NTRU Challenge - Answers

Challenge #1 107r0

Challenge:

h=

[115 268 109 260 232 2 290 391 330 249 300 419 222 310 241 428 359 378 267
452 105 394 347 318 164 492 346 298 290 164 126 474 353 54 440 67 459 30 122
46 192 18 332 431 422 362 22 154 273 254 498 135 330 107 425 120 135 440 83
399 228 370 40 441 365 501 187 3 72 318 28 478 472 355 153 348 221 28 205 398
75 211 194 236 390 199 351 498 279 459 110 301 389 203 470 257 48 142 110 111
166 155 59 508 280 107 388]

Answer:

f'=
[0 0 0 0 0 0 6 -3 3 0 0 3 -6 -3 -6 0 -3 0 3 3 0 -3 -3 0 -3 3 3 3 0 3 0 -3 -3
0 3 3 0 3 3 0 0 0 3 -3 3 3 0 0 0 0 -3 -3 3 -3 3 -6 -3 -6 -3 -3 0 6 0 0 0 -3 0
-3 3 12 0 -3 3 0 -6 0 0 -3 -3 0 3 6 0 0 3 0 3 -6 3 0 2 3 3 -3 -6 -3 3 0 0 -3
0 0 -3 1 0 0 3]

such that

f'*h mod q=
[-2 -1 -1 1 0 1 -1 1 0 0 -2 0 -1 2 0 0 0 0 -1 0 0 -1 -1 0 0 0 0 0 1 -1 -2 2 -
1 -1 -1 0 1 -2 0 -1 0 0 1 0 -2 2 0 2 2 0 1 0 -1 2 1 -1 0 2 -2 2 -2 -1 -1 0 1
0 0 -1 -1 1 -1 1 0 1 1 0 1 0 0 -2 -1 0 -1 1 1 -1 0 -2 1 0 -2 2 1 2 2 1 -1 -1
1 -1 2 -1 0 1 0 1 0]

which isn't trinary, but it is (1-x^13)*g, where

g=
[-1 -1 -1 1 1 0 0 0 1 0 -1 1 0 1 -1 -1 1 1 -1 0 0 0 -1 -1 1 0 1 -1 0 0 -1 1 -
1 -1 -1 -1 0 -1 0 0 -1 0 1 -1 -1 1 -1 1 1 0 0 0 -1 1 1 0 -1 1 -1 1 -1 0 -1 0
1 -1 1 0 -1 0 0 0 1 0 1 -1 1 1 -1 -1 -1 -1 -1 1 1 0 0 -1 0 1 -1 1 0 1 1 0 0 0
1 -1 1 -1 1 0 1 1 1]

and f*g=h mod q. The fact g, h are not invertible makes me submit the above solution (which is way smaller than the expected small vectors) rather than take the time to f = hg^{-1} (although this should be very possible to find if you need me to).

Discovery:

I want to apply the hybrid technique to the lattice

q
ph 1

i.e. look for (F g) close to (p^{-1} 0)

but 1+p*F is harder to enumerate that g, so I first transformed this to an equivalent lattice

 q
1 X

and then removed the last 20 rows.

For reduction I started with NTL with BKZ blocksize around 22 (but tried smaller blocks with blocksize going up to 30). The reduction I ended up with had |b_i^*| ranging from

```
324.9769223
320.4735098
322.957304
299.9457301
300.2212507
```

to

```
2.442338059
2.455610966
2.399027935
```

I launched the full hybrid method on this basis, but no small vector was found - so I am not including the time to do that in the time for completion.

I resolve to increase b_n^* a little more, so I downloaded fplll, and played with it.

I gave it the previous basis, and asked it to BKZ blocksize 24 reduce the basis, printing out the answer ever 2 hours.

I resolved to wait until |b_n^*| got to about 4.

