

Dima Zettel

Definitionen und Sätze

- Binomialkoeffizient: $\binom{n}{k} := \frac{n!}{k!(n-k)!}$
- $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$
- Binomische Formel: $(x+y)^n = \sum_{i=0}^n \binom{n}{i} x^{n-i} y^i$
- Kleiner Fermat: $a^p \equiv a \pmod{p}$
 - $\Leftrightarrow a^{p-1} \equiv 1 \pmod{p}$ falls $p \nmid a$
- Satz von Euler: $a^{\varphi(n)} \equiv 1 \pmod{n}$ falls $\gcd(a, n) = 1$
- $m^{k\varphi(pq)+1} \equiv m \pmod{pq}$
- Lemma von Bezout: $ax + by = d$ ist lösbar $\Leftrightarrow \gcd(a, b) \mid d$

Kombinatorik

- Permutation von n Elementen: $n!$
- Werden k Elemente der Permutation nicht unterschieden, dann durch $k!$ teilen
- Urnenmodelle
 - $RW : n^k$
 - $R|(W) : \frac{n!}{(n-k)!}$
 - $|(RW) : \binom{n+k-1}{k}$
 - $|(R)|(W) : \binom{n}{k}$

```
# returns gcd, x, y s.t. a*x + b*y = gcd(a,b)
def extended_gcd(a, b):
    x, y, lastx, lasty = 0, 1, 1, 0
    while b != 0:
        quo, a, b = a // b, b, a % b
        x, lastx, y, lasty = lastx - quo * x, x, lasty - quo * y, y
    return a, lastx, lasty

# returns x s.t. x = a_i mod m_i for all i
def crt(a : list, m : list):
    M, s = prod(m), 0
    for a_i, m_i in zip(a, m):
        M_i = M // m_i
        e_i = M_i * extended_gcd(m_i, M_i)[2]
        s += a_i * e_i
    return s % M
```

Diffie Hellman

Sei p eine große Primzahl und $g \in (\mathbb{Z}/p\mathbb{Z})^*$ öffentlich. A und B wählen geheim jeweils einen Schlüssel $a, b \in N$. A berechnet $A = g^a \pmod{p}$ und B berechnet $B = g^b \pmod{p}$. A und B tauschen A und B öffentlich aus. A berechnet $B^a \pmod{p}$ und B berechnet $A^b \pmod{p}$. Das ist das gemeinsame Geheimnis, da $B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \pmod{p}$.

RSA

- B wählt Primzahlen p, q geheim und bildet $N = pq$. B wählt $e \in [2, N]$ mit $\gcd(\varphi(N), e) = 1$. (N, e) ist der öffentliche Schlüssel von B.
- B berechnet seinen privaten Schlüssel d mit $de \equiv 1 \pmod{\varphi(N)}$ mit dem EEA. Das geht weil $\gcd(\varphi(N), e) = 1$ und B $\varphi(N) = (p-1)(q-1)$ kennt.
- A will die Nachricht m schicken. A berechnet $c = m^e \pmod{N}$ und schickt das an B.
- B kann die Nachricht mit $c^d \equiv m \pmod{N}$ entschlüsseln, da $c^d \equiv m^{ed} \equiv m^{k\varphi(N)+1} \equiv m \pmod{N}$