Open access • Book Chapter • DOI:10.1007/978-3-662-53887-6_26

Optimization of $$\mathsf {LPN}$$ Solving Algorithms — Source link

Sonia Bogos, Serge Vaudenay

Institutions: École Polytechnique Fédérale de Lausanne

Published on: 04 Dec 2016 - International Conference on the Theory and Application of Cryptology and Information Security

Topics: Path (graph theory), Time complexity, Graph (abstract data type) and Linear code

Optimization of LPN Solving Algorithms
Additional material

We present below the chains without *guess-secret*. The codes used in the *code-reduce* correspond to the perfect and quasi-perfect codes presented in the paper, to which we add random codes that we generated. The list of random codes can be found in this document as well.

1 Solving LPN without *guess-secret*

```
% below, lsumc and lmaxc are the logarithmic total and max- complexities

% what follows uses precision .1, without guess

k=32 tau=0.050 theta=0.33 precision=0.10 lmaxc=11.26 lsumc=13.89:
  (32,11.2)-drop(1)-(31,10.2)-drop(4)-(27,6.2)-xor(5)-(22,6.4)-xor(5)
      -(17,6.8)-xor(6)-(11,6.6)-xor(5)-(6,7.2)-xor(5)-(1,8.4)-WHT
k=32 tau=0.100 theta=0.33 precision=0.10 lmaxc=12.70 lsumc=15.04:
  (32,12.7)-drop(1)-(31,11.7)-drop(4)-(27,7.7)-xor(7)-(20,7.4)-xor(6)
      -(14,7.8)-xor(6)-(8,8.6)-xor(7)-(1,9.2)-WHT
k=32 tau=0.125 theta=0.33 precision=0.10 lmaxc=13.52 lsumc=15.66:
  (32,13.3)-drop(1)-(31,12.3)-drop(4)-(27,8.3)-xor(7)-(20,8.6)-xor(7)
      -(13,9.2)-xor(8)-(5,9.4)-drop(4)-(1,5.4)-WHT
k=32 tau=0.200 theta=0.33 precision=0.10 lmaxc=14.80 lsumc=17.01:
  (32,14.8)-drop(1)-(31,13.8)-drop(4)-(27,9.8)-xor(9)-(18,9.6)-xor(8)
      -(10,10.2)-xor(8)-(2,11.4)-drop(1)-(1,10.4)-WHT
k=32 tau=0.250 theta=0.33 precision=0.10 lmaxc=16.30 lsumc=18.42:
  (32,16.3)-drop(1)-(31,15.3)-drop(4)-(27,11.3)-xor(11)-(16,10.6)-xor(8)
      -(8,12.2)-WHT

k=48 tau=0.050 theta=0.33 precision=0.10 lmaxc=12.94 lsumc=14.52:
  (48,5.8)-sparse-(48,2.9)-xor(1)-(47,3.6)-xor(1)-(46,5.1)-xor(2)-(44,7.2)-
      code([44,1,r])-(1,7.2)-WHT
k=48 tau=0.100 theta=0.33 precision=0.10 lmaxc=16.43 lsumc=18.58:
  (48,16)-drop(1)-(47,15)-drop(4)-(43,11)-xor(10)-(33,11)-xor(10)-(23,11)-
      xor(10)-(13,11)-xor(9)-(4,12)-drop(3)-(1,9)-WHT
k=48 tau=0.125 theta=0.33 precision=0.10 lmaxc=17.00 lsumc=19.29:
  (48,16.5)-drop(1)-(47,15.5)-drop(4)-(43,11.5)-xor(11)-(32,11)-xor(9)
      -(23,12)-xor(11)-(12,12)-xor(10)-(2,13)-drop(1)-(1,12)-WHT
k=48 tau=0.200 theta=0.33 precision=0.10 lmaxc=19.23 lsumc=21.25:
  (48,18.8)-drop(1)-(47,17.8)-drop(4)-(43,13.8)-xor(13)-(30,13.6)-xor(12)
      -(18,14.2)-xor(13)-(5,14.4)-drop(4)-(1,10.4)-WHT
k=48 tau=0.250 theta=0.33 precision=0.10 lmaxc=20.43 lsumc=22.34:
```

```
  (48,20)-drop(1)-(47,19)-drop(4)-(43,15)-xor(14)-(29,15)-xor(14)-(15,15)-
      drop(2)-(13,13)-WHT

k=64 tau=0.050 theta=0.33 precision=0.10 lmaxc=14.43 lsumc=16.04:
  (64,6.3)-sparse-(64,3.9)-xor(1)-(63,5.7)-xor(2)-(61,8.4)-code([61,1,r])
      -(1,8.4)-WHT
k=64 tau=0.100 theta=0.33 precision=0.10 lmaxc=19.38 lsumc=21.58:
  (64,9.3)-sparse-(64,9.1)-xor(4)-(60,13.2)-xor(12)-(48,13.4)-xor(12)
      -(36,13.8)-code([36,1,r])-(1,13.8)-WHT
k=64 tau=0.125 theta=0.33 precision=0.10 lmaxc=20.50 lsumc=22.94:
  (64,20.5)-drop(1)-(63,19.5)-drop(5)-(58,14.5)-xor(14)-(44,14)-xor(12)
      -(32,15)-xor(14)-(18,15)-xor(13)-(5,16)-drop(4)-(1,12)-WHT
k=64 tau=0.200 theta=0.33 precision=0.10 lmaxc=22.00 lsumc=24.42:
  (64,22)-drop(1)-(63,21)-drop(5)-(58,16)-xor(15)-(43,16)-xor(15)-(28,16)-
      xor(14)-(14,17)-WHT
k=64 tau=0.250 theta=0.33 precision=0.10 lmaxc=24.58 lsumc=26.86:
  (64,24.5)-drop(1)-(63,23.5)-drop(5)-(58,18.5)-xor(18)-(40,18)-xor(16)
      -(24,19)-xor(17)-(7,20)-WHT

k=100 tau=0.050 theta=0.33 precision=0.10 lmaxc=18.46 lsumc=20.47:
  (100,7.9)-sparse-(100,7.1)-xor(2)-(98,11.2)-xor(10)-(88,11.4)-xor(10)
      -(78,11.8)-code([78,1,r])-(1,11.8)-WHT
k=100 tau=0.100 theta=0.33 precision=0.10 lmaxc=25.39 lsumc=27.61:
  (100,14.9)-sparse-(100,14.9)-xor(11)-(89,17.8)-xor(16)-(73,18.6)-xor(17)
      -(56,19.2)-code([18,5,rnd150927][19,6,rnd150927][19,6,rnd150927])
      -(17,19.2)-WHT
k=100 tau=0.125 theta=0.33 precision=0.10 lmaxc=26.30 lsumc=28.91:
  (100,26.3)-drop(1)-(99,25.3)-drop(6)-(93,19.3)-xor(18)-(75,19.6)-xor(19)
      -(56,19.2)-xor(17)-(39,20.4)-xor(19)-(20,20.8)-drop(2)-(18,18.8)-WHT
k=100 tau=0.200 theta=0.33 precision=0.10 lmaxc=29.75 lsumc=32.06:
  (100,29.1)-drop(1)-(99,28.1)-drop(5)-(94,23.1)-xor(22)-(72,23.2)-xor(22)
      -(50,23.4)-xor(22)-(28,23.8)-drop(6)-(22,17.8)-WHT
k=100 tau=0.250 theta=0.33 precision=0.10 lmaxc=30.75 lsumc=32.94:
  (100,30.1)-drop(1)-(99,29.1)-drop(5)-(94,24.1)-xor(23)-(71,24.2)-xor(23)
      -(48,24.4)-xor(23)-(25,24.8)-drop(3)-(22,21.8)-WHT

k=256 tau=0.050 theta=0.33 precision=0.10 lmaxc=34.45 lsumc=36.75:
  (256,21.8)-sparse-(256,21.8)-xor(17)-(239,25.6)-xor(24)-(215,26.2)-xor
      (25)-(190,26.4)-xor(25)-(165,26.8)-code([165,1,r])-(1,26.8)-WHT
k=256 tau=0.100 theta=0.33 precision=0.10 lmaxc=44.22 lsumc=46.75:
  (256,32.3)-sparse-(256,32.3)-xor(28)-(228,35.6)-xor(34)-(194,36.2)-xor
      (35)-(159,36.4)-xor(35)-(124,36.8)-code([11,4,S][18,5,rnd150927
      ][19,5,rnd150927][19,5,rnd150927][19,5,rnd150927][19,5,rnd150927
      ][19,5,rnd150927])-(34,36.8)-WHT
k=256 tau=0.125 theta=0.33 precision=0.10 lmaxc=47.35 lsumc=49.90:
  (256,35.6)-sparse-(256,35.6)-xor(31)-(225,39.2)-xor(38)-(187,39.4)-xor
      (38)-(149,39.8)-xor(39)-(110,39.6)-code([15,5,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926][19,7,rnd150926][19,7,
      rnd150926])-(37,39.6)-WHT
```

```
k=256 tau=0.200 theta=0.33 precision=0.10 lmaxc=53.82 lsumc=56.31:
  (256,42.4)-sparse-(256,42.4)-xor(38)-(218,45.8)-xor(45)-(173,45.6)-xor
      (44)-(129,46.2)-xor(45)-(84,46.4)-code([15,7,2BCH][25,15,W][25,15,W
      ][19,7,rnd150927])-(44,46.4)-drop(1)-(43,45.4)-WHT
k=256 tau=0.250 theta=0.33 precision=0.10 lmaxc=56.88 lsumc=59.47:
  (256,56.7)-drop(1)-(255,55.7)-drop(7)-(248,48.7)-xor(48)-(200,48.4)-xor
      (47)-(153,48.8)-xor(47)-(106,49.6)-xor(59)-(47,39.2)-WHT

k=512 tau=0.050 theta=0.33 precision=0.10 lmaxc=55.09 lsumc=57.77:
  (512,41.3)-sparse-(512,41.3)-xor(36)-(476,45.6)-xor(44)-(432,46.2)-xor
      (46)-(386,45.4)-xor(44)-(342,45.8)-xor(44)-(298,46.6)-code([9,1,r
      ][9,1,r][11,1,r][11,1,r][11,1,r][19,3,rnd150927][19,3,rnd150927
      ][19,3,rnd150927][19,3,rnd150927][19,3,rnd150927][19,3,rnd150927
      ][19,3,rnd150927][19,3,rnd150927][19,3,rnd150927][19,3,rnd150927
      ][19,3,rnd150927][19,3,rnd150927][19,3,rnd150927])-(44,46.6)-WHT
k=512 tau=0.100 theta=0.33 precision=0.10 lmaxc=70.92 lsumc=73.68:
  (512,57.4)-sparse-(512,57.4)-xor(52)-(460,61.8)-xor(61)-(399,61.6)-xor
      (60)-(339,62.2)-xor(61)-(278,62.4)-xor(61)-(217,62.8)-code([8,2,iGop
      ][19,5,rnd150927][19,5,rnd150927][19,5,rnd150927][19,6,rnd150927
      ][19,6,rnd150927][19,5,rnd150927][19,5,rnd150927][19,5,rnd150927
      ][19,5,rnd150927][19,5,rnd150927][19,5,rnd150927])-(59,62.8)-WHT
k=512 tau=0.125 theta=0.33 precision=0.10 lmaxc=76.22 lsumc=78.85:
  (512,63.3)-sparse-(512,63.3)-xor(59)-(453,66.6)-xor(65)-(388,67.2)-xor
      (66)-(322,67.4)-xor(66)-(256,67.8)-xor(67)-(189,67.6)-code([18,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,7,rnd150926][19,7,rnd150926][19,7,
      rnd150926][19,7,rnd150926])-(64,67.6)-WHT
k=512 tau=0.200 theta=0.33 precision=0.10 lmaxc=86.38 lsumc=89.04:
  (512,73.6)-sparse-(512,73.6)-xor(69)-(443,77.2)-xor(76)-(367,77.4)-xor
      (76)-(291,77.8)-xor(77)-(214,77.6)-xor(76)-(138,78.2)-code([23,12,G
      ][25,15,W][25,15,W][25,15,W][25,15,W][15,4,rnd150927])-(76,78.2)-drop
      (2)-(74,76.2)-WHT
k=512 tau=0.250 theta=0.33 precision=0.10 lmaxc=91.97 lsumc=94.66:
  (512,79.3)-sparse-(512,79.3)-xor(75)-(437,82.6)-xor(81)-(356,83.2)-xor
      (82)-(274,83.4)-xor(82)-(192,83.8)-xor(83)-(109,83.6)-code([31,26,H
      ][16,15,Chop][31,20,W][31,20,W])-(81,83.6)-drop(1)-(80,82.6)-WHT

k=768 tau=0.050 theta=0.33 precision=0.10 lmaxc=74.03 lsumc=76.63:
  (768,60)-sparse-(768,60)-xor(55)-(713,64)-xor(63)-(650,64)-xor(63)
      -(587,64)-xor(63)-(524,64)-xor(62)-(462,65)-code([11,1,r][11,1,r
      ][19,3,rnd150927][19,3,rnd150927][19,3,rnd150927][19,3,rnd150927
      ][19,3,rnd150927][19,3,rnd150927][19,3,rnd150927][13,1,r][13,1,r
      ][13,1,r][13,1,r][13,1,r][13,1,r][13,1,r][13,1,r][13,1,r][19,3,
      rnd150927][19,3,rnd150927][19,3,rnd150927][19,3,rnd150927][19,3,
      rnd150927][19,3,rnd150927][19,3,rnd150927][19,3,rnd150927][19,3,
      rnd150927][19,3,rnd150927])-(62,65)-WHT
k=768 tau=0.100 theta=0.33 precision=0.10 lmaxc=96.04 lsumc=98.97:
  (768,82.1)-sparse-(768,82.1)-xor(77)-(691,86.2)-xor(85)-(606,86.4)-xor
      (85)-(521,86.8)-xor(86)-(435,86.6)-xor(85)-(350,87.2)-xor(86)
```

```
      -(264,87.4)-code([17,5,rnd150927][19,6,rnd150927][19,6,rnd150927
      ][19,6,rnd150927][19,6,rnd150927][19,6,rnd150927][19,6,rnd150927
      ][19,6,rnd150927][19,6,rnd150927][19,6,rnd150927][19,6,rnd150927
      ][19,6,rnd150927][19,6,rnd150927][19,6,rnd150927])-(83,87.4)-WHT
k=768 tau=0.125 theta=0.33 precision=0.10 lmaxc=103.01 lsumc=105.89:
  (768,89.1)-sparse-(768,89.1)-xor(84)-(684,93.2)-xor(92)-(592,93.4)-xor
      (92)-(500,93.8)-xor(93)-(407,93.6)-xor(92)-(315,94.2)-xor(93)
      -(222,94.4)-code([25,15,W][25,15,W][25,15,W][14,3,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926])-(90,94.4)-WHT
k=768 tau=0.200 theta=0.33 precision=0.10 lmaxc=118.18 lsumc=121.04:
  (768,104.7)-sparse-(768,104.7)-xor(100)-(668,108.4)-xor(107)-(561,108.8)-
      xor(108)-(453,108.6)-xor(107)-(346,109.2)-xor(108)-(238,109.4)-xor
      (108)-(130,109.8)-code([63,57,H][16,15,Chop][20,13,W][31,20,W])
      -(105,109.8)-WHT
k=768 tau=0.250 theta=0.33 precision=0.10 lmaxc=124.63 lsumc=127.35:
  (768,111.3)-sparse-(768,111.3)-xor(107)-(661,114.6)-xor(113)-(548,115.2)-
      xor(114)-(434,115.4)-xor(114)-(320,115.8)-xor(115)-(205,115.6)-code
      ([5,1,r][25,15,W][25,15,W][25,15,W][25,15,W][25,15,W][25,15,W][25,15,
      W][25,15,W])-(121,115.6)-drop(9)-(112,106.6)-WHT
```