The end of the b_i^*'s looked like this:

```
> tail -5 diag6_24_002hrs.dat
2.723524089
2.674156709
2.519634014
2.310050701
2.476350194
> tail -5 diag6_24_004hrs.dat
2.735465754
2.524798366
2.462141096
2.194307381
2.456038063
> tail -5 diag6_24_006hrs.dat
2.526344657
2.613929458
2.436445281
2.283290665
2.338143311
> tail -5 diag6_24_008hrs.dat
2.509917077
2.461420683
2.455892618
2.274964103
2.116750922
> tail -5 diag6_24_010hrs.dat
2.565747318
2.472501222
2.431957464
2.840271314
34.05773339
```

This |b_n^*| of 34, was way larger than the 4 I was waiting for, and broke the GSA completely, so was a sign of reaching the solution.

NTRU Challenge - Answers

```
As is typical with these things - if you find a large b_n* in the primal,
then it corresponds to a small b_1 is the dual, and then I reversed the
coefficients, and got the above solution.
```

NTRU Challenge - Answers

Challenge #2 113r0

Challenge:

[910 191 25 288 935 686 171 790 849 922 477 528 126 464 466 488 242 326 909 727 388 887 778 807 68 20 442 306 248 241 443 826 70 118 432 962 656 196 413 330 499 349 618 388 502 653 353 1015 634 629 910 520 127 924 185 378 776 28 254 917 93 147 789 534 652 372 1009 859 39 391 682 314 733 718 579 628 809 515 383 1 636 1018 729 929 613 0 682 977 557 463 750 559 588 518 526 64 211 248 812 768 303 1005 729 116 744 71 130 339 642 743 83 509 65]

Answer:

f'=
[-9 6 -6 0 6 0 -9 3 6 -3 0 0 0 3 -3 0 0 -3 3 -3 6 -3 0 -6 6 3 0 -3 3 3 0 -3 -6 0 6 -3 -3 0 3 0 0 -6 6 -3 3 0 -3 0 3 3 -6 3 -3 0 6 -3 -3 3 3 -3 -3 3 3 0 -3 -3 3 0 0 0 0 0 0 3 -3 0 -3 3 0 3 0 -3 -3 3 -3 2 -2 3 -3 3 -3 0 6 -3 3 3 -9 6 -3 3 -3 0 0 6 -6 -3 3 0 0 3 -6 6 3]

such that

f' * h mod q=
[-1 2 0 0 -2 2 -1 1 0 -1 -1 1 0 0 1 -2 0 0 2 -2 2 0 -2 2 0 0 -2 2 -2 0 0 1 1 -1 0 -1 1 0 0 0 1 -2 2 -2 2 -2 0 2 -1 1 -2 1 1 0 -1 0 1 -1 -1 1 1 -2 0 1 0 1 -1 1 -2 1 1 -2 0 2 -1 -1 2 0 0 -2 2 -2 0 1 0 -1 2 -1 0 0 0 1 0 -1 -1 2 0 0 -1 -1 1 -1 1 -1 0 0 0 1 1 -2 1 -1 1]

which isn't trinary, but it is (1-x)*g, where

g=
[-1 1 1 1 -1 1 0 1 1 0 -1 0 0 0 1 -1 -1 -1 1 -1 1 1 -1 1 1 1 -1 1 -1 -1 -1 0 1 0 0 -1 0 0 0 0 1 -1 1 -1 1 -1 -1 1 0 1 -1 0 1 1 0 0 1 0 -1 0 1 -1 -1 0 0 1 0 1 -1 0 1 -1 -1 1 0 -1 1 1 1 -1 1 -1 -1 0 0 -1 1 0 0 0 0 1 1 0 -1 1 1 1 0 -1 0 -1 0 -1 -1 -1 -1 0 1 -1 0 -1 0]

and f*g=h mod q. The fact g, h are not invertible makes me submit the above solution (which is way smaller than the expected small vectors) rather than take the time to f = hg^{-1} (although this should be very possible to find if you need me to).

Discovery:

I want to apply the hybrid technique to the lattice

```
q
ph 1
```

i.e. look for (F g) close to (p^{-1} 0)

but 1+p*F is harder to enumerate that g, so I first transformed this to an equivalent lattice

```
  q
1 X
```

and then removed the last 30 rows.

I reduce the basis using fplll BKZ blocksize 24 reduce the basis, printing out the answer ever 2 hours.

