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 \( F_{2^{p \times n}} \).
* The encapsulation algorithm chooses \( e \leftarrow F_{2^n} \) 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, Appendix K]. For \( i = 0, \ldots, n-1 \), 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 \( i = 1, \ldots, n-1 \), 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 \( s' = L_{(n_0-1)} \) 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 \( Hash(\text{secret-seed},s) \) (or \( KDF(\text{secret-seed},s) \)).
[30]: Edoardo Persichetti:
"Secure and Anonymous Hybrid Encryption 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:
(http://eprint.iacr.org/2016/461)

Regards,
Keita Xagawa