Polynomial Chains in Gentry-Szydlo Algorithm

Setting

- R ring of integers in m-th cyclotomic field K
- n degree of K
- v element of R
- <v> ideal generated by v as lattice in HNF
- $\tilde{v}$ complex conjugate
- $v\tilde{v}$ - norm of v in real subfield

Short Multiple Lemma

- "Implicit Lattice Reduction"
- For vectors v in R
- Given v ṽ and HNF of v, <v>
- We can produce a multiple of v
  - w= v a
  - a is 'LLL short' = norm <= $2^{(n-1)/2}$ sqrt(n)
  - Poly time in bit length of v and dim(R)

Congruence Lemma

- P prime =1 mod m
- v not zero divisor of R_p
- v^{P-1}=1 mod P in R
- For elements a with small coefs, |a| <P/2
- Knowledge of: a v^{P-1} mod P reveals a

Small Primes Euclidean Lemma

- $\{p_i\}$ bunch of small primes
- P and P’ both =1 mod 2m
- GCD (P-1, P’-1)=2m
- Knowledge of v^{P-1} and $v^{P'-1}$ gives v^{2m} mod $\{p_i\}$
- Suppose product of primes > 2 $|v^{2m}|$
- v^{2m} computable exactly in R

2m-th root Lemma

- Knowledge of v^{2m} gives v
- v defined up to 2m-th root of 1
- Describe proof later

Strategy for extracting v

- Choose big primes P P' bigger than LLL bound
  - =1 mod 2m and GCD (P-1, P'-1)=2m
  - Avoid P's where v zero divisor in R_p
  - Computing v^{P-1} is futile as $P > 2^n$
- For P, P' create special chains of polynomials using Short Multiple Lemma
  - Reasonable sized coefs
- Calculate v^{P-1} and $v^{P'-1}$ mod $\{p_i\}$ for small primes using Congruence Lemma
- Calculate v^{2m} then v up to root of 1

Defining Chains for P

- Goal allow expressions with v^{P-1} mod small primes
  - Motivated by square and multiply
- Write P-1 in binary as $k_0 + 2k_1 + 4k_2 \ldots. 2^r k_r$
- Each term will encode a bit k_{r-i} and an unknown v_i with known norm $v_i \tilde{v}_i$ and ideal $\langle v_i \rangle$
- These v_i build up information about v^{P-1}
- $w_1 = v\hat{}(k_{r-1})\ v^2\ \tilde{v}_1$ comes with $v_1 \tilde{v}_1$ and $\langle v_1 \rangle$
- $w_2 = v\hat{}(k_{r-2})\ v_1{}^2\ \tilde{v}_2$ comes with $v_2 \tilde{v}_2$ and $\langle v_2 \rangle$
- …..
- $w_r = v\hat{}(k_0)\ v_{r-1}{}^2\ \tilde{v}_r$ comes with $v_r \tilde{v}_r$ and $\langle v_r \rangle$

Computing terms

- First term needs $w_1$and $v_1\tilde{v}_1$and $<v_1>$
- Use known ideal $<v>$ and $v\tilde{v}$
- Create $<v\text{^}(k_{r-1}+2)>$ and $v\text{^}(k_{r-1}+2)\ \tilde{v}\text{^}(k_{r-1}+2)$
- Short Multiple Lemma gives w_1 in R
- $w_1=v\text{^}(k_{r-1})\ v^2\ \tilde{v}_1$ where $\tilde{v}_1$ is short-ish
  - Try again if $\tilde{v}_1$ is a zero divisor in R_p
- Divide out terms of $w_1 w_1\text{~}$ to get $v_1\tilde{v}_1$
- Divide out ideal terms $<w_1>$ to get $<v_1>$

General terms

- i-th term for $i>1$ needs w_i and $v_i\tilde{v}_i$ and $<v_i>$
- Use known ideal $<v_{i-1}>$ and $v_{i-1}\tilde{v}_{i-1}$
- Create $<v_{i-1}$^$(k_{r-i}+2)>$ & v_{i-1}^$(k_{r-i}+2)$ $\tilde{v}_{i-1}$^$(k_{r-i}+2)$
- Short Multiple Lemma gives w_i in R
- $w_i=v$^(k_{r-i}) $v^2\tilde{v}_i$ where $\tilde{v}_i$ is short
- Divide out terms of $w_i w_i$~ to get $v_i\tilde{v}_i$
- Divide out ideal terms $<w_i>$ to get $\langle v_i \rangle$

