Kryptanalyse Teil II

Alexander May

Fakultät für Mathematik
Ruhr-Universität Bochum

Wintersemester 2010/11

Pollards $p-1$ Methode

Szenario:

- Sei $N = pq$ und $p-1$ zerfalle in kleine Primfaktoren, $q-1$ nicht.
- D.h. es existieren Schranken B_1, B_2 moderater Größe, so dass
$$p - 1 = \Pi_i p_i^{e_i} \text{ mit } p_i \leq B_1 \text{ und } p_i^{e_i} \leq B_2.$$
- Für jedes $a \in \mathbb{Z}_N^*$ und jedes Vielfache k von $p-1$ gilt
$$a^k = 1 \bmod p.$$
- Falls $a^k \neq q$, dann erhalten wir $\mathrm{ggT}(N, a^k - 1) = p$.

Algorithmus Pollards $p-1$-Methode

EINGABE: $N = pq$ mit p, q gleicher Bitgröße.

1. Wähle Schranken $B_1, B_2 \in \mathbb{N}$ mit $B_2 = 2\sqrt{N}$. Wähle $a \in_R \mathbb{Z}_N^*$.
2. Für alle Primzahlen $p_i \leq B_1$ berechne $a := a^{p_i^{e_i}} \bmod N$, so dass e_i maximal ist mit $p_i^{e_i} \leq B_2$.
3. Falls $\mathrm{ggT}(a^k - 1, N) \notin \{1, N\}$, Ausgabe des ggTs.

AUSGABE: p, $q = \frac{N}{p}$ oder *Kein Faktor gefunden*.

Korrektheit der $p-1$-Methode

Satz Korrektheit der $p-1$-Methode

Sei $N = pq$ und $B_1, B_2 \in \mathbb{N}$, so dass $p-1$ B_1-glatt ist mit Primpotenzen beschränkt durch B_2, $q-1$ jedoch nicht B_1-glatt ist. Dann berechnet die $p-1$ Methode p in Zeit $\mathcal{O}(B_1 \log^3 N)$ mit Erfolgws mind. $1 - \frac{1}{B_1}$.

Beweis:

- Wir definieren $k := \prod_{\text{Primzahlen } p_i \le B_1} p_i^{e_i}$.
- Da $q-1$ nicht B_1-glatt, existiert ein Primfaktor $r \mid q-1$ mit $r > B_1$.
- Falls $r \mid \mathrm{ord}_{\mathbb{Z}_q^*}(a)$, so gilt $\mathrm{ord}_{\mathbb{Z}_q^*}(a) \nmid k$ und damit $a^k \neq 1 \bmod q$.
- Andererseits ist k aber ein Vielfaches von $p-1$.
- Daher gilt $a^k = 1 \bmod p$ und es folgt $\mathrm{ggT}(a^k, N) = p$.
- Bleibt zu zeigen, dass $r \mid \mathrm{ord}_{\mathbb{Z}_q^*}(a)$ mit hoher Ws für $a \in_R \mathbb{Z}_N^*$.
- Da $\mathbb{Z}_q^*$ zyklisch, gilt $\mathbb{Z}_q^* = \{\alpha^1, \ldots, \alpha^{q-1}\}$ für einen Generator α.
- D.h. $(a \bmod q) = \alpha^i$ für ein $i \in_R [q-1]$ und α^i besitzt

$$ord_{\mathbb{Z}_q^*}(\alpha^i) = \frac{q-1}{\mathrm{ggT}(i, q-1)}. \quad \text{(Übung)}$$

Korrektheit der $p-1$-Methode

Beweis: (Fortsetzung)

- Falls i Vielfaches von r ist, so wird Fakor r in $ord_{\mathbb{Z}_q^*}(\alpha^i)$ eliminiert.
- Dies geschieht mit Ws $\frac{1}{r}$. D.h. r verbleibt in $ord_{\mathbb{Z}_q^*}(\alpha^i)$ mit Ws
$$1 - \frac{1}{r} > 1 - \frac{1}{B_1}.$$
- **Laufzeit:** Es gibt sicherlich höchstens B_1 Primzahlen $\leq B_1$.
- Wegen $p_i^{e_i} = \mathcal{O}(B_2) = \mathcal{O}(\log N)$, kann $a^{p_i^{e_i}} \bmod N$ in jeder Iteration von Schritt 2 in Zeit $\mathcal{O}(\log^3 N)$ berechnet werden.
- Damit benötigen wir für $a^k - 1 \bmod N$ Gesamtzeit $\mathcal{O}(B_1 \log^3 N)$.

Problem der $p-1$-Methode

- Erfolgsws und Laufzeit sind abhängig von der Ordnung von $\mathbb{Z}_p^*$.
- Falls $\frac{p-1}{2}$ prim ist, so benötigen wir $B_1 \approx p$.
- D.h. in diesem Fall ist die Laufzeit nicht besser als Brute-Force.
- **Ausweg:** Bei elliptischen Kurven E variiert die Ordnung von $E \bmod p$ in einem großen Intervall, in dem glatte Zahlen liegen.

Elliptische Kurven

Definition Elliptische Kurve

Sei $p \neq 2, 3$ prim, $f(x) = x^3 + ax + b \in \mathbb{Z}_p[x]$, $4a^3 + 27b^2 \neq 0 \mod p$. Wir definieren für $f(x)$ eine *elliptische Kurve* E als

$$\{(x, y) \in \mathbb{Z}_p \mid y^2 = x^3 + ax + b\} \cup \{\mathbf{O}\},$$

wobei **O** der Punkt im Unendlichen heißt.

Anmerkungen:

- Die Bedingung $4a^3 + 27b^2$ ist äquivalent zu der Forderung, dass $f(x)$ in $\mathbb{Z}_p^*$ keine mehrfachen Nullstellen besitzt. (Übung)
- Für jeden Punkt $P = (x, y)$ auf E liegt auch $(x, -y)$ auf E.
- Wir definieren $-P = (x, -y)$.
- Für $P = \mathbf{O}$ definieren wir $-P = \mathbf{O}$ und $P + Q = Q$ für alle Q auf E.

Addition von Punkten

Algorithmus Addition von Punkten auf E

EINGABE: $P = (x_1, y_1)$, $Q = (x_2, y_2)$ auf E mit $P, Q \neq \mathbf{O}$

1. Falls $x_1 = x_2$ und $y_1 = -y_2$, Ausgabe $\mathbf{O}$.
2. Setze $\alpha := \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{für } x_1 \neq x_2 \\ \frac{3x_1^2 + a}{2y_1} & \text{für } x_1 = x_2 \end{cases}$. Setze $\beta = y_1 - \alpha x_1$.
3. Berechne $x_3 = \alpha^2 - x_1 - x_2$ und $y_3 = -(\alpha x_3 + \beta)$.

AUSGABE: $P + Q = (x_3, y_3)$

Anmerkungen:

- Sei $P \neq Q$. Wir betrachten die Gerade G durch P, Q.
- Falls $Q = -P$, so liegt G parallel zur y-Achse. Wir definieren
$$P + (-P) = \mathbf{O}.$$
- Sonst ist G definiert durch $y = \alpha x + \beta$ mit Steigung $\alpha = \frac{y_2 - y_1}{x_2 - x_1}$.
- Für $P = Q$ besitzt die Tangente im Punkt P Steigung $\alpha = \frac{3x_1^2 + a}{2y_1}$.

Addition von Punkten

Lemma Addition von Punkten auf E

Seien P, Q auf E mit $P \neq -Q$. Dann schneidet die Gerade durch P, Q die Kurve E in einem dritten Punkt R mit $-R := P + Q$.

Beweis:

- Wir zeigen nur $P \neq Q$. Der Beweis für $P = Q$ folgt analog.
- Wie zuvor setzen wir $P = (x_1, y_1)$, $Q = (x_2, y_2)$ und $R = (x_3, y_3)$.
- Sei G die Gerade $y = \alpha x + \beta$ durch P, Q. Dann gilt für $i = 1, 2$
$$(\alpha x_i + \beta)^2 = x_i^3 + a x_i + b.$$
- x_1, x_2 sind damit Nullstellen des Polynoms $g(x) = x^3 - \alpha^2 + \ldots$.
- Das Polynom $g(x)$ besitzt aber 3 verschiedene Nullstellen
$$g(x) = (x - x_1)(x - x_2)(x - x_3) = x^3 - (x_1 + x_2 + x_3)x^2 + \ldots$$
- Durch Koeffizientenvergleich folgt $x_1 + x_2 + x_3 = \alpha^2$.
- Wir erhalten $y_3 = \alpha x_3 + \beta$ und damit $-R = (x_3, -y_3)$.

Eigenschaften der Addition auf E

Korollar Effizienz der Addition

Sei E eine elliptische Kurve mit Punkten P, Q. Dann kann $P + Q$ in Laufzeit $\mathcal{O}(\log^2 p)$ berechnet werden.

- Wir benötigen nur Addition, Multiplikation und Division in $\mathbb{Z}_p$.

Satz von Mordell

Jede elliptische Kurve E bildet mit der definierten Addition eine abelsche Gruppe.

Beweis:

- Abgeschlossenheit: $P + Q$ liefert wieder einen Punkt auf E.
- Neutrales Element ist der Punkt $\mathbf{O}$.
- Inverses von $P \neq \mathbf{O}$ ist $-P$ und $-\mathbf{O} = \mathbf{O}$.
- Abelsch: Berechnung von G unabhängig von Reihenfolge P, Q.
- Assoziativität kann durch Nachrechnen gezeigt werden.

Gruppenordnung einer elliptischen Kurve

Satz von Hasse

Sei E eine elliptische Kurve über $\mathbb{F}_p$. Dann gilt

$$|E| \leq p + 1 + t \text{ mit } |t| \leq 2\sqrt{p}.$$

Anmerkungen: (ohne Beweis)

- Sei $x \in \mathbb{Z}_p$ und $f(x) = x^3 + ax + b$.
- Falls $f(x)$ ein quadratischer Rest modulo p ist, dann existieren genau zwei Lösungen $\pm y$ der Gleichung $y^2 = f(x) \bmod p$, d.h.
$$(x, y) \text{ und } (x, -y) \text{ liegen in } E.$$
- Falls $f(x)$ ein Nichtrest ist, besitzt E keinen Punkt der Form $(x, \cdot)$.
- Genau die Hälfte aller Elemente in $\mathbb{Z}_p^*$ ist ein quadratischer Rest.
- Falls $x \mapsto g(x)$ sich zufällig verhält auf $\mathbb{Z}_p$, erwarten wir $\frac{p}{2} \cdot 2 = p$ Punkte. Hinzu kommt der Punkt $\mathbf{O}$, d.h. $|E| \approx p + 1$.
- Satz von Hasse: $x \mapsto g(x)$ ist fast zufällig mit Fehler $|t| \leq 2\sqrt{p}$.

Verteilung und Berechnung der Gruppenordnung

Satz von Deuring

Sei $p \neq 2, 3$ prim. Für jedes $t \in \mathbb{Z}$, $|t| \leq 2\sqrt{p}$ ist die Anzahl der elliptischen Kurven E modulo p mit $|E| = p + 1 - t$ Punkten $\Omega\left(\frac{p^{\frac{3}{2}}}{\log p}\right)$.

Anmerkungen: (ohne Beweis)

- Die Anzahl aller Kurven E modulo p beträgt $p^2 - p$. (Übung)
- Es gibt $4\sqrt{p} + 1$ viele $t \in \mathbb{Z}$ mit $|t| \leq 2\sqrt{p}$.
- D.h. für jedes feste t gibt es durchschnittlich $\frac{p^2-p}{4p+1} = \Omega(p^{\frac{3}{2}})$ elliptische Kurven E mit Ordnung $|E| = p + 1 + t$.
- Satz von Deuring: Durchschnittsargument korrekt bis auf $\log p$.
- Sei E definiert mittels zufällig gewählter $(a, b) \in \mathbb{Z}_p^2$, $4a^3 \neq 27b^2$.
- Dann ist $|E|$ fast uniform verteilt in $[p + 1 - 2\sqrt{p}, p + 1 + 2\sqrt{p}]$.

Satz von Schoof (1985)

Für E modulo p kann $|E|$ in Zeit $\mathcal{O}(\log^8 p)$ berechnet werden.

Elliptische Kurven modulo N

Definition Elliptische Kurve über $\mathbb{Z}_n$

Sei $N \in \mathbb{N}$ mit

$\mathrm{ggT}(6, N) = 1$, $f(x) = x^3 + ax + b \in \mathbb{Z}_N[x]$ und $4a^3 + 27b^2 \neq 0 \bmod N$.

Wir definieren für $f(x)$ eine *elliptische Kurve E modulo N* als

$$\{(x, y) \in \mathbb{Z}_N \mid y^2 = x^3 + ax + b\} \cup \{\mathbf{O}\},$$

wobei $\mathbf{O}$ der Punkt im Unendlichen heißt.

- **Vorsicht:** Die Punkte von E definieren mit der zuvor definierten Addition **keine** Gruppe.
- Bsp: Sei $N = 55$ und E definiert durch $f(x) = x^3 + 1$.
- Dann liegt $P = (10, 11)$ auf E.
- Die Berechnung von $2P$ erfordert $(2y)^{-1} = 22^{-1} \bmod 55$.
- Wegen $\mathrm{ggT}(22, 55) = 11$ existiert dieses Inverse in $\mathbb{Z}_{55}$ nicht.
- D.h. E ist nicht abgeschlossen bezüglich der Addition.

Addition von Punkten auf $E \bmod N$

Algorithmus Addition von Punkten auf $E \bmod N$

EINGABE: $P = (x_1, y_1)$, $Q = (x_2, y_2)$ auf E mit $P, Q \neq \mathbf{O}$

1. Falls $x_1 = x_2$ und $y_1 = -y_2$, Ausgabe $\mathbf{O}$.
2. Berechne $d = \mathrm{ggT}(x_1 - x_2, N)$. Falls $d \notin \{1, N\}$, Ausgabe d.
3. Falls $x_1 = x_2$, $d = \mathrm{ggT}(y_1 + y_2, N)$. Falls $d > 1$, Ausgabe d.
4. Setze $\alpha := \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{für } x_1 \neq x_2 \\ \frac{3x_1^2 + a}{y_1 + y_2} & \text{für } x_1 = x_2 \end{cases}$. Setze $\beta = y_1 - \alpha x_1$.
5. Berechne $x_3 = \alpha^2 - x_1 - x_2$ und $y_3 = -(\alpha x_3 + \beta)$.

AUSGABE: $P + Q = (x_3, y_3)$ oder nicht-trivialer Teiler d von N

Addition verträglich mit zuvor definierter Addition

Satz Verträglichkeit der Additionsdefinitionen

Sei P, Q auf $E \bmod N$, so dass nicht für genau einen Teiler $p \mid N$ gilt $P + Q = \mathbf{O}$ auf $E \bmod p$. Dann ist $P + Q$ auf $E \bmod N$ identisch mit der Addition auf $E \bmod p$ und $E \bmod q$ oder liefert einen Teiler von N.

Beweis:

- Sei $P = (x_1, y_1)$ und $Q = (x_2, y_2)$.
- **Fall 1:** Sei $P + Q = \mathbf{O}$ auf $E \bmod p$ und $E \bmod q$.
- Dann gilt $\left| \begin{array}{rl} x_1 &= x_2 \\ y_1 &= -y_2 \end{array} \right| \bmod p$ und $\bmod q$ und damit auch $\bmod N$.
- Es folgt $P + Q = \mathbf{O}$ auf $E \bmod p$ und $E \bmod q$.
- Unser Algorithmus berechnet analog $P + Q = \mathbf{O}$ auf $E \bmod N$.

Addition verträglich mit zuvor definierter Addition

Beweis: (Fortsetzung)

- **Fall 2:** Sei $P + Q \neq \mathbf{O}$ auf $E \bmod p$ und $E \bmod q$.
- **Fall 2a:** $x_1 \neq x_2 \bmod p$ und $x_1 \neq x_2 \bmod q$.
- Die Additionsformel ist identisch auf $E \bmod p$ und $E \bmod N$. (analog für $E \bmod q$ und $E \bmod N$)
- **Fall 2b:** $\left| \begin{array}{ll} x_1 = x_2 & \bmod p \\ y_1 \neq -y_2 & \bmod p \end{array} \right| \text{ und } \left| \begin{array}{ll} x_1 = x_2 & \bmod p \\ y_1 \neq -y_2 & \bmod p \end{array} \right|.$
- Gleichung $y^2 = x_1^3 + ax_1 + b$ besitzt $\bmod p$ Lösungen $y_1 \neq -y_2$.
- Da wir genau 2 Lösungen $\pm y \bmod p$ erhalten, gilt $y_1 = y_2 \mod p$.
- Es folgt $y_1 + y_2 = 2y_1 \bmod p$, d.h. die Additionsformel ist identisch. (analog modulo q)
- **Fall 2c:** $x_1 \neq x_2 \bmod p$ und $\left| \begin{array}{ll} x_1 = x_2 & \bmod q \\ y_1 \neq -y_2 & \bmod q \end{array} \right|$ (und vice versa).
- Es folgt $\mathrm{ggT}(x_1 - x_2, N) = q$ in Schritt 2.

