충남대학교
CHUNGNAM NATIONAL UNIVERSITY

RSA

2022년 10월 12일 수요일

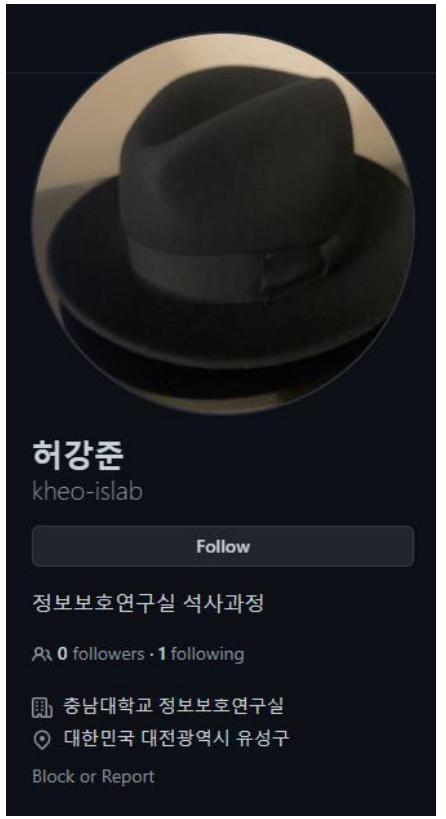
정보보호

충남대학교 정보보호연구실 허강준

과제 관련 공지

- 과제 제출 기한은 언제나 다음 수업시간 전까지입니다.
- 레포 Collaborator 추가여부 확인 요망!!

