

1. 일방향 해쉬 함수의 사용

2. 블록암호의 사용

3. 수학적 난제에 기반한 생성 방법

- **CSPRNG(Cryptographically Secure Pseudo Random Number Generator)**

1. 일방향 해쉬 함수의 사용

2. 블록암호의 사용

3. 수학적 난제에 기반한 생성 방법

- **수학적 난제 →**

NP문제: 다항시간에 해결할 수 있는 결정적 알고리즘이 밝혀지지 않음

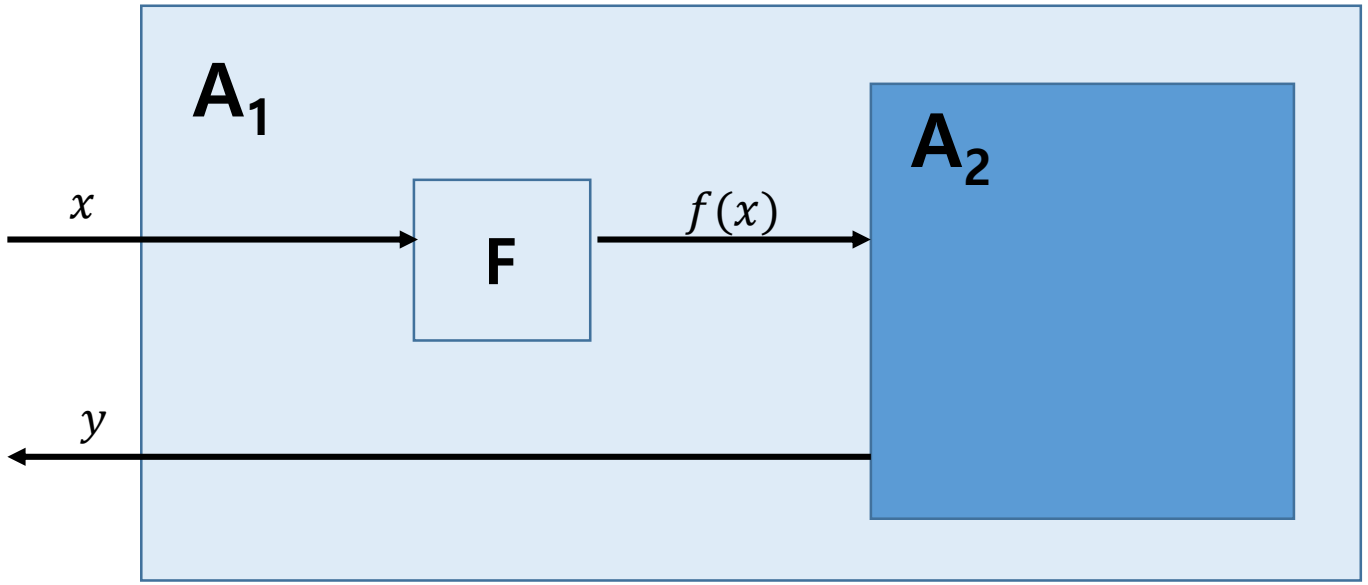
1. **소인수분해 문제**

2. **이산 대수 문제 : $\mathrm{p}, \mathrm{g}, y$가 주어졌을 때, $g^x \equiv y \ (mod\ p)$ → x를 계산하는 문제**

