

Gröbnerbaser og kryptosystemet HFE

Jon Inge Kolden

Master i fysikk og matematikk
Oppgaven levert: Juni 2009
Hovedveileder: Aslak Bakke Buan, MATH
Biveileder(e): Petter Andreas Bergh, MATH

Oppgavetekst

Oppgaven omhandler Gröbner-baser og kryptosystemet Hidden Field Equation (HFE), samt Gröbner-basisangrep på dette systemet.

Oppgaven gitt: 26. januar 2009

Hovedveileder: Aslak Bakke Buan, MATH

Sammendrag

I denne masteroppgaven ser vi nærmere på Gröbner-baser og kryptosystemet Hidden Field Equations.

Vi begynner med å se på flere algebraiske begreper knyttet til Gröbner-baser, og hvordan Gröbner-baser kan beregnes vha. Buchbergers algoritme. I den siste delen av første kapittel ser vi hvordan Gröbner-baser kan brukes til å løse polynomiske ligningssystemer.

Videre ser vi på kryptosystemet Hidden Field Equations. Vi begynner med den matematiske teorien som ligger bak HFE-systemet, for deretter å gi en beskrivelse av selve kryptingen. For å øke forståelsen av kryptosystemet, ser vi på et enkelt eksempel. Flere angrep på HFE har blitt forsøkt, og disse kan grovt sett deles inn i to klasser. Den ene typen angrep utnytter bestemte egenskaper i det konkrete kryptosystemet, mens den andre typen består av generelle algoritmer for å løse multivariate ligningssystemer. I denne oppgaven fokuserer vi på den siste typen, nærmere bestemt algoritmer som beregner en Gröbner-basis for et gitt ligningssystem. I kapittel 3 gjennomgår vi et Gröbner-basisangrep på HFE.

I den siste delen av oppgaven ser vi på koblingen mellom Gröbner-baser og lineær algebra. Vi ser deretter på forbedringer av Buchbergers originale algoritme. Vi studerer F4-algoritmen som tar i bruk lineær algebra, og en videreutvikling av F4, kalt F5. F5 tar utgangspunkt i å kutte ut unødvendige beregninger ved å bruke det såkalte F5-kriteriet. Et tilsvarende kriterium, formulert av Gebauer og Möller, blir også gjennomgått.

Forord

Denne oppgaven er skrevet for algebragruppen ved det matematiske institutt ved Norges teknisk-naturvitenskapelige universitet (NTNU). Oppgaven avslutter et fem år langt masterstudium, med fokusering på algebra de siste to årene. Dette har vært en fortsettelse av min prosjektoppgave forrige semester, som omhandlet Gröbner-baser og kryptosystemet HFE. Det har vært mye å gjøre, men når man får jobbe med noe av interesse, tenker man ikke så mye på slikt.

Jeg vil rette en takk til veileder Petter Andreas Bergh, som har kommet med tips og innspill underveis. Samtidig må jeg takke Aslak Bakke Buan, som har stått oppført som hovedveileder, og som også har vært tilgjengelig for hjelp. Til slutt vil jeg takke min medstudent Jarle Kvåle for så vel faglige diskusjoner som sosiale avbrekk.

Innhold

1 Gröbner-teori	1
1.1 Introduksjon	1
1.2 Affine varieteteter og idealer	1
1.3 Sortering av monomer i P	4
1.4 Divisjonsalgoritmen i P	6
1.5 Monomidealer	9
1.6 Gröbnerbaser	10
1.7 Egenskaper ved Gröbner-baser	12
1.8 Buchbergers algoritme	15
1.9 Løsing av polynomiske ligningssystem	18
2 Hidden Field Equations (HFE)	23
2.1 Introduksjon	23
2.2 Matematisk bakgrunn	23
2.3 Beskrivelse av krypteringen i HFE	24
2.4 Eksempel på bruk av HFE	26
2.5 Fordeler og ulemper med HFE	28
3 Kryptoanalyse av Hidden Field Equations (HFE) vha. Gröbner-baser	29
3.1 Introduksjon	29
3.2 Første HFE-utfordring knekt	30
3.3 Eksempel på Gröbner-basisangrep på HFE	30
4 Avanserte Gröbner-baseteknikker	33
4.1 Introduksjon	33
4.2 Kobling av Gröbner-baser og lineær algebra	33
4.3 Ensartet Buchberger-algoritme	34
4.4 Faugères F4-algoritme	37
4.5 Gebauer og Möller-installasjon og F5	42
5 Konklusjon	49
Referanser	50

Kapittel 1

Gröbner-teori

1.1 Introduksjon

Dette kapittelet omhandler temaet Gröbner-baser og teori som blir brukt i forbindelse med dette.

Teorien om Gröbner-baser for polynomringer ble utviklet av østerrikeren Bruno Buchberger i 1965. Han kalte opp denne teorien etter sin veileder Wolfgang Gröbner. I 2007 mottok Buchberger “ACM Paris Kanellakis Theory and Practice Award” for hans teori om Gröbner-baser.

En Gröbner-basis er en helt spesiell genererende undermengde av et ideal I i en polynomring R . Vi skal se her at et ideal kan ha (uendelig) mange Gröbner-baser, men at det finnes én basis, den reduserte Gröbner-basisen, som er bedre enn de andre. Buchberger har selv laget en algoritme for å finne Gröbner-baser i et ideal i en polynomring.

Men før vi kan gå løs på dette er det en del ting som må defineres og gjøres klart. Dette vil være affine varieteter, ordning av monomer i en polynomring $k[x_1, \dots, x_n]$, divisjonsalgoritmen i denne polynomringen, monomidealer m.m.

Gröbner-baser har blitt brukt i kryptosystemer, og angrep på kryptosystemer, noe vi kommer nærmere inn på i senere kapitler.

1.2 Affine varieteter og idealer

I dette delkapittelet introduseres idealer og affine varieteter, stort sett med samme notasjon som i [4]. En ring som dannes av mengden polynomer med koeffisienter i en ring, kalles en polynomring. Hvis vi opererer med én variabel, x , med koeffisienter p_i i en kropp k , skriver vi

$$p = \sum_i p_i x^i.$$

Vi sier da at $p \in k[x]$. Ringen $k[x]$ kalles da polynomringen i én variabel, x over k .

Et polynom i n variable x_1, \dots, x_n , med koeffisienter i kroppen k , blir definert på samme måte som et polynom i én variabel. Notasjonen er som følger:

$$f = \sum_{\alpha} c_{\alpha} x^{\alpha}, c_{\alpha} \in k.$$

Mengden med alle polynomer i x_1, \dots, x_n , med koeffisienter i kroppen k , skriver vi da $k[x_1, \dots, x_n]$. Her tas summen over et endelig antall n-tupler $\alpha = (\alpha_1, \dots, \alpha_n)$. Fra nå av skriver vi for enkelhets skyld kun P nå vi refererer til polynomringen over kroppen k i n variable, $k[x_1, \dots, x_n]$ (hvis ikke annet blir spesifisert).

Muligheten til å betrakte et polynom som en funksjon er det som gjør det mulig å koble algebra og geometri. Derfor introduseres det affine rommet, hvor geometriske figurer eksisterer.

Definisjon: Gitt en kropp k og et positivt heltall n , så definerer vi det affine rommet til å være mengden

$$k^n = \{(a_1, \dots, a_n) : a_1, \dots, a_n \in k\}.$$

Å evaluere et polynom f i $(a_1, \dots, a_n) \in k^n$, er en funksjon

$$f : k \rightarrow k^n,$$

hvor alle x_i er erstattet av a_i , for $1 \leq i \leq n$.

Når vi skal se på kompleksiteten til det å løse polynomiske ligningssystem, er begrepet varietet sentralt. Mengden av alle løsninger av et ligningssystem

$$f_1(x_1, \dots, x_n) = \dots = f_m(x_1, \dots, x_n) = 0.$$

kalles en affin varietet og defineres på følgende måte.

Definisjon: La k være en kropp og la f_1, \dots, f_m være polynomer i polynomringen over k i n variable, P . Da kan vi definere mengden av løsninger i k som den affine varieteten:

$$V(f_1, \dots, f_m) = \{(a_1, \dots, a_n) \in k^n : f_i(a_1, \dots, a_n) = 0, i = 1, \dots, m\}$$

Som vi skal se i lemmaet som følger, er både snittet av to varieteter og unionen av to varieteter, selv varieteter.

Lemma 1.2.1. *Hvis $V = V(f_1, \dots, f_s)$, $W = V(g_1, \dots, g_t) \subset k^n$ er affine varieteter, så er også*

$$\begin{aligned} V \cap W &= V(f_1, \dots, f_s, g_1, \dots, g_n) \\ V \cup W &= V(f_i, g_j : 1 \leq i \leq s, 1 \leq j \leq t) \end{aligned}$$

varieteter.

Bevis: Se [4]. □

For å kunne regne med varieteter er det nødvendig å definere et ideal.

Definisjon: En undermengde $I \subset P$ er et ideal hvis den tilfredsstiller følgende krav:

- (i) $0 \in I$.
- (ii) Hvis $f, g \in I$, så er $f + g \in I$.
- (iii) Hvis $f \in I$ og $h \in P$, så er $hf \in I$.

Idealet generert av et endelig antall polynomer er definert på følgende måte.

Definisjon: La f_1, \dots, f_m være polynomer i P . Definerer idealet

$$\langle f_1, \dots, f_m \rangle = \left\{ \sum_{i=1}^m h_i f_i : h_1, \dots, h_m \in P \right\}.$$

Hvis det finnes en endelig mengde av polynomer i P som genererer et gitt ideal, kaller vi denne mengden en basis. Hilberts basisteorem sier at ethvert ideal er endelig generert. Man skal imidlertid legge merke til at et gitt ideal kan ha mange ulike basiser.

Proposisjon 1.2.2. *Hvis f_1, \dots, f_s og g_1, \dots, g_t er basiser til det samme idealet i P , slik at $\langle f_1, \dots, f_s \rangle = \langle g_1, \dots, g_t \rangle$, så er $V(f_1, \dots, f_s) = V(g_1, \dots, g_t)$.*

Bevis: Enhver $f \in \langle f_1, \dots, f_s \rangle$ er også i $\langle g_1, \dots, g_t \rangle$ og kan derfor skrives som $h_1 g_1 + \dots + h_t g_t$, $h_i \in P$. Dermed vil enhver $a = (a_1, \dots, a_n) \in V(g_1, \dots, g_t)$ tilfredsstille $f(a) = 0$. Det samme argumentet gjelder for en $g \in \langle g_1, \dots, g_t \rangle$. Dette viser at begge varietetene består av de samme punktene, og de er dermed lik hverandre. □

Muligheten til å bytte basis for et ideal uten å endre varieteten, spiller en avgjørende rolle når det gjelder å løse polynomiske ligningssystemer. For å se på hvilke polynomer som kanselleres i den samme varieteten, introduseres et nytt algebraisk begrep.

Definisjon: La $V \subset k^n$ være en affin varietet. Da definerer vi

$$I(V) = \{f \in P : f(a_1, \dots, a_n) = 0 \text{ for alle } (a_1, \dots, a_n) \in V\}.$$

Den viktigste observasjonen her er at dette også er et ideal (se [4]). Noe som imidlertid ikke er så lett å se, er at $I(V(f_1, \dots, f_m))$ ikke behøver å være lik $\langle f_1, \dots, f_m \rangle$.

Lemma 1.2.3. *Hvis $f_1, \dots, f_m \in P$, så er $\langle f_1, \dots, f_m \rangle \subseteq I(V(f_1, \dots, f_m))$ (men likheten trenger ikke å intrefje).*

Bevis: Se [4]. □

1.3 Sortering av monomer i P

Det er viktig å få klargjort hvordan polynomene skal settes opp før man begynner å jobbe med dem. Det første vi gjør er å definere hvilke krav en sortering av monomer må oppfylle.

Definisjon: En sortering $>$ av monomer i P er en relasjon $>$ på $\mathbb{Z}_{\geq 0}^n$, eller ekvivalent, en relasjon på mengden av monomer x^α , $\alpha \in \mathbb{Z}_{\geq 0}^n$, som tilfredsstiller:

- (i) $>$ er en total (eller lineær) sortering i $\mathbb{Z}_{\geq 0}^n$.
- (ii) Hvis $\alpha > \beta$ og $\gamma \in \mathbb{Z}_{\geq 0}^n$, så er $\alpha + \gamma > \beta + \gamma$.
- (iii) $>$ er velordnet i $\mathbb{Z}_{\geq 0}^n$. Dvs. at alle ikke-tomme undermengder av $\mathbb{Z}_{\geq 0}^n$ har et minste element under $>$.

Etter hvert vil vi gjøre beregninger hvor det er avgjørende hva vi velger som ledende ledd i polynomet. Vil x_2^2 være større enn x_1x_2 ?

Det finnes mange måter å sortere monomer på, og her skal vi ta for oss tre av disse sorteringstypene.

Definisjon (leksikografisk sortering): La $\alpha = (\alpha_1, \dots, \alpha_n)$ og $\beta = (\beta_1, \dots, \beta_n)$ være elementer i $\mathbb{Z}_{\geq 0}^n$. Vi skriver $\alpha >_{\text{lex}} \beta$ hvis det første tallet fra venstre forskjellig fra null, i vektordifferansen $\alpha - \beta$, er positivt. Vi skriver $x^\alpha >_{\text{lex}} x^\beta$ hvis $\alpha >_{\text{lex}} \beta$.

Under følger et par eksempler på lex-sorteringen.

1. $(2, 0, -3) >_{\text{lex}} (1, 2, -3)$ fordi $\alpha - \beta = (1, -2, 0)$.
2. $(2, 1, -2) >_{\text{lex}} (2, 0, -1)$ fordi $\alpha - \beta = (0, 1, -1)$.

Variablene x_1, \dots, x_n blir sortert på vanlig måte når man bruker lex-sorteringen:

$$(1, 0, \dots, 0) >_{\text{lex}} (0, 1, 0, \dots, 0) >_{\text{lex}} \cdots >_{\text{lex}} (0, \dots, 0, 1) \Rightarrow x_1 >_{\text{lex}} x_2 >_{\text{lex}} \cdots >_{\text{lex}} x_n.$$

Når vi regner med tre variable, x_1, x_2, x_3 , vil vi bruke x, y, z . Vi antar også at den alfabetiske sorteringen $x > y > z$ brukes til å definere den leksikografiske sorteringen dersom ikke annet blir påpekt. Det er viktig å notere seg *hvilken* sortering som brukes. F.eks. vil en lex-sortering $z > y > x$ gi at $z^2 >_{\text{lex}} zy^5x^7$.

I enkelte sammenhenger vil det imidlertid være hensiktsmessig å sortere etter den totale graden til monomene. En måte å gjøre dette på er å ta i bruk såkalt leksikografisk totalsortering, som vi definerer slik:

Definisjon (leksikografisk totalsortering): La α og $\beta \in \mathbb{Z}_{\geq 0}^n$. Vi skriver $\alpha >_{\text{grlex}} \beta$ hvis

$$|\alpha| = \sum_{i=1}^n \alpha_i > |\beta| = \sum_{i=1}^n \beta_i, \text{ eller } |\alpha| = |\beta| \text{ og } \alpha >_{\text{lex}} \beta$$

Her kommer to eksempler på grlex-sortering:

1. $(1, 4, 7) >_{\text{grlex}} (2, 8, 0)$ fordi $|(1, 4, 7)| = 12 > |(2, 8, 0)| = 10$.
2. $(1, 2, 3) >_{\text{grlex}} (1, 1, 4)$ fordi $|(1, 2, 3)| = 6 = |(1, 1, 4)|$ og $(1, 2, 3) >_{\text{lex}} (1, 1, 4)$

Legg merke til at variablene x_1, \dots, x_n blir sortert på vanlig måte siden alle disse er av grad én.

Definisjon (Omvendt leksikografisk totalsortering): La α og $\beta \in \mathbb{Z}_{\geq 0}^n$. Vi sier da at $\alpha >_{\text{grevlex}} \beta$ hvis

$$|\alpha| = \sum_{i=1}^n \alpha_i > |\beta| = \sum_{i=1}^n \beta_i, \text{ eller } |\alpha| = |\beta|$$

og tallet lengst til høyre, som ikke er null, i differansen $\alpha - \beta$, er negativt.

Et par eksempler som viser hvordan revlex-sorteringen virker:

1. $(1, 2, 3) >_{\text{grevlex}} (2, 3, 0)$ fordi $|(1, 2, 3)| = 6 > |(2, 3, 0)| = 5$.
2. $(1, 2, 3) >_{\text{grevlex}} (1, 1, 4)$ fordi $|(1, 2, 3)| = 6 = |(1, 1, 4)|$ og $(1, 2, 3) - (1, 1, 4) = (0, 1, -1)$.

Vi ser at sorteringen ligner veldig på grlex. Forskjellen mellom dem oppstår når den totale graden til monomene er like. Da vil grlex sortere slik at det monomet med høyest grad på den største variabelen kommer først, mens revlex ordner monomene sånn at monomet hvor den minste variabelen er minst, kommer først.

Definisjon: La $f = \sum_a \alpha_a x^a$ være et polynom, forskjellig fra null, i P og la $>$ være en monomsortering.

(i) Multigraden til f er

$$\text{multideg}(f) = \max(\alpha \in \mathbb{Z}_{\geq 0}^n : \alpha_a \neq 0)$$

(ii) Ledende koeffisient til f er

$$\text{LC}(f) = \alpha_{\text{multideg}(f)} \in k.$$

(iii) Ledende monom til f er

$$\text{LM}(f) = x^{\text{multideg}(f)}$$

(med koeffisient 1).

