From: Keita Xagawa <xagawa.keita@lab.ntt.co.jp>
Sent: Tuesday, January 09, 2018 11:42 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: LEDAkem

Dear authors, deal all,

The current specification (and implementation) of LEDAkem seems to fail to achieve CCA security.
LEDAkem tries to construct an IND-CCA-secure KEM by applying the conversion in [30] to a OW-CPA-secure deterministic PKE.
The authors would not notice the chosen-ciphertext attacks in [A1] and [A3, Appendix K] against KEM/Hybrid PKE in [30].

LEDAkem
==========
* The public key is M in F_2^{p x n}.
* The encapsulation algorithm chooses e <- F_2^n with HW(e) = t, and outputs a ciphertext s = M e^T and a session key K = KDF(e).
* The decapsulation algorithm recovers e from s by using the secret key and outputs k_s = KDF(e). If s is invalid, the decapsulation algorithm returns a "pseudorandom" key k_s = KDF(s).

The footnote 1 of [30] suggests k_s = KDF(s), which is not pseudorandom.

Chosen-Ciphertext Attack against the current LEDAkem ========== The following CCA exists even if the scheme is perfectly correct. See [A1] and [A3, Apendix K].
For i = 0,...,n-1, let u_i be the i-th unit vector of dimension n.

* Assume that a ciphertext s = M e^T is given and assume that e[0] = 0.
* For i = 1, ..., n-1, we query s_i = s + M {u_0+u_i}^T and obtain the result.
* Set e[i] = 0 if k_s == KDF(s_i); else set e[i] = 1.
* Compute K = KDF(e)

If e[i] = 1, then HW(e + u_0 + u_i) = t. On the other hand, if e[i] = 0, then HW(e + u_0 + u_i) = t + 2 > t.
This breaks the onewayness of KEM.

Note
==========
If DFR is 0, it is easy to fix the problem.

* Persichetti's thesis suggests to use KDF(s'), where s' = L_{n_0-1}^{-1} s in the LEDAkem context.
* [A1] and [A2] suggests to use KDF(\pi(s)), where \pi is a random permutation. Notice that this \pi should be pseudorandom. Otherwise, one can still check if a ciphertext is valid or invalid by checking the answer is random or deterministic.
* [HHK17] and [SXY17] suggests to use Hash(secret-seed,s) (or KDF(secret-seed,s)).

[30]: Edoardo Persichetti:
"Secure and Anonymous Hybrid Encrytpion from Coding Theory" in PQCrypto 2013
[A1]: Pierre-Louis Cayrel, Cheikh Thiecoumba Gueye, El Hadji Modou Mboup, Ousmane Ndiaye, and Edoardo Persichetti:
"Efficient Implementation of Hybrid Encryption from Coding Theory" in C2SI 2017
[A2]: Edoardo Persichetti:
"Code-based Key Encapsulation from McEliece's Cryptosystem" in MACIS2017
[A3]: Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, Christine van Vredendaal:
"NTRU prime: reducing attack surface at low cost" in SAC 2017.
(http://eprint.iacr.org/2016/461)

Regards,
Keita Xagawa

From: Gerardo Pelosi <gerardo.pelosi@polimi.it>
Sent: Wednesday, January 10, 2018 12:31 PM
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: LEDAkem

Dear Keita,

thank you for the observations provided and the clarity in pointing them out.

In the LEDAkem supporting documentation Sect. 2.4, we claimed that LEDAkem provides the NIST required IND-CPA security and that it is possible to adapt it to achieve IND-CCA security employing a KDF to hide the case where a decoding failure (in your observation, induced by the
attacker) takes place.

We acknowledge that the KDF to provide such a guarantee requires the addition of a secret bitstring of some kind as a parameter to the KDF, as pointed out by Persichetti's thesis, [A1], [A2] (can you provide references for [HHK17] and [SXY17] to us privately?), and this is not pointed out in the specification and reference implementation.

While the presence of a secret bitstring has no impact on the IND-CPA security of the scheme in a case where ephemeral keys are used, it provides a useful resiliency feature in case of accidental key reuse, in addition to allowing to achieve IND-CCA.

We are thus planning to provide an up-to date specification document including this clarification, and adapt the implementation accordingly on our website as soon as possible.

Kind Regards,
--the LEDAkem team

Il 01/10/18 05:42, Keita Xagawa ha scritto:
> Dear authors, deal all,
>
> The current specification (and implementation) of LEDAkem seems to
> fail to achieve CCA security.
> LEDAkem tries to construct an IND-CCA-secure KEM by applying the
> conversion in [30] to a OW-CPA-secure deterministic PKE.
> The authors would not notice the chosen-ciphertext attacks in [A1] and
> [A3, Appendix K] against KEM/Hybrid PKE in [30].
>
> LEDAkem
> ==========
> * The public key is M in F_2^{p x n}.
> * The encapsulation algorithm chooses e <- F_2^n with HW(e) = t, and
> outputs a ciphertext s = M e^T and a session key K = KDF(e).
> * The decapsulation algorithm recovers e from s by using the secret
> key and outputs k_s = KDF(e). If s is invalid, the decapsulation
> algorithm returns a "pseudorandom" key k_s = KDF(s).
>
> The footnote 1 of [30] suggests k_s = KDF(s), which is not pseudorandom.
>
>
> Chosen-Ciphertext Attack against the current LEDAkem ========== The