RSA 알고리즘

유튜브 주소 : https://youtu.be/e1E2oPkE5ls

공개키 암호

RSA 암호

RSA 코드 분석

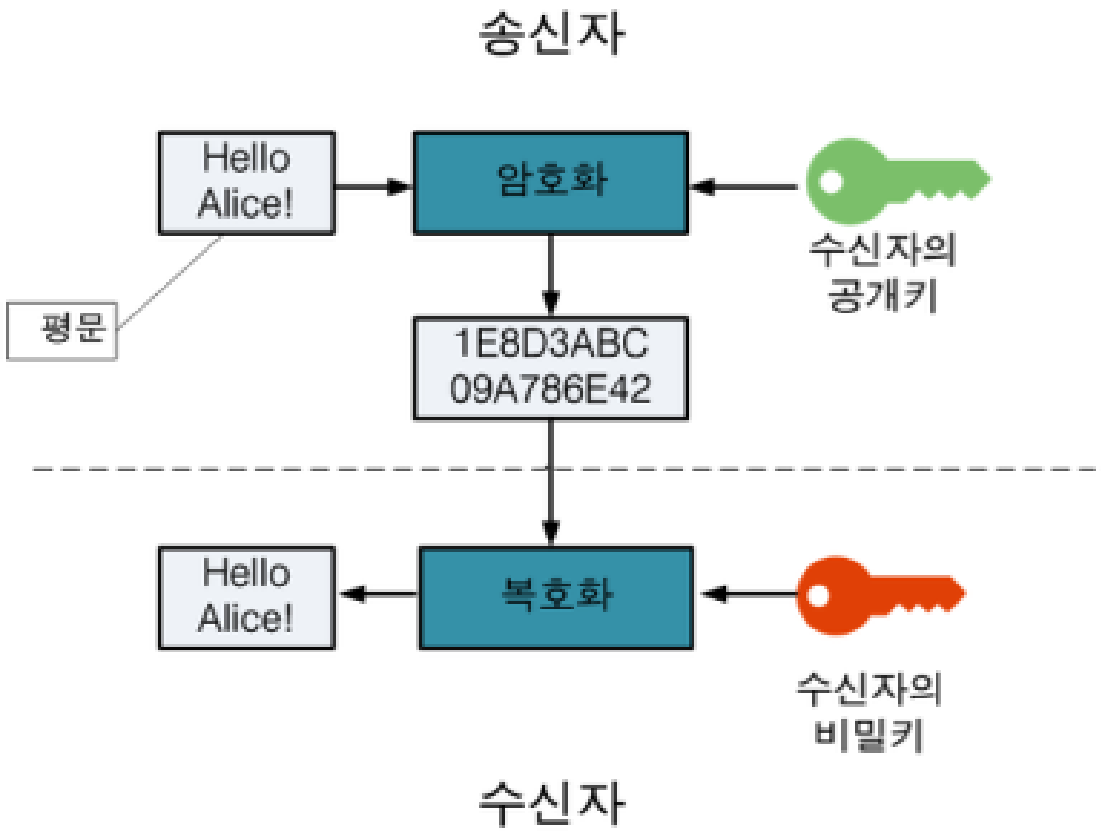
공개키 암호

- 두 개의 다른 키를 사용
  - 공개키 : 모든 사람이 접근 가능한 키
  - 개인키 : 사용자 자신만 소유하는 키(비공개)

대칭키 암호	공개키 암호
암호화, 복호화 시 동일한 키 사용	암호화, 복호화 시 서로 다른 키를 사용
수신자, 송신자 간 키 교환 필요	키 교환 필요 X(공개키 이용)
공유한 키는 비밀로 유지	개인키만 비밀로 유지
공개키에 비해 속도 빠름(약 1000배)	대칭키에 비해 속도 느림(약 1000배)
키 분배가 어려움	공개키만 공개하면 됨

공개키 암호 모델

- 송신자 - 수신자의 공개키를 받아 데이터를 암호화 후 전달
- 수신자 – 전달받은 암호화된 데이터를 자신의 개인키로 복호화



공개키 암호 장단점

장점

- 키 관리가 쉬움
  - 키가 공개되어도 상관 없음
- 대칭키에 비해 키의 개수가 적음
  - 공개키 - 2n개 필요, 대칭키 – n(n-1)/2개 필요
- 전자 서명에도 이용가능

