Layered ROLLO-I: Open Source Code Guideline

작성자: 김찬기(조선대, 정보통신공학부)

최종 업데이트: 22/10/27

1. 주요 특징

* 해당 코드는 랭크 기반 부호의 LGRL 라이선스 오픈 소스로 공개된 Rank-based-cryptography (RBC) [1]을 기반으로 하되, 기존 RBC 부호에서 지원되지 않는 연산들을 추가하거나 관련된 코드을 수정하여 제작
* Rank-based-cryptography (RBC)는 기본적으로 C 기반으로 랭크 기반의 Maximum rank distance (MRD) 혹은 Low-rank parity-check (LRPC) 부호에 기반한 암호 기법인 RQC [2] 및 ROLLO [3] 의 Optimized implementation 수준의 구현 및 관련 라이브러리를 생성할 수 있는 코드임. 다만, 해당 변형에서는 주요 비교 대상인 ROLLO-I 및 제안 기법에 대해서만 생성되도록 수정함 [1]
* 성능향상에 주요한 부분인 RSR 복호 알고리즘 및 다항식 환 연산 알고리즘은 RBC 내 라이브러리를 대부분 수정없이 활용하였으나, 다항식 환 및 간의 연결성을 위한 추가적인 함수들이 몇몇 정의되어 있음
* 기존 코드인 ROLLO-I도 같이 생성되며, AVX-2 인스트럭션을 기반으로 생성하였으며, 동작 시 같은 보안 수준에서는 프로세싱 사이클 값 기준 매우 일관적인 성능 향상을 확인할 수 있음
* RBC 및 구현 알고리즘은 대부분의 동작이 상수 시간 동작을 고려하여 구현하였으나, 구현 상에서 상수 시간을 방해하는 요소가 완전히 배제되어 있지는 않으며, 추후 코드 개선을 통해 구현 예정

(예를 들어, Constant-time 동작을 방해하는 원인인 조건 분기문을 완전히 없애지는 못함)

* Known answer test (KAT)를 지원하며, 빌드 생성 과정에서 KAT 관련 실행 파일도 같이 빌드됨(2장 Getting started 참고)

1. Getting Started

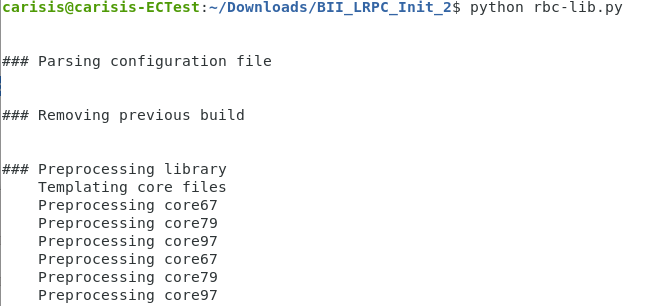
시험 환경: x86 12th Gen. Intel Core i9-12900K multi-core CPU with 32GB 4,800MHz DDR5 memory. Ubuntu Linux 20.04

* AVX-2 지원 환경의 경우에만 동작하도록 구현되어 있음
* 필요 라이브러리: Python3, python3-yaml, cmake, make, gcc, openssl

(기타 aes, minuit, randombyte 라이브러리 등을 활용하나 따로 설치 없이 백엔드에 포함)

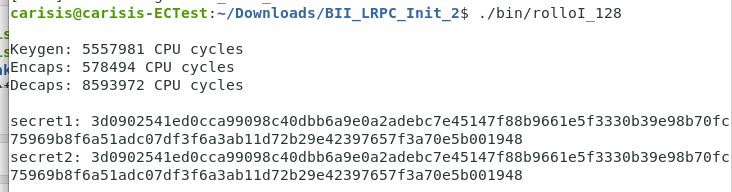
라이브러리가 설치 되지 않은 경우 apt 명령어를 이용하여 설치



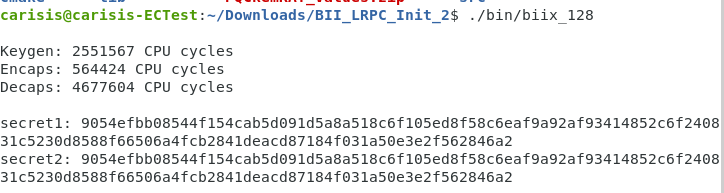
* 빌드: 설치 이후, 해당 디렉토리 내 터미널을 실행하고, `python rbc-lib.py’ 실행을 통해 빌드 진행
* 