본계: 0x00000FF
총 든 이상한 아저씨 (x)
이거 아님



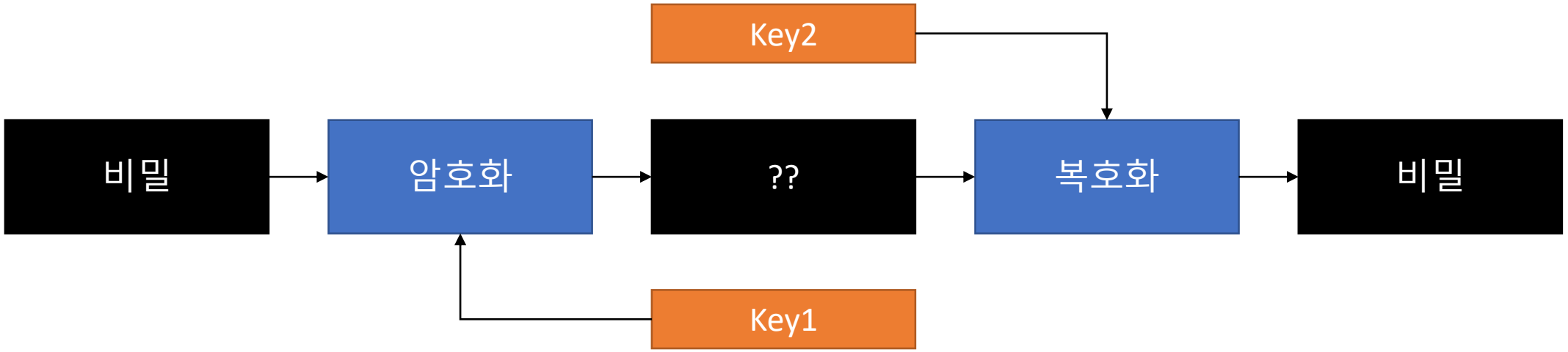
채점계: kheo-islab
검정페도라 (o)
이걸로 추가할 것

공개키 암호

- **대칭**키암호 → 암호화 했던 키로 복호화가 가능

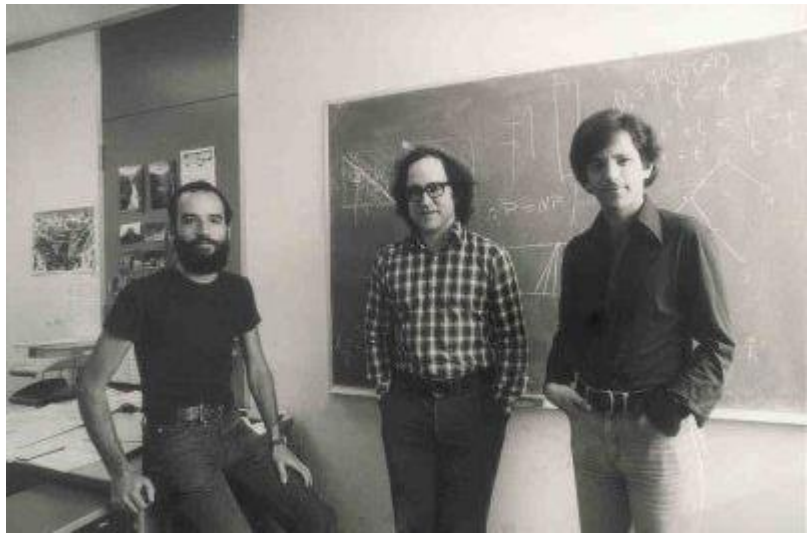
공개키 암호

- 이건 안돼?



RSA

- Ronald Rivest, Adi Shamir and Leonard Adleman



이상하게도 공학쪽으로 유명한 사람들은 죄다 앞머리가 허전합니다

- 1977년에 발표
- 대표적인 공개키 암호 시스템 (Public-key Cryptosystem)

- 소수(Prime Number)와 소인수분해
- 60을 인수분해

$$60 = 12 \times 5 = 4 \times 3 \times 5 = 2^2 \times 3 \times 5$$

- 소수? 1과 자기 자신만이 약수인 자연수!

RSA

- 두 소수 P, Q (아주 크고 서로 다른) 를 선택한 후 $N = PQ$를 만들기
- 예를 들어…

N=

25195908475657893494027183240048398571429282126204032027777137836043662020707595556264018525880784406918290641249515082189298559149176184502808489120072844992687392807287776735971418347270261896375014971824691165077613379859095700097330459748084284017974291006424586918171951187461215151726546322822168699875491824224336372590851418654620435767984233871847744479207399342365848238242811981638150106748104516603773060562016196762561338441436038339044149526344321901146575444541784240209246165157233507787077498171257246796292638635637328991215483143816789988504044536402352738195137863656439121201039712282212072035 7

RSA-2048로 만든 N(617자리)의 예시. 이거 깨면 상금 준다고 했던거 같은데 깨지긴 했나?

- 오일러 함수 $\varphi(N) = (P-1)(Q-1)$
- 적절한($\varphi(N) > e$, $\varphi(N)$ 와 e는 서로소) 지수 e 를 선택
  - 보통 3이나 65537(표준임!)을 많이 씀

- e 와 N 을 공개 → 공개키!

```
def generate(bits, randfunc=None, e=65537):
    """Create a new RSA key pair.

    The algorithm closely follows NIST `FIPS 186-4`_ in its
    sections B.3.1 and B.3.3. The modulus is the product of
    two non-strong probable primes.
    Each prime passes a suitable number of Miller-Rabin tests
    with random bases and a single Lucas test.

    Args:
      bits (integer):
        Key length, or size (in bits) of the RSA modulus.
        It must be at least 1024, but **2048 is recommended.**
        The FIPS standard only defines 1024, 2048 and 3072.
      randfunc (callable):
        Function that returns random bytes.
        The default is :func:`Crypto.Random.get_random_bytes`.
      e (integer):
        Public RSA exponent. It must be an odd positive integer.
        It is typically a small number with very few ones in its
        binary representation.
        The FIPS standard requires the public exponent to be
        at least 65537 (the default).

    Returns: an RSA key object (:class:`RsaKey`, with private key).

    .. _FIPS 186-4: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
    """
```

pyCryptodome 의 예시

- 공개키가 있으니 암호화 해봅시다

$$c = m^e \ mod \ N$$

- 이 값들을 사용해보면…
  - 평문 M=8
  - P = 3, Q = 5 → N = 15
  - e = 3

- 512 mod 15 = 2 → 암호문 c!