Reihenfolge der Addition auf E modulo N

Vorsicht:

- Auf E modulo N ist die Addition von Punkten nicht assoziativ.
- D.h. es kann $2P + 3P \neq P + 4P$ gelten. (Übung)

Definition Reihenfolge der Addition auf E modulo N

Sei P ein Punkt auf E modulo N. Für $m \in \mathbb{N}$ definieren wir

$$mP = \begin{cases} (m-1)P + P & \text{für } m \text{ ungerade} \\ \frac{m}{2}P + \frac{m}{2}P & \text{für } m \text{ gerade, } m > 0 \\ \mathbf{O} & \text{für } m = 0. \end{cases}$$

Anmerkung:

- mP kann in Zeit $\mathcal{O}(\log m \log^2 N)$ berechnet werden.

ECM Faktorisierungssatz

Satz ECM Faktorisierungssatz

Sei $P + Q = \mathbf{O}$ auf $E \bmod p$ und $P + Q \neq \mathbf{O}$ auf $E \bmod q$. Dann liefert die Addition $P + Q$ auf $E \bmod N$ einen Teiler von N.

Beweis:

- Wegen $P + Q = \mathbf{O}$ auf $E \bmod p$ gilt
$$x_1 = x_2 \bmod p \text{ und } y_1 = -y_2 \bmod q.$$
- Aus $P + Q \neq \mathbf{O}$ auf $E \bmod q$ folgt
$$x_1 \neq x_2 \bmod q \text{ oder } y_1 \neq -y_2 \bmod q.$$
- **Fall 1:** $x_1 \neq x_2 \bmod q$. Dann liefert Schritt 2 $\mathrm{ggT}(x_1 - x_2, N) = p$.
- **Fall 2:** $y_1 \neq -y_2 \bmod q$. Dann liefert Schritt 3 $\mathrm{ggT}(y_1 + y_2, N) = q$.

ECM Faktorisierung

Algorithmus ECM Faktorisierung

EINGABE: $N = pq$ mit p, q gleicher Bitgröße

1. Wähle Schranken $B_1, B_2 \in \mathbb{N}$.
2. Wähle $(a, x, y) \in_R \mathbb{Z}_N^3$ und berechne $b = y^2 - x^3 - ax \bmod N$.
3. Falls $\text{ggT}(4a^3 + 27b^2, N) = \begin{cases} 1 & \text{Setze } P = (x, y). \\ N & \text{Gehe zu Schritt 3.} \\ \text{sonst} & \text{Ausgabe } p, q. \end{cases}$
4. Für alle Primzahlen $p_i \leq B_1$, berechne $P := p_i^{e_i} P$ auf $E \bmod N$, wobei e_i maximal mit $p_i^{e_i} \leq B_2 + 2\sqrt{B_2} + 1$.
   Falls eine der Berechnungen scheitert, Ausgabe p, q.
5. Sonst zurück zu Schritt 3 oder Ausgabe *Kein Faktor gefunden.*

AUSGABE: p, q oder *Kein Faktor gefunden*.

Man beachte:

In Schritt 3 wird eine zufällige Kurve E mit zufälligem P auf E gewählt.

Korrektheit der ECM Faktorisierung

Satz Korrektheit der ECM Faktorisierung

Sei $N = pq$ und E eine elliptische Kurve über $\mathbb{Z}_N$, so dass $|E \bmod p|$ B_1-glatt und $|E \bmod q|$ nicht B_1-glatt ist. Dann liefert ECM die Faktorisierung von N in Zeit $\mathcal{O}(B_1 \log^3 N)$ mit Erfolgsws mind. $1 - \frac{1}{B_1}$.

Beweis:

- Wir definieren $k := \prod_{\text{Primzahlen } p_i \leq B_1} p_i^{e_i}$.
- Da $|E \bmod q|$ nicht B_1-glatt, gilt $r \mid |E \bmod q|$ für ein primes $r > B_1$.
- Falls $r \mid \mathrm{ord}_{E \bmod q}(P)$, so folgt $kP \neq \mathbf{O}$ auf $E \bmod q$.
- Andererseits ist k ein Vielfaches von $|E \bmod p|$.
- Damit gilt $kP = \mathbf{O}$ auf $E \bmod p$.
- D.h. wir erhalten bei Berechnung von kP auf $(E \bmod N)$ P', Q' mit
$$P' + Q' = \mathbf{O} \text{ auf } E \bmod p \text{ und } P' + Q' \neq \mathbf{O} \text{ auf } E \bmod q.$$
- Mit vorigem Satz liefert dies die Faktorisierung von N.
- Laufzeitanalyse und Erfolgws sind analog zur $p-1$-Methode.

Wahl der Schranken B_1, B_2 und Laufzeit

Laufzeit von ECM:

- Wir wählen B_2 so dass $B_2 \geq p$.
- Tradeoff: Kleine B_1 führen zu kleiner Laufzeit einer ECM-Iteration.
- Große B_1 erhöhen die Ws, dass $E \bmod p$ B_1-glatt ist. D.h. für große B_1 müssen weniger ECM-Iterationen durchlaufen werden.
- Optimale Wahl: $B_1 \approx L_p[\frac{1}{2}, \frac{1}{\sqrt{2}}] = e^{\frac{1}{\sqrt{2}}\sqrt{\log p \log\log p}}$.
- Unter einer Annahme für die Glattheit von Zahlen in $[p+1-2\sqrt{p}, p+1+2\sqrt{p}]$ erhalten wir Gesamtlaufzeit $L_p[\frac{1}{2}, \sqrt{2}]$.
- Besser als Laufzeit $L_N[\frac{1}{2}, 1]$ für Quadratisches Sieb falls $p \ll \sqrt{N}$.
- ECM ist die beste Methode, um kleine Primfaktoren zu finden.

Quadratische Reste und das Legendre Symbol

Definition Quadratischer Rest

Sei p prim. Ein Element $a \in \mathbb{Z}_p$ heißt *quadratischer Rest* in $\mathbb{Z}_p^*$, falls es ein $b \in \mathbb{Z}_p^*$ gibt mit $b^2 = a \bmod n$. Wir definieren

$$QR_p = \{a \in \mathbb{Z}_p^* \mid a \text{ ist ein quadratischer Rest }\} \text{ und } QNR_p = \mathbb{Z}_p^* \setminus QR.$$

Definition Legendre Symbol

Sei $p > 2$ prim und $a \in \mathbb{N}$. Das *Legendre Symbol* ist definiert als

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{falls } p|a \\ 1 & \text{falls } (a \bmod p) \in QR_p \\ -1 & \text{falls } (a \bmod p) \in QNR_p. \end{cases}$$

Berechnung von $\mathrm{dlog}_\alpha(\beta) \bmod 2$

Satz Berechnung des niederwertigsten Bits

Sei p prim, α Generator von $\mathbb{Z}_p^*$ und $\beta = \alpha^a \bmod p$. Dann gilt

$$\left(\frac{\beta}{p}\right) = \beta^{\frac{p-1}{2}} \bmod p = \begin{cases} 1 & \text{falls } a = 0 \bmod 2 \\ -1 & \text{falls } a = 1 \bmod 2 \end{cases}.$$

Beweis:

- Es gilt $\mathbb{Z}_p^* = \{\alpha, \alpha^2, \ldots, \alpha^{p-1}\}$. Damit folgt

$$QR_p = \{\alpha^2, \alpha^4, \ldots, \alpha^{2\cdot\frac{p-1}{2}}, \underbrace{\alpha^{2\cdot\frac{p+1}{2}}}_{\alpha^2}, \underbrace{\alpha^{2\cdot\frac{p+3}{2}}}_{\alpha^4}, \ldots, \underbrace{\alpha^{2(p-1)}}_{\alpha^{p-1}}\}$$

- D.h. β ist ein quadratischer Rest gdw a gerade ist.
- Es gilt $\beta^{\frac{p-1}{2}} = \pm 1$, da die 1 in $\mathbb{Z}_p^*$ Quadratwurzeln ± 1 besitzt.
- Ferner ist $\beta^{\frac{p-1}{2}} = \alpha^{\frac{a(p-1)}{2}} = 1$ gdw $\frac{a(p-1)}{2}$ Vielfaches von $p-1$.
- D.h. $\beta^{\frac{p-1}{2}} = 1$ gdw a gerade ist.

Korollar: Wir können $\mathrm{dlog}_\alpha(\beta) \bmod 2$ in Zeit $\mathcal{O}(\log^3 p)$ berechnen.

Lernen von $\mathrm{dlog}_\alpha(\beta)$ modulo Teiler von $p-1$

Idee des Pohlig Hellman Algorithmus:

- Wir nehmen an, dass die Zerlegung $p-1=\prod_{i=1}^{k} p_i^{e_i}$ bekannt ist.
- Bestimmen $a = a_i \bmod p_i^{e_i}$ für alle i. Wir ermitteln a mittels CRT.
- Zur Bestimmung von a_i verwenden wir die p_i-adische Zerlegung
$$a_i = a_{i0} + a_{i1}p_i + a_{i2}p_i^2 + \ldots + a_{ie_i-1}p_i^{e_i-1} \text{ mit } 0 \le a_{ij} < p_i.$$
- Die a_{ij} werden sukzessive für $j=0,\ldots,e_i-1$ berechnet.

Elemente in der p_i-adischen Entwicklung

Bestimmung von a_{i0}:

- Es gilt

$$\begin{aligned}\beta^{\frac{p-1}{p_i}} &= \alpha^{a\cdot\frac{p-1}{p_i}} = \alpha^{(a \bmod p_i)\cdot\frac{p-1}{p_i}} \cdot \underbrace{\alpha^{\lfloor\frac{a}{p_i}\rfloor\cdot p_i\cdot\frac{p-1}{p_i}}}_{1}\\ &= \alpha^{(a \bmod p_i)\cdot\frac{p-1}{p_i}} = \alpha^{(a_i \bmod p_i)\cdot\frac{p-1}{p_i}} = \alpha^{a_{i0}\cdot\frac{p-1}{p_i}}.\end{aligned}$$

- Wir berechnen $\alpha^{\ell\cdot\frac{p-1}{p_i}}$ für $\ell = 0, \ldots, p_i - 1$ und vergleichen mit $\beta^{\frac{p-1}{p_i}}$.

Bestimmung von a_{ij}:

- Angenommen, wir haben bereits $a_{i0}, \ldots, a_{ij-1}$ bestimmt.
- Setze $r = a_0 + \ldots + a_{ij-1}p^{j-1}$ und $\beta' := \beta \cdot \alpha^{-r}$.
- Analog zum obigen Fall berechnen wir

$$\beta'^{\frac{p-1}{p_i^{j+1}}} = \alpha^{(a-r)\cdot\frac{p-1}{p_i^{j+1}}} = \alpha^{(a-r \bmod p_i^{j+1})\cdot\frac{p-1}{p_i^{j+1}}} = \alpha^{(a_i-r \bmod p_i^{j+1})\cdot\frac{p-1}{p_i^{j+1}}} = \alpha^{a_{ij}\cdot\frac{p-1}{p_i}}.$$

- Durch Vergleich mit $\alpha^{\ell\cdot\frac{p-1}{p_i}}$, $\ell = 0, \ldots, p_i - 1$ bestimmen wir a_{ij}.

Pohlig-Hellman Algorithmus

Algorithmus Pohlig-Hellmann

EINGABE: p, α, $\beta = \alpha^a$ und $p-1 = \prod_{i=1}^{k} p_i^{e_i}$

1. FOR $i = 1, \ldots, k$ und $\ell = 0, \ldots, p_i - 1$ berechne $c_{i\ell} = \alpha^{\ell \cdot \frac{p-1}{p_i}}$.
2. FOR $i = 1, \ldots, k$ und $j = 0, \ldots, e_i - 1$
   1. Bestimme $c_{i\ell}$ mit $c_{i\ell} = \beta^{\frac{p-1}{p_i^{j+1}}}$. Setze $a_{ij} = \ell$ und $\beta := \beta \cdot \alpha^{-a_{ij} p_i^j}$.
3. Für $i = 1, \ldots, k$ berechne $a_i = a_{i0} + a_{i1} p_i + \ldots + a_{ie_i - 1} p_i^{e_i - 1}$.
4. Bestimme $a = CRT(a_1, \ldots, a_k) \bmod p - 1$.

AUSGABE: $a = \mathrm{dlog}_\alpha \beta$

Laufzeit:

- Schritt 1: $(p_1 + \ldots + p_k) \cdot \mathcal{O}(\log^3 p)$.
- Schritt 2,3,4: $(e_1 + \ldots + e_k) \cdot \mathcal{O}(\log^3 p) = \mathcal{O}(\log^4 p)$.
- D.h. wir erhalten Gesamtlaufzeit $\mathcal{O}((p_1 + \ldots + p_k) \cdot \mathcal{O}(\log^4 p))$.
- Damit ist unsere Laufzeit polynomiell falls $p_i = \mathcal{O}(\log p)$ für alle i.

Cold boot attacks

Szenario: Halderman et al 2008

- Computer wird inkorrekt runtergefahren, z.B. durch AUS-Schalter.
- DRAM erhält seinen Speicherinhalt für wenige Sekunden.
- Insbesondere stehen geheime Schlüssel im DRAM.
- Massives Kühlen erhält die Speicherinhalte stundenlang.
- Prozess induziert Ausfälle und Fehler bei einzelnen Bits.
- D.h. wir benötigen einen Algorithmus zur Ausfall-/Fehlerkorrektur.
- **Ziel:** Korrekturalgorithmen für Faktorisierung (p, q).

2-adische Faktorisierung

Algorithmus 2-adische Faktorisierung

EINGABE: $N = pq$ mit Bitlänge $2n$

- FOR i=1 to n bestimme $M = \{(p', q') \mid p'q' = N \bmod 2^n\}$.
- Für alle $(p', q') \in M$ mit Bitlänge jeweils n: Teste ob $p'q' = N$.

AUSGABE: p, q

Laufzeit:

- Sei p' ungerade. Dann ist $(p', q') \in M$ mit $q' = (p')^{-1} N \bmod 2^n$.
- Damit ist $|M| \geq 2^{n-1} = \Omega(\sqrt{N})$.
- D.h. 2-adische Faktorisierung ist nicht besser als triviales Raten.

Heninger-Shacham Algorithmus

Szenario:

- Erhalten $\tilde{p}$ mit Bits von p und Ausfällen, z.B. $\tilde{p} = 1?0??1$.

Algorithmus Heninger-Shacham

EINGABE: $N = pq$ mit Bitlänge $2n$, Bitmaterial $\tilde{p}, \tilde{q}$.

- FOR i=1 to n bestimme $M = \{(p', q') \mid p'q' = N \bmod 2^n\}$. Verwerfe solche (p', q'), die inkonsistent mit dem Bitmaterial $\tilde{p}, \tilde{q}$ sind.
- Für alle $(p', q') \in M$ mit Bitlänge jeweils n: Teste ob $p'q' = N$.

AUSGABE: p, q

Bsp: Faktorisiere $N = 10100101$ mittels $\tilde{p} = 101?$ und $\tilde{q} = 1??1$.

Satz Heninger-Shacham 2009

Sei $N = pq$ und $\tilde{p}, \tilde{q}$ beinhalten mindestens 43% der Bits, gleichverteilt über den Bitvektor. Dann kann N mit großer Ws in polynomieller Zeit faktorisiert werden.

Fehlerkorrektur

Szenario: (Henecka, May, Meurer 2010)

- Physikalische Messung liefert $\tilde{p}, \tilde{q}$ mit fehlerhaften Bits.
- Jedes Bit flippt mit bekannter Fehlerrate $\delta < \frac{1}{2}$.
- Man beachte: Für $\delta = \frac{1}{2}$ liefern $\tilde{p}, \tilde{q}$ keine Information.

Algorithmus FEHLERKORREKTUR

EINGABE: $N = pq$ mit Bitlänge $2n$, fehlerhaftes Bitmaterial $\tilde{p}, \tilde{q}$

1. Wähle t und Hamming Distanz d geeignet.
2. FOR i=1 to $\frac{n}{t}$
   1. Berechne $M = \{(p', q') \mid p'q' = N \bmod 2^{it}\}$. Verwerfe (p', q') mit Hamming-Distanz $H((p', q'), (\tilde{p}, \tilde{q})) > d$ im letzten t-Bit Fenster.
3. Für alle $(p', q') \in M$ mit Bitlänge jeweils n: Teste ob $p'q' = N$.

AUSGABE: p, q

Bsp: Faktorisiere $10100101 = 1011 \cdot 1111$ mittels $\tilde{p} = 10\mathbf{0}1$, $\tilde{q} = \mathbf{0}111$.

