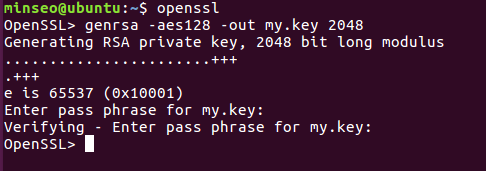
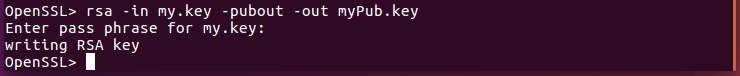
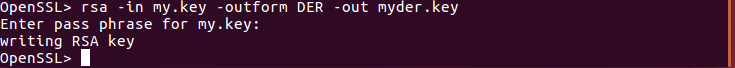
1. **Openssl** 
   1. **CA 인증서 / 클라이언트 인증서 및 클라이언트 키 파일 / 서버 인증서 및 서버 키 파일 생성하기**
      * **key 생성**



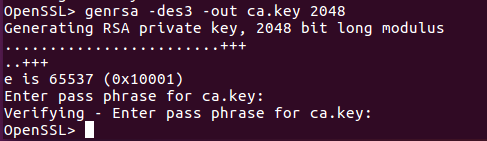
* + - **my.key 파일에서 RSA 공개키를 추출**



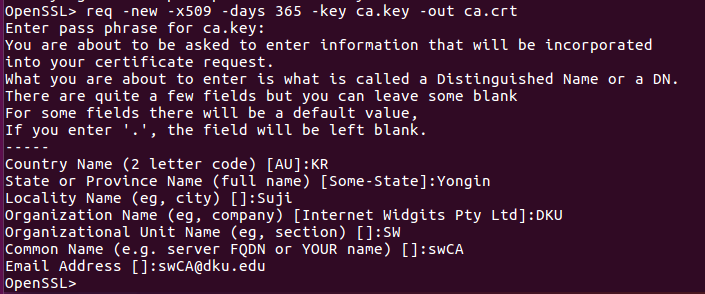
* + - **my.key 파일의 포맷을 PEM에서 DER 형식으로 변환하여 myder.key에 저장**



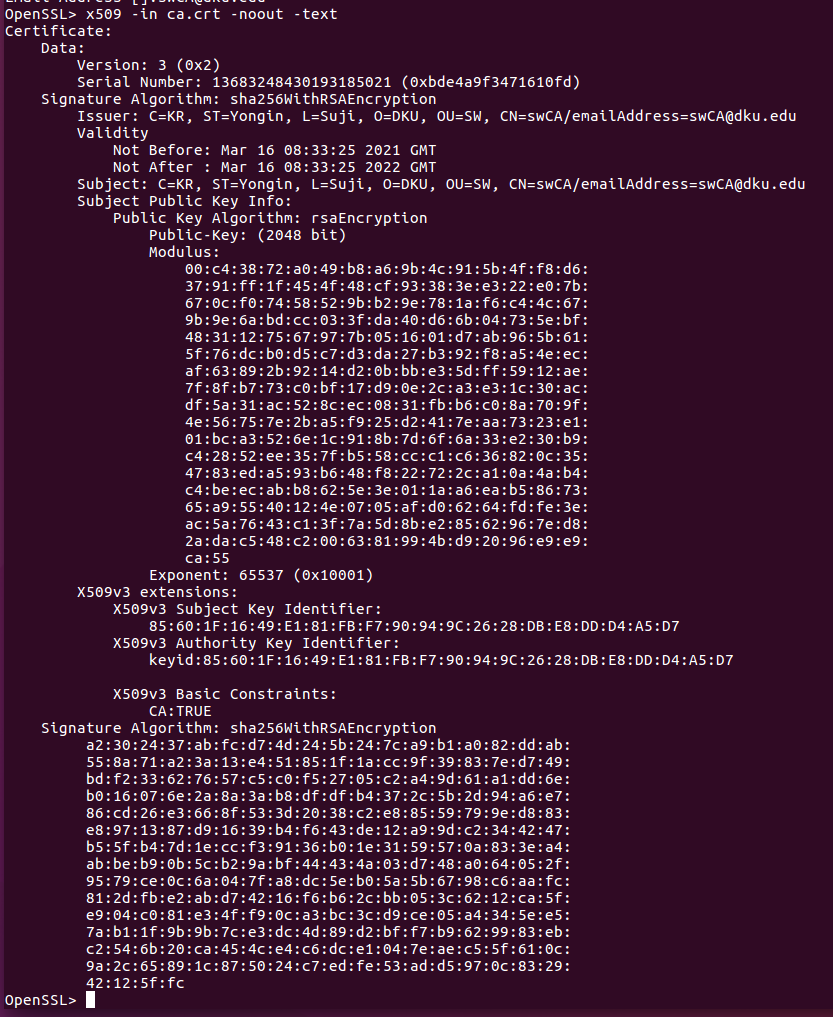
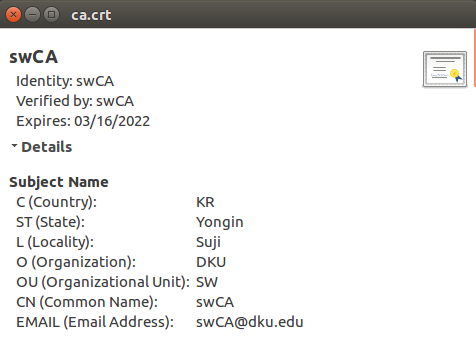
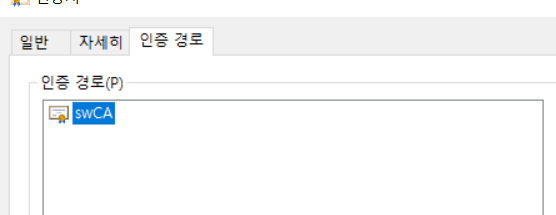
* + - **CA(인증기관) RSA개인키 생성**



