**From:** oscar.garcia-morchon@philips.com
**Sent:** Saturday, August 04, 2018 11:00 AM
**To:** pqc-forum
**Subject:** [pqc-forum] OFFICIAL COMMENT: Round5 = Round2 + Hila5

Dear all,

with this email we want to officially announce Round5, the merging of the Round2 and Hila5 proposals.

Technically, the merged proposal is based on Round2 combined with Hila5's techniques such as error correction to bring down the failure probability so that Round5 can achieve better bandwidth and CPU performance. The Round5 parameters are adapted to fully comply with NIST security levels.

We also want to announce that in addition to the members of the original Hila5 and Round2 proposals, Thijs Laarhoven (TU Eindhoven) has also joined the Round5 team.

All information is available at https://round5.org/ In the next weeks, we will post further updates.

In the website you can already find two papers. The first paper describes Round5 - including algorithms, security analysis and parameters. The second paper details an optimized implementation - due to Markku J.O. Saarinen - on Cortex M4. This implementation is available at https://github.com/round5/r5nd_tiny. The papers show that Round5 is a leading lattice-based candidate in terms of security, bandwidth and CPU performance.

Best regards,

Oscar, Markku, and the rest of the Round5 team.

**From:** Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
**Sent:** Saturday, August 04, 2018 12:29 PM
**To:** pqc-forum
**Subject:** [pqc-forum] Re: OFFICIAL COMMENT: Round5 = Round2 + Hila5

Hello All,

As announced by Oscar, my "Hila5" project has merged with the "Round2" effort to produce **Round5**, and this has been acknowledged by NIST (we gave them an early heads-up about our merger plans). I will no longer work on Hila5, so all analytic efforts should be directed to the new candidate. Round5 does inherit some design features from Hila5, perhaps most importantly the constant-time error correction code XEf.

Oscar will be the principal submitter of the new merged proposal (or tweak), so my opinions and communications should be taken as unofficial. However, I'd like to fill in some details:

We have worked hard on the new Round5 proposal. At least in my opinion it is clearly superior to either of the old ones; it has the best message sizes of lattice-based candidates, and also leading performance characteristics of *all* candidates.

This is largely because the flexibility of the design allowed a fine-tuned parameter search to meet the post-quantum and classical security levels while optimizing parameters, primarily for bandwidth (public key and ciphertext message sizes). The team has done a fantastic job at it.

* Hayo Baan (Philips, NL)
* Sauvik Bhattacharya (Philips, NL)
* Oscar Garcia-Morchon (Philips, NL)
* Thijs Laarhoven (TU/e, NL)
* Markku-Juhani O. Saarinen (PQShield, UK)
* Ronald Rietman (Philips, NL)
* Ludo Tolhuizen (Philips, NL)
* Jose Luis Torre Arce (Philips, NL)
* Zhenfei Zhang (OnboardSecurity, US)

The official homepage is at https://round5.org/. I also intend to keep some unofficial implementation-related materials at https://mjos.fi/round5/

While we prepare for the official NIST tweak proposal package, we are putting two preprints out at this time:

On parameter selection and security analysis:

* “Round5: Compact and Fast Post-Quantum Public-Key Encryption” by S. Bhattacharya, O. Garcia-Morchon, T. Laarhoven, R. Rietman, M.-J. Saarinen, L. Tolhuizen, and Z. Zhang. https://round5.org/doc/round5paper.pdf

Implementation aspects (of the ring variant) are discussed in:

* “Shorter Messages and Faster Post-Quantum Encryption with Round5 on Cortex M” by M.-J. Saarinen, S. Bhattacharya, O. Garcia-Morchon, R. Rietman, L. Tolhuizen, and Z.

Zhang. https://round5.org/doc/r5m4text.pdf

I'm still working on optimizing implementations. However, the Cortex M4 implementation discussed in the latter paper is already available at https://github.com/round5/r5nd_tiny

Again, please note that all parameter selections etc before the actual NIST submission are still subject to change.