2 Solving LPN with *guess-secret*

We present below the chains that use *guess-secret*.

```
% below, lsumc and lmaxc are the logarithmic total and max- complexities
% lrep is the logarithmic number of repetitions applied to WHT

% what follows uses precision .1, with guess

k=32 tau=0.050 theta=0.33 precision=0.10 lmaxc=10.90 lsumc=11.85 lrep=8.57:
  (32,5.1)-sparse-(32,1.2)-guess(13,3)-(19,1.2)-code([19,1,r])-(1,1.2)-WHT
k=32 tau=0.100 theta=0.33 precision=0.10 lmaxc=11.65 lsumc=12.41 lrep=8.87:
  (32,5.1)-sparse-(32,1.2)-guess(23,2)-(9,1.2)-code([9,1,r])-(1,1.2)-WHT
k=32 tau=0.125 theta=0.33 precision=0.10 lmaxc=12.40 lsumc=13.30 lrep=9.96:
  (32,5.1)-sparse-(32,1.2)-guess(26,2)-(6,1.2)-code([6,1,r])-(1,1.2)-WHT
k=32 tau=0.200 theta=0.33 precision=0.10 lmaxc=14.80 lsumc=17.01 lrep=0.00:
  (32,14.8)-drop(1)-(31,13.8)-drop(4)-(27,9.8)-xor(9)-(18,9.6)-xor(8)
      -(10,10.2)-xor(8)-(2,11.4)-drop(1)-(1,10.4)-WHT
k=32 tau=0.250 theta=0.33 precision=0.10 lmaxc=16.30 lsumc=18.42 lrep=0.00:
  (32,16.3)-drop(1)-(31,15.3)-drop(4)-(27,11.3)-xor(11)-(16,10.6)-xor(8)
      -(8,12.2)-WHT

k=48 tau=0.050 theta=0.33 precision=0.10 lmaxc=12.52 lsumc=13.01 lrep=8.27:
  (48,5.7)-sparse-(48,2)-guess(23,2)-(25,2)-code([25,1,r])-(1,2)-WHT
k=48 tau=0.100 theta=0.33 precision=0.10 lmaxc=14.25 lsumc=15.23 lrep
    =11.35:
  (48,5.7)-sparse-(48,2)-guess(37,2)-(11,2)-code([11,1,r])-(1,2)-WHT
```

```
k=48 tau=0.125 theta=0.33 precision=0.10 lmaxc=15.49 lsumc=16.49 lrep
    =12.68:
  (48,5.7)-sparse-(48,2)-guess(39,2)-(9,2)-code([9,1,r])-(1,2)-WHT
k=48 tau=0.200 theta=0.33 precision=0.10 lmaxc=19.23 lsumc=21.25 lrep=0.00:
  (48,18.8)-drop(1)-(47,17.8)-drop(4)-(43,13.8)-xor(13)-(30,13.6)-xor(12)
      -(18,14.2)-xor(13)-(5,14.4)-drop(4)-(1,10.4)-WHT
k=48 tau=0.250 theta=0.33 precision=0.10 lmaxc=20.43 lsumc=22.34 lrep=0.00:
  (48,20)-drop(1)-(47,19)-drop(4)-(43,15)-xor(14)-(29,15)-xor(14)-(15,15)-
      drop(2)-(13,13)-WHT

k=64 tau=0.050 theta=0.33 precision=0.10 lmaxc=13.74 lsumc=14.44 lrep
    =10.04:
  (64,6.1)-sparse-(64,2.2)-guess(38,2)-(26,2.2)-code([26,1,r])-(1,2.2)-WHT
k=64 tau=0.100 theta=0.33 precision=0.10 lmaxc=16.76 lsumc=17.71 lrep
    =13.80:
  (64,6.1)-sparse-(64,2.2)-guess(52,2)-(12,2.2)-code([12,1,r])-(1,2.2)-WHT
k=64 tau=0.125 theta=0.33 precision=0.10 lmaxc=18.61 lsumc=20.57 lrep
    =12.07:
  (64,6.8)-sparse-(64,5.6)-xor(1)-(63,9.2)-xor(8)-(55,9.4)-guess(36,2)
      -(19,9.4)-code([19,4,rnd150926])-(4,9.4)-drop(3)-(1,6.4)-WHT
k=64 tau=0.200 theta=0.33 precision=0.10 lmaxc=22.00 lsumc=24.42 lrep=0.00:
  (64,22)-drop(1)-(63,21)-drop(5)-(58,16)-xor(15)-(43,16)-xor(15)-(28,16)-
      xor(14)-(14,17)-WHT
k=64 tau=0.250 theta=0.33 precision=0.10 lmaxc=24.58 lsumc=26.86 lrep=0.00:
  (64,24.5)-drop(1)-(63,23.5)-drop(5)-(58,18.5)-xor(18)-(40,18)-xor(16)
      -(24,19)-xor(17)-(7,20)-WHT

k=100 tau=0.050 theta=0.33 precision=0.10 lmaxc=16.19 lsumc=17.20 lrep
    =13.37:
  (100,6.7)-sparse-(100,2)-guess(75,2)-(25,2)-code([25,1,r])-(1,2)-WHT
k=100 tau=0.100 theta=0.33 precision=0.10 lmaxc=22.14 lsumc=24.02 lrep
    =17.74:
  (100,6.9)-sparse-(100,4.3)-xor(1)-(99,6.5)-xor(3)-(96,9)-guess(77,2)
      -(19,9)-code([19,6,rnd150927])-(6,9)-drop(5)-(1,4)-WHT
k=100 tau=0.125 theta=0.33 precision=0.10 lmaxc=24.80 lsumc=27.14 lrep
    =16.87:
  (100,11.2)-sparse-(100,11.1)-xor(6)-(94,15.2)-xor(14)-(80,15.4)-xor(15)
      -(65,14.8)-guess(47,3)-(18,14.8)-code([18,8,rnd150926])-(8,14.8)-drop
      (7)-(1,7.8)-WHT
k=100 tau=0.200 theta=0.33 precision=0.10 lmaxc=29.75 lsumc=32.06 lrep
    =0.00:
  (100,29.1)-drop(1)-(99,28.1)-drop(5)-(94,23.1)-xor(22)-(72,23.2)-xor(22)
      -(50,23.4)-xor(22)-(28,23.8)-drop(6)-(22,17.8)-WHT
k=100 tau=0.250 theta=0.33 precision=0.10 lmaxc=30.75 lsumc=32.94 lrep
    =0.00:
  (100,30.1)-drop(1)-(99,29.1)-drop(5)-(94,24.1)-xor(23)-(71,24.2)-xor(23)
      -(48,24.4)-xor(23)-(25,24.8)-drop(3)-(22,21.8)-WHT
```

```
k=256 tau=0.050 theta=0.33 precision=0.10 lmaxc=28.02 lsumc=30.13 lrep
    =17.28:
  (256,8.1)-sparse-(256,4.2)-xor(1)-(255,6.3)-xor(2)-(253,9.6)-xor(8)
      -(245,10.2)-guess(178,1)-(67,10.2)-code([67,1,r])-(1,10.2)-WHT
k=256 tau=0.100 theta=0.33 precision=0.10 lmaxc=43.49 lsumc=45.99 lrep
    =27.36:
  (256,26)-sparse-(256,26)-xor(21)-(235,30)-xor(29)-(206,30)-xor(29)
      -(177,30)-xor(29)-(148,30)-guess(100,4)-(48,30)-code([11,4,S][18,5,
      rnd150927][19,6,rnd150927])-(15,30)-drop(14)-(1,16)-WHT
k=256 tau=0.125 theta=0.33 precision=0.10 lmaxc=47.35 lsumc=49.90 lrep
    =0.00:
  (256,35.6)-sparse-(256,35.6)-xor(31)-(225,39.2)-xor(38)-(187,39.4)-xor
      (38)-(149,39.8)-xor(39)-(110,39.6)-code([15,5,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926][19,7,rnd150926][19,7,
      rnd150926])-(37,39.6)-WHT
k=256 tau=0.200 theta=0.33 precision=0.10 lmaxc=53.82 lsumc=56.34 lrep
    =1.00:
  (256,42.4)-sparse-(256,42.4)-xor(38)-(218,45.8)-xor(45)-(173,45.6)-xor
      (44)-(129,46.2)-xor(45)-(84,46.4)-guess(1,1)-(83,46.4)-code([14,6,FP
      ][25,15,W][25,15,W][19,7,rnd150927])-(43,46.4)-drop(1)-(42,45.4)-WHT
k=256 tau=0.250 theta=0.33 precision=0.10 lmaxc=56.88 lsumc=59.47 lrep
    =0.00:
  (256,56.7)-drop(1)-(255,55.7)-drop(7)-(248,48.7)-xor(48)-(200,48.4)-xor
      (47)-(153,48.8)-xor(47)-(106,49.6)-xor(59)-(47,39.2)-WHT

% what follows uses precision 1

k=512 tau=0.050 theta=0.33 precision=1.00 lmaxc=47.29 lsumc=49.56 lrep
    =39.23:
  (512,10)-sparse-(512,9)-xor(2)-(510,15)-xor(14)-(496,15)-xor(14)-(482,15)
      -guess(417,2)-(65,15)-code([13,1,r][14,1,r][19,3,rnd150927][19,3,
      rnd150927])-(8,15)-drop(7)-(1,8)-WHT
k=512 tau=0.100 theta=0.33 precision=1.00 lmaxc=71.09 lsumc=73.68 lrep
    =1.60:
  (512,58)-sparse-(512,58)-xor(53)-(459,62)-xor(61)-(398,62)-xor(61)
      -(337,62)-xor(61)-(276,62)-xor(61)-(215,62)-guess(2,1)-(213,62)-code
      ([5,1,r][18,5,rnd150927][19,5,rnd150927][19,5,rnd150927][19,6,
      rnd150927][19,6,rnd150927][19,5,rnd150927][19,5,rnd150927][19,5,
      rnd150927][19,5,rnd150927][19,5,rnd150927][19,5,rnd150927])-(58,62)-
      WHT
k=512 tau=0.125 theta=0.33 precision=1.00 lmaxc=76.24 lsumc=78.97 lrep
    =0.19:
  (512,63)-sparse-(512,63)-xor(58)-(454,67)-xor(66)-(388,67)-xor(66)
      -(322,67)-xor(66)-(256,67)-xor(65)-(191,68)-guess(1,0)-(190,68)-code
      ([19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,7,rnd150926][19,7,rnd150926
      ][19,7,rnd150926][19,7,rnd150926])-(64,68)-WHT
k=512 tau=0.200 theta=0.33 precision=1.00 lmaxc=86.79 lsumc=89.28 lrep
    =2.16:
```

```
  (512,74)-sparse-(512,74)-xor(70)-(442,77)-xor(76)-(366,77)-xor(75)
      -(291,78)-xor(77)-(214,78)-xor(77)-(137,78)-guess(3,1)-(134,78)-code
      ([25,15,W][25,15,W][25,15,W][25,15,W][25,15,W][9,4,BBD])-(79,78)-drop
      (6)-(73,72)-WHT
k=512 tau=0.250 theta=0.33 precision=1.00 lmaxc=92.36 lsumc=94.85 lrep
    =1.68:
  (512,79)-sparse-(512,79)-xor(74)-(438,83)-xor(82)-(356,83)-xor(82)
      -(274,83)-xor(81)-(193,84)-xor(83)-(110,84)-guess(2,1)-(108,84)-code
      ([15,11,H][31,26,H][31,20,W][31,20,W])-(77,84)-WHT

k=768 tau=0.050 theta=0.33 precision=1.00 lmaxc=65.98 lsumc=68.15 lrep
    =58.89:
  (768,10)-sparse-(768,8)-xor(1)-(767,14)-xor(12)-(755,15)-xor(14)-(741,15)
      -guess(682,2)-(59,15)-code([7,1,r][7,1,r][19,3,rnd150927][7,1,r
      ][19,3,rnd150927])-(9,15)-drop(8)-(1,7)-WHT
k=768 tau=0.100 theta=0.33 precision=1.00 lmaxc=96.34 lsumc=99.21 lrep
    =0.76:
  (768,82)-sparse-(768,82)-xor(77)-(691,86)-xor(85)-(606,86)-xor(85)
      -(521,86)-xor(85)-(436,86)-xor(85)-(351,86)-xor(84)-(267,87)-guess
      (5,0)-(262,87)-code([15,5,rnd150927][19,6,rnd150927][19,6,rnd150927
      ][19,6,rnd150927][19,6,rnd150927][19,6,rnd150927][19,6,rnd150927
      ][19,6,rnd150927][19,6,rnd150927][19,6,rnd150927][19,6,rnd150927
      ][19,6,rnd150927][19,6,rnd150927][19,6,rnd150927])-(83,87)-WHT
k=768 tau=0.125 theta=0.33 precision=1.00 lmaxc=103.42 lsumc=106.18 lrep
    =2.44:
  (768,89)-sparse-(768,89)-xor(84)-(684,93)-xor(92)-(592,93)-xor(91)
      -(501,94)-xor(93)-(408,94)-xor(93)-(315,94)-xor(92)-(223,95)-guess
      (4,1)-(219,95)-code([25,15,W][25,15,W][18,6,rnd150926][18,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,7,rnd150926
      ][19,7,rnd150926][19,7,rnd150926][19,7,rnd150926])-(88,95)-drop(1)
      -(87,94)-WHT
k=768 tau=0.200 theta=0.33 precision=1.00 lmaxc=118.57 lsumc=121.12 lrep
    =0.00:
  (768,116)-drop(7)-(761,109)-xor(108)-(653,109)-xor(108)-(545,109)-xor
      (108)-(437,109)-xor(108)-(329,109)-xor(107)-(222,110)-xor(109)
      -(113,110)-drop(7)-(106,103)-WHT
k=768 tau=0.250 theta=0.33 precision=1.00 lmaxc=125.01 lsumc=127.63 lrep
    =2.25:
  (768,111)-sparse-(768,111)-xor(106)-(662,115)-xor(114)-(548,115)-xor(114)
      -(434,115)-xor(113)-(321,116)-xor(115)-(206,116)-guess(3,1)-(203,116)
      -code([3,1,r][25,15,W][25,15,W][25,15,W][25,15,W][25,15,W][25,15,W
      ][25,15,W][25,15,W])-(121,116)-drop(11)-(110,105)-WHT
```