(iv) Ledende ledd til f er

$$\text{LT}(f) = \text{LC}(f) \cdot \text{LM}(f)$$

For å illustrere disse definisjonene kan vi se på et enkelt eksempel.

La $f = 9x^2y^3 - 2x^6 + 7z^4 - 5x^3y^2z^4$ og la $>$ være lex-sortering. Vi har da:

$$\text{multideg}(f) = (6, 0, 0),$$

$$\text{LC}(f) = -2,$$

$$\text{LM}(f) = x^6,$$

$$\text{LT} = -2x^6$$

1.4 Divisjonsalgoritmen i P

I en polynomring med én variabel har vi en kjent divisjonsalgoritme - vanlig polynomdivisjon. Når vi nå jobber med polynomringen med flere variable kan vi ikke bruke polynomdivisjon på denne måten. Vi må lage oss en ny divisjonsalgoritme. Den nye algoritmen skal også gjøre divisjon av et polynom på flere polynomer mulig. Vi vil altså dividere $f \in P$ på $f_1, \dots, f_s \in P$. Grunntanken bak den nye algoritmen er den samme som for standard polynomdivisjon. Teoremet som følger angir den generelle divisjonsalgoritmen.

Teorem 1.4.1 (Divisjonsalgoritmen i P). *Velg en monomsortering $> i \mathbb{Z}_{\geq 0}^n$ og la $F = (f_1, \dots, f_s)$ være et sortert s-tuppel av polynomer i P . Da kan alle $f \in P$ skrives som*

$$f = a_1f_1 + \dots + a_sf_s + r,$$

hvor $a_i, r \in P$, hvor r er enten 0 eller en lineærkombinasjon (med koeffisienter i k) av monomer, og ingen av disse er delelig med noen av $\text{LT}(f_1), \dots, \text{LT}(f_s)$. Vi kaller r for resten til f ved divisjon med F . Videre har vi at hvis $a_if_i \neq 0$, så er

$$\text{multideg}(f) \geq \text{multideg}(a_if_i \neq 0).$$

Bevis: Se [4] for bevis. □

Eksempel: I dette eksempelet skal vi dele $f = x^3y^2 + x^2y^3 + y^2$ med $f_1 = x^2y^2 - 1$ og $f_2 = y^2 - 1$. Vi bruker lex-sortering med $x > y$. Når man bruker divisjonsalgoritmen for flere variable må man holde orden på kvotientene, i dette tilfellet k_1 og k_2 , mens man dividerer.

Divisjonen foregår på følgende måte: Man sjekker om $\text{LT}(f_1) = x^2y^2$ eller $\text{LT}(f_2) = y^2$ deler $\text{LT}(f) = x^3y^2$. Her ser vi at $\text{LT}(f_1)$ gjør det, så vi dividerer først med den. (Dersom begge de to ledende leddene deler $\text{LT}(f)$, så bruker man den funksjonen som listes opp først, altså f_1 .)

$\text{LT}(f) = x \cdot \text{LT}(f_1)$, der x er kvotienten, og k_1 settes lik x . Når man deler f på $x \cdot f_1$ står man igjen med $f^* = x^2y^3 + x + y^2$.

$$\begin{aligned} k_1 &= x \\ f_1 &= x^2y^2 - 1 \\ f_2 &= y^2 - 1 \\ f^* &= x^2y^3 + x + y^2 \end{aligned}$$

Ser nå at $\text{LT}(f_1)$ igjen deler ledende ledd i uttrykket f^* og at $y \cdot \text{LT}(f_1) = \text{LT}(f^*)$. Deler derfor f^* på $y \cdot f_1$ og får en ny $f^* = x + y^2 + y$.

$$\begin{aligned} k_1 &= x + y \\ f_1 &= x^2y^2 - 1 \\ f_2 &= y^2 - 1 \\ f^* &= x + y^2 + y \end{aligned}$$

Ser nå at hverken $\text{LT}(f_1)$ eller $\text{LT}(f_2)$ deler $\text{LT}(f^*)$. Men f^* er likevel ikke resten. $\text{LT}(f_2)$ deler nemlig y^2 , så vi flytter det ledende leddet, x , over til resten r , før vi fortsetter å dividere.

$$\begin{aligned} r &= x \\ k_1 &= x + y \\ f_1 &= x^2y^2 - 1 \\ f_2 &= y^2 - 1 \\ f^* &= y^2 + y \end{aligned}$$

NB! Dette er en situasjon som aldri oppstår i divisjonsalgoritmen for én variabel.

Etter å ha fjernet x , står vi igjen med $f^* = y^2 + y$. $\text{LT}(f^*) = \text{LT}(f_2)$, så vi deler f^* på f_2 og får $f^* = y + 1$.

$$\begin{aligned} r &= x \\ k_1 &= x + y \\ k_2 &= 1 \\ f_1 &= x^2y^2 - 1 \\ f_2 &= y^2 - 1 \\ f^* &= y + 1 \end{aligned}$$

Nå har vi igjen et uttrykk som ingen av de to funksjonene deler. Dermed flyttes $\text{LT}(f^*) = y$

til resten r .

$$\begin{aligned} r &= x + y \\ k_1 &= x + y \\ k_2 &= 1 \\ f_1 &= x^2y^2 - 1 \\ f_2 &= y^2 - 1 \\ f^* &= 1 \end{aligned}$$

Det samme gjentas for den nye $f^* = 1$ slik at $f^* = 0$. Da har vi funnet resten $r = x + y + 1$.

$$\begin{aligned} r &= x + y + 1 \\ k_1 &= x + y \\ k_2 &= 1 \\ f_1 &= x^2y^2 - 1 \\ f_2 &= y^2 - 1 \\ f^* &= 0 \end{aligned}$$

Dermed har vi

$$x^3y^2 + x^2y^3 + y^2 = (x + y) \cdot (x^2y^2 - 1) + 1 \cdot (y^2 - 1) + x + y + 1.$$

Legg merke til at restleddet er en sum av monomer, hvor ingen av dem er delelig med hverken $\text{LT}(f_1)$ eller $\text{LT}(f_2)$. Dette er en egenskap som resten i divisjonsalgoritmen alltid skal ha. Resten r er imidlertid ikke unik, slik som i tilfellet med bare én variabel. Den er avhengig av rekkefølgen på divisorene f_1 og f_2 , noe vi ser i følgende eksempel.

Eksempel: La oss bruke de samme tre polynomene som i forrige eksempel, bare med den forskjell at vi bytter rekkefølgen på divisorene. Vi har altså at $f = x^3y^2 + x^2y^3 + y^2$ skal deles på $f_1 = y^2 - 1$ og $f_2 = x^2y^2 - 1$. Igjen bruker vi lex-sortering med $x > y$. Divisjonsalgoritmen gir oss nå at f kan skrives slik:

$$f = (x^3 + x^2y + 1) \cdot (y^2 - 1) + 0 \cdot (x^2y^2 - 1) + x^3 + x^2y + 1.$$

Som vi ser, så er resten nå $r = x^3 + x^2y + 1$. Vi legger merke til at denne resten er forskjellig fra den vi fikk i det andre eksempelet. Dette er et problem vi gjerne vil gjøre noe med, og i denne sammenhengen vil Gröbner-baser komme inn i bildet.

1.5 Monomidealer

Definisjon: Et ideal $I \subset P$ er et monomideal hvis det finnes en undermengde $A \subset \mathbb{Z}_{\geq 0}^n$ slik at I består av alle polynomer som er endelige summer på formen $\sum_{\alpha \in A} h_\alpha x^\alpha$, hvor $h_\alpha \in P$. I dette tilfellet skriver vi $I = \langle x^\alpha : \alpha \in A \rangle$.

Lemma 1.5.1. La $I = \langle x^\alpha : \alpha \in A \rangle$ være et monomideal. Da ligger et monom x^β i I hvis og bare hvis x^β er delelig med x^α for en $\alpha \in A$.

Bevis: Hvis x^α deler x^β for en $\alpha \in A$, så er $x^\beta \in I$ ved definisjonen av et ideal. Tilsvarende har vi at hvis $x^\beta \in I$, så er $x^\beta = \sum_{i=1}^s h_i x^{\alpha(i)}$, hvor $h_i \in P$ og $\alpha(i) \in A$. Dersom vi utvider hver h_i til en lineærkombinasjon av monomer, så ser vi at hvert ledd på høyresiden i ligningen er delelig på en $x^{\alpha(i)}$. Dermed må også venstresiden, x^β , ha denne egenskapen. \square

Om et gitt polynom f ligger i et monomideal eller ikke, kan bestemmes ved å se på monomene i f , noe neste lemma viser.

Lemma 1.5.2. La I være et monomideal, og la $f \in P$. Da er følgende utsagn ekvivalente:

- (i) $f \in I$.
- (ii) Alle ledd i f ligger i I .
- (iii) f er en k -lineær kombinasjon av monomene i I .

Bevis: Implikasjonene $(\text{iii}) \Rightarrow (\text{ii}) \Rightarrow (\text{i})$ er lette å se. Implikasjonen $(\text{i}) \Rightarrow (\text{iii})$ følger av beviset for Lemma 1.5.1. \square

En umiddelbar konsekvens av del (iii) i lemmaet er at et monomideal er unikt bestemt av monomene sine. Dette gir oss følgende korollar:

Korollar 1.5.3. To monomidealer er like hvis og bare hvis de inneholder de samme monomene.

Hovedresultatet i dette underkapittelet sier at alle monomidealer i P er endiggenererte. Dette slås fast i det som kalles Dicksons lemma.

Teorem 1.5.4 (Dicksons lemma). La $I = \langle x^\alpha : \alpha \in A \rangle \subseteq P$ være et monomideal. Da kan I skrives på formen $I = \langle x^{\alpha(1)}, \dots, x^{\alpha(s)} \rangle$, hvor $\alpha(1), \dots, \alpha(s) \in A$. Sagt på en annen måte; I har en endelig generatormengde.

Bevis: Se [4]. \square

1.6 Gröbnerbaser

Nå skal vi se på såkalte ledende ledd-ideal. Vi har lyst til å finne gode generatormengder til idealene vi jobber med, slik at divisjonsalgoritmen vår vil fungere godt. Den kanskje viktigste ideen er at når vi velger en monomsortering, så har alle polynomer $f \in P$ et unikt ledende ledd $\text{LT}(f)$. Da kan vi, for ethvert ideal I , definere idealet av ledende ledd på følgende måte:

Definisjon: La $I \subset P$ være et ideal ulik null.

- (i) Mengden av ledende ledd av elementer i I kaller vi $\text{LT}(I)$. Altså

$$\text{LT}(I) = \{cx^\alpha : \text{det eksisterer } f \in I \text{ med } \text{LT}(f) = cx^\alpha\}.$$

- (ii) Idealet generert av elementene i $\text{LT}(I)$ kaller vi $\langle \text{LT}(I) \rangle$.

Det er viktig å merke seg at dersom vi har et ideal I med en endelig genererende mengde $\langle f_1, \dots, f_i \rangle = I$, så kan $\langle \text{LT}(f_1), \dots, \text{LT}(f_i) \rangle$ og $\langle \text{LT}(I) \rangle$ være ulike ideal. Men uansett er $\langle \text{LT}(f_1), \dots, \text{LT}(f_i) \rangle$ inneholdt i $\langle \text{LT}(I) \rangle$. Vi tar for oss et eksempel:

La $I = \langle f_1, f_2 \rangle$, hvor $f_1 = x^3 - 2xy$ og $f_2 = x^2y - 2y^2 + x$, og bruk grlex-sorteringen på monomene i $k[x, y]$. Da er

$$x \cdot (x^2y - 2y^2 + x) - y \cdot (x^3 - 2xy) = x^2$$

slik at $x^2 \in I$. Altså er $x^2 = \text{LT}(x^2) \in \langle \text{LT}(I) \rangle$. Men $\text{LT}(f_1) = x^3$ og $\text{LT}(f_2) = x^2y$, så $x^2 \notin \langle \text{LT}(f_1), \text{LT}(f_2) \rangle$.

Det viser seg at $\langle \text{LT}(I) \rangle$ er et monomideal.

Proposisjon 1.6.1. *La $I \subset P$ være et ideal.*

- (i) $\langle \text{LT}(I) \rangle$ er et monomideal.

- (ii) Det finnes elementer $g_1, \dots, g_t \in I$ slik at $\langle \text{LT}(I) \rangle = \langle \text{LT}(g_1), \dots, \text{LT}(g_t) \rangle$.

Bevis: (i) De ledende monomene $\text{LM}(g)$ av elementer $g \in I - \{0\}$ genererer monomidealet $\langle \text{LM}(g) : g \in I - \{0\} \rangle$. Siden $\text{LM}(g)$ er en konstant ulik null multiplisert med $\text{LT}(g)$, så har vi $\langle \text{LM}(g) : g \in I - \{0\} \rangle = \langle \text{LT}(I) \rangle$. Dermed er $\langle \text{LT}(I) \rangle$ et monomideal.

(ii) Siden $\langle \text{LT}(I) \rangle$ er generert av monomene $\text{LM}(g)$ for $g \in I \setminus \{0\}$, så gir Dicksons lemma at $\langle \text{LT}(I) \rangle = \langle \text{LM}(g_1), \dots, \text{LM}(g_t) \rangle$ for endelig mange $g_1, \dots, g_t \in I$. Siden $\text{LM}(g_i)$ er $\text{LT}(g_i)$ multiplisert med en konstant ulik null, så følger det at $\langle \text{LT}(I) \rangle = \langle \text{LT}(g_1), \dots, \text{LT}(g_t) \rangle$. \square

Nå kan vi bruke Proposisjon 1.6.1 og divisjonsalgoritmen til å bevise eksistensen til en endelig genererende mengde for ethvert polynomisk ideal.

Teorem 1.6.2 (Hilberts basisteoremet). *Ethvert ideal $I \subset P$ har en endelig genererende mengde. Dvs. at $I = \langle g_1, \dots, g_t \rangle$ for noen $g_1, \dots, g_t \in I$.*

Bevis: Hvis $I = \{0\}$, så velger vi $\{0\}$ som vår genererende mengde, og denne er åpenbart endelig. Dersom I inneholder polynomer forskjellige fra null, så kan en generende mengde, g_1, \dots, g_t , til I konstrueres på følgende vis:

Ved Proposisjon 1.6.1 finnes det $g_1, \dots, g_t \in I$ slik at $\langle \text{LT}(I) \rangle = \langle \text{LT}(g_1), \dots, \text{LT}(g_t) \rangle$. Vi hevder at $I = \langle g_1, \dots, g_t \rangle$. Det er opplagt at $\langle g_1, \dots, g_t \rangle \subset I$ siden hver $g_i \in I$. Så velger vi et vilkårlig polynom $f \in I$. Vi bruker divisjonsalgoritmen fra underkapittel 1.4 til å dele f på $\langle g_1, \dots, g_t \rangle$. Da får vi et uttrykk på formen

$$f = a_1 g_1 + \dots + a_t g_t + r$$

hvor ingen av leddene i r er delelig med $\text{LT}(g_i)$ for $1 \leq i \leq t$. Vi hevder nå at $r = 0$. For å se dette lettere, så skriver vi om uttrykket til

$$r = f - a_1 g_1 - \dots - a_t g_t \in I.$$

Dersom $r \neq 0$, så vil $\text{LT}(r) \in \langle \text{LT}(I) \rangle = \langle \text{LT}(g_1), \dots, \text{LT}(g_t) \rangle$, og ved Lemma 1.5.1 følger det at $\text{LT}(r)$ må være delelig med en $\text{LT}(g_i)$. Dette strider mot definisjonen av en rest, så r må være null. Da har vi

$$f = a_1 g_1 + \dots + a_t g_t \in \langle g_1, \dots, g_t \rangle,$$

noe som viser at $I \subset \langle g_1, \dots, g_t \rangle$. Dermed har vi bevist det vi skulle. \square

Den genererende mengden $\langle g_1, \dots, g_t \rangle$ som vi brukte i beviset til Hilberts basisteorem, har den spesielle egenskapen at $\langle \text{LT}(g_1), \dots, \text{LT}(g_t) \rangle = \langle \text{LT}(I) \rangle$. Slike generatormengder definerer vi på følgende vis:

Definisjon: Gitt en monomsortering. En endelig undermengde $G = \{g_1, \dots, g_t\}$ av et ideal I sies å være en Gröbner-basis hvis

$$\langle \text{LT}(g_1), \dots, \text{LT}(g_t) \rangle = \langle \text{LT}(I) \rangle.$$

Ekvivalent, men mer uformelt, kan vi si at en mengde $\{g_1, \dots, g_t\} \subset I$ er en Gröbner-basis til I hvis og bare hvis ledende ledd til ethvert element i I er delelig med en av $\text{LT}(g_i)$ -ene. Beviset til Teorem 1.6.2 gir også følgende resultat:

Korollar 1.6.3. *Gitt en monomsortering. Da har alle idealer $I \subset P$ utenom $\{0\}$ en Gröbner-basis. Videre, så er enhver Gröbner-basis for et ideal I en genererende mengde i I .*

Til slutt i dette delkapittelet tar vi med et par viktige konsekvenser av Hilberts basisteorem. Den første omhandler en økende følge av idealer i P ,

$$I_1 \subset I_2 \subset I_3 \subset \dots$$

og sier at denne såkalte stigende kjeden stabiliseres ved et bestemt tidspunkt, dvs. at alle idealer i kjeden til slutt vil være like. Dette er en viktig betingelse når man skal vise at algoritmer som beregner Gröbner-baser, terminerer.

Teorem 1.6.4 (Stigende kjede-betingelsen). La

$$I_1 \subset I_2 \subset I_3 \subset \dots$$

være en stigende kjede av idealer i P . Da eksisterer det en $N \geq 1$ slik at

$$I_N = I_{N+1} = I_{N+2} = \dots$$

Bevis: Se [4]. □

Mens den første konsekvensen er algebraisk, er den andre av geometrisk art. Den sier at varieteten som er knyttet til en mengde av polynomer er like varieteten av idelet generert av polynomengden.

Definisjon: La $I \subset P$ være et ideal. Med $V(I)$ betegner vi mengden

$$\{(a_1, \dots, a_n) \in k^n : f(a_1, \dots, a_n) = 0 \text{ for alle } f \in I\}.$$

Proposisjon 1.6.5. La $V(I)$ være en affin varietet. Da har vi at hvis $I = \langle f_1, \dots, f_s \rangle$, så er $V(I) = V(f_1, \dots, f_s)$.

Bevis: Se [4] for bevis. □

1.7 Egenskaper ved Gröbner-baser

I forrige underkapittel definerte vi hva Gröbner-baser er. Nå skal vi se på fordelene ved bruken av Gröbner-basene. Vi vil også vise hvordan vi kan kontrollere om en gitt generatormengde er en Gröbner-basis. La oss begynne med å bevise at resten er unik når vi deler på en Gröbner-basis.

Proposisjon 1.7.1. La $G = \{g_1, \dots, g_t\}$ være en Gröbner-basis for et ideal $I \subset P$ og la $f \in P$. Da finnes det en unik $r \in P$ med følgende egenskaper:

- (i) Ingen av leddene i r er delelig med noen av $LT(g_1), \dots, LT(g_t)$.
- (ii) Det finnes en $g \in I$ slik at $f = g + r$.

Polynomet r er resten av en divisjon av f med G når man bruker divisjonsalgoritmen, uansett hvordan elementene i G er listet.

Bevis: Divisjonsalgoritmen gir $f = a_1g_1 + \dots + a_tg_t + r$, der r tilfredsstiller (i). Ved å sette $g = a_1g_1 + \dots + a_tg_t \in I$. Dette beviser eksistensen til r .

For å bevise unikhet antar vi at $f = g + r = g' + r'$ tilfredsstiller (i) og (ii). Da har vi $r - r' = g - g' \in I$, slik at hvis $r \neq r'$, så er $LT(r - r') \in \langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_t) \rangle$. Ved Lemma 1.5.1 følger det at $LT(r - r')$ er delelig med en eller annen $LT(g_i)$. Dette er umulig siden ingen ledd av hverken r eller r' er delelig med en av $LT(g_1), \dots, LT(g_t)$. Dermed må $r - r' = 0$ og unikhet er bevist.

Den siste delen av proposisjonen følger av unikheten til resten r . □

Korollar 1.7.2. La $G = \{g_1, \dots, g_t\}$ være en Gröbner-basis for et ideal $I \subset P$ og la $f \in P$. Da er $f \in I$ hvis og bare hvis resten av en divisjon av f med G er null.

Bevis: Hvis resten er null, så har vi allerede slått fast at $f \in I$. Tilsvarende har vi at dersom $f \in I$, så tilfredsstiller $f = f + 0$ de to kravene i Proposisjon 1.7.1. Da følger det at resten til f , når man deler på G , er 0. \square

Selv om resten er unik kan koeffisientene, a_i , som oppstår ved bruk av divisjonsalgoritmen, variere hvis generatorene bytter rekkefølge. Siden resten blir unik forsvinner problemet med at resten til et polynom i idealet blir ulik null.

Nå ser vi på hvordan vi kan sjekke om en gitt generatormengde er en Gröbner-basis.

Definisjon: Vi skriver \bar{f}^F for resten av en divisjon av f med det sorterte s-tupplet $F = (f_1, \dots, f_s)$. Hvis F er en Gröbner-basis, så kan vi betrakte F som en mengde (uten noen særskilt sortering).

Med $F = (x^2y - y^2, x^4y^2 - y^2) \in k[x, y]$ og lex-sortering, får vi f.eks. $\overline{x^5y}^F = xy^3$ siden divisjonsalgoritmen gir at

$$x^5y = (x^3 + xy)(x^2y - y^2) + 0 \cdot (x^4y^2 - y^2) + xy^3.$$

Nå skal vi se på hvordan vi kan se om en genererende mengde, f_1, \dots, f_s , til et ideal er en Gröbner-basis. Årsaken til at det kan være vanskelig å se dette, er at polynomkombinasjoner av f_i -er, der de ledende leddene ikke er med i idealet generert av $\text{LT}(f_i)$, kan oppstå. Dette kan f.eks. skje ved at de ledende leddene i $ax^\alpha f_i - bx^\beta f_j$ kansellerer, slik at bare mindre ledd står igjen. Samtidig er $ax^\alpha f_i - bx^\beta f_j \in I$, så de ledende leddene er i $\langle \text{LT}(I) \rangle$. For å jobbe med dette kanselleringsfenomenet definerer vi følgende:

Definisjon: La $f, g \in P$ være polynomer som ikke er null.

(i) Hvis $\text{multideg}(f) = \alpha$ og $\text{multideg}(g) = \beta$, la $\gamma = (\gamma_1, \dots, \gamma_n)$, hvor $\gamma_i = \max(\alpha_i, \beta_i)$ for hver i . Vi kaller x^γ det minste felles multiplum til $\text{LM}(f)$ og $\text{LM}(g)$, og skriver $x^\gamma = \text{LCM}(\text{LM}(f), \text{LM}(g))$.

(ii) S-polynomet til f og g er kombinasjonen

$$S(f, g) = \frac{x^\gamma}{\text{LT}(f)} \cdot f - \frac{x^\gamma}{\text{LT}(g)} \cdot g.$$

La f.eks. $f = 5xz^2 + 3y$, $g = xy^2z - xz^4$ med lex-sortering. Da ser vi at $\text{multideg}(f) = (1, 0, 2)$ og $\text{multideg}(g) = (1, 2, 1)$, så $\gamma = (1, 2, 2)$. Dermed får vi

$$S(f, g) = \frac{xy^2z^2}{5xz^2} \cdot f - \frac{xy^2z^2}{xy^2z} \cdot g.$$

$$\begin{aligned}
&= \frac{1}{5}y^2 \cdot f - z \cdot g \\
&= xz^5 + \frac{3}{5}y^3
\end{aligned}$$

Et S-polynom $S(f, g)$ er “designet” for å produsere kansellering av ledende ledd. Det neste lemmaet viser faktisk at enhver kansellering av ledende ledd i polynomer med samme multigrad, kommer fra en slik type kansellering.

Lemma 1.7.3. *Anta at vi har en sum $\sum_{i=1}^s c_i f_i$, hvor $c_i \in k$ og $\text{multideg}(f_i) = \delta \in \mathbb{Z}_{\geq 0}^n$ for alle i . Hvis $\text{multideg}(\sum_{i=1}^s c_i f_i) < \delta$, så er $\sum_{i=1}^s c_i f_i$ en lineærkombinasjon, med koeffisienter i k , av S-polynomene $S(f_j, f_k)$ for alle i . Videre har hver $S(f_i, f_k)$ multideg $< \delta$.*

Bevis: Se [4]. □

Dersom f_1, \dots, f_s tilfredsstiller hypotesen i Lemma 1.7.3, så får vi en ligning på formen

$$\sum_{i=1}^s c_i f_i = \sum_{j,k} c_{jk} S(f_j, f_k).$$

La oss se nærmere på hvor kanselleringen skjer. I summen på venstre side har alle tillegg $c_i f_i$ multigrad δ , så kanselleringen skjer etter addisjonen. På høyre side, derimot, har alle tilleggene $c_{jk} S(f_j, f_k)$ multigrad mindre enn δ , så kanselleringen har allerede skjedd. Dette betyr at all kansellering skyldes S-polynomene. Ved å bruke S-polynomer og Lemma 1.7.3 kan vi nå komme fram til en betingelse, kalt Buchbergers kriterium, for når en generatormengde til et ideal er en Gröbner-basis.

Teorem 1.7.4 (Buchbergers kriterium). *La I være et ideal i en polynomring. Da er en genererende mengde, $G = \{g_1, \dots, g_t\}$, for I en Gröbner-basis for I hvis og bare hvis det for alle par $i \neq j$ er slik at resten fra divisjon av $S(g_i, g_j)$ med G er null.*

Bevis: Se [4]. □

Buchbergers kriterium er et av nøkkelresultatene innen Gröbner-teori. Foreløpig har vi sett at Gröbner-baser har mange fine egenskaper, men det har vært vanskelig å bestemme om en gitt genererende mengde til et ideal er en Gröbner-basis eller ikke. Buchbergers kriterium gjør dette lett. Som et eksempel på hvordan kriteriet kan brukes, ser vi på idealet $I = \langle y - x^2, z - x^3 \rangle$ i $\mathbb{R}[x, y, z]$. Vi hevder at $G = \{y - x^2, z - x^3\}$ er en Gröbner-basis med lex-sortering $y > z > x$. For å vise dette, ser vi på S-polynomet

$$S(y - x^2, z - x^3) = \frac{yz}{y}(y - x^2) - \frac{yz}{z}(z - x^3) = -zx^2 + yx^3.$$

Ved bruk av divisjonsalgoritmen får vi

$$-zx^2 + yx^3 = x^3 \cdot (y - x^2) + (-x^2)(z - x^3) + 0,$$

slik at $\overline{S(y - x^2, z - x^3)}^G = 0$. Ved Buchbergers kriterium er dermed G en Gröbner-basis for I . Det kan nevnes at G ikke er en Gröbner-basis med lex-sortering $x > y > z$.

1.8 Buchbergers algoritme

Vi har sett at ethvert ideal i P forskjellig fra 0, har en Gröbner-basis. Men hvordan skal vi lage en Gröbner-basis for et gitt ideal i P ? Vi illustrerer ved hjelp av et eksempel.

Eksempel: Se på ringen $k[x, y]$ med grlex-sortering, og la $I = \langle f_1, f_2 \rangle = \langle x^3 - 2xy, x^2y - 2y^2 + x \rangle$. Siden $\text{LT}(S(f_1, f_2)) = x^2 \notin \langle \text{LT}(f_1), \text{LT}(f_2) \rangle$, er ikke $\langle f_1, f_2 \rangle$ en Gröbner-basis for I .

For å konstruere en Gröbner-basis er det naturlig å først prøve å utvide den originale genererende mengden ved å legge til polynomer i I . På en måte legger vi ikke til noe nytt, men dersom vi kommer fram til en Gröbner-basis vil det være verdt det. Men hva skal vi legge til? Vi tar utgangspunkt i S-polynomene. Vi har at $S(f_1, f_2) = -x^2 \in I$, og resten ved divisjon med $F = (f_1, f_2)$ er $-x^2$, som ikke er null. Derfor inkluderer vi denne resten i den genererende mengden, som en ny generator $f_3 = -x^2$. Hvis vi setter $F_1 = (f_1, f_2, f_3)$, så kan vi bruke Buchbergers kriterium til å sjekke om denne nye mengden er en Gröbner-basis for I . Vi finner følgende:

$$S(f_1, f_2) = f_3, \text{ så}$$

$$\overline{S(f_1, f_2)}^{F_1} = 0$$

$$S(f_1, f_3) = (x^3 - 2xy) - (-x)(-x^2) = -2xy, \text{ men}$$

$$\overline{S(f_1, f_3)}^{F_1} = -2xy \neq 0.$$

Siden ikke alle restene er null, må vi legge til $f_4 = -2xy$ til den genererende mengden. Vi får da en ny mengde $F_2 = (f_1, f_2, f_3, f_4)$. Nå har vi

$$\overline{S(f_1, f_2)}^{F_2} = \overline{S(f_1, f_3)}^{F_2} = 0$$

$$S(f_1, f_4) = y(x^3 - 2xy) - \left(-\frac{1}{2}\right)x^2(-2xy) = -2xy^2 = yf_4, \text{ så}$$

$$\overline{S(f_1, f_4)}^{F_2} = 0$$

$$S(f_2, f_3) = (x^2y - 2y^2 + x) - (-y)(-x^2) = -2y^2 + x, \text{ men}$$

$$\overline{S(f_2, f_3)}^{F_2} = -2y^2 + x \neq 0.$$

Derfor må vi legge til $f_5 = -2y^2 + x$ i den genererende mengden F_2 , og får da $F_3 = \{f_1, f_2, f_3, f_4, f_5\}$. Nå kan vi finne at

$$\overline{S(f_i, f_j)}^{F_3} = 0 \text{ for alle } 1 \leq i \leq j \leq 5.$$

Buchbergers kriterium gir nå at F_3 er en Gröbner-basis med grlex for I .

$$\{f_1, f_2, f_3, f_4, f_5\} = \{x^3 - 2xy, x^2y - 2y^2 + x, -x^2, -2xy, -2y^2 + x\}.$$

Dette eksempelet illustrerer en generell algoritme, kalt Buchbergers algoritme.

Teorem 1.8.1 (Buchbergers algoritme). La $I = \langle f_1, \dots, f_s \rangle \neq 0$ være et ideal i en polynomring. Da kan en Gröbner-basis for I konstrueres via et endelig antall steg i følgende algoritme:

```

Input:  $F = (f_1, \dots, f_s)$ 
Output: En Gröbner-basis  $G = (g_1, \dots, g_t)$  for  $I$ , med  $F \subset G$ 
 $G := F$ 
repeat
     $G' := G$ 
    for hvert par  $\{p, q\}, p \neq q$  i  $G'$  do
         $S := \overline{S(p, q)}^{G'}$ 
        if  $S \neq 0$  then
             $G := G \cup \{S\}$ 
        end if
    end for
until  $G = G'$ 
```

Bevis: Se [4]. □

Dette er den første av flere algoritmer for å beregne en Gröbner-basis. Senere i denne oppgaven skal vi se på noen av de andre algoritmene med dette formålet. Mange av forbedringene av Buchbergers algoritme dreier seg om hvordan man velger, og reduserer, de kritiske parene, ofte kalt utvelgelsesstrategi. Andre algoritmer vedrører de såkalte "kriteriene" for å unngå kritiske par som reduseres til null mhp. den foreløpige basisen G' . F4 er et velkjent eksempel på en algoritme med forbedret utvelgelsesstrategi, og denne algoritmen vil bli introdusert i kapittel 4. Gebauer og Möller og Faugère formulerte kriterier som også blir kommentert senere.

Algoritmer som beregner en Gröbner-basis, legger, som Buchbergers algoritme, reduserte S-polynomer til en mengde G' for å danne en foreløpig basis for det originale idealet I , samtidig som et større monomideal, utspent av de ledende leddene i idealet, lages. Heretter vil uttrykket "foreløpig basis" referere til en mengde som tilsvarer G' i Buchbergers algoritme.

Definisjon: Gitt en monomsortering og la $G = \{g_1, \dots, g_m\} \subset P$. Gitt et polynom $f \in P$, så sier vi at f reduseres til null modulo G , og skriver dette på følgende måte:

$$f \rightarrow_G 0,$$

hvis f kan skrives på formen

$$f = a_1g_1 + \dots + a_mg_m,$$

slik at når $a_i g_i \neq 0$, så har vi

$$\text{multideg}(f) \geq \text{multideg}(a_i g_i).$$

Definisjon: Normalformen r til f mhp. G er den refleksiv-transitive tillukningen av reduksjonen av elementer i G , \rightarrow_G . Dette betyr at hvis det er en reduksjonskjede fra f til r ved divisjon med elementer fra G , og r ikke kan reduseres videre av noen elementer fra G (mhp. en monomsortering), så kalles r for normalformen til f mhp. G .

Det nye kriteriet formuleres i teoremet som følger.

Teorem 1.8.2. La G være en endelig delmengde av P . G er en Gröbner-basis hvis og bare hvis normalformen til $S(g_1, g_2)$ er lik 0, eller $S(g_1, g_2) \rightarrow_G 0$, for alle $g_1, g_2 \in G$.

Bevis: Se [4]. □

I mange tilfeller vil Buchbergers algoritme gi en Gröbner-basis som er større enn nødvendig. Følgende resultat medfører at vi kan avgjøre om et element i en Gröbner-basis er overflødig.

Lemma 1.8.3. La G være en Gröbner-basis for idealet I i en polynomring. La $p \in G$ være et polynom slik at $LT(p) \in \langle LT(G - \{p\}) \rangle$. Da er $G - \{p\}$ også en Gröbner-basis for I .

Bevis: Vi vet at $\langle LT(G) \rangle = \langle LT(I) \rangle$. Hvis $LT(p) \in \langle LT(G - \{p\}) \rangle$, så har vi $\langle LT(G - \{p\}) \rangle = \langle LT(G) \rangle$. Da følger det av definisjonen at $G - \{p\}$ er en Gröbner-basis for I . □

Ved å tilpasse konstanter slik at alle ledende koeffisienter blir 1 og i tillegg fjerne enhver p med $LT(p) \in \langle LT(G - \{p\}) \rangle$ fra G , så kommer vi fram til det som kalles en minimal Gröbner-basis.

Definisjon: En minimal Gröbner-basis for et ideal I i en polynomring er en Gröbner-basis G for I slik at:

- (i) $LC(p) = 1$ for alle $p \in G$.
- (ii) $LT(p) \notin \langle LT(G - \{p\}) \rangle$ for alle $p \in G$.

Vi kan alltid finne en minimal Gröbner-basis, for et gitt ideal, ved å bruke Buchbergers algoritme og deretter anvende Lemma 1.8.3 for å eliminere de unødvendige generatorene som måtte ha kommet med. Dette kan vi illustrere ved å gå tilbake til idealet I vi så på i eksempelet i delkapittel 1.4. Ved bruk av grlex-sortering fant vi følgende Gröbner-basis:

$$\{f_1, f_2, f_3, f_4, f_5\} = \{x^3 - 2xy, x^2y - 2y^2 + x, -x^2, -2xy, -2y^2 + x\}.$$

Siden to av de ledende koeffisientene er ulik 1, må vi først multiplisere disse generatorene (f_4, f_5) med passende konstanter. Legg nå merke til at $LT(f_1) = x^3 = -x \cdot LT(f_3)$. Lemma 1.8.3 sier da at vi kan fjerne f_1 . Videre har vi $LT(f_2) = x^2y = -\frac{1}{2}x \cdot LT(f_4)$, så vi kan på samme måte fjerne f_2 . Det er ingen andre tilfeller hvor ledende ledd til en generator deler ledende ledd til en annen generator. Dermed er

$$\{\tilde{f}_3, \tilde{f}_4, \tilde{f}_5\} = \{x^2, xy, y^2 - \frac{1}{2}x\}$$

en minimal Gröbner-basis for I .

Denne minimale Gröbner-basisen til et ideal er ikke nødvendigvis unik. For idealet i eksempelet vårt er det lett å kontrollere at

$$\{\tilde{f}_3, \tilde{f}_4, \tilde{f}_5\} = \{x^2 + axy, xy, y^2 - \frac{1}{2}x\}$$

også er en minimal Gröbner-basis, hvor $a \in k$ er en konstant. Hvis vi antar at k er uendelig kan vi dermed produsere uendelig mange minimale Gröbner-basiser. Heldigvis er det mulig å finne en unik basis som er bedre enn de andre, en såkalt redusert Gröbner-basis. Denne definerer vi på følgende måte:

Definisjon: En redusert Gröbner-basis for et ideal I i en polynomring er en Gröbner-basis G for I slik at:

- (i) $\text{LC}(p) = 1$ for alle $p \in G$.
- (ii) For alle $p \in G$, så ligger ingen monom av p i $\langle \text{LT}(G - \{p\}) \rangle$.

I eksempelet ovenfor ser vi at den eneste reduserte Gröbner-basisen vil være nettopp den med $a = 0$. Til slutt tar vi med en fin egenskap som gjelder generelt for Gröbner-basiser.

Proposisjon 1.8.4. *La $I \neq \{0\}$ være et ideal i en polynomring. Da har I , gitt en monomsortering, en unik redusert Gröbner-basis.*

Bevis: Se [4]. □

1.9 Løsing av polynomiske ligningssystem

Dette avsnittet forklarer hvordan Gröbner-baser kan brukes til å løse polynomiske ligningssystem, og vil derfor ta for seg teori som ofte ikke nevnes i artikler om algebraiske angrep på kryptosystemer. Vi hevder at en Gröbner-basis med lex-sortering vil bringe det polynomiske ligningssystemet over på en “triangulær form”. Dette blir bekreftet i form-lemmaet, som kommer senere i avsnittet. For å bevise dette lemmaet, må vi introdusere noen fundamentale egenskaper som omhandler idealer.

Definisjon: Gitt $I = \langle f_1, \dots, f_m \rangle \in P$, så er det l-te eliminasjonsidealset I_l , idealset i $k[x_{l+1}, \dots, x_n]$ definert ved

$$I_l = I \cap k[x_{l+1}, \dots, x_n].$$

Definisjon: En kropp k kalles en perfekt kropp hvis enten kroppens karakteristikk er 0, eller karakteristikken er $p > 0$ og $k = k^p$, dvs. alle elementene har en p-te rot i k .

Legg merke til at endelige kropper $k = \mathbb{F}_q$, hvor $q = p^e$ og $e > 0$, er perfekte siden avbildningen $x \mapsto x^{p^{e-1}}$ har p-te røtter, fordi $(x^{p^{e-1}})^p = x$ for alle $x \in k$.

Det viser seg å være viktig om ligningssystemet knyttet til det kryptografiske problemet, har et endelig antall løsninger. Idealet som de korresponderende polynomene i et slikt ligningssystem utspenner, kalles nulldimensjonalt.

Proposisjon 1.9.1 (Endelighetskriteriet). *La $>$ være en sortering av monomene $T(P)$ i polynomringen $P = \bar{k}[x_1, \dots, x_n]$. For et ligningssystem som korresponderer med et ideal $I = \langle f_1, \dots, f_m \rangle$, er følgende utsagn ekvivalente:*

- (i) *Ligningssystemet har et endelig antall løsninger.*
- (ii) *For $i = 1, \dots, n$, har vi $I \cap \bar{k}[x_i] \neq \emptyset$.*
- (iii) *Mengden av monomer $T(P) \setminus \{LT_{>}(f) : f \in I\}$ er endelig.*
- (iv) *Dimensjonen til \bar{k} -vektorrommet P/I er endelig.*

Bevis: Se [12] for bevis. □

Dette kriteriet impliserer at for kryptografi bruk, så vil de såkalte “kroppslingene”

$$\{x_i^q - x_i = 0 : 1 \leq i \leq n\}$$

sikre at idealet er nulldimensjonalt.

For å kunne si noe om mulige polynomer i idealet beskrevet av en mengde polynomer, vil teoremet som følger være svært viktig. Det forteller at et polynom over en algebraisk lukket kropp som deler nullpunkter med polynomene i $F = \{f_1, \dots, f_m\}$, forekommer i en eller annen grad av idealet utspent av F .

Teorem 1.9.2 (Hilberts nullstellensatz). *La k være en algebraisk lukket kropp. Hvis $f, f_1, \dots, f_m \in P$ er slik at $f \in I(V(f_1, \dots, f_m))$, så eksisterer det et heltall $e \geq 1$ slik at*

$$f^e \in \langle f_1, \dots, f_m \rangle.$$

Tilsvarende gjelder implikasjonen den andre veien.

Bevis: Se [4]. □

Definisjon: La $I \subset P$ være et ideal. Radikalet til I , \sqrt{I} , er mengden

$$\{f : f^e \in I \text{ for et heltall } e \geq 1\}.$$

La \bar{k} være den algebraiske tillukningen av k . Anta at vi jobber med et kryptosystem over $k = \mathbb{F}_q$, hvor q er graden til et primtall p . Anta videre at $F = \{f_1, \dots, f_m\} \subset \bar{k}[x_1, \dots, x_n]$,

og at ligningene

$$\begin{aligned} y_1 &= f_1(x_1, \dots, x_n) \\ y_2 &= f_2(x_1, \dots, x_n) \\ &\vdots \\ y_m &= f_m(x_1, \dots, x_n) \end{aligned}$$

beskriver sammenhengene mellom komponentene man får ut, $y_1, \dots, y_m \in k$, og meldingskomponentene $x_1, \dots, x_n \in k$. Siden meldingsbitsene er elementer i k , er vi ikke interessert i mulige løsninger som måtte eksistere i $\overline{k} \setminus k$. Derfor legges mengden

$$\{x_i^q - x_i = 0 : 1 \leq i \leq n\}$$

til F , og den lager et radikal-ideal som er slik at meldingsbitsene fremdeles er mulige å løse. Dette blir behandlet i Seidenbergs lemma.

Proposisjon 1.9.3 (Seidenbergs lemma). *La k være en kropp, la $P = k[x_1, \dots, x_n]$ og la $I \subseteq P$ være et nulldimensjonalt ideal. Anta at for enhver $i \in \{1, \dots, n\}$, så eksisterer det et polynom forskjellig fra null, $g_i \in I \cap k[x_i]$, slik at største felles divisor til g_i og den deriverte til g_i er lik 1. Da er I et radikal-ideal.*

Bevis: Se [12]. □

Når man legger til kroppsligningene, så eksisterer g_i som er relativt primiske til sine deriverte, slik Seidenbergs lemma beskriver. Derfor er idealet I radikalt og, ifølge endelighet-skriteriet, nulldimensjonalt. Siden $x_i^q - x_i$ faktoriseres helt over k , så vil ikke den tilhørende varieteten, V , inneholde noen punkter $p \in V$ med koordinater i $\overline{k} \setminus k$.

Vi er nå klare for å presentere form-lemmaet.

Teorem 1.9.4 (Form-lemmaet). *La k være en perfekt kropp og la $I \subseteq P$ være et nulldimensjonalt radikal-ideal, slik at de x_n koordinatene av punkter i $V(I)$ er distinkte. La $g_n \in k[x_n]$ være den moniske generatoren av eliminasjonsidealset $I \cap k[x_n]$, og la $d = \text{grad}(g_n)$.*

(i) Den reduserte Gröbner-basisen til idealet I mhp. lex-sorteringen $x_1 > \dots > x_n$ er på formen

$$\{x_1 - g_1, \dots, x_{n-1} - g_{n-1}, g_n\},$$

hvor $g_1, \dots, g_n \in k[x_n]$.

(ii) Polynomet g_n har d distinkte røtter $a_1, \dots, a_d \in \overline{k}$, og mengden av røtter til I er

$$\{(g_1(a_i), \dots, g_{n-1}(a_i), a_i) : i = 1, \dots, d\}.$$

Bevis: Se [12].

□

Gröbner-baser har vist seg å være et av de viktigste verktøyene når det gjelder å løse algebraiske systemer. Som vi har sett tidligere er Gröbner-basisen til et system avhengig av hvilken sortering som brukes. De ulike sorteringene har ulike fordeler. Når det gjelder kompleksiteten til beregningen av Gröbner-basisen, har grevlex-sorteringen vist seg å være den mest effektive. Men som form-lemmaet forklarer, så er det lex-sorteringen som egner seg best når løsninger til algebraiske systemer skal beregnes.

Kapittel 2

Hidden Field Equations (HFE)

2.1 Introduksjon

Alle multivariate kvadratiske offentlige nøkkelsystem kan deles inn i fire typer: Unbalanced Oil and Vinegar Systems (UOV), Stepwise Triangular Systems (STS), Matsumoto-Imai Scheme A (MIA) og Hidden Field Equations (HFE). Vi skal her se på systemet Hidden Field Equations, forkortet HFE. MIA og HFE tar begge i bruk kroppsutvidelse mens UOV og STS ikke gjør det.

Etter å ha knekt systemet MIA, utviklet franskmannen Jacques Patarin et system som ble kalt Hidden Field Equations. Han tok utgangspunkt i MIA da han bygde opp HFE, og skriver i innledningen av sin artikkel [16] at en annen tittel på denne publikasjonen kunne vært “How to repair Matsumoto-Imai algorithm with the same kind of public polynomials”. Sikkerheten i HFE ligger i problemet med å løse et system av multivariate kvadratiske ligninger over en endelig kropp (f.eks. \mathbb{F}_2). Det generelle problemet å løse et tilfeldig valgt system av multivariate kvadratiske ligninger over \mathbb{F}_2 er NP-komplett [9] (men å få igjen en klartekst fra en kryptert HFE-tekst er ikke et NP-komplett problem, riktignok er dette problemet forventet å være eksponensielt vanskelig). Videre vil HFE, med noen godt valgte parametere, gi en algoritmekandidat for asymmetriske signaturer av 128 bits, og til og med 64 bits!

Som nevnt tidligere, så er ikke HFE det første forsøket på å bruke multivariate kvadratiske ligninger over \mathbb{F}_2 som et asymmetrisk kryptosystem. Japanerne Matsumoto og Imai hadde allerede laget en slik algoritme, som de kalte C^* [15], en algoritme som Patarin knakk i [17]. HFE ligner på mange områder på MIA, men har blitt spesialkonstruert for å motstå alle ideene ved angrepene på MIA.

2.2 Matematisk bakgrunn

Før vi ser på selve krypteringen i HFE, vil vi ta for oss funksjonen f som lager polynomene som brukes i krypteringen.

Funksjonen f :

La K være en endelig kropp med kardinalitet q og karakteristikk p . La L_N være en kroppsutvidelse av grad N av kroppen K , og la $B = \{b_1, \dots, b_N\}$ være en basis for L_N som vektorrom over K . La β_{ij}, α_i og μ_0 være elementer i L_N , og la $\theta_{ij}, \varphi_{ij}$ og ξ_i være heltall. Til slutt, la f være funksjonen:

$$f : \begin{cases} L_N \rightarrow L_N \\ x \mapsto \sum_{i,j} \beta_{ij} x^{q^{\theta_{ij}} + q^{\varphi_{ij}}} + \sum_i \alpha_i x^{q^{\xi_i}} + \mu_0. \end{cases}$$

Gitt et element $x = \sum_{i=1}^N x_i b_i$ i L_N , hvor $x_i \in K$, har vi:

$$f(x) = (p_1(x_1, \dots, x_N), \dots, p_N(x_1, \dots, x_N))$$

hvor p_1, \dots, p_N er N kvadratiske polynomer i N variable.

Årsaken til at polynomene blir kvadratiske er at for ethvert heltall λ , så er $x \mapsto x^{q^\lambda}$ en lineær funksjon $L_N \rightarrow L_N$. Polynomene p_1, \dots, p_N finnes typisk ved å velge et irreduksibelt polynom $i_N(X)$ over K , av grad N , og deretter identifisere L_N med $K[X]/(i_N(X))$.

2.3 Beskrivelse av krypteringen i HFE

Her skal vi se på den grunnleggende HFE-algoritmen for kryptering.

Representasjon x av meldingen M :

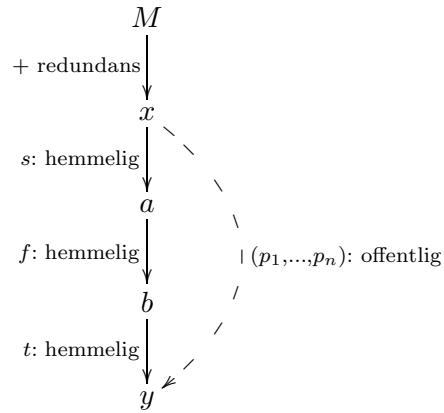
En kropp K med $q = p^m$ elementer er offentlig. Hver melding M er representert ved en verdi x som er en string av n elementer fra K . Dvs. at hvis $p = 2$, så vil hver melding være representert ved nm bits. Videre vil det enkelte ganger antas at redundans har blitt lagt inn i representasjonen av meldingene. Tilfellene hvor vi må legge til redundans er når meldingen er på $k < nm$ bits. Da må vi legge til $nm - k$ bits slik vi får en unik løsning når vi dekrypterer.

Kryptering av x :

I krypteringen av x er følgende hemmelig:

1. En utvidelse L_n av K av grad n .
2. En funksjon f fra L_n til L_n , med en grad d , “ikke for stor” (typisk er $d \leq 1024$).
3. To affine bijeksjoner s og $t: K^n \rightarrow K^n$. (Disse affine bijeksjonene kan bli representeret i en basis som polynomer av total grad én og med koefisienter i K).

Krypteringen er beskrevet i Figur 1 (som skal leses fra toppen og nedover). Den krypterte teksten y er gitt ved $y = t(f(s(x)))$.



Figur 1: Figuren viser standard HFE for kryptering.

En viktig ting å legge merke til er at siden s og t er av grad én, og siden f er kvadratisk, så vil sammensetningen av alle disse operasjonene fremdeles være en kvadratisk funksjon. Denne funksjonen kan bli gitt ved n polynomer med koeffisienter i K , (p_1, \dots, p_n) . Disse polynomene gir komponentene y_1, \dots, y_n fra den krypterte teksten y fra komponentene x_1, \dots, x_n av x :

$$y = \begin{cases} y_1 = p_1(x_1, \dots, x_n) \\ y_2 = p_2(x_1, \dots, x_n) \\ \vdots \\ y_n = p_n(x_1, \dots, x_n) \end{cases}$$

Dette er den offentlige nøkkelen:

1. Kroppen K med $q = p^m$ elementer, og lengden n .
2. De n polynomene p_1, \dots, p_n i n variable over K .
3. Algoritmen for å produsere redundans i meldingen.

Dette viser at vi ikke trenger noe hemmelig informasjon for å kryptere, altså kan hvem som helst kryptere en melding M .

Videre ser vi at dersom den hemmelige informasjonen er kjent, så vil dekryptering være lett, siden vi da bare trenger å invertere alle operasjonene i Figur 1. Når f skal inverteres, må vi løse et system av polynomligninger med én variabel i L_n . Men siden f ikke trenger å være en bijeksjon, kan vi finne mer enn en løsning. Vi kan finne opptil d løsninger, siden f er et polynom av grad d i en kropp. Dersom meldingen som sendes er på under nm bits, så vil ligningssystemet ha flere løsninger. For å være sikker på å finne korrekt melding M , vil redundans legges til meldingen slik at meldingen og redundansen tilsammen utgjør nm bits.

Redundans:

Redundans kan legges til både i klarteksten x og i den krypterte teksten y .

Hvis meldingen er k bits og vi legger til redundansen i klarteksten, så vil klarteksten som sendes være på $k + r = nm$ bits. En måte å legge til redundans på, er å ta i bruk en "feilrettingskode". Det er også mulig å bruke en hash-funksjon, som f.eks. MD5 eller SHA. Redundansen vil da typisk være de første r bits-ene i hash-funksjonen. Det er også mulig å bruke enklere funksjoner, men det er viktig at hvert bit som legges til M , er avhengig av bits-ene i M på en ikke-lineær måte.

Hvis redundansen legges til i y , vil $x = M$, mens den krypterte teksten vil være $y + \omega$, der ω er en enveis funksjon av x .

2.4 Eksempel på bruk av HFE

For at det hele ikke skal bli alt for komplisert og uoversiktlig, velger vi å se på den enkleste kroppen, nemlig $K = \mathbb{F}_2$. I dette eksempelet bruker vi det irreducibele polynomet $i_n(x) = x^3 + x^2 + 1$ av grad 3 over K . Derved vil kroppsutvidelsen L_n være en tredjegradsutvidelse av grunnkroppen K . Vi har $L_n = K[X]/(i_n(x))$, og ser at L_n består av restklasser av polynomer av grad maksimalt lik 2. Deretter ser vi på funksjonen f som lager "HFE-polynomet". Polynomet vil være på formen

$$f(x) = \sum_{i,j} \beta_{ij} x^{q^{\theta_{ij}} + q^{\varphi_{ij}}} + \sum_i \alpha_i x^{q^{\xi_i}} + \mu_0$$

hvor β_{ij}, α_i og μ_0 er elementer i L_n og $\theta_{ij}, \varphi_{ij}$ og ξ_i er heltall.

Vi ser at $f(x) = x + x^3 + x^5$ er et slikt polynom med $d = 5$. Polynomet kan også skrives

$$f(x_1 + x_2x + x_3x^2) = (x_1 + x_2x + x_3x^2) + (x_1 + x_2x + x_3x^2)^3 + (x_1 + x_2x + x_3x^2)^5$$

hvor $x_i = 0$ eller $x_i = 1$ for $i = 1, 2, 3$ (siden vi jobber over kroppen $K = \mathbb{F}_2$).

Nå må vi finne ut hva polynomet $f(x)$ blir i kroppsutvidelsen L_n . Vi husker fra definisjonen av f at

$$f(x_1 + x_2x + x_3x^2) = (p_1(x_1, \dots, x_n), \dots, p_n(x_1, \dots, x_n))$$

er uttrykket til f i basisen B til L_n .

For å finne disse p_i -ene må vi ta i bruk det irreducibele polynomet, $i_n(x)$, som vi har valgt oss. Vi observerer at $i_n(x) = x^3 + x^2 + 1$ gir $x^0 \mapsto 1$, $x \mapsto x$, $x^2 \mapsto x^2$, $x^3 \mapsto x^2 + 1$, $x^4 \mapsto x^2 + x + 1$, $x^5 \mapsto x + 1$, $x^6 \mapsto x^2 + x$, $x^7 \mapsto 1$, $x^8 \mapsto x$, $x^9 \mapsto x^2$, $x^{10} \mapsto x^2 + 1$ osv.

Dette bruker vi til å finne at polynomet kan skrives

$$f(x_1 + x_2x + x_3x^2) = (x_3 + x_1 + x_3x_1 + x_2x_1) + (x_3 + x_3x_2 + x_2x_1)x + (x_3 + x_2 + x_3x_2 + x_3x_2)x^2.$$

Da har vi at:

$$\begin{aligned} p_1(x_1, x_2, x_3) &= x_1 + x_3 + x_1x_2 + x_1x_3 \\ p_2(x_1, x_2, x_3) &= x_3 + x_1x_2 + x_2x_3 \\ p_3(x_1, x_2, x_3) &= x_2 + x_3 + x_1x_3 + x_2x_3 \end{aligned}$$

Meldingen vi vil sende kan, som nevnt tidligere, bestå av maksimalt nm bits. Her er $n = 3$ og $m = 1$, så meldingen vår kan ikke være lengre enn 3 bits.

Så velger vi de to affine bijeksjonene s og t . Vi definerer $s: x_i \mapsto x_{n-i}$ og $t: x_i \mapsto x_{i+1}$ for $i = 1, 2, 3$, hvor indeksene regnes "syklisk". I t vil dette f.eks. bety at $x_3 \mapsto x_1$.

For å finne de tre polynomene som skal være offentlige må vi gå gjennom algoritmen for en tilfeldig melding på 3 bits.

Vi kaller den tilfeldige meldingen (x_1, x_2, x_3) . Når vi anvender s får vi (x_3, x_2, x_1) . Dette tilsvarer $x_3 + x_2x + x_1x^2$ i kroppsutvidelsen L_n .

Så bruker vi funksjonen f til å skrive meldingen på formen $p_1(x_3, x_2, x_1) + p_2(x_3, x_2, x_1)x + p_3(x_3, x_2, x_1)x^2$. Dette uttrykket svarer til $(p_1(x_3, x_2, x_1), p_2(x_3, x_2, x_1), p_3(x_3, x_2, x_1))$ i K^n . Til slutt vil den affine bijeksjonen t gi oss $(p_3(x_3, x_2, x_1), p_1(x_3, x_2, x_1), p_2(x_3, x_2, x_1))$.

Dette gir oss følgende polynomer som gis ut offentlig:

$$\begin{aligned} p_1(x_3, x_2, x_1) &= x_1 + x_3 + x_1x_3 + x_2x_3 \\ p_2(x_3, x_2, x_1) &= x_1 + x_1x_2 + x_2x_3 \\ p_3(x_3, x_2, x_1) &= x_1 + x_2 + x_1x_2 + x_1x_3 \end{aligned}$$

Vi vil nå sende meldingen $(1\ 0\ 1)$.

Ved hjelp av de tre offentlige polynomene krypterer vi denne meldingen. Vi ser at $x_1 = x_3 = 1$ og $x_2 = 0$. Dermed blir $p_1(x_3, x_2, x_1) = 1$, $p_2(x_3, x_2, x_1) = 1$ og $p_3(x_3, x_2, x_1) = 0$. Siden rekkefølgen på polynomene er $(p_3\ p_1\ p_2)$, vil den krypterte meldingen være $(0\ 1\ 1)$.

Nå går vi baklengs gjennom algoritmen for å finne den sendte meldingen. Først vil den inverse affine bijeksjonen t^{-1} gi $(1\ 1\ 0)$. Dette er det samme som $1 + x$ i L_n .

Ved å anvende f^{-1} på $1 + x$ får vi ligningssystemet

$$\begin{aligned} p_1(x_1, x_2, x_3) &= 1 \\ p_2(x_1, x_2, x_3) &= 1 \\ p_3(x_1, x_2, x_3) &= 0 \end{aligned}$$

og finner at $x_1 = x_3 = 1$ og $x_2 = 0$. Da har vi $1 + x^2$ i L_n , som er det samme som $(1\ 0\ 1)$ i K^n . Ved å anvende s^{-1} skjer ingenting, og vi står igjen med den opprinnelige meldingen $(1\ 0\ 1)$.

Her har vi altså ikke lagt til redundans, siden meldingen vi ville sende var $nm = 3$ bits.

2.5 Fordeler og ulemper med HFE

Som alle kryptosystemer finnes det både fordeler og ulemper med HFE. Vi ser først på ulempene:

Den offentlige nøkkelen er relativt stor.

Operasjoner med de hemmelige nøklene er nokså trege. Det er ikke noe problem for en PC, men systemer med små ressurser, som f.eks. smartkort, vil ikke egne seg.

En åpenbar ulempe med HFE er at det er et relativt nytt kryptosystem. Det vil si at det ennå ikke er så mange som har forsøkt å knekke systemet. Her har f.eks. RSA en stor fordel fordi dette kryptosystemet, i motsetning til HFE, i lang tid har vært utsatt for angrepsforsøk uten at disse har lyktes.

HFE-systemet har følgende fordeler:

En stor fordel er at problemet med å løse et tilfeldig valgt system av multivariate kvadratiske ligninger over kroppen \mathbb{F}_2 er NP-komplett [9]. I praksis finnes det ikke noen kjent metode, for å løse systemet, som er vesentlig raskere enn å prøve seg fram.

HFE har, som nevnt i innledningen, blitt spesialkonstruert for å motstå angrepene som knakk MIA. HFE er altså mye sterkere enn MIA, og alle angrep som knekker HFE, vil også knekke MIA.

Ellers gir HFE mulighet for raske og korte signaturer. Med noen godt valgte parametere, vil HFE gi en algoritmekandidat for asymetriske signaturer av 128 bits, og til og med 64 bits. [16]

I 1994 viste Peter Shor at kvantedatamaskiner kan knekke alle signaturskjemaer som blir brukt i dag [19]. Så selv om RSA er sikkert i dag, så vil dette kryptosystemet bryte sammen hvis man tar i bruk kvantedatamaskiner. Foreløpig finnes det ikke noen kjent algoritme for å løse multivariate ligninger, som gir kvantedatamaskiner en fordel framfor vanlige datamaskiner. Det betyr at mens f.eks. RSA ikke er kvanteresistant, så kan HFE være det.

Kapittel 3

Kryptoanalyse av Hidden Field Equations (HFE) vha. Gröbner-baser

3.1 Introduksjon

Sikkerheten til HFE, særlig de ulike variantene som Patarin presenterte i [16], er ikke veldig godt undersøkt i dag. Noen av systemene har blitt knekt, mens andre ser ut til å stå imot angrepene som har blitt forsøkt. Men det er likevel vanskelig å trekke et klart skille mellom de sikre og usikre systemene i HFE-familien.

De kjente angrepene som har blitt forsøkt, kan vi dele inn i to klasser. Den ene klassen består av angrepene som fokuserer på en spesiell variant, og knekker akkurat denne varianten. Den andre klassen består av angrep som inneholder generelle algoritmer for å løse multivariate ligningssystem. I denne klassen har vi relineæreriseringsteknikken til Kipnis og Shamir [11], XL-algoritmen [3] og angrep vha. Gröbner-baser.

Kipnis og Shamirs relineæreriseringsteknikk passer veldig godt til det grunnleggende HFE-systemet, beskrevet i kapittel 2. Her skal vi imidlertid konsentrere oss om angrep som tar i bruk Gröbner-baser. Gröbner-baser er en veletablert og generell metode for å løse polynomiske ligningssystem. Hvor effektivt et angrep som bruker Gröbner-baser er, vil dermed være avhengig av hvor rask algoritmen som produserer Gröbner-basisen er. I [16] kommer Patarin med en utfordring, kjent som den første HFE-utfordringen, som det ikke er mulig å knekke hvis man bruker Buchbergers algoritme for å beregne Gröbner-basisen. Utfordringen med Gröbner-baseangrep på et HFE-system med gitte parametre, er å beregne kompleksiteten til angrepene.

Her skal vi vise at svakheten til ligningssystemet som kommer fra HFE-kryptosystemer, kan forklares ved de algebraiske egenskapene til den hemmelige nøkkelen. Dette har gjort det mulig å beregne den maksimale graden som oppstår når Gröbner-basisen beregnes. Det har også blitt gjort flere simuleringer på PC av HFE-problemer av realistisk størrelse (opp til 160 bits), slik at man kan beregne en nøyaktig kompleksitet av Gröbner-baseangrepene,

og sammenligne med de teoretiske grensene.

3.2 Første HFE-utfordring knekt

Den første HFE-utfordringen ble presentert av Patarin i [16], og en pemie på 500\$ ble lovet den som først kunne knekke systemet. Systemet som ble brukt i utfordingen bestod av 80 kvadratiske ligninger i 80 variable over \mathbb{F}_2 og med $d = 96$. Skulle man brukt Buchbergers algoritme, ville dette krevd $\geq 2^{80}$ operasjoner, så dette problemet er totalt uoppnåelig for den originale algoritmen.

Jean-Charles Faugère og Antoine Joux [8] ble i 2003 de første som klarte å løse utfordringen. De brukte en algoritme kalt F5, implementert i programmeringsspråket C. F5 tar i bruk avanserte Gröbner-baseteknikker, og algoritmen vil bli presentert nærmere i delkapittel 4.5. Det tok dem to dager og fire timer å knekke systemet med en datamaskin med 1 Ghz prosessor og 4 GB RAM. Kort forklart prøver F5 å utvide det oprinnelige ligningssystemet ved å velge ut ekstra ligninger og monomer, for deretter å omforme det utvidede systemet til matriseform. Det som tar mest tid i algoritmen er lineær algebraen som utføres.

Måten HFE-utfordringen ble løst på, involverte en 307126×1667009 -matrise som måtte løses over \mathbb{F}_2 . Det at matrisen ble løst, vil i denne sammenhengen si at man fant en Gröbner-basis for denne matrisen. Bare det å lagre en matrise på denne størrelsen, uten å komprimere den, krever 64 GB minne. Siden 80 kvadratiske ligninger tidligere hadde vært totalt utenfor rekkevidde, må denne måten å beregne en Gröbner-basis på, kunne regnes som et gjenombrudd når det gjelder å løse polynomiske ligningssystemer. I 2004 løste Allan Steel den første HFE-utfordringen ved å bruke Faugères F4-algoritme. Dette tok knapt 25 og en halv time med en 750 Mhz prosessor og 15 GB minne. Både F4- og F5-algoritmen skal vi se nærmere på i kapittel 4.

3.3 Eksempel på Gröbner-basisangrep på HFE

Det siste underkapittelet i kapittel 3 tar for seg et eksempel på et enkelt angrep på HFE samt litt om kompleksiteten til F4 og F5.

Anta at kroppen k er \mathbb{F}_q med $q = 2$. Vi velger $n = 5$ og $g(x) = x^5 + x^4 + x^3 + x + 1$ slik at \mathbb{F}_{q^n} er isomorf med $k[x]/\langle g(x) \rangle$. La θ være en rot av g . Da vil basisen $\{1, \theta, \theta^2, \theta^3, \theta^4\}$ bestemme sammenhengen mellom kroppsutvidelsen \mathbb{F}_{q^n} og vektorrommet k^n . Definer videre de affine transformasjonene s og t ved

$$s(x) = Ax + c \text{ og } t(x) = B^{-1}(x - d) \text{ for } x \in k^n,$$

hvor

$$A = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad c = (1, 0, 1, 1, 1), \quad d = (1, 0, 1, 0, 0).$$

Til slutt velges det hemmelige HFE-polynomet til å være $f(x) = x \cdot x^8 + x^4 \cdot x^{16}$ for elementer x i kroppsutvidelsen \mathbb{F}_{2^5} . Legg her merke til at leddene her er skrevet som faktorer av x med en eksponent $q = 2$ opphøyd i noe. For en klartekstvektor $x = (x_1, \dots, x_n)$, beregn den affine transformasjonen $s(x)$. Bruk basisen til å representere $s(x) \in k^5$ som et element i kroppsutvidelsen, og evaluér f i dette punktet. Så lages den offentlige nøkkelen ved å konvertere elementet tilbake til en vektor, og deretter anvende den affine transformasjonen t . Dette fører til følgende ligninger for den offentlige nøkkelen $y = (y_1, \dots, y_5) = (p_1(x), \dots, p_5(x))$:

$$\begin{aligned} y_1 &= x_1^2 + x_1x_2 + x_1x_3 + x_1 + x_2x_5 + x_3 + x_4^2 + x_5 + 1 \\ y_2 &= x_1x_2 + x_1x_3 + x_1x_4 + x_1x_5 + x_2x_3 + x_2 + x_3^2 + x_3x_5 + x_3 + x_4^2 + x_4 + x_5^2 + x_5 + 1 \\ y_3 &= x_1^2 + x_1x_4 + x_1x_5 + x_1 + x_2x_3 + x_2x_5 + x_2 + x_3^2 + x_3x_5 + x_4 + x_5^2 \\ y_4 &= x_1^2 + x_1x_2 + x_1x_4 + x_1 + x_2x_3 + x_2x_4 + x_2 + x_3x_5 + x_5^2 + 1 \\ y_5 &= x_1^2 + x_2^2 + x_2x_3 + x_2 + x_3x_4 + x_3x_5 + x_5 \end{aligned}$$

Anta nå at Alice vil kryptere klarteksten $(1, 1, 1, 1, 1) \in k^5$. Det hun da må gjøre er å evaluere ligningene for den offentlige nøkkelen i dette punktet. Dette gir henne den krypterte meldingen $(1, 0, 1, 0, 0)$ som hun kan sende til Bob. For dette lille eksemplet er det algebraiske anget til Eve enkelt, men det tar i bruk de tekniske detaljene som ble diskutert i underkapittel 1.9. Alt Eve trenger å gjøre, er å løse ligningssystemet for den kjente (y_1, \dots, y_5) . Hvis hun regner ut en redusert Gröbner-basis G' mhp. lex-sortering av $F = \{p_i - y_i : 1 \leq i \leq 5\}$, så vil ikke denne være nulldimensjonal, og dermed tilfredsstiller den ikke form-lemmaet. Ved å legge "kroppsligningene" $\{x_i^2 - x_i : 1 \leq i \leq 5\}$ til i mengden F , vil hun lage et nulldimensjonalt radikal-ideal. Den reduserte leksikografiske Gröbner-basisen, G , for dette idealet er

$$G = \{g_1, g_2, g_3, g_4, g_5\} = \{x_1 + x_5, x_2 + 1, x_3 + x_5, x_4 + x_5, x_5^2 + x_5\}$$

og løsningene $(0, 1, 0, 0, 0), (1, 1, 1, 1, 1)$ i k^5 følger. Her ser vi at redundans er nødvendig for å bestemme hvilken av løsningene som er den korrekte meldingen.

I 2004 kom Allan Steel med en veldig effektiv implementasjon av F4 i programmet Magma 2.11. I sin masteroppgave [18] testet A.J.M. Segers ut Steels implementasjon ved å kjøre bergninger på HFE-ligninger for nøkkellengder på mellom 10 og 30 bits. Det hemmelige HFE-polynomet ble valgt tilfeldig med maksimal grad 64 (gjennomsnittgraden var 40) og en Gröbner-basis ble beregnet. Tjue beregninger ble kjørt per nøkkellengde. Disse

simuleringene viste at å finne en Gröbner-basis vha. F4, har eksponensiell kompleksitet mhp. nøkkellengden. Allan Steel forklarer på sin hjemmeside [20] at hans forbedrede F4-implementasjon i Magma 2.11 er sammenlignbar med implementasjonen som beskrives i [5]. Man kan således si at denne siste Magma-versjonen inneholder en av de raskeste Gröbner-basisimplementasjonene som er kjent. Et viktig punkt når det gjelder kompleksiteten til HFE blir tatt opp i artikkelen [8]. Her viser Faugère og Joux at med deres F5-algoritme, så vil graden til polynomenene som dukker opp underveis i Gröbner-basisberegningen, holdes nede på et minimum. Denne påstanden blir støttet av angrepet de utførte da de løste Patarins HFE-utfordring. Under kjøringen av F5-algoritmen dukket det ikke opp polynomer av grad høyere enn 4, selv om graden til det hemmelige HFE-polynommet var 96. I sin artikkel beregner Faugère og Joux den maksimale graden til polynomene underveis i F5 brukt i forbindelse med HFE. Gjennom simuleringer kom de fram til resultatene som er gjengitt i tabellen under.

d	$3 \leq d \leq 12$	16	$17 \leq d \leq 96$	128	$129 \leq d \leq 512$	$513 \leq d \leq 1280$
d_{F5}	3	3	4	4	5	5

Tabell 3.1: Maksimal grad av mellomregningspolynomene under utførelsen av F5, d_{F5} , for det hemmelige HFE-polynomet, d .

Kapittel 4

Avanserte Gröbner-baseteknikker

4.1 Introduksjon

I dette kapittelet skal vi se på hvordan algoritmer for å finne Gröbner-baser er koblet til gausseliminasjon. Dette blir gjort ved å ta i bruk teorien som presenteres i underkapittel 4.2. I delkapittel 4.3 diskuteres den normale utvelgelsesstrategien i forbindelse med en algoritme kalt den ensartede Buchberger-algoritmen. Faugères F4-algoritme, en svært sentral algoritme når det gjelder å beregne Gröbner-baser, presenteres i 4.4. I det siste delkapittelet ser vi på to forbedrede utvelgelsesstrategier - Gebauer og Möller-installasjonen og strategien knyttet til F5-algoritmen til Faugère.

4.2 Kobling av Gröbner-baser og lineær algebra

Sammenhengen mellom lineær algebra og det å finne en Gröbner-basis ble introdusert av Daniel Lazard i [14]. For å forstå likheten, kan det være greit å begynne med å se på det enkleste tilfellet av to polynomer i én variabel:

$$\begin{aligned} f &= a_0 + a_1x + \cdots + a_lx^l, \quad a_l \neq 0, \\ g &= b_0 + b_1x + \cdots + b_mx^m, \quad b_l \neq 0. \end{aligned}$$

I dette tilfellet er standardmetoden for å beregne en Gröbner-basis det samme som Euklids algoritme for å beregne største felles divisor og resultanten av polynomer. Den andre måten å beregne resultanten på, er å se på determinanten til sylvestermatrisen, som blir definert her.

Definisjon: Den følgende $(m + l) \times (m + l)$ -matrisen, kalles sylvestermatrisen til f og

$g.$

$$\left(\begin{array}{cccccc} a_l & & b_m & & & \\ a_{l-1} & a_l & & b_{m-1} & b_m & \\ a_{l-2} & a_{l-1} & \ddots & b_{m-2} & b_{m-1} & \ddots \\ & & \ddots & a_l & & \ddots & b_m \\ \vdots & & & a_{l-1} & \vdots & & b_{m-1} \\ & & & \vdots & & \vdots & \\ a_0 & & & b_0 & & & \\ a_0 & & & b_0 & & & \vdots \\ & & & a_0 & & & b_0 \\ & & & & & & \end{array} \right)$$

Matrisen har m kolonner med koeffisientene a_i fra f og l kolonner med koeffisientene b_i fra g . Kolonnene i sylvestermatrisen kan ses på som representasjoner av polynomene $x^i f(x)$ for $i \in \{0, \dots, m-1\}$ og $x^j g(x)$ for $j \in \{0, \dots, l-1\}$.

Gausseliminasjonen starter ved at man b_m/a_l ganger trekker den første kolonnen fra kolonne $m+1$. Hvis $m \geq l$, så kan vi b_m/a_l ganger trekke den første i -te kolonnen fra kolonne $m+i$, for $i \in \{1, \dots, l\}$. Denne operasjonen kan ses på som å bytte ut koeffisientene i g med $h = g - (b_m/a_l)x^{m-l}f$. I dette tilfellet er sylvesterdeterminanten til f og g redusert til a_l ganger sylvesterdeterminanten til f og h . Hvis vi fortsetter denne prosessen, simulerer vi Euklids algoritme. Samtidig kan Euklids algoritme ses på som en måte å gjøre reduseringen av sylvestermatrisen mer effektiv.

La nå P være polynomringen $k[x_1, \dots, x_n]$ over en kropp k og $I = \langle f_1, \dots, f_m \rangle$ et ideal generert av polynomene $f_i \in P$, $0 \leq i \leq m$. Som et k -vektorrom, er I generert av

$$x^\alpha \text{ for alle } i \in \{1, \dots, m\} \text{ og } \alpha \in \mathbb{Z}_{\geq 0}^n.$$

En egenskap ved Gröbner-baser er at de sørger for en endelig beskrivelse av den lineære basisen til I , noe som blir beskrevet i den påfølgende proposisjonen. Det viser seg at det er nyttig å beregne dimensjonen til vektorrommet av elementer i idealet, opp til en gitt grad.

Proposisjon 4.2.1. *Mengden $F = \{f_1, \dots, f_m\} \subset P = k[x_1, \dots, x_n]$ er en Gröbner-basis for idealet $I = \langle F \rangle$ hvis og bare hvis den følgende mengden, B , er en basis for vektorrommet tilhørende I ,*

$$B = \{x^\alpha f_i : i \in \{1, \dots, m\}, \alpha \in \mathbb{Z}_{\geq 0}^n \text{ og samtidig ikke } LM(f_j) | LM(x^\alpha f_i) \text{ for } i < j\}.$$

Bevis: Se [14]. □

4.3 Ensartet Buchberger-algoritme

Den ensartede Buchberger-algoritmen danner et rammeverk for mange avanserte algoritmer som beregner Gröbner-baser. Denne algoritmen gir oss også mulighet til å beregne Gröbner-

baser for deler av idealer opp til elementer av en viss grad. Før vi ser på selve algoritmen, må et par definisjoner og en proposisjon på plass.

Definisjon: Et polynom f i $P = k[x_1, \dots, x_n]$ kalles ensartet av total grad d hvis hvert ledd som opptrer i f har total grad d . For en generell $g \in P$, så er den ensartede komponenten av grad d av g , lik summen av leddene som har total grad d .

Definisjon: Vi sier at I er et ensartet ideal hvis det for hver $f \in I$ er slik at de ensartede komponentene også er i I .

Proposisjon 4.3.1. La P være en polynomring og $I \subset P$ et ideal generert av en mengde av ensartede polynomer $F = \{f_1, \dots, f_m\}$. Da gjelder følgende:

- (i) Buchbergers algoritme, anvendt på F , returnerer en ensartet Gröbner-basis til I .
- (ii) Den reduserte Gröbner-basisen til I består av ensartede vektorer.

Bevis: Se [13] for bevis. □

Nå kan vi formulere den ensartede Buchberger-algoritmen som et teorem.

Teorem 4.3.2 (Ensartet Buchberger-algoritme). La $\{f_1, \dots, f_m\}$ være en mengde av ensartede polynomer som utspenner idealet I . Da kan en Gröbner-basis for I konstrueres vha. et endelig antall steg i algoritmen som følger.

Input: $F = \{f_1, \dots, f_m\}$

Output: Et tuppel $G = (g_1, \dots, g_{s'})$, hvor elementene tilfredsstiller

$$\text{totalgrad}(g_1) \leq \text{totalgrad}(g_2) \leq \dots \leq \text{totalgrad}(g_{s'})$$

og danner en Gröbner-basis til idealet I

$B := \{\}$

$G := \{\}$

$s' := 0$

repeat

$d_1 := \min\{\text{totalgrad}(f) : f \in F\}$

$d_2 := \min\{\text{totalgrad}(\text{LCM}(\text{LT}(g_i), \text{LT}(g_j))) : (i, j) \in B\}$

$d := \min\{d_1, d_2\}$

$B_d := \{(i, j) \in B : \text{totalgrad}(\text{LCM}(\text{LT}(g_i), \text{LT}(g_j))) = d\}$

$B := B \setminus B_d$

$F_d := \{f \in F : \text{totalgrad}(f) = d\}$

$F := F \setminus F_d$

repeat

if $B_d = \{\}$ **then**

Velg $f \in F_d$ og fjern denne fra F_d

else

```

Velg et par  $(i, j) \in B_d$  og fjern dette fra  $B_d$ 
 $f := S(g_i, g_j)$ 
if  $\bar{f}^G = 0$  then
    Gå tilbake tilbake til if  $B_d = \{\}$ 
else
     $s' := s' + 1$ 
     $g_{s'} := \bar{f}^G$  og legg  $g_{s'}$  til  $G$ 
    Legg par  $(1, s'), \dots, (s' - 1, s')$  til  $B$ 
end if
end if
until  $B_d = \{\}$  eller  $F_d = \{\}$ 
until  $B = \{\}$  eller  $F = \{\}$ 

```

Bevis: Se [13]. □

La $I_{\leq d}$ være elementene i et ideal I av grad mindre eller lik d . Den ensartede Buchberger-algoritmen er i stand til å beregne en basis med de samme egenskapene som en Gröbner-basis begrenset til elementene i I av grad $\leq d$.

Definisjon: La G være resultatet av den ensartede Buchberger-algoritmen for et ideal I . Elementene i G av total grad $\leq d$ danner mengden $G_{\leq d}$, som fra nå av kalles en d -trunkert Gröbner-basis for idealet I .

Dersom vi ikke er i et ensartet tilfelle, kan en d -trunkert Gröbner-basis defineres som følger.

Definisjon: La G være en endelig undermengde av idealet $I \subset P$. G kalles en d -trunkert Gröbner-basis for idealet I hvis det er slik at for alle $f, g \in G$ med total grad av de tilhørende kritiske parene mindre eller lik d , så vil S-polynomet $S(f, g)$ reduseres til null av G .

Hvis den ensartede Buchberger-algoritmen avbrytes etter at en grad d , så danner elementene i G en trunkert Gröbner-basis $G_{\leq d}$. Hovedegenskapen til en slik basis er formulert i proposisjonen under.

Proposisjon 4.3.3. *La G være en mengde av ensartede polynomer, forskjellig fra null, som genererer idealet $I \subset P$, og la $d \in \mathbb{Z}_{\geq 0}$. Da er følgende ekvivalent:*

- (i) $G_{\leq d}$ er en trunkert Gröbner-basis for I .
- (ii) Alle ensartede elementer $f \in I_{\leq d}$ tilfredsstiller

$$\exists g \in G_{\leq d} : LT(g) | LT(f)$$

Bevis: Se [13] for bevis. □

Det å velge ut kritiske par etter økende grad, kalles den normale utvelgelsesstrategien. Denne strategien er, i henhold til det kommende teoremet, med på å redusere beregningene under reduksjonen av S-polynomene.

Teorem 4.3.4. *La I være et ideal i P og $F = \{f_1, \dots, f_m\}$ en ensartet basis for I . La en Gröbner-basis for I bli beregnet fra F vha. en algoritme som behandler kritiske par etter stigende grad (som f.eks. den normale utvelgelsesstrategien). Hvis et S-polynom, underveis i algoritmen, enten har grad mindre enn, eller blir redusert til en grad mindre enn det tilhørende kritiske paret, så vil det etter hvert reduseres til 0.*

- (i) *Når graden minskes i en reduksjonsløkke, kan reduksjonen avbrytes siden den etter hvert vil reduseres til 0.*
- (ii) *Nye basiselementer dukker opp etter økende grad.*

Bevis: Se [21]. □

Den ensartede Buchberger-algoritmen kan, som nevnt tidligere, brukes som et rammeverk for andre forbedringer av Buchbergers algoritme. Faugère anbefaler å implementere sin F4-algoritme med den normale utvelgelsesstrategien. Neste delkapittel tar for seg algoritmen med denne utvelgelsesstrategien.

4.4 Faugères F4-algoritme

Algoritmen F5, som Faugère introduserer i [6] er basert på et effektivt utvelgelseskriterium for kritiske par, kombinert med en reduksjonsstrategi basert på lineær algebra. Denne reduksjonsstrategien ble presentert av Faugère i en tidligere algoritme, kalt F4 [5]. I dette underkapittelet forklares hovedideene bak F4 i detalj.

Vanligvis vil en reduksjonsstrategi inneholde to deler. Det man først og fremst trenger, er en metode for å bestemme hvilke kritiske par som skal reduseres. I Buchbergers algoritme vil dette valget bestemme hvilke reduserte S-polynomer som skal legges til i den foreløpige basisen, noe som igjen har innvirkning på de senere reduseringene i algoritmen. På samme måte som i divisjonsalgoritmen, kan man velge rekkefølgen på elementene i den midlertidige basisen som redusererer de nye S-polynomene.

I stedet for å beregne ett og ett nytt redusert S-polynom om gangen, er hovedtanke bak F4 å redusere en mengde kritiske par samtidig mhp. en forprosessert mengde av den foreløpige basisen, kalt reduktorer. Teorien bak beregningen av reduktorene stammer opprinnelig fra en del av FGLM-algoritmen [7].

En sammenheng mellom mengden av polynomer og matriser er beskrevet i definisjonen som følger.

Definisjon: La $F = (f_1, \dots, f_m)$ være et tuppel av polynomer i $P = k[x_1, \dots, x_n]$ og la $T(F)$ være mengden av parvis distinkte monomer i F og $T_\sigma(F)$ være den samme mengden, men mhp. sorteringen σ . Antallet distinkte monomer i F , $|T(F)|$, kaller vi s .

La et generelt polynom $f \in P$ skrives som

$$f = \sum_{i=1}^s c_i x^{\alpha_i}, \text{ med } \alpha_i \in \mathbb{Z}_{\geq 0}^n \text{ og } c_i \in k.$$

Definer vektorrepresentasjons-avbildningen

$$\psi_{T_\sigma(F)} : P \rightarrow k^s$$

av f mhp. $T_\sigma(F)$ som følger:

$$\psi_{T_\sigma(F)}(f) = (c_1, \dots, c_s)$$

og matriserepresentasjonen av et tuppel av polynomer F som

$$\psi_{T_\sigma(F)} : P^m \rightarrow \text{Mat}_{m,s}(k), (f_1, \dots, f_m) \mapsto \begin{pmatrix} \psi_{T_\sigma(F)}(f_1) \\ \vdots \\ \psi_{T_\sigma(F)}(f_m) \end{pmatrix}.$$

Dersom det går klart fram av sammenhengen er det vanlig å sløyfe subskriptet.

I F4-algoritmen blir de reduserte S-polynomene vi kjener fra Buchbergers algoritme, erstattet med matrisen \tilde{F}^+ , som defineres på følgende måte:

Definisjon: La F være en undermengde av polynomringen P .

- (i) Mengden av polynomer som korresponderer til trappeformen til $\psi(F)$ kalles \tilde{F} .
- (ii) La \tilde{F}^+ betegne $\{g \in \tilde{F} : \text{LT}(g) \notin \text{LT}(F)\}$.

Elementene i \tilde{F}^+ er knyttet til en undermengde H , av den opprinnelige mengden F , slik at

$$\text{LT}(H) = \text{LT}(F) \text{ og } |H| = |\text{LT}(F)|$$

holder. Som en konsekvens av det påfølgende teoremet er idealet $\langle F \rangle$ utspent av $H \cup \tilde{F}^+$.

Teorem 4.4.1. La k være en kropp, F en endelig mengde av elementer $P = k[x_1, \dots, x_n]$ og la s betegne kardinaliteten av $T_\sigma(F)$. For enhver undermengde $H \subseteq F$ slik at $|H| = |\text{LT}(F)|$ og $\text{LT}(H) = \text{LT}(F)$, så danner vektorene

$$\psi(g) \in k^s, \text{ for } g \in \tilde{F}^+ \cup H$$

en triangulær basis av underrommet til vektorrommet k^s utspent av vektorene $\psi(f)$ for $f \in F$.

Bevis: Sett $G = \tilde{F}^+ \cup H$. Alle elementene $g \in G$ har distinkte ledende ledd og er lineærkombinasjoner av elementer i F . Dermed er mengden $\{\psi(g) : g \in G\}$ lineært uavhengig og inkludert i undermengden utspent av vektorer som korresponderer med elementer i F . Videre, la f betegne rangen til underrommet utspent av $\psi(f)$ for $f \in F$. I tillegg gjelder

$$\text{LT}(G) = \text{LT}(\tilde{F}^+) \cup \text{LT}(H) = \text{LT}(\tilde{F}),$$

noe som impliserer $|\text{LT}(G)| = |\text{LT}(\tilde{F})| = r$ og teoremet følger. \square

Som nevnt tidligere ligger hovedideen bak F4 i hvordan S-polynomene reduseres. I stedet for å gjøre reduseringen for ett og ett S-polynom, så lager algoritmen utvalg av kritiske par $b = (b_1, b_2)$, for b_1, b_2 , i den foreløpige basisen G' og sender de to polynomene

$$\frac{\text{LCM}(\text{LT}(b_1), \text{LT}(b_2))}{\text{LT}(b_1)} b_1, \frac{\text{LCM}(\text{LT}(b_1), \text{LT}(b_2))}{\text{LT}(b_2)} b_2$$

til reduseringsfunksjonen. La oss anta at den vanlige utvelgesesstrategien er tatt i bruk. De kritiske parene som korresponderer til grad d er da

$$B_d = \{(b_1, b_2) : b_1, b_2 \in G' \text{ hvor } \text{totalgrad}(\text{LCM}(\text{LT}(b_1), \text{LT}(b_2))) = d, b_1 \neq b_2\}.$$

Dermed blir følgende mengde sendt til reduseringsrutinen i F4:

$$L_d = \bigcup_{(b_1, b_2) \in B_d} \left\{ \frac{\text{LCM}(\text{LT}(b_1), \text{LT}(b_2))}{\text{LT}(b_1)} b_1, \frac{\text{LCM}(\text{LT}(b_1), \text{LT}(b_2))}{\text{LT}(b_2)} b_2 \right\}.$$

Reduseringen i F4 tar i bruk forprosesserte reduktorer av en foreløpig basis G' . Reduktorene legges sammen i en rutine kalt symbolsk forprosessering.