Cheers,
- markku

Dr. Markku-Juhani O. Saarinen <mjos@iki.fi>

**From:** Leo Ducas <leo.ducas1@gmail.com>
**Sent:** Tuesday, August 07, 2018 2:07 AM
**To:** pqc-forum
**Subject:** [pqc-forum] Re: OFFICIAL COMMENT: Round5 = Round2 + Hila5

Dear authors,

I note that the failure analysis assumes that ``bit failures occur independently'', but I'm unconvinced it would be the case, especially in the ring setting. I have searched for solution to this issue for a long time, and still don't know how to properly address this issue theoretically.

May I suggest to resort to experimental analysis to test how close or not to independent these failure events are, at least in a regime where failures are statistically measurable ?

Best regards
-- Leo Ducas

Le samedi 4 août 2018 18:29:10 UTC+2, Markku-Juhani O. Saarinen a écrit :

> Hello All,
>
> As announced by Oscar, my "Hila5" project has merged with the "Round2" effort to produce **Round5**, and this has been acknowledged by NIST (we gave them an early heads-up about our merger plans). I will no longer work on Hila5, so all analytic efforts should be directed to the new candidate. Round5 does inherit some design features from Hila5, perhaps most importantly the constant-time error correction code XEf.
>
> Oscar will be the principal submitter of the new merged proposal (or tweak), so my opinions and communications should be taken as unofficial. However, I'd like to fill in some details:
>
> We have worked hard on the new Round5 proposal. At least in my opinion it is clearly superior to either of the old ones; it has the best message sizes of lattice-based candidates, and also leading performance characteristics of *all* candidates.
>
> This is largely because the flexibility of the design allowed a fine-tuned parameter search to meet the post-quantum and classical security levels while optimizing parameters, primarily for bandwidth (public key and ciphertext message sizes). The team has done a fantastic job at it.
>
> * Hayo Baan (Philips, NL)
> * Sauvik Bhattacharya (Philips, NL)
> * Oscar Garcia-Morchon (Philips, NL)
> * Thijs Laarhoven (TU/e, NL)
> * Markku-Juhani O. Saarinen (PQShield, UK)
> * Ronald Rietman (Philips, NL)
> * Ludo Tolhuizen (Philips, NL)
> * Jose Luis Torre Arce (Philips, NL)
> * Zhenfei Zhang (OnboardSecurity, US)

**From:** Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
**Sent:** Tuesday, August 07, 2018 6:44 AM
**To:** pqc-forum
**Subject:** [pqc-forum] Re: OFFICIAL COMMENT: Round5 = Round2 + Hila5

On Tuesday, August 7, 2018 at 7:06:42 AM UTC+1, Leo Ducas wrote:

> I note that the failure analysis assumes that ``bit failures occur independently'',
> but I'm unconvinced it would be the case, especially in the ring setting. I have
> searched for solution to this issue for a long time, and still don't know how to
> properly address this issue theoretically.

Hi Leo,

Same thing here, it's a known open issue. Also I find that my inconclusive results based on probability convolutions ( Section 3.2, https://eprint.iacr.org/2017/424 ) are now being cited in other works ( for example upcoming SAC paper https://eprint.iacr.org/2018/150 ).

Another, more easily avoidable problem is that some authors also appear to completely ignore the side-channel aspect of error correction codes, not fully realising that retrofitting side-channel resistance to error correction codes of certain type can incur a significant cost. Let me remind that this is one of the additional evaluation criteria (Section 4.A.6 of the Call).

In symmetric cryptographic design, which is arguably more mature, such a design omission would be unthinkable. However on some probability distributions one unfortunately still has to rely on heuristic statistical arguments there too.

> May I suggest to resort to experimental analysis to test how close or not to
> independent these failure events are, at least in a regime where failures are
> statistically measurable ?

I have done this, and it pretty much confirmed the analysis performed by the team. However affirmative computations of this type are rarely reported with analytic work, and the computations were not performed on the exact final parameters.