$(t = 2, d = 1)$

Hoeffding Schranke

Wahl von t und d:

- $|M|$ soll polynomiell beschränkt sein, d.h. $t = \mathcal{O}(\log n)$.
- Korrekte Lösung p, q darf nicht verworfen werden: t und d groß.
- Wenige inkorrekte Lösungen sollen in M verbleiben: d klein.

Satz Hoeffding

Seien $X_1, \ldots, X_{2t}$ unabhängige 0,1-wertige Zufallsvariablen mit $\mathrm{Ws}[X_i = 1] = p$. Sei $X = X_1 + \ldots\ldots X_{2t}$. Dann gilt

1. $\mathrm{Ws}[X \geq 2t(p+\gamma)] \leq e^{-4t\gamma^2}$,
2. $\mathrm{Ws}[X \leq 2t(p+\gamma)] \leq e^{-4t\gamma^2}$.

Erhalt der korrekten Lösung

Lemma Erhalt der korrekten Lösung

Sei $t = \frac{\ln n}{4\epsilon^2}$ für ein konstantes $\epsilon > 0$ und $d = 2t(\delta + \epsilon)$. Dann bleibt die korrekte Lösung in FEHLERKORREKTUR mit Ws $\geq 1 - \frac{1}{t}$ erhalten.

Beweis:

- Sei $p, q \bmod 2^{it}$ die korrekte partielle Lösung in Iteration i.
- In jeder Iteration vergleichen wir $2t$ Bits von p, q mit $\tilde{p}, \tilde{q}$.
- Definiere X_i als XOR der Bits in Position i für $i = 1, \ldots, 2t$.
- D.h. $X = X_1 + \ldots + X_{2t}$ bezeichnet die Anzahl verschiedener Bits.
- Jedes Bit kippt mit Ws δ, d.h. $E[X] = 2t \cdot E[X_i = 1] = 2t\delta$.
- Wir verwerfen (p, q) falls die Distanz zu $(\tilde{p}, \tilde{q})$ größer d ist.
- Nach Hoeffding Schranke geschieht dies pro Runde mit Ws
$$\mathrm{Ws}[X > d] = \mathrm{Ws}[X > 2t(\delta + \epsilon)] \leq e^{-4t\epsilon^2} = e^{-\ln n} = \frac{1}{n}.$$
- D.h. FEHLERKORREKTUR verwirft (p, q) nicht in $\frac{n}{t}$ Runden mit
$$\mathrm{Ws}[\mathrm{Erfolg}] \geq (1 - \tfrac{1}{n})^{\frac{n}{t}} \geq 1 - \frac{1}{t}.$$

Inkorrekte Lösungen werden eliminiert

Lemma Elimination inkorrekter Lösungen

Under der Annahme, dass sich fehlerhafte Lösungen zufällig verhalten, werden für $t = \frac{\ln n}{4\epsilon^2}$, $d = 2t(\delta + \epsilon)$ alle inkorrekte Lösungen mit großer Ws eliminiert, sofern $\delta < \frac{1}{2}(1 - \sqrt{\ln(2)}) - \epsilon \approx 0.084 - \epsilon$.

Beweis:

- Sei (p', q') inkorrekt. Wir vergleichen $2t$ Bits von p', q' und $\tilde{p}, \tilde{q}$.
- Sei X_i eine Zufallsvariable für das XOR der Bitposition.
- D.h. $X = X_1 + \ldots + X_{2t}$ ist die Anzahl der verschiedenen Bits.
- Unter unserer Annahme für (p', q') gilt $E[X] = 2t \cdot E[X_i = 1] = t$.
- Wir eliminieren (p', q') nicht, falls $X \leq d$. D.h. mit

$$\mathrm{Ws}[X \leq d] = \mathrm{Ws}[X \leq 2t(\delta + \epsilon)] = \mathrm{Ws}[X \leq 2t(\tfrac{1}{2} - \underbrace{(\frac{1}{2} - \delta - \epsilon)}_{\gamma})] \leq e^{-4t\gamma^2}.$$

- Falls $\gamma^2 > \frac{\ln 2}{4}$, so erhalten wir $\mathrm{Ws}[X \leq d] < 2^{-t}$.
- D.h. alle 2^t inkorrekten Lösungen werden mit großer Ws eliminiert.
- Wir benötigen $(\frac{1}{2} - \delta - \epsilon)^2 > \frac{\ln 2}{4}$ bzw $\delta < \frac{1}{2}(1 - \sqrt{\ln(2)}) - \epsilon$.

Fehlerkorrektur bei Faktorisierung

Satz Henecka, May, Meurer 2010

Sei $N = pq$ und $\tilde{p}, \tilde{q}$ mit Fehlerrate $\delta < 0.084 - \epsilon$ behaftet. Dann faktorisiert FEHLERKORREKTUR N mit großer Ws in Zeit $\mathcal{O}(\log^{2+\mathcal{O}(\frac{1}{\epsilon^2})} N)$.

Resultate für RSA-Schlüssel mit mehr Information

Schlüssel	Fehlerrate δ
(p, q)	0.084
(p, q, d)	0.160
(p, q, d, d_p)	0.206
(p, q, d, d_p, d_q)	0.237

Das Generalized Birthday Problem

Problem Birthday

Gegeben: L_1, L_2 Listen mit Elementen aus $\{0,1\}^n$

Gesucht: $x_1 \in L_1$ und $x_2 \in L_2$ mit $x_1 \oplus x_2 = \mathbf{0}$.

Anwendungen:

- Meet-in-the-Middle Angriffe (z.B. für RSA, ElGamal)
- Kollisionen für Hashfunktionen $h : \{0,1\}^* \rightarrow \{0,1\}^n$

Problem Generalized Birthday

Gegeben: $L_1, \ldots, L_k$ Listen mit Elementen aus $\{0,1\}^n$, unabhängig und gleichverteilt gezogen

Gesucht: $x_1 \in L_1, \ldots, x_k \in L_k$ mit $x_1 \oplus \ldots \oplus x_k = \mathbf{0}$

- Listen können auf beliebige Länge erweitert werden.
- Wir erwarten die Existenz einer Lösung sobald $|L_1| \cdot \ldots \cdot |L_k| > 2^n$.

Zusammenfügen zweier Listen

Definition Join-Operator

Wir bezeichnen mit $\mathrm{low}_\ell(x)$ die ℓ niederwertigsten Bits von x. Wir definieren für zwei Listen L_1, L_2 den Join-Operator

$$L_1 \bowtie_\ell L_2 = \{(x_1, x_2, x_1 \oplus x_2) \in L_1 \times L_2 \times \{0,1\}^n \mid \mathrm{low}_\ell(x_1) = \mathrm{low}_\ell(x_2)\}.$$

Eigenschaften:

- Es gilt $\mathrm{low}_\ell(x_1 \oplus x_2) = \mathbf{0}$ gdw $\mathrm{low}_\ell(x_1) = \mathrm{low}_\ell(x_2)$.
- Bei Eingabe L_1, L_2 kann $L_1 \bowtie L_2$ leicht berechnet werden.
- Falls $x_1 \oplus x_2 = x_3 \oplus x_4$, dann gilt $x_1 \oplus x_2 \oplus x_3 \oplus x_4 = \mathbf{0}$.
- Falls $\mathrm{low}_\ell(x_1 \oplus x_2) = \mathbf{0}$ und $\mathrm{low}_\ell(x_3 \oplus x_3) = \mathbf{0}$, dann gilt

$$\mathrm{low}_\ell(x_1 \oplus x_2 \oplus x_3 \oplus x_4) = \mathbf{0} \text{ und } \mathrm{Ws}[x_1 \oplus x_2 \oplus x_3 \oplus x_4 = \mathbf{0}] = \frac{1}{2^{n-\ell}}.$$

Algorithmus für das 4-Listen Problem

Algorithmus 4-Listen Problem

EINGABE: L_1, L_2, L_3, L_4 der Länge $|L_i| = 2^{\frac{n}{3}}$ mit Elementen aus $\{0,1\}^n$

1. Setze $\ell := \frac{n}{3}$.
2. Berechne $L_{12} = L_1 \bowtie_\ell L_2$ und $L_{34} = L_3 \bowtie_\ell L_4$.
3. Berechne $L_{1234} = L_{12} \bowtie_n L_{34}$.

AUSGABE: Elemente von L_{1234}

Korrektheit des 4-Listen Problem Algorithmus

Korrektheit:

- Elemente von L_{12}, L_{34} erfüllen $\mathrm{low}_{\frac{n}{3}}(x_1 \oplus x_2) = \mathrm{low}_{\frac{n}{3}}(x_3 \oplus x_4) = \mathbf{0}$.
- Wir erwarten Listenlänge $E[|L_{12}|] = \sum_{(x_1,x_2)\in L_1 \times L_2} \mathrm{Ws}[\mathrm{low}_{\frac{n}{3}}(x_1 \oplus x_2) = \mathbf{0}] = \frac{|L_1|\cdot|L_2|}{2^{\frac{n}{3}}} = 2^{\frac{n}{3}}$.
- Analog gilt $E[|L_{34}|] = 2^{\frac{n}{3}}$.
- Elemente von L_{1234} erfüllen $x_1 \oplus x_2 \oplus x_3 \oplus x_4 = \mathbf{0}$.
- Die erwartete Listenlänge $E[|L_{1234}|]$ von L_{1234} ist

$$\sum_{(x_1,\ldots,x_4)\in L_{12} \times L_{34}} \mathrm{Ws}[x_1 \oplus \ldots \oplus x_4 = \mathbf{0} \mid \mathrm{low}_{\frac{n}{3}}(x_1 \oplus x_2) = \mathrm{low}_{\frac{n}{3}}(x_3 \oplus x_4)]$$
$$= \frac{E(|L_{12}|)\cdot E(|L_{34}|)}{2^{\frac{2n}{3}}} = 1.$$

- D.h. wir erwarten, dass L_{1234} mindestens eine Lösung enthält.

Laufzeitanalyse des 4-Listen Problem Algorithmus

Laufzeit und Speicherplatz:

- Die Listen $L_1, \ldots, L_4, L_{12}, L_{34}$ benötigen jeweils Platz $\tilde{\mathcal{O}}(2^{\frac{n}{3}})$.
- Die Konstruktion von L_{12}, L_{34} geht in Laufzeit $\tilde{\mathcal{O}}(2^{\frac{n}{3}})$.
- Konstruktion von L_{1234} benötigt ebenfalls Laufzeit $\tilde{\mathcal{O}}(2^{\frac{n}{3}})$.
- **Gesamt:** Zeit und Platz $\tilde{\mathcal{O}}(2^{\frac{n}{3}})$

Übungen: Modifizieren Sie den Algorithmus, so dass

- $\mathrm{low}_\ell(x_1 \oplus x_2) = \mathrm{low}_\ell(x_3 \oplus x_4) = c$ für ein $c \in \{0,1\}^\ell$.
- wir $x_1 \oplus x_2 \oplus x_3 \oplus x_4 = c'$ für ein $c' \in \{0,1\}^n$ lösen können.
- wir jede Instanz mit $k \geq 4$ in Zeit und Platz $\tilde{\mathcal{O}}(2^{\frac{n}{3}})$ lösen können.

4-Listen Problem in $\mathbb{Z}_{2^n}$

Ziel: Verwende Gruppe $(\mathbb{Z}_{2^n}, +)$ statt $(\{0,1\}^n, \oplus) = (\mathbb{F}_{2^n}, +)$.

Sei $-L = \{-x \in \mathbb{Z}_{2^n} \mid x \in L\}$.

Algorithmus 4-Listen Problem

EINGABE: L_1, L_2, L_3, L_4 mit Elementen aus $\{0,1\}^n$ der Länge $|L_i| = 2^{\frac{n}{3}}$

1. Setze $\ell := \frac{n}{3}$.
2. Berechne $L_{12} = L_1 \bowtie_\ell -L_2$ und $L_{34} = L_3 \bowtie_\ell -L_4$.
3. Berechne $L_{1234} = L_{12} \bowtie_n -L_{34}$.

AUSGABE: Elemente von L_{1234}

Korrektheit:

- Wir erhalten $(x_1, x_2, x_1 + x_2) \in L_{12}$ mit $x_1 + x_2 = 0 \bmod 2^\ell$.
- Man beachte: Für $x_1 + x_2 = 0 \bmod 2^\ell$ und $x_3 + x_4 = 0 \bmod 2^\ell$ gilt

$$x_1 + x_2 + x_3 + x_4 = 0 \bmod 2^\ell.$$

Algorithmus k-Listen Problem, $k = 2^m$

Algorithmus k-Listen Problem

EINGABE: $L_1, \ldots, L_{2^m}$ mit Elementen aus $\{0,1\}^n$, Länge $|L_i| = 2^{\frac{n}{m+1}}$

1. Setze $\ell := \frac{n}{m+1}$.
2. For $i := 1$ to $m-1$
   1. FOR $j := 1$ to 2^m step 2^i
      /* Join aller benachbarten Listen auf Level i des Baumes */
   2. Berechne $L_{j\ldots j+2^i-1} = L_{j\ldots j+2^{i-1}-1} \bowtie_\ell L_{j+2^{i-1}\ldots j+2^i-1}$.
3. Berechne $L_{1\ldots 2^m} = L_{1\ldots 2^{m-1}} \bowtie_n L_{2^{m-1}+1\ldots 2^m}$.

AUSGABE: Elemente von $L_{1\ldots 2^m}$

Beispiel für $k = 2^3$:

- Join für $i = 1$: $L_{12} = L_1 \bowtie_\ell L_2$, $L_{34} = L_3 \bowtie_\ell L_4, \ldots, L_{78} = L_7 \bowtie_\ell L_8$.
- Join für $i = 2$: $L_{1234} = L_{12} \bowtie_\ell L_{34}$, $L_{5678} = L_{56} \bowtie_\ell L_{78}$.
- Join in Schritt 3: $L_{1\ldots 8} = L_{1\ldots 4} \bowtie_n L_{5\ldots 8}$.

Analyse des k-Listen Algorithmus

Korrektheit:

- Alle Startlisten besitzen Länge 2^ℓ.
- D.h. durch das Join auf unterster Ebene entstehen Listen mit erwarteter Länge $\frac{2^\ell \cdot 2^\ell}{2^\ell} = 2^\ell$.
- Damit entstehen in Schritt 2 stets Listen mit erwarteter Länge 2^ℓ.
- In Schritt 3 entsteht eine Liste $L_{1\ldots k}$ mit erwarteter Länge

$$\sum_{(x_1,\ldots,x_k)} \mathrm{Ws}[x_1 \oplus \ldots \oplus x_k = \mathbf{0} \mid \mathrm{low}_{(m-1)\ell}(x_1 \oplus \ldots \oplus x_{\frac{k}{2}}) = \mathrm{low}_{(m-1)\ell}(x_{\frac{k}{2}+1} \oplus x_k)] = \frac{2^{2\ell}}{2^{n-(m-1)\ell}} = 1.$$

Analyse des k-Listen Algorithmus

Laufzeit und Platz:

- Die Listen $L_1, \ldots, L_k$ besitzen benötigen jeweils Platz $\tilde{\mathcal{O}}(2^\ell)$.
- In Schritt 2 berechnen wir $k-2$ Listen mit erwarteter Länge $\tilde{\mathcal{O}}(2^\ell)$.
- Damit erhalten wir Speicherplatzbedarf $\tilde{\mathcal{O}}(k2^\ell) = \tilde{\mathcal{O}}(k2^{\frac{n}{\log k+1}})$.
- Die Laufzeit für alle $k-1$ Join-Operationen beträgt $\tilde{\mathcal{O}}(2^\ell)$.
- Damit ist die Gesamtlaufzeit ebenfalls $\tilde{\mathcal{O}}(k2^\ell) = \tilde{\mathcal{O}}(k2^{\frac{n}{\log k+1}})$
- Für $k = 2^{\sqrt{n}}$ erhalten wir Zeit und Speicherplatz Komplexität

$$\tilde{\mathcal{O}}(2^{\sqrt{n}} \cdot 2^{\frac{n}{\sqrt{n}+1}}) = \tilde{\mathcal{O}}(2^{2\sqrt{n}})$$

- Dies ist eine subexponentielle Funktion in n.

Übung: Konstruieren Sie einen Algorithmus für $k = 2^m + j, 0 < j < 2^m$ mit Komplexität $\tilde{\mathcal{O}}(k2^{\frac{n}{\log k+1}})$.

Offenes Problem:
Geht es für $k = 2^m + j$ besser? Für $k = 3$ besser als $\tilde{\mathcal{O}}(2^{\frac{n}{2}})$?