Using Chain

- Want $v^{P-1}\ \tilde{v}_r$ mod P (*or another prime)*
- Set $x_1=w_1=v\hat{}(2+k_{r-1})\ \tilde{v}_1$ ($v^{\text{some bits}}$ times fudge)
  - Exponent of v has 2 most significant bits of P-1
- Set $x_2=x_1^2 w_2 /(v_1\tilde{v}_1)^2$ mod P
- $=(v\hat{}(2+k_{r-1})\ \tilde{v}_1)^2\ v\hat{}(k_{r-2})\ v_1^2\,\tilde{v}_2\ /\ (v_1\tilde{v}_1)^2$
- $= v\hat{}(4+2k_{r-1}+k_{r-2})\ \tilde{v}_2$
  - Exponent of v has 3 most significant bits of P-1
- Continue so $x_r=v^{P-1}\ \tilde{v}_r$ mod P
- This = $\tilde{v}_r$ mod P.
- Since $\tilde{v}_r$ is snall get $\tilde{v}_r$ exactly in R
- Details – Make sure didn't divide by 0

Reuse for small primes

- Let q be a prime where no $v_i\tilde{v}_i$ are zero divisors in R_q
- Same chain gives $v^{P-1}\ \tilde{v}_r$ mod q
- Divide by known $\tilde{v}_r$ to get v^{P-1} mod q
- Choose many primes $\{p_i\}$ with product> $|v^{2m}|$
- Small Primes Euclidean Lemma gives v^{2m}
- 2m-th root Lemma gives v
- Done!

2m-th root Lemma Details

- v^{2m} defines v up to a 2m-th root of 1
- Embedding into **C** at a root defines v uniquely
- Compute ratios $v(s)/v(s^b)$ efficiently.
  - Take large Q= 2mc-b. $v(x)^Q=v(x^Q)$ in R_q
  - Compute $(v^{2m})^c = v^Q v^b = v(x^{-b})\, v^b$ mod Q
  - Since Q large get $z_{-b} = v(x^{-b})\, v^b$ in R
- Let s be m-th root of 1.
- Take $v^{2m}(s)$ and take an m-th root v(s) in Complex
  - $v(s^{-b})=z_{-b}(s)/v(s)^b$
- Use all n values $v(s^b)$ to find coefs of v using
  - Using linear algebra

Another Look at GS result

- Often work in $Z[X]/(X^N-1)$ Ring instead of R
  - N prime
  - Decompose as R + Z
  - Finding f from ff~ in Z is easy!
- Neglected $\{a_i\}$ (coordinate embedding)
  - Unitary Matrix

GS focus on Lattice

- GS says given $\{a_i\ f\}$ and f*f~ you can recover f and all the a_i's up to a unit u |uu~ =1
- Two easily derivable quantites
- 1. Note from $\{a_i\ f\}$ and f*f~ we easily obtain $\{a_i\ a_j\text{\textasciitilde}\}$ in R
- Const term - CT($a_i\ a_j$~)= dot product $\langle a_i\ ,\ a_j \rangle$
  - That is Gram Matrix A_{ij} = CT($a_i\ a_j$~)
  - Threw away other terms of $a_i\ a_j$~
- 2. Since we have polys $\{a_i\ f\}$, we can define x a_i
  - Map x: a_i -> Sum ($g_{i,j}\ a_i$), define $g_{i,j}$
  - Take x a_i f in Ideal, find $g_{i,j}$ so it equals Sum (g_i,j a_i) f
- This is rest of information thrown out in Gram

Gram + Group versus GS classic

- Let's focus on a_1 and get all the $a_1\, a_j\sim$
- X^e –th term of $a_1\, a_j\sim$ = $CT(x^{-e}\, a_1\, a_j\sim)$
- Use group law to mult $x^{-e} = x^{N-e}$ by a_1
  - $x^{-e}\, a_1 = Sum\,(h_i\, a_i)$ for easily computable integers h_i
- X^e –th term of $a_1\, a_j\sim$

  = CT (Sum $(h_i\, a_i)\, a_j\sim$)

  = Sum (h_i CT($a_1\, a_j\sim$)) = Sum ($h_i\, A_{ij}$)

  Gram + Group = ($a_1\, a_1\sim$) & {$a_1\, a_j\sim$ } [ideal <a_1> This is GS!
- Gram and Group Law will recover basis (mod rotation)
- Hendrik will generalize!

