From: 赵运磊 <ylzhao@fudan.edu.cn>
Sent: Wednesday, December 27, 2017 6:36 AM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: OKCN/AKCN/CNKE

Follow Up Flag: Follow up
Flag Status: Flagged

We are the owner of the proposal. Originally, we used the name ``OKCN/AKCN/CNKE" for our proposal, as OKCN, AKCN, CNKE (together with SEC and E_8 lattice-code for error correction) are the key building blocks of this proposal. But the name of ``OKCN/AKCN/CNKE" is lengthy and is inconvenient for reference. For simplicity, we would like the proposal to be called KCL that stands for ``Key Consensus from Lattice".

Best regards
Yunlei

From: Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
Sent: Saturday, May 26, 2018 2:23 PM
To: pqc-forum
Subject: [pqc-forum] OFFICIAL COMMENT: Bit-flip ciphertext and key malleability in AKCN and OKCN

Hi,

Let's recall the basic notation for a KEM:

- Alice computes (PK, SK) = KeyGen() and sends public key PK to Bob.
- Bob computes (CT, K) = Encaps(PK) and sends ciphertext CT to Alice.
- Alice computes shared key K = Decaps(SK, CT).

We informally define two forms of malleability:

1. Ciphertext malleability (CT-Mal): A traditional definition of malleability is that it is possible to transform ciphertext CT to another ciphertext CT' so that K' = Decaps(SK, CT') is related to K = Decaps(SK, CT).

2. Public key malleability (PK-Mal): A second form of malleability occurs if a transformed public key PK' will yield transformed ciphertext (CT', K) = Encaps(PK') *and* decryption will yield K' = Decaps(SK, CT') where K and K' are related.

We note that there may be multiple distinct representations of the same public key (resulting in exactly same ciphertext). Also many algorithms apply error correction, and are tolerant to some modifications of ciphertext. Therefore we say that K is "related" to K' only when K != K'.

I subjected most of the NIST PQC Project KEM candidates to a trivial experiment where a single bit is flipped in either public key or ciphertext, and the Hamming distance of the resulting shared secrets is observed (expected value is n/2). It is easy to assign a P value to the distances based on Chi^2 statistic.

Most candidates apply a hash function (or a similar mechanism) to remove observable biases in K vs K', but not all. Turns out that sometimes a simple bit flip in PK or CT will result in easily recognizable change in K. In case of AKCN and OKCN, the change in shared secret can be as small as a single bit (or none). There are many real-life protocol scenarios where this can be very dangerous.

Tabulating the findings:

AKCN-MLWE:
- CT-Mal - Extreme (distance often 1)
- PK-Mal - Significant (distance often small <= 12)

AKCN-SEC:
- CT-Mal - Extreme (distance often 1)
- PK-Mal - Extreme (distance often 1)

OKCN-MLWE:
- CT-Mal - Extreme (distance often 1)
- PK-Mal - Extreme (distance often 1)

OKCN-SEC:
- CT-Mal - Extreme (distance often 2)
- PK-Mal - Extreme (distance often 1)

LIMA-EncapCPA:

CT-Mal - Extreme (distance often 1)

CFPKM:

CT-Mal and PK-Mal: Extreme. This cipher is broken in too many other ways as well.

Cheers,
- markku

--

From: 赵运磊 <ylzhao@fudan.edu.cn>
Sent: Saturday, June 16, 2018 11:29 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov; mjos.crypto@gmail.com
Subject: OFFICIAL COMMENT: KCL (pka OKCN/AKCN/CNKE) Re: Bit-flip ciphertext and key malleability in AKCN and OKCN

Dear Markku-Juhani:
Thanks for your comments, and sorry that I just noticed your comment.

I suggest what you pointed out on key malleability is a common issue applied to almost all KC/AKC mechnisms for KEM from LWE and its variants, and actually should be viewed as an advantage of OKCN/AKCN. It has little effect in practice.