단점

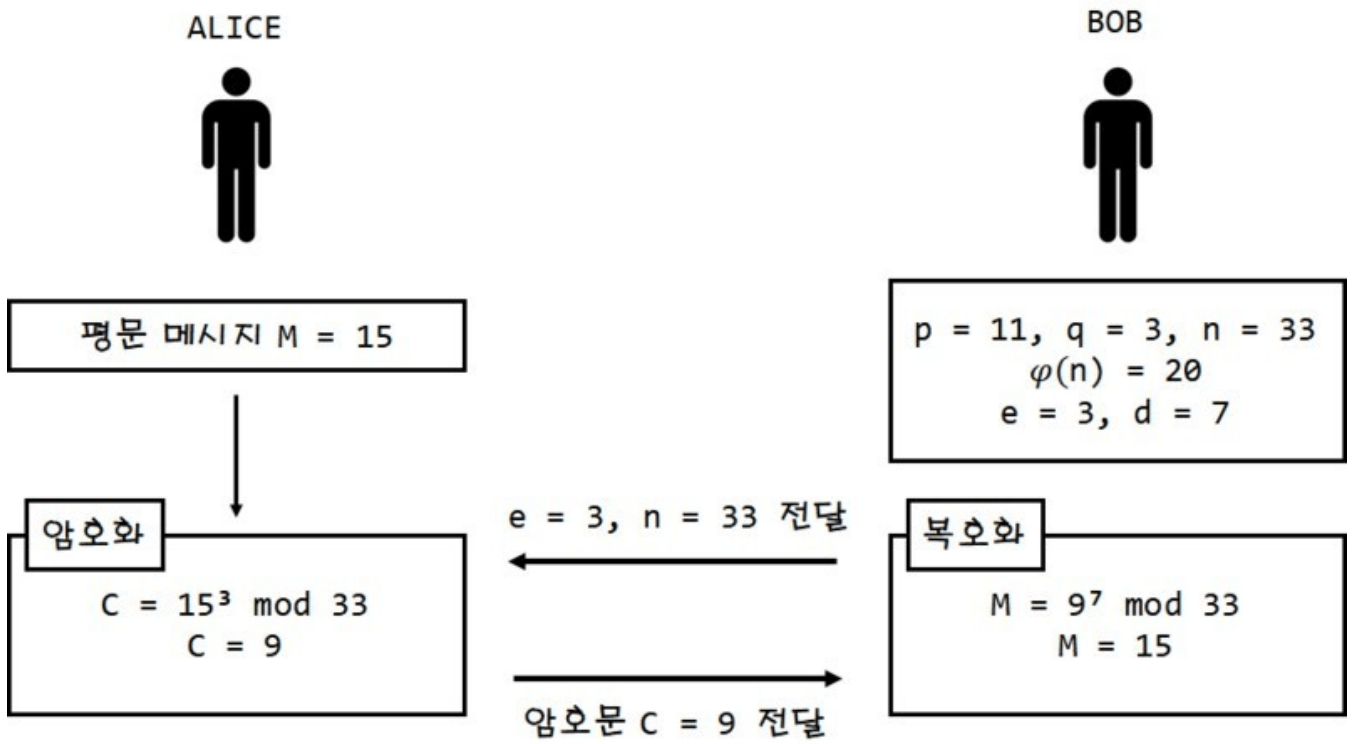
- 브루트 포스 공격에 취약
  - 키의 크기를 크게 함으로써 방지할 수 있음 But 속도는 느려짐
- 대칭키에 비해 속도가 매우 느림
  - 실시간 암호화 통신에는 사용이 힘듦

RSA 암호

- 가장 대표적인 공개키 알고리즘
  - 전자 상거래에서 가장 흔히 사용되고 있음(RSA-2048)
- Rivest, Shamir, Adelman 세 사람의 이니셜을 따서 작명
  - 1983년 미국 MIT에서 특허로 등록
- 큰 정수의 소인수 분해가 어렵다는 점을 이용하여 암호화
  - 곱하는 것은 쉽지만 다시 분해하는 것은 어려움
- 쇼어 알고리즘 – 소인수 분해를 빠르게 처리할 수 있는 양자 알고리즘
  - 양자 컴퓨터가 실용화되면 RSA의 무력화 가능

RSA 암호

- d -> 개인키 : e mod(p-1)(q-1)의 역
- n -> 공개키 : p * q(p,q는 소수)
- e -> 공개키 : (p-1)*(q-1)과 서로소
  (1 < e < n)



