Criptografía

David Arroyo Guardeño

Escuela Politécnica Superior de la Universidad Autónoma de Madrid

September 29, 2016

GMP

Installation

sudo apt-get install libgmp-dev

Use

```
#include <gmp.h>
```

```
#include <stdio.h>
#include <time.h>
#include <gmp.h>
#include <stdlib.h>

#define LENGTH 128
#define REP 10
#define TAM 10

int main(int argc, char *argv[])
{
  long seed;  int count;  gmp_randstate_t state;  mpz_t n, temp;
  gmp_randinit(state, 0, LENGTH); mpz_init(n);  mpz_init(temp);

  if (argc != 2){
      printf ("Integer, please...\n");
      exit(1);
  }else{
    mpz_set_str(n,argv[1],TAM);
  }
  time (&seed);  gmp_randseed_ui (state, seed);
  for(count=REP; count; count--){
    mpz_urandomm (temp, state, n);
    mpz_out_str (stdout, TAM, temp);
     printf ("\n");
  }

  gmp_randclear (state); mpz_clear(n); mpz_clear(temp);
  return 0;
}
```

```
gcc -oexample01 example01.c -lgmp
```

GMP's function name usually looks like:

mpz_funxx_dataxx

For convenience the GMP functions are divided in several categories:

Integer Functions Functions for signed integer arithmetic, begin with mpz (about 150 functions). The associated type is mpz_t

Rational Functions Functions for rational number arithmetic, begin with mpq (about 40 functions). The associated type is mpq_t

Floating-point Functions Functions for oating-point arithmetic, begin with mpf (about 60 functions). The associated type is mpf_t

BSD Compatible Functions Functions compatible with Berkeley MP.The associated type is MINT

Low-level Functions Fast low-level functions for natural numbers arithmetic, begin with mpn (about 30 functions). The associated type is mp_limb_t

Miscellaneous Functions

```
mpz_t t; // this is the type of statement is needed to declare a variable
mpz_init (t); // initialization of a variable
mpz_set_ui (t, 2); // assignment of 2 to already initialized variable t
mpz_set_str (t, "1234"); // string assignment
mpz_add (t, a , b); // a + b is assigned to t, that is, t = a + b
mpz_sub (t, a, b); // subtraction
mpz_mul (t, a, b); // multiplication
gmp_printf ("% Zd ", t); // print output t
```

Bibliography