Information Lost in Gram

- Let a_i be vectors spanning L
  - Matrix A, AU U unimodular, different basis.
- Signed Permutation of coordinates
  - O A permutation of coords.
- Lattices Z^N equiv if B = O A U
- Gram A = A^T A
  - Gram (OA) = Gram (A)
- Group Law rigidifies lattice – nails down signed permutation

Factoring Gram Matrices

Shortest vectors in orthogonal lattices

Flavors of Lattices

- Subset of Z^N – integral basis presented
- Positive Definite Quadratic Form (PDQF).
  - Span of N independent real valued vectors (basis).
  - Discrete Subgroup of $R^{N.}$
  - Gram Matrix of some basis: $G = A * A^T$. (A has real coefs)
- Gram Matrix with *Integral* Entries.
  - (more restrictive class).
- Gram of a *Basis* with *Integral* Entries.
  - (even more restrictive class).
- Det. 1 Gram of a *Basis* with Integral Entries.

Lattice Problems

- Standard Problem Formulation.
  - Given L find Shortest vector (SVP).
  - Given L,v find Closest vector (CVP).
  - Approaches : LLL & Schnorr variants.
- *Presentation* & Conditions affect hardness
  - Standard: Know a **Basis** (Embed into R^N).
  - Alternate: Know Gram matrix of Basis.
    - Conveys *less* information.

    LLL & Schnorr use Gram data
  - If an Integral Basis *Exists,* may be hard to find!

The Orthogonal Lattice

- A lattice isomorphic to $\mathbf{Z}^N$ is called *Orthogonal*, or *Trivial*, or *Standard*
- **An easy case**: Suppose L presented as span of integer-valued basis of vectors $\{\mathbf{v}_i\}$.
  - Arrange Basis in columns as *Unimodular* Matrix U
  - Gaussian elimination for all shortest vectors!

Orthogonal Lattice Problem

- **<u>Harder Case</u>**: L is *not* presented with basis matrix.
- L: span of N *unknown* integer vectors $\{v_i\}$, only specified by Gram Matrix: $G = U^T U$.
  - Only have geometric data: dot products $\{v_i \cdot v_j\}$.
- **Problem** (OLP) : <u>Given Gram matrix of det. 1,</u> G of <u>integer-valued basis of L, find L's short vectors.</u>
  - Given $U^T U$ find $U' = OU$, O signed permutation matrix
  - Also called Embedding Prob., Gram Factorization Prob.
- Exponential lattice reduction (using G) recovers U.
  - Seems generally Infeasible !
  - This is a less studied special case though

Example of Averaging

- Old variants - GGH, NTRUSign. $Z[x]/(x^N-1)$
  - Let A be private basis (with columns $\mathbf{v}_i$).
  - Let M = AU be public basis – U unimodular.
  - Suppose 'transcript' consists of vectors s:
  - $s = Ab = \Sigma\, b_i \mathbf{v}_i$ where the b_i are <u>uniformly</u> distributed.
- $Avg(s_j s_j^T) = A\, avg(b_j\ b_j^T)\ A^T = \lambda\, AA^T$.
  - Where λ is a constant.
- Define $G = M^T (AA^T)^{-1} M = U^T U$
- G is Gram matrix of rows spanned by U.
  - Recovering OU produces AO^{-1}- reordered basis

Approach – Embedding in Z^N.

- Interest in 'factoring' $G = U^T U$.
- Standard LLL /BKZ – small dimension only
- Attempt to embed v_i in **Z**. (*know one exists*).
- Postulate tool – '**Lattice Distinguisher**'.
  - Consider Oracle Algorithms.
  - Discuss feasibility; use of Theta Functions to help realize Oracle.
  - With Oracle, we can recover v_i coordinates (mod sign,order).

Lattice Isomorphism

- We need a definition to distinguish:
  - A and B lattices bases define *isomorphic* lattices if B = O A U
  - Decisional Problem – maybe hard
- Easy cases for non-isomorphism:.
  - Determinant differs.
  - Only one of the 2 contains vectors v: $|v|^2 = n_0$.
    - Theta functions differ

Orthogonal Lattice Theorem

- Suppose we have oracle to decide if Lattices defined by Gram matrices G_1, G_2 are isomorphic
- Then there is an oracle algorithm that will produce $U' = OU$ in polynomial time from $G = U^T U$

Example - HNF

- With *integral* basis – can be easy to distinguish.
- Isomorphism class distinguished by HNF.
- Example of non-isomorphic L's. (v's in rows)

Lattice 1	Lattice 2 .
1 1 1 0 0 0 …	1 1 1 1 1 1 1 1 1 0 …
0 2 0 0 0 0 …	0 2 0 0 0 0 0 0 0 0 …
0 0 2 0 0 0 …	0 0 2 0 0 0 0 0 0 0 …
0 0 0 2 0 0 …	0 0 0 2 0 0 0 0 0 0 …

