If I understand correctly, the current version of ac-main.pdf was completed on 2018. And I assume there is no update on your draft since 2019. Could you confirm this?

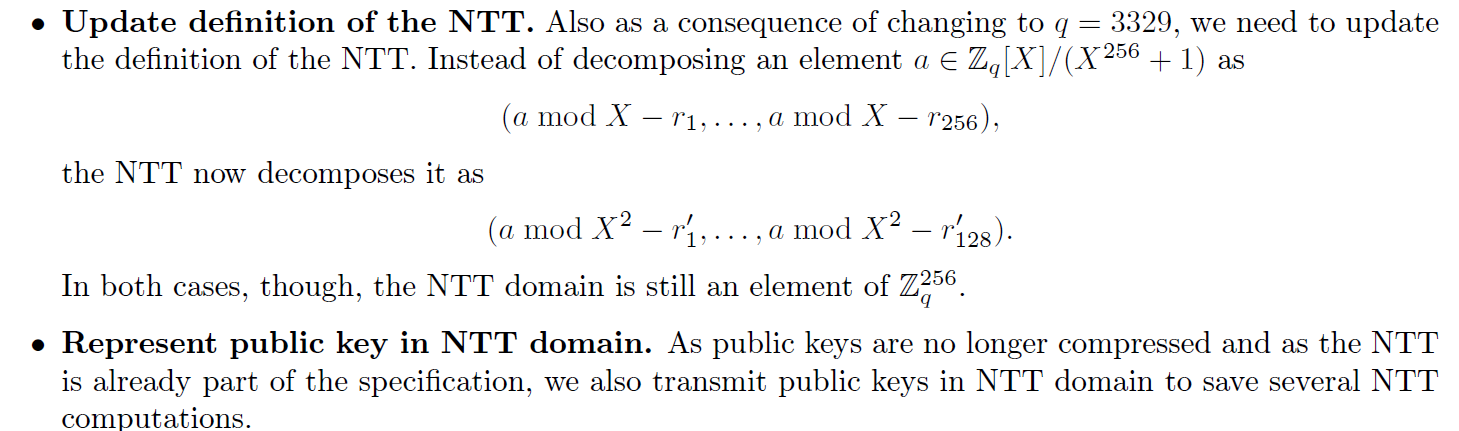
[NewHope]

After the announcement of round 1 winners, there are two updates for NewHope: 2019-04-10: Version 1.02 and 2019-07-10: Version 1.03. Did you already aware the difference from the version change that will cause the previous attack in ac-main.pdf to fail? I have not taken a close look into it yet. If you know, could you specify that difference for me?

I already checked the latest specification in 2019-07-10: Version 1.03. From the errata, I found that there is no significant update on the NTT of NewHope. Do this mean there is a high probability that your previous fault analysis can work well on the latest version?

[KYBER]

After the announcement of round 1 winners, I think there is no major revision on the NTT part of KYBER. The only difference I found from the change log is listed as below. In my opinion, it just updated the expression and did not change the implementation of NTT. I assume your previous fault analysis can directly work on the latest version of KYBER, am I right?



[Dilithium]

The round 2 version of Dilithium had a significant change. As the website says: “**As an update for round 2 of the NIST project we propose a variant of Dilithium, called Dilithium-AES, that uses AES-256 in counter mode instead of SHAKE to expand the matrix and the masking vectors, and to sample the secret polynomials.**” Due to this change, I assume your previous attack method may not work or it may need to adjust accordingly? Shall we start to work this target first?

The specification also highlight such change, which can also be found in the public website: “**The other changes were in the implementation. We made various optimizations in the signing algorithm. The most important optimization is how the rejection condition based on the low part of the vector w and the hint vector is computed. Our AVX2 optimized implementation now makes more use of vectorization and includes a simpler assembler NTT implementation using macros.**”

As for me, I think the implementation of the NTT does matter the fault analysis.

After I read your version, I think I understood where the fault is injected and what impact such fault will bring. But I haven’t fully understood how you conduct the fault analysis and how to recover the key. I may need more time to explore the attack.

At your convenience, could you comment what I have concluded and give me some guidance. We can leave the discussion of physical experiment later.

Bolin