Recall that for both KC and AKC, the communicating two parties are trying to get key consensus from key materails with noise. The noise can be just from the LWE problem, but also can be transmit errors (just the scenarios you considered as key malleating attack). The design goal of all KC/AKC is to overcome such errors to a maximum extent. For KC, we proved that the upper-bound has to obey: $2kd<q(1-1/g)$. In other words, when q (dominating security leverl), g (dominating bandwidth), and k (dominating key length) are fixed, the maximus distance d (dominating error correction) between the key material of Alice and that of bob, has to obey: $d<[q(1-1/g)]/2k$. This a rule that any KC has to obey. Similarly, we proved that for any AKC, it is: $d<[q(1-k/g)]/2k$.

Both OKCN and AKCN (almost) achieved this optimal upperbound, which means that they have the (almost) maxium ability of error correction (as least correcting one-dimension error is considered). For error correction in D4 and E8, they can perform ever better.

So, we suggest what you pointed out are actually the advantageous feature of OKCN/AKCN (specfically, its optimality in error correction), and a common issue related key KC/AKC from LWE and its variants.

When using a KEM (like Diffie-Hellman) for actual authenticated key exchange, the actual session-key is derived from the resultant key of OKCN/AKCN and (part or whole of) the transcript. This is just the case of TLS1.3 (where its session-key is dependent upon the whole transcript). This is also just the philosopy of our identity-concealed non-malleable AKE: CNKE-MLWE (specified in pages 68-69 of our proposal).

BTW, KC-based KEM corresponds to Diffie-Hellman in the lattice world, while AKC-based to El Gamal (or key transport). We note that, in TLS1.3, key transport is explicitly abandoned. From our view, KC-based is more versatile, and is more compatible with TLS1.3.

Thanks!
Yunlei

From: 赵运磊 <ylzhao@fudan.edu.cn>
Sent: Wednesday, December 05, 2018 11:36 AM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: On the Desirability and Preferability of OKCN-KEM vs. AKCN-KEM

Dear All:

Recently, we show an additional application of OKCN developed with our KCL-KEM proposal. Specifically, we show that the deterministic version of OKCN (where the random e is fixed to be zero) can be used to generalize and optimize Dilithium (one of the most promising lattice-based signature proposals to NIST). This further justifies and highlights the desiarability of OKCN as the same routine can be used for both KEM and signatures, which is useful to simplify the system complexity of lattice-based cryptography. This result is now available from https://eprint.iacr.org/2018/1180

With is new observation, we would like to summarize the desirability/preferability of OKCN-KEM over AKCN-KEM.

(1) KC-based KEM corresponds to Diffie-Hellman key exchange in the lattice world, while AKC-based to El Gamal key transport.

(2) When deploying AKC-based KEM in practice, if the randomness used by the responder (e.g., a low-power device like smart card) is poor, it will significantly ruin the session-key security. In comparison, with KC-based KEM, the two players play a symmetric role in generating the session-key, and thus the damage caused by poor randomness can be alleviated. In particularly, symmetry is usually a desirable feature for cryptographic schemes in practice.

(3) On the same parameters (q,m,g) (which imply the same bandwidth), OKCN-based KEM has lower error probability than AKCN-based. Or, on the same parameters (q,m,d) (which imply the same error probability), OKCN-based KEM has smaller bandwidth than AKCN-based. This comparison is enabled by the upper-bounds on these parameters proved in our KCL proposal.

(4) KC-based KEM is more versatile, in the sense that it can also be straightforwardly adapted into a key transport protocol or a CPA-secure PKE scheme. And in the above recent work, we show that the deterministic version of OKCN is also a fundamental building tool for lattice-based signature.

(5) KC-based KEM is more appropriate for incorporating into the existing standards like IKE and TLS that are based on Diffie-Hellman via the SIGMA mechanism. We note that key transport is explicitly abandoned with TLS1.3.

Best regards
Yunlei