Algorithm Specifications And Supporting Documentation

1. **A complete written specification**: See attached paper RLCEspec.pdf with the totle:
   - *RLCE Key Encapsulation Mechanism (RLCE-KEM) Specification*
2. **A detailed performance analysis**: The details could be found in Section 9 "Appendix B: RLCE Performance evaluation (Informative), pages 38--40" in the paper RLCEspec.pdf
3. **Known Answer Test values**: See the KAT directory on digital media
4. **A thorough description of the expected security strength**: The details could be found in Table 2 (page 7) of Section 3 in the paper RLCEspec.pdf
5. **An analysis of the algorithm with respect to known attacks**: The details could be found in Section 8 "Appendix A: RLCE Security Analysis (Informative), pages 25--38" in the paper RLCEspec.pdf
6. **A statement of advantages and limitations**:
   - **Advantages:**
     i. The proposed RLCE scheme has smaller public keys compared with Goppa code based McEliece scheme. For example, for the equivalence of AES-128 security, binary Goppa based McEliece scheme has a public key of 188KB while RLCE has a public key size of 115KB. For the equivalence of AES-192 security, binary Goppa based McEliece scheme has a public key of 490KB while RLCE has a public key size of 280KB.
     ii. Goppa code based McEliece scheme has the assumption that Goppa codes behave like random codes while RLCE does not have this kind of assumption for the underlying linear code.
     iii. RLCE scheme is based on widely deployed Reed-Solomon codes. RLCE scheme is very efficient for encryption and decryption. The efficiency for RLCE could be further improved using Reed-Solomon hardware decoders and vector instructions. It is noted that the most expensive parts in RLCE primitives (key generation, encryption, and decryption) are Reed-Solomon coding/decoding operations and echelon computation for large matrices (which is essentially vector operations). Industry has extensive experience on speeding up these primitives using hardware.
     iv. Industry has extensive experience in implementing Reed-Solomon code on 8-bit processors (e.g., smartcards), voice applications, satellite applications, and other constraint environments with low power and constrained memory. RLCE schemes could be easily deployed in these environments.
     v. Though some McEliece based encryption schemes (e.g., MDPC/LDPC code based McEliece scheme) may have short public key size or short cipher texts, their security strength depends on stronger assumption that certain structured codes are hard to decode. For the proposed RLCE scheme, the generator matrix is completely randomized. To decrypt a cipher-text without the private key (randomization parameters), it is expected to be as hard as decoding a random linear code which is NP-hard. In other words, the RLCE's security does not depends on any specific structure of underlying linear codes, instead its security is believed to depend on the NP-hardness of decoding random linear codes.

vi. The security of the RLCE scheme is thoroughly analyzed in Section 8 “Appendix A: RLCE Security Analysis (Informative), pages 25--38” in the paper RLCEspec.pdf.

- **Limitations:**
  i. RLCE public key sizes are still large. For AES-128 equivalent security, the public key size is 110KB and the cipher text is 785 bytes. For AES-192 equivalent security, the public key is 280KB and the cipher text is 1238 bytes