3 Solving $\mathsf{LPN}_{512,0.125}$

We present here the chains for a complete recovery of the secret in the case of $\mathsf{LPN}_{512,0.125}$.

```
Step 1:
k=512 tau=0.125 theta=0.33 precision=0.10 lmaxc=76.22 lsumc=78.85:
  (512,63.3)-sparse-(512,63.3)-xor(59)-(453,66.6)-xor(65)-(388,67.2)-xor
       (66)-(322,67.4)-xor(66)-(256,67.8)-xor(67)-(189,67.6)-code([18,6,
       rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,
       rnd150926][19,6,rnd150926][19,7,rnd150926][19,7,rnd150926][19,7,
       rnd150926][19,7,rnd150926])-(64,67.6)-WHT

Step 2:
k=448 tau=0.125 theta=0.11 precision=0.10 lmaxc=69.20 lsumc=71.99:
  (448,56.6)-sparse-(448,56.6)-xor(52)-(396,60.2)-xor(59)-(337,60.4)-xor
       (59)-(278,60.8)-xor(60)-(218,60.6)-xor(59)-(159,61.2)-code([25,15,W
       ][25,15,W][14,3,rnd150926][19,4,rnd150926][19,4,rnd150926][19,6,
       rnd150926][19,6,rnd150926][19,6,rnd150926])-(59,61.2)-drop(1)
       -(58,60.2)-WHT

Step 3:
k=390 tau=0.125 theta=0.04 precision=0.10 lmaxc=62.70 lsumc=65.39:
  (390,50.3)-sparse-(390,50.3)-xor(46)-(344,53.6)-xor(52)-(292,54.2)-xor
       (53)-(239,54.4)-xor(53)-(186,54.8)-xor(54)-(132,54.6)-code([25,15,W
       ][13,4,rnd150926][18,6,rnd150926][19,6,rnd150926][19,6,rnd150926
       ][19,7,rnd150926][19,7,rnd150926])-(51,54.6)-WHT

Step 4:
k=339 tau=0.125 theta=0.01 precision=0.10 lmaxc=57.06 lsumc=59.78:
  (339,44.7)-sparse-(339,44.7)-xor(40)-(299,48.4)-xor(47)-(252,48.8)-xor
       (48)-(204,48.6)-xor(47)-(157,49.2)-xor(48)-(109,49.4)-code([3,1,r
       ][25,15,W][25,15,W][18,5,rnd150926][19,4,rnd150926][19,6,rnd150926])
       -(46,49.4)-WHT

Step 5:
k=293 tau=0.125 theta=0.00 precision=0.10 lmaxc=51.81 lsumc=54.52:
  (293,39.7)-sparse-(293,39.7)-xor(35)-(258,43.4)-xor(42)-(216,43.8)-xor
       (43)-(173,43.6)-xor(42)-(131,44.2)-xor(43)-(88,44.4)-code([25,15,W
       ][25,15,W][19,5,rnd150926][19,6,rnd150926])-(41,44.4)-WHT

Step 6:
k=252 tau=0.125 theta=0.00 precision=0.10 lmaxc=47.00 lsumc=49.61:
  (252,34.8)-sparse-(252,34.8)-xor(30)-(222,38.6)-xor(37)-(185,39.2)-xor
       (38)-(147,39.4)-xor(38)-(109,39.8)-code([14,5,rnd150926][19,6,
       rnd150926][19,6,rnd150926][19,6,rnd150926][19,7,rnd150926][19,7,
       rnd150926])-(37,39.8)-WHT

Step 7:
k=215 tau=0.125 theta=0.00 precision=0.10 lmaxc=42.36 lsumc=44.86:
  (215,30.7)-sparse-(215,30.7)-xor(26)-(189,34.4)-xor(33)-(156,34.8)-xor
       (34)-(122,34.6)-xor(33)-(89,35.2)-code([7,1,r][25,15,W][19,6,
       rnd150926][19,4,rnd150926][19,6,rnd150926])-(32,35.2)-WHT
```

```
Step 8:
k=183 tau=0.125 theta=0.00 precision=0.10 lmaxc=38.35 lsumc=40.76:
  (183,27)-sparse-(183,27)-xor(23)-(160,30)-xor(28)-(132,31)-xor(30)
      -(102,31)-xor(30)-(72,31)-code([25,15,W][13,4,rnd150926][16,5,
      rnd150926][18,5,rnd150926])-(29,31)-WHT

Step 9:
k=154 tau=0.125 theta=0.00 precision=0.10 lmaxc=34.38 lsumc=36.94:
  (154,23.5)-sparse-(154,23.5)-xor(19)-(135,27)-xor(26)-(109,27)-xor(26)
      -(83,27)-xor(25)-(58,28)-code([25,15,W][15,5,rnd150926][18,5,
      rnd150926])-(25,28)-WHT

Step 10:
k=129 tau=0.125 theta=0.00 precision=0.10 lmaxc=31.34 lsumc=33.82:
  (129,20.6)-sparse-(129,20.6)-xor(16)-(113,24.2)-xor(23)-(90,24.4)-xor(23)
      -(67,24.8)-xor(24)-(43,24.6)-code([25,15,W][18,7,rnd150926])
      -(22,24.6)-WHT

Step 11:
k=107 tau=0.125 theta=0.00 precision=0.10 lmaxc=27.85 lsumc=30.15:
  (107,27)-drop(1)-(106,26)-drop(5)-(101,21)-xor(20)-(81,21)-xor(20)
      -(61,21)-xor(20)-(41,21)-xor(19)-(22,22)-drop(2)-(20,20)-WHT

Step 12:
k=87 tau=0.125 theta=0.00 precision=0.10 lmaxc=24.54 lsumc=27.05:
  (87,24.1)-drop(1)-(86,23.1)-drop(5)-(81,18.1)-xor(17)-(64,18.2)-xor(17)
      -(47,18.4)-xor(17)-(30,18.8)-xor(17)-(13,19.6)-WHT

Step 13:
k=74 tau=0.125 theta=0.00 precision=0.10 lmaxc=23.30 lsumc=25.55:
  (74,22.9)-drop(1)-(73,21.9)-drop(5)-(68,16.9)-xor(16)-(52,16.8)-xor(15)
      -(37,17.6)-xor(21)-(16,13.2)-WHT

Step 14:
k=58 tau=0.125 theta=0.00 precision=0.10 lmaxc=19.80 lsumc=22.19:
  (58,19.8)-drop(1)-(57,18.8)-drop(5)-(52,13.8)-xor(13)-(39,13.6)-xor(12)
      -(27,14.2)-xor(13)-(14,14.4)-drop(1)-(13,13.4)-WHT

Step 15:
k=45 tau=0.125 theta=0.00 precision=0.10 lmaxc=17.29 lsumc=19.69:
  (45,17)-drop(1)-(44,16)-drop(5)-(39,11)-xor(9)-(30,12)-xor(11)-(19,12)-
      drop(1)-(18,11)-xor(8)-(10,13)-WHT

Step 16:
k=35 tau=0.125 theta=0.00 precision=0.10 lmaxc=15.97 lsumc=17.96:
  (35,15.7)-drop(1)-(34,14.7)-drop(4)-(30,10.7)-xor(10)-(20,10.4)-xor(9)
      -(11,10.8)-drop(1)-(10,9.8)-WHT
```

```
Step 17:
k=25 tau=0.125 theta=0.00 precision=0.10 lmaxc=13.30 lsumc=15.33:
  (25,12.9)-drop(1)-(24,11.9)-drop(3)-(21,8.9)-xor(8)-(13,8.8)-xor(7)
      -(6,9.6)-WHT

Step 18:
k=19 tau=0.125 theta=0.00 precision=0.10 lmaxc=12.21 lsumc=14.30:
  (19,11.9)-drop(1)-(18,10.9)-drop(3)-(15,7.9)-xor(9)-(6,5.8)-xor(1)
      -(5,9.6)-WHT

Step 19:
k=14 tau=0.125 theta=0.00 precision=0.10 lmaxc=10.93 lsumc=10.93:
  (14,4)-sparse-guess(14)
```

Overall, the complexity of going through all chains is

$$2^{78.85}+2^{71.99}+2^{65.39}+2^{59.78}+2^{54.52}+2^{49.61}+2^{44.86}+2^{40.76}+2^{36.94}$$
$$+2^{33.82}+2^{30.15}+2^{27.05}+2^{25.55}+2^{22.19}+2^{19.69}+2^{17.96}+2^{15.33}$$
$$+2^{14.30}+2^{10.93}\approx 2^{78.86}$$

with a probability of success at least $\frac{1}{2}$.

4 Random codes

We present below a list of random codes and their bc. The way we generate these codes is the following: for a $[k,k']$ code, we generate, step by step, random vectors of size k until we have k' independent ones. This will form the generator matrix G. Once we generate all the $2^{k'}$ codewords we use our formula to compute the bc. After many trials, we keep the code with the largest value for bc.