* + - **자가 서명한 CA 인증서 생성**

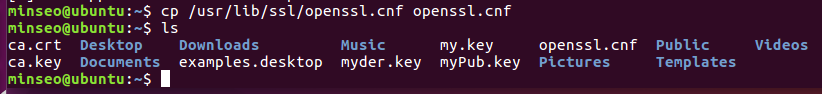


* + - **생성된 CA 인증서 보기**

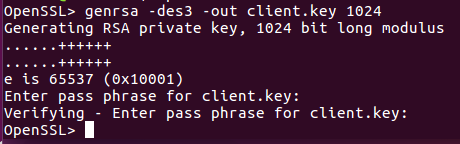


**<CA인증서>**

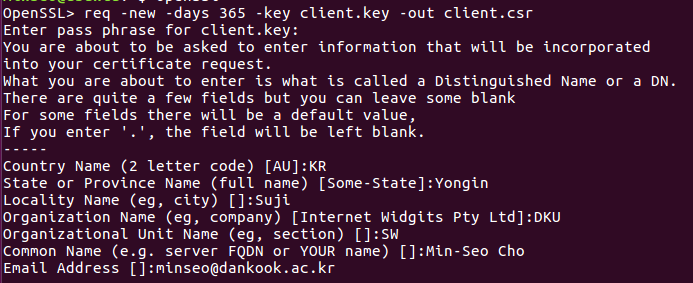
* + - **openssl.cnf 파일 복사**

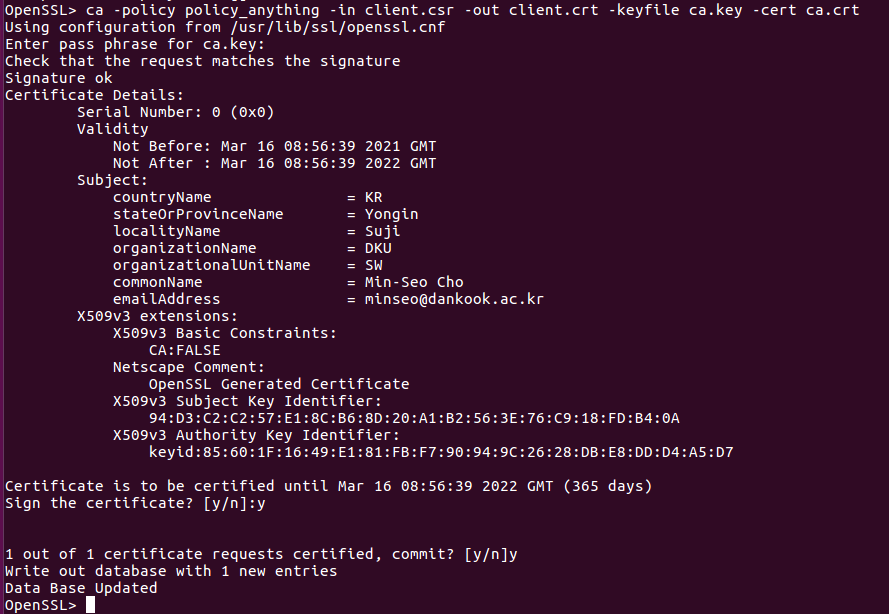


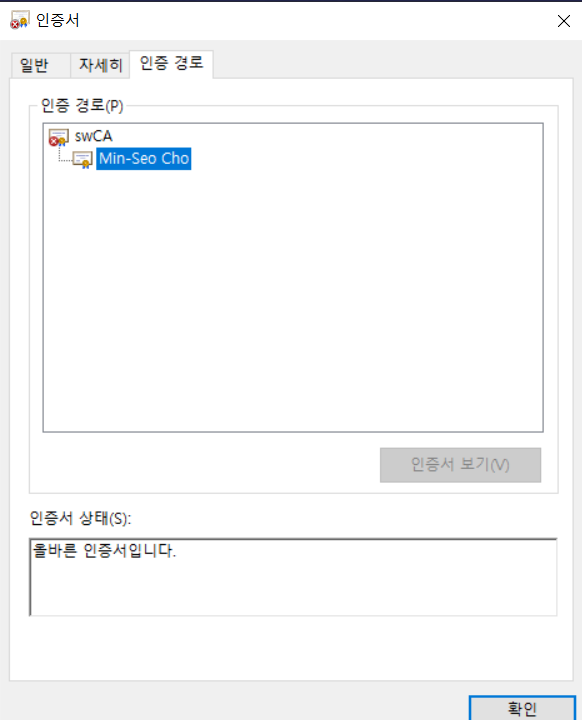
* + - **일반 사용자(client) RSA 개인키 생성**

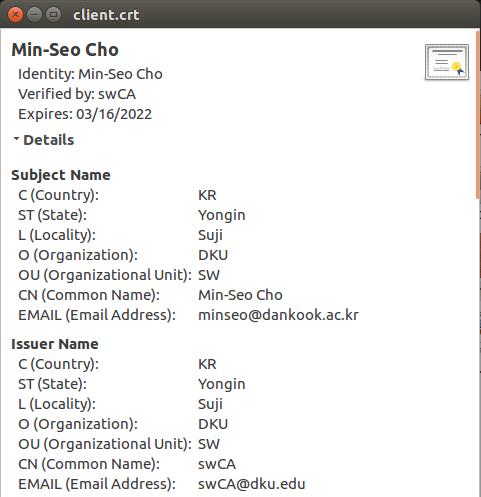


