[CSE3029] 암호학 과제 #2

mini RSA 구현

2017-11-14

1. 목표

- 간단한 RSA를 구현해보며 RSA 암호화 알고리즘의 동작 원리를 이해한다.

2. 문제

- 제한사항에서 제시한 함수를 작성하여 AES 암·복호화 알고리즘을 구현한다.
- 과제의 RSA는 기존 RSA와 다르게 알고리즘 내부에서 다루게 될 파라미터들의 길이가 32비트를 넘지 않는다.
- 과제는 C언어로 작성하도록 하며, 코드 수행 중 중간값이 정상적으로 연산되는지 확인하기 위한 중간 처리 과정을 보여준다.

3. 제한 사항

- 과제의 함수명과 변수명은 반드시 뼈대코드와 동일하게 구현한다.

```
uint modMul(uint x, uint y, uint mod);
uint modPow(uint base, uint exp, uint mod);
bool isPrime(uint n, uint repeat);
uint gcd(uint a, uint b);
uint modInv(uint a, uint m);
void miniRSAKeygen(uint *p, uint *q, uint *e, uint *d, uint *n);
uint miniRSA(uint data, uint key, uint n);
```

- 파라미터들의 기본 자료형으로써 4byte 'unsigned int'를 'uint'로 정의하여 사용하도록 하며, 이 보다 더 큰(4byte 이상의) 자료형은 사용하지 않도록 한다.
- 주어진 뼈대코드에서 임의 수 생성을 위한 RNG(Random Number Generator)는 rsa.h 파일에 구현된 $0 \leq r < 1$ 사이의 값을 duoble 형으로 반환해주는 'WELLRNG512a' 함수를 사용하도록 한다.
- 전체 알고리즘에서 사용되는 나눗셈, 나머지(모듈러) 연산을 c언어에서 지원하는 연산자('/', '%')를 사용하지 않고 비트 연산으로 처리 하도록 한다.
- 거듭제곱 연산을 할 때 'square and multiply' 알고리즘을 이용하여 빠르게 연산되도록 한다.
- 모듈러 값 n이 $2^{31} \leq n < 2^{32}$(32bit 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 $\bmod \Phi(n)$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.

4. 참고 사항

- 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.
- 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.

5. 제출물

"[학번_이름]_miniRSA.zip" - 과제의 소스 파일(miniRSA.h, miniRSA.c 또는 추가 구현 소스 파일)을 앞의 이름과 같이 압축해 마감일을 엄수하여 joontaechoi@gmail.com으로 제출한다.

6. 마감기한

가. 마감일: 2017년 12월 02일 토요일 자정 전까지

나. 마감일을 넘겨 제출할 경우, 하루 단위로 총점의 20%씩 감점된다.

7. 참고자료

가. R. L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Comm. ACM 21, 2(Feb. 1978), 120-126.

8. 문의사항

가. 제 3 공학관 519-2호 정보보호연구실 최준태 조교

나. joontaechoi@gmail.com, 메일 보낼 시에 학번과 성명 표기해주세요.

- 결과 예제

```
0. Key generation is Success!
 p : 54623
 q : 62189
 e : 3152570619
 d : 3028275219
 N : 3396949747

 input data : 2018915346
 output data : 2543547194
1. plain text : 2018915346
2. encrypted plain text : 2543547194

 input data : 2543547194
 output data : 2018915346
3. cipher text : 2543547194
4. Decrypted plain text : 2018915346

RSA Decryption: SUCCESS!
```