Random Codes for $\tau = 0.05$:

$[4,2]$ bc $= 0.925000000000000$	$[8,3]$ bc $= 0.830843750000002$	$[13,4]$ bc $= 0.717848281250054$
$[7,2]$ bc $= 0.820437500000001$	$[9,3]$ bc $= 0.795250000000000$	$[14,4]$ bc $= 0.686005234375007$
$[9,2]$ bc $= 0.755670312499999$	$[10,3]$ bc $= 0.764220312499999$	$[15,4]$ bc $= 0.659313910156184$
$[10,2]$ bc $= 0.717886796875003$	$[11,3]$ bc $= 0.734533281250005$	$[16,4]$ bc $= 0.635095716308526$
$[11,2]$ bc $= 0.689080273437506$	$[12,3]$ bc $= 0.707727851562494$	$[17,4]$ bc $= 0.609055215502953$
$[12,2]$ bc $= 0.658278378906265$	$[13,3]$ bc $= 0.677121250000042$	$[18,4]$ bc $= 0.583034990081217$
$[13,2]$ bc $= 0.627343367480487$	$[14,3]$ bc $= 0.644854535644540$	$[19,4]$ bc $= 0.560423084937082$
$[14,2]$ bc $= 0.605104670898428$	$[15,3]$ bc $= 0.622457954101448$	$[8,5]$ bc $= 0.912500000000003$
$[15,2]$ bc $= 0.574849437353450$	$[16,3]$ bc $= 0.592374557714663$	$[11,5]$ bc $= 0.810718749999989$
$[16,2]$ bc $= 0.550932480072046$	$[17,3]$ bc $= 0.570423946075392$	$[12,5]$ bc $= 0.780814062499967$
$[17,2]$ bc $= 0.522348102187504$	$[18,3]$ bc $= 0.542640210536539$	$[13,5]$ bc $= 0.756492578125057$
$[18,2]$ bc $= 0.503960971625559$	$[19,3]$ bc $= 0.523425509870706$	$[14,5]$ bc $= 0.723323867187460$
$[19,2]$ bc $= 0.479500628284949$	$[10,4]$ bc $= 0.799187499999992$	$[15,5]$ bc $= 0.696788447265571$
$[5,3]$ bc $= 0.925000000000000$	$[12,4]$ bc $= 0.747566406249973$	$[16,5]$ bc $= 0.670566863281211$

[17,5] bc = 0.642065001220673
[18,5] bc = 0.621900535155624
[19,5] bc = 0.596215895090632
[8,6] bc = 0.925000000000003
[9,6] bc = 0.912499999999993
[12,6] bc = 0.813390624999974
[13,6] bc = 0.785820312500064
[15,6] bc = 0.734159374999888
[16,6] bc = 0.708464716796699
[17,6] bc = 0.682572705077685
[18,6] bc = 0.656482291502392
[19,6] bc = 0.633382513669716
[9,7] bc = 0.924999999999994
[10,7] bc = 0.901249999999986
[14,7] bc = 0.789750781249873
[16,7] bc = 0.741090429687176
[17,7] bc = 0.714251874999825
[18,7] bc = 0.687121879880629
[19,7] bc = 0.664058937986979
[10,8] bc = 0.924999999999989
[11,8] bc = 0.912500000000028
[15,8] bc = 0.797351562500136
[17,8] bc = 0.746043945312908
[18,8] bc = 0.720697158201192
[19,8] bc = 0.696926562499299
[11,9] bc = 0.925000000000025
[12,9] bc = 0.912500000000054
[16,9] bc = 0.803749999999664
[18,9] bc = 0.754477929686187

Random Codes for $\tau = 0.1$:

[4,2] bc = 0.850000000000000
[7,2] bc = 0.650250000000000
[9,2] bc = 0.550462500000000
[10,2] bc = 0.501896250000001
[11,2] bc = 0.451706624999987
[12,2] bc = 0.414441562499999
[13,2] bc = 0.384841406249986
[14,2] bc = 0.347599265625008
[15,2] bc = 0.315965939062601
[16,2] bc = 0.292209785156346
[17,2] bc = 0.255932410640333
[18,2] bc = 0.235804919976357
[19,2] bc = 0.217948973378862
[5,3] bc = 0.850000000000000
[8,3] bc = 0.662249999999999
[9,3] bc = 0.620124999999997
[10,3] bc = 0.570712499999995
[11,3] bc = 0.523481249999980
[12,3] bc = 0.483963124999979
[13,3] bc = 0.439562812499995
[14,3] bc = 0.406910531250004
[15,3] bc = 0.374183078125198
[16,3] bc = 0.338270570312456
[17,3] bc = 0.305143173281109
[18,3] bc = 0.282576971953005
[19,3] bc = 0.257462926757663
[10,4] bc = 0.620124999999991
[12,4] bc = 0.533606249999987
[13,4] bc = 0.498295625000009
[14,4] bc = 0.456234062499976
[15,4] bc = 0.423086156250238
[16,4] bc = 0.388249140624795
[17,4] bc = 0.357040026562527
[18,4] bc = 0.326219003906170
[19,4] bc = 0.298997873515392
[8,5] bc = 0.805000000000002
[11,5] bc = 0.647124999999997
[12,5] bc = 0.602012500000006
[13,5] bc = 0.557631250000012
[14,5] bc = 0.514058124999978
[15,5] bc = 0.477960312500255
[16,5] bc = 0.432172781249713
[17,5] bc = 0.404812453125008
[18,5] bc = 0.370472807812691
[19,5] bc = 0.342473907030773
[8,6] bc = 0.850000000000003
[9,6] bc = 0.785000000000002
[12,6] bc = 0.657625000000007
[13,6] bc = 0.604262500000019
[15,6] bc = 0.523140625000133
[16,6] bc = 0.483241562499645
[17,6] bc = 0.449978906249989
[18,6] bc = 0.415639015625284
[19,6] bc = 0.385346414062386
[9,7] bc = 0.850000000000006
[10,7] bc = 0.804999999999990
[14,7] bc = 0.610312500000011
[16,7] bc = 0.535203124999684
[17,7] bc = 0.492602812500183
[18,7] bc = 0.459242031250249
[19,7] bc = 0.427382828126207
[10,8] bc = 0.849999999999990
[11,8] bc = 0.824999999999974
[15,8] bc = 0.624562499999774
[17,8] bc = 0.543265625000525
[18,8] bc = 0.505529062500634
[19,8] bc = 0.466630156251362
[11,9] bc = 0.849999999999978
[12,9] bc = 0.825000000000046
[16,9] bc = 0.625812499999804

Random Codes for $\tau = 0.125$:

[4,2] bc = 0.812500000000000
[7,2] bc = 0.590820312500000
[9,2] bc = 0.469238281250000
[10,2] bc = 0.410583496093750
[11,2] bc = 0.363045692443848
[12,2] bc = 0.331746578216553
[13,2] bc = 0.295313835144043
[14,2] bc = 0.262624084949493
[15,2] bc = 0.229796074330807
[16,2] bc = 0.199777818284929
[17,2] bc = 0.178411910077557
[18,2] bc = 0.161525560077280
[19,2] bc = 0.142899490048876
[5,3] bc = 0.812500000000000
[8,3] bc = 0.596679687500000
[9,3] bc = 0.538574218750000
[10,3] bc = 0.484985351562500
[11,3] bc = 0.438781738281250
[12,3] bc = 0.395263671875000
[13,3] bc = 0.344362258911133
[14,3] bc = 0.318330764770508
[15,3] bc = 0.280164837837219
[16,3] bc = 0.250766009092331
[17,3] bc = 0.222185984253883
[18,3] bc = 0.201188247650862
[19,3] bc = 0.173051953781396
[10,4] bc = 0.556884765625000
[12,4] bc = 0.459289550781250
[13,4] bc = 0.416761398315430
[14,4] bc = 0.364337921142578
[15,4] bc = 0.324589252471924
[16,4] bc = 0.295842528343201
[17,4] bc = 0.266516566276550
[18,4] bc = 0.238161936402321
[19,4] bc = 0.215231126174331
[8,5] bc = 0.734375000000000
[11,5] bc = 0.568603515625000
[12,5] bc = 0.515930175781250
[13,5] bc = 0.467575073242188
[14,5] bc = 0.427307128906250
[15,5] bc = 0.384399414062500
[16,5] bc = 0.345722198486328
[17,5] bc = 0.308802604675293
[18,5] bc = 0.278767108917236
[19,5] bc = 0.249812759459019
[8,6] bc = 0.812500000000000
[9,6] bc = 0.757812500000000
[12,6] bc = 0.584716796875000

[13,6] bc = 0.533325195312500
[15,6] bc = 0.436737060546875
[16,6] bc = 0.391594886779785
[17,6] bc = 0.354474067687988
[18,6] bc = 0.323782920837402
[19,6] bc = 0.291754990816116
[9,7] bc = 0.812500000000000
[10,7] bc = 0.757812500000000
[14,7] bc = 0.539184570312500
[16,7] bc = 0.451797485351562
[17,7] bc = 0.404412269592285
[18,7] bc = 0.368431568145752
[19,7] bc = 0.336303114891052
[10,8] bc = 0.812500000000000
[11,8] bc = 0.781250000000000
[15,8] bc = 0.542846679687500
[17,8] bc = 0.452558517456055
[18,8] bc = 0.415075302124023
[19,8] bc = 0.374835968017578
[11,9] bc = 0.812500000000000
[12,9] bc = 0.757812500000000
[16,9] bc = 0.547607421875000
[18,9] bc = 0.462966918945312

Random Codes for $\tau = 0.2$:

[4,2] bc = 0.700000000000000
[7,2] bc = 0.405500000000000
[9,2] bc = 0.272749999999999
[10,2] bc = 0.221980000000003
[11,2] bc = 0.177584000000003
[12,2] bc = 0.154187500000002
[13,2] bc = 0.126058437500001
[14,2] bc = 0.102578124999976
[15,2] bc = 0.0806773999999974
[16,2] bc = 0.0672851875000390
[17,2] bc = 0.0541784750000697
[18,2] bc = 0.0430625200001377
[19,2] bc = 0.0349497860001292
[5,3] bc = 0.700000000000000
[8,3] bc = 0.405499999999999
[9,3] bc = 0.343000000000001
[10,3] bc = 0.291275000000004
[11,3] bc = 0.240512500000004
[12,3] bc = 0.197742500000001
[13,3] bc = 0.156951999999985
[14,3] bc = 0.137524624999980
[15,3] bc = 0.111005200000025
[16,3] bc = 0.0928898000000887
[17,3] bc = 0.0767771256251488
[18,3] bc = 0.0609095375002112
[19,3] bc = 0.0505532889999707
[10,4] bc = 0.364750000000007
[12,4] bc = 0.258325000000002
[13,4] bc = 0.213679999999971
[14,4] bc = 0.176458750000002
[15,4] bc = 0.147204875000059
[16,4] bc = 0.121787406250114
[17,4] bc = 0.102391512500167
[18,4] bc = 0.0842524562500867
[19,4] bc = 0.0690329674999138
[8,5] bc = 0.589999999999999
[11,5] bc = 0.369250000000010
[12,5] bc = 0.318649999999986
[13,5] bc = 0.268149999999959
[14,5] bc = 0.225544999999992
[15,5] bc = 0.188910625000082
[16,5] bc = 0.157532562500143
[17,5] bc = 0.130860062500169
[18,5] bc = 0.109656978125099
[19,5] bc = 0.0912561109374092
[8,6] bc = 0.699999999999998
[9,6] bc = 0.650000000000002
[12,6] bc = 0.379749999999974
[13,6] bc = 0.328624999999967
[15,6] bc = 0.235025000000079
[16,6] bc = 0.197588125000182
[17,6] bc = 0.169437500000210
[18,6] bc = 0.139357843750180
[19,6] bc = 0.119218062500138
[9,7] bc = 0.700000000000003
[10,7] bc = 0.620000000000010
[14,7] bc = 0.343999999999986
[16,7] bc = 0.245225000000221
[17,7] bc = 0.207013750000245
[18,7] bc = 0.175233125000239
[19,7] bc = 0.151714656250229
[10,8] bc = 0.700000000000010
[11,8] bc = 0.619999999999996
[15,8] bc = 0.356000000000120
[17,8] bc = 0.255875000000282
[18,8] bc = 0.214251250000277
[19,8] bc = 0.185132187500286
[11,9] bc = 0.699999999999990
[12,9] bc = 0.649999999999956
[16,9] bc = 0.366875000000244

Random Codes for $\tau = 0.25$:

[4,2] bc = 0.625000000000000
[7,2] bc = 0.292968750000000
[9,2] bc = 0.184570312500000
[10,2] bc = 0.143066406250000
[11,2] bc = 0.107299804687500
[12,2] bc = 0.0848999023437500
[13,2] bc = 0.0620727539062500
[14,2] bc = 0.0496444702148438
[15,2] bc = 0.0366325378417969
[16,2] bc = 0.0274744033813477
[17,2] bc = 0.0209437608718872
[18,2] bc = 0.0164231657981873
[19,2] bc = 0.0120856314897537
[5,3] bc = 0.625000000000000
[8,3] bc = 0.320312500000000
[9,3] bc = 0.250000000000000
[10,3] bc = 0.196289062500000
[11,3] bc = 0.150146484375000
[12,3] bc = 0.121704101562500
[13,3] bc = 0.0912780761718750
[14,3] bc = 0.0709533691406250
[15,3] bc = 0.0566749572753906
[16,3] bc = 0.0420742034912109
[17,3] bc = 0.0335798263549805
[18,3] bc = 0.0254254341125488
[19,3] bc = 0.0197929143905640
[10,4] bc = 0.263671875000000
[12,4] bc = 0.168457031250000
[13,4] bc = 0.129882812500000
[14,4] bc = 0.101013183593750
[15,4] bc = 0.0771789550781250
[16,4] bc = 0.0606994628906250
[17,4] bc = 0.0467262268066406
[18,4] bc = 0.0371999740600586
[19,4] bc = 0.0295500755310059
[8,5] bc = 0.531250000000000
[11,5] bc = 0.275390625000000
[12,5] bc = 0.218750000000000
[13,5] bc = 0.170166015625000
[14,5] bc = 0.138671875000000
[15,5] bc = 0.111480712890625
[16,5] bc = 0.0867004394531250
[17,5] bc = 0.0682830810546875
[18,5] bc = 0.0538368225097656
[19,5] bc = 0.0424060821533203
[8,6] bc = 0.625000000000000
[9,6] bc = 0.500000000000000
[12,6] bc = 0.292968750000000
[13,6] bc = 0.230468750000000
[15,6] bc = 0.147705078125000
[16,6] bc = 0.118072509765625
[17,6] bc = 0.0952911376953125
[18,6] bc = 0.0742111206054688
[19,6] bc = 0.0585670471191406

[9,7] bc = 0.625000000000000
[10,7] bc = 0.531250000000000
[14,7] bc = 0.240234375000000
[16,7] bc = 0.154541015625000
[17,7] bc = 0.119049072265625
[18,7] bc = 0.0965576171875000
[19,7] bc = 0.0788497924804688
[10,8] bc = 0.625000000000000
[11,8] bc = 0.562500000000000
[15,8] bc = 0.251953125000000
[17,8] bc = 0.162719726562500
[18,8] bc = 0.134338378906250
[19,8] bc = 0.106536865234375
[11,9] bc = 0.625000000000000
[12,9] bc = 0.531250000000000
[16,9] bc = 0.255859375000000
[18,9] bc = 0.171264648437500

The random codes used for $\mathsf{LPN}_{512,0.125}$ have the following generator matrices:

```
[18,6]
G is:
(0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1)
(0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0)
(0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0)
(1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0)
(0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1)
(0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1)


[19,7]
G is:
(0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0)
(1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0)
(0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0)
(0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0)
(0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1)
(0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1)
(0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1)

[19,6]
G is:
(0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)
(0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1)
(1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1)
(1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0)
(1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0)
(1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0)
```

5 Heuristic approximation

In our work, we make the following assumption on the failure probability:

$$E\left(1-(1-\varphi(Z(s)))^{2^{k'}-1}\right) \approx 1-(1-\varphi(E(Z(s))))^{2^{k'}-1},$$

where s is the secret, of k' bits, we recover with the Walsh Hadamard Transform (WHT), $\varphi(x) = \frac{1}{2} + \frac{1}{2}\mathsf{erf}(\frac{x}{\sqrt{2}})$ and erf is the Gauss error function, $Z(s) = -\frac{\delta'(s)}{\sqrt{2-\delta'(s)^2}}\sqrt{n'}$. So, we consider that the average probability of error is very

close to the probability of error, i.e. $E(p(s)) \approx p$, where $p = 1-(1-\varphi(Z))^{2^{k'}-1}$. and $Z = -\frac{\delta'}{\sqrt{2-\delta'^2}}\sqrt{n'}$. The values n' and δ' are computed as from our formulae for the biases.

