TECHNISCHE UNIVERSITÄT DRESDEN

Fachrichtung Mathematik · **Institut für Algebra** · Modul Math Ma ANGALG WiSe 2016/17

Dr. Jens Zumbrägel 26. Jan. 2017

6. Übungsblatt zur Angewandten Algebra

diskrete Logarithmen und Faktorisierung

1. In $G = \mathbb{Z}_{101}^*$ finde $\log_2(42)$ mit der Methode Baby-Step-Giant-Step.

2. Betrachte $\mathbb{F}_{64} = \mathbb{F}_2[X]/\langle X^6 + X + 1\rangle$. Dann ist $\alpha := [X]$ ein Erzeuger von $\mathbb{F}_{64}^*$. Betrachte $\beta := [X^4+X^3+X^2+X+1]$ und verwende den Indexkalkül mit Faktorbasis $S := \{X, X+1, X^2+X+1\}$, um $\log_\alpha \beta$ zu berechnen.

   (Weg zum Erfolg: Berechne zunächst α^6 und α^{32}, und dann $\beta\alpha^2$.)

3. Bezeichne $N_q(d)$ die Anzahl der normierten irreduziblen Polynome in $\mathbb{F}_q[X]$ vom Grad d. Begründe, dass der n-te Koeffizient der formalen Potenzreihe

$$F = \prod_{d=1}^{b} \Big(\frac{1}{1-Z^d}\Big)^{N_q(d)} \in \mathbb{Q}[[Z]]$$

   die Anzahl der b-glatten normierten Polynome vom Grad n angibt. (Erinnerung: $\frac{1}{1-Z} = 1 + Z + Z^2 + \ldots$.)

   Ein zufälliges Polynom vom Grad n zerfällt in Linearfaktoren mit Wahrscheinlichkeit

$$P = \frac{1}{n!} \prod_{i=0}^{n-1} (1 + \tfrac{i}{q}) \,.$$

4. Sei n eine zu faktorisierende Zahl.

   a) Ist $n = p \cdot q$ mit $p \approx q$, so ist die Faktorisierung einfach. Warum? — Faktorisiere die Zahl $n = 99\,997\,501$.

   b) Warum muss man beim quadratischen Sieb nur Primzahlen p mit $(\frac{n}{p}) = 1$ für die Faktorbasis berücksichtigen?

5. Faktorisiere mit dem quadratischen Sieb die Zahl $n = 29\,041$.

   (Verwende hierfür die Faktorbasis $S := \{-1, 2, 3, 5, 11\}$.)

6. Betrachte und teste den folgenden Sage-Code für die diskreten Logarithmen von Faktorbasis-Elementen in $\mathbb{F}_{2^{35}}$. Wie arbeitet die Funktion `smooth`? Teste auch andere Faktorbasis-Parameter, etwa `B=7` und `B=9`.

```
m = 35; B = 8

Polynomials.<X> = GF(2)[]; P = X^m + X^2 + 1
Field.<x> = GF(2^m, name = 'x', modulus = P)
n = 2^m - 1; alpha = X
print "n = 2^m - 1, m =", m, " alpha =", alpha
print "generator?", multiplicative_order(Field(alpha)) == n
```

```
S = []  # factor-base
for p in Polynomials.monics( max_degree = B ):
    if p.is_irreducible(): S.append(p)
N = len(S); print "#factor-base =", N

def smooth(f):
    g = f
    for p in S:
        while g % p == 0:
            g = g // p
    return g == 1

Av = []; bv = []  # lists for relations
M = 0    # relations found
while M < N+10:
    k = randint(1,n)
    t = Polynomials(Field(alpha)^k)  # t = alpha^k % P -- was too slow
    if not smooth(t): continue
    Av.append([ valuation(t, p) for p in S ])
    bv.append(k); M += 1

IMR = IntegerModRing(n)
A = Matrix(IMR, Av)
b = vector(IMR, bv)
print M, "x", N, "matrix"

z = A.solve_right(b)
print "sol: ", z
for i in range(10):
    print Field(alpha)^z[i]
```