For example: One larger experiment was performed on an experimental set of parameters, but sufficiently large to be in similar scale to the ones we are using. I ran a version with n=d=700, h=196, q=2^14, p=2^8, t=2^4 with mu=700 message bits for m=2570540000 (2^31.26) keygen-encrypt-decrypt iterations, which produced *one* message with a single bit error. This puts the per-bit error rate at c=1/(mu*m) = 2^-40.71 and was consistent with the bounds we had analyzed for those parameters. Note that not all mu bits are actually used to carry a message and error correction codes were being used etc.

The computation took 30 CPU hours on a 16-node cluster and ran until that single error was found. It did provide evidence that errors do not happen radically more often than we expected, but of course statistical conclusions cannot be drawn from a single bit error.

Best Regards,
- markku

Dr. Markku-Juhani O. Saarinen <mjos@iki.fi>

**From:** Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
**Sent:** Tuesday, August 07, 2018 7:29 AM
**To:** pqc-forum
**Subject:** [pqc-forum] Re: OFFICIAL COMMENT: Round5 = Round2 + Hila5

Hi Leo,

Should add this is of course not only related to our submission, and I was answering only in a personal capacity (I'm getting slack from team for giving an initial reply without even consulting them -- they have more detailed answer to give than the few experiments I did.)

However we will probably instantiate and report a more robust experimental regime to answer the question of independence specifically in our case.

Cheers,
- markku

On Tuesday, August 7, 2018 at 11:43:45 AM UTC+1, Markku-Juhani O. Saarinen wrote:

> On Tuesday, August 7, 2018 at 7:06:42 AM UTC+1, Leo Ducas wrote:
>
> > I note that the failure analysis assumes that ``bit failures occur independently'',
> > but I'm unconvinced it would be the case, especially in the ring setting. I have
> > searched for solution to this issue for a long time, and still don't know how to
> > properly address this issue theoretically.
>
> Hi Leo,
>
> Same thing here, it's a known open issue. Also I find that my inconclusive results based on probability convolutions ( Section 3.2, https://eprint.iacr.org/2017/424 ) are now being cited in other works ( for example upcoming SAC paper https://eprint.iacr.org/2018/150 ).
>
> Another, more easily avoidable problem is that some authors also appear to completely ignore the side-channel aspect of error correction codes, not fully realising that retrofitting side-channel resistance to error correction codes of certain type can incur a significant cost. Let me remind that this is one of the additional evaluation criteria (Section 4.A.6 of the Call).
>
> In symmetric cryptographic design, which is arguably more mature, such a design omission would be unthinkable. However on some probability distributions one unfortunately still has to rely on heuristic statistical arguments there too.
>
> > May I suggest to resort to experimental analysis to test how close or not to
> > independent these failure events are, at least in a regime where failures are
> > statistically measurable ?
>
> I have done this, and it pretty much confirmed the analysis performed by the team. However affirmative computations of this type are rarely reported with analytic work, and the computations were not performed on the exact final parameters.
>
> For example: One larger experiment was performed on an experimental set of parameters, but sufficiently large to be in similar scale to the ones we are using. I ran a version with n=d=700, h=196, q=2^14, p=2^8, t=2^4 with mu=700 message bits for m=2570540000 (2^31.26) keygen-encrypt-decrypt iterations, which produced

**From:** Leo Ducas <leo.ducas1@gmail.com>
**Sent:** Tuesday, August 07, 2018 2:29 PM
**To:** pqc-forum
**Subject:** [pqc-forum] Re: OFFICIAL COMMENT: Round5 = Round2 + Hila5

Thanks for the pointers and your early experiments.

Beyond testing the full scheme itself, I think what would be interesting is to test independence for intermediate results. Indeed, I suspect that the product terms es' and e's have strong correlation, but they start to fade off a bit when summing es'+e's, and in the end, this may be mostly drown out by the rounding error term, whose coordinates are fully independent.