- *RSA는 소인수분해 문제에 기반, ECDSA는 이산 대수 문제에 기반하여 안전성을 확보*

- **수학적 난제 → <제곱근문제>: y 가 주어졌을 때, $x^2 \equiv y \,(mod\, n)$ → x를 계산하는 문제**

-제곱근 문제가 인수분해 문제만큼 어려움이 증명된다면, 제곱근 문제도 CSPRNG를 만족하게 됨

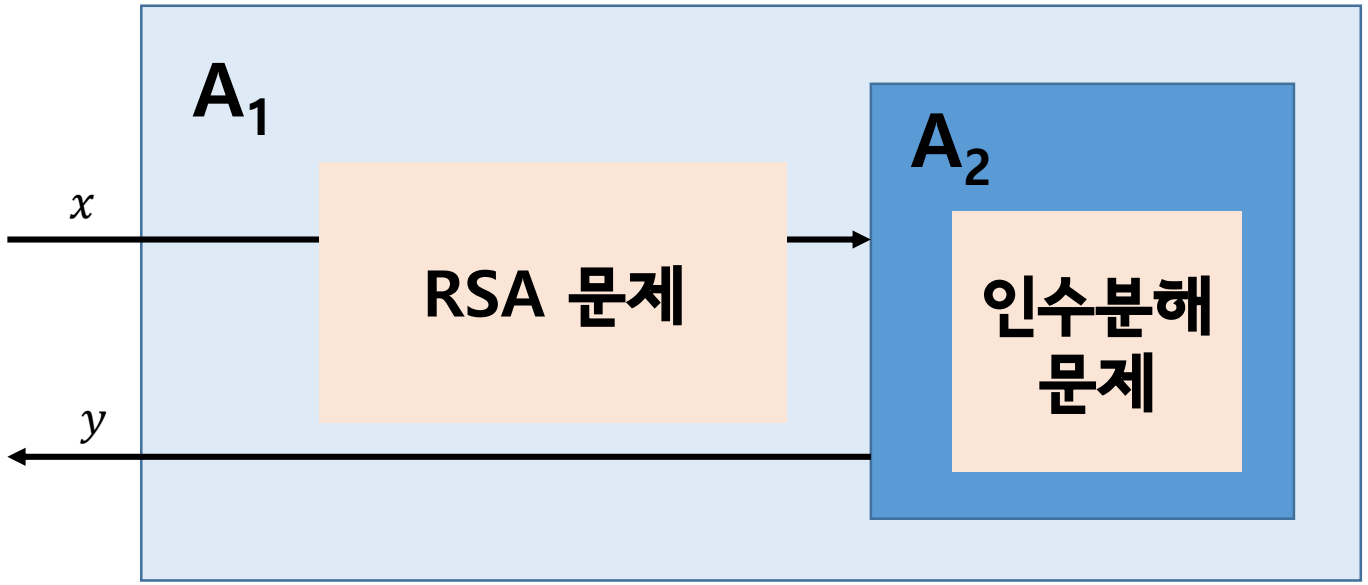


A_1문제를 풀기 위해서 A_2문제를 풀어야 한다면, 문제 A_1이 문제 A_2로 리덕션된다.

즉, A_1이 A_2보다 어렵지 않다 = A_2가 A_1 만큼 어렵다!