- 복호화를 위한 개인키 만들기? *확장 유클리드 호제법*

$$am + bn = GCD(m, n)$$

수학시간 아닌거 아는데 이번에만 참아주셔요...

- 이걸 이용해서 개인키 d를 구해보자

$$ex + \varphi(N)y = 1 \ \ (x = d)$$

시험에 안나와요 이거

$$de \equiv 1 \ mod \ \varphi(N)$$

이건 나올지도

- e 와 곱해서 $\varphi(N)$으로 나눴을 때 1이 되는 수 → 개인키!

- 그러면

$$3d \equiv 1 \; mod \; 8$$

- 가능한 자연수 d의 값은… 3, 11 등
- 이 중에 하나 골라잡아 N과 함께 개인키로 사용!
  - 3은 e와 같으니 11을 써보자

- N=15, d=11, c=2

$$m = c^d \ mod \ N$$

$$c^d = 2048$$

$$c^d \ mod \ N = 8 = m$$

이렇게 평문을 복호화 할 수 있다

RSA 구현하기

```python
def make_keys(p: BigNumber, q: BigNumber):
    # place your own implementation of make_keys
    # use e = 65537 as if FIPS standard

    return [e, d, n]

def rsa_encrypt(plain: BigNumber, e: BigNumber, n: BigNumber):
    # place your own implementation of rsa_encrypt
    pass

def rsa_decrypt(cipher: BigNumber, d: BigNumber, n: BigNumber):
    # place your own implementation of rsa_decrypt
    pass
```

```
PS C:\Users\patche\Desktop\sdes> & C:/Users/patche/AppData/Local/Programs/Python/Python310/python.exe c:/Users/patche/Desktop/sdes/rsa.py
P = 233, Q = 523, N = 121859, M = 7, e = 65537, d = 54017, C = 82213, M2 = 7
RSA Success!!
PS C:\Users\patche\Desktop\sdes> & C:/Users/patche/AppData/Local/Programs/Python/Python310/python.exe c:/Users/patche/Desktop/sdes/rsa.py
P = 929, Q = 349, N = 324221, M = 10, e = 65537, d = 54017, C = 285430, M2 = 10
RSA Success!!
PS C:\Users\patche\Desktop\sdes> & C:/Users/patche/AppData/Local/Programs/Python/Python310/python.exe c:/Users/patche/Desktop/sdes/rsa.py
P = 277, Q = 149, N = 41273, M = 13, e = 65537, d = 15233, C = 28227, M2 = 13
RSA Success!!
```

죄다 랜덤이니 함수만 구현하면 OK

이후 일정

주차	실습 주제	과제	날짜
1	오리엔테이션 & 썰풀기	과제를 위한 GitHub 설정	9/7
2	카이사르&비즈네르 암호	ENIGMA	9/14
3	XOR과 블록암호	Simplified DES 구현하기	9/21
4	여러가지 블록암호	블록암호를 이용하여 암호통신기 완성하기	9/28
5	블록암호 운용모드	S-DES-CBC, S-DES-ECB 구현하기	10/5
6	RSA	RSA 구현하기, ~~저강도 RSA 크랙하기~~	10/12
7	해시	암호통신기에 무결성 검증 기능 추가하기	10/19
8	중 간 고 사 (10/24)		공강
9	메세지 인증코드(MAC)	HMAC 구현하기	11/2
10	디지털 서명	사설인증서 생성 및 프로그램 코드 서명	11/9
11	하이브리드 암호	하이브리드 기반 암호 통신기	11/16
12	난수	시드값 추측을 이용한 암호문 크랙	11/23
13	블록체인과 머클 트리	머클트리 구현하기	11/30
14	TLS와 PGP(GPG)	GPG를 이용하여 암호 메일 보내기	12/7
15	기 말 고 사 (12/12)		종강

질문?

- 없으면 자리에서 일어나셔도 좋습니다 :)
- 대학원 입학 문의는 언제나 환영
  - 블록체인, Web 3, 해킹 관심있거나 유경험자 우대

입학문의

- 류재철 교수님 (jcryou [at] cnu.ac.kr)
- 허강준 조교 (knowledge [at] o.cnu.ac.kr)