From: zhenfei zhang <zhangzhenfei@gmail.com>
Sent: Tuesday, April 23, 2019 8:10 AM
To: pqc-comments
Cc: pqc-forum@list.nist.gov; luxianhui@iie.ac.cn
Subject: ROUND 2 OFFICIAL COMMENT: LAC

Hi all,

We would like to thank Oscar Garcia-Morchon, Ludo Tolhuizen and Sauvik Bhattacharya for pointing out a mistake in our submission.

In the submission, we claimed that hybrid attacks are no better than lattice reductions under core sieving model. This is not correct for LAC192 under classic core sieving model. Under this model, hybrid attacks takes roughly 2^278 operations, which exceeds the pure lattice reduction at 286 bit operations. This is because in LAC192 parameters, we have used a very sparse secret/error distribution from fixed hamming weight ternary distribution (i.e., 128 +/-1s, 768 0s). We overlooked the fact that hybrid attack is more efficient for this sparse secret.
We have given a revised estimation for our parameter set:
https://eprint.iacr.org/2018/1009.pdf

In summery,
* the analysis of LAC128 and LAC256 remain intact.
* the security of LAC192 against quantum computers also remains unchanged.
* the security of LAC192 against classical computers dropped to 278 from 286.
This revision does not affect the security category that each parameter set is aiming for, thanks to the adequate security margin we have build in.

Regards,
Zhenfei (on behalf of the LAC team)

From: Leo Ducas <leo.ducas1@gmail.com>
Sent: Tuesday, April 23, 2019 12:42 PM
To: pqc-forum
Cc: pqc-comments; luxianhui@iie.ac.cn
Subject: [pqc-forum] Re: ROUND 2 OFFICIAL COMMENT: LAC

Dear LAC team,

could you maybe clarify how was the cost of the hybrid attack estimated ? In particular, some schemes assume for simplicity (and/or conservativeness) a collision probability of 1 (NTRUprime maybe ?), though it can sometime be *much* lower according to https://eprint.iacr.org/2016/733.pdf .

Unfortunately the link to Thomas Wunderer' script seems dead, I'll poke him to see if it can be dug up... Cross-checks would be valuable.

More nitpicky: are the calls to Nearest-plane algorithm costed to 1, or to ~d^2 (or maybe something else) ?

Best regards
-- Leo Ducas

Le mardi 23 avril 2019 14:10:27 UTC+2, zhenfei zhang a écrit :

> Hi all,
>
> We would like to thank Oscar Garcia-Morchon, Ludo Tolhuizen and Sauvik Bhattacharya for pointing out a mistake in our submission.
>
> In the submission, we claimed that hybrid attacks are no better than lattice reductions under core sieving model. This is not correct for LAC192 under classic core sieving model. Under this model, hybrid attacks takes roughly 2^278 operations, which exceeds the pure lattice reduction at 286 bit operations. This is because in LAC192 parameters, we have used a very sparse secret/error distribution from fixed hamming weight ternary distribution (i.e., 128 +/-1s, 768 0s). We overlooked the fact that hybrid attack is more efficient for this sparse secret.
> We have given a revised estimation for our parameter set:
> https://eprint.iacr.org/2018/1009.pdf
>
> In summery,
> * the analysis of LAC128 and LAC256 remain intact.
> * the security of LAC192 against quantum computers also remains unchanged.
> * the security of LAC192 against classical computers dropped to 278 from 286.
> This revision does not affect the security category that each parameter set is aiming for, thanks to the adequate security margin we have build in.
>
> Regards,
> Zhenfei (on behalf of the LAC team)

From: zhenfei zhang <zhangzhenfei@gmail.com>
Sent: Tuesday, April 23, 2019 1:30 PM
To: Leo Ducas
Cc: pqc-forum; pqc-comments; luxianhui@iie.ac.cn
Subject: Re: [pqc-forum] Re: ROUND 2 OFFICIAL COMMENT: LAC

Hi Leo,

For conservative purpose
* the cost of NP is set to 1
* the probability of collision is also 1

We take the usual approach of estimating the cost.
1. cut the lattice basis B into two sublattices with basis B1 and B2, with dim(B1) = \ell, dim(B2) = dim(B)-\ell
2. find the best \ell such that

a. BKZ(B1) = Search(B2)
b. BDD can be solved with reduced B1 (under GSA assumption) and NP algorithm
// note Wunderer suggested GSA is different for q-array lattices; our analysis didn't take this into account

In both classic and quantum setting, the cost of search is set to the square root of the entropy.
The cost of BKZ is estimated by either classical core sieving or quantum core sieving model.

Zhenfei

On Tue, Apr 23, 2019 at 12:42 PM Leo Ducas <leo.ducas1@gmail.com> wrote:

> Dear LAC team,
>
> could you maybe clarify how was the cost of the hybrid attack estimated ? In particular, some schemes assume for simplicity (and/or conservativeness) a collision probability of 1 (NTRUprime maybe ?), though it can sometime be *much* lower according to https://eprint.iacr.org/2016/733.pdf .
>
> Unfortunately the link to Thomas Wunderer' script seems dead, I'll poke him to see if it can be dug up... Cross-checks would be valuable.
>
> More nitpicky: are the calls to Nearest-plane algorithm costed to 1, or to ~d^2 (or maybe something else) ?
>
> Best regards
> -- Leo Ducas
>
> Le mardi 23 avril 2019 14:10:27 UTC+2, zhenfei zhang a écrit :
>
>> Hi all,
>>
>> We would like to thank Oscar Garcia-Morchon, Ludo Tolhuizen and Sauvik Bhattacharya for pointing out a mistake in our submission.
>>
>> In the submission, we claimed that hybrid attacks are no better than lattice reductions under core sieving model. This is not correct for LAC192 under classic core sieving model. Under this model, hybrid attacks takes roughly 2^278

From: Leo Ducas <leo.ducas1@gmail.com>
Sent: Tuesday, April 23, 2019 1:58 PM
To: pqc-forum
Cc: leo.ducas1@gmail.com; pqc-comments; luxianhui@iie.ac.cn
Subject: Re: [pqc-forum] Re: ROUND 2 OFFICIAL COMMENT: LAC

Thanks for the prompt and precise answer Zhenfei,

> // note Wunderer suggested GSA is different for q-array lattices; our analysis didn't take this into account

Could you elaborate a bit more ? Do you mean that you use a straight GSA line rather than a broken line: flat for q vectors followed by the GSA slope ? Or something more subtle ?

Best regards
-- Leo
--

From: zhenfei zhang <zhangzhenfei@gmail.com>
Sent: Tuesday, April 23, 2019 6:37 PM
To: Leo Ducas
Cc: pqc-forum; pqc-comments; luxianhui@iie.ac.cn
Subject: Re: [pqc-forum] Re: ROUND 2 OFFICIAL COMMENT: LAC

> Could you elaborate a bit more ? Do you mean that you use a straight GSA line rather than a broken line: flat for q vectors followed by the GSA slope ? Or something more subtle ?

My bad. It is still a flat then slope line.

I meant to say that we didn't use the formula (2) and (3) of A.2 from https://eprint.iacr.org/2016/733.pdf .
We used John Schanck's script https://github.com/jschanck/estimator
The precise fomular
is https://github.com/jschanck/estimator/blob/fbf5f7181a6583dd22927fc4a1c69501214f6c29/estimate.gp#L101

Zhenfei

On Tue, Apr 23, 2019 at 1:57 PM Leo Ducas <leo.ducas1@gmail.com> wrote:

> Thanks for the prompt and precise answer Zhenfei,
>
> > // note Wunderer suggested GSA is different for q-array lattices; our analysis didn't take this into account
>
> Could you elaborate a bit more ? Do you mean that you use a straight GSA line rather than a broken line: flat for q vectors followed by the GSA slope ? Or something more subtle ?
>
> Best regards
> -- Leo
> --
>

--
Zhenfei Zhang
Cryptography Engineer
W: zhenfei@algorand.com
P: zhangzhenfei@gmail.com
https://www.algorand.com
https://zhenfeizhang.github.io
--

From: 이종혁 <n_seeu@naver.com>
Sent: Friday, August 16, 2019 11:01 AM
To: pqc-comments
Cc: pqc-forum
Subject: ROUND 2 OFFICIAL COMMENT: LAC

Dear LAC team,

In your LAC code, is `ecc_bytes' calculated by $\frac{m*t+8-1}{8}$?
According your description, the byte length of error correcting code is 18 in LAC128 because of t is 16.

However, on 'Constant-time BCH Error-Correcting Code' (eprint.iacr.org/2019/155.pdf), their ecc is only 31 bytes for 29 bits error.
BCH(511, 264, 59)
$\frac{9*29+8-1}{8}=33!=31$

Best regards,
JongHyeok Lee

From: Mike Hamburg <mike@shiftleft.org>
Sent: Monday, September 16, 2019 2:10 PM
To: pqc-comments
Cc: pqc-forum
Subject: [pqc-forum] ROUND 2 OFFICIAL COMMENT: LAC

Hello LAC authors,

While poking through the LAC implementation, I noticed that encryption isn't constant-time. In particular, sampling the error code with gen_psi_fix_ham takes an amount of time that depends on the seed.

This might or might not leak a useful amount of information during keygen and encrypt; with advanced timing attacks a surprising amount of data can leak from a single function call. But where it is really a problem is in re-encrypt with the FO transform. Like the earlier BCH timing leak, this enables an attacker to determine whether a given message decoded correctly or not.

Of course, a full attack is made more difficult by the presence of the BCH code, but it seems dangerous to assume that it is infeasible. At least, this would require another track of cryptanalysis.

Furthermore, it is difficult to re-implement around this issue without significantly hurting performance, because the sampling logic must be functionally equivalent for interoperability.

I suggest that the authors change the specification to use sorting, or on some other operation which is efficient to implement in constant time.

The same issue is present in Round5.

Regards,
— Mike

From: xianhui lu <luxianhui@gmail.com>
Sent: Monday, September 16, 2019 8:30 PM
To: Mike Hamburg
Cc: pqc-comments; pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: LAC

Hi Mike,

Thank you very much for pointing this out, and for the suggestion of sorting. We will try constant time implementation of the sampling algorithm.

Best wishes
Xianhui from LAC Team

Mike Hamburg <mike@shiftleft.org> 于2019年9月17日周二 上午2:09写道：

> Hello LAC authors,
>
> While poking through the LAC implementation, I noticed that encryption isn't constant-time. In particular, sampling the error code with gen_psi_fix_ham takes an amount of time that depends on the seed.
>
> This might or might not leak a useful amount of information during keygen and encrypt; with advanced timing attacks a surprising amount of data can leak from a single function call. But where it is really a problem is in re-encrypt with the FO transform. Like the earlier BCH timing leak, this enables an attacker to determine whether a given message decoded correctly or not.
>
> Of course, a full attack is made more difficult by the presence of the BCH code, but it seems dangerous to assume that it is infeasible. At least, this would require another track of cryptanalysis.
>
> Furthermore, it is difficult to re-implement around this issue without significantly hurting performance, because the sampling logic must be functionally equivalent for interoperability.
>
> I suggest that the authors change the specification to use sorting, or on some other operation which is efficient to implement in constant time.
>
> The same issue is present in Round5.
>
> Regards,
> — Mike
>
>