과제 #2
미니 RSA 구현

목표

알고리즘 내부에서 다루게 될 파라미터들의 길이가 64bit를 넘지 않는 간단한 RSA를 구현해 본다.
이렇게 구현된 RSA는 키의 길이가 너무 짧지만, 알고리즘의 동작 원리를 이해하기에는 충분하다.

구현요약

사용되는 숫자 값의 크기가 최대 64bit를 넘지 않는 mini RSA를 구현한다.
과제는 c언어로 작성하도록 하며, 제한사항에 주어진 조건들을 만족하도록 구현한다.
코드 수행 중 값들이 정상적으로 연산되는지 확인하기 위한 중간 처리 과정을 보여준다.

제한사항

- 구동 및 테스트 환경은 Linux 환경에서 gcc를 사용하여 컴파일 하는 것을 기본으로 하며, Windows 에서는 Cygwin을 설치하여 컴파일을 수행하도록 한다.(MS Visual Studio 이용해 작성하지 않도록 한다.)
- 파라미터들의 기본 자료형으로써 8byte 'long long int'를 'llint'로 정의하여 사용하도록 하며, 이 보다 더 큰(8byte 이상의) 자료형은 사용하지 않도록 한다.
- 주어진 뼈대코드에서 임의 수 생성을 위한 RNG(Random Number Generator)는 rsa.h 파일에 구현 $0 \leqq r \leqq 1$ 사이의 값을 duoble 형으로 반환해주는 'WELLRNG512a' 함수를 사용하도록 한다.
- 전체 알고리즘에서 사용되는 나눗셈, 나머지(모듈러) 연산을 c언어에서 지원하는 연산자('/', '%')를 사용하지 않고 비트 연산으로 처리하도록 한다.
- 거듭제곱 연산을 할 때 'square and multiply' 알고리즘을 이용하여 빠르게 연산 되도록 한다.
- 모듈러 n 값이 $2^{53} < n < 2^{64}$(64bit 수)가 되도록 두 소수 p, q를 임의로 선택한다.
- p, q는 Miller-Rabin 소수 판별법과 같은 확률적인 방법을 사용하여, 이론적으로 4N(99.99%) 이상 되는 값을 선택하도록 한다.
- 조건을 만족하는 적절한 e값을 임의로 선택하여 사용하고, e의 mod Phi(n)에서 역수 d를 찾는 방법은 확장 유클리드 알고리즘을 사용하도록 한다.
- 키 생성에 성공하면 (e, n)이 공개키가 되고 (d, n)이 개인키가 되도록 하여 암•복호화에 사용한다.
- 암•복호화에 사용되는 데이터는 64bit 한 개의 데이터블록을 사용하도록 한다.
- 데이터의 값이 모듈러 n값 보다 큰 경우엔 암•복호화할 수 없으므로 오류로 처리한다.

참고사항

- 타인의 코드를 전체 혹은 일부 사용하여 작성하는 경우에는 이유 불문하고 상호 F학점으로 처리한다.
- mini RSA에 관한 이론적인 기준은 수업 PPT를 중심으로 위의 제한사항을 참고하도록 한다.
- 모듈러 곱 연산 간 발생할 수 있는 오버플로우에 대한 처리가 되어 있어야 한다.
- 다른 함수들 또한 입출력 파라미터를 지정해두었으나, 구현과정에서 추가 혹은 삭제, 수정하여 사용 가능하다.
- 개인키와 공개키를 위한 e,d,n 값과 이를 검산하기 위한 값 또한 출력 결과물에 포함되어 있어야 한다.
- 모든 입력과 출력에 대한 예외처리가 되어 있어야 한다.

제출물

미니 RSA 소스코드와 테스트 실행 결과가 담긴 파일을 마감기한 전까지 “학번_이름_miniRSA.zip”의 형태로 압축한 뒤 “암호학_과제_#2_학번_이름”의 제목으로 [ehdals614@naver.com]에 제출한다.

마감기한

- 마감일 : 2019년 11월 10일 일요일 자정까지
- 페널티 : 마감일을 넘겨 제출할 경우, 해당 과제의 최고점부터 채점을 시작하여 하루 단위로 20%씩 감점된다.

코드 수행 예시

```
mRSA key generator start.

random-number1 727626063 selected.
727626063 is not Prime.

random-number1 3584435539 selected.
3584435539 is not Prime.

random-number1 3595317721 selected.
3595317721 is not Prime.

random-number1 3659172347 selected.
3659172347 may be Prime.

random-number2 2527772237 selected.
2527772237 is not Prime.

random-number2 3243076043 selected.
3243076043 is not Prime.

random-number2 2230938491 selected.
2230938491 is not Prime.

random-number2 2698217407 selected.
2698217407 is not Prime.
```

```
random-number2 371601529 selected.
371601529 is not Prime.

random-number2 1354693547 selected.
1354693547 is not Prime.

random-number2 3098596737 selected.
3098596737 is not Prime.

random-number2 1905010131 selected.
1905010131 is not Prime.

random-number2 566466309 selected.
566466309 is not Prime.

random-number2 2301208841 selected.
2301208841 is not Prime.

random-number2 1839796327 selected.
1839796327 may be Prime.

finally selected prime p, q = 3659172347, 1839796327.
thus, n = 6732131843870569469
```

```
e : 1923581200907737 selected.
GCD(3641129127469482876, 1923581200907737)
GCD(1923581200907737, 1713495352044472)
GCD(1713495352044472, 210085848863265)
GCD(210085848863265, 32808561138352)
GCD(32808561138352, 13234482033153)
GCD(13234482033153, 6339597072046)
GCD(6339597072046, 555287889061)
GCD(555287889061, 231430292375)
GCD(231430292375, 92427304311)
GCD(92427304311, 46575683753)
GCD(46575683753, 45851620558)
GCD(45851620558, 724063195)
GCD(724063195, 235639273)
GCD(235639273, 17145376)
GCD(17145376, 12749385)
GCD(12749385, 4395991)
GCD(4395991, 3957403)
GCD(3957403, 438588)
GCD(438588, 10111)
GCD(10111, 3815)
GCD(3815, 2481)
GCD(2481, 1334)
GCD(1334, 1147)
GCD(1147, 187)
GCD(187, 25)
GCD(25, 12)
GCD(12, 1)
GCD(1, 0)

d : 292053776150736181 selected.
```

```
e, d, n, pi_n : 1923581200907737        292053776150736181      3641129131357442029     3641129127469482876
e*d mod pi_n  : 1

0. Key generation is Success!
 p : 1572593707
 q : 2315365447
 e : 1923581200907737
 d : 292053776150736181
 N : 3641129131357442029

 input data : 2018915346
 output data : 3572564566883167206
1. plain text : 2018915346
2. encrypted plain text : 3572564566883167206

 input data : 3572564566883167206
 output data : 2018915346
3. cipher text : 3572564566883167206
4. Decrypted plain text : 2018915346

RSA Decryption: SUCCESS!
```