In order to validate this assumption, we compare our theory with what we observe in practice. For this, we take several chains that contain a code reduction (as the bias introduced by this operation depends on the secret s). The following chains were analysed:

```
k=48, tau=0.005, theta=0.33:
(48,5.8)-sparse-(48,2.9)-xor(1)-(47,3.6)-xor(1)-(46,5.1)-xor(2)-(44,7.2)-
    code-(1,7.2)-WHT
with the codes [44,1,r]

k=48, tau=0.05, theta=0.33:
(48,6.2100)-sparse-(48,4.70)-xor( 1)-(47,7.35)-xor( 1)-(46,12.69)-code
    -(24,12.69)-xor(14)-(10,10.37)-WHT
with the codes [23,12,G][23,12,G]

k=64, tau=0.05, theta=0.28:
(64,11.30)-sparse-(64,11.3)-code-(10,11.3)-WHT
with the codes [5,1,r][5,1,r][5,1,r][7,1,r][7,1,r][7,1,r][7,1,r][7,1,r
    ][7,1,r][7,1,r]

k=64, tau=0.1, theta=0.32:
(64,8.88)-sparse-(64,8.68)- xor( 4)-(60,12.35)-xor(11)-(49,12.70)- xor(11)
    -(38,13.40)-code-(23,13.40)-WHT
with the codes [15,11,H][23,12,G]

k=64, tau=0.1, theta = 0.33:
(64,9.3)-sparse-(64,9.1)-xor(4)-(60,13.2)-xor(12)-(48,13.4)-xor(12)
    -(36,13.8)-code-(1,13.8)-WHT
with the codes [36,1,r]

k=256, tau=0.2, theta=0.12:
(256,41.94)-sparse-(256,41.94)-xor(37)-(219,45.88)-xor(45)-(174,45.76)-xor
    (45)-(129,45.52)-xor(44)-(85,46.04)-code-(45,46.04)-WHT
with the codes [2,1,r][7,4,H][7,4,H][23,12,G][23,12,G][23,12,G]
```

We obtain the following results:
$\mathsf{LPN}_{48,0.05}$ with the codes $[44,1,r]$

- $E(p(s)) = 0.4231$
- $1-(1-\varphi(E(Z(s))))^{2^{k'}-1} = 0.3220$
- $1-(1-\varphi(Z))^{2^{k'}-1} = 0.310893249450781$
- $E(\delta(s)) = 0.0514$
- $\delta = 0.0573009521884174$
- $E(Z(s)) = -0.461981714958069$

– $Z = -0.493326542156718$

$\mathsf{LPN}_{48,0.05}$ with the codes $[23,12,G][23,12,G]$

– $E(p(s)) = 0.4780$
– $1-(1-\varphi(E(Z(s))))^{2^{k'}-1} = 0.0318$
– $1-(1-\varphi(Z))^{2^{k'}-1} = 0.334275183642941$
– $E(\delta(s)) = 0.1298$
– $\delta = 0.129822931628186$
– $E(Z(s)) = -4.00049008907774$
– $Z = -3.35442479599458$

$\mathsf{LPN}_{64,0.05}$ with the codes $[5,1,r][5,1,r][5,1,r][7,1,r][7,1,r][7,1,r][7,1,r][7,1,r][7,1,r][7,1,r]$

– $E(p(s)) = 0.8670$
– $1-(1-\varphi(E(Z(s))))^{2^{k'}-1} = 0.074$
– $1-(1-\varphi(Z))^{2^{k'}-1} = 0.282948298443234$
– $E(\delta(s)) = 0.10654$
– $\delta = 0.09580$
– $E(Z(s)) = -3.79016746911067$
– $Z = -3.40978448268245$

$\mathsf{LPN}_{64,0.1}$ with the codes $[15,11,H][23,12,G]$

– $E(p(s)) = 0.5353$
– $1-(1-\varphi(E(Z(s))))^{2^{k'}-1} = 0.3274$
– $1-(1-\varphi(Z))^{2^{k'}-1} = 0.335578150280515$
– $E(\delta(s)) = 0.0724$
– $\delta = 0.0724220723200000$
– $E(Z(s)) = -5.33685553225594$
– $Z = -5.33137227108397$

$\mathsf{LPN}_{64,0.1}$ with the codes $[36,1,r]$

– $E(p(s)) = 0.4461$
– $1-(1-\varphi(E(Z(s))))^{2^{k'}-1} = 0.3159$
– $1-(1-\varphi(Z))^{2^{k'}-1} = 0.316793578967859$
– $E(\delta(s)) = 0.0056$
– $\delta = 0.00564445020331737$

- $E(Z(s)) = -0.478971239492193$
- $Z = -0.476684216638090$

$\mathsf{LPN}_{256,0.2}$ with the codes $[2,1,r][7,4,H][7,4,H][23,12,G][23,12,G][23,12,G]$

- $E(p(s)) = 0.7609$
- $1-(1-\varphi(E(Z(s))))^{2^{k'}-1} = 0.1209$
- $1-(1-\varphi(Z))^{2^{k'}-1} = 0.120943626782345$
- $E(\delta(s)) = 1.2933e-6$
- $\delta = 1.2933e-6$
- $E(Z(s)) = -7.77878914531831$
- $Z = -7.77878914490038$

From this we conclude that the approximations $E(\delta(s)) \approx \delta$ and $E(Z(s)) \approx Z$ are correct but that there is a small gap between $E(p(s))$ and p. The function $x \mapsto 1-(1-\varphi(x))^{2^{k'}-1}$ is convex in the region $x \leq 0$ so we rather have $E(p(s)) \geq 1-(1-\varphi(E(Z(s))))^{2^{k'}-1}$.

6 Other Reduction Steps

In this section, we present other reduction steps which have not been useful to find optimal chains.

Reduction step from BKW *[1].* *partition-reduce*(b) is used by the BKW algorithm. Recall that the queries received from the oracle are of the form $(v, \langle v,s\rangle \oplus d)$. In this reduction, the v vectors are sorted in equivalence classes according to their values on a block of b bits. These b positions are chosen randomly. Two queries in the same equivalence class have the same values on the b positions. In each equivalence class, we choose a representative vector and xor it with the rest of the queries. This will give vectors with only 0 on this window of b bits. Afterwards the representative vector is dropped. This operation reduces the secret to $k-b$ effective bits (since b bits of the secret are always xored with zero). The new bias is δ^2 as the new queries are xor of two initial ones and the number of queries is reduced by 2^b (as there are 2^b equivalence classes).

partition-reduce$(b) : k' = k-b;\ n' = n-2^b;\ \delta' = \delta^2;\ \delta'_s = \delta_s$ Complexity: $O(kn)$

The *xor-reduce*(b) reduction is always better than *partition-reduce*(b).

The *partition-reduce*(b) edge is $(k, \log_2 n) \to (k-b, \mathsf{Round}_L(\log_2(n-2^b))$ with label $\alpha = 2, \beta = 0$.

In the automaton defining valid chains, *partition-reduce* appears exactly at the same place as *xor-reduce*.

Reduction step $\mathsf{LF}(4)$ *from [4].* The $\mathsf{LF}(4)$ reduction tries to reduce the size of the secret by b bits in the following manner: given the n queries, it creates a new list of queries by computing the XOR of all the possible pairs. This new list has a size of $\frac{n(n-1)}{2}$. In this new list it looks for all collisions, i.e. see if there are two queries that have the same value on the b-bit window. The new queries are obtained by the xor of 4 old queries.

$$\mathsf{LF}(4)(b) : k' = k-b;\ n' = \binom{n}{4}2^{-b};\ \delta' = \delta^4;\ \delta'_s = \delta_s$$
$$\text{Complexity: } O\left(k\tfrac{n(n-1)}{2} + kn'\right)$$

The $\mathsf{LF}(4)(b)$ edge is $(k, \log_2 n) \to (k-b, \mathsf{Round}_L(\log_2(\binom{n}{4}2^{-b}))$ with label $\alpha = 4, \beta = 0$.

In the automaton defining valid chains, $\mathsf{LF}(4)(b)$ appears exactly at the same place as *xor-reduce*.

Reduction step *trunc-reduce.* Given the queries $(v, \langle v, s\rangle \oplus d)$, one can just pick an arbitrary position, j, and reduce the secret by 1 bit if the j^{th} bit of v is 0. If the j^{th} bit is 1, then we can still reduce the size by 1 by adding s_j to the error term; i.e. the updated query is $(v', \langle v', s'\rangle \oplus d \oplus s_j)$, where $s'(v')$ is $s($ respectively $v)$ vector without the j^{th} bit. When the bit of v is 0 this change does not affect the original query. When it is 1, then the new bias introduced is δ_s, the bias of s_j. We can take $\delta_s = 1 - 2\tau$. Thus, the expected bias is $\frac{1+\delta_s}{2}$. We can generalize this reduction and zero b bits of the secret by introducing an error term that has a bias of $(\frac{1+\delta_s}{2})^b$. One point is that we must make the secret sparse (thus use *sparse-secret*) and also randomize the b bits which are truncated so that they do have a bias of δ_s and are not constant. We do this by a special treatment during the *sparse-secret* step: for each output equation we use b new input equations and use these errors bits as the randomized b secret bits.

trunc-reduce(b) introduces b bits of the secret in the error term. We can easily see that, by this operation, the number of queries remains the same, the size of the secret is reduced by b bits and the new bias is $\delta \cdot (\frac{1+\delta_s}{2})^b$.

As we do not follow the evolution of δ_s beyond *code-reduce*, the *trunc-reduce* step can only be done in between *sparse-secret* and *code-reduce* in a chain. When considering *trunc-reduce*(b) outside this case, we under-estimate δ_s to 0 and have $\delta' = \delta \cdot 2^{-b}$. As for *guess-secret*, the *trunc-reduce* step requires a transition from state 3 to itself in the automaton.

$$\textit{trunc-reduce}(b) : k' = k-b;\ n' = n;\ \delta' = \delta \cdot (\tfrac{1+\delta_s}{2})^b;\ \delta'_s = \delta_s$$
$$\text{Complexity: } O(1)$$

The *trunc-reduce*(b) edge is $(k, \log_2 n) \to (k-b, \log_2 n)$ with label $\alpha = 1, \beta = b\log_2 \frac{1+\delta_s}{2}$.

In the automaton defining valid chains, *trunc-reduce*(b) appears exactly at the same place as *guess-secret*.

Solving algorithm majority. *majority* is the solving algorithm which is used by the BKW algorithm. For this solving algorithm we have queries of the form $(1, s_i \oplus d_j)$, where s_i is the i^{th} bit of the secret and $d_j \leftarrow \mathsf{Ber}_{(1-\delta')/2}$. For this, from the n' queries, we chose only those that have Hamming weight 1. Given that the probability for a noise bit to be set on 1 is smaller than $\frac{1}{2}$, most of the queries will be $(1, s_i)$. The majority of the $\frac{n'}{2^{k'}}$ queries decide what is the value of s_i. The number of queries needed to make the correct guess is given by the Chernoff bounds. According to [2], the *majority* needs $n' = 2\ln\left(\frac{k'}{\theta}\right) \cdot \delta'^{-2} 2^{k'}$ queries in order to bound the failure probability of guessing wrong any of k' bits by θ, with $0 < \theta < 1$. The complexity of *majority* is $O(n')$. The WHT algorithm is always better than *majority*.

7 Chains from [3] and [4]

In this section we present the analysis on the results from ASIACRYPT'14 [3] and EUROCRYPT'16 [4]. We compute complexities and biases following our formulas, and always use the largest pool of codes that we had to have the most favorable results. Sometimes, the number of initial queries is either too small or too large to have a failure probability close to 33% so we correct it to be able to compare chains. Together with these chains, we provide the best ones that we found.