- **수학적 난제 → <제곱근문제>: y 가 주어졌을 때, $x^2 \equiv y \,(mod\, n)$ → x를 계산하는 문제**

-제곱근 문제가 인수분해 문제만큼 어려움이 증명된다면, 제곱근 문제도 CSPRNG를 만족하게 됨

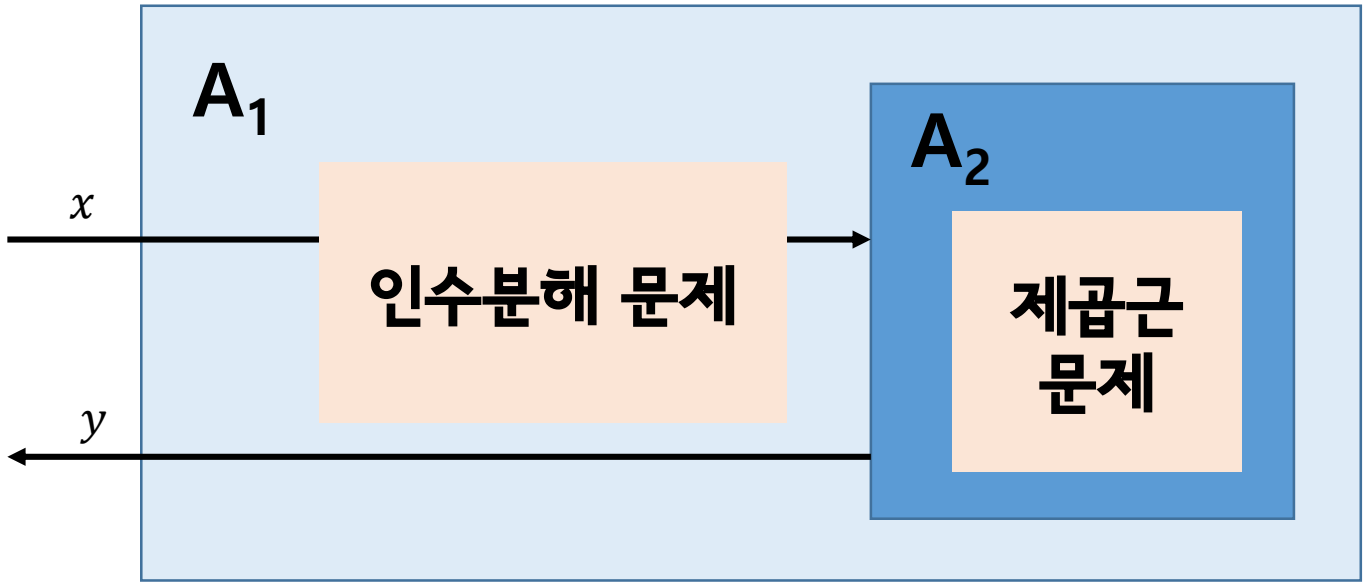


A_1문제를 풀기 위해서 A_2문제를 풀어야 한다면, 문제 A_1이 문제 A_2로 리덕션된다.

즉, A_1이 A_2보다 어렵지 않다 = 두 문제는 비슷한 복잡도를 갖고 있으며 서로 동치류이다!

- **수학적 난제 → <제곱근문제>: y 가 주어졌을 때, $x^2 \equiv y \,(mod\, n)$ → x를 계산하는 문제**

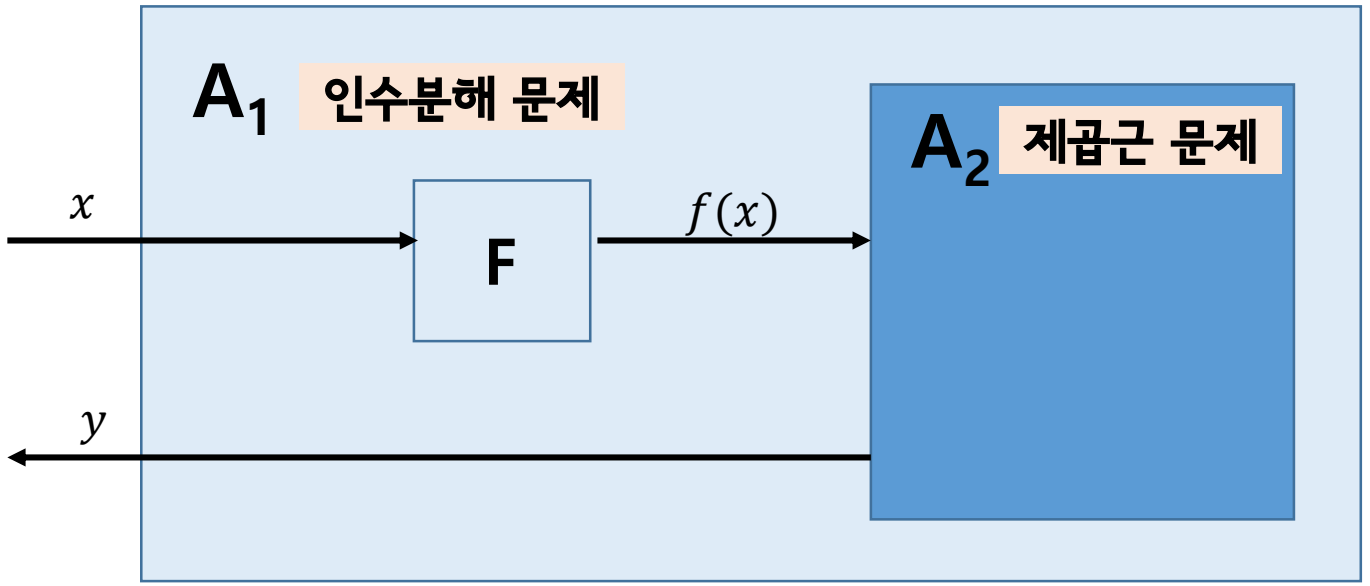
-제곱근 문제가 인수분해 문제만큼 어려움이 증명된다면, 제곱근 문제도 CSPRNG를 만족하게 됨



A_1문제를 풀기 위해서 A_2문제를 풀어야 한다면, 문제 A_1이 문제 A_2로 리덕션된다.

즉, A_1이 A_2보다 어렵지 않다 = 두 문제는 비슷한 복잡도를 갖고 있으며 서로 동치류이다!