The end of the b_i^*'s looked like this:

```
> tail -5 diag_113_002hrs.dat
4.366552965
3.980021689
4.151024102
4.238854704
4.160573947
> tail -5 diag_113_004hrs.dat
4.469161544
4.389693485
3.904888212
3.979958444
4.099543537
> tail -5 diag_113_006hrs.dat
4.140395668
4.247721305
4.014150526
3.96951532
4.311512066
> tail -5 diag_113_008hrs.dat
4.169200638
4.131102625
3.9378865
3.991580617
61.30527643
```

This |b_n^*| of 61 broke the GSA completely, so was a sign of reaching the solution.

As is typical with these things - if you find a large b_n* in the primal, then it corresponds to a small b_1 is the dual, and then I reversed the coefficients, and got the above solution.

NTRU Challenge - Answers

Challenge #3 131r0

Challenge:

```
[754, 311, 612, 431, 914, 748, 62, 714, 128, 872, 349, 1021, 700, 854, 742,
1005, 420, 955, 881, 667, 793, 412, 837, 114, 411, 29, 182, 845, 748, 756,
513, 823, 114, 691, 308, 145, 980, 962, 45, 907, 309, 578, 671, 665, 236,
920, 229, 319, 498, 566, 421, 329, 728, 943, 50, 752, 654, 309, 395, 872,
691, 463, 225, 361, 359, 635, 47, 271, 958, 70, 932, 546, 917, 597, 952, 652,
982, 948, 972, 901, 551, 418, 8, 119, 232, 863, 90, 91, 647, 309, 97, 488,
698, 837, 574, 891, 944, 822, 395, 310, 378, 23, 401, 589, 316, 438, 927,
270, 363, 781, 948, 616, 515, 878, 372, 701, 556, 605, 514, 528, 228, 83,
469, 200, 405, 641, 892, 203, 220, 465, 70]
```

Answer:

```
f = [ 2  1  1  1  0  0  0  0  0  1  0  1 -1  0  1  0  0  1  0  0  0  1 -1  0
-1  0  0 -1  0 -1  0 -1 -2  0  0  0  0  1  0  0  0  1  0  2  0  1 -1  0  0  0
0  0  0  0 -1  0 -2  2  0  0  0  1  0  0 -1  0 -1 -1 -1  0  1  0  0 -1  0  1
0  1  0  0  0  0  0  1  0  2  1  0 -2  1 -1  1  0  0  0  1  1 -1 -1 -1 -1 -2
-2  0  0 -1 -1 -1 -1  0  0  0  0  0  1  0  1  1  0  2  0 -2  0  0  0  0  0  0
0 -1  1]
```

```
g = [-1  1  0  0 -1  0 -1  1  1  1  1  0  0  1  0 -1 -1 -1  1  0  1 -1 -1 -1
1  1 -1 -1  0  0  1  1  0  0 -1  1 -1 -1  1  0  0  0  1  1 -1  0 -1 -1 -1  1
-1  0  1  0  1 -1  1  1 -1  0 -1 -1  0 -1  1  1  0  1 -1 -1  1  1  1  0  1 -1
-1 -1  1  1  1 -1  1 -1 -1  0 -1  0  1  0  1  0  0  0 -1  1 -1  0  0  0  0  0
0  1 -1 -1  0  0  0  1  0  0  1  0  0  1  1 -1 -1  1 -1  0  1  0 -1  1  1 -1
1 -1 -1]
```

Discovery:

We first used BKZ pre-processing on the Primal Lattice (the one generated by (3h, -1) and (q,0)) and this took 12 hours. Then, we search for a solution to 3 h f - g = r * h where r is the inverse of 3 mod q. We did so by a CVP pruned enumeration (tweaked linear pruning) and it took about 6 hours. Single Core, 3.2 GHz. The BKZ pre-processing was ran using fplll 4.0 (available on github), while the enumeration ran on homebrew software. The pruned enumeration failed on the first trial, but thanks to rotational symmetry, we may run several trials from the same BKZ-reduced basis. The first trial took 4 hours and was unsuccesful, the second on found a close vector after about 1 hour.

Note: Using the f and g specified in this solution, the element (3*f + 1, g) is a short element of the lattice.