Urbild Angriff auf Inkrementelle Hashfunktionen

AdHash Konstruktion: (Bellare, Micciancio 1997)

- Hashe Nachricht $x = (x_1, \ldots, x_k)$ als
$$H(x) = \sum_{i=1}^{k} h(i, x_i) \bmod M.$$
- **Inkrementell:** Block x_i kann leicht durch x_i' ersetzt werden.
- NASD Instantiierung: $M = 2^{256}$

Algorithmus: Urbild Angriff auf AdHash

EINGABE: Modul $M = 2^{256}$, Hashwert c

1. Generiere Listen $L_1, \ldots, L_k$ mit $|L_i| = 2^{\frac{n}{\log k+1}}$.
2. Liste L_i enthält $y_j^{(i)} = h(i, x_j)$ für zufällig gewählte x_j.
3. k-Listen Algorithmus liefert $y_{j_1}^{(1)}, \ldots, y_{j_k}^{(k)}$ mit
$$y_{j_1}^{(1)} + \ldots + y_{j_k}^{(k)} = c \bmod 2^{256} \text{ und } y_{j_i}^{(i)} = h(i, x_{j_i}).$$

AUSGABE: $x = (x_{j_1}, \ldots, x_{j_k})$ mit $H(x) = c \bmod M$

Urbild Angriff auf Inkrementelle Hashfunktionen

Komplexität:

- Naive Urbildberechnung benötigt erwartet 2^{256} H-Auswertungen.
- Für unseren Angriff ist der k-Listen Algorithmus laufzeitbestimmend.
- Auswerten von $k \cdot 2^{\frac{n}{\log k+1}}$ für $k = 128$ liefert $2^7 \cdot 2^{32} = 2^{39}$.
- Allgemein: Erhalten einen Angriff mit Komplexität $\tilde{\mathcal{O}}(2^{2\sqrt{\log M}})$.
- D.h. für 80-Bit Sicherheit muss $M > 2^{1600}$ gewählt werden.

Fälschen von einfachen Ringsignaturen

Idee: Ringsignatur

- Sei $U = \{u_1, \ldots, u_k\}$ eine Menge von Usern.
- Ein User u_i möchte eine Unterschrift im Namen von U leisten.
- Eine Ringsignatur schützt die Anonymität von u_i in U.

Ringsignatur von Back (1997)

Sei H eine Hashfunktion.

1. **Gen:** Generiere RSA Schlüssel (N_i, e_i, d_i) für alle User u_i.
2. **Sign:** User u_i wählt $m_j \in_R \mathbb{Z}_{N_j}$, $j \neq i$, Nachricht m, und berechnet
$$m_i = \left(H(m) \oplus \bigoplus_{j \neq i} (m_j^{e_j} \bmod N_j)\right)^{d_i} \bmod N_i.$$
Ausgabe von (m, σ) mit der Signatur $\sigma = (m_1, \ldots, m_k)$.
3. **Vrfy:** Prüfe für (m, σ) die Identität
$$\bigoplus_{i=1}^{k} (m_i^{e_i} \bmod N_i) \stackrel{?}{=} H(m).$$

Fälschen von Ringsignaturen

Algorithmus Universelles Fälschen von Ringsignaturen

EINGABE: Nachricht m, (N_i, e_i) für $i = 1, \ldots, k$

1. Berechne Listen L_i für $i = 1, \ldots, k$ mit Elementen
$$x_j^{(i)} = m_j^{e_i} \bmod N_i \text{ für } m_j \in_R \mathbb{Z}_{N_i}.$$
2. k-Listen Algorithmus liefert $x_{j_1}^{(1)}, \ldots, x_{j_n}^{(n)}$ mit
$$x_{j_1}^{(1)} \oplus \ldots \oplus x_{j_n}^{(n)} = H(m).$$

AUSGABE: (m, σ) mit $\sigma = (m_{j_1}, \ldots, m_{j_n})$.

Komplexität:

- Sei $N = \max_i\{N_i\}$. Wir erhalten Komplexität $\mathcal{O}(k \cdot 2^{\frac{\log N}{\log k+1}})$.
- D.h. für $k = \theta(\log N)$ erhalten wir einen subexponentiellen Angriff.

Polynomielle Vielfache mit kleinem Gewicht

Definition Gewicht eines Polynoms

Sei $p(x) = \sum_{i=0}^{n} p_i x^i \in \mathbb{F}_2[x]$. Das *Gewicht* $w(p)$ von $p(x)$ ist definiert als das Hamminggewicht des Koeffizientenvektors von $p(x)$, d.h.

$$w(p) = w((p_0, \ldots, p_n)).$$

- Betrachten Korrelationsattacken auf Stromchiffren.
- Diese benötigen Polynomvielfache sehr kleinen Gewichts.

Problem Polynomvielfache mit kleinem Gewicht

Gegeben: $p(x) \in \mathbb{F}_2[x]$ irreduzibel vom Grad n, Gradschranke $d > n$, Gewicht k

Gesucht: $m(x) \in \mathbb{F}_2[x]$ mit $p(x) \mid m(x)$, Grad $\leq d$ und $w(m) \leq k$.

Konstruktion von Polynomvielfachen

Wir identifizieren Polynome in $\mathbb{F}_2[x]$ mit ihren Koeffizientenvektoren.

Algorithmus Polynomvielfache

EINGABE: $p(x) \in \mathbb{F}_2[x]$, Gewicht k

1. Setze die Gradschranke $d := 2^{\frac{n}{\log k+1}}$
2. Generiere Listen L_i, $i = 1, \ldots, k$ mit Elementen der Form
$$y_j^{(i)} = x^{a_j} \bmod p(x) \text{ für zufällig gewählte } a_j \leq d.$$
3. k-Listen Algorithmus liefert $y_{j_1}^{(1)}, \ldots, y_{j_k}^{(k)}$ mit
$$y_{j_1}^{(1)} \oplus \ldots \oplus y_{j_k}^{(k)} = \mathbf{0}.$$

AUSGABE: $m(x) = x^{a_{j_1}} + \ldots + x^{a_{j_k}}$

Konstruktion von Polynomvielfachen

Korrektheit:

- Wir definieren $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/p(x)$. Addition zweier Polynome in $\mathbb{F}_{2^n}$ entspricht dem XOR ihrer Koeffizientenvektoren.
- Damit gilt $m(x) = x^{a_{j_1}} + \ldots + x^{a_{j_k}} = 0$ in $\mathbb{F}_{2^n}$, d.h. $p(x)$ teilt $m(x)$.
- Wegen $a_j \leq d$ besitzt $m(x)$ Grad höchstens d.
- Ferner besteht $m(x)$ aus k Monomen, d.h. besitzt Gewicht k.
- Für die Listengröße im k-Listen Alg. benötigen wir $d = 2^{\frac{n}{\log k+1}}$.
- D.h. unser Algorithmus funktioniert nur für hinreichend großes d.

Komplexität:

- Der k-Listen Algorithmus liefert Komplexität $\tilde{\mathcal{O}}(k \cdot 2^{\frac{n}{\log k+1}})$.
- Bsp.: $\mathrm{grad}(p) = 120$ und wir suchen Vielfaches mit Gewicht $k = 4$.
- Wir wählen $d = 2^{\frac{n}{\log k+1}} = 2^{\frac{120}{3}} = 2^{40}$ erhalten $k \cdot 2^{\frac{n}{\log k+1}} = 2^{42}$.

k-Listen Problem über $\mathbb{F}_{2^n}$ für $k \geq n$

Problem Generalized Birthday für $k \geq n$

Gegeben: $L_1, \ldots, L_k$ mit Elementen aus $\{0,1\}^n$, $|L_i| \geq 2$, $k \geq n$.
Gesucht: $\mathbf{x}_1 \in L_1, \ldots, \mathbf{x}_k \in L_k$ mit $\mathbf{x}_1 \oplus \ldots \oplus \mathbf{x}_k = \mathbf{0}$

Idee: (Algorithmus von Bellare, Micciancio 1997)

- ObdA $L_i = \{\mathbf{x}_{i,0}, \mathbf{x}_{i,1}\}$ für alle i, sonst entferne Elemente aus L_i.
- Definiere $b_i = \begin{cases} 0 & \text{falls } \mathbf{x}_{i,0} \text{ in } L_i \text{ ausgewählt wird.} \\ 1 & \text{falls } \mathbf{x}_{i,1} \text{ in } L_i \text{ ausgewählt wird.} \end{cases}$
- D.h. wird müssen $\mathbf{b} = (b_1, \ldots, b_n) \in \{0,1\}^n$ finden mit
$$b_1\mathbf{x}_{1,1} + (1-b_1)\mathbf{x}_{1,0} + \ldots + b_n\mathbf{x}_{n,1} + (1-b_n)\mathbf{x}_{n,0} = \mathbf{0}$$
$$\Leftrightarrow \quad b_1(\mathbf{x}_{1,1} - \mathbf{x}_{1,0}) + \ldots + b_n(\mathbf{x}_{n,1} - \mathbf{x}_{n,0}) = -(\mathbf{x}_1 + \ldots + \mathbf{x}_n)$$
- Dies ist ein lineares Gleichungssystem in den b_i.
- Falls die Matrix definiert durch die Vektoren $\mathbf{x}_{i,1} - \mathbf{x}_{i,0}$ vollen Rang besitzt, so können wir das System in Zeit $\mathcal{O}(n^3 + kn)$ lösen.

Affine Varietät

Definition Affine Varietät

Seien $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ für einen Körper $\mathbb{F}$. Wir bezeichnen

$$\mathbf{V}(f_1, \ldots, f_m) = \{(a_1, \ldots, a_n) \in \mathbb{F}^n \mid f_i(a_1, \ldots, a_n) = 0 \text{ für } i = 1, \ldots, m\}$$

als die durch $f_1, \ldots, f_m$ definierte *affine Varietät*.

Anmerkungen:

- $\mathbf{V}(f_1, \ldots, f_m)$ ist die gemeinsame Nullstellenmenge von $f_1, \ldots, f_m$.
- Für Beispiele verwenden wir oft den Körper $\mathbb{F} = \mathbb{R}$, für die Kryptographie $\mathbb{F} = \mathbb{F}_p$.

Beispiele:

- $\mathbf{V}(x^2 + y^2 - 1)$ ist in $\mathbb{R}^2$ der Einheitskreis mit Mittelpunkt $\mathbf{0}$.
- $\mathbf{V}(x^2 + y^2 - z^2)$ liefert im $\mathbb{R}^3$ einen Doppelkegel.
- $\mathbf{V}(y - x^2, z - x^3)$ liefert als Schnitt zweier Flächen eine Kurve.
- $\mathbf{V}(xz, yz)$ ist die Vereinigung der (x, y)-Ebene mit der z-Achse.

Spezialfall Lineare Varietät

Definition Lineare Varietät

Sei $A \in \mathbb{F}^{m \times n}$ und $\mathbf{b} \in \mathbb{F}^m$. Dann definieren die Lösungen $\mathbf{V} = \{\mathbf{x} \in \mathbb{F}^n \mid A\mathbf{x} = \mathbf{b}\}$ eine *lineare Varietät*.

Anmerkungen:

- Sei $\mathrm{rang}(A) = r$. Dann besitzt $\mathbf{V}$ Dimension $n - r$. D.h. $\dim(\mathbf{V})$ wird von der Anzahl linear unabhängiger Gleichungen bestimmt.

Mehr Ziele:

1. **Lösbarkeit:**
   Gilt $\mathbf{V}(f_1, \ldots, f_m) \neq \emptyset$, d.h. ist $f_1 = \ldots = f_m = 0$ lösbar?
2. **Endlichkeit:**
   Ist $\mathbf{V}(f_1, \ldots, f_m)$ endlich? Können wir alle Lösungen bestimmen?

Abgeschlossenheit unter Vereinigung und Schnitt

Satz Abgeschlossenheit unter Vereinigung und Schnitt

Seien V, W affine Varietäten. Dann sind auch $V \cap W$ und $V \cup W$ affine Varietäten.

Beweis:

- Seien $V = \mathbf{V}(f_1, \ldots, f_m)$ und $W = \mathbf{W}(g_1, \ldots, g_\ell)$. Sei $\mathbf{x} \in V \cap W$.
- Dann verschwindet $\mathbf{x}$ sowohl auf $f_1, \ldots, f_m$ als auch auf $g_1, \ldots, g_\ell$.
- Damit verschwindet $\mathbf{x}$ auf $f_1, \ldots, f_m, g_1, \ldots, g_\ell$, d.h.
$$V \cap W = \mathbf{V}(f_1, \ldots, f_m, g_1, \ldots, g_\ell).$$
- Wir zeigen weiterhin: $V \cup W = \mathbf{V}(f_i g_j \mid i = 1, \ldots, m, j = 1, \ldots, \ell)$.
- $V \cup W \subseteq \mathbf{V}(f_i g_j)$: Sei $\mathbf{x} \in V \cup W$, oBda $\mathbf{x} \in V$.
- Dann verschwindet $\mathbf{x}$ auf allen f_i und damit auf allen $f_i g_j$.
- $\mathbf{V}(f_i g_j) \subseteq V \cup W$: Sei $\mathbf{x} \in \mathbf{V}(f_i g_j)$.
- Falls $\mathbf{x} \in V$, gilt $\mathbf{x} \in V \cup W$. Sonst folgt $f_{i'}(\mathbf{x}) \neq 0$ für ein $i' \in [m]$.
- Andererseits verschwindet $\mathbf{x}$ auf allen $f_{i'} g_j$.
- Damit verschwindet $\mathbf{x}$ auf allen g_j. D.h. es gilt $\mathbf{x} \in W$.

Ideal

Definition Ideal

Eine Menge $I \subseteq \mathbb{F}[x_1, \ldots, x_n]$ heißt *Ideal* falls Folgendes gilt.

1. $0 \in I$.
2. Falls $f, g \in I$, dann ist $f + g \in I$.
3. Für $f \in I$ und $h \in \mathbb{F}[x_1, \ldots, x_n]$ gilt $hf \in I$.

Definition Polynomideal

Seien $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$. Dann bezeichnen wir mit

$$\langle f_1, \ldots, f_m \rangle = \{\textstyle\sum_{i=1}^m h_i f_i \mid h_i \in \mathbb{F}[x_1, \ldots, x_n]\}$$

das *von* $f_1, \ldots, f_m$ *generierte Ideal*.

Anmerkung: $I = \langle f_1, \ldots, f_m \rangle$ ist ein Ideal.

- Sei $I = \langle f_1, \ldots, f_m \rangle$. $0 \in I$ wegen $0 = \sum_i 0 \cdot f_i$.
- Seien $f = \sum_i p_i f_i$, $g = \sum q_i f_i \in I$ und $h \in \mathbb{F}[x_1, \ldots, x_n]$. Dann gilt
$$f + g = \textstyle\sum_i (p_i + q_i) f_i \in I \ \text{ und } \ hf = \sum_i (hp_i) f_i \in I.$$

Varietäten und Ideale

Definition Basis eines Ideals

Ein Ideal I heißt *endlich erzeugt mit Basis* $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$, falls $I = \langle f_1, \ldots, f_m \rangle$.

Satz Varietäten hängen nur vom Ideal ab

Seien $f_1, \ldots, f_m$ und $g_1, \ldots, g_\ell$ Basen eines Ideals I. Dann gilt

$$\mathbf{V}(f_1, \ldots, f_m) = \mathbf{V}(g_1, \ldots, g_\ell).$$

Beweis:

- Zeigen $\mathbf{V}(f_1, \ldots, f_m) \subseteq \mathbf{V}(g_1, \ldots, g_\ell)$. Umkehrung folgt analog.
- Sei $\mathbf{x} \in \mathbf{V}(f_1, \ldots, f_m)$. D.h. $f_i(\mathbf{x}) = 0$ für alle $i = 1, \ldots, m$.
- Da die f_i eine Basis von I bilden, können wir jedes g_j schreiben als
  $$g_j = \sum_{i=1}^m h_i f_i \text{ für } j = 1, \ldots, \ell.$$
- Damit gilt $g_j(\mathbf{x}) = \sum_i h_i(\mathbf{x}) \cdot f_i(\mathbf{x}) = 0$. D.h. $x \in \mathbf{V}(g_1, \ldots, g_\ell)$.

Bsp: Es gilt $\langle 2x^2 + 3y^2 - 11, x^2 - y^2 - 3 \rangle = \langle x^2 - 4, y^2 - 1 \rangle$ (Übung), d.h. $\mathbf{V}(2x^2 + 3y^2 - 11, x^2 - y^2 - 3) = \mathbf{V}(x^2 - 4, y^2 - 1) = \{(\pm 2, \pm 1)\}$.

Das Ideal einer Varietät

Frage: Welche Polynome verschwinden auf $\mathbf{V}(f_1, \ldots, f_m)$?

Definition Ideal einer Varietät

Sei V eine affine Varietät. Dann ist das Ideal von V definiert als

$$\mathbf{I}(V) = \{f \in \mathbb{F}[x_1, \ldots, x_n] \mid f(\mathbf{x}) = 0 \text{ für alle } x \in V\}.$$

Satz $\mathbf{I}(V)$ ist ein Ideal

Sei V eine affine Varietät. Dann ist $\mathbf{I}(V)$ ein Ideal.

Beweis:

- $0 \in \mathbf{I}(V)$, da das Nullpolynom auf allen Punkten verschwindet.
- Seien $f, g \in \mathbf{I}(V)$ und $h \in \mathbb{F}[x_1, \ldots, x_n]$. Für $\mathbf{x} \in V$ folgt
$$\underbrace{f(\mathbf{x})}_{=0} + \underbrace{g(\mathbf{x})}_{=0} = 0 \text{ und } h(\mathbf{x}) \cdot \underbrace{f(\mathbf{x})}_{=0} = 0.$$
- Damit gilt $f + g \in \mathbf{I}(V)$ und $hf \in \mathbf{I}(V)$.

Beispiel: Ideal einer Varietät

Bsp Ideal einer Varietät

$\mathbf{I}(\{(0,0)\}) = \langle x, y\rangle \subseteq \mathbb{F}[x,y]$.

Beweis:

- $\langle x, y\rangle \subseteq \mathbf{I}(\{(0,0)\})$: Sei $f \in \langle x, y\rangle$. Dann gilt
$$f(x,y) = h_1(x,y)\cdot x + h_2(x,y)\cdot y.$$
- Damit ist $f(0,0) = 0$ und es folgt $f \in \mathbf{I}(\{(0,0)\})$.
- $\mathbf{I}(\{(0,0)\}) \subseteq \langle x, y\rangle$: Sei $f \in \mathbf{I}(\{(0,0)\})$. Dann gilt
$$f(x,y) = \sum_{i,j} a_{ij}x^i y^j \text{ mit } f(0,0) = 0.$$
- Es folgt $a_{00} = 0$ und damit
$$f(x,y) = \left(\sum_{i,j,i>0} a_{ij}x^{i-1}y^j\right)\cdot x + \left(\sum_{j>0} a_{0j}y^{j-1}\right)\cdot y \in \langle x, y\rangle.$$

Polynome → Varietät → Ideal

Frage: Gilt $\langle f_1, \ldots, f_m \rangle = \mathbf{I}(\mathbf{V}(f_1, \ldots, f_m))$? Antwort: Leider nicht.

Satz

Es gilt $\langle f_1, \ldots, f_m \rangle \subset \mathbf{I}(\mathbf{V}(f_1, \ldots, f_m))$, aber i. Allg. keine Gleichheit.

Beweis:

- Sei $f \in \langle f_1, \ldots, f_m \rangle$, d.h. $f = \sum_{i=1}^{n} h_i f_i$ für Polynome h_i.
- Die Polynome $f_1, \ldots, f_m$ verschwinden auf allen $\mathbf{x} \in \mathbf{V}(f_1, \ldots, f_m)$.
- Damit gilt $f(\mathbf{x}) = 0$ für $\mathbf{x} \in \mathbf{V}(f_1, \ldots, f_m)$, d.h. $f \in \mathbf{I}(\mathbf{V}(f_1, \ldots, f_m))$.

- **Gegenbeispiel** für Gleichheit: $\mathbf{I}(\mathbf{V}(x^2, y^2)) \not\subseteq \langle x^2, y^2 \rangle$.
- Die Gleichungen $x^2 = y^2 = 0$ implizieren $\mathbf{V}(x^2, y^2) = \{(0,0)\}$.
- Aus dem Beispiel zuvor folgt $\mathbf{I}(\mathbf{V}(x^2, y^2)) = \mathbf{I}(\{(0,0)\}) = \langle x, y \rangle$.
- Es gilt aber $\langle x, y \rangle \not\subseteq \langle x^2, y^2 \rangle$, da z.B. x nicht in der Form $h_1 \cdot x^2 + h_2 \cdot y^2$ dargestellt werden kann.

Beziehung zwischen Varietäten und ihren Idealen

Satz

Seien $V, W \subseteq \mathbb{F}^n$ affine Varietäten. Dann gilt

1. $V \subseteq W$ gdw $\mathbf{I}(W) \subseteq \mathbf{I}(V)$.
2. $V = W$ gdw $\mathbf{I}(V) = \mathbf{I}(W)$.

Beweis:

- $\Rightarrow$: Sei $V \subseteq W$ und $f \in \mathbf{I}(W)$.
- Dann verschwindet f auf allen $\mathbf{x} \in W$ und damit auf allen $\mathbf{x} \in V$.
- Damit folgt $f \in \mathbf{I}(V)$.
- $\Leftarrow$: Sei $\mathbf{I}(W) \subseteq \mathbf{I}(V)$.
- Sei die affine Varietät W definiert durch die Polynome $f_1, \ldots, f_m$.
- Dann gilt $f_1, \ldots, f_m \in \mathbf{I}(W) \subseteq \mathbf{I}(V)$.
- D.h. $f_1, \ldots, f_m$ verschwinden insbesondere auf den Punkten aus V.
- Da W aus *allen* gemeinsamen Nst. der f_i besteht, folgt $V \subseteq W$.
- 2 folgt aus 1: $V = W$ gilt gdw $V \subseteq W$ und $W \subseteq V$ gdw $V = W$.

Interessante Probleme

Ziel: Löse die folgenden Probleme algorithmisch.

1. **Basisdarstellung:**
   Stelle jedes Ideal I mittels einer endlichen Basis $\langle f_1, \ldots, f_m \rangle$ dar.
2. **Idealzugehörigkeit:**
   Entscheide, ob f im Ideal $\langle f_1, \ldots, f_m \rangle$ liegt.
3. **Lösbarkeit von polynomiellen Gleichungssystemen:**
   Bestimme alle gemeinsamen Lösungen von

$$\left| \begin{array}{ccc} f_1 & = & 0 \\ & \vdots & \\ f_m & = & 0 \end{array} \right| .$$

Polynomdivision

Definition führender Term

Sei $f = a_m x^m + \ldots + a_0 \in \mathbb{F}[x]$. Dann bezeichnen wir den *führenden Term* von f mit $LT(f) = a_m x^m$.

Anmerkung:

- Für $f, g \in \mathbb{F}[x]$ gilt: $\mathrm{grad}(f) \leq \mathrm{grad}(g) \Leftrightarrow \mathrm{LT}(f)$ teilt $\mathrm{LT}(g)$.

Algorithmus Polynomdivision

EINGABE: $f, g \in \mathbb{F}[x]$ mit $\mathrm{grad}(g) < \mathrm{grad}(f)$

1. Setze $q := 0$ und $r := f$.
2. WHILE ($r \neq 0$ und $\mathrm{LT}(g)$ teilt $\mathrm{LT}(r)$)
    1. Setze $q := q + \frac{\mathrm{LT}(r)}{\mathrm{LT}(g)}$ und $r - \frac{\mathrm{LT}(r)}{\mathrm{LT}(g)} \cdot g$.

AUSGABE: q, r mit $\mathrm{grad}(r) < \mathrm{grad}(g)$ und $f = qg + r$

Invariante: $f = qg + r = (q + \frac{\mathrm{LT}(r)}{\mathrm{LT}(g)}) \cdot g + r - \frac{\mathrm{LT}(r)}{\mathrm{LT}(g)} \cdot g$.

Jedes Ideal in $\mathbb{F}[x]$ wird von einem Polynom erzeugt.

Satz Jedes Ideal in $\mathbb{F}[x]$ ist ein Hauptideal.

Für jedes Ideal I in $\mathbb{F}[x]$ gilt $I = \langle f \rangle$ für ein $f \in \mathbb{F}[x]$, wobei f eindeutig ist bis auf Multiplikation mit Konstanten ungleich Null.

Beweis:

- Sei $I = \{0\}$, dann gilt $I = \langle 0 \rangle$.
- Andernfalls wähle $f \in I \setminus \{0\}$ minimalen Grads.
- Behauptung: $I = \langle f \rangle$. Es gilt $\langle f \rangle \subseteq I$, da I ein Ideal ist.
- $I \subseteq \langle f \rangle$: Sei $g \in I$ beliebig. Wir berechnen q, r mit $g = qf + r$.
- Da I ein Ideal ist, gilt $qf \in I$ und ferner $r = g - qf \in I$.
- Wegen $\deg(r) < \deg(f)$, folgt $r = 0$ aufgrund der Minimalität von f.
- Daher gilt $g = qf \in \langle f \rangle$.

Jedes Ideal in $\mathbb{F}[x]$ wird von einem Polynom erzeugt.

Beweis der Eindeutigkeit:

- Angenommen $\langle f \rangle = \langle g \rangle$.
- Aus $f \in \langle g \rangle$ folgt $f = hg$ für ein $h \in \mathbb{F}[x]$.
- Damit gilt $\mathrm{grad}(f) = \mathrm{grad}(h) + \mathrm{grad}(g)$, d.h. $\mathrm{grad}(g) \leq \mathrm{grad}(f)$.
- Vertauschen von f und g liefert analog $\mathrm{grad}(f) \leq \mathrm{grad}(h)$.
- Damit gilt $\mathrm{grad}(g) = \mathrm{grad}(f)$ und f, g unterscheiden sich durch Multiplikation mit einem konstanten Polynom h, $\mathrm{grad}(h) = 0$.

Definition Hauptideal

Ein Ideal, das von einem Polynom erzeugt wird, heißt *Hauptideal*.

Problem:

Wie finden wir z.B. im Hauptideal $\langle x^4 - 1, x^6 - 1 \rangle$ einen Generator?

Der ggT ist ein Generator

Satz ggT ist Generator

Seien $f, g \in \mathbb{F}[x]$. Dann gilt $\langle f, g \rangle = \langle \mathrm{ggT}(f, g) \rangle$.

Beweis:

- Jedes Ideal I in $\mathbb{F}[x]$ ist ein Hauptideal.
- D.h. $I = \langle f, g \rangle = \langle h \rangle$ für ein $h \in \mathbb{F}[x]$.
- Der Generator h ist ein gemeinsamer Teiler von f, g, da $f, g \in \langle h \rangle$.
- Um zu zeigen, dass $h = \mathrm{ggT}(f, g)$, müssen wir zeigen, dass jeder gemeinsame Teiler von f, g auch h teilt und h somit der ggT ist.
- Sei p ein beliebiger gemeinsamer Teiler von f, g.
- D.h. $f = ap$ und $g = bp$ für $a, b \in \mathbb{F}[x]$.
- Wegen $h \in \langle f, g \rangle$ existieren $c, d \in \mathbb{F}[x]$ mit $h = cf + dg$. Es folgt
$$h = cap + dbp = (ca + dp)p.$$
- Damit teilt p das Polynom h, und es muss $h = \mathrm{ggT}(f, g)$ gelten.

Beispiele für Basisdarstellung und Idealzugehörigkeit

Bsp Basisdarstellung:

- Wir berechnen einen Generator von $I = \langle x^4 - 1, x^6 - 1 \rangle$.
- Der Euklidische Algorithmus für Polynome liefert
$$\mathrm{ggT}(x^4 - 1, x^6 - 1) = x^2 - 1.$$
- Damit gilt $I = \langle x^2 - 1 \rangle$.

Bsp Idealzugehörigkeit:

- Sei $I = \langle x^3 - 3x + 2, x^4 - 1, x^6 - 1 \rangle$. Ist $x^2 + 2x + 1 \in I$?
- Es gilt $\mathrm{ggT}(x^3 - 3x + 2, x^4 - 1, x^6 - 1) = x - 1$. D.h. $I = \langle x - 1 \rangle$.
- Division mit Rest liefert $x^2 + 2x + 1 = (x + 3)(x - 1) + 4$.
- D.h. $x^2 + 2x + 1$ ist nicht in I, da es nicht von $x - 1$ geteilt wird.

Bsp Lösbarkeit:

$\{1\}$ ist die Lösungsmenge des polynomiellen Gleichungssystems
$$\left| \begin{array}{rcr} x^3 - 3x & = & -2 \\ x^4 & = & 1 \\ x^6 & = & 1 \end{array} \right|.$$

Monomordnung

Ziel: geeignete Monomordnung in $\mathbb{F}[x_1, \ldots, x_n]$

- Monomordnung soll verträglich mit der Polynommultiplikation sein.
- Wir identifizieren Monome $\mathbf{x}^\alpha := x_1^{\alpha_1} \ldots x_n^{\alpha_n}$ mit ihrem Exponentenvektor $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}_0^n$.

Definition Monomordnung

Eine Monomordnung auf $\mathbb{F}[x_1, \ldots, x_n]$ ist eine Relation $>$ auf $\mathbb{N}_0^n$ mit:

1. $>$ ist eine totale Ordnung auf $\mathbb{N}_0^n$.
2. Seien $\alpha, \beta \in \mathbb{N}_0^n$ mit $\alpha > \beta$. Dann gilt für alle $\gamma \in \mathbb{N}_0^n$

   $$\alpha + \gamma > \beta + \gamma \text{ (Verträglichkeit mit Monommultiplikation).}$$
3. $>$ ist eine Wohlordnung auf $\mathbb{N}_0^n$. D.h. jede nicht-leere Teilmenge von $\mathbb{N}_0^n$ enthält ein kleinstes Element.

Bsp:

- Die Ordnung $\ldots > 2 > 1 > 0$ erfüllt obige Bedingungen auf $\mathbb{N}_0$.
- Damit ist die Gradordnung eine Monomordnung auf $\mathbb{F}[x]$.

Wohlordnung

Anmerkung:
Wohlordnung wird uns Terminierung von Algorithmen liefern.

Lemma zur Wohlordnung

Eine Relation $>$ ist eine Wohlordnung gdw jede strikt fallende Sequenz $\alpha_1 > \alpha_2 > \ldots$ in $\mathbb{N}_0^n$ terminiert.

Beweis:

- keine Wohlordung $\Rightarrow$ Sequenz terminiert nicht:
- Sei $S \subseteq \mathbb{N}_0^n$ eine Menge ohne minimales Element.
- Wähle $\alpha_1 \in S$. Da α_1 nicht minimal in S ist, existiert $\alpha_2 < \alpha_1$, usw.
- Sequenz terminiert nicht $\Rightarrow$ keine Wohlordung:
- Sei $\alpha_1 > \alpha_2 > \ldots$ eine Sequenz. Definiere $S = \{\alpha_i \mid i \in \mathbb{N}\}$.
- S besitzt kein minimales Element, d.h. $>$ ist keine Wohlordnung.

Lexikographische Ordnung

Definition Lexikographische Ordnung $>_{lex}$

Seien $\alpha, \beta \in \mathbb{N}_0^n$. Definiere $\alpha >_{lex} \beta$, falls in $\alpha - \beta$ der von links erste Nicht-Null Eintrag positiv ist. Wir schreiben $x^\alpha >_{lex} x^\beta$ für $\alpha >_{lex} \beta$.

Bsp:

- $(2, 3, 4) >_{lex} (1, 5, 6)$ und $(2, 3, 4) >_{lex} (2, 1, 5)$.
- $(1, 0, \ldots, 0) >_{lex} (0, 1, 0 \ldots, 0) >_{lex} \ldots >_{lex} (0, \ldots, 0, 1)$, so dass
$$x_1 >_{lex} \ldots >_{lex} x_n.$$
- Wir verwenden ebenfalls $x >_{lex} y >_{lex} z$. Damit gilt z.B. $x > y^3z^5$.
- Für die alphabetische Ordnung $a > b > \ldots > z$, erhalten wir eine Wörterbuchsortierung mit z.B. Kryptanalyse > Kryptographie.

Satz

Die lexikographische Ordnung $>_{lex}$ ist eine Monomordnung.

Beweis: Übungsaufgabe.

Andere wichtige Monomordnungen

Definition Grad-Lexikographische Ordnung $>_{grlex}$

Seien $\alpha, \beta \in \mathbb{N}_0^n$ und $|\alpha| = \sum_i \alpha_i, |\beta| = \sum_i \beta_i$. Definiere $\alpha >_{grlex} \beta$ falls

$$|\alpha| > |\beta| \quad \text{oder} \quad |\alpha| = |\beta| \text{ und } \alpha >_{lex} \beta.$$

- **Bsp:** $(1,2,3) >_{grlex} (2,2,1)$ und $(1,3,2) >_{grlex} (1,2,3)$.
- Wie bei der lexikographischen Ordnung gilt $x_1 >_{grlex} \ldots >_{grlex} x_n$.

Definition Gradreverse-Lexikographische Ordnung $>_{grevlex}$

Seien $\alpha, \beta \in \mathbb{N}_0^n$. Wir definieren $\alpha >_{grevlex} \beta$ falls

$|\alpha| > |\beta|$ oder $|\alpha| = |\beta|$ und der von rechts erste Nicht-Null Eintrag in $\alpha - \beta$ ist negativ.