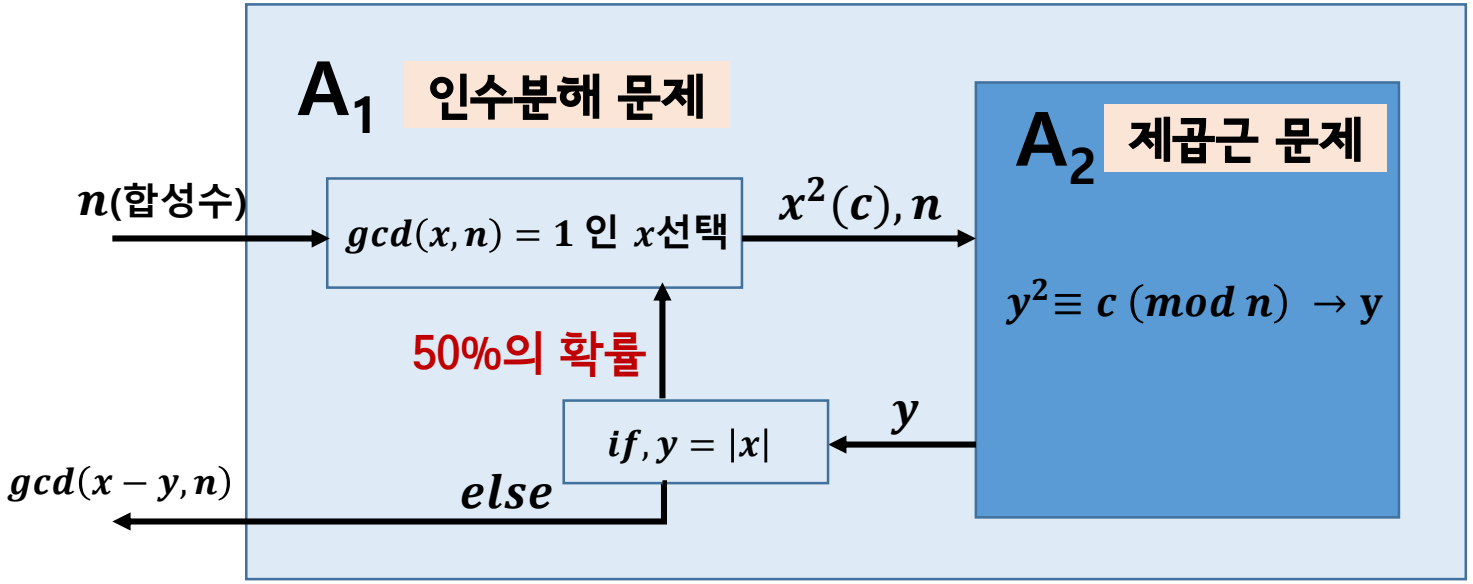
- **수학적 난제 → <제곱근문제>: y 가 주어졌을 때, $x^2 \equiv y\ (mod\ n)$ → x를 계산하는 문제**



정리 1) 정수 x와 y에 대하여 $x^2 \equiv y^2\ (mod\ n)$ 이고 $x \neq y\ (mod\ n)$ 이면, $\gcd(x-y, n)$ 혹은 $\gcd(x+y, n)$는 n의 1이 아닌 인수이다.

정리 2) $n = p \cdot q$ 이고 $\gcd(y, n) = 1$ 인 경우, $x^2 \equiv y^2\ (mod\ n)$ 는 4개의 해가 존재하며, 그 중 2개는 $x \equiv y\ (mod\ n), x \equiv -y\ (mod\ n)$ 이다.

- **수학적 난제 → <제곱근문제>: y 가 주어졌을 때, $x^2 \equiv y\ (mod\ n)$ → x를 계산하는 문제**



정리 1) 정수 x와 y에 대하여 $x^2 \equiv y^2\ (mod\ n)$ 이고 $x \neq y\ (mod\ n)$ 이면,
$\gcd(x-y,n)$ 혹은 $\gcd(x+y,n)$는 n의 1이 아닌 인수이다.

정리 2) $n = p \cdot q$ 이고 $\gcd(y,n) = 1$ 인 경우, $x^2 \equiv y^2\ (mod\ n)$ 는 4개의 해가 존재하며,
그 중 2개는 $x \equiv y\ (mod\ n), x \equiv -y\ (mod\ n)$ 이다.