In the data below, `lcomp` denotes the logarithmic complexity in each step of the chain, `lbc2` denotes the logarithmic value of δ_s^2, `lr_all` denotes the logarithmic number of necessary average of repetitions of the overall chain for guessing to succeed, `lr_wht` denotes the logarithmic number of repetitions which apply to WHT only, `lmaxcomp` denotes the logarithmic max-complexity, `ltotcomp` denotes the logarithmic total complexity of the chain. The name of the reduction steps differ a bit: `sparse` denotes *sparse-secret*, `part` denotes *partition-reduce*, `xor` denotes *xor-reduce*, `guess` denotes *guess-secret*, and `code` denotes *code-reduce*. For each chain, we put the computations for the exact chain instead of the rounded one. Each reduction step is followed by the indication of the vertex $(k, \log_2 n)$.

7.1 Chains for $\mathsf{LPN}_{512,1/8}$

Here is the chain proposed in [3] (proceedings) with a claimed complexity of $2^{79.9}$:

```
chain for LPN_{512,0.125} with n=2^66.3000
      sparse-(512,66.30) lcomp=79.245 lbc2=-0.830
     part(63)-(449,66.15) lcomp=75.300 lbc2=-1.660
     part(63)-(386,65.97) lcomp=74.956 lbc2=-3.320
     part(63)-(323,65.78) lcomp=74.565 lbc2=-6.641
     part(63)-(260,65.55) lcomp=74.111 lbc2=-13.281
     part(63)-(197,65.28) lcomp=73.571 lbc2=-26.562
     part(63)-(134,64.94) lcomp=72.900 lbc2=-53.125
  guess(10u2)-(124,64.94) lcomp=0.000 lbc2=-53.125
         code-(64,64.94) lcomp=71.899 lbc2=-65.809
       WHT(theta=100.00) lcomp=75.125
          lr_all=0.184 lr_wht=5.807
                    lmaxcomp=81.12
                    ltotcomp=81.58
  for the used code, log_2 bc^2 =-12.684598566058570684917951628899665523
  used code family: old+QP+rnd
  params=[5,1,r][25,15,W][25,15,W][25,12,FP][25,15,W][19,6,rnd150926]
```

Here is the corrected one to reach $\theta = 33\%$:

```
chain for LPN_{512,0.125} with n=2^73.2200
      sparse-(512,73.22) lcomp=85.769 lbc2=-0.830
     part(63)-(449,73.22) lcomp=82.220 lbc2=-1.660
     part(63)-(386,73.22) lcomp=82.029 lbc2=-3.320
     part(63)-(323,73.22) lcomp=81.810 lbc2=-6.641
     part(63)-(260,73.22) lcomp=81.552 lbc2=-13.281
     part(63)-(197,73.21) lcomp=81.238 lbc2=-26.562
     part(63)-(134,73.21) lcomp=80.836 lbc2=-53.125
  guess(10u2)-(124,73.21) lcomp=0.000 lbc2=-53.125
         code-(64,73.21) lcomp=80.167 lbc2=-65.809
        WHT(theta=28.00) lcomp=79.300
          lr_all=0.184 lr_wht=5.807
                    lmaxcomp=85.95
                    ltotcomp=86.96
  for the used code, log_2 bc^2 =-12.684598566058570684917951628899665523
  used code family: old+QP+rnd
  params=[5,1,r][25,15,W][25,15,W][25,12,FP][25,15,W][19,6,rnd150926]
```

Here is the chain presented by [3] (presented at the conference) with a claimed complexity of $2^{79.7}$:

```
chain for LPN_{512,0.125} with n=2^63.7000
      sparse-(512,63.70) lcomp=76.467 lbc2=-0.830
     xor(62)-(450,64.40) lcomp=73.400 lbc2=-1.660
     xor(62)-(388,65.80) lcomp=74.614 lbc2=-3.320
     xor(62)-(326,68.60) lcomp=77.200 lbc2=-6.641
     xor(62)-(264,74.20) lcomp=82.549 lbc2=-13.281
     xor(62)-(202,85.40) lcomp=93.444 lbc2=-26.562
  guess(22u2)-(180,85.40) lcomp=0.000 lbc2=-26.562
         code-(60,85.40) lcomp=92.892 lbc2=-58.986
         WHT(theta=0.00) lcomp=91.307
```

```
         lr_all=1.091 lr_wht=7.989
                  lmaxcomp=100.39
                  ltotcomp=100.43
  for the used code, log_2 bc^2 =-32.42392571159892747060626060259404448 1
  used code family: old+QP+rnd
  params=[3,1,r][25,15,W][19,4,rnd150926][19,4,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926]
```

Here is the corrected one to reach $\theta = 33\%$:

```
chain for LPN_{512,0.125} with n=2^63.1030
      sparse-(512,63.10) lcomp=76.101 lbc2=-0.830
     xor(62)-(450,63.21) lcomp=72.206 lbc2=-1.660
     xor(62)-(388,63.41) lcomp=72.226 lbc2=-3.320
     xor(62)-(326,63.82) lcomp=72.424 lbc2=-6.641
     xor(62)-(264,64.65) lcomp=72.997 lbc2=-13.281
     xor(62)-(202,66.30) lcomp=74.340 lbc2=-26.562
  guess(22u2)-(180,66.30) lcomp=0.000 lbc2=-26.562
        code-(60,66.30) lcomp=73.788 lbc2=-58.986
       WHT(theta=26.00) lcomp=72.717
         lr_all=1.091 lr_wht=7.989
                  lmaxcomp=81.80
                  ltotcomp=81.90
  for the used code, log_2 bc^2 =-32.42392571159892747060626060259404448 1
  used code family: old+QP+rnd
  params=[3,1,r][25,15,W][19,4,rnd150926][19,4,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926]
```

Here is the chain from [4] based on LF1 with a claimed complexity of $2^{75.897}$:

```
chain for LPN_{512,0.125} with n=2^71.2910
    drop( 5)-(507,66.29) lcomp=72.245 lbc2=-0.830
      sparse-(507,66.29) lcomp=79.225 lbc2=-0.830
    part(63)-(444,66.14) lcomp=75.277 lbc2=-1.660
    part(63)-(381,65.96) lcomp=74.930 lbc2=-3.320
    part(63)-(318,65.76) lcomp=74.535 lbc2=-6.641
    part(63)-(255,65.53) lcomp=74.076 lbc2=-13.281
    part(63)-(192,65.26) lcomp=73.527 lbc2=-26.562
  guess(20u1)-(172,65.26) lcomp=0.000 lbc2=-26.562
        code-(62,65.26) lcomp=72.686 lbc2=-55.531
        WHT(theta=0.00) lcomp=73.371
         lr_all=1.905 lr_wht=4.392
                  lmaxcomp=81.13
                  ltotcomp=81.79
  for the used code, log_2 bc^2 =-28.96829063258713082771780760278761345 4
  used code family: old+QP+rnd
  params=[25,15,W][14,5,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,
      rnd150926]
```

Here is the corrected one to reach $\theta = 33\%$:

```
chain for LPN_{512,0.125} with n=2^70.5660
    drop( 5)-(507,65.57) lcomp=71.520 lbc2=-0.830
      sparse-(507,65.57) lcomp=78.383 lbc2=-0.830
    part(63)-(444,65.30) lcomp=74.552 lbc2=-1.660
    part(63)-(381,64.97) lcomp=74.094 lbc2=-3.320
    part(63)-(318,64.55) lcomp=73.545 lbc2=-6.641
    part(63)-(255,63.94) lcomp=72.860 lbc2=-13.281
    part(63)-(192,62.88) lcomp=71.937 lbc2=-26.562
  guess(20u1)-(172,62.88) lcomp=0.000 lbc2=-26.562
        code-(62,62.88) lcomp=70.309 lbc2=-55.531
       WHT(theta=31.00) lcomp=73.032
         lr_all=1.905 lr_wht=4.392
                  lmaxcomp=80.29
                  ltotcomp=81.07
  for the used code, log_2 bc^2 =-28.968290632587130827717807602787613454
  used code family: old+QP+rnd
  params=[25,15,W][14,5,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,
      rnd150926]
```

For comparison, here is the same computation using only perfect codes:

```
chain for LPN_{512,0.125} with n=2^70.8630
    drop( 5)-(507,65.86) lcomp=71.817 lbc2=-0.830
      sparse-(507,65.86) lcomp=78.578 lbc2=-0.830
    part(63)-(444,65.65) lcomp=74.849 lbc2=-1.660
    part(63)-(381,65.40) lcomp=74.444 lbc2=-3.320
    part(63)-(318,65.10) lcomp=73.973 lbc2=-6.641
    part(63)-(255,64.71) lcomp=73.409 lbc2=-13.281
    part(63)-(192,64.19) lcomp=72.706 lbc2=-26.562
  guess(20u1)-(172,64.19) lcomp=0.000 lbc2=-26.562
        code-(62,64.19) lcomp=71.612 lbc2=-56.834
       WHT(theta=31.00) lcomp=73.169
         lr_all=1.905 lr_wht=4.392
                  lmaxcomp=80.48
                  ltotcomp=81.27
  for the used code, log_2 bc^2 =-30.272072056802635203237971232374080351
  used code family: old
  params=[5,1,r][5,1,r][5,1,r][5,1,r][5,1,r][5,1,r][5,1,r][5,1,r][5,1,r
      ][7,1,r][7,1,r][7,1,r][7,1,r][7,1,r][23,12,G][23,12,G][23,12,G
      ][23,12,G]
```

Here is the chain from [4] based on LF2 with a claimed complexity of $2^{74.732}$:

```
chain for LPN_{512,0.125} with n=2^69.9870
    drop( 5)-(507,64.99) lcomp=70.941 lbc2=-0.830
      sparse-(507,64.99) lcomp=78.045 lbc2=-0.830
     xor(64)-(443,64.97) lcomp=73.973 lbc2=-1.660
     xor(64)-(379,64.95) lcomp=73.765 lbc2=-3.320
```

```
    xor(64)-(315,64.90) lcomp=73.514 lbc2=-6.641
    xor(64)-(251,64.79) lcomp=73.195 lbc2=-13.281
    xor(64)-(187,64.58) lcomp=72.764 lbc2=-26.562
 guess(17u1)-(170,64.58) lcomp=0.000 lbc2=-26.562
       code-(62,64.58) lcomp=71.993 lbc2=-54.892
       WHT(theta=0.00) lcomp=73.232
        lr_all=1.497 lr_wht=4.170
                 lmaxcomp=79.54
                 ltotcomp=80.45
 for the used code, log_2 bc^2 =-28.32984537421874253423830411393851492 8
 used code family: old+QP+rnd
 params=[25,15,W][13,4,rnd150926][18,6,rnd150926][19,6,rnd150926][19,6,
     rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,7,
     rnd150926]
```

Here is the corrected one to reach $\theta = 33\%$:

```
chain for LPN_{512,0.125} with n=2^69.9140
    drop( 5)-(507,64.91) lcomp=70.868 lbc2=-0.830
     sparse-(507,64.91) lcomp=77.605 lbc2=-0.830
    xor(64)-(443,64.83) lcomp=73.900 lbc2=-1.660
    xor(64)-(379,64.66) lcomp=73.619 lbc2=-3.320
    xor(64)-(315,64.31) lcomp=73.222 lbc2=-6.641
    xor(64)-(251,63.62) lcomp=72.611 lbc2=-13.281
    xor(64)-(187,62.25) lcomp=71.596 lbc2=-26.562
 guess(17u1)-(170,62.25) lcomp=0.000 lbc2=-26.562
       code-(62,62.25) lcomp=69.657 lbc2=-54.892
      WHT(theta=28.00) lcomp=72.990
        lr_all=1.497 lr_wht=4.170
                 lmaxcomp=79.10
                 ltotcomp=80.09
 for the used code, log_2 bc^2 =-28.32984537421874253423830411393851492 8
 used code family: old+QP+rnd
 params=[25,15,W][13,4,rnd150926][18,6,rnd150926][19,6,rnd150926][19,6,
     rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,7,
     rnd150926]
```

For comparison, here is the same computation using only perfect codes:

```
chain for LPN_{512,0.125} with n=2^69.9540
    drop( 5)-(507,64.95) lcomp=70.908 lbc2=-0.830
     sparse-(507,64.95) lcomp=78.027 lbc2=-0.830
    xor(64)-(443,64.91) lcomp=73.940 lbc2=-1.660
    xor(64)-(379,64.82) lcomp=73.699 lbc2=-3.320
    xor(64)-(315,64.63) lcomp=73.382 lbc2=-6.641
    xor(64)-(251,64.26) lcomp=72.931 lbc2=-13.281
    xor(64)-(187,63.53) lcomp=72.236 lbc2=-26.562
 guess(17u1)-(170,63.53) lcomp=0.000 lbc2=-26.562
       code-(62,63.53) lcomp=70.937 lbc2=-56.158
      WHT(theta=20.00) lcomp=73.090
        lr_all=1.497 lr_wht=4.170
                 lmaxcomp=79.52
```

```
                  ltotcomp=80.37
  for the used code, log_2 bc^2 =-29.595714087454877581434997641085959931
  used code family: old
  params=[5,1,r][5,1,r][5,1,r][5,1,r][5,1,r][5,1,r][5,1,r][5,1,r][5,1,r
      ][5,1,r][7,1,r][7,1,r][7,1,r][7,1,r][23,12,G][23,12,G][23,12,G
      ][23,12,G]
```