- **Bsp:** $(1,2,4) >_{grevlex} (3,2,1)$ und $(1,2,3) >_{grevlex} (0,3,3)$.
- Man beachte, dass z.B. $xy^2z^3 >_{lex} y^3z^3$ und $xy^2z^3 >_{grevlex} y^3z^3$.
- Es gilt $x_1 >_{grevlex} \ldots >_{grevlex} x_n$.

Multigrad

Definition Multigrad, führender Term

Sei $f = \sum_\alpha a_\alpha x^\alpha \in \mathbb{F}[x_1, \ldots, x_n]$ und sei $>$ eine Monomordnung.

1. Der *Multigrad* von f ist $\mathrm{multigrad}(f) = \max\{\alpha \in \mathbb{N}_0^n \mid a_\alpha \neq 0\}$.
2. Der *führende Koeffizient* von f ist $LC(f) = a_{\mathrm{multigrad}(f)}$.
3. Das *führende Monom* von f ist $LM(f) = x^{\mathrm{multigrad}(f)}$.
4. Der *führende Term* von f ist $LT(f) = LC(f) \cdot LM(f)$.

Bsp: Sei $f = x^2yz^3 + 2x^3 + 3y^2z$. Dann gilt für $>_{lex}$

$$\mathrm{multigrad}(f) = (3,0,0),\ LC(f) = 2,\ LM(f) = x^3 \text{ und } LT(f) = 2x^3.$$

Satz Eigenschaften des Multigrads

Seien $f, g \in \mathbb{F}[x_1, \ldots, x_n] \setminus \{0\}$. Dann gilt:

1. $\mathrm{multigrad}(fg) = \mathrm{multigrad}(f) + \mathrm{multigrad}(g)$.
2. $\mathrm{multigrad}(f+g) \leq \max\{\mathrm{multigrad}(f), \mathrm{multigrad}(g)\}$ für $f + g \neq 0$.

Beweis: Übungsaufgabe.

High-Level Beschreibung für Division in $\mathbb{F}[x_1, \ldots, x_n]$

Ziel: Algorithmus für Polynomdivision in $\mathbb{F}[x_1, \ldots, x_n]$.

Gegeben: $f, f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$

Gesucht: Darstellung $f = a_1 f_1 + \ldots + a_m f_m + r$ mit $a_1, \ldots, a_m, r \in \mathbb{F}[x_1, \ldots, x_n]$ und keiner der Terme in r ist teilbar von einem der Terme $LT(f_1), \ldots, LT(f_m)$.

Algorithmus High-Level Beschreibung Polynomdivision

EINGABE: $f, f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$

1. Teile sukzessive durch die Polynome $f_1, \ldots, f_m$ mit Rest r.
2. Falls $r \neq 0$ und r nicht weiter teilbar, entferne $LM(r)$ und iteriere.

AUSGABE: $f = a_1 f_1 + \ldots + a_m f_m + r$

Bsp: Wir verwenden lexikographische Ordnung.

- Sei $f = x^2y + xy^2 + y^2$, $f_1 = xy - 1$, $f_2 = y - 1$.
- $f : f_1 = x + y$ mit Rest $r = x + y^2 + y$. Wir entfernen x aus r.
- $(y^2 - y) : f_2 = y + 2$ mit Rest $r = 2$. Wir entfernen 2 aus r.
- Wir erhalten insgesamt $f = (x + y) \cdot f_1 + 1 \cdot f_2 + x + 2$.

Divisionsalgorithmus für $\mathbb{F}[x_1, \ldots, x_n]$

Algorithmus Division

EINGABE: $f, f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$

1. Setze $p := f, r := 0$ und $a_1 := 0, \ldots, a_m := 0$.
2. WHILE $p \neq 0$
   1. Falls $LT(f_i)$ teilt $LT(p)$, setze $a_i := a_i + \frac{\mathrm{LT}(p)}{\mathrm{LT}(f_i)}$ und $p := p - \frac{\mathrm{LT}(p)}{\mathrm{LT}(f_i)} \cdot f_i$. (Teste Teilbarkeit von $LT(p)$ in der Reihenfolge $f_1, \ldots, f_m$.)
   2. Sonst setze $p := p - LT(p)$ und $r := r + LT(p)$.

AUSGABE: $f = a_1 f_1 + \ldots + a_m f_m + r$

Korrektheit:

- Invariante $f = a_1 f_1 + \ldots + a_m f_m + p + r$ gilt in Schritt 1.
- Schritt 2.1 erhält die Invariante, falls $LT(f_i)$ den Term $LT(p)$ teilt, da
$$a_i f_i + p = (a_i + \frac{\mathrm{LT}(p)}{\mathrm{LT}(f_i)}) f_i + p - \frac{\mathrm{LT}(p)}{\mathrm{LT}(f_i)} \cdot f_i.$$
- Schritt 2.2 erhält die Invariante: $p + r = (p - LT(p)) + (r + LT(p))$.

Divisionsalgorithmus für $\mathbb{F}[x_1, \ldots, x_n]$

Terminierung:

- z.z.: Modifikationen verringern $\mathrm{multigrad}(p)$ oder erzeugen $p = 0$.
- Schritt 2.1 eliminiert $LT(p)$ mittels $p := p - \frac{\mathrm{LT}(p)}{\mathrm{LT}(f_i)} \cdot f_i$.
- Schritt 2.2 eliminiert ebenfalls $LT(p)$ mittels $p := p - LT(p)$.
- Damit verringert sich der Multigrad in Schritt 2.1 und in Schritt 2.2.
- Monomordnung: Die Sequenz der Multigrade muss terminieren.
- D.h. wir erhalten $p = 0$ und damit $f = a_1 f_1 + \ldots + a_m f_m + r$.

Reihenfolge ist wichtig

Bsp: Wie zuvor $f = x^2y + xy^2 + y^2$, $f_1 = xy - 1$ und $f_2 = y - 1$.

- Wir vertauschen aber nun die Reihenfolge in f_2, f_1 bei der Division.
- Wir erhalten $f : f_2 = x^2 + xy + x + y + 1$ mit Rest $p = 1$.
- Dies liefert die Darstellung
$$f = (x^2 + xy + x + y + 1) \cdot f_2 + 0 \cdot f_1 + 1.$$
- Bei Reihenfolge (f_1, f_2) erhielten wir dagegen die Darstellung
$$f = (x + y) \cdot f_1 + (y + 2) \cdot f_2 + (x + 2).$$
- D.h. der Rest r hängt von der Reihenfolge der Division ab.

Idealzugehörigkeit

Idealzugehörigkeit:

$f \in \langle f_1, \ldots, f_m \rangle$ falls $f = a_1 f_1 + \ldots + a_m f_m$. D.h. falls $r = 0$.

Bsp: Wir betrachten $f = xy^2 - x$, $f_1 = xy + 1$ und $f_2 = y^2 - 1$.

- Mit lexikographischer Ordnung und Reihenfolge (f_1, f_2) erhalten wir
$$f = y \cdot f_1 + 0 \cdot f_2 - x - y.$$
- Reihenfolge (f_2, f_1) liefert aber
$$f = x \cdot f_2 + 0 \cdot f_1.$$
- D.h. f ist im Ideal $\langle f_1, f_2 \rangle$.
- Allerdings liefert nur (f_2, f_1) die hinreichende Bedingung $r = 0$.

Ziel:

- Definiere geeignete Generatormenge G für $I = \langle f_1, \ldots, f_m \rangle$.
- Beim Teilen durch G soll der Rest r eindeutig bestimmt sein.
- Rest $r = 0$ soll äquivalent zur Zugehörigkeit im Ideal I sein.
- Sogenannte Gröbnerbasen sind geeignete Generatormengen.

Monomideal

Definition Monomideal

Ein Ideal $I \subseteq \mathbb{F}[x_1, \ldots, x_n]$ heißt *Monomideal* falls eine (unendliche) Menge $A \subseteq \mathbb{N}_0^n$ existiert, so dass I aus Polynomen der Form $\sum_{\alpha \in A} h_\alpha x^\alpha$ besteht. Wir schreiben dann $I = \langle x^\alpha \mid \alpha \in A \rangle$.

Bsp: Für $A = \{(1,4), (2,2), (3,1)\}$ erhalten wir $I = \langle xy^4, x^2y^2, x^3y \rangle$.

Satz Teilbarkeitssatz

Sei $I = \langle x^\alpha \mid \alpha \in A \rangle$ ein Monomideal. Ein Monom x^β liegt in I gdw x^α teilt x^β für ein $\alpha \in A$.

Beweis:

- $\Leftarrow$: Falls $x^\beta = x^\gamma \cdot x^\alpha$, dann folgt $x^\beta \in I$.
- $\Rightarrow$: Sei $x^\beta \in I$, d.h. $x^\beta = \sum_i h_i x^{\alpha^{(i)}}$ mit $h_i \in \mathbb{F}[x_1, \ldots, x_n]$, $\alpha^{(i)} \in A$.
- Multipliziere $h_i x^{\alpha^{(i)}}$ aus. Jedes Monom ist teilbar durch ein $x^{\alpha^{(i)}}$.
- Die Summe kollabiert aber zu einem einzigen Monom x^β.
- Damit muss auch das Monom x^β durch ein $x^{\alpha^{(i)}}$ teilbar sein.

Gleichheit von Monomidealen

Satz Darstellung aus Monomen

Sei I ein Monomideal und $f \in \mathbb{F}[x_1, \ldots, x_n]$. Dann gilt $f \in I$ gdw f eine $\mathbb{F}$-Linearkombination von Monomen in I ist.

Beweis:

- $\Rightarrow$: Sei $f = \sum_i h_i x^{\alpha^{(i)}}$.
- Ausmultiplizieren von $h_i x^{\alpha^{(i)}}$ liefert Monome der Form cx^γ mit $c \in \mathbb{F}$ und $x^{\alpha^{(i)}} \mid x^\gamma$. Nach Teilbarkeitssatz ist x^γ ein Monom in I.
- Damit können wir f in der gewünschten Form schreiben
$$f = \sum_i c_i x^{\gamma^{(i)}} \text{ mit } c_i \in \mathbb{F},\, x^{\gamma^{(i)}} \in I.$$
- $\Leftarrow$: Folgt aus der Abgeschlossenheit von I gegenüber Addition.

Korollar Gleichheit von Monomidealen

Zwei Monomideale sind gleich gdw sie dieselben Monome enthalten.

Dicksons Lemma

Lemma Dicksons Lemma

Jedes Monomideal $I = \langle x^\alpha \mid \alpha \in A\rangle \subset \mathbb{F}[x_1, \ldots, x_n]$ besitzt eine endliche Basis $I = \langle x^{\alpha^{(1)}}, \ldots, x^{\alpha^{(m)}}\rangle$.

Beweis per Induktion über die Anzahl der Variablen n:

- $n = 1$: $I = \langle x_1^\alpha \mid \alpha \in A\rangle$. Sei β das kleinste Element in $A \subseteq \mathbb{N}_0$.
- Daher gilt $x_1^\beta \mid x_1^\alpha$ für alle $\alpha \in A$. D.h. $I = \langle x_1^\beta\rangle$.
- $n - 1 \to n$: Wir verwenden die Variablen $x_1, \ldots, x_{n-1}, y$.
- D.h. Monome besitzen die Form $x^\alpha y^t$ mit $\alpha \in \mathbb{N}_0^{n-1}$ und $t \in \mathbb{N}_0$.
- Sei J die Projektion von I auf $\mathbb{F}[x_1, \ldots, x_{n-1}]$. D.h. J wird generiert von denjenigen Monomen x^α, für welche $x^\alpha y^t \in I$ für ein $t \geq 0$.
- IV: Wir schreiben $J = \langle x^{\alpha^{(1)}}, \ldots, x^{\alpha^{(m)}}\rangle$. Für $i = 1, \ldots, m$ gilt
$$x^{\alpha^{(i)}} y^{t_i} \in I \text{ für ein festes } t_i \geq 0. \text{ Sei } t = \max_i\{t_i\}.$$
- Für jedes feste $k = 0, \ldots, t-1$ definiere $J_k \subseteq \mathbb{F}[x_1, \ldots, x_{n-1}]$ als die Projektion derjenigen Monome in I, die genau y^k enthalten.

Dicksons Lemma

Beweis: (Fortsetzung)

- Nach IV: $J_k = \langle x^{\alpha_k^{(1)}}, \ldots, x^{\alpha_k^{(m_k)}} \rangle$ für $k = 0, \ldots, t-1$.
- Wir behaupten, dass I von folgender Monomliste L generiert wird.

$$
\begin{array}{rlcl}
\text{aus } J: & x^{\alpha^{(1)}} y^t, & \ldots & , x^{\alpha^{(m)}} y^t \\
\text{aus } J_0: & x^{\alpha_0^{(1)}} y^0, & \ldots & , x^{\alpha_0^{(m_0)}} y^0 \\
\vdots & & \vdots & \\
\text{aus } J_{t-1}: & x^{\alpha_{t-1}^{(1)}} y^{t-1}, & \ldots & , x^{\alpha_{m-1}^{(m_{t-1})}} y^{t-1}
\end{array}
$$

- $\langle L \rangle \subseteq I$: Die Monome in unserer Liste L sind alle in I. Dies folgt für die Elemente $x^{\alpha_k^{(i)}} y^k$ nach Konstruktion der Elemente in J_k.
- Für die Elemente $x^{\alpha^{(i)}} y^t$ gilt dies aufgrund der Maximalität von t.
- $I \subseteq \langle L \rangle$: Jedes $x^\alpha y^p \in I$ wird von einem Listenmonom geteilt.
- Sei $p \geq t$. Dann teilt ein $x^{\alpha^{(i)}} y^t$ nach Konstruktion von J.
- Sei $p < t$. Dann teilt ein $x^{\alpha_p^{(i)}} y^p$ nach Konstruktion von J_p.
- D.h. $\langle L \rangle$ und I enthalten dieselben Monome und sind daher gleich.

Idealzugehörigkeit in Monomidealen

Lemma Dicksons Lemma (Teil II)

Jedes Monomideal $I = \langle x^\alpha \mid \alpha \in A \rangle \subset \mathbb{F}[x_1, \ldots, x_n]$ besitzt eine endliche Basis $I = \langle x^{\alpha^{(1)}}, \ldots, x^{\alpha^{(m)}} \rangle$ mit $a^{(i)} \in A$.

Beweis: Übungsaufgabe.

Satz Idealzugehörigkeit in Monomidealen

Sei $I = \langle x^{\alpha^{(1)}}, \ldots, x^{\alpha^{(m)}} \rangle$ ein Monomideal. Dann gilt $f \in I$ gdw f bei Division durch $x^{\alpha^{(1)}}, \ldots, x^{\alpha^{(m)}}$ Rest 0 lässt.

Beweis:

- $\Leftarrow$: Aus $f = h_1 \cdot x^{\alpha^{(1)}} + \ldots + h_m \cdot x^{\alpha^{(m)}} + 0$ folgt $f \in I$.
- $\Rightarrow$: Nach *Satz zur Darstellung aus Monomen* folgt, dass $f \in I$ gwd
$$f = \sum_i c_i x^{\gamma^{(i)}} \text{ mit } x^{\gamma^{(i)}} \in I.$$
- Andererseits ist $x^{\gamma^{(i)}} \in I$ gwd $x^{\alpha^{(j)}}$ teilt $x^{\gamma^{(i)}}$ für ein $j \in [m]$.
- Damit wird jeder Term in f von einem der $x^{\alpha^{(j)}}$ geteilt.
- Sukzessives Teilen von f durch $x^{\alpha^{(1)}}, \ldots, x^{\alpha^{(m)}}$ liefert also Rest 0.

Das Ideal der führenden Terme

Definition Ideal der führenden Terme

Sei $I \subseteq \mathbb{F}[x_1, \ldots, x_n] \setminus \{0\}$ ein Ideal, $LT(I)$ die Menge führender Terme

$$LT(I) = \{cx^\alpha \mid \text{es existiert } f \in I \text{ mit } LT(f) = cx^\alpha\}.$$

Dann heißt $\langle LT(I) \rangle$ das *Ideal der führenden Monome von* I.

Anmerkung:

- Sei $I = \langle f_1, \ldots, f_m \rangle$. Es gilt $LT(f_i) \in LT(I) \subseteq LT(I)$ für alle $i \in [m]$.
- Daher folgt $\langle LT(f_1), \ldots, LT(f_n) \rangle \subseteq \langle LT(I) \rangle$.
- Andererseits kann $LT(I)$ weitere Element enthalten.
- Sei $I = \langle f_1, f_2 \rangle$ mit $f_1 = x^3 - 2xy$ und $f_2 = x^2y + x - 2y^2$.
- Es gilt $x^2 \in I$ wegen $x^2 = -y \cdot f_1 - x \cdot f_2$. D.h. $x^2 \in \langle LT(I) \rangle$.
- Aber x^2 wird weder von $LT(f_1) = x^3$ noch von $LT(f_2) = x^2y$ geteilt.
- Daraus folgt, dass x^2 nicht im Monomideal $\langle LT(f_1), LT(f_2) \rangle$ ist.

