



Soutenance de TX

Printemps 2015

Développement d'une librairie de fonctions de chiffrement en C

Pergoud Florent

Labate Aurélien

- 1 Introduction
- 2 Environnement de programmation
- 3 Les algorithmes principaux
- 4 Conclusion

- Acquérir des connaissances en cryptographie
- Création d'une librairie en C avec des fonctions de cryptographie
- Faire un programme d'exemple avec un menu pour présenter les algorithmes

- 1 Introduction
- 2 Environnement de programmation
- 3 Les algorithmes principaux
- 4 Conclusion

- Projet originalement proposé en C avec GMP
- Mais la syntaxe est très lourde en utilisant la librairie GMP en C
- Par exemple pour $42 + 3172371623812 \times 0x1A$:

```
mpz_t a, b, c, res;  
mpz_init_set_ui(a, 42);  
mpz_init_set_str(b, "3172371623812", 10);  
mpz_init_set_str(c, "1A", 16);  
  
mpz_mul(res, b, c);  
mpz_add(res, a, res);  
  
printf("%s\n", mpz_get_str (NULL, 10, res));
```

- Alors qu'avec la version C++ de GMP :

```
mpz_class a, b, c, res;  
a = 42;  
b = "3172371623812";  
c = "0x1A";  
  
res = a + b * c;  
  
std::cout << "Résultat : " << res << std::endl;
```

- Cependant, le C++ est moins connu est peut gêner

- La version C++ de GMP manque certaines fonctionnalités
- Ainsi pour calculer 2^3 :

```
mpz_class a, res;  
a = 2;  
  
res = mpz_pow_ui(a.get_mpz_t(), 3);  
  
std::cout << "Résultat : " << res << std::endl;
```

- On doit utiliser les fonction de la version C

- Nous avons donc créer une librairie qui évite cela tout en gardant les avantages :

```
mpz2_class a, res;  
a = 2;  
  
res = a.pow(3);  
  
std::cout << "Résultat : " << res << std::endl;
```


- 1 Introduction
- 2 Environnement de programmation
- 3 Les algorithmes principaux**
- 4 Conclusion

Les différents algorithmes

- Exponentiation rapide
- Test de primalité de Rabin-Miller
- Diffie-Hellmann
- ElGamal
- RSA

Test de primalité de Rabin-Miller

- Algorithme probabiliste
- En entrée un entier n à tester et un entier k (précision)
- En sortie "probablement premier" ou "composé"
- Probabilité de se tromper : 4^{-k}

ElGamal

- Algorithme servant à chiffrer et déchiffrer un message
- Clé publique d'Alice : (p, g, A)
- Clés privées a et b avec $A = g^a$ et $B = g^b$
- Message chiffré de Bob : (M, B)
- Déchiffrement $m = M/B^a$

RSA

- Algorithme servant à chiffrer et déchiffrer un message
- Génération d'un couple de clés : clé publique (n,e) , clé privée (n,d)
- Généralement on prend un e constant
- $n = p \times q$ où p et q sont premiers
- $d = e^{-1} \mod \phi(n)$ où $\phi(n) = (p-1)(q-1)$
- chiffrement : $C = M^e \mod (n)$
- déchiