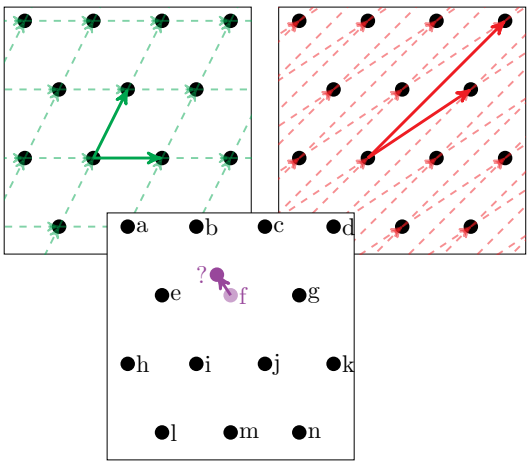
Lattice cryptography: from cryptanalysis to New Foundations

Woerden, W.P.J. van



Samenvatting

Dit proefschrift gaat hoofdzakelijk over roosters. Niet in dimensie twee of drie waar we ons nog een beeld van kunnen scheppen, maar in honderden, soms wel duizenden dimensies. Deze hoge dimensies brengen een hoop computationele problemen met zich mee. Vragen die makkelijk zijn in twee dimensies, kunnen erg lastig zijn in hogere dimensies. En deze moeilijke problemen geven aanleiding tot cryptografie.

Cryptografie heeft als doel om communicatie veilig te maken in het geval de verbinding afgeluisterd of beïnvloed wordt. Verrassend genoeg kan dit zonder dat de communicerende partijen vooraf een geheime codetaal afgesproken hebben. Dit vereist problemen die in het algemeen lastig zijn, maar die toch opgelost kunnen worden als bepaalde geheime sleutelinformatie bekend is. Als alleen de ontvanger bekend is met die geheime sleutel, dan kan alleen die het bericht lezen, en niemand anders. Denk aan een kluis met een hangslot. Iedereen kan een bericht in de kluis stoppen en het slot dichtdrukken, maar alleen de eigenaar van de sleutel kan de kluis (moeiteloos) ontgrendelen.

Roosterproblemen zijn uitermate geschikt voor zo'n versleuteling, met als extra voordeel dat ze zelfs niet door kwantumcomputers efficiënt opgelost lijken te kunnen worden. Het idee is als volgt. Een rooster kan, ondanks de hoge dimensie, omschreven worden door een paar pijlen (zie afbeelding). Een omschrijving met *lange pijlen* (rood) vormt hierbij de kluis met hangslot en een omschrijving met *korte pijlen* (groen) vormt hierbij de sleutel. Het vinden van zo'n korte omschrijving, gegeven een lange omschrijving, heet *reduceren* en dit is een moeilijk computationeel probleem. Als we een roosterpunt zien als een letter, dan kan deze versleuteld worden door het roosterpunt

iets te verplaatsen naar buiten het rooster. Het terugvinden van de oorspronkelijke locatie, en dus het ontcijferen van het bericht, heet *decoderen*. Decoderen is in het algemeen moeilijk, behalve met een korte omschrijving, de sleutel.

Deel II en III gaan over cryptanalyse met de centrale vraag: hoe moeilijk is het precies om het versleutelde bericht te lezen? We kunnen de sleutel achterhalen (reduceren), of het slot met geweld open breken (decoderen). In Hoofdstuk 4 laten we zien hoe we met moderne algoritmes en grafische kaarten in staat zijn om een rooster in dimensie 180 te reduceren, 25 dimensies meer dan het oude record. In Hoofdstuk 5 analyseren we een algoritme, dat na wat voorwerk, veel punten kan decoderen in hetzelfde rooster. In Hoofdstuk 6 bespreken we dat roosters die gebruikt worden in de cryptografie vaak wat vreemd gevormd zijn, bijvoorbeeld doordat ze roosterpunten bevatten die te dicht bij elkaar liggen. Deze afwijkingen leiden ertoe dat het reduceren of decoderen vaak aanzienlijk makkelijker is dan je zou verwachten, gegeven de hoge dimensie. We behandelen in Hoofdstuk 7 zogenoemde NTRU roosters, die al in de cryptografie gebruikt worden sinds 1996. Deze roosters hebben in sommige gevallen een nog grotere afwijking dan verwacht en wij voorspellen voor het eerst precies wat hiervan de invloed is op hun veiligheid.

Helaas zijn deze afwijkingen inherent aanwezig in alle roosters die momenteel gebruikt worden. Speciale roosters, die minder grote afwijkingen hebben, maar nog wel decodeerbaar zijn, bestaan wel en zouden in theorie veiliger zijn. Het probleem is dat deze speciale roosters voor iedereen bekend zijn, dus ook hun korte omschrijving, waardoor ze momenteel niet bruikbaar zijn. In Deel IV laten we zien dat de korte omschrijving alsnog verstopt kan worden, simpelweg door het rooster te roteren, waarbij nu de rotatie de sleutel vormt. In Hoofdstuk 9 onderbouwen we waarom het vinden van de rotatie lastig is. In Hoofdstuk 10 laten we zien hoe dit in theorie gebruikt kan worden voor het versleutelen of het digitaal ondertekenen van berichten en we geven een praktische demonstratie in de vorm van een zeer efficiënte digitale handtekening genaamd HAWK.

In het losstaande Hoofdstuk 8 laten we zien dat bekende begrippen en algoritmes voor roosters, ook vertaalbaar zijn naar lineaire codes, een soort eindige analogie van roosters. We dragen onder andere het beroemde LLL-reductiealgoritme over naar lineaire codes.