* + - **일반 사용자(client) 개인키를 이용한 CSR(인증서 서명 요청서) 생성**

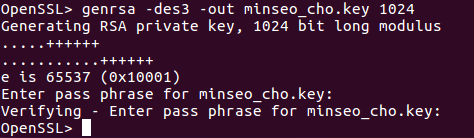


* + - **CA 개인키로 서명한 사용자(client) 인증서 생성**

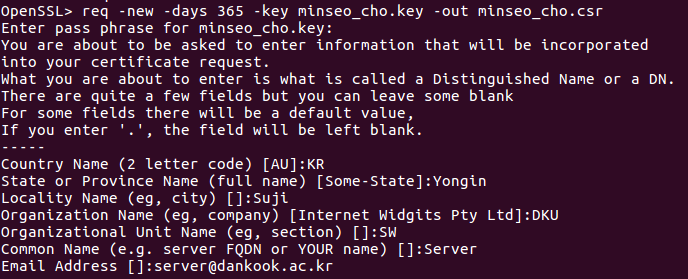
**<클라이언트 인증서>**



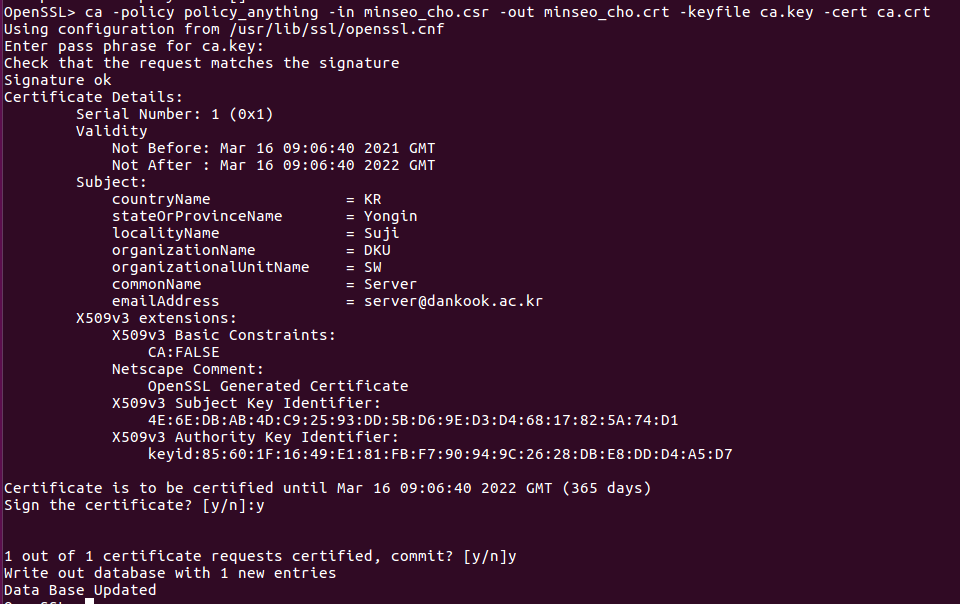
* + - **서버 RSA 개인키 생성**



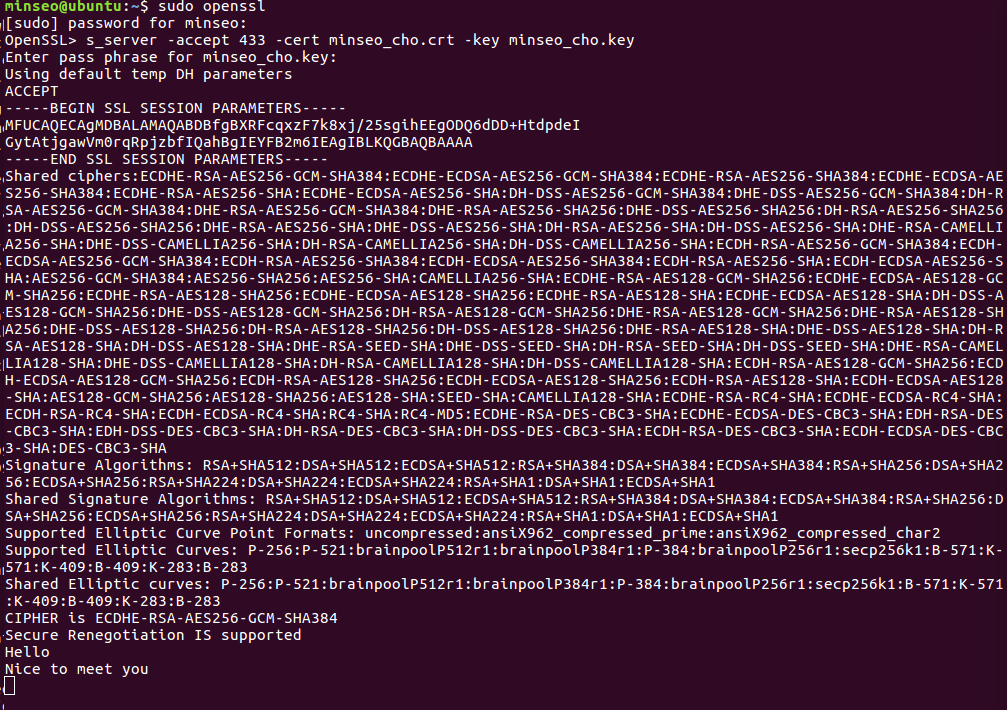
* + - **서버 개인키를 이용한 CSR(인증서 서명 요청서) 생성**