NTRU Challenge - Answers

Challenge #4 139r1

Challenge:

[689, 612, 772, 630, 337, 459, 68, 285, 31, 519, 148, 804, 575, 609, 122, 517, 119, 640, 749, 138, 654, 840, 27, 172, 1, 488, 394, 597, 388, 847, 703, 167, 465, 983, 664, 165, 881, 453, 591, 973, 911, 722, 605, 877, 837, 808, 838, 747, 856, 988, 814, 675, 1019, 455, 323, 299, 407, 853, 490, 654, 150, 199, 790, 903, 271, 431, 469, 128, 982, 649, 268, 648, 86, 659, 332, 766, 54, 930, 650, 217, 957, 733, 948, 897, 997, 255, 179, 467, 309, 526, 235, 617, 816, 282, 928, 730, 860, 481, 681, 928, 575, 109, 990, 894, 293, 589, 505, 438, 592, 330, 450, 231, 221, 175, 2, 960, 191, 640, 836, 820, 640, 950, 167, 710, 61, 498, 196, 537, 678, 996, 134, 991, 176, 614, 175, 259, 953, 321, 1018]

Answer:

[1 0 3 0 -3 0 0 0 0 -3 0 -3 0 0 -3 0 0 3 6 0 3 -3 0 3 -3 -3 0 0 0 0 0 3 -3 0 0 -3 0 0 0 3 -3 3 -6 -3 -3 0 0 3 0 3 0 0 6 -3 3 -3 0 0 0 -6 0 -3 0 6 3 9 0 6 0 6 -3 -3 -3 0 -3 0 0 0 0 -3 0 -3 0 -3 3 -3 3 0 0 0 0 0 0 -3 -3 0 -3 3 0 0 0 -3 0 -3 0 0 0 0 3 0 6 0 0 0 0 0 -3 -3 0 0 -3 0 -3 3 0 3 0 0 3 0 -3 0 0 0 0 -3 6 6 3]

= 1+3F

where F = [0 0 1 0 -1 0 0 0 0 -1 0 -1 0 0 -1 0 0 1 2 0 1 -1 0 1 -1 -1 0 0 0 0 0 1 -1 0 0 -1 0 0 0 1 -1 1 -2 -1 -1 0 0 1 0 1 0 0 2 -1 1 -1 0 0 0 -2 0 -1 0 2 1 3 0 2 0 2 -1 -1 -1 0 -1 0 0 0 0 -1 0 -1 0 -1 1 -1 1 0 0 0 0 0 0 -1 -1 0 -1 1 0 0 0 -1 0 -1 0 0 0 0 1 0 2 0 0 0 0 0 -1 -1 0 0 -1 0 -1 1 0 1 0 0 1 0 -1 0 0 0 0 -1 2 2 1]

g = [-1 1 0 0 -1 0 -1 1 1 1 1 0 0 1 0 -1 -1 -1 1 0 1 -1 -1 -1 1 1 -1 -1 0 0 1 1 0 0 -1 1 -1 -1 1 0 0 0 1 1 -1 0 -1 -1 -1 1 -1 0 1 0 1 -1 1 1 -1 0 -1 -1 0 -1 1 1 0 1 -1 -1 1 1 1 0 1 -1 -1 -1 1 1 1 -1 1 -1 -1 0 -1 0 1 0 1 0 0 0 -1 1 -1 0 0 0 0 0 0 1 -1 -1 0 0 0 1 0 0 1 0 0 1 1 -1 -1 1 -1 0 1 0 -1 1 1 -1 1 -1 -1]

Discovery:

It is well-known that finding the secret key can be heuristically reduced to bounded distance decoding (BDD) over the lattice L generated by (3h, -1) and (q,0). We first computed a BKZ-reduced basis of L. Then we used this basis to solve BDD by pruned enumeration using a tailored bounding function [GNR10][1]. The BKZ pre-processing was performed by a homebrew-BKZ along the lines of [CN11][2]: this pre-processing took 10 hours on a single 1.3-Ghz core. And the search of bounding function took less than one hour on a single 1.3-Ghz core. The 9th pruned enumeration succeeded, where we took advantage of NTRU

[1] [GNR10] Nicolas Gama, Phong Q. Nguyen, Oded Regev: Lattice Enumeration Using Extreme Pruning. EUROCRYPT 2010: 257-278

[2] [CN11] Yuanmi Chen, Phong Q. Nguyen: BKZ 2.0: Better Lattice Security Estimates. ASIACRYPT 2011: 1-20

symmetries like for chl131r1. Each pruned enumeration took approximately 40 seconds on a 2.5-Ghz core, so the global enumeration time was very small, less than 10 minutes. The total running time for chl131r1 can be seen to be less than 12 hours on a 1.3-Ghz core.

