Øving K18

Oppgave 1

a) Skriv 72 og 136 på binærform

In [306]:
```
numbers = [72, 136]

for num in numbers:
    print(f'{num:<3} = {bin(num)}')
```

```
72  = 0b1001000
136 = 0b10001000
```

b) Regn ut a = 11^{72} og b = 11^{136}, begge mod 10001

In [307]:
```
def square_and_multiply(a, c, n):
    z = 1

    for j in range(c.bit_length() - 1, -1, -1):
        z = (z**2) % n
        if (c & (1 << j)) >> j:
            z = (z*a) % n

    return z
```

In [308]:
```
for num in numbers:
    print(f'11^{num:<3} mod 10001 = {square_and_multiply(11, num, 10001)}')
```

```
11^72  mod 10001 = 804
11^136 mod 10001 = 9454
```

c) Regn ut gcd(a, 10001) og gcd(b, 10001)

In [309]:
```
def gcd(a, b):
    if a == 0: return b
    return gcd(b%a, a)
```

In [310]:
```
print(f'gcd(11^72,  10001) = gcd(804,  10001) = {gcd(804, 10001)}')
print(f'gcd(11^136, 10001) = gcd(9454, 10001) = {gcd(9454, 10001)}')
```

```
gcd(11^72,  10001) = gcd(804,  10001) = 1
gcd(11^136, 10001) = gcd(9454, 10001) = 1
```

d) Regn ut ab (mod 10001)

In [311]:
```
print(f'(11^72)*(11^136) (mod 10001) = 804*9454 (mod 10001) = {(804 * 9454) % 1000
1}')
```

```
(11^72)*(11^136) (mod 10001) = 804*9454 (mod 10001) = 256
```

Oppgave 2

Sett opp et RSA-kryptosystem med følgende parametre:

- p og q skal ha minst 8 bits hver
- e skal være liten, men større enn 3

a) Skriv ut (hele) offentlig nøkkel

In [312]:
```python
p = 1087
q = 3671

n = p*q
e = 7

print(f'Public key (n, e) = ({n}, {e})')
```

```
Public key (n, e) = (3990377, 7)
```

b) Finn ved Euklids algoritme d og skriv ut (hele) private nøkkel

In [313]:
```python
def ø(q, p):
    return (q-1) * (p-1)
```

In [314]:
```python
def egcd(a, b):
    if a == 0:
        return (b, 0, 1)

    g, y, x = egcd(b % a, a)
    return (g, x - (b // a) * y, y)

def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1: return -1
    else: return x % m
```

In [315]:
```python
d = modinv(e, ø(p, q))

print(f'Private key (p, q, d) = ({p}, {q}, {d})')
```

```
Private key (p, q, d) = (1087, 3671, 1708123)
```

c) Krypter 42 og dekrypter igjen. Bruk kvadrer-og-multipliser-algoritmen for å regne ut potenser

In [316]:
```python
def encrypt(M, n, e):
    return square_and_multiply(M, e, n)
```

In [317]:
```python
M = 42
N = encrypt(M, n, e)

print(f'Message: {M} -> Encrypted: {N}')
```

```
Message: 42 -> Encrypted: 3282827
```

In [318]:
```python
def decrypt(N, p, q, d):
    return square_and_multiply(N, d, p*q)
```

In [319]:
```python
M = decrypt(N, p, q, d)

print(f'Encrypted: {N} -> Decrypted: {M}')
```

```
Encrypted: 3282827 -> Decrypted: 42
```

Oppgave 3

a) La n = 1829 og B = 5. Kan du finne en primtallsfaktor i n ved Pollard p − 1?

In [335]:
```python
def pollard_p_1(n, B):
    a = 2

    for j in range(2, B+1, 1):
        a = a**j % n
        d = gcd(a-1, n)

        if d > 1 and d < n:
            return d

    return -1
```

In [336]:
```python
n = 1829
B = 5

d = pollard_p_1(n, B)
print(f'd = {d}')
```

```
d = 31
```

b) La n_1 = 18779 og n_2 = 42583. Ved bruk av Pollard p − 1, finn B'er som er garantert å fungere for hver av disse, uten å utføre testen

Gitt at n er et oddetall, så vil største mulige faktor være gitt ved $\frac{n-1}{2}$. En B lik største felles faktor vil garantert fungere. Dermed har vi:

$$B_1 = \frac{n_1 - 1}{2} = \frac{18779 - 1}{2} = 9389$$
$$B_2 = \frac{n_2 - 1}{2} = \frac{42583 - 1}{2} = 26291$$

c) La n = 6319. Forsøk å finne en faktor i dette tallet ved Pollard p − 1. Prøv deg frem med B

In [337]:
```python
def recursive_pollard(n, B):
    d = pollard_p_1(n, B)
    if d > 0:
        return (d, B)
    return recursive_pollard(n, B+1)
```

In [338]:
```
n = 6319

d, B = recursive_pollard(n, 1)
print(f'd = {d}, B = {B}')
```

```
d = 71, B = 7
```

Oppgave 4

Finn en faktor i tallene under med $f(x) = x^2 + 1$ og startverdi $x_1 = 1$ i Pollard rho. Hvor mange iterasjoner trenger du?