* + - **CA 인증서로 서명한 서버 인증서 생성**



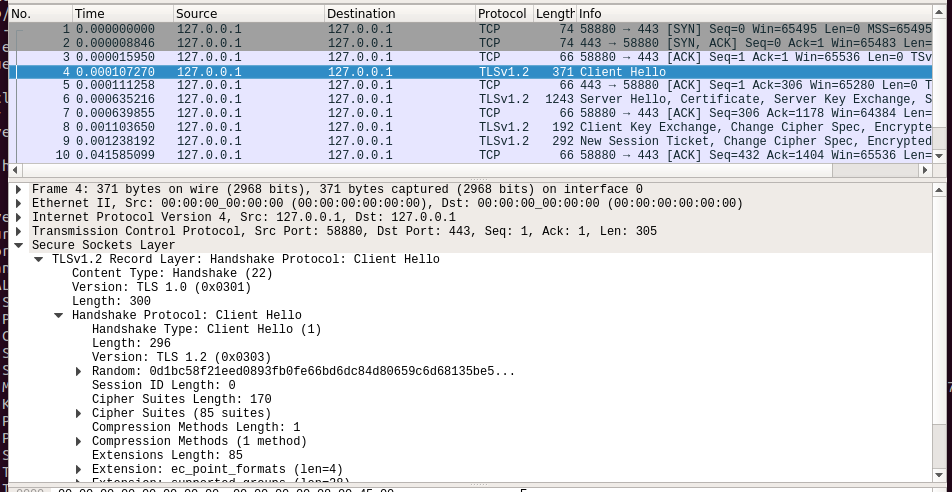
* 1. **TLS Handshake & application session실행**
     + **Server**



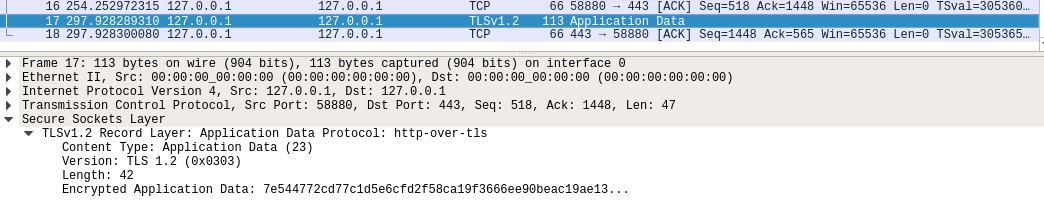
* + - **Client**



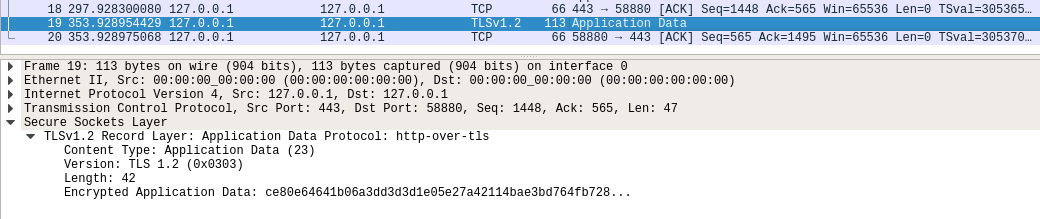
* 1. **Wireshark으로 Capture (Hello server)**



* + - **Client에서 Server로 메시지 보내기**

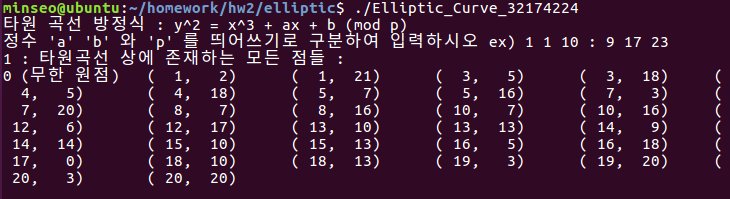


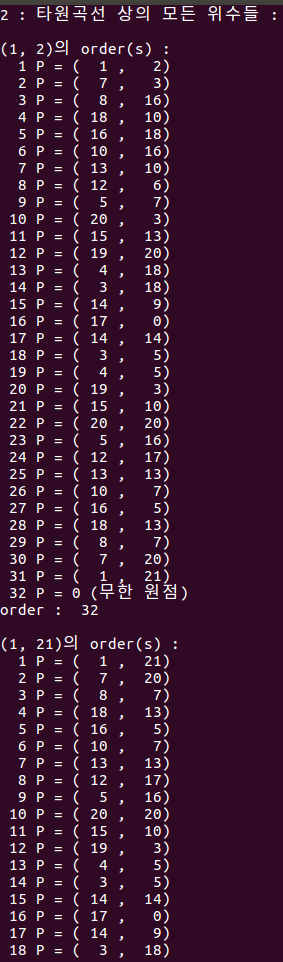
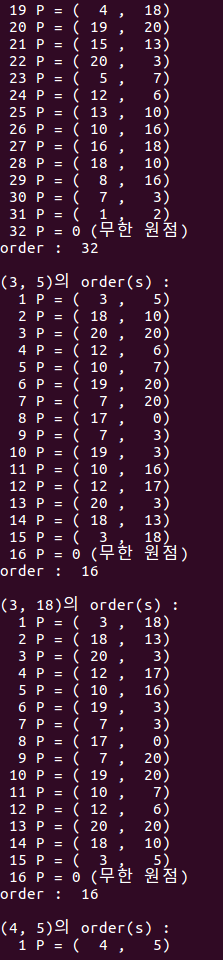
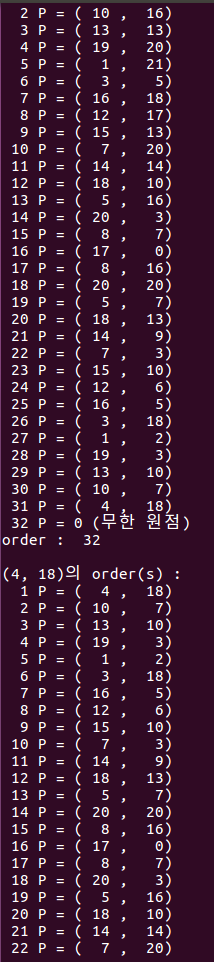
* + - **Server에서 Client로 메시지 보내기**

