NTRU Prime: modifications for round 3

20201007

NTRU Prime is a small lattice system. Subject to this constraint, our primary objective is to eliminate unnecessary complications in security review. We correctly predicted that such complications would lead to security failures in NISTPQC lattice submissions. We evaluated a variety of trapdoor functions from this perspective before submission, again during round 1, and again during round 2.

On this basis we have once again decided against decryption failures; modules; errors; and all other changes that we have considered to our family of trapdoor functions. We are therefore submitting **the same family of trapdoor functions in round 3**. NTRU Prime therefore has **an unchanged family of trapdoor functions throughout round 1, round 2, and round 3**.

Our CCA conversion includes various hashing safeguards, some already in round 1 and some added in round 2. These safeguards cost 32 bytes in ciphertext size and a considerable fraction of our CPU time. However, even with these safeguards, NTRU Prime often outperforms other small lattice KEMs. More importantly, the costs of our hashing safeguards are negligible in applications. We are therefore submitting the same CCA conversion in round 3. NTRU Prime is thus **fully compatible between round 2 and round 3, when users choose the same parameters**.

The only changes to the text of the algorithm specification (Section 2) are as follows. The statement "Multiply by v in $\mathcal{R}/3$" has been clarified to "Multiply e by v in $\mathcal{R}/3$". "CCA transforms modified" has been clearly labeled as a change for round 2. One occurrence of "$x^p - x - 1$ is irreducible in the polynomial ring $(\mathbb{Z}/q)[x]$" incorrectly said "$x^p - x - 1$ is irreducible in the polynomial ring $\mathcal{R}/q$"; this has been corrected. There is now a note "some objects in the tables are defined later" regarding the tables of notation.

There are many updates elsewhere in the documentation. There are several additional pages of analysis in the rationale; several additional pages of analysis of parameter selection; and, in the analysis of advantages, a summary of six new attack papers published after the beginning of round 2. There are several additions to the security analysis reflecting exciting new enumeration speedups, a new analysis of constant factors in the cost of memory, and the recent collapse of the idea (which we had already criticized as being unjustified) that $(3/2)^{\beta/2}$ is a "conservative lower bound" on the number of pre-quantum operations used in sieving. Our security-estimate script is updated and has been applied to an even wider range of parameters, producing new graphs and tables. The performance analysis shows various performance improvements, including new Haswell speeds (e.g., 166000 Haswell cycles for Streamlined NTRU Prime key generation, integrated into TLS 1.3, OpenSSL, and a web browser), new Cortex-M4 speeds, and a complete new constant-time FPGA implementation.

The updated security estimates do not affect our Core-SVP calculations ("non-hybrid sieving free", pre-quantum and post-quantum). The Core-SVP results from our script are the same as in round 2. Errors in the widely used "Estimate" scripts led to Core-SVP miscalculations for SABER; those errors did not appear in our script.

Parameter selection. We are concerned that pre-quantum Core-SVP levels 2^{100}, 2^{106}, and 2^{111}, proposed for category 1 for Dilithium, NTRU, and Kyber respectively, will turn out to be inadequate against generic lattice attacks. We will not add dimensions below our 653 (pre-quantum Core-SVP 2^{129}). We recommend our original dimension 761 (pre-quantum Core-SVP 2^{153}) for an extra security margin.

We have seen various requests for larger dimensions, even larger than our dimension 857 (pre-quantum Core-SVP 2^{175}). To accommodate these requests and prevent any accusations of a lack of flexibility, we have now added some **larger dimensions as a supplement to our current dimensions**. See the documentation for a full description of the selection process and the resulting dimensions.

We have considered adding intermediate parameter sets to further illustrate NTRU Prime's flexibility, showing that NTRU Prime offers even larger advantages in security level under various size limits compared to, e.g., Kyber. The call for proposals explicitly allowed multiple parameter sets per category. However, NIST has recently made an announcement that seems to discourage "too many parameter sets", and has not answered the question of what "too many" means.

Software. The only changes in the reference C software are as follows:

- Adding the new parameter sets.
- Porting the NTRU LPRime software to big-endian CPUs (a few extra lines in the `Expand` function).
- Namespacing (8 extra lines in `.h` files), as required for SUPERCOP and for wide deployment.
- Following the SUPERCOP naming for subroutines (symmetric primitives and sorting), and moving the subroutines into `ref/subroutines`.
- Renaming `Hash` as `Hash_prefix` for clarity.
- Removing an unnecessary `extern`.
- Aborting if `crypto_stream_aes256ctr` fails (meaning that OpenSSL ran out of memory; note that there are alternative `aes256ctr` implementations that cannot fail).

The round-2 KATs are identical to the round-3 KATs for the same parameter sets.

Our round-3 Sage reference implementation `ntruprime.sage` is included in the submission as supporting documentation. This implementation includes all current parameter sets. There is a close match between the structure of the specification, the structure of the Sage reference implementation, and the structure of the C reference implementation, as in round 2. For continuity, the Sage implementation also supports an option to use the round-1 parameter sets, and is structured to help reviewers see that this option uses exactly the same trapdoor functions. The Sage implementation matches the C implementation (and the round-1 C implementation) in various tests, including SUPERCOP's checksums.