Tristan Erney
November 29th, 2021
Intro to Cryptology
Hands On Exercise 13

1)

p = 13
root = 2

primititve roots = 2, 6, 7, 11

primitive_roots.py
```
#!/usr/bin/env python

# assume all p's are prime
def primitive_roots(p):
    roots = []
    phi = p - 1
    for i in range(2, p):
        k = 0
        for j in range(1, p):
            k = i ** j % p
                if k == 1 and j != phi:
                        break
               elif k == 1 and j == phi:
                        roots.append(i)

    return roots

roots = primitive_roots(13)
print("primitive roots = {}".format(roots))
```

$dlog_{2,13}(3) \rightarrow 3 = 2^i \bmod 13 \rightarrow i = 4$
$dlog_{2,13}(11) \rightarrow 11 = 2^i \bmod 13 \rightarrow i = 7$

2)

$6^5 \pmod{11} = 10 \pmod{11}$

3)

q = 71
a = 7
$X_A = 5$
$X_B = 12$

a)

$Y_A = (a)^{XA} = 7^5 \bmod 71 = 51 \pmod{71}$

b)

$Y_B = (a)^{XB} = 7^{12} \bmod 71 = 4 \pmod{71}$

c)

$K = (Y_A)^{XB} = 51^{12} \bmod 71 = 30$
$K = (Y_B)^{XA} = 4^5 \bmod 71 = 30$

4)

$q = 11$
$a = 2$
$Y_A = 9$
$Y_B = 3$

a)

phi(11) = 10

$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 5$
$2^5 = 10$
$2^6 = 9$
$2^7 = 7$
$2^8 = 3$
$2^9 = 6$
$2^{10} = 1$

b)

$Y_A = (a)^{XA} = 2^{XA} \bmod 11 = 9 \pmod{11}$
$X_A = 6$

c)

$K = (Y_B)^{XA} = 3^6 \bmod 11 = 3 \pmod{11}$

5)

$a = 3$

$Y_A = 27 \quad \rightarrow 3^{XA} = 27 \quad \rightarrow X_A = 3$
$Y_B = 243 \quad \rightarrow 3^{XB} = 243 \quad \rightarrow X_B = 5$

$K = (Y_A)^{XB} = 27^5 = 14348907$
$K = (Y_B)^{XA} = 243^3 = 14348907$

6)

$q = 71$
$a = 7$

a)

$Y_B = 3$
$k = 2$
$M = 30$

$K = (Y_B)^k \bmod q = 3^2 \bmod 71 = 9 \pmod{71}$

$C_1 = a^k = 7^2 \bmod 71 = 49 \pmod{71}$
$C_2 = KM = 9(30) \bmod 71 = 57 \pmod{71}$

$C = (49, 57)$

b)

$M = 30$
$a = 7$
$C = (59, C_2)$

$C_1 = a^k = 7^k \bmod 71 = 59 \pmod{71}$
$k = 3$
$K = (Y_B)^k = 3^3 \bmod 71 = 27 \pmod{71}$

$C_2 = KM = 27(30) \bmod 71 = 29 \pmod{71}$