Auxiliary Lattice

- Define AUX: **span v_1, $\{2v_i\}$** (i=1,…N).
  - Gram matrix easy to make
- For embedding $\{v_i\}$ into Z^N
  - Span$\{2 v_i\}$, ~ span (2 **I**), with HNF:

2L ~

2	**0**	**0…**
0	**2**	**0…**
0	**0**	**2…**

- AUX contains 2L, with index 2.
  - We know shape of its HNF

HNF of Aux Lattice

- Very few choices for HNF in embedding
  - Allowing coordinate permutations
- Last vector has Λ 1's for some Λ in {1,..N}
- (using rows for vectors)

1	**1**	**1**	**1**	**1**	**1**	**0**	**…**
0	**2**	**0**	**0**	**0**	**0**	**0**	**…**
0	**0**	**2**	**0**	**0**	**0**	**0**	**…**

- Λ = **#** **Odd** coordinates of v_1 !

First Oracle

- Can form Aux lattice for any vector v.
  - Using given Gram matrix
- Aux lattice is isomorphic to above type.
  - For some Λ in {1,2…N}.
- Postulate that we can tell which one.
  - *Special Case* of distinguish / isomorphism problem.
- Formalize as an oracle **O**: computing:
  - Λ(v) = # odd coordinates of v.

Embedding Basis Vectors

- Start modulo 2.
- Given $\Lambda(v_1)$, write coordinates (mod 2).
  - WLOG assume first coordinates =1 mod 2
  - We are making inherent coordinate order choices.

1 1 1 1 1 0 0 0 0 0 0.

- Given $\Lambda(v_2)$, try to write next row.
  - **Q:** For how many coords. are v_1,v_2 ***both*** odd?
- Ask Oracle $\Lambda(v_1+ v_2)$, apply linear algebra.
  - **A:** ½ [$\Lambda(v_1) + \Lambda(v_2) - \Lambda(v_1+ v_2)$.]

Oracle Feasibility?

- We have reduced OLP to Oracle existence.
- Easy cases: $\Lambda(v)$ is not $\Lambda(v')$ mod 4.
  - Many witness vectors of given length mod 4.
- General oracle is more difficult.
  - $L(v) = L(v')$ mod 4.
  - Lattices still may be distinguished via differing number of vectors of given length.

Distinguishing with Lengths

Lattice 1	Lattice 2 .
1 1 1 1 1 0 …	1 1 1 1 1 1 1 1 1 0 …
0 2 0 0 0 0 …	0 2 0 0 0 0 0 0 0 0 …
0 0 2 0 0 0 …	0 0 2 0 0 0 0 0 0 0 …
0 0 0 2 0 0 …	0 0 0 2 0 0 0 0 0 0 …

<u>How many vectors of a given length?</u>

#v $\|v\|^2=5$	2^5	0
#v $\|v\|^2=9$	2^6 (N-5)	2^9

Theta Functions

- Systematic analysis of vector distributions.
- We saw huge differences - for very short vectors.
  - But not so useful – can't find them.
- The differences persist for larger vectors.
  - Question : How many vectors of length i?
- Theta function: $f(z)= \Sigma\, a_i\, z^{i}$. (z dummy variable).
  - Encoding via coefficients $a_i = \#v \mid |v^2| = a_i$.
  - Usually hard to compute, some easy cases.

Theta Examples

- Direct sum Lattices – Multiplicative Theta
- E.g: Trivial Lattice
  - $f(z) = (1 + 2z^1 + 2z^4 + 2z^9 + 2z^{16} \ldots)^N$
  - **2 I** is $f(z^2)$, previous f.
- Count vectors whose first k entries are odd
  - $f(z) = (2z^1 + 2z^9 + 2z^{25} \ldots)^{N-k} (1 + 2z^4 + 2z^{16} \ldots)^{N-k}$
- Similarly easy for all our special lattices

Using Theta Functions

- Use a very large number of *medium* length vectors $|v| \le B_{0.}$
  - Assume uniform distribution.
- Use Theta- write probability density funct.
  - Differing Statistical pdf of vectors $|v|^2 \le B_{0.}$
- Realize oracle using sample pdf.
  - Compare to candidates' pdfs. & get isom. class.

Final Thoughts

- We looked at $G = U^T U$ with no group structure!
  - But knowledge of Z^N isomorphism special
- Reduced to Lattice Distinguishing Problem
  - Interesting in its own right
- Interesting case: many approximate short vectors help find exact shortest vector