빌드 이후 단일 (Keygen,encap,decap) 동작 Cycle 속도를 측정하고자 할 때 아래 실행 파일 실행

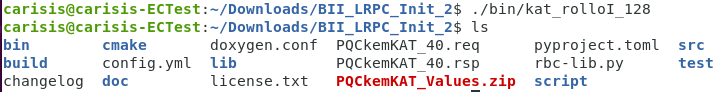
* + ROLLO-I의 경우: ./bin/rolloI\_XXX(128,192,256 중 선택)



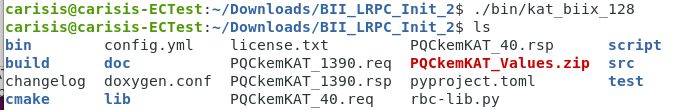
* + 제안 알고리즘의 경우: ./bin/biix\_XXX(128,192,256 중 선택)



* KAT 파일을 생성하고자 할 때, 실행 파일
  + ROLLO-I의 경우: ./bin/kat\_rolloI\_XXX(128,192,256 중 선택)으로 실행. 그 이후, PQCKem\_40\_kat.req 및 PQCKem\_40\_kat.rsp 파일 생성



* + 제안 알고리즘의 경우: ./bin/kat\_biix\_XXX(128,192,256 중 선택) 으로 실행. 그 이후, PQCKem\_(secret 키 길이)\_kat.req 및 PQCKem\_(secret 키 길이)\_kat.rsp 파일 생성

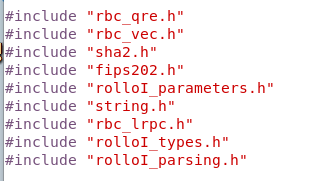


1. 코드 알고리즘 소개

- 대부분의 알고리즘 및 원리는 [1]에 설명된 Manual을 참고하면 되며 해당 구현 과정에서 기존 알고리즘 대비 추가된 부분을 위주로 설명하고자 함

* + ROLLO-I 내 암호 동작이 정의되어 있는 코드: ./src/schemes/rolloI/rolloI\_kem.c

사용 헤더:



주요 함수:

* + - rolloI\_keygen(unit8\_t\* pk, unit8\_t\* sk): 키 생성
    - rolloI\_encaps(unit8\_t\* ct, unit8\_t\* ss, unit8\_t\* pk)
    - rolloI\_decaps(unit8\_t\* ss, const unit8\_t\* ct, const unit8\_t\* sk)
  + 제안 알고리즘 내 암호 동작이 정의되어 있는 코드: ./src/scheme/biix/biix\_kem.c

사용 헤더:

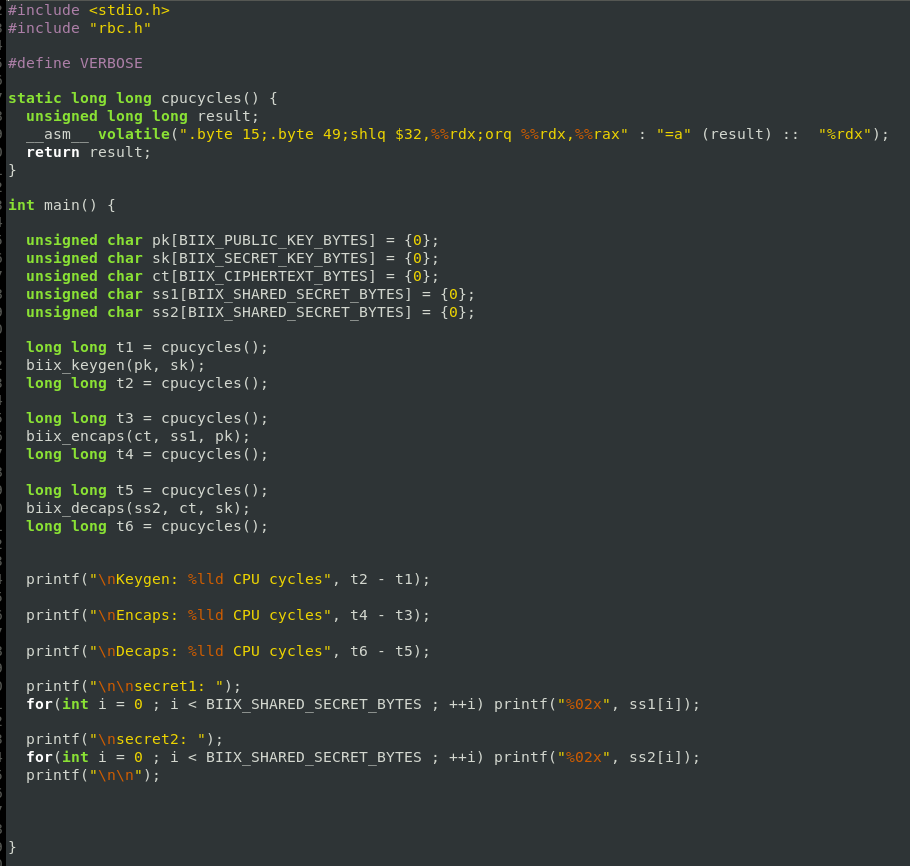


주요 함수:

* + - biix\_keygen(unit8\_t\* pk, unit8\_t\* sk): 및 모듈러 상에서 키 생성 연산 수행
      * biix\_sk\_generate(-): 모듈러 상에서의 연산 및 생성)만을 분리
    - biix\_encaps(unit8\_t\* ct, unit8\_t\* ss, unit8\_t\* pk): 및 모듈러 상에서의 encapsulation 연산 수행
      * biix\_encaps\_errgen(-): 모듈러 상에서의 연산(E support 생성)만을 분리
    - biix\_decaps(unit8\_t\* ss, const unit8\_t\* ct, const unit8\_t\* sk): 및 모듈러 상에서의 decapsulation 연산 수행
    - biix\_decaps\_multisym(-): 모듈러 상에서의 decapsulation( 계산) 연산 수행

그 외 및 모듈러 내부 변환 및 기존 연산 호환을 위해 rbc\_qre\_mod\_sparse\_from\_string(-), rbc\_qre\_set\_arb\_random(-), rbc\_qre\_set\_arb\_random\_apri\_from\_support(-) 함수가 정의됨

* + ROLLO-I 내 암호 동작 사이클 측정 코드: ./test/example/rolloI/main.c
  + 제안 알고리즘 동작 사이클 측정 코드 : ./test/example/biix/main.c



* + ROLLO-I 및 제안 내 암호 동작 사이클 측정 코드: ./test/kat/kat/main.c

1. 참고 문헌

[1] N. Aragon, S. Bettaieb, L. Bidoux, Y. Connan, J. Coulaud, P. Gaborit, and A. Kominiarz. ``The rank-based cryptography library,'' *Code-Based Cryptography Workshop (CBCrypto),* Munich:Germany, Jun 21-22 2021.

[2] Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Olivier Blazy Loïc Bidoux, Jean-Christophe Deneuville, Philippe Gaborit, Gilles Zémor, Alain Couvreur, and Adrien Hauteville. NIST PQC submisssion : Rank Quasi-Cyclic (RQC), 2017.

[3] Rollo, ``ROLLO specification,'' Available on [online], https://pqc-rollo.org/documentation.html, 2020.