Best regards
-- Leo Ducas

Le mardi 7 août 2018 13:29:25 UTC+2, Markku-Juhani O. Saarinen a écrit :

> Hi Leo,
>
> Should add this is of course not only related to our submission, and I was answering only in a personal capacity (I'm getting slack from team for giving an initial reply without even consulting them -- they have more detailed answer to give than the few experiments I did.)
>
> However we will probably instantiate and report a more robust experimental regime to answer the question of independence specifically in our case.
>
> Cheers,
> - markku
>
> On Tuesday, August 7, 2018 at 11:43:45 AM UTC+1, Markku-Juhani O. Saarinen wrote:
>
> > On Tuesday, August 7, 2018 at 7:06:42 AM UTC+1, Leo Ducas wrote:
> >
> > > I note that the failure analysis assumes that ``bit failures occur independently'',
> > > but I'm unconvinced it would be the case, especially in the ring setting. I have
> > > searched for solution to this issue for a long time, and still don't know how to
> > > properly address this issue theoretically.
> >
> > Hi Leo,
> >
> > Same thing here, it's a known open issue. Also I find that my inconclusive results based on probability convolutions ( Section 3.2, https://eprint.iacr.org/2017/424 ) are now being cited in other works ( for example upcoming SAC paper https://eprint.iacr.org/2018/150 ).
> >
> > Another, more easily avoidable problem is that some authors also appear to completely ignore the side-channel aspect of error correction codes, not fully realising that retrofitting side-channel resistance to error correction codes of certain type can incur a significant cost. Let me remind that this is one of the additional evaluation criteria (Section 4.A.6 of the Call).
> >
> > In symmetric cryptographic design, which is arguably more mature, such a design omission would be unthinkable. However on some probability distributions one unfortunately still has to rely on heuristic statistical arguments there too.

**From:** Mike Hamburg <mike@shiftleft.org>
**Sent:** Friday, August 24, 2018 2:30 PM
**To:** Leo Ducas
**Cc:** pqc-forum
**Subject:** Re: [pqc-forum] OFFICIAL COMMENT: Round5 = Round2 + Hila5

Dear PQC,

I concur that independence of intermediate results is an important consideration, and in Round5 it may be a serious problem. I have discussed the issue with the Round5 team, and we have concluded the following.

The usual errors in a LWE/LWR scheme are of the form

s(As'+e') - s'(As+e) = se' - s'e

where in this case s,s' are the sparse ternary secrets and e,e' are the rounding error. The calculation is done modulo the cyclotomic polynomial \Phi_{n+1}(x).

If we were to instead calculate this in the NTRU ring — i.e. mod the polynomial x^{n+1} - 1 — each coefficient would be the sum of h rounding terms times +-1, where h is the fixed Hamming weight of the secret. In that case, while there might still be correlations between the magnitudes of the terms, they would probably be somewhat close to i.i.d..

To reduce the resulting value mod \Phi_{n+1}(x), we take the n'th coefficient and subtract it from the others. This means that if the n'th coefficient is large, we will have a greatly increased error probability in every position. This might cause many errors, so that the error-correcting code would be overwhelmed.

Concretely, we have estimated this probability in SAGE, and gotten 4-error probabilities around 2^-68, 26-56, 2^-55 for R5ND_PKE{1,3,5} respectively. This is high enough that it may allow an attack, especially if the adversary iterates to find ciphertexts with larger than expected amounts of rounding noise. I have discussed this with the Round5 team, who have confirmed the calculations. I have also confirmed this by setting s=700 in R5ND_PKE3. The same script estimates 2^-11.56 for these parameters, and indeed tests on Markku's code found errors a 2^-12.38 fraction of the time. The discrepancy may be explained by xe3 sometimes correcting more than 3 errors. But if errors were independent, the same script would predict a failure rate below 2^-32.

****

The above issues don't affect the original Round2, but they frustrate application of XEf or other error correction as in Round5.