NTRU Challenge - Answers

Challenge #5 149r1

Challenge:

```
[571, 837, 462, 662, 107, 172, 608, 364, 163, 980, 674, 720, 95, 17, 101,
456, 686, 808, 968, 478, 905, 749, 278, 414, 406, 651, 566, 238, 698, 724,
751, 19, 505, 694, 652, 370, 222, 172, 212, 502, 422, 745, 439, 802, 703,
731, 286, 1005, 867, 621, 601, 223, 984, 917, 895, 869, 511, 525, 682, 160,
22, 472, 378, 389, 170, 937, 9, 25, 933, 185, 717, 444, 140, 697, 274, 636,
188, 628, 350, 580, 853, 204, 747, 408, 604, 251, 844, 659, 572, 519, 130,
238, 771, 218, 661, 964, 507, 195, 313, 975, 588, 828, 312, 252, 405, 467,
997, 508, 24, 480, 356, 756, 13, 4, 370, 277, 981, 550, 188, 407, 896, 699,
966, 804, 202, 415, 839, 624, 782, 225, 565, 237, 790, 530, 27, 1018, 278,
450, 131, 366, 597, 170, 563, 319, 653, 204, 496, 373, 319]
```

Answer:

```
f = [ 1   3   3   0   0   0   0   0   0   0 -3   0   0   0   0   0   9 -3   3 -6   0   0   0   0
0   0   0 -3   0   0   6   3   0   3   0   0 -6   6 -6   0   0 -3   3   0   0 -3   3   0   0 -6
0   0   3   0   0 -3   3   3 -3   3   0   0 -3   0   0   0 -3   3 -3   0 -3   0   0   0   3   0
0   0   0 -3   0   3   0 -3 -3 -6   0   0   0 -3   0   0 -3   3   0   3 -3   3   0 -3   0 -3
0   0   3   0   0   3 -3   3   3 -3   0   0   0 -6   3 -3   0   3   0   0   3   3   0   0   0 -3
0 -3 -3 -3   0   3   0   3   3   0   0   6   0   3   3   0   0   0   0 -6   3]

= 1+3F


where F = [ 0   1   1   0   0   0   0   0   0   0 -1   0   0   0   0   0   3 -1   1 -2   0   0
0   0   0   0   0 -1   0   0   2   1   0   1   0   0 -2   2 -2   0   0 -1   1   0   0 -1   1   0
0 -2   0   0   1   0   0 -1   1   1 -1   1   0   0 -1   0   0   0 -1   1 -1   0 -1   0   0   0
1   0   0   0   0 -1   0   1   0 -1 -1 -2   0   0   0 -1   0   0 -1   1   0   1 -1   1   0 -1
0 -1   0   0   1   0   0   1 -1   1   1 -1   0   0   0 -2   1 -1   0   1   0   0   1   1   0   0
0 -1   0 -1 -1 -1   0   1   0   1   1   0   0   2   0   1   1   0   0   0   0 -2   1];


g = [ 1 -1   1   0   1   0   1   0 -1   1   1   1   0   0 -1 -1   0   0 -1   0   1   0   0 -1
-1 -1   0   1   0   0 -1   0   1   1   0   0   0   1 -1   0 -1   1   1   0   1   1 -1 -1   1   0
0   1   1   1 -1   1 -1   1   1   0 -1 -1   1 -1   0   0   0 -1 -1 -1   0 -1   1   1   0   1
1   0   0   1   0 -1   0 -1 -1 -1   0   1 -1   0   0   1   0 -1 -1   1   1   1   1 -1   1   0
-1 -1   0 -1   1   1   1   0   1   0 -1 -1   0 -1   1   0   1   1   0 -1   1   1 -1 -1 -1 -1
1 -1   0   1   1 -1 -1   0 -1   0 -1 -1   1   0 -1   0   0 -1 -1   1   1]
```