Existenz einer Gröbnerbasis

Definition Gröbnerbasis

Eine Menge $G = \{g_1, \ldots, g_m\} \subseteq I$ heißt *Gröbnerbasis* falls

$$\langle LT(I)\rangle = \langle LT(g_1), \ldots, LT(g_n)\rangle.$$

Satz Existenz einer Gröbnerbasis

Sei I ein Ideal. Dann ist $\langle LT(I)\rangle$ ein Monomideal und es existiert eine Gröbnerbasis $\{g_1, \ldots, g_m\} \subseteq I$ mit $\langle LT(I)\rangle = \langle LT(g_1), \ldots, LT(g_m)\rangle$.

Beweis:

- Es gilt $\langle\{LT(g) \mid g \in I \setminus \{0\}\}\rangle = \langle\{LM(g) \mid g \in \setminus\{0\}\}\rangle$.
- Die führenden Monome von I generieren aber ein Monomideal.
- Dickson's Lemma: $\langle LT(I)\rangle$ wird endlich generiert, d.h.
  $$\langle LT(I)\rangle = \langle LM(g_1), \ldots, LM(g_m)\rangle \text{ für } g_1, \ldots, g_m \in I.$$
- Aus obiger Gleichung folgt $\langle LT(I)\rangle = \langle LT(g_1), \ldots, LT(g_m)\rangle$.

Hilbert Basissatz

Satz Hilbert Basissatz

Jedes Ideal $I \subseteq \mathbb{F}[x_1, \ldots, x_n]$ wird endlich generiert, d.h.

$$I = \langle g_1, \ldots, g_m \rangle \text{ für } g_1, \ldots, g_m \in I.$$

Beweis:

- Falls $I = \{0\}$, verwende 0 als Generator. Sei also $I \neq \{0\}$.
- Wir wissen, dass $\langle LT(I) \rangle = \langle LT(g_1), \ldots, LT(g_m) \rangle$ für $g_i \in I$.
- Behauptung: $I = \langle g_1, \ldots, g_m \rangle$. Es gilt $\langle g_1, \ldots, g_m \rangle \subseteq I$, da $g_i \in I$.
- $I \subseteq \langle g_1, \ldots, g_m \rangle$: Sei $f \in I$ beliebig.
- Teilen von f durch $g_1, \ldots, g_m$ liefert $f = a_1 g_1 + \ldots + a_m g_m + r$.
- Kein Term von r wird von einem der $LT(g_1), \ldots, LT(g_m)$ geteilt.
- Angenommen $r \neq 0$. Es gilt $r = f - a_1 g_1 - \ldots - a_m g_m \in I$.
- Aus $r \in I$ folgt $LT(r) \in \langle LT(I) \rangle = \langle LT(g_1), \ldots, LT(g_m) \rangle$.
- Dann muss aber nach Teilbarkeitskeitssatz $LT(r)$ von einem der Terme $LT(g_i)$ geteilt werden. (Widerspruch)
- D.h. es folgt $r = 0$ und damit $f \in \langle g_1, \ldots, g_m \rangle$.

Charakterisierung von Gröbnerbasen

Satz Charakterisierung von Gröbnerbasen

Eine Menge $G = \{g_1, \ldots, g_m\} \subseteq I$ ist eine Gröbnerbasis gdw für jedes $f \in I$ der Term $LT(f)$ von einem der $LT(g_i)$, $i = 1, \ldots, m$ geteilt wird.

Beweis:

- $\Rightarrow$: Sei $G = \{g_1, \ldots, g_m\}$ eine Gröbnerbasis, d.h.
$$\langle LT(I)\rangle = \langle LT(g_1), \ldots, LT(g_m)\rangle.$$
- Für jedes $f \in I$ gilt $LT(f) \in \langle LT(I)\rangle = \langle LT(g_1), \ldots, LT(g_m)\rangle$.
- Nach Teilbarkeitssatz ist $LT(f) \in \langle LT(g_1), \ldots, LT(g_n)\rangle$ gdw $LT(f)$ von einem der Terme $LT(g_i)$ geteilt wird.
- $\Leftarrow$: Sei $f \in I$ beliebig. Es gilt $LT(g_i) \mid LT(f)$ für ein $i \in [m]$.
- Daraus folgt $\langle LT(f)\rangle \subseteq \langle LT(g_1), \ldots, LT(g_m)\rangle$.
- Da stets auch $\langle LT(g_1), \ldots, LT(g_m)\rangle \subseteq LT(I)$ gilt, folgt
$$\langle LT(I)\rangle = \langle LT(g_1), \ldots, LT(g_m)\rangle.$$

Beispiel einer Gröbnerbasis

Bsp: Gröbnerbasis. Wir verwenden lex-Ordnung in $\mathbb{R}[x, y, z]$.

- Sei $I = \langle g_1, g_2 \rangle = \langle x + z, y - z \rangle$. Zeigen: $\{g_1, g_2\}$ ist Gröbnerbasis.
- D.h. wir müssen zeigen, dass $\langle LT(g_1), LT(g_2) \rangle = \langle x, y \rangle = \langle LT(I) \rangle$.
- Es gilt offenbar $\langle x, y \rangle \subseteq \langle LT(I) \rangle$, bleibt $\langle LT(I) \rangle \subseteq \langle x, y \rangle$ zu zeigen.
- Sei $f \in I$. Wir müssen zeigen, dass $LT(f)$ von x oder y geteilt wird.
- Annahme: $f \in \mathbb{R}[z] \setminus \{0\}$.
- Wegen $f \in I$ verschwindet f auf $\mathbf{V}(x + z, y - z)$.
- D.h. f verschwindet auf allen Punkten $(-t, t, t) \in \mathbb{R}^3$. Das einzige Polynom $f \in \mathbb{R}[z]$ mit dieser Eigenschaft ist $z = 0$ (Widerspruch).
- D.h. jedes Polynom $f \in I$ enthält einen x oder einen y-Term.

ACC – Ascending Chain Condition

Satz Ascending Chain Condition (ACC)

Sei $I_1 \subseteq I_2 \subseteq \ldots$ eine aufsteigende Kette von Idealen in $\mathbb{F}[x_1, \ldots, x_n]$.
Dann existiert ein $N \geq 1$ mit $I_N = I_M$ für alle $M \geq N$.

Beweis:

- Wir definieren $I = \bigcup_{i=1}^{\infty} I_i$. Wir zeigen, dass I ein Ideal ist.
- Seien $f, g \in I$. Sei $f \in I_i$ und $g \in I_j$. ObdA $i \leq j$.
- Dann gilt $f, g \in I_j$ und damit $f + g \in I_j \subseteq I$.
- Analog folgt für $f \in I$, dass $f \in I_i$ für ein i und damit $hf \in I_i \subseteq I$.
- Da I ein Ideal ist, wird es endlich erzeugt. D.h. $I = \langle g_1, \ldots, g_m \rangle$.
- Jeder Generator $g_i \in I$ ist in einem Ideal I_{j_i}. Sei $N = \max_i \{j_i\}$.
- Dann sind $g_1, \ldots, g_m \in I_N$. Damit gilt

$$I = \langle g_1, \ldots, g_m \rangle \subseteq I_N \subseteq I_{N+1} \subseteq \ldots \subseteq I.$$

Ideale definieren Varietäten

Definition Varietät eines Ideals $\mathbf{V}(I)$

Sei $I \subseteq \mathbb{F}[x_1, \ldots, x_n]$ ein Ideal. Wir definieren

$$\mathbf{V}(I) = \{(a_1, \ldots, a_n) \in \mathbb{F}^n \mid f(a_1, \ldots, a_n) = 0 \text{ für alle } f \in I\}.$$

Satz Varietät eines Ideals $\mathbf{V}(I)$

$\mathbf{V}(I)$ ist eine Varietät. Insbesondere gilt für $I = \langle f_1, \ldots, f_m \rangle$, dass

$$\mathbf{V}(I) = \mathbf{V}(f_1, \ldots, f_m).$$

Beweis:

- $\mathbf{V}(I) \subseteq \mathbf{V}(f_1, \ldots, f_m)$: Sei $(a_1, \ldots, a_n) \in \mathbf{V}(I)$. Dann gilt $f(a_1, \ldots, a_m) = 0$ für alle $f \in I$, d.h. insbesondere für $f_1, \ldots, f_m \in I$.
- $\mathbf{V}(f_1, \ldots, f_m) \subseteq \mathbf{V}(I)$: Sei $(a_1, \ldots, a_n) \in \mathbf{V}(f_1, \ldots, f_m)$ und $f \in I$.
- Wir schreiben $f = \sum_i h_i f_i$ und damit gilt
$$f(a_1, \ldots, a_m) = \sum_{i=1}^{m} h_i(a_1, \ldots, a_m) \cdot \underbrace{f_i(a_1, \ldots, a_m)}_{0} = 0.$$

Eindeutigkeit des Rests für Gröbnerbasen

Satz Eindeutigkeit des Rests

Sei $G = \{g_1, \ldots, g_m\}$ eine Gröberbasis für $I \subseteq \mathbb{F}[x_1, \ldots, x_n]$ und $f \in \mathbb{F}[x_1, \ldots, x_n]$. Es existiert ein eindeutiger Rest r mit

1. Kein Term von r is teilbar von einem der $LT(g_1), \ldots, LT(g_m)$.
2. Es existiert ein $g \in I$ mit $f = g + r$.

Beweis:

- **Existenz:** Polynomdivision mit $g_1, \ldots, g_m$ liefert
  $f = \underbrace{a_1 g_1 + \ldots + a_m g_m}_{g} + r$, wobei r Eigenschaft 1 besitzt.
- **Eindeutigkeit:** Seien $r \neq r'$ Reste mit $f = g + r = g' + r'$.
- Es gilt $r - r' = g' - g \in I$, d.h.
  $$LT(r - r') \in \langle LT(I) \rangle = \langle LT(g_1), \ldots, LT(g_m) \rangle.$$
- Damit ist $LT(r - r')$ teilbar von einem $LT(g_i)$. D.h. einer der Terme von r oder r' wird von einem $LT(g_i)$ geteilt. (Widerspruch)

Man beachte: r ist eindeutig unabhängig von der Reihenfolge der g_i.

Idealzugehörigkeit mittels Gröbnerbasis

Satz Idealzugehörigkeit mittels Gröbnerbasis

Sei $G = \{g_1, \ldots, g_m\}$ eine Gröbnerbasis für I. Es gilt $f \in I$ gdw f bei Division durch die Polynome in G Rest 0 lässt.

Beweis:

- $\Leftarrow$: Sei $f = a_1 g_1 + \ldots + a_m g_m$. Dann gilt $f \in \langle g_1, \ldots, g_m \rangle = I$.
- $\Rightarrow$: Sei $f \in I$. Dann erfüllt die Wahl $g = f$ und $r = 0$ beide Eigenschaften des Satzes zuvor.
- Da der Rest r eindeutig bestimmt ist, muss $r = 0$ gelten.

Ziel: Konstruktion Gröbnerbasis

- Konstruiere für $f_1, \ldots, f_m$ eine Gröbnerbasis $g_1, \ldots, g_t$ mit
$$\langle f_1, \ldots, f_m \rangle = \langle g_1, \ldots, g_t \rangle.$$
- Erzeuge dazu eine Linearkombinationen g der f_i, deren führender Term *nicht* im durch die $LT(f_i)$ erzeugten Ideal ist.
- Wir eliminieren dazu die führenden Koeffizienten der f_i.
- Füge g zu $f_1, \ldots, f_m$ hinzu und iteriere.

Syzygien-Polynom

Definition kgV, S-Polynom (Syzygien-Polynom)

Seien $f, g \in \mathbb{F}[x_1, \ldots, x_n]$ mit Multigraden $\alpha, \beta \in \mathbb{N}_0^n$.

1. Das *kleinste gemeinsame Vielfache* von $LM(f)$ und $LM(g)$ ist definiert als x^γ, wobei $\gamma = (\gamma_1, \ldots, \gamma_n)$ mit $\gamma_i = \max_i\{\alpha_i, \beta_i\}$.
2. Das S-Polynom von f und g ist definiert als
$$S(f,g) = \frac{x^\gamma}{LT(f)} \cdot f - \frac{x^\gamma}{LT(g)} \cdot g.$$

Bsp:

- Seien $f = x^3y^2 + x^4, g = 3x^4y + y^2 \in \mathbb{R}[x,y]$ in grlex-Ordnung.
- Es gilt $\alpha = (3,2), \beta = (4,1)$ und $\gamma = (4,2)$. Damit ist
$$S(f,g) = \frac{x^4y^2}{x^3y^2} \cdot f - \frac{x^4y^2}{3x^4y} \cdot g = xf - \frac{1}{3}yg = x^5 - \frac{1}{3}y^3.$$

Buchberger Kriterium

Satz Buchberger Kriterium

Sei I ein Ideal. Eine Basis $G = \{g_1, \ldots, g_m\}$ ist eine Gröbnerbasis gdw für alle $i \neq j$ beim Teilen von $S(g_i, g_j)$ durch G der Rest 0 ist.

Beweisskizze:

- $\Rightarrow$: Sei G eine Gröbnerbasis.
- Da $S(g_i, g_j) \in I$ liefert die Teilung durch G Rest 0.
- $\Leftarrow$: Sei $f \in I$ beliebig. Wir müssen zeigen, dass
$$LT(f) \in \langle LT(g_1), \ldots, LT(g_m) \rangle.$$
- Da $f \in I = \langle g_1, \ldots, g_m \rangle$ gilt $f = \sum_i h_i g_i$. Daraus folgt
$$\mathrm{multigrad}(f) \leq \max_i \{\mathrm{multigrad}(h_i g_i)\}.$$
- Müssen zeigen: $\mathrm{multigrad}(f) = \max_i \{\mathrm{multigrad}(h_i g_i)\}$ für ein i.
- Damit $LT(g_i) \mid LT(f)$, woraus $LT(f) \in \langle LT(g_1), \ldots, LT(g_m) \rangle$ folgt.
- Annahme: $\mathrm{multigrad}(f) < \max_i \{\mathrm{multigrad}(h_i g_i)\}$. D.h. es werden Terme eliminiert. Dies kann nur durch S-Polynome geschehen.
- Aufgrund der Teilbarkeit der S-Polynome gilt $S(g_i, g_j) = \sum_k h'_k g_k$.
- D.h. wir können alle Eliminationen entfernen. (Widerspruch)

Beispiel Gröbnerbasis

Bsp:

- Wir verifizieren erneut die Basis $f_1 = x + z$, $f_2 = y - z$ in $\mathbb{R}[x, y, z]$.
- Es gilt $S(f_1, f_2) = y \cdot f_1 - x \cdot f_2 = yz + xz$.
- Division mit f_1, f_2 liefert $S(f_1, f_2) = z \cdot f_1 + z \cdot f_2$.
- Damit ist $\{f_1, f_2\}$ wirklich eine Gröbnerbasis für $\langle f_1, f_2 \rangle$.

Buchberger Algorithmus

Algorithmus BUCHBERGER

EINGABE: $F = \{f_1, \ldots, f_m\}$ mit $I = \langle f_1, \ldots, f_m \rangle$

1. Setze $G := F$.
2. WHILE ($\exists g_i \neq g_j \in G$, so dass $S(g_i, g_j) : G$ Rest $r \neq 0$ lässt)
   1. $G := G \cup \{r\}$.

AUSGABE: Gröbnerbasis G für I mit $F \subseteq G$

Beispiel Gröbnerbasen-Berechnung

Bsp:

- Seien $f_1 = x^2y + xy$, $f_2 = xy^2 + 1 \in \mathbb{R}[x, y]$ in grlex-Ordnung.
- $S(f_1, f_2) = yf_1 - xf_2 = xy^2 - x$. Division liefert
$$S(f_1, f_2) = 1 \cdot f_2 - x - 1.$$
- Wir fügen $f_3 = -x - 1$ zur Basis hinzu.
- $S(f_1, f_3) = f_1 + xyf_3 = 0$ und $S(f_2, f_3) = f_2 + y^2 f_3 = -y^2 + 1$.
- Wir fügen $f_4 = -y^2 + 1$ zur Basis hinzu.
- $S(f_1, f_4)$, $S(f_2, f_4)$, $S(f_3, f_4)$ verschwinden bei Basisdivision.
- D.h. $\{x^2y + xy, xy^2 + 1, -x - 1, -y^2 + 1\}$ ist Gröbnerbasis für I.

Notation für Ideale und Division

Sei $G = \{g_1, \ldots, g_m\}$ und $f \in \mathbb{F}[x_1, \ldots, x_n]$. Wir schreiben vereinfacht
$$\langle G \rangle = \langle g_1, \ldots, g_m \rangle \text{ und } \langle LT(G) \rangle = \langle LT(g_1), \ldots, LT(g_m) \rangle.$$
Wir notieren mit $\overline{f}^G$ den Rest der Division von f durch G.