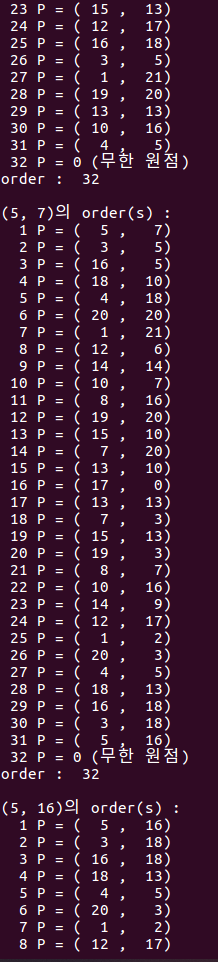
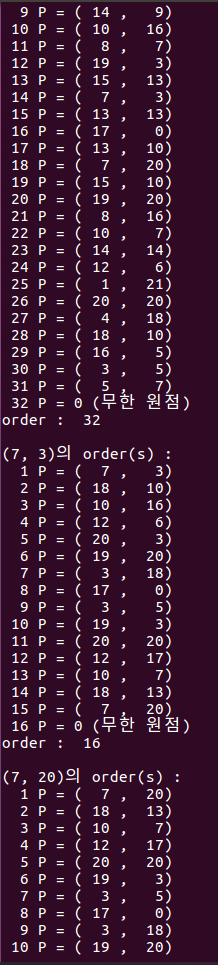
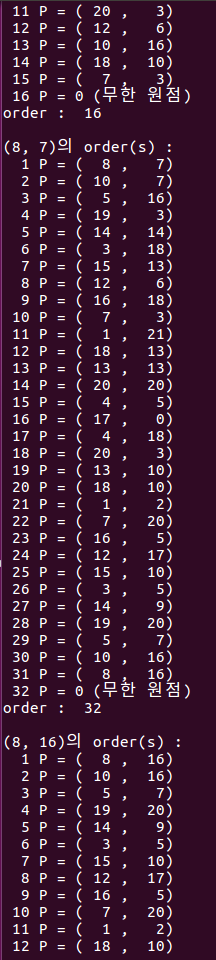


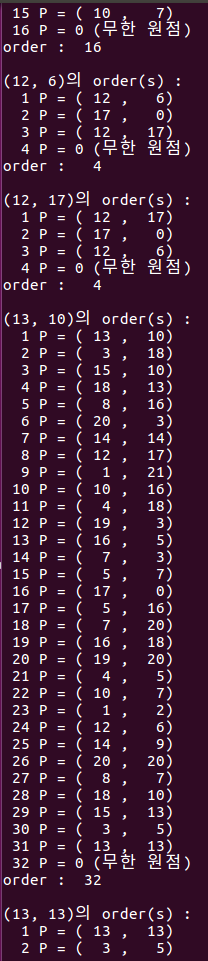
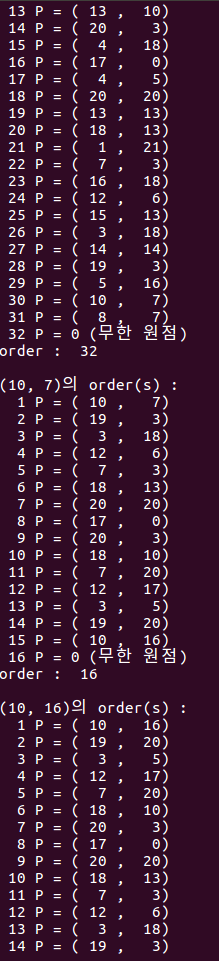
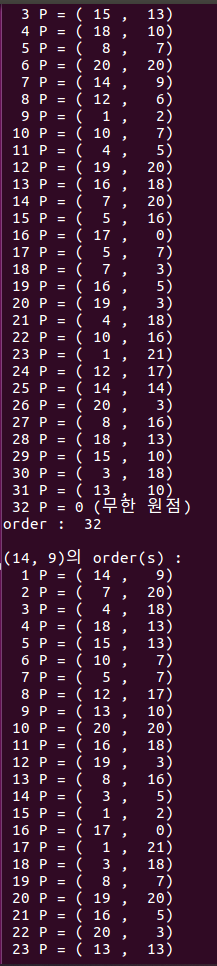
**2. Elliptic Curve Cryptosystem**

