Hello NTRU team
Can you say me, why you didn't keep the NTRUencrypt-1024 release, is it because of speed performance or security performance.
Best regards.

You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
Visit this group at https://groups.google.com/a/list.nist.gov/group/pqc-forum/.
Dear El Hassane Laaji,

* EL HASSANE LAAJI <e.laaji@ump.ac.ma> [2019-05-26 00:34:09 +0000]:
> Can you say me, why you didn't keep the NTRUencrypt-1024 release, is
> it because of speed performance or security performance.

Thanks for your question. To clarify for others, the "NTRUencrypt-1024" parameter set was proposed in the first round NTRUEncrypt submission for use with the ss-ntru-pke and ss-ntru-kem schemes. I'll split your question into two parts:
- Why didn't we recommend ss-ntru?
- Why didn't we recommend an NTRU variant that uses Z[x]/(x^1024 + 1)?

Regarding ss-ntru:

At a fixed security level, NTRU and LWE schemes have a trade-off triangle between
1. the correctness of the decryption procedure,
2. the width of the coefficient distributions,
3. the compactness of public keys and ciphertexts.

The second round NTRU team wanted a compact scheme with a correct decryption procedure. The coefficient distribution used in ss-ntru is not compatible with that goal.

Regarding Z[x]/(x^1024 + 1):

It's not clear to us that there's a real need for an NTRU parameter set with such a large n. The largest n that we recommend is 821.

Best,
John (on behalf of the NTRU team)

--
You received this message because you are subscribed to the Google Groups "pqc-forum" group. To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov. Visit this group at https://groups.google.com/a/list.nist.gov/group/pqc-forum/.
Hello NTRU team;
I have a remark:
In "NTRU Algorithm specification..." document, page 17 DPE_Public_Key algorithm, in operation subsection you compute: 

V0=Sq(G.f); // you compute V0 in Sq but in implementation you compute G.f in Rq.
the same for V1=Sq_inverse(V0); // you compute V1 in Sq but in implementation you compute it in Rq.
also hq=Rq(V1.f.f); you compute hq in Rq but in implementation you compute it Sq.
see function:

void owcpa_keypair(unsigned char *pk,
    unsigned char *sk,
    const unsigned char seed[NTRU_SAMPLE_FG_BYTES])
{
    ....

    poly_Rq_mul(Gf, G, f);//v0=f.G should be computed in Sq ?
    poly_Rq_inv(invGf, Gf);//v1=invGf should be computed in Sq ?
    poly_Rq_mul(tmp, invGf, f);
    poly_Sq_mul(invh, tmp, f);//hq=invh should be also computed in Rq ?
    ....
}
Can you say me which are the right operations?

Best regards
H.LAAJI
Ph.D student
Mohammed First University Morocco.

Garanti sans virus. www.avast.com
Hello El Hassane,

> poly_Rq_inv(invGf, Gf);\//v1=invGf should be computed in Sq?

> Can you say me which are the right operations?

The "poly_Rq" in the name just refers to the fact that Gf and invGf are encoded as elements of Rq. As long as Gf is not zero, our implementation of poly_Rq_inv will return invGf that satisfies invGf * Gf = 1 mod (q, Phi_n).

If Gf happens to be invertible in Rq, then our implementation of poly_Rq_inv will return invGf that satisfies invGf * Gf = 1 mod (q, Phi_1Phi_n). This condition is not necessary, which is why the specification only asks for an inverse in Sq.

Cheers,
John
Dear NTRU team;
I have some remarks:
1- In NTRU4096821 release, in file owcpa.c in owcpa_keypair(..) function.
I remark that the poly_S3_inv(invf_mod3, f); is very costly,
the speed of key generation function is about 215 ms(test performed in PC-I7 2Ghz, 8go).
I successfully reduce the cost to only 106 ms, by using another inverse function ntru_ring_inv(...);
(you find attached to this message the files modified: owcpa.c ; poly.c; poly.h )

2- Another idea, why you do not choose F=3*f+1 to void computing inverse of f mod 3.
3- In my opinion, the use of S domain is unnecessary; why you do not use only R domain.
Best regards
--
Dear El Hassane,

Thank you for these comments.