Discovery:

It is well-known that finding the secret key can be heuristically reduced to bounded distance decoding (BDD) over the lattice L generated by (3h, -1) and (q,0). We first computed a BKZ-reduced basis of L. Then we used this basis to solve BDD by pruned enumeration using a tailored bounding function [GNR10][3]. The BKZ pre-processing was performed by a homebrew-BKZ along the lines of [CN11][4]: this pre-processing took about 48 hours on a single 1.3-Ghz core. And the search of bounding function took less than one hour on a single 1.3-Ghz

[3] [GNR10] Nicolas Gama, Phong Q. Nguyen, Oded Regev: Lattice Enumeration Using Extreme Pruning. EUROCRYPT 2010: 257-278

[4] [CN11] Yuanmi Chen, Phong Q. Nguyen: BKZ 2.0: Better Lattice Security Estimates. ASIACRYPT 2011: 1-20

core. The 16th pruned enumeration succeeded, where we took advantage of NTRU symmetries like for chl139r1. Each pruned enumeration took approximately 760 seconds on a 2.5-Ghz core, so the global enumeration time was not so large, less than 4 hours. The total running time for chl149r1 can be seen to be less than 50 hours on a 1.3-Ghz core.

NTRU Challenge - Answers

Challenge #6 163r1

Challenge:

[391, 858, 742, 459, 973, 558, 532, 799, 233, 936, 523, 614, 304, 481, 978, 513, 356, 737, 593, 43, 944, 277, 273, 667, 979, 150, 302, 898, 826, 141, 672, 203, 281, 494, 995, 771, 839, 946, 805, 21, 481, 504, 20, 450, 596, 465, 22, 577, 149, 240, 169, 226, 72, 380, 61, 736, 735, 665, 91, 488, 51, 288, 753, 69, 351, 325, 459, 740, 860, 316, 504, 71, 934, 277, 555, 1007, 920, 612, 771, 753, 842, 347, 758, 328, 49, 216, 466, 826, 810, 524, 336, 815, 693, 486, 867, 543, 557, 359, 199, 729, 445, 209, 142, 460, 741, 619, 777, 612, 209, 574, 419, 440, 9, 225, 113, 388, 836, 341, 933, 524, 144, 372, 977, 908, 600, 866, 232, 133, 240, 552, 134, 456, 158, 371, 465, 503, 932, 385, 829, 838, 389, 249, 624, 207, 926, 621, 539, 863, 664, 32, 77, 1008, 215, 401, 297, 1011, 1001, 571, 436, 340, 968, 951, 868]

Answer:

Let F = [0 1 0 0 0 0 -1 0 1 0 0 0 0 1 0 1 1 1 -2 -1 0 1 -2 0 -2 0 0 0 -1 0 0 0 -1 -2 -1 0 -1 -1 -1 0 1 0 -1 0 -1 0 1 -2 0 0 0 -2 0 0 1 0 0 0 0 -1 -1 0 0 -2 -1 -2 0 -1 1 -1 0 0 0 -1 0 0 1 0 0 0 0 0 0 1 0 0 0 -1 0 -1 0 0 0 0 0 0 1 0 0 0 0 0 0 -1 1 0 0 0 0 1 1 0 0 1 1 0 -1 0 0 4 0 0 0 1 1 1 0 1 2 0 0 0 -1 2 0 1 0 1 0 0 -1 1 1 0 0 0 0 0 0 1 0 0 0 0 -1 0 0 1 0 0 1 0 1]