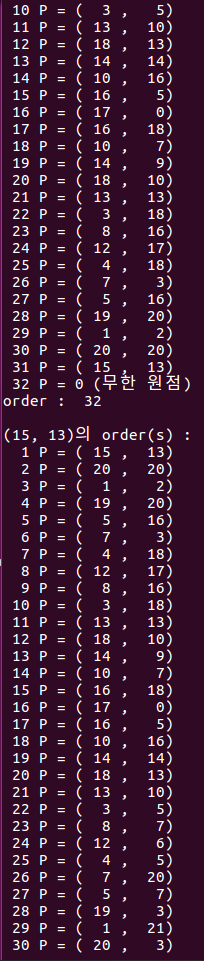
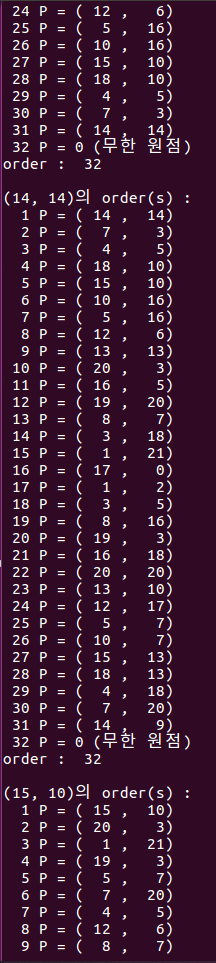
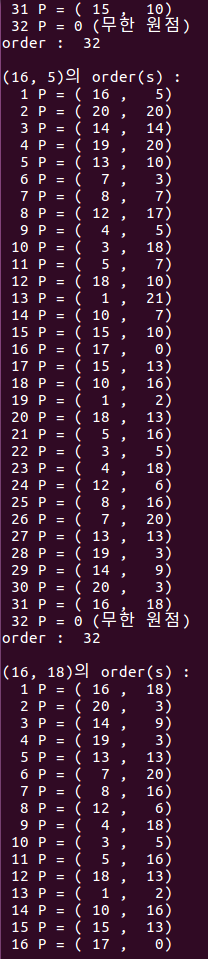
1. **무한원점을 포함하여 타원곡선 상에 존재하는 모든 점들을 나열하시오**

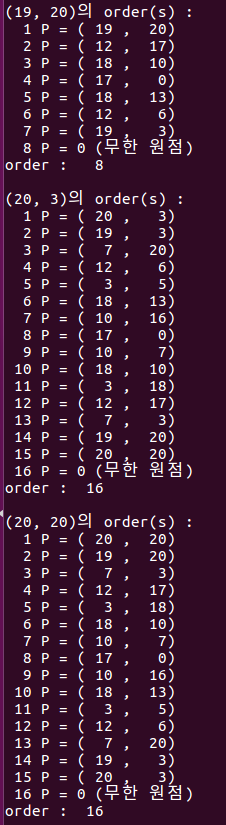
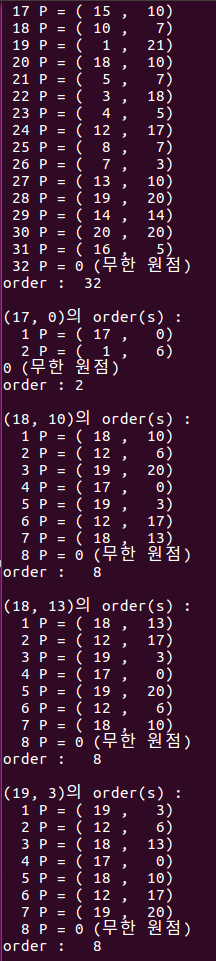


1. **모든 점들의 위수(order)계산하시오 (빨간 색 박스: 구성 원소 노란 박수: 구성원소 위수)**



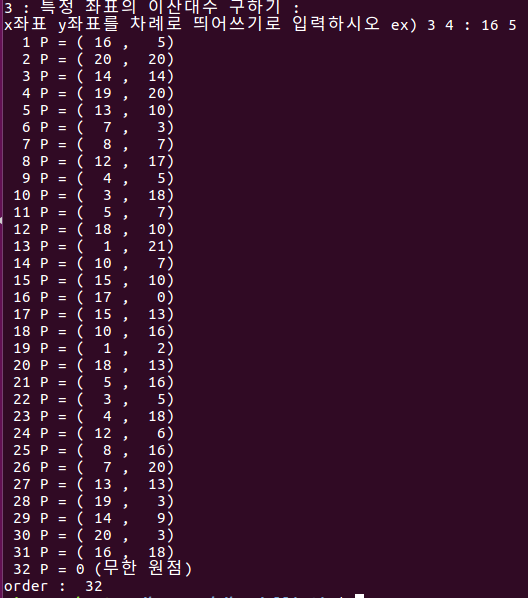






1. **Y = (4, 5) G = (16, 5)에서 이산대수x (Y=xG) 구하시오**

답 : x = 9일 때 Y = xG가 성립한다. (Y =9G)



X = 9일 때,

Y(4,5) = 9G(16,5)

