암호프리미티브구현

GMP의 연산 함수 및 실습

장남수

1. GMP의 정수 연산함수

GMP의 정수 연산함수

- 정수 덧셈(Addition) : rop = op1+op2
  - void mpz_add (mpz t rop, mpz t op1, mpz t op2)
  - void mpz_add_ui (mpz t rop, mpz t op1, unsigned long int op2)
- 정수 뺄셈(Subtraction) : rop = op1-op2
  - void mpz_sub (mpz t rop, mpz t op1, mpz t op2)
  - void mpz_sub_ui (mpz t rop, mpz t op1, unsigned long int op2)
  - void mpz_ui_sub (mpz t rop, unsigned long int op1, mpz t op2)
- 정수 곱셈(multiplication) : rop = op1*op2
  - void mpz_mul (mpz t rop, mpz t op1, mpz t op2)
  - void mpz_mul_si (mpz t rop, mpz t op1, long int op2)
  - void mpz_mul_ui (mpz t rop, mpz t op1, unsigned long int op2)

Gmp(지엠피)의 연산 함수에 대하여 살펴보도록 하겠습니다. GMP에서 제공하는 연산 함수 중에서 암호 알고리즘에서 필요로 하는 함수는 정수연산과 난수생성 함수이므로 이에 대하여 중점적으로 살펴보겠습니다. 먼저 덧셈의 경우 op1(오피원)과 op2(오피투)를 더하여 rop(알오피에) 저장하는 함수로 mpz_add(엠피지 에드) 함수를 사용합니다. 마찬가지로 뺄셈함수는 op1(오피원)에서 op2(오피투)를 빼서 결과를 rop(알오피에) 저장하는 함수로 mpz_sub(엠피지 서브) 함수를 사용합니다. 마지막으로 곱셈함수는 op1(오피원)과 op2(오피투)를 곱하여 rop(알오피에) 저장하는 함수로 mpz_mul(엠피지 멀) 함수를 사용합니다.

GMP의 정수 연산함수

- 관련 기타 함수
  - rop = rop+(op1*op2)
    - void mpz_addmul (mpz t rop, mpz t op1, mpz t op2)
    - void mpz_addmul_ui (mpz t rop, mpz t op1, unsigned long int op2)
  - rop = rop-(op1*op2)
    - void mpz_submul (mpz t rop, mpz t op1, mpz t op2) [Function]
    - void mpz_submul_ui (mpz t rop, mpz t op1, unsigned long int op2)

- 정수 나눗셈(Division) : n = q*d + r, d>r>0
  - void mpz_tdiv_q (mpz t q, mpz t n, mpz t d) [Function]
  - void mpz_tdiv_r (mpz t r, mpz t n, mpz t d) [Function]
  - void mpz_tdiv_qr (mpz t q, mpz t r, mpz t n, mpz t d)
  - void mpz_mod (mpz t r, mpz t n, mpz t d)

다음으로는 덧셈과 뺄셈 연산과 곱셈 연산이 결합된 함수와 나눗셈 함수에 대하여 살펴보겠습니다. 먼저 mpz_addmul(엠피지 에드멀) 함수는 op1(오피원)과 op2(오피투)를 곱하고 곱한 결과를 rop(알오피)와 더하여 다시 rop에 반환하는 함수입니다. 다음으로 mpz_submul(엠피지 서브멀) 함수는 op1(오피원)과 op2(오피투)를 곱하고 곱한 결과를 rop(알오피)에서 뺀 후 다시 rop에 반환하는 함수입니다. 이와 같이 연산이 결합된 함수는 알고리즘 내에서 추가적인 변수 선언 없이 보다 편리하게 연산이 가능하도록 하는 기능을 제공합니다. 다음으로 나눗셈 함수를 살펴보겠습니다. 나눗셈의 경우 n(엔)을 d(디)로 나눈 몫 q(큐)와 나머지 r(알)로 표현되며 이때 나머지는 0보다 크고 d 보다 작은 값을 가집니다. 나눗셈 함수는 몫 또는 나머지만을 반환하거나 이두 값을 모두 반환하는 함수가 있으며, 암호알고리즘에서는 나머지 만을 반환하는 mpz_mod(엠피지 모드) 함수를 주로 상용하게 됩니다.

GMP의 정수 연산함수

- 지수승 함수(Exponentiation) : rop = $base^{exp}$ (mod mod)
  - void mpz_powm (mpz t rop, mpz t base, mpz t exp, mpz t mod)

- 기타 함수
  - int mpz_probab_prime_p (mpz t n, int reps)
  - void mpz_nextprime (mpz t rop, mpz t op)
  - void mpz_gcd (mpz t rop, mpz t op1, mpz t op2)
  - void mpz_gcdext (mpz t g, mpz t s, mpz t t, mpz t a, mpz t b)
  - int mpz_invert (mpz t rop, mpz t op1, mpz t op2)

다음으로는 암호 알고리즘에서 직접적으로 사용하게 되는 지수승과 기타 함수들에 대하여 살펴보겠습니다. RSA 암복호화에 사용되는 지수승 함수인 mpz_powm(엠피지 파워엠) 함수는 base(베이스)에 exp(이엑스피) 승한 결과는 mod(엠오디) 값으로 모듈러 연산한 결과를 rop(알오피)에 반환합니다. 다음으로 mpz_probab_prime_p(엠피지 프로바블 프라임 피) 함수는 n이 소수인지 아닌지 밀러라빈 알고리즘으로 reps(알이피에스)번 테스트 한 후 소수이면 1을 합성수이면 0을 반환하는 함수로 RSA 키쌍을 생성할 때 사용하는 함수입니다. 그리고 추가적으로 최대공약수와 역원을 구하는 함수 등이 있습니다.

학습활동

학습활동

✓ 큰 정수 연산의 예제

1. 2048비트 데이터 n과 1024비트 데이터 d를 랜덤하게 생성한다.
2. n을 d로 나누어 몫 q와 나머지 r을 구한다.
3. 구해진 몫 q에 d를 곱하고 r을 더한 결과를 cmp_n에 저장한다.
4. 계산에 사용된 모든 데이터를 출력하고 n과 cmp_n을 비교한 결과를 출력한다.

test.c

```
void main()
{
        /* mpz 자료형 초기화*/
        /* 난수 생성을 위한 시드 생성*/
        /* 난수 생성*/
        /* n = q*d + r */
        /* 값 비교를 위한 계산*/
        /* 데이터 출력 및 비교*/
        /* 자료형 해제*/
}
```