- **수학적 난제 → <제곱근문제>: y 가 주어졌을 때, $x^2 \equiv y\ (mod\ n)$ → x를 계산하는 문제**

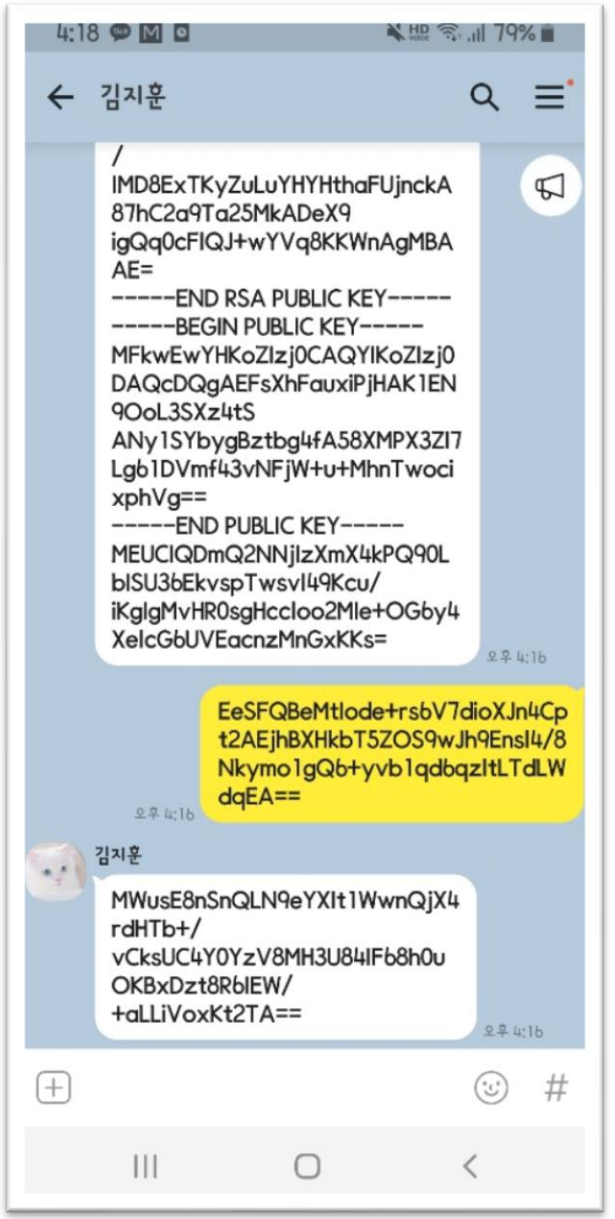
1. **$4k+3$ 의 형태인 두 개의 큰 소수 p, q 를 선택한다 (p와 q는 1024비트)**

   ->밀러-라빈 소수 판별법: $(1-\left(\frac{1}{4}\right)^n)$의 확률을 가진다 → 56번 진행

2. **$n = p \cdot q$ 를 계산한다.**
3. **n 과 서로소인 임의의 수 r을 선택한다. $(\gcd(n,r)=1)$**
4. **난수생성기의 초기값으로 사용하는 $x_0 \equiv r^2\ (mod\ n)$ 을 계산한다.**
5. **$x_{i+1} = x_i^2\ (mod\ n)$을 재귀적으로 계산하고 각 x_i 의 최하위 비트를 의사 난수의 비트로 선택한다.**
6. **1~5번을 8번 진행해서 random byte 1바이트 생성**

Demo

Demo


4:18
79%
김지훈
/
IMD8ExTKyZuLuYHYHthaFUjnckA
87hC2a9Ta25MkADeX9
igQq0cFIQJ+wYVq8KKWnAgMBA
AE=
-----END RSA PUBLIC KEY-----
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0
DAQcDQgAEFsXhFauxiPjHAK1EN
9OoL3SXz4tS
ANy1SYbygBztbg4fA58XMPX3ZI7
Lgb1DVmf43vNFjW+u+MhnTwoci
xphVg==
-----END PUBLIC KEY-----
MEUCIQDmQ2NNjIzXmX4kPQ90L
bISU3bEkvspTwsvI49Kcu/
iKgIgMvHR0sgHccIoo2MIe+OGby4
XeIcGbUVEacnzMnGxKKs=
오후 4:16
EeSFQBeMtIode+rsbV7dioXJn4Cp
t2AEjhBXHkbT5ZOS9wJh9EnsI4/8
Nkymo1gQb+yvb1qdbqzItLTdLW
dqEA==
오후 4:16
김지훈
MWusE8nSnQLN9eYXIt1WwnQjX4
rdHTb+/
vCksUC4Y0YzV8MH3U84IFb8h0u
OKBxDzt8RbIEW/
+aLLiVoxKt2TA==
오후 4:16

감사합니다