Here is the chain from [4] based on $\mathsf{LF}(4)$ with a claimed complexity of $2^{72.844}$:

```
chain for LPN_{512,0.125} with n=2^63.5260
    drop(10)-(502,53.53) lcomp=64.525 lbc2=-0.830
      sparse-(502,53.53) lcomp=66.734 lbc2=-0.830
    lf4(156)-(346,53.52) lcomp=115.024 lbc2=-3.320
    lf4(156)-(190,53.49) lcomp=114.473 lbc2=-13.281
  guess(16u1)-(174,53.49) lcomp=0.000 lbc2=-13.281
        code-(60,53.49) lcomp=60.934 lbc2=-43.739
        WHT(theta=0.00) lcomp=70.675
         lr_all=1.366 lr_wht=4.087
                  lmaxcomp=116.39
                  ltotcomp=117.14
  for the used code, log_2 bc^2 =-30.457352130329587529315901036020611109
  used code family: old+QP+rnd
  params=[25,15,W][19,4,rnd150926][16,5,rnd150926][19,6,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,
      rnd150926]
```

Here is the corrected one to reach $\theta = 33\%$:

```
chain for LPN_{512,0.125} with n=2^63.3730
    drop(10)-(502,53.37) lcomp=64.372 lbc2=-0.830
      sparse-(502,53.37) lcomp=66.329 lbc2=-0.830
    lf4(156)-(346,52.91) lcomp=114.718 lbc2=-3.320
    lf4(156)-(190,51.04) lcomp=113.249 lbc2=-13.281
  guess(16u1)-(174,51.04) lcomp=0.000 lbc2=-13.281
        code-(60,51.04) lcomp=58.486 lbc2=-43.739
       WHT(theta=30.00) lcomp=70.609
         lr_all=1.366 lr_wht=4.087
                  lmaxcomp=116.08
                  ltotcomp=116.53
  for the used code, log_2 bc^2 =-30.457352130329587529315901036020611109
  used code family: old+QP+rnd
  params=[25,15,W][19,4,rnd150926][16,5,rnd150926][19,6,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,
      rnd150926]
```

Here is the chain we obtain by running our algorithm with precision 0.1, after removing the roundings and adjusting the number of queries:

```
chain for LPN_{512,0.125} with n=2^63.2990
      sparse-(512,63.30) lcomp=76.215 lbc2=-0.830
     xor(59)-(453,66.60) lcomp=75.598 lbc2=-1.660
```

```
    xor(65)-(388,67.20) lcomp=76.019 lbc2=-3.320
    xor(66)-(322,67.39) lcomp=75.992 lbc2=-6.641
    xor(66)-(256,67.78) lcomp=76.115 lbc2=-13.281
    xor(67)-(189,67.57) lcomp=75.784 lbc2=-26.562
       code-(64,67.57) lcomp=75.130 lbc2=-60.165
      WHT(theta=28.00) lcomp=75.528
                  lmaxcomp=76.22
                  ltotcomp=78.84
  for the used code, log_2 bc^2 =-33.602837640638774272597586977976944615
  used code family: old+QP+rnd
  params=[18,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,7,rnd150926][19,7,rnd150926
      ][19,7,rnd150926][19,7,rnd150926]
```

For comparison, here is the same computation using only perfect and quasi-perfect codes:

```
chain for LPN_{512,0.125} with n=2^63.3400
     sparse-(512,63.34) lcomp=76.240 lbc2=-0.830
    xor(59)-(453,66.68) lcomp=75.680 lbc2=-1.660
    xor(65)-(388,67.36) lcomp=76.183 lbc2=-3.320
    xor(66)-(322,67.72) lcomp=76.320 lbc2=-6.641
    xor(66)-(256,68.44) lcomp=76.771 lbc2=-13.281
    xor(67)-(189,68.88) lcomp=76.880 lbc2=-26.562
       code-(64,68.88) lcomp=76.442 lbc2=-61.459
     WHT(theta=18.00) lcomp=76.009
                  lmaxcomp=76.88
                  ltotcomp=79.36
  for the used code, log_2 bc^2 =-34.896841882956364951443303534972935944
  used code family: old+QP
  params=[7,1,r][7,1,r][7,1,r][7,1,r][7,1,r][7,1,r][7,1,r][7,1,r][7,1,r
      ][7,1,r][8,2,iGop][11,4,S][25,12,FP][25,12,FP][25,12,FP][25,12,FP]
```

For comparison, here is the same computation using only perfect codes:

```
chain for LPN_{512,0.125} with n=2^63.3490
     sparse-(512,63.35) lcomp=76.245 lbc2=-0.830
    xor(59)-(453,66.70) lcomp=75.698 lbc2=-1.660
    xor(65)-(388,67.40) lcomp=76.219 lbc2=-3.320
    xor(66)-(322,67.79) lcomp=76.392 lbc2=-6.641
    xor(66)-(256,68.58) lcomp=76.915 lbc2=-13.281
    xor(67)-(189,69.17) lcomp=77.168 lbc2=-26.562
       code-(64,69.17) lcomp=76.730 lbc2=-61.749
      WHT(theta=18.00) lcomp=76.150
                  lmaxcomp=77.17
                  ltotcomp=79.51
  for the used code, log_2 bc^2 =-35.186278395343000828324443022088734514
  used code family: old
  params=[5,1,r][5,1,r][5,1,r][5,1,r][5,1,r][5,1,r][5,1,r][7,1,r][7,1,r
      ][7,1,r][7,1,r][7,1,r][6,1,r][7,1,r][7,1,r][7,1,r][23,12,G][23,12,G
      ][23,12,G][23,12,G]
```

7.2 Chains for $\mathsf{LPN}_{532,1/8}$

Here is the chain proposed in [3] with a claimed complexity of $2^{81.82}$:

```
chain for LPN_{532,0.125} with n=2^68.0000
      sparse-(532,68.00) lcomp=81.153 lbc2=-0.830
    part(65)-(467,67.81) lcomp=77.055 lbc2=-1.660
    part(65)-(402,67.58) lcomp=76.675 lbc2=-3.320
    part(65)-(337,67.32) lcomp=76.236 lbc2=-6.641
    part(65)-(272,67.00) lcomp=75.719 lbc2=-13.281
    part(65)-(207,66.58) lcomp=75.087 lbc2=-26.562
    part(65)-(142,66.00) lcomp=74.278 lbc2=-53.125
  guess(12u2)-(130,66.00) lcomp=0.000 lbc2=-53.125
        code-(66,66.00) lcomp=73.022 lbc2=-66.904
       WHT(theta=100.00) lcomp=77.153
         lr_all=0.290 lr_wht=6.304
                  lmaxcomp=83.75
                  ltotcomp=84.06
  for the used code, log_2 bc^2 =-13.77919304211275368712222681033526497 9
  used code family: old+QP+rnd
  params=[25,15,W][25,15,W][25,15,W][23,12,G][13,4,rnd150926][19,5,
      rnd150926]
```

Here is the corrected one to reach $\theta = 33\%$:

```
chain for LPN_{532,0.125} with n=2^74.3600
      sparse-(532,74.36) lcomp=87.314 lbc2=-0.830
    part(65)-(467,74.36) lcomp=83.415 lbc2=-1.660
    part(65)-(402,74.36) lcomp=83.225 lbc2=-3.320
    part(65)-(337,74.35) lcomp=83.007 lbc2=-6.641
    part(65)-(272,74.35) lcomp=82.750 lbc2=-13.281
    part(65)-(207,74.35) lcomp=82.439 lbc2=-26.562
    part(65)-(142,74.35) lcomp=82.042 lbc2=-53.125
  guess(12u2)-(130,74.35) lcomp=0.000 lbc2=-53.125
        code-(66,74.35) lcomp=81.369 lbc2=-66.904
       WHT(theta=33.00) lcomp=80.549
         lr_all=0.290 lr_wht=6.304
                  lmaxcomp=87.60
                  ltotcomp=88.62
  for the used code, log_2 bc^2 =-13.77919304211275368712222681033526497 9
  used code family: old+QP+rnd
  params=[25,15,W][25,15,W][25,15,W][23,12,G][13,4,rnd150926][19,5,
      rnd150926]
```

Here is the chain from [4] based on LF1 with a claimed complexity of $2^{78.182}$:

```
chain for LPN_{532,0.125} with n=2^73.5840
    drop( 5)-(527,68.58) lcomp=74.538 lbc2=-0.830
      sparse-(527,68.58) lcomp=81.476 lbc2=-0.830
    part(65)-(462,68.46) lcomp=77.626 lbc2=-1.660
    part(65)-(397,68.32) lcomp=77.310 lbc2=-3.320
```

```
    part(65)-(332,68.17) lcomp=76.954 lbc2=-6.641
    part(65)-(267,68.00) lcomp=76.544 lbc2=-13.281
    part(65)-(202,67.81) lcomp=76.059 lbc2=-26.562
  guess(20u1)-(182,67.81) lcomp=0.000 lbc2=-26.562
        code-(64,67.81) lcomp=75.313 lbc2=-57.882
        WHT(theta=0.00) lcomp=75.597
         lr_all=1.905 lr_wht=4.392
                  lmaxcomp=83.38
                  ltotcomp=84.06
  for the used code, log_2 bc^2 =-31.319243871934051230338929039571812625
  used code family: old+QP+rnd
  params=[5,1,r][25,15,W][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926]
```

Here is the corrected one to reach $\theta = 33\%$:

```
chain for LPN_{532,0.125} with n=2^72.6360
    drop( 5)-(527,67.64) lcomp=73.590 lbc2=-0.830
      sparse-(527,67.64) lcomp=80.500 lbc2=-0.830
    part(65)-(462,67.38) lcomp=76.678 lbc2=-1.660
    part(65)-(397,67.08) lcomp=76.235 lbc2=-3.320
    part(65)-(332,66.69) lcomp=75.709 lbc2=-6.641
    part(65)-(267,66.15) lcomp=75.060 lbc2=-13.281
    part(65)-(202,65.28) lcomp=74.209 lbc2=-26.562
  guess(20u1)-(182,65.28) lcomp=0.000 lbc2=-26.562
        code-(64,65.28) lcomp=72.790 lbc2=-57.882
       WHT(theta=30.00) lcomp=75.153
         lr_all=1.905 lr_wht=4.392
                  lmaxcomp=82.41
                  ltotcomp=83.19
  for the used code, log_2 bc^2 =-31.319243871934051230338929039571812625
  used code family: old+QP+rnd
  params=[5,1,r][25,15,W][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926]
```

Here is the chain from [4] based on LF2 with a claimed complexity of $2^{76.902}$:

```
chain for LPN_{532,0.125} with n=2^73.9830
    drop( 7)-(525,66.98) lcomp=74.972 lbc2=-0.830
      sparse-(525,66.98) lcomp=80.115 lbc2=-0.830
     xor(66)-(459,66.97) lcomp=76.019 lbc2=-1.660
     xor(66)-(393,66.93) lcomp=75.808 lbc2=-3.320
     xor(66)-(327,66.86) lcomp=75.550 lbc2=-6.641
     xor(66)-(261,66.73) lcomp=75.217 lbc2=-13.281
     xor(66)-(195,66.46) lcomp=74.756 lbc2=-26.562
  guess(17u1)-(178,66.46) lcomp=0.000 lbc2=-26.562
        code-(64,66.46) lcomp=73.932 lbc2=-56.630
        WHT(theta=0.00) lcomp=75.293
```

```
         lr_all=1.497 lr_wht=4.170
                  lmaxcomp=81.61
                  ltotcomp=82.53
  for the used code, log_2 bc^2 =-30.067365090154126592900296031961631480
  used code family: old+QP+rnd
  params=[3,1,r][25,15,W][18,6,rnd150926][18,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926]
```

Here is the corrected one to reach $\theta = 33\%$:

```
chain for LPN_{532,0.125} with n=2^73.9080
    drop( 7)-(525,66.91) lcomp=74.897 lbc2=-0.830
      sparse-(525,66.91) lcomp=79.666 lbc2=-0.830
     xor(66)-(459,66.82) lcomp=75.944 lbc2=-1.660
     xor(66)-(393,66.63) lcomp=75.658 lbc2=-3.320
     xor(66)-(327,66.26) lcomp=75.250 lbc2=-6.641
     xor(66)-(261,65.53) lcomp=74.617 lbc2=-13.281
     xor(66)-(195,64.06) lcomp=73.556 lbc2=-26.562
  guess(17u1)-(178,64.06) lcomp=0.000 lbc2=-26.562
        code-(64,64.06) lcomp=71.532 lbc2=-56.630
       WHT(theta=15.00) lcomp=75.069
         lr_all=1.497 lr_wht=4.170
                  lmaxcomp=81.16
                  ltotcomp=82.17
  for the used code, log_2 bc^2 =-30.067365090154126592900296031961631480
  used code family: old+QP+rnd
  params=[3,1,r][25,15,W][18,6,rnd150926][18,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926]
```

Here is the chain from [4] based on $\mathsf{LF}(4)$ with a claimed complexity of $2^{74.709}$:

```
chain for LPN_{532,0.125} with n=2^70.5040
    drop(15)-(517,55.50) lcomp=71.504 lbc2=-0.830
      sparse-(517,55.50) lcomp=68.792 lbc2=-0.830
    lf4(162)-(355,55.43) lcomp=119.022 lbc2=-3.320
    lf4(162)-(193,55.14) lcomp=118.334 lbc2=-13.281
  guess(13u1)-(180,55.14) lcomp=0.000 lbc2=-13.281
        code-(61,55.14) lcomp=62.631 lbc2=-45.275
        WHT(theta=0.00) lcomp=71.743
         lr_all=0.990 lr_wht=3.807
                  lmaxcomp=120.01
                  ltotcomp=120.71
  for the used code, log_2 bc^2 =-31.994004843189402519390178262046306566
  used code family: old+QP+rnd
  params=[3,1,r][25,15,W][19,4,rnd150926][19,5,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926]
```