Korrektheit von BUCHBERGER

Satz

Algorithmus BUCHBERGER terminiert nach endlich vielen Schritten mit einer Gröbnerbasis.

Beweis:

- **Korrektheit:** Als Invariante gilt, dass G das Ideal I generiert.
- Sei $S(g_i, g_j) = \sum_i a_i g_i + r$. Da $S(g_i, g_j), \sum_i a_i g_i \in I$ ist auch $r \in I$.
- Wir fügen also nur Element aus I zu G hinzu.
- Buchberger Kriterium: G ist bei Terminierung eine Gröbnerbasis.
- **Terminierung:** Sei $G = \{g_1, \ldots, g_m\}$.
- Sei $G' = G \cup \{r\}$ in Schritt 2.1. Da r in G aufgenommen wird, wird $LT(r)$ von keinem der $LT(g_i)$ geteilt. D.h.
$$\langle LT(G) \rangle \subset \langle LT(G') \rangle, \text{ da } G \subset G' \text{ und } LT(r) \in \langle LT(G') \rangle.$$
- Damit entsteht in Schritt 2.1 eine aufsteigende Kette von Idealen
$$\langle LT(G) \rangle \subset \langle LT(G') \rangle \subset \langle LT(G'') \rangle \subset \ldots.$$
- Nach ACC stabilisiert die Kette nach endlichen vielen Schritten.

Minimale Gröbnerbasis

Beobachtung: Gröbnerbasen enthalten oft unnötige Generatoren.

Satz Elimination von Generatoren

Sei G eine Gröbnerbasis für I. Sei $g \in G$ mit $LT(g) \in \langle LT(G \setminus \{g\})\rangle$. Dann ist $G \setminus \{g\}$ eine Gröbnerbasis von I.

Beweis:

- Da G eine Gröbnerbasis ist, gilt $\langle LT(G)\rangle = \langle LT(I)\rangle$.
- Wegen $LT(g) \in \langle LT(G \setminus \{g\})\rangle$ folgt
$$\langle LT(G \setminus \{g\})\rangle = \langle LT(G)\rangle = \langle LT(I)\rangle.$$
- Damit ist auch $G \setminus \{g\}$ eine Gröbnerbasis.

Definition Minimale Gröbnerbasis

Wir nennen eine Gröbnerbasis G *minimal*, falls für alle $g \in G$ gilt:

1. $LT(g) \notin \langle LT(G \setminus \{g\})\rangle$.
2. $LC(g) = 1$.

Minimierung einer Gröbnerbasis

Algorithmus MINIMIERE GRÖBNER

EINGABE: Gröbnerbasis B

1. Für alle $g \in G$: Falls $LT(g) \in \langle LT(G \setminus \{g\}) \rangle$, setze $G := G \setminus \{g\}$.
2. Für alle $g \in G$: Setze $g := \frac{g}{LC(g)}$.

AUSGABE: minimale Gröbnerbasis

Beispiel: Gröbnerbasis $\{x^2y + xy, xy^2 + 1, -x - 1, -y^2 + 1\}$ (grlex)

- Wir können g_1 eliminieren, da $LT(g_1) = x^2y = -xy \cdot LT(g_3)$.
- Ferner können wir g_2 eliminieren, da $LT(g_2) = xy^2 = -x \cdot LT(g_4)$.
- Damit ist $\{x + 1, y^2 - 1\}$ eine minimale Gröbnerbasis.
- Leider sind minimale Gröbnerbasen nicht eindeutig.
- Die folgenden Basen sind ebenfalls minimal für die grlex-Ordnung

$$\{x + 1, y^2 + a(x + 1) - 1\} \text{ mit } a \in \mathbb{Z}.$$

Reduzierte Gröbnerbasis

Definition reduzierte Gröbnerbasis

Wir nennen eine Gröbnerbasis G *reduziert*, falls für alle $g \in G$ gilt:

1. Kein Monom von g liegt in $LT(G \setminus \{g\})\rangle$.
2. $LC(g) = 1$.

Algorithmus REDUZIERE GRÖBNER

EINGABE: minimale Gröbnerbasis G

1. Für alle $g \in G$
   1. Sei $g' := \overline{g}^{G \setminus \{g\}}$.
   2. Setze $G := G \setminus \{g\} \cup \{g'\}$.

AUSGABE: reduzierte Gröbnerbasis G

Reduzierte Gröbnerbasis

Satz Korrektheit REDUZIERE GRÖBNER

Algorithmus REDUZIERE GRÖBNER berechnet eine reduzierte Gröbnerbasis.

Beweis:

- Wir bezeichnen ein Polynom $g \in G$ als reduziert, falls kein Monom von g in $\langle LT(G \setminus \{g\}) \rangle$ liegt (Eigenschaft 1).
- Ein reduziertes g bleibt reduziert, sofern sich die führenden Terme von G nicht ändern.
- In Schritt 1.1 gilt $LT(g') = LT(g)$, da aufgrund von G's Minimalität $LT(g)$ von keinem der führenden Terme in $LT(G \setminus \{g\})$ geteilt wird.
- D.h. führendeTerme bleiben unverändert und $\langle LT(G') \rangle = \langle LT(G) \rangle$.
- Damit ist G' in Schritt 1.2 ebenfalls eine minimale Gröbnerbasis.
- Da wir alle $g \in G$ reduzieren, ist G bei Terminierung reduziert.

Eindeutigkeit reduzierter Gröbnerbasen

Satz Existenz und Eindeutigkeit reduzierter Gröbnerbasen

Jedes Ideal $I \subseteq \mathbb{F}[x_1, \ldots, x_n]$ besitzt eine eindeutige reduzierte Gröbnerbasis.

Beweis:

- **Existenz:** Hilbert Basissatz: $I = \langle G \rangle$ mit endlicher Basis G. Das G aus dem Beweis zum Basissatz ist bereits eine Gröbnerbasis.
- Anwendung der Algorithmen MINIMIERE GRÖBNER und REDUZIERE GRÖBNER führt zu einer reduzierten Basis G.
- **Eindeutigkeit:** Seien G und G' reduzierte Gröbnerbasen von I.
- Da G, G' Gröbnerbasen sind, gilt $\langle LT(G) \rangle = \langle LT(G') \rangle = \langle LT(I) \rangle$.
- $LT(I)$ ist ein Monomideal. Zwei Monomideal sind gleich gdw sie dieselben Monome enthalten. D.h es gilt $LT(G) = LT(G')$.
- Daher existiert für jedes $g \in G$ ein $g' \in G'$ mit $LT(g) = LT(g')$.

Gleichheit von Idealen

Beweis: (Fortsetzung)

- Es genügt zu zeigen, dass $g = g'$.
- Wegen $LT(g) = LT(g')$, wird $LT(g - g')$ eliminiert.
- Da G, G' reduziert sind, wird keiner der Terme in $g - g'$ von einem der $LT(g_i)$ geteilt. D.h.
$$\overline{g - g'}^G = g - g'.$$
- Da $g, g' \in I$, gilt $g - g' \in I$.
- Da G eine Gröbnerbasis ist, folgt damit
$$\overline{g - g'}^G = 0.$$
- Dies zeigt $g = g'$ und damit sind G und G' identisch.

Algorithmus GLEICHHEIT IDEALE

EINGABE: $I_1 = \langle f_1, \ldots, f_\ell \rangle$, $I_2 = \langle g_1, \ldots, g_m \rangle$.

1. Fixiere eine beliebige Monomordnung.
2. Berechne reduzierte Gröbnerbasen G_1, G_2 für I_1, I_2.

AUSGABE: $I_1 = I_2$ gdw $G_1 = G_2$.

Algorithmische Betrachtungen

Anmerkung: Effizienz

- Ziel: Effizienzsteigerung des BUCHBERGER-Algorithmus durch Vermeidung von unnötigen S-Polynom Berechnungen.
- Verwendet Verallgemeinerung von S-Polynomen.
- Implementierungen im F4- und F5-Algorithmus.

Laufzeit von BUCHBERGER:

- Sei I ein Ideal mit Generatoren vom Multigrad α.
- Sei der Grad definiert als $d = \sum_i \alpha$.
- Gröbnerbasis von I kann Polynome vom Grad 2^{2^d} enthalten.
- D.h. BUCHBERGER besitzt doppelt exponentielle Laufzeit.
- Probleme in der Praxis können aber oft effizient gelöst werden.
- grevlex-Ordnung erzeugt meist Polynome minimalen Grads.

BUCHBERGER versus GAUSS-ELIMINATION

Bsp: $I = \langle 3w - 6x - 2y, 2w - 4x + 4z, w - 2x - y - z\rangle \subseteq \mathbb{R}[w,x,y,z]$

- Wir stellen I in Matrixform dar.
$$\begin{pmatrix} 3 & -6 & -2 & 0 \\ 2 & -4 & 0 & 4 \\ 1 & -2 & -1 & -1 \end{pmatrix}$$
- Die normierte Stufenform davon ist
$$\begin{pmatrix} 1 & -2 & -1 & -1 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$
- Liefert eine minimale Gröbnerbasis $G = \{w - 2x - y - z, y + 3z\}$.
- Wir stellen sicher, dass führende Einsen in ihrer Spalte der einzige Nicht-Null Eintrag sind.
$$\begin{pmatrix} 1 & -2 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{pmatrix}$$
- Liefert die reduzierte Gröbnerbasis $G' = \{w - 2x + 2z, y + 3z\}$.
- Die Gauß-Elimination ist ein Spezialfall von BUCHBERGER.
- G' erlaubt einfaches Lösen des Gleichungssystems.

Lösen polynomieller Gleichungssysteme

Bsp:

- Wir suchen alle Lösungen in $\mathbb{C}$ des Gleichungssystems
$$\left|\begin{array}{rcl} x^2+y^2+z^2 & = & 1 \\ x^2+z^2 & = & y \\ x & = & z \end{array}\right|.$$
- Sei $I = \langle x^2+y^2+z^2-1, x^2-y+z^2, x-z\rangle$.
- Wir wollen $\mathbf{V}(I)$ bestimmen.
- BUCHBERGER liefert die reduzierte lex-Gröbnerbasis
$$G = \{x-z, y-2z^2, z^4+\tfrac{1}{2}z^2-\tfrac{1}{4}\}.$$
- Offenbar eliminiert die lex-Ordnung x in g_2 und x, y in g_3.
- Der Generator g_3 hängt nur von z ab und liefert
$$z = \pm\tfrac{1}{2}\sqrt{\pm\sqrt{5}-1}.$$
- Rücksubstitution von z in g_1, g_2 führt zu Lösungen in x und y.
- Damit erhalten wir alle Lösungen unseres Gleichungssystems.

Eliminationsideal

Definition Eliminationsideal

Sei $I = \langle g_1, \ldots, g_m \rangle \subseteq \mathbb{F}[x_1, \ldots, x_n]$. Das *$\ell$-te Eliminationsideal I_ℓ* ist

$$I_\ell = I \cap \mathbb{F}[x_{\ell+1}, \ldots, x_n].$$

Anmerkung:

- In I_ℓ sind die Variablen $x_1, \ldots, x_\ell$ eliminiert.
- D.h. zum sukzessiven Lösen polynomieller Gleichungssysteme müssen wir Basen für I_ℓ für $\ell = 1, \ldots, n$ berechnen.

Eliminationstheorem

Satz Eliminationstheorem

Sei G eine lex-Gröbnerbasis für $I \subseteq \mathbb{F}[x_1, \ldots, x_n]$. Dann ist

$$G_\ell = G \cap \mathbb{F}[x_{l+1}, \ldots, x_n] \text{ für } \ell = 0, \ldots, n$$

eine Gröbnerbasis des ℓ-ten Eliminationsideals I_ℓ.

Beweis:

- $\langle LT(G_\ell) \rangle \subseteq \langle LT(I_\ell) \rangle$: Nach Konstruktion gilt $G_\ell \subseteq I_\ell$. Daraus folgt
$$\langle LT(G_\ell) \rangle \subseteq \langle LT(I_\ell) \rangle.$$
- $\langle LT(I_\ell) \rangle \subseteq \langle LT(G_\ell) \rangle$: Sei $f \in I_\ell \subseteq \mathbb{F}[x_{\ell+1}, \ldots, x_\ell]$.
- zu zeigen: $LT(f)$ wird von einem der $LT(g)$ mit $g \in G_\ell$ geteilt.
- Da $f \in I$, wird $LT(f)$ von einem der $LT(g)$ mit $g \in G$ geteilt.
- Damit ist $LT(g) \in \mathbb{F}[x_{\ell+1}, \ldots, x_n]$. Da aber $x_1 > \ldots > x_{\ell+1}$, folgt
$$g \in \mathbb{F}[x_{\ell+1}, \ldots, x_n].$$
- D.h. insgesamt gilt $g \in G \cap \mathbb{F}[x_{\ell+1}, \ldots, x_n] = G_\ell$.

Erweitern partieller Lösungen

Bsp: Sei $I = \langle xy - 1, xz - 1 \rangle \subseteq \mathbb{C}[x, y, z]$.

- Das Ideal I besitzt Gröbnerbasis $G = \{xy - 1, xz - 1, y - z\}$.
- $G_1 = G \cap \mathbb{C}[y, z] = y - z$ und $G_2 = G \cap \mathbb{C}[z] = \emptyset$, d.h. $I_2 = \{0\}$.
- Damit ist jedes $z \in \mathbb{C}$ eine partielle Lösung.
- Wegen $y = z$ ist jedes $(y, z) = (c, c) \in \mathbb{C}^2$ eine partielle Lösung.
- Da $x = \frac{1}{y} = \frac{1}{z}$ lässt sich diese Lösung zu $(\frac{1}{c}, c, c) \in \mathbb{C}^3$ erweitern.
- Allerdings sind diese nur für $c = 0$ eine Lösung.
- D.h. alle Lösungen $(y, z) = (c, c), c \neq 0$ sind erweiterbar.

Erweiterungssatz

Satz Erweiterungssatz

Sei $I = \langle f_1, \ldots, f_m \rangle \subseteq \mathbb{C}[x_1, \ldots, x_n]$. Für $i = 1, \ldots, m$ sei

$$f_i = h_i(x_2, \ldots, x_n) x_1^{N_i} - \text{(Terme mit } \mathrm{grad}(x_1) \leq N_i) \text{ für } h_i \neq 0,\ N_i \in \mathbb{N}_0.$$

Sei $(a_2, \ldots, a_n) \in \mathbf{V}(I_1)$. Es existiert $a_1 \in \mathbb{C}$ mit $(a_1, \ldots, a_n) \in \mathbf{V}(I)$ falls

$$(a_1, \ldots, a_n) \notin \mathbf{V}(h_1, \ldots, h_m).$$

(ohne Beweis)

Beispiel: $I = \langle xy - 1, xz - 1 \rangle \subseteq \mathbb{C}[x, y, z]$

- $I_2 = \{0\}$ ist das erste Eliminationsideal von $I_1 = \langle y - z \rangle \subseteq \mathbb{C}[y, z]$.
- Es gilt $y - z = h(z) \cdot y - z$ mit $h(z) = 1$. D.h. $h(z) \neq 0$ für alle z.
- Damit lassen sich alle Lösungen $z = c$ zu $(y, z) = (c, c)$ erweitern.
- Es gilt $f_1 = \underbrace{y}_{h_1(y,z)} \cdot x - 1$ und $f_2 = \underbrace{z}_{h_2(y,z)} \cdot x - 1$.
- Ferner ist $\mathbf{V}(h_1(y, z), h_2(y, z)) = \{(0, 0)\}$.
- D.h. alle Lösungen außer $(y, z) = (0, 0)$ sind erweiterbar.

Hilberts schwacher Nullstellensatz

Satz Hilberts schwacher Nullstellensatz

Sei $I \in \mathbb{C}[x_1, \ldots, x_n]$ mit $\mathbf{V}(I) = \emptyset$. Dann gilt $I = \mathbb{C}[x_1, \ldots, x_n]$.

(ohne Beweis)

Satz Lösbarkeit von Gleichungssystemen in $\mathbb{C}$

Sei $I = \langle f_1, \ldots, f_m \rangle \in \mathbb{C}[x_1, \ldots, x_n]$, G reduzierte Gröbnerbasis von I. Falls $G \neq \{1\}$, dann besitzt das System $f_1 = \ldots = f_m = 0$ eine Lösung.

Beweis:

- Es gilt $\mathbb{C}[x_1, \ldots, x_n] = \langle 1 \rangle$. $\{1\}$ ist eine reduzierte Gröbnerbasis.
- D.h. falls $G \neq \{1\}$, dann gilt $I \neq \mathbb{C}[x_1, \ldots, x_n]$.
- Daraus folgt $\mathbf{V}(I) \neq \emptyset$ mit schwachem Nullstellensatz.
- Damit besitzt das Gleichungssystem mindestens eine Lösung.