At the PQC conference, I was working on a low-bandwidth KEM, called "Glowstick", which makes heavy use of error correction. After PQC, I discussed extending Glowstick to cyclotomics with Sauvik, Oscar and Ludo of the Round2 team. We developed a ring-switching trick, which may be able to repair the error correction in Round5, but which currently lacks a security proof. Consider that with a balanced sparse ternary secret, as in Round2, the coefficients of the secret s sum to 0. In other words, s is a multiple of x-1.

[The same is true with the "balanced balls-in-bins" distribution, \sum_{i=0}^{h-1} (-1)^i x^{random_i mod n}, in development versions of Glowstick.]

The trick is that we still send (As+e mod \Phi_{n+1}(x)) and the same for As'+e', just as in the current Round2 and Round5. But the approximate shared secret is s(As'+e') mod the NTRU polynomial x^{n+1} - 1, instead of mod \Phi. This is well-defined, because mod x^{n+1} - 1:

As+e mod \Phi_{n+1}(x) = As+e+k\Phi_{n+1}(x)

s'(As+e mod \Phi_{n+1}(x)) = s'(As+e) + ks'\Phi_{n+1}(x)

But indeed s'\Phi_{n+1}(x) = 0 mod x^{n+1}-1, because s' is a multiple of x-1. So the public key is mod \Phi_{n+1}(x), but the approximate shared secret is mod x^{n+1} - 1. This may make the security proof difficult from RLWE mod \Phi, but the LPR10 data is more heavily rounded and only has \mu < n coefficients, so it seems difficult for the attacker to make use of the different ring. It may even be possible to turn this into a proof, but we haven't managed to do it.

This is a trivial implementation change, since internally Round2 and Round5 are computing mod x^{n+1} - 1 anyway. So the change is just don't reduce mod \Phi at the end.

If this fix is employed, it prevents both the noise amplification and the major correlation problem described above, so error correction will be effective again. It also results in much lower failure rates than the current draft of Round5, or alternatively higher noise and better security. We would still need to check for smaller correlations between failures, caused by eg ciphertexts with larger than expected rounding noise.

Cheers,
— Mike Hamburg

> On Aug 7, 2018, at 11:29 AM, Leo Ducas <leo.ducas1@gmail.com> wrote:
>
> Thanks for the pointers and your early experiments.
>
> Beyond testing the full scheme itself, I think what would be interesting is to test independence for intermediate results. Indeed, I suspect that the product terms es' and e's have strong correlation, but they start to fade off a bit when summing es'+e's, and in the end, this may be mostly drown out by the rounding error term, whose coordinates are fully independent.
>
> Best regards
> -- Leo Ducas

**From:** Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
**Sent:** Friday, August 24, 2018 3:35 PM
**To:** pqc-forum
**Cc:** leo.ducas1@gmail.com
**Subject:** Re: [pqc-forum] OFFICIAL COMMENT: Round5 = Round2 + Hila5

(Talking in purely personal capacity here, just like to keep people updated.)

Thanks Mike!

As you know, we already have running code for the proposed fix (which is really simple, as you note), and smarter people than me are looking at the security proof adjustments.

As you note, this doesn't affect Round2 -- I would like to add that it doesn't affect Hila5 either (where the XEf error codes come from), since a ring of type x^n+1 was used there.

Since Round5 has lots of flexibility (with dimension and distribution parameters d, n, h, p, q, t etc) it takes a while to find the best parameters to match each security level and failure probability. I'm pretty sure that we'll update the tables soon after that's done; probably the implementation oriented paper https://eprint.iacr.org/2018/723 first, then the theory paper https://eprint.iacr.org/2018/725. The actual Round5 proposal to NIST is still months away.