1. **Elliptic curve 문제를 해결하기 위하여 사용한 코드**

**<** **Elliptic\_Curve\_32174224.c>**

#include <stdio.h>

#include <math.h>

int mod (int val, int p);

void all\_point\_orders(int a,int b, int p);

void all\_point\_list(int a,int b, int p);

void one\_point\_orders(int a,int b,int x,int y, int p);

int main(){

    int a,b,p;

    int x,y;

    printf("타원 곡선 방정식 : y^2 = x^3 + ax + b (mod p)\n");

    printf("정수 'a' 'b' 와 'p' 를 띄어쓰기로 구분하여 입력하시오 ex) 1 1 10 : ");

    scanf("%d %d %d", &a, &b, &p);

    printf("1 : 타원곡선 상에 존재하는 모든 점들 : \n");

    all\_point\_list(a,b,p);

    printf("\n2 : 타원곡선 상의 모든 위수들 : \n");

    all\_point\_orders(a,b,p);

    printf("\n3 : 특정 좌표의 이산대수 구하기 : \n");

    printf("x좌표 y좌표를 차례로 띄어쓰기로 입력하시오 ex) 3 4 : ");

    scanf("%d %d" ,&x, &y);

    one\_point\_orders(a,b,x,y,p);

    return 0;

}

int mod (int val, int p){

    //mod연산 함수

    while (1)

    {

        if(val >=p) val = val%p;

        else if(val <0) val = val +p;

        if(val >=0 && val <p) break;

    }

    return val;

}

void all\_point\_list(int a,int b, int p){

    //타원 곡선 위의 모든 점

    int x,y;

    int temp\_x3, temp\_x, temp\_y, temp, i,j;

    printf("0 (무한 원점)\t");

    for(int i=0; i< p; i++){

        x=i;

        //3차항, 1차항 연산

        temp\_x3 =mod(x\*x\*x,p);

        temp\_x = mod(a\*x,p);

        temp = mod(temp\_x3 + temp\_x +b, p);

        for(j =0; j<p; j++){

            y=j;

            temp\_y = y\*y;

            if(temp == temp\_y||temp == mod(temp\_y,p)){

                printf("(%3d, %3d)    \t",x,y);

            }

        }

    }

    printf("\n");

    return;

}

void all\_point\_orders(int a,int b, int p){

    //타원 곡선 위의 모든 점의 위수 구하기

    //all\_point\_list 와 거의 동일

    int x,y;

    int temp\_x3, temp\_x, temp\_y, temp, i,j;

    for(int i=0; i< p; i++){

        x=i;

        temp\_x3 =mod(x\*x\*x,p);

        temp\_x = mod(a\*x,p);

        temp = temp\_x3 + temp\_x +b;

        temp = mod(temp, p);

        for(j =0; j<p; j++){

            y=j;

            temp\_y = y\*y;

            if(temp == temp\_y||temp == mod(temp\_y,p)){

                printf("\n(%d, %d)의 order(s) : \n",x,y);

                //all\_point\_list와 유일한 차이점 각 점의 위수 구하는 함수 호출

                one\_point\_orders(a,b,x,y,p);

            }

        }

    }

    return;

}

void one\_point\_orders(int a,int b,int x1,int y1, int p){

    //위수를 구하는 함수

    int x2, y2, x3, y3, f\_x, f\_y, first,s\_x, s\_y, t\_x, t\_y, sub\_x;

    int inver\_x, inver\_y; // 곱셈 값의 역원

    int order = 1, i; //order는 위수 계산

    for(i=0;;i++) {

        if(order == 1) {

            if(x1 == 0 && y1 == 0) {

                printf("  1 P = (0 , 0)\n");

                printf("무한 원점 \n order : 2\n");

                return ;

            }

            f\_x = 3 \* x1 \* x1 + a;

            f\_x = mod(f\_x, p);

            for(int k=1; k<p; k++) {

                int result;

                result = 2 \* y1 \* k;

                if(mod(result, p) == 1) {

                    inver\_y = k;

                    break;

                }

            }

            s\_x = mod(f\_x \* inver\_y, p);

            t\_x=mod(s\_x \* s\_x, p);

            first = mod(2 \* x1, p);

            x3 = mod(t\_x - first, p);

            //y3

            f\_y = mod(x1 - x3, p);

            s\_y = mod(s\_x \* f\_y, p);

            y3 = mod(-y1 + s\_y, p);

            x2 = x3;

            y2 = y3;

            printf("  1 P = (%3d , %3d)\n", x1, y1);

            printf("  2 P = (%3d , %3d)\n", x2, y2);

            if(y1 == 0) {

                printf("0 (무한 원점) \norder : 2\n");

                return ;

            }

            order = order + 1;

        }

        else {

            //무한 원점 판별

            if(x1 == x2 && ((y1 + y2) % p == 0)) {

                printf("%3d P = 0 (무한 원점) \norder : %3d\n", order + 1, order + 1);

                return ;

            }

            f\_y = mod(y2 - y1, p);

            f\_x = mod(x2 - x1, p);

            for(int k = 1; k<p; k++) {

                int result;

                result = f\_x \* k;

                if(mod(result, p) == 1) {

                    inver\_x = k;

                    break;

                }

            }

            s\_x = mod(f\_y \* inver\_x, p);

            t\_x = mod(s\_x \* s\_x, p);

            x3 = mod(t\_x - x1 - x2, p);

            //y3

            sub\_x = mod(x1 - x3, p);

            s\_y = mod(s\_x \* sub\_x, p);

            y3 = mod(-y1 + s\_y, p);

            x2 = x3;

            y2 = y3;

            printf("%3d P = (%3d , %3d)\n", order + 1, x3, y3);

            order = order + 1;