Definisjon: Underveis i gjennomføringen av en algoritme som beregner Gröbner-baser, er en reduktor r av mengden F et polynom som tilfredsstiller

$$\text{LT}(r) \in T(F) \setminus \text{LT}(F).$$

Definisjon (Symbolisk forprosessering): Den følgende algoritmen føyer reduktorer til mengden F mhp. en foreløpig basis G' .

```

Input: En mengde  $F \subset P$  og en foreløpig basis,  $G'$ 
Output: Mengden  $F \cup R$  for en mengde av reduktorer,  $R$ 
 $D := \text{LT}(F)$ 
 $R := \{\}$ 
while  $(F \cup R) \neq D$  do
    Velg  $m \in T(F \cup R) \setminus D$ 
     $D := D \cup \{m\}$ 
    if  $\text{LT}(m)$  er delelig med et element  $g \in \text{LT}(G')$  then

```

```

 $m' := m/\text{LT}(g)$ 
 $R := R \cup \{gm'\}$ 
end if
end while

```

Nå kan vi formulere funksjonen F4 som reduserer polynomer som korresponderer til kritiske par.

Definisjon (ReduseringF4): Delrutinen ReduseringF4 returnerer \tilde{F}^+ , hvor F er “outputen” fra delrutinen symbolsk forprosessering for en mengde L_d , som definert tidligere, mhp. en foreløpig basis G' .

S-polynomer som ikke blir redusert til null i Buchbergers algoritme, utvider idealet utsspent av de ledende leddene i den foreløpige basisen. På denne måten oppdages en stigende kjede av ledende ledd-ideal. På samme måte bidrar de ledende leddene til elementer i \tilde{F}^+ til idealet utsspent av de ledende leddene i den foreløpige basisen. Dette blir klargjort i det følgende lemmaet.

Lemma 4.4.2. La \tilde{F}^+ betegne “outputen” av ReduseringF4 brukt på L_d mhp. G' . For alle $f \in \tilde{F}^+$, er $\text{LT}(f)$ ikke et element i $\langle \text{LT}(G') \rangle$.

Bevis: La $f \in \tilde{F}^+$ og la “outputen” av den symbolske forprosesseringen av L_d mhp. G' , betegnes ved F . For å tvinge fram en motsigelse antar vi at $\text{LT}(f) \in \langle \text{LT}(G') \rangle$. Denne antagelsen sammen med $\text{LT}(f) \in T(\tilde{F}^+) \subset T(F)$ impliserer at den symbolske forprosesseringen må ha lagt til en reduktør $\frac{\text{LT}(f)}{\text{LT}(g)}g$ til F , for en passende $g \in G'$. Dette ville bety at $\text{LT}(f) \in \text{LT}(F)$, noe som er en motsigelse til definisjonen av \tilde{F}^+ . Dermed er ikke $\text{LT}(f)$ et element i $\langle \text{LT}(G') \rangle$. \square

Det neste lemmaet forsikrer at elementer som legges til den foreløpige basisen, er med i idealet $\langle G' \rangle$.

Lemma 4.4.3. La \tilde{F}^+ være som i Lemma 4.4.2. Da er $\tilde{F}^+ \subset \langle G' \rangle$.

Bevis: Enhver $f \in \tilde{F}^+$ er en lineærkombinasjon av elementer fra L_d og reduktorer, R , som begge er undermengder av $\langle G' \rangle$. \square

Neste lemma erklærer at alle S-polynomer i mengden av mulige k -lineære kombinasjoner av L_d reduseres til null av en undermengde av $\tilde{F}^+ \cup G'$. Dette blir brukt til å bevise algoritmens korrekthet ved kriteriet formulert i Teorem 1.8.2.

Lemma 4.4.4. La \tilde{F}^+ være som i Lemma 4.4.2. For alle k -lineære kombinasjoner, f , av elementer fra L_d , så vil normalformen være lik null mhp. $\tilde{F}^+ \cup G'$.

Bevis: La f være en lineærkombinasjon av elementer fra L_d . Anta at F er “outputen” fra den symbolske forprosesseringen av L_d mhp. G' . Fra konstruksjonen av L_d har vi at denne mengden er en undermengde av F og ifølge Teorem 4.4.1 er derfor disse elementene

en lineærkombinasjon av den triangulære basisen $\tilde{F}^+ \cup H$ for en passende mengde $H \subset F$. Elementer fra H er enten elementer fra L_d eller på formen $x^\alpha g$ for en $g \in G'$ og en $\alpha \in \mathbb{Z}_{\geq 0}^n$, og f kan dermed skrives som

$$f = \sum_i a_i f_i + \sum_j a_j x^{\alpha_j} g_j,$$

for $f_i \in \tilde{F}^+$, $g_j \in G'$, $a_i, a_j \in k$ og $\alpha_j \in \mathbb{Z}_{\geq 0}^n$. Dermed gir divisjonsalgoritmen en rest på null for et passende tuppel av elementer i $\tilde{F}^+ \cup G'$, og det eksisterer dermed en reduseringsfølge til null. \square

Nå er vi klare for å beskrive F4-algoritmen for deretter å bevise at den stemmer. Utvelgesstrategien i algoritmen er ikke fastsatt her (strategien er bare betegnet med "Velg(B)"). Man kan velge ut alle de kritiske parene som er ledige, eller f.eks. bruke den normale utvelgesstrategien fra seksjon 4.3.

Teorem 4.4.5 (F4). *Algoritmen F4 beregner, i løpet av et endelig antall steg, en Gröbner-basis G av et ideal utspenn av F, slik at $F \subseteq G$.*

```

Input:  $F = \{f_1, \dots, f_m\}$ 
Output: En Gröbner-basis  $G$  for  $\langle F \rangle$ , som tilfredsstiller  $F \subseteq G$ .
 $G' := F$ 
 $\tilde{F}_0^+ := F$ 
 $d := 0$ 
 $B = \{(b_1, b_2) : b_1, b_2 \in G' \text{ med } b_1 \neq b_2\}$ 
while  $B \neq \emptyset$  do
     $d := d + 1$ 
     $B_d := \text{Velg}(B)$ 
     $B := B \setminus B_d$ 
     $L_d = \bigcup_{(b_1, b_2) \in B_d} \left\{ \frac{\text{LCM}(LT(b_1), LT(b_2))}{LT(b_1)} b_1, \frac{\text{LCM}(LT(b_1), LT(b_2))}{LT(b_2)} b_2 \right\}$ 
     $\tilde{F}^+ := \text{ReduseringF4}(L_d, G')$ 
    for  $f \in \tilde{F}^+$  do
         $B := B \cup \{(f, g) : g \in G'\}$ 
         $G' := G' \cup \{f\}$ 
    end for
end while
 $G := G'$ 

```

Bevis: Korrekthet og terminering av algoritmen blir bevist ved å observere følgende:

- (i) Lemma 4.4.3 impliserer at under steget $d = d'$ i algoritmen, så vil den foreløpige basisen tilfredsstille

$$G' = \bigcup_{d=1}^{d'} \tilde{F}_d^+ \subset \langle F \rangle.$$

(ii) Lemma 4.4.2 viser at

$$\langle \text{LT}(\tilde{F}_1^+) \rangle \subset \langle \text{LT}(\tilde{F}_1^+ \cup \tilde{F}_1^+) \rangle \subset \dots$$

er en stigende kjede av monomidealer. Stigende kjede-betingelsen, Teorem 1.6.4, sier den skal stabilisere seg etter hvert. Dette betyr at while-loopen må terminere siden vi til slutt vil gå tom for kritiske par.

- (iii) Anta at algoritmen terminerer ved $d = d_{F4}$. Siden ethvert par (g_1, g_2) for $g_1, g_2 \in G = \cup_{d=1}^{d_{F4}} \tilde{F}_d^+$ blir betraktet, så er $S(g_1, g_2)$ i det lineære spennet av elementer fra G . Lemma 4.4.4 sier at normalformen er lik null, og dermed er Gröbner-basiskriteriet i Teorem 1.8.2 tilfredsstilt.

□

Et eksempel på bruk av algoritmen F4 finnes i [5].

Faugère foreslår å implementere F4 med et sterkt kriterium for å unngå ubrukelige kritiske par. Gebauer og Möller formulerete i [10] et slikt kriterium. Dette kriteriet kan brukes i enhver utvidelse av Buchbergers algoritme, som iterativt velger ut kritiske par og beregner de tilhørende reduserte S-polynomene. Det blir også anbefalt å lagre matrisene i en komprimert form i tillegg til å bruke skreddersydde teknikker når det gjelder radredusering av sparse-matriser. I neste delkapittel vil den såkalte Gebauer og Möller-installasjonen forklares nærmere.

4.5 Gebauer og Möller-installasjon og F5

Gebauer og Möller-installasjonen presenteres for første gang i [10] og er en forbedring av to kriterier som Buchberger selv la fram i [1]. I [2] viste Massimo Caboara, Martin Kreuzer og Lorenzo Robbiano hvordan man kan beregne en minimal mengde av kritiske par. Konklusjonen deres var at Gebauer og Möller-kriteriet var nesten optimalt. Dette underkapittelet vil presentere Gebauer og Möller-installasjonen samt kriteriet som brukes i F5 for å unngå ubrukelige kritiske par.

For å kunne jobbe med Gebauer og Möller-installasjonen trenger vi noe som kalles syzygyer, og for å kunne definere syzygyer må begrepet modul introduseres. Modulen til en ring kan sammenlignes med vektorrommet til en kropp, og er definert slik.

Definisjon: For en ring R , er en R -modul en abelsk gruppe $(M, +)$ med en operasjon $\cdot : R \times M \rightarrow M$, kalt skalarmultiplikasjon, slik at $1 \cdot m = m$ for alle $m \in M$ og slik at både den assosiative og distributive lov gjelder. En abelsk undergruppe $N \subseteq M$ kalles en R -undermodul hvis vi har $R \cdot N \subseteq N$.

Nå som definisjonen av en modul er på plass, kan vi se på begrepet syzygy. Syzygyer er

sterkt knyttet til S-polynomer og denne sammenhengen blir forklart nærmere i definisjonen som kommer.

Definisjon: La $F = (f_1, \dots, f_m) \in P^m$. En syzygy til de ledende leddene $\text{LT}(f_1), \dots, \text{LT}(f_m)$ er et m -tuppel av polynomer $S = (h_1, \dots, h_m) \in P^m$ slik at

$$\sum_{i=1}^m h_i \text{LT}(f_i) = 0.$$

Vi lar $S(F)$ være undermengden av P^m som består av alle syzygynene til de ledende leddene i F .

La vektoren $e_i = (0, \dots, 0, 1, 0, \dots, 0) \in P^m$, hvor 1 er på plass nummer i . Da kan en syzygy $S \in S(F)$ skrives som $S = \sum_{i=1}^m h_i e_i$, $h_i \in P$. Ved å bruke denne notasjonen, kan en syzygy, som kommer fra S-polynomer, defineres på følgende måte.

Definisjon: La F være tuppelet $(f_1, \dots, f_m) \in P^m$. Syzyggen, S_{ij} , knyttet til S-polynomet $S(f_i, f_j)$ defineres da som

$$S_{ij} = \frac{x^\gamma}{\text{LT}(f_i)} e_i - \frac{x^\gamma}{\text{LT}(f_j)} e_j,$$

hvor x^γ som vanlig er $\text{LCM}(\text{LM}(f_i), \text{LM}(f_j))$.

En fin egenskap hos mengden $S(F)$ er at den har en endelig basis. Før dette kan bevises, trenger vi å definere hva en ensartet syzygy er.

Definisjon: Et element $S \in S(F)$ er ensartet av multigrad α , hvor $\alpha \in \mathbb{Z}_{\geq 0}$, dersom

$$S = (c_i x^{\alpha(1)}, \dots, c_m x^{\alpha(m)}),$$

hvor c_i er et element i kroppen k og $\alpha(i) + \text{multideg}(f_i) = \alpha$ når $c_i \neq 0$.

Proposisjonen som følger forteller at S_{ij} -ene danner en basis bestående av alle syzygynene til de ledende leddene $S(F) \subset P^m$.

Proposisjon 4.5.1. *Gitt $F = (f_1, \dots, f_m)$, så kan enhver syzygy, $S \in S(F)$, skrives som*

$$S = \sum_{i < j} u_{ij} S_{ij},$$

med $u_{ij} \in P$. Derved er

$$\{S_{ij} : 1 \leq i < j \leq m\},$$

med S_{ij} ensartet av multigrad γ , en ensartet basis til $S(F)$.

Bevis: Se [4]. □

Basisen i Proposisjon 4.5.1 kalles en Taylor-basis. Metoder for å oppdage en redusert basis fra Taylor-basisen gir en avansert Gröbner-basistest, siden man ikke trenger å betrakte så mange S-polynomer. Dette følger fra det neste teoremet, som er et nytt algoritmekriterium for Gröbner-baser.

Teorem 4.5.2. *La $G = (g_1, \dots, g_m) \in P^m$. En basis $\{g_1, \dots, g_m\}$ for et ideal I er en Gröbner-basis hvis og bare hvis vi for ethvert element $S = (h_1, \dots, h_m)$ i en ensartet basis for syzyggen $S(G)$, har*

$$S \cdot G = \sum_{i=1}^m h_i g_i \rightarrow_G 0.$$

Bevis: Se [4]. □

Syzyggene S_{ij} svarer til akkurat de S-polynomene $S(f_i, f_j)$ som man vil kunne komme til å betrakte under kjøringen av Buchbergers algoritme, eller lignende algoritmer for beregning av Gröbner-baser. For å kunne utnytte Teorem 4.5.2, så må man kjenne til hvordan man lager mindre ensartede basiser for $S(G)$. Vi vil vise at når man begynner med en Taylorbasis, så finnes det en systematisk måte å forutsi når elementer kan utelates. Beviset til proposisjonen som følger, er tatt med siden det omhandler en viktig idé om hvordan kritiske par kan utelates.

Proposisjon 4.5.3. *Gitt en $G = (g_1, \dots, g_m) \in P^m$, anta at vi har en undermengde $L \subset \{S_{ij} : 1 \leq i < j \leq m\}$ som er en basis for $S(G)$. Anta i tillegg at vi har distinkte elementer g_i, g_j og $g_k \in G$ slik at*

$$\text{LT}(g_k) | \text{LCM}(\text{LT}(g_i), \text{LT}(g_j)).$$

Hvis $S_{ij}, S_{jk} \in L$, så er $L \setminus \{S_{ij}\}$ også en basis for $S(G)$. Legg her merke til at dersom $i < j$, så setter vi $S_{ij} = S_{ji}$.

Bevis: Anta, uten tap av generalitet, at $i < j < k$. Sett

$$x^{\gamma_{ij}} = \text{LCM}(\text{LT}(g_i), \text{LT}(g_j))$$

og la $x^{\gamma_{ik}}$ og $x^{\gamma_{jk}}$ definieres på samme måte. Videre, la

$$x^{\gamma_{ijk}} = \text{LCM}(\text{LT}(g_i), \text{LT}(g_j), \text{LT}(g_k)).$$

Hypotesen vår impliserer at både $x^{\gamma_{ik}}$ og $x^{\gamma_{jk}}$ deler $x^{\gamma_{ij}}$. Vi skal bevise at

$$S_{ij} = \frac{x^{\gamma_{ij}}}{x^{\gamma_{ik}}} S_{ik} - \frac{x^{\gamma_{ij}}}{x^{\gamma_{jk}}} S_{jk}.$$

Det er $\frac{m(m-1)}{2}$ syzygger S_{ij} . Hvis det finnes noen form for avhengighet mellom noen av disse generatorene av $S(G)$, så vil det eksistere høyere ordens syzygger i modulen $P^{m(m-1)/2}$ som

kansellerer de aktuelle generatorene. For å kunne skape en kanonisk basis for denne modulen av syzygyer, så vil de $\frac{m(m-1)}{2}$ syzygyene være sortert ved $<_1$, som defineres slik:

$$S_{ab} <_1 S_{cd} \Leftrightarrow x^{\gamma_{ab}} < x^{\gamma_{cd}} \text{ eller } (x^{\gamma_{ab}} = x^{\gamma_{cd}}, b \leq d, \text{ hvor } b = d \Rightarrow a < c).$$

Ved å bruke denne sorteringen, betegner vi ikke lenger den kanoniske i -te enhetsvektoren i $P^{m(m-1)/2}$ med e_i . Med denne sorteringen brukes i stedet e_{ab} dersom S_{ab} er den i -te syzygyen. Modulen av syzygyer

$$S^{(2)}(G) = \left\{ \sum_{i,j=1, i < j}^m h_{ij} e_{ij} \in P^{m(m-1)/2} : \sum_{i,j=1, i < j}^m h_{ij} S_{ij} = 0 \right\}$$

har Taylor-basisen $L^{(2)} = \{S_{ijk} : 1 \leq i < j < k \leq m\}$ med

$$S_{ijk} = \frac{x^{\gamma_{ijk}}}{x^{\gamma_{ij}}} e_{ij} - \frac{x^{\gamma_{ijk}}}{x^{\gamma_{ik}}} e_{ik} + \frac{x^{\gamma_{ijk}}}{x^{\gamma_{jk}}} e_{jk}.$$

Elementene S_{ijk} er ensartede av multigrad γ_{ijk} .

