CRYPTOGRAPHY

MESIGI3906
WORKING WITH GMP AND OPENSSL

JOHN CAGNOL / AMAURY DARSCH
DER CALCUL SCIENTIFIQUE / DER GÉNIE INFORMATIQUE

AGENDA

- First contact
  - Using OpenSSL
  - Basic difference with GPG
  - Playing with encryption command
  - Random number generation
- GMP
  - Multi-precision arithmetic before John's wrap-up
  - Simple conversion and predicate

OPENSSL / GPG

- OpenSSL
  - SSL : Secure Socket Layer
  - TLS : Transport Layer Security
- GPG
  - PGP : Pretty Good Privacy
  - GPG : Gnu Privacy Guard
- Two complementary tools
  - Secure data transport
  - Messages encryption and key distribution

OPENSSL COMMANDS

- Openssl supports a wide range of commands
- Syntax
  - openssl command [options] [arguments]
- Command list
  - openssl list-standard-commands
  - openssl list-message-digest-commands
  - openssl list-cipher-commands
- Most commands have a cryptic syntax
  - Cipher specific options
  - Crypto mode and standard reference

OPENSSL CIPHERS

- Numerous ciphers
  - Syntax: cipher-key size-mode
  - Example: aes-192-cbc
- Common ciphers
  - AES: advance encryption standard (reference)
  - DES: data encryption standard (obsolete)
  - DES3: triple DES
  - CAST: (GPG)
  - RC5: (SSL/TLS)

OPENSSL ENCRYPTION

- Command: enc
  - Encryption: -e
  - Decryption: -d
  - Salting: -salt -nosalt
- Example
  - openssl enc -e -nosalt -des -in msg.txt -out msg.cpy
  - openssl enc -d -nosalt -des -in msg.cpy -out msg.dcy
- The crypted file size depends on the cipher and the salt option

COMPATIBILITY WITH GPG

- Openssl is not really compatible with GPG
  - Two different goals
  - GPG header with encrypted file
- A file encrypted with GPG cannnot be decrypted with Openssl
  - GPG uses CAST5 by default, no mode
  - GPG inserts a header to help in the decryption
- Example
  - gpg -c –cipher-algo 3DES msg.txt (encryption)
  - gpg msg.txt.gpg (decryption)

RANDOM DATA

- Command: rand
  - Base 64: -base64
- Example
  - openssl rand -base64 -out rand.txt 128
  - openssl enc -d -base64 -in rand.txt -out rand.bin
- Random data generation is an art
  - PRNG: pseudo random number generator
  - Kernel: entropy based generator
  - Need to be properly seeded

OPENSSL: PRIME NUMBER

- Large prime numbers are essential with certain cryptographic operations
- OPENSSL
  - Generation : impossible
  - Verification : possible [decimal]
    - openssl prime [num]
- GPG
  - Generation : possible [hexadecimal]
    - gpg –gen-prime 1 [bits]
  - Verification : impossible

GMP

- GNU multi-precision library
  - Fast large integer arithmetic
- Standard operations
  - Usual operations, conversions
  - Modular arithmetic
  - Primality testing and generation
- Available on most platforms in native mode
  - gcc [file to compile] -lgmp
  - Can crash if badly compiled or installed

TRANSFORMATION GMP

```
/*
 * ios: simple i/o demo with gmp
 *      ESILV S9 2008-2009
 *      John Cagnol - Amaury Darsch
 */

#include <stdlib.h>
#include <stdio.h>
#include "gmp.h"

/*
 * usage: print a usage message
 */

static void usage () {
  fprintf (stderr,
           "usage: ios number\n");
  exit (1);
}
```

```
/*
 * main program
 */

int main (int argc, const char** argv) {
  // declarations
  mpz_t n;

  // check number of arguments
  if (argc != 2) usage ();

  // initialize and load
  mpz_init (n);
  if (mpz_set_str (n, argv[1], 0) != 0)
    usage ();
  // print value
  mpz_out_str (stdout, 10, n);
  fprintf (stdout, "\n");

  // everything is fine
  return 0;
}
```

```
gcc -Wall -o gmp-ios gmp-ios.c -lgmp
```

PREDICATE WITH GMP

- GMP can be used to test if a number is prime :
  - mpz_probab_prime
- Exercise
  - Write a program to test if a number is prime
  - Find the mean distribution of prime numbers between two bounds
  - Generate a random number and check for its primality. How many trials are needed before finding a prime number.
  - How long does it take to find a probable prime?