Here is the corrected one to reach $\theta = 33\%$:

```
chain for LPN_{532,0.125} with n=2^70.3460
    drop(15)-(517,55.35) lcomp=71.346 lbc2=-0.830
      sparse-(517,55.35) lcomp=68.278 lbc2=-0.830
    lf4(162)-(355,54.80) lcomp=118.706 lbc2=-3.320
    lf4(162)-(193,52.61) lcomp=117.070 lbc2=-13.281
  guess(13u1)-(180,52.61) lcomp=0.000 lbc2=-13.281
        code-(61,52.61) lcomp=60.103 lbc2=-45.275
       WHT(theta=25.00) lcomp=71.675
         lr_all=0.990 lr_wht=3.807
                  lmaxcomp=119.70
                  ltotcomp=120.10
  for the used code, log_2 bc^2 =-31.994004843189402519390178262046306566
  used code family: old+QP+rnd
  params=[3,1,r][25,15,W][19,4,rnd150926][19,5,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926]
```

Here is the chain we obtain by running our algorithm with precision 0.1, after removing the roundings and adjusting the number of queries:

```
chain for LPN_{532,0.125} with n=2^65.3000
      sparse-(532,65.30) lcomp=78.290 lbc2=-0.830
     xor(61)-(471,68.60) lcomp=77.655 lbc2=-1.660
     xor(67)-(404,69.20) lcomp=78.080 lbc2=-3.320
     xor(68)-(336,69.40) lcomp=78.058 lbc2=-6.641
     xor(68)-(268,69.80) lcomp=78.192 lbc2=-13.281
     xor(69)-(199,69.60) lcomp=77.866 lbc2=-26.562
        code-(67,69.60) lcomp=77.237 lbc2=-62.111
       WHT(theta=17.00) lcomp=78.436
                  lmaxcomp=78.44
                  ltotcomp=81.02
  for the used code, log_2 bc^2 =-35.548346362647631165140054416118030948
  used code family: old+QP+rnd
  params=[3,1,r][25,15,W][19,4,rnd150926][19,5,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926]
```

7.3 Chains for $\mathsf{LPN}_{592,1/8}$

Here is the chain proposed in [3] with a claimed complexity of $2^{88.07}$:

```
chain for LPN_{592,0.125} with n=2^72.7000
      sparse-(592,72.70) lcomp=85.748 lbc2=-0.830
    part(70)-(522,72.46) lcomp=81.909 lbc2=-1.660
    part(70)-(452,72.17) lcomp=81.487 lbc2=-3.320
    part(70)-(382,71.81) lcomp=80.989 lbc2=-6.641
    part(70)-(312,71.32) lcomp=80.384 lbc2=-13.281
    part(70)-(242,70.58) lcomp=79.606 lbc2=-26.562
    part(70)-(172,68.99) lcomp=78.502 lbc2=-53.125
  guess(35u3)-(137,68.99) lcomp=0.000 lbc2=-53.125
```

```
        code-(64,68.99) lcomp=76.092 lbc2=-69.942
      WHT(theta=100.00) lcomp=76.063
         lr_all=1.524 lr_wht=12.809
                  lmaxcomp=90.40
                  ltotcomp=90.58
  for the used code, log_2 bc^2 =-16.816717432567051155707898232895271319
  used code family: old+QP+rnd
  params=[5,1,r][25,15,W][25,15,W][25,15,W][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926]
```

Here is the corrected one to reach $\theta = 33\%$:

```
chain for LPN_{592,0.125} with n=2^77.3910
      sparse-(592,77.39) lcomp=90.513 lbc2=-0.830
    part(70)-(522,77.38) lcomp=86.600 lbc2=-1.660
    part(70)-(452,77.37) lcomp=86.410 lbc2=-3.320
    part(70)-(382,77.36) lcomp=86.194 lbc2=-6.641
    part(70)-(312,77.36) lcomp=85.942 lbc2=-13.281
    part(70)-(242,77.35) lcomp=85.642 lbc2=-26.562
    part(70)-(172,77.34) lcomp=85.266 lbc2=-53.125
  guess(35u3)-(137,77.34) lcomp=0.000 lbc2=-53.125
        code-(64,77.34) lcomp=84.437 lbc2=-69.942
       WHT(theta=33.00) lcomp=83.344
         lr_all=1.524 lr_wht=12.809
                  lmaxcomp=97.68
                  ltotcomp=97.71
  for the used code, log_2 bc^2 =-16.816717432567051155707898232895271319
  used code family: old+QP+rnd
  params=[5,1,r][25,15,W][25,15,W][25,15,W][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926]
```

Here is the chain from [4] based on LF1 with a claimed complexity of $2^{84.715}$:

```
chain for LPN_{592,0.125} with n=2^79.5570
    drop( 4)-(588,75.56) lcomp=80.464 lbc2=-0.830
      sparse-(588,75.56) lcomp=88.675 lbc2=-0.830
    part(73)-(515,75.29) lcomp=84.757 lbc2=-1.660
    part(73)-(442,74.96) lcomp=84.297 lbc2=-3.320
    part(73)-(369,74.53) lcomp=83.746 lbc2=-6.641
    part(73)-(296,73.91) lcomp=83.056 lbc2=-13.281
    part(73)-(223,72.82) lcomp=82.124 lbc2=-26.562
  guess(16u1)-(207,72.82) lcomp=0.000 lbc2=-26.562
        code-(72,72.82) lcomp=80.517 lbc2=-62.542
        WHT(theta=0.00) lcomp=83.443
         lr_all=1.366 lr_wht=4.087
                  lmaxcomp=90.04
                  ltotcomp=90.75
  for the used code, log_2 bc^2 =-35.979931993882924904548450029191500372
  used code family: old+QP+rnd
```

```
  params=[18,6,rnd150926][25,15,W][13,4,rnd150926][18,5,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926]
```

Here is the corrected one to reach $\theta = 33\%$:

```
chain for LPN_{592,0.125} with n=2^79.3610
    drop( 4)-(588,75.36) lcomp=80.268 lbc2=-0.830
      sparse-(588,75.36) lcomp=88.561 lbc2=-0.830
    part(73)-(515,75.05) lcomp=84.561 lbc2=-1.660
    part(73)-(442,74.65) lcomp=84.057 lbc2=-3.320
    part(73)-(369,74.10) lcomp=83.437 lbc2=-6.641
    part(73)-(296,73.19) lcomp=82.623 lbc2=-13.281
    part(73)-(223,70.14) lcomp=81.395 lbc2=-26.562
  guess(16u1)-(207,70.14) lcomp=0.000 lbc2=-26.562
        code-(72,70.14) lcomp=77.829 lbc2=-62.542
       WHT(theta=19.00) lcomp=83.334
         lr_all=1.366 lr_wht=4.087
                  lmaxcomp=89.93
                  ltotcomp=90.62
  for the used code, log_2 bc^2 =-35.979931993882924904548450029191500372
  used code family: old+QP+rnd
  params=[18,6,rnd150926][25,15,W][13,4,rnd150926][18,5,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926]
```

Here is the chain from [4] based on LF2 with a claimed complexity of $2^{83.843}$:

```
chain for LPN_{592,0.125} with n=2^77.9850
    drop( 4)-(588,73.99) lcomp=78.892 lbc2=-0.830
      sparse-(588,73.98) lcomp=86.931 lbc2=-0.830
     xor(73)-(515,73.97) lcomp=83.185 lbc2=-1.660
     xor(73)-(442,73.94) lcomp=82.978 lbc2=-3.320
     xor(73)-(369,73.88) lcomp=82.728 lbc2=-6.641
     xor(73)-(296,73.76) lcomp=82.407 lbc2=-13.281
     xor(73)-(223,73.52) lcomp=81.969 lbc2=-26.562
  guess(14u1)-(209,73.52) lcomp=0.000 lbc2=-26.562
        code-(72,73.52) lcomp=81.227 lbc2=-63.159
        WHT(theta=0.00) lcomp=83.496
         lr_all=1.112 lr_wht=3.907
                  lmaxcomp=88.52
                  ltotcomp=89.46
  for the used code, log_2 bc^2 =-36.596897467910004630929717148293030280
  used code family: old+QP+rnd
  params=[25,15,W][13,4,rnd150926][19,5,rnd150926][19,6,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926]
```

Here is the corrected one to reach $\theta = 33\%$:

```
chain for LPN_{592,0.125} with n=2^77.8980
```

```
    drop( 4)-(588,73.90) lcomp=78.805 lbc2=-0.830
      sparse-(588,73.90) lcomp=86.873 lbc2=-0.830
     xor(73)-(515,73.80) lcomp=83.098 lbc2=-1.660
     xor(73)-(442,73.59) lcomp=82.804 lbc2=-3.320
     xor(73)-(369,73.18) lcomp=82.380 lbc2=-6.641
     xor(73)-(296,72.37) lcomp=81.711 lbc2=-13.281
     xor(73)-(223,70.74) lcomp=80.577 lbc2=-26.562
  guess(14u1)-(209,70.74) lcomp=0.000 lbc2=-26.562
        code-(72,70.74) lcomp=78.443 lbc2=-63.159
       WHT(theta=30.00) lcomp=83.351
         lr_all=1.112 lr_wht=3.907
                  lmaxcomp=88.37
                  ltotcomp=89.32
  for the used code, log_2 bc^2 =-36.596897467910004630929717148293030280
  used code family: old+QP+rnd
  params=[25,15,W][13,4,rnd150926][19,5,rnd150926][19,6,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,
      rnd150926][19,6,rnd150926][19,6,rnd150926]
```

Here is the chain from [4] based on $\mathsf{LF}(4)$ with a claimed complexity of $2^{81.963}$:

```
chain for LPN_{592,0.125} with n=2^78.5130
    drop(18)-(574,60.51) lcomp=79.513 lbc2=-0.830
      sparse-(574,60.51) lcomp=73.677 lbc2=-0.830
    lf4(177)-(397,60.47) lcomp=129.191 lbc2=-3.320
    lf4(177)-(220,60.28) lcomp=128.567 lbc2=-13.281
  guess(16u1)-(204,60.28) lcomp=0.000 lbc2=-13.281
        code-(68,60.28) lcomp=67.956 lbc2=-50.044
        WHT(theta=0.00) lcomp=79.025
         lr_all=1.366 lr_wht=4.087
                  lmaxcomp=130.56
                  ltotcomp=131.28
  for the used code, log_2 bc^2 =-36.762390670153894058017705531513633529
  used code family: old+QP+rnd
  params=[19,5,rnd150926][8,2,iGop][25,15,W][19,4,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926]
```

Here is the corrected one to reach $\theta = 33\%$:

```
chain for LPN_{592,0.125} with n=2^78.3410
    drop(18)-(574,60.34) lcomp=79.341 lbc2=-0.830
      sparse-(574,60.34) lcomp=73.561 lbc2=-0.830
    lf4(177)-(397,59.78) lcomp=128.847 lbc2=-3.320
    lf4(177)-(220,57.53) lcomp=127.191 lbc2=-13.281
  guess(16u1)-(204,57.53) lcomp=0.000 lbc2=-13.281
        code-(68,57.53) lcomp=65.204 lbc2=-50.044
       WHT(theta=33.00) lcomp=78.959
         lr_all=1.366 lr_wht=4.087
                  lmaxcomp=130.21
```

```
                  ltotcomp=130.61
  for the used code, log_2 bc^2 =-36.76239067015389405801770553151363352 9
  used code family: old+QP+rnd
  params=[19,5,rnd150926][8,2,iGop][25,15,W][19,4,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926]
```

Here is the chain we obtain by running our algorithm with precision 0.1, after removing the roundings and adjusting the number of queries:

```
chain for LPN_{592,0.125} with n=2^71.6910
      sparse-(592,71.69) lcomp=84.735 lbc2=-0.830
     xor(67)-(525,75.38) lcomp=84.591 lbc2=-1.660
     xor(74)-(451,75.76) lcomp=84.800 lbc2=-3.320
     xor(75)-(376,75.53) lcomp=84.581 lbc2=-6.641
     xor(74)-(302,76.06) lcomp=84.611 lbc2=-13.281
     xor(75)-(227,76.11) lcomp=84.350 lbc2=-26.562
         code-(73,76.11) lcomp=83.939 lbc2=-68.504
        WHT(theta=22.00) lcomp=84.751
                  lmaxcomp=84.80
                  ltotcomp=87.57
  for the used code, log_2 bc^2 =-41.94154951694101951676162163044818738 2
  used code family: old+QP+rnd
  params=[18,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926
      ][19,6,rnd150926][19,6,rnd150926][19,6,rnd150926][19,7,rnd150926]
```

References

1. Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 435–440. ACM, 2000.
2. Sonia Bogos, Florian Tramèr, and Serge Vaudenay. On solving LPN using BKW and variants - Implementation and analysis. *Cryptography and Communications*, 8(3):331–369, 2016.
3. Qian Guo, Thomas Johansson, and Carl Löndahl. Solving LPN Using Covering Codes. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2014.
4. Bin Zhang, Lin Jiao, and Mingsheng Wang. Faster algorithms for solving LPN. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 168–195. Springer, 2016.