Cheers,
- markku

On Friday, August 24, 2018 at 7:30:45 PM UTC+1, Mike Hamburg wrote:

> Dear PQC,
>
> I concur that independence of intermediate results is an important consideration, and in Round5 it may be a serious problem. I have discussed the issue with the Round5 team, and we have concluded the following.
>
> The usual errors in a LWE/LWR scheme are of the form
>
> s(As'+e') - s'(As+e) = se' - s'e
>
> where in this case s,s' are the sparse ternary secrets and e,e' are the rounding error. The calculation is done modulo the cyclotomic polynomial \Phi_{n+1}(x).
>
> If we were to instead calculate this in the NTRU ring — i.e. mod the polynomial x^{n+1} - 1 — each coefficient would be the sum of h rounding terms times +-1, where h is the fixed Hamming weight of the secret. In that case, while there might still be correlations between the magnitudes of the terms, they would probably be somewhat close to i.i.d..
>
> To reduce the resulting value mod \Phi_{n+1}(x), we take the n'th coefficient and subtract it from the others. This means that if the n'th coefficient is large, we will have a greatly increased error probability in every position. This might cause many errors, so that the error-correcting code would be overwhelmed.
>
> Concretely, we have estimated this probability in SAGE, and gotten 4-error probabilities around 2^-68, 26-56, 2^-55 for R5ND_PKE{1,3,5} respectively. This is high enough that it may allow an attack, especially if the adversary iterates to find ciphertexts with larger than expected amounts of rounding noise. I have discussed this with the Round5 team, who have confirmed the calculations. I have also confirmed this by setting s=700 in R5ND_PKE3. The same script

**From:** Leo Ducas <leo.ducas1@gmail.com>
**Sent:** Saturday, August 25, 2018 11:53 AM
**To:** pqc-forum
**Cc:** leo.ducas1@gmail.com
**Subject:** Re: [pqc-forum] OFFICIAL COMMENT: Round5 = Round2 + Hila5

PS: while sampling preserving this symmetry is rather trivial, when it comes to rounding it is a bit more tricky, yet efficiently doable. Essentially it consist of solving CVP in root lattices A_n and/or A_n*, which can be done in time n \log n.

Some pointers:
https://www.math.leidenuniv.nl/scripties/BachVanWoerden.pdf
https://arxiv.org/abs/0801.1364

Best of luck
-- Leo

Le samedi 25 août 2018 17:46:14 UTC+2, Leo Ducas a écrit :

> I think the cleanest way of doing things is to lift everything in the cyclic ring (as in NTRU) in a symmetry preserving manner. Proof and discussion on such technique (which I call the NTRU trick)
> are given in section A of https://eprint.iacr.org/2017/996.pdf.
>
> Hope this helps
>
> Best
> -- Leo
>
> Le vendredi 24 août 2018 20:30:45 UTC+2, Mike Hamburg a écrit :
>
> > Dear PQC,
> >
> > I concur that independence of intermediate results is an important consideration, and in Round5 it may be a serious problem. I have discussed the issue with the Round5 team, and we have concluded the following.
> >
> > The usual errors in a LWE/LWR scheme are of the form
> >
> > s(As'+e') - s'(As+e) = se' - s'e
> >
> > where in this case s,s' are the sparse ternary secrets and e,e' are the rounding error. The calculation is done modulo the cyclotomic polynomial \Phi_{n+1}(x).
> >
> > If we were to instead calculate this in the NTRU ring — i.e. mod the polynomial x^{n+1} - 1 — each coefficient would be the sum of h rounding terms times +-1, where h is the fixed Hamming weight of the secret. In that case, while there might still be correlations between the magnitudes of the terms, they would probably be somewhat close to i.i.d..
> >
> > To reduce the resulting value mod \Phi_{n+1}(x), we take the n'th coefficient and subtract it from the others. This means that if the n'th coefficient is large, we will have a greatly increased error probability in every position. This might cause many errors, so that the error-correcting code would be overwhelmed.
> >
> > Concretely, we have estimated this probability in SAGE, and gotten 4-error probabilities around 2^-68, 26-56, 2^-55 for R5ND_PKE{1,3,5} respectively. This is high enough that it may allow an attack, especially if the adversary iterates