f = 1+3*F = [1 3 0 0 0 0 -3 0 3 0 0 0 0 3 0 3 3 3 -6 -3 0 3 -6 0 -6 0 0 0 -3 0 0 0 -3 -6 -3 0 -3 -3 -3 0 3 0 -3 0 -3 0 3 -6 0 0 0 -6 0 0 3 0 0 0 0 -3 -3 0 0 -6 -3 -6 0 -3 3 -3 0 0 0 -3 0 0 3 0 0 0 0 0 0 3 0 0 0 -3 0 -3 0 0 0 0 0 0 3 0 0 0 0 0 0 -3 3 0 0 0 0 3 3 0 0 3 3 0 -3 0 0 12 0 0 0 3 3 3 0 3 6 0 0 0 -3 6 0 3 0 3 0 0 -3 3 3 0 0 0 0 0 0 3 0 0 0 0 -3 0 0 3 0 0 3 0 3]

g = [0 1 1 0 0 1 0 -1 -1 -1 -1 0 -1 1 1 -1 -1 -1 1 1 -1 0 -1 -1 0 -1 0 0 1 -1 -1 0 0 1 -1 1 0 -1 -1 -1 -1 1 1 1 1 -1 0 0 1 1 -1 1 -1 -1 1 1 1 0 1 0 1 0 1 0 -1 -1 0 -1 0 1 0 -1 0 1 0 1 -1 -1 -1 1 1 -1 -1 0 0 1 -1 0 0 0 -1 0 -1 1 1 0 -1 0 0 -1 -1 0 0 1 -1 0 0 1 0 1 -1 1 1 1 0 1 1 0 -1 1 -1 0 -1 1 0 -1 1 1 1 1 0 0 1 -1 1 -1 -1 1 0 -1 0 1 1 -1 -1 1 -1 0 1 0 0 0 0 0 0 0 -1 1 1 -1 -1 0 1]

Then h*f = g mod q

Discovery:

Similar to ch131r1, ch139r1 and ch149r1.

The BKZ preprocessing was performed with a homebrew-BKZ and the BKZ 2.0 implementation of [CN11][5]: the preprocessing running time was approximately 30 core-days on a 2.53GHz core. The BDD enumeration was performed by a slightly modified version of the [LN13][6] algorithm, which is a BDD adaptation of the SVP pruned-enumeration algorithm [GNR10][7]. The secret key was found after approximately 100 enumerations, and each enumeration took about 9000s on a 2.53GHz core, so the global enumeration running time was 11 core-days. Hence, the total running time is about 41 core-days.

[5] [GNR10] Nicolas Gama, Phong Q. Nguyen, Oded Regev: Lattice Enumeration Using Extreme Pruning. EUROCRYPT 2010: 257-278

[6] [CN11] Yuanmi Chen, Phong Q. Nguyen: BKZ 2.0: Better Lattice Security Estimates. ASIACRYPT 2011: 1-20

[7] [LN13] Mingjie Liu, Phong Q. Nguyen: Solving BDD by Enumeration: An Update. CT-RSA 2013: 293-309.

NTRU Challenge - Answers

Challenge #7 173r1

Challenge:

[349, 1012, 730, 343, 582, 26, 372, 113, 748, 272, 510, 490, 926, 792, 89,
152, 330, 513, 798, 254, 998, 232, 421, 584, 590, 96, 735, 81, 616, 279, 212,
654, 5, 844, 551, 63, 960, 153, 68, 109, 586, 422, 497, 786, 6, 558, 405,
208, 141, 324, 6, 322, 123, 392, 235, 440, 737, 881, 980, 856, 226, 472, 763,
244, 419, 201, 62, 547, 575, 660, 516, 788, 666, 968, 922, 729, 166, 751,
936, 951, 783, 27, 704, 654, 883, 970, 475, 193, 660, 728, 250, 1008, 532,
264, 963, 147, 659, 9, 607, 415, 21, 888, 15, 460, 369, 998, 855, 687, 66,
928, 218, 235, 266, 429, 74, 413, 685, 1, 733, 46, 846, 381, 584, 155, 475,
419, 252, 1012, 1018, 265, 892, 118, 65, 354, 600, 213, 793, 600, 7, 413,
671, 103, 515, 342, 643, 878, 579, 398, 895, 297, 719, 174, 462, 71, 71, 319,
744, 632, 571, 731, 573, 793, 255, 290, 380, 471, 1006, 544, 975, 81, 13,
204, 646]

Answer:

Let F = [0 1 -1 2 -1 -1 -1 0 0 -1 1 1 0 0 1 0 1 0 0 -1 0 -1 -1 -1 1 -2 0 0 0
1 0 1 1 0 1 0 0 1 0 1 -1 0 -2 0 0 -1 0 1 -2 1 0 0 -1 -1 0 -1 -1 0 0 0 0 0 0 1
0 0 -1 1 -2 0 0 -1 0 1 0 1 1 1 0 0 -1 1 0 -1 0 -1 0 1 0 0 0 1 0 0 0 0 -2 0 -2
1 -1 0 -1 0 0 1 0 0 -1 0 1 1 -1 -1 1 0 0 0 0 0 0 0 0 0 0 1 -1 0 2 -1 0 -1 1
-1 1 0 0 1 0 1 -1 0 0 0 -1 0 1 -1 0 0 0 1 1 1 0 0 1 1 0 -1 1 0 0 0 0 1 0 0 0
-1 1 0]

f = 1+3*F = [1 3 -3 6 -3 -3 -3 0 0 -3 3 3 0 0 3 0 3 0 0 -3 0 -3 -3 -3 3 -6 0
0 0 3 0 3 3 0 3 0 0 3 0 3 -3 0 -6 0 0 -3 0 3 -6 3 0 0 -3 -3 0 -3 -3 0 0 0 0 0
0 3 0 0 -3 3 -6 0 0 -3 0 3 0 3 3 3 0 0 -3 3 0 -3 0 -3 0 3 0 0 0 3 0 0 0 0 -6
0 -6 3 -3 0 -3 0 0 3 0 0 -3 0 3 3 -3 -3 3 0 0 0 0 0 0 0 0 0 0 3 -3 0 6 -3 0
-3 3 -3 3 0 0 3 0 3 -3 0 0 0 -3 0 3 -3 0 0 0 3 3 3 0 0 3 3 0 -3 3 0 0 0 0 3 0
0 0 -3 3 0]

g = [-1 0 1 -1 1 1 1 -1 1 1 0 -1 -1 -1 1 -1 1 -1 -1 -1 -1 -1 -1 -1 1 1 0 0 0
0 -1 1 -1 0 1 0 -1 1 1 -1 1 0 1 0 1 1 -1 1 1 -1 -1 1 -1 0 1 0 1 0 1 0 1 0 0 0
1 0 0 0 1 0 -1 -1 0 -1 0 1 1 -1 0 1 0 0 -1 -1 0 -1 -1 0 1 -1 -1 1 -1 0 1 0 0
1 0 1 -1 0 -1 -1 -1 0 0 0 -1 1 -1 0 0 1 1 -1 -1 -1 1 0 0 -1 1 1 0 -1 0 1 0 0
1 1 1 0 -1 -1 1 -1 0 1 0 -1 -1 -1 -1 0 1 0 1 1 1 1 1 1 1 -1 -1 0 -1 -1 0 1 -1
0 0 1 1 1 -1 0 -1 0 0]

Then h*f = g mod q

Discovery:

Similar to ch163r1.

The BKZ preprocessing was performed with a homebrew-BKZ and the BKZ 2.0 implementation of [CN11][8]: the preprocessing running time was approximately 240 core-days on a 2.53GHz core. The BDD enumeration was performed by a

[8] [CN11] Yuanmi Chen, Phong Q. Nguyen: BKZ 2.0: Better Lattice Security Estimates. ASIACRYPT 2011: 1-20

slightly modified version of the algorithm presented in [LN13][9], which is an adaptation of the SVP-enumeration algorithm presented in [GNR10][10].

The secret key was found after approximately 400 enumerations, and each enumeration took about 3 hours on a 2.53GHz core, so the global enumeration running time was 50 core-days. The total running time is therefore about 290 core-days.

Compared to chl163r1, there was one noticeable difference: several reduced bases were used.

[9] [LN13] Mingjie Liu, Phong Q. Nguyen: Solving BDD by Enumeration: An Update. CT-RSA 2013: 293-309.
[10] [GNR10] Nicolas Gama, Phong Q. Nguyen, Oded Regev: Lattice Enumeration Using Extreme Pruning. EUROCRYPT 2010: 257-278