RSA 암호

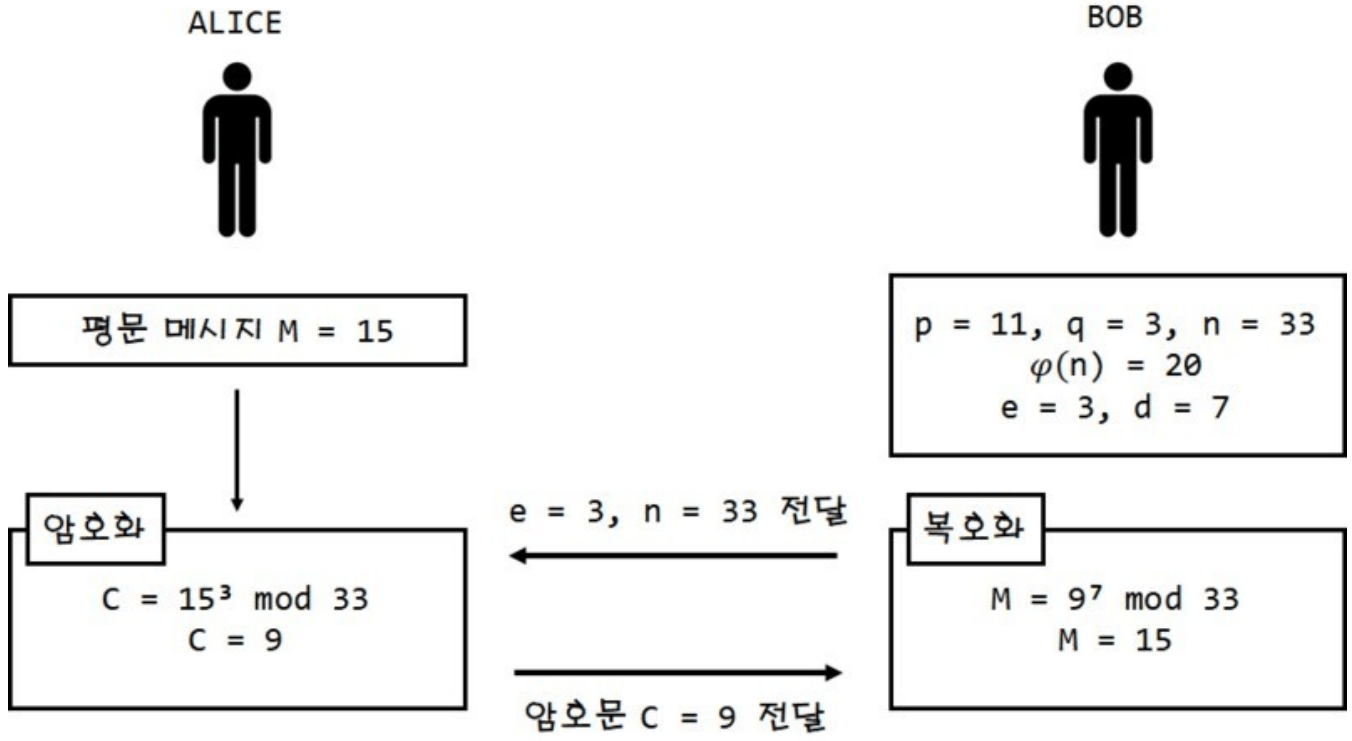
- 암호화 공식(M이 메시지일 때)
  - M^e mod n = C (0 <= M <= n-1)
- 복호화 공식(C가 암호문일 때)
  - C^d mod n = M
- 공개 키로 개인 키 구하기
  - e*d mod(p-1)(q-1) 일 때
    d 값 구하면 됨

Input

$4782969 \bmod 33$

Result

15



Input

$3375 \bmod 33$

Result

9

RSA 코드 분석

```c
#include <stdio.h>
#include <stdlib.h>
#define size 3000
#define n 6012707
#define e 3000037

long long mod(long long a, long long b) {
    if (a > 0)
        return a % b;
    else {
        return a - b * -((-a / b) + 1);
    }
}
int main(void) {
    printf( _Format: " n= %d, e=%d\n", n,e);

    int *store = (int)malloc( _Size: sizeof(int)*size);
    int length = 0;
    for (int i = 2; i < 2 * size; i++) {
        for (int j = 1; j <= i; j++) {
            if (j != 1 && j != i && i%j == 0)
                break;
            else if (j == i) {
                store[length++] = i;
            }
        }
    }
```

```c
    long long p = 0;
    long long q = 0;
    for (int i = 0; i < length; i++) {
        for (int j = i; j < length; j++) {
            if (store[i] > 0 && store[j] > 0) {
                if (store[i] * store[j] == n) {
                    p = store[i]; q = store[j];
                    break;
                }
            }
        }
    }
    free(store);
    for (long long d = 1; d < 500000; d++) {
        if (mod( a: e*d,  b: (p - 1)*(q - 1)) == 1) {
            printf( _Format: "\np=%lld\n", p);
            printf( _Format: "q=%lld\n",q);
            printf( _Format: "d=%lld\n",d);
            printf( _Format: "%d * %lld mod (%lld-1)(%lld-1) = %lld\n",
                    e,d,p,q,mod( a: e*d,  b: (p - 1)*(q - 1)));
            break;
        }
    }
}
```

```
 n= 6012707, e=3000037

p=2357
q=2551
d=327373
3000037 * 327373 mod (2357-1)(2551-1) = 1
```

RSA 코드 분석

```c
#include<stdio.h>

long pow_(long i, long j, long k) {
    double l, temp, p = 1;
    for (temp = 0; temp < j; temp++) {
        p = (p * ((double) i));
        l = (long) (p / k);
        p = p - (l * k);
    }
    return (long) p;
}

int encryption(int input, int e, int n) {
    int i = pow_(input, e, n);
    printf( _Format: "encrypt result= %d\n", i);
    return i;
}

int Decryption(int input, int d, int n) {
    int i = pow_(input, d, n);
    printf( _Format: "decrypt result= %d\n", i);
    return i;
}
```

```c
int main() {
    int input;
    scanf( _Format: "%d", &input);
    int p = 17, q = 11;
    int e = 7, d = 23, N = p * q;
    input = encryption(input, e, N);
    Decryption(input, d, N);
    return 0;
}
```

```
88
encrypt result= 11
decrypt result= 88
```

Q & A