        }

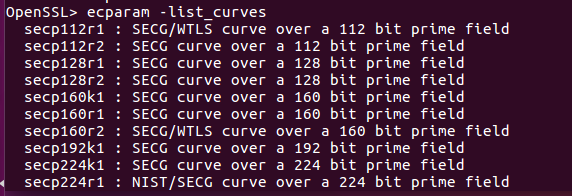
    }

}

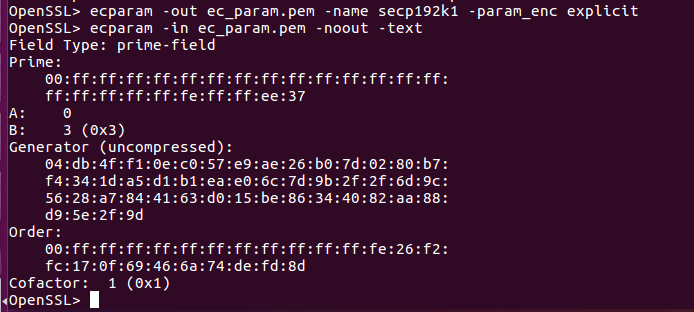
**3. Elliptic Curve Integrated Encryption Scheme(ECIES)**

**EC parameter생성 & 256 비트 타원 곡선 parameters**

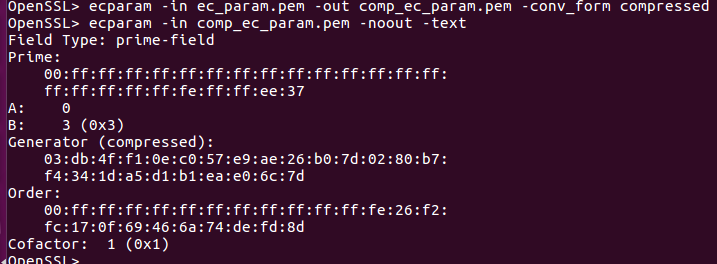
* + - **256비트 타원 곡선parameters**



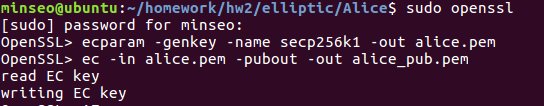
* + - **EC parameters생성**



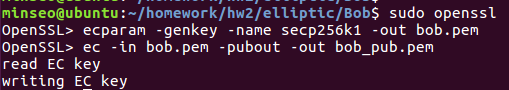
* **압축된 형태의 EC parameter**



* 1. **ECDH key 생성**
     + **Alice의 개인키/ 공개키 생성**



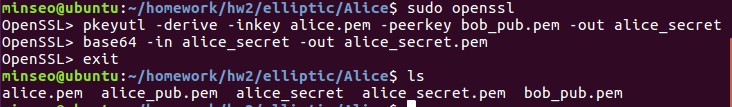
* + - **Bob의 개인키 / 공개키 생성**



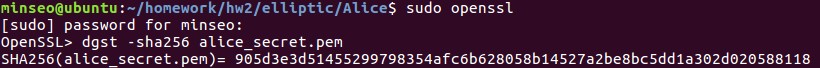
* + - **Alice에게 Bob의 공개키 전달**



* 1. **Alice의 암호화된 데이터 전송**
     + **Bob의 공개키로 K값 도출 및 확인**



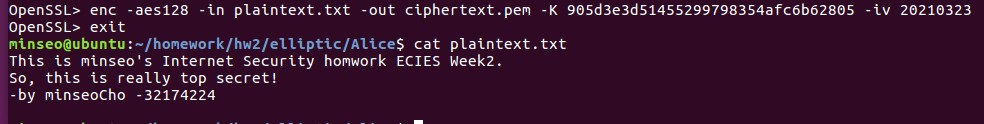
* + - **KDF 함수를 이용한 Ke와 Km 도출**



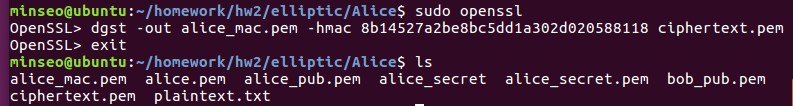
**Ke = 905d3e3d51455299798354afc6b62805**

**Km = 8b14527a2be8bc5dd1a302d020588118**

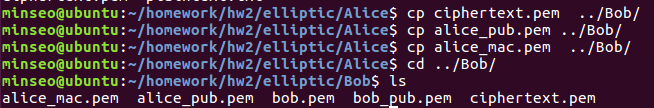
* + - **메시지 Encryption**



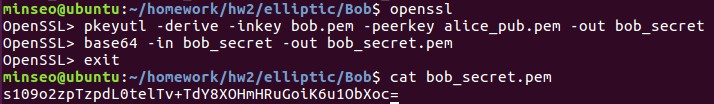
* + - **MAC값 도출**



* 1. **Bob의 암호화된 데이터 확인**
     + **Bob에게 메시지/ Alice의 공개키 /MAC값 전달**



* + - **Alice의 공개키로 K값 도출 및 확인**



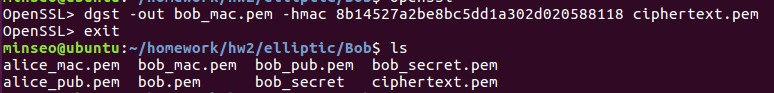
* + - **KDF 함수를 이용한 Ke와 Km 도출**



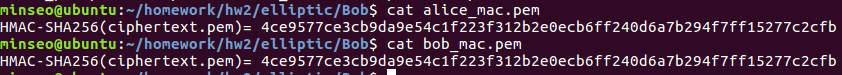
**Ke = 905d3e3d51455299798354afc6b62805**

**Km = 8b14527a2be8bc5dd1a302d020588118**

* + - **MAC값 도출**



* + - **MAC값 비교**



* + - **암호화된 메시지 확인**

