From: Leo Ducas <leo.ducas1@gmail.com>
Sent: Tuesday, December 26, 2017 6:12 AM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: DRS

Dear Authors,

I am trying to understand and reproduce your reduction algorithm for signatures. I've had trouble with the document, where I think there are a few typos (marked by comments in the following python code).

My implementation ends up getting stuck in an infinite loop. While most signatures are produced within less than 10000 iterations of the reduction loop, one sample in (say) 30 does not stop before a million iteration.

I suspect that my implementation is still not conform to yours. In particular I would be grateful if you could specify more precisely the desired behavior for the rounded division w[i]/D (toward 0 ? toward -oo ?). Please also let me know if the typo I suspect are indeed typoes and if I addressed them properly.

Best regards
-- Leo Ducas

==== tentative re-implementation in python

```
import random
from numpy import zeros, int64
import sys

# params
n = 912
D = n
b = 28  // typos ? Sometime referred as B in the document
Nb = 16
N1 = 432
delta = 28

Samples = 100000


def RandomSign():
    return random.randint(0, 1)*2 -1

def KeyGen():
    L = Nb*[b] + N1 *[1] + (n - 1 - Nb - N1) * [0]
    random.shuffle(L)
    t = [D] + L
    S = zeros((n, n), dtype=int64)

    for i in xrange(n):    # Typo ? i should start at 0 ?
        S[i, i] = t[0]
```

```
        for j in xrange(1, n):
            S[i, (i+j) % n] = t[j] * RandomSign() # typo ? t is one-dimensional
    return S


# For efficiency, I'm using directly vector operation rather than loops.
# I rewrote this assuming that the lign w_l <- w_l + q M_{i,j} is typoed,
# and that the correct instruction is w_l <- w_l + q M_{i,l}


# Unlike C, Python does not round toward 0 but toward -oo, adding an option
# to force this behavior

ROUND_TOWARD_ZERO = True

def Sign(S, v):
    w = v
    i = 0
    it = 0
    while True:
        it += 1
        if it % 1000 == 0:
            print it,
            sys.stdout.flush()

        if ROUND_TOWARD_ZERO and w[i] < 0:
            q = - ((-w[i])/D)
        else:
            q = w[i]/D

        w -= q * S[i]

        if (max([abs(x) for x in w]) < D):
            return w

        i = (i + 1) % n


def SignRandom(S):
    v = zeros(n, dtype=int64)
    f = 2**delta - 1
    for i in xrange(n):
        # Simulate random hash of fresh message
        # Always chosen positive, as the version with sign is commented out
        # in the reference implementation
        v[i] = random.randint(0, f)
    return Sign(S, v)

S = KeyGen()

for a in range(100):
    print
    print "Sample ", a
    SignRandom(S)
```

From: Leo Ducas <leo.ducas1@gmail.com>
Sent: Sunday, February 04, 2018 10:42 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov; Han Zhao; Yang Yu; thomaspl@uow.edu.au
Subject: OFFICIAL COMMENT: DRS

Dear authors,

I think I have managed to reproduce your algorithm (it is now always terminating), I would nevertheless appreciate if you could confirm that I have properly interpreted potential typos in your report.

In addition, my colleagues and I have a further question regarding the specifications of your scheme:

In your PKC2008 paper, the message was hashed to a symmetric space [-W, W]^n, but this is less clear in the NIST document. It seems that the reference implementation samples in [0, W]^n for W=2^28. More precisely, there is a line to also randomize the sign (line 312 of char_key_conv.h) but it has been commented out.

Could you please clarify the intended hashing space ?

Thanks in advance
-- Leo Ducas, Yang Yu and Han Zhao

From: Thomas Plantard <thomaspl@uow.edu.au>
Sent: Friday, February 09, 2018 3:34 AM
To: Leo Ducas; pqc-comments
Cc: pqc-forum@list.nist.gov; Han Zhao; Yang Yu
Subject: Re: OFFICIAL COMMENT: DRS

Dear Leo, Yang and Han,
regarding the typos, I think Arnaud answered it or should answer it really soon.
Sorry if there was too many for understanding clearly our scheme.
As I told you, I can confirm that our "intentions" was a signed message.
However, if you think that a positive only message could be an issue, please do not hesitate to communicate it (specially to us) as at this stage we want to minimize the number of corrections we need to do to our scheme. I am not sure of the process to modify submission: we may have to keep it to the necessary change only.
Please do not hesitate if you have further questions.
Best regards.
Thomas Plantard.

From: Leo Ducas <leo.ducas1@gmail.com>
Sent: Wednesday, March 28, 2018 2:44 AM
To: pqc-forum
Cc: pqc-comments
Subject: OFFICIAL COMMENT: DRS

Dear all,

We (Yang Yu and myself) have finalized our pre-print on a statistical attack against the DRS scheme:

https://eprint.iacr.org/2018/294.pdf

The attack is not devastating, but it significantly decreases the concrete security of the scheme. While countermeasures could be develop, we believe they should come with a thorough statistical analysis, and ideally a provable statement.

Best regards
-- Yang Yu and Leo Ducas

From: YUYANG <y-y13@mails.tsinghua.edu.cn>
Sent: Thursday, April 19, 2018 9:05 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: [pqc-forum] OFFICIAL COMMENT: DRS

Dear all,

Leo and I have updated our preprint on an attack against the DRS signature scheme:

https://eprint.iacr.org/2018/294.pdf.

A link of open source scripts is provided in the updated version.

Best regards,
-- Leo Ducas and Yang Yu
--

From: Thomas Plantard <thomas.plantard@gmail.com>
Sent: Friday, November 30, 2018 4:32 AM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: DRS

Dear Leo, Yang and all,

having taken into consideration your work, we have decided to change our key setup to provide secret keys without the structure you have attacked.
Vectors of the noise matrix of the diagonal dominant basis are now randomly taken in the set of all the possible ones i.e. all vectors of taxicab norm inferior to the diagonal coefficient.
As a result, the noise matrix is as random as the definition of a diagonal dominant matrix allows.
As discuss at the conference, changing the secret key implies larger dimensions. Detailed specification with proper parameters are provided in our website linked below.
Furthermore, we have improved other parts of our implementation and provided timing with and without AVX.
Here is the link to our website:
https://www.uow.edu.au/~thomaspl/drs.html

Best Regards.
Thomas Plantard on behalf of DRS submitters.