Hypotesen vår impliserer at S_{ijk} og S_{ij} er ensartede av samme multigrad $\gamma_{ij} = \gamma_{ijk}$. Hvis det stemmer, så kan S_{ij} uttrykkes ved leddene til syzygyene S_{ik} og S_{jk} , fordi S_{ijk} er et element i den andre modulen av syzygyer, $S^{(2)}(G)$. Dermed er

$$S_{ij} = \frac{x^{\gamma_{ijk}}}{x^{\gamma_{ik}}} S_{ik} + \frac{x^{\gamma_{ijk}}}{x^{\gamma_{jk}}} S_{jk}.$$

Dette tillater oss å fjerne S_{ij} fra L og mengden $L \setminus \{S_{ij}\}$ genererer fremdeles $S(G)$ fordi S_{ij} kan erstattes av S_{ik} og S_{jk} i enhver basisrepresentasjon av en $S \in S(G)$. \square

Teoremet som følger presenterer Gebauer og Möller-installasjonen. Den kan beskrives som en modifisering av Buchbergers algoritme. En fordel ved Gebauer og Möller-installasjonen er at alle algoritmer som beregner en Gröbner-basis og velger ut kritiske par på en lignende måte, kan benytte seg av dette utvelgelseskriteriet. For mer detaljert informasjon henvises det til [10].

Teorem 4.5.4 (Gebauer og Möller-installasjon). *La $I = \langle f_1, \dots, f_m \rangle \neq \{0\}$ være et polynom-ideal. Da kan en Gröbner-basis for I finnes vha. et endelig antall steg i algoritmen som følger.*

```

Input:  $F = \{f_1, \dots, f_m\}$ 
Output: En Gröbner-basis  $G$  for  $\langle \{f_1, \dots, f_m\} \rangle$ .
 $G := \{f_1\}$ 
 $D := \{\}$ 
for  $t := 2$  til  $m$  do
     $OppdaterPar(D, t)$ 
     $G := G \cup \{f_t\}$ 
end for

```

```

 $r := m$ 
while  $\exists (i, j) \in D$  do
   $h := \overline{S(f_i, f_j)}^G$ 
  if  $h \neq 0$  then
     $f_{r+1} := h$ 
     $D := OppdaterPar(D, r + 1)$ 
     $G := G \cup \{f_{r+1}\}$ 
     $r := r + 1$ 
  end if
end while
return  $G$ 

```

Prosedyren *OppdaterPar* er definert som følger:

Input: En mengde av par, D , og et positivt heltall, t .

Output: En oppdatert mengde av kritiske par D , slik at $\{S_{ij} : (i, j) \in D\}$ sammen med en S_{ij} , $1 \leq i < j \leq t$, og $LT(f_i)LT(f_j) = x^{\gamma_{ij}}$, danner en basis for modulen av syzygger

$$\{(g_1, \dots, g_t) \in P^t : \sum_{i=1}^t g_i LT(f_i) = 0\}.$$

Fjern alle par (i, j) i D , som tilfredsstiller

$$x^{\gamma_{ij}} = x^{\gamma_{ijt}} \text{ og } x^{\gamma_{it}} \neq x^{\gamma_{ij}} \neq x^{\gamma_{jt}}.$$

Betegn denne mengden av gjenværende par ved D .

Sett $D1 := \{(i, t) : 1 \leq i \leq t\}$.

Fjern hvert par (i, t) i D , hvor en $(j, t) \in D1$ eksisterer, slik at

$$x^{\gamma_{jt}} | x^{\gamma_{it}} \text{ og } x^{\gamma_{jt}} \neq x^{\gamma_{it}}.$$

I hver ikke-tom undermengde $\{(j, t) : x^{\gamma_{jt}} = \tau\} \subset D1'$ med monom $\tau \in T(P)$, fiksér et element (i, t) som tilfredsstiller

$$LT(f_i)LT(f_j) = x^{\gamma_{it}}.$$

Hvis ingen slik (i, t) finnes, fiksér en vilkårlig (i, t) . Fjern de andre elementene i $\{(j, t) : x^{\gamma_{jt}} = \tau\} \subset D1'$. Slett til slutt alle (i, t) i $D1'$ med

$$LT(f_i)LT(f_t) = x^{\gamma_{it}}.$$

og la den gjenværende undermengden av $D1'$ nå betegne $D1'$.

return $D := D1' \cup D'$

Bevis: Se [10]. □

Anta at vi vil beregne en Gröbner-basis for mengden $\{f_1, \dots, f_m\}$. På samme måte som i Buchbergers algoritme, så har vi at hvis et kritisk par ikke reduseres til null, så kalles det reduserte S-polynomet for f_{m+1} og blir lagt til i vår opprinnelige mengde. Denne prosessen blir repetert og S-polynomer blir indeksert etter rekkefølgen de blir lagt til i den oprinnelige basisen. I stedet for å nå prøve alle kombinasjoner av par (i, j) , for $i \leq j < m$, kjører Gebauer og Möller-installasjonen gjennom et utvalg av alle mulige kombinasjoner.

Den siste delen av dette underkapittelet vil være en forklaring av utvelgelseskriteriet som brukes i F5. Hvis "inputen" er en regulær sekvens, så hevdes det at denne algoritmen ikke lager noen ubrukelige kritiske par.

Definisjon: La P være en polynomring og $I = \langle f_1, \dots, f_m \rangle$ et ideal i P . Da gjelder følgende:

- (i) Et ideal I blir kalt et ekte ideal hvis det er forskjellig fra P .
- (ii) Et element $f \in P$ kalles en ikke-nulldivisor hvis $fg = 0 \Rightarrow g = 0$.
- (iii) En sekvens av elementer $f_1, \dots, f_m \in R$ kalles en regulær sekvens hvis idealet I er et ekte ideal, og bildet til f_i er en ikke-nulldivisor i $P/\langle f_1, \dots, f_i - 1 \rangle$ for $i = 1, \dots, m$.

For å forklare F5-kriteriet er det nødvendig å utvide den originale sorteringen $<$ til en ny sortering $<_{P^m}$ for modulen P^m over polynomringen P . Som tidligere refererer e_i til den kanoniske i -te enhetsvektoren $(0, \dots, 0, 1, 0, \dots, 0)$ i P^m .

Definisjon: For to elementer $H = \sum_{i=j}^m h_i e_i$ og $H' = \sum_{i=j}^m h'_i e_i$ i P^m , med h_j, h'_j , i P , så er modulreddsorteringen definert som følger:

$$H <_{P^m} H' \Leftrightarrow j < j' \text{ eller } (j = j' \text{ og } \text{LT}(h_j) < \text{LT}(h'_{j'})).$$

Med en sortering av elementene i P^m , kan man snakke om et ledende modulredd.

Definisjon: Det ledende modulreddet til et element $H = \sum_{i=j}^m h_i e_i$, med $h_j \in P$, er definert som

$$\text{LMT}(H) = \text{LT}(h_j)e_j.$$

Ideen om en signatur til et polynom er både essensiell og spesiell for algoritmen F5. Faugère introduserer også begrepet indeksen til et polynom. Begge disse begrepene presenteres i definisjonen som følger.

Definisjon: Under beregningen av en Gröbner-basis for et tuppel $F = (f_1, \dots, f_m)$ vha. F5, så er signaturen til et polynom f , $Si(f)$, lik det ledende modulreddet til det minste modulelementet $H = \sum_{i=j}^m h_i e_i$, som tilfredsstiller

$$\text{LT}(H \cdot F) = \text{LT}\left(\sum_{i=j}^m h_i f_i\right) = \text{LT}(f).$$

Signaturen, $Si(f)$, har formen te_j for et ledd $t \in P$ og et heltall $j \in \{1, \dots, m\}$.

Indeksen til polynomet f er indeksen til den kanoniske enhetsvektoren i signaturen, så hvis $Si(f) = te_j$, så er $indeks(f) = j$.

I teoremet som følger blir F5-kriteriet formulert. Teoremet består av tre kriterier. Hvis det andre kriteriet holder for et element g i Gröbner-basisen G , så kalles g for passende (admissible på engelsk) i [6].

Teorem 4.5.5. La $F = \{f_1, \dots, f_m\}$ og $G = \{g_1, \dots, g_{m_G}\} \subset P$ spenne ut idealet I . Definer $x^{\gamma_{ij}} = LCM(LM(g_i), LM(g_j))$, for $i, j \in \{1, \dots, m_G\}$.

Mengden G er en Gröbner-basis hvis følgende tre kriterier holder:

(i) $F \subset G$.

(ii) For enhver $g \in G$, så eksisterer det et modulelement $H = \sum_{i=1}^m h_i e_i \in P^m$ med

$$H \cdot F = \sum_{i=1}^m h_i f_i = g,$$

slik at $LMT(H) = Si(g)$.

(iii) S-polynomet $S(g_i, g_k)$ er null eller har en t -representasjon $\sum_{l=1}^m b_l f_l$, med

(1) $t < x^{\gamma_{ij}}$

(2) $Si(t) \leq_{P^m} Si(\frac{x^{\gamma_{ij}}}{LT(g_i)} g_i)$ og $Si(t) \leq_{P^m} Si(\frac{x^{\gamma_{ij}}}{LT(g_j)} g_j)$

(3) $Si(b_l f_l) \leq_{P^m} Si(S(g_i, g_j))$, for $1 \leq l \leq m$
for alle par (i, j) tilfredsstiller

(1) $Si(g_j) <_{P^m} Si(g_i)$

(2) Hvis $Si(\frac{x^{\gamma_{ij}}}{LT(g_i)} g_i) = t_i e_{i'}$ og $Si(\frac{x^{\gamma_{ij}}}{LT(g_j)} g_j) = t_j e_{j'}$, så er ikke t_i og t_j delelig på
elementer av hhv.

$$\{LT(f) : f \in \langle f_{i'+1}, \dots, f_m \rangle\} \text{ og } \{LT(f) : f \in \langle f_{j'+1}, \dots, f_m \rangle\}.$$

Bevis: Se [6]. □

Algoritmen F5 tar et tuppel $F = (f_1, \dots, f_m)$ av polynomer som en “input” og beregner en foreløpig Gröbner-basis G_i for enhver mengde $\{f_i, \dots, f_m\}$ for $i = m$ ned til $i = 1$. Under den i -te gjennomgangen i algoritmen, tas basisen G_{i+1} og polynomet f_i som en “input”, og en Gröbner-basis G_i blir beregnet. Mens den regner ut den i -te foreløpige basisen G_i , velger algoritmen ut kritiske par etter økende grad. Et par (i, j) blir forkastet hvis ett av leddene t_i eller t_j , som dukker opp i de tilhørende signaturene (se tredje kriterium i Teorem 4.5.4) er reduserbar mhp. G_{i+1} .

Et eksempel på bruk av F5-algoritmen er å finne i [6].

Kapittel 5

Konklusjon

I denne oppgaven har vi i hovedsak sett på Gröbner-baser og hvordan disse kan brukes i angrep på kryptosystemet HFE. Her følger en oppsummering:

I kapittel 1 introduserte, og definerte, vi en rekke algebraiske begreper. Det gjaldt i første rekke varieteter, den multivariable polynomringen og en divisjonsalgoritme for den multivariable polynomringen. Dette ble brukt til å forstå det som kalles Gröbner-baser. Buchbergers algoritme ga oss en oppskrift på hvordan vi kan finne Gröbner-basisen til et ideal, og i tillegg har vi sett hvordan vi kan finne den unike reduserte Gröbner-basisen. I den siste delen av kapittelet kom vi inn på hvordan Gröbner-baser kan brukes til å løse polynomiske ligningssystem. Form-lemmaet fortalte oss at en Gröbner-basis med lex-sortering vil bringe det polynomiske ligningssystemet over på en “triangulær form”. Dette lemmaet er sentralt når vi skal forstå Faugères algoritmer F4 og F5.

Det andre kapittelet tar for seg kryptosystemet Hidden Field Equations, utviklet av Jacques Patarin. Vi så først på den historiske bakgrunnen for HFE. Siden gikk vi gjennom den matematiske bakgrunnen for systemet og ble bedre kjent med algoritmen gjennom et eksempel. Kryptosystemet HFE baserer seg på at det å løse multivariate kvadratiske ligningssett er et NP-komplett problem. Dette gjør HFE til et interessant kryptosystem, fordi RSA og de andre systemene som brukes i dag, vil bryte sammen hvis kvantedatamaskiner blir en realitet.

Kapittel 3 gir oss en smakebit på hvordan HFE kan angripes vha. Gröbner-basisberegniner. Vi fikk et lite sammendrag av hvordan Faugère knakk Patarins første HFE-utfordring. I tillegg så vi gjennom et eksempel hvordan kan Gröbner-baser kan brukes i angrep på dette kryptosystemet HFE. I forbindelse med dette så vi at graden til polynomene som oppstår under kjøring av F5, blir holdt nede på et minimum.

I det fjerde og siste kapittelet tok vi for oss avanserte metoder for å beregne Gröbner-baser. Effektiviteten til angrepene som bruker Gröbner-baser vil være avhengig av algoritmen som brukes til å beregne Gröbner-basisen. I delkapittel 4.3 så vi på den såkalte ensartede Buchberger-algoritmen. Den velger ut de kritiske parene vha. det som kalles den normale utvelgelsesstrategien. Det at den danner et rammeverk for flere andre algoritmer som beregner Gröbner-baser, gjør algoritmen sentral. I seksjon 4.4 presenteres Faugères F4-

algoritme som er en ledende algoritme når det gjelder å beregne Gröbner-baser. Den var utgangspunktet for F5 som løste Patarins utfordring. I det siste underkapittelet forklares to forbedrede utvelgelsesstrategier, nemlig Gerbauer og Möller-installasjonen og kriteriet som brukes i F5-algoritmen for å unngå ubrukelige kritiske par.

Faugère hevder i [6] at hans F5 er nesten optimal på den måten at den, gitt en regulær sekvens av polynomer, ikke lager noen kritiske par som reduseres til null. Dette gjør algoritmen interessant for videre studier. Gerbauer og Möller-installasjonen har vist seg å være nesten optimal, men den kan ikke konkurrere med F5 når det gjelder graden til polynomene underveis i algoritmen.

Mye av teorien rundt dette er såpass ny at det sannsynligvis vil komme nye ideer og algoritmer i de nærmeste årene. Det vil bl.a. kunne være interessant å se om man kan finne en optimal Gröbner-basisalgoritme når det gjelder graden underveis, og som har et effektivt kriterium for å unngå ubrukelige kritiske par.

Referanser

- [1] B. Buchberger. Multidimensional Systems Theory, D. Reidel Publishing Company (1985), side 184-232.
- [2] M. Caboara, M. Kreuzer og L. Robbiano. Efficiently Computing Minimal Sets of Critical Pairs, Journal of Symbolic Computation 38 (2004), side 1169-1190.
- [3] N. Courtois, A. Klimov, J. Patarin og A. Shamir. Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations, Advances in Cryptology - Eurocrypt 2000 1807 (2000), side 392-407.
- [4] D.A. Cox, J.B. Little og D. O'Shea. Ideals, Varieties and Algorithms, Springer forlag (2007). ISBN 9783540978473.
- [5] J.-C. Faugère. A New Efficient Algorithm for Computing Gröbner Bases (F4), Journal of Pure and Applied Algebra 139 (1999), side 61-88.
- [6] J.-C. Faugère. A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5), Proceedings of the 2002 international symposium on Symbolic and algebraic computation, ACM Special Interest Group on Symbolic and Algebraic Manipulation (2002).
- [7] J.-C. Faugère, P. Gianni, D. Lazard og T. Mora. Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering, Journal of Symbolic Computation (1993), side 329-344.
- [8] J.-C. Faugère og A. Joux. Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases, Advances in Cryptology - Crypto 2003, LNCS 2729 (2003), side 44-60.
- [9] M. Garey, D. Johnson. Computers and Intractability, a Guide to the Theory of NP-Completeness, Freeman, side 251.
- [10] R. Gebauer og H.M. Möller. On an Installation of Buchberger's Algorithm, Journal of Symbolic Computation 6 (1988), side 275-286.
- [11] A. Kipnis og A. Shamir. Cryptanalysis of the HFE Public Key Cryptosystem by Re-linearization, Advances in Cryptology - Crypto 1999 1666 (1999), side 19-30.

- [12] M. Kreuzer og L. Robbiano. Computational Commutative Algebra 1, Springer forlag (2000).
- [13] M. Kreuzer og L. Robbiano. Computational Commutative Algebra 2, Springer forlag (2004).
- [14] D. Lazard. Gröbner Bases, Gaussian Elimination and Resolution of Systems of Algebraic Equations (1983), side 146-157.
- [15] T. Matsumoto og H. Imai. Public Quadratic Polynomial-tuples for efficient signature-verification and message-encryption, EUROCRYPT'88, Springer forlag (1988), side 419-453.
- [16] J. Patarin. Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): two new Families of Asymmetric Algorithms. Advances in Cryptology EUROCRYPT'96, volume 1070 i Lecture Notes in Computer Science, side 33-48.
- [17] J. Patarin. Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88, CRYPTO'95, side 248-261.
- [18] A.J.M. Segers. Algebraic Attacks from a Gröbner Basis Perspective, Masteroppgave ved Technische Universiteit Eindhoven (oktober 2004).
- [19] P.W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. Revidert utgave av den opprinnelige utgivelsen til Peter Shor. Dette er en utvidet versjon av artikkelen som ble trykket i Proceedings of the 35th Annual Symposium on Foundation of Computer Science, Santa Fe, New Mexico (november 1994), <http://arxiv.org/abs/quant-ph/9508027>.
- [20] A.Steel. Allan Steel's Gröbner Basis Timings Page (oktober 2004), <http://magma.maths.usyd.edu.au/users/allan/gb/>.
- [21] C. Traverso. Hilbert Functions and the Buchberger Algorithm, Journal of Symbolic Computation 22 (1997), side 355-376.