> 1- In NTRU4096821 release, in file owcpa.c in owcpa_keypair(..) function .
> I remark that the poly_S3_inv(invf_mod3, f); is very costly,

This is true. But note that the reference implementation is not currently an optimization target for us, and we are unlikely to release an optimized plain c implementation in the near future. Our main focus is on platform specific optimizations.

That said, we plan on restructuring our supercop submission to make it easier for you and others to contribute optimized subroutines for benchmarking. The "factored" implementations of Streamlined NTRU Prime should give you a sense of what this will look like.

> 2- Another idea, why you do not choose F=3*f+1 to void computing
> inverse of f mod 3.

We discuss this in the submission document. See Sections 2.4.1 and 5.2.

Best,
John

PS. It looks like you are working on an old version of our source code.
I keep a public github repository up to date with our private development repository: https://github.com/jschanck/ntru Please feel free to get in touch off-list if you'd like to learn more about where your efforts may be most useful.

--
You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/20191010183126.t5ipeksu2v23wpql%40weil.
I work on NTRU scheme that was submitted to the second round of the NIST's Post-Quantum Cryptography project in March 2019.
Is this really old release? What are the principal changes on NTRU4096821?

Le jeudi 10 octobre 2019, John Schanck <jschanck@uwaterloo.ca> a écrit :
* EL HASSANE LAAJI <e.laaji@ump.ac.ma> [2019-10-09 18:10:46 +0100]:
> Dear NTRU team;

Dear El Hassane,

Thank you for these comments.

> 1- In NTRU4096821 release, in file owcpa.c in owcpa_keypair(..) function .
> I remark that the  poly_S3_inv(invf_mod3, f); is very costly,

This is true. But note that the reference implementation is not currently an optimization target for us, and we are unlikely to release an optimized plain c implementation in the near future. Our main focus is on platform specific optimizations.

That said, we plan on restructuring our supercop submission to make it easier for you and others to contribute optimized subroutines for benchmarking. The "factored" implementations of Streamlined NTRU Prime should give you a sense of what this will look like.

> 2- Another idea, why you do not choose F=3*f+1 to void computing inverse
> of f mod 3.

We discuss this in the submission document. See Sections 2.4.1 and 5.2.

Best,
John

PS. It looks like you are working on an old version of our source code.
I keep a public github repository up to date with our private development repository: https://github.com/jschanck/ntru
Please feel free to get in touch off-list if you'd like to learn more about where your efforts may be most useful.

--
You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/CAAVXnrvajVTHgZQS_iwTU20mvVxPgot1iPKT64F8BON53LtK-Q%40mail.gmail.com.
From: John Schanck <jschanck@uwaterloo.ca>  
Sent: Friday, October 11, 2019 3:16 PM  
To: EL HASSANE LAAJI  
Cc: pqc-forum  
Subject: [pqc-forum] Re: NTRU comments

* EL HASSANE LAAJI <e.laaji@ump.ac.ma> [2019-10-11 19:49:02 +0100]:  
> I work on NTRU scheme that was submitted to the second round of the  
> NIST's Post-Quantum Cryptography project  
> <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Subm  
> isions>  
> in  
> March 2019.  
> Is this really old release? What are the principal changes on NTRU4096821?

I announced new software on this list on August 4th, and I announced further improvements to the software at the Second PQC Standardization Conference.

You can see a detailed list of changes here:  
https://github.com/jschanck/ntru/commits/master

Some commits that affect ntruhps4096821:  
* Aug 26: "Factor inversion routines out of ref-common/poly.c"  
* Aug 27: "Factor optimization targets out of poly.c"  
* Aug 29: "Update avx2-hps4096821"  
* Aug 30: "avx2-hps4096821: better poly_r2_mul"

Best,  
John

--
You received this message because you are subscribed to the Google Groups "pqc-forum" group.  
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.  
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/20191011191556.zqti2qxglwlevwrz%40weil.