In [340]:
```
def f(x):
    return x**2 + 1
```

In [382]:
```
def pollard_rho(n, x, i):
    x.append(f(x[i-1]) % n)

    if i % 2 == 0:
        d = gcd(x[i] - x[int(i/2)], n)

        if d == 1:
            return pollard_rho(n, x, i+1)
        elif d == n:
            return -1
        else:
            return d, i

    return pollard_rho(n, x, i+1)
```

a) 851

In [386]:
```
d, i = pollard_rho(851, [1], 1)
print(f'd = {d} i = {i}')
```

```
d = -37 i = 8
```

b) 1517

In [387]:
```
d, i = pollard_rho(1517, [1], 1)
print(f'd = {d} i = {i}')
```

```
d = 37 i = 8
```

c) 31861

In [389]:
```
d, i = pollard_rho(31861, [1], 1)
print(f'd = {d} i = {i}')
```

```
d = 151 i = 10
```

Oppgave 5

a) Vis følgende multiplikative egenskap til RSA:

$$e_K(x_1)e_K(x_2) \bmod n = e_K(x_1x_2) \bmod n$$

Lar offentlig nøkkel er gitt ved $K = (n, e)$. Da har vi:

$$\begin{aligned} e_K(x_1)e_K(x_2) \bmod n &= x_1^e x_2^e \bmod n \\ &= (x_1x_2)^e \bmod n \\ &= (x_1x_2 \bmod n)^e \bmod n \\ &= e_K(x_1x_2 \bmod n) \bmod n \\ &= e_K(x_1x_2) \bmod n \end{aligned}$$

b) Vis hvordan RSA er usikker mot valgt chiffertekst-angrep:

Gitt en chiffertekst y, beskriv hvordan en angriper kan velge chiffertekst $y' \neq y$, slik at kjennskap til klarteksten $x' = d_K(y')$ lar ham beregne $x = d_K(y)$

Chifferteksten y er gitt ved:

$$y = x^e \bmod n$$

Chifferteksten y' er gitt ved:

$$y' = x'^e \bmod n$$

Kan videre regne ut:

$$C = y' \cdot y$$
$$C = x'^e \cdot x^e \bmod n$$

Regner ut $d_K(C)$:

$$C^d = (x'^e \cdot x^e)^d = (x'^e)^d \cdot (x^e)^d \bmod n$$

Vet at for en hver plaintext p, har vi

$$C \equiv p^e$$
$$p \equiv C^d$$

Dette gir:

$$(p^e)^d \equiv p \bmod n$$

For C har vi da:

$$C^d = x' \cdot x \bmod n$$

Dermed kan vi regne ut x ved:

$$x = C^d \cdot x'^{-1} \bmod n$$

NB. Dette fungere kun mot skolebok-RSA. Ordentlig RSA bruker padding.

Oppgave 6

I denne oppgaven skal vi se på måte å angripe RSA på, hvis differansen $q - p$ er liten. Anta $q > p$.

a) Forklar hvorfor vi kan skrive q – p = 2d, hvor d er et heltall.

q og p velges som primtall. Primtall er alltid oddetall (for $n > 2$), og differansen mellom to oddetall vil være et partall. Dermed vil differansen $q - p$ være et partall, som kan skrives som $2d$ der $d \in \mathbb{Z}$

b) Vis at n + d^2 er et kvadrattall

Har $n - p = 2d$ og $n = pq$. Kan skrive om $n + d^2$ til:

$$n + d^2 = pq + \sqrt{\frac{q-p}{2}}$$

Som kan faktoriseres videre til:

$$\begin{aligned} &pq + \frac{q^2 - 2pq + p^2}{4} \\ &= \frac{q^2 + 2pq + p^2}{4} \\ &= \frac{(q+p)^2}{2^2} \\ &= (\frac{q+p}{2})^2 \end{aligned}$$

c) Vis hvordan vi kan faktorisere n hvis n + d^2 er et kvadrattall. Vi antar her at at d^2 er "liten nok".

Vet at $n + d^2$ er et kvadrattall, altså er $n + d^2 = m^2$. Da er m gitt ved:

$$m = \sqrt{n + d^2}$$

Videre kan vi faktorisere n:

$$\begin{aligned} n &= m^2 - d^2 \\ &= (m+d)(m-d) \\ &= pq \end{aligned}$$

Altså er:

$$\begin{aligned} p &= m + d \\ q &= m - d \end{aligned}$$

d) Faktoriser n = 152416580095517 med denne metoden.

For $d = 2$ har vi:

$$m = \sqrt{152416580095517 + 2^2} = 12345711$$

Da er p og q gitt ved:

$$p = m + d = 12345711 + 2 = 12345709$$
$$q = m - d = 12345711 - 2 = 12345713$$

Sjekker faktoriseringen:

$$n = pq = 12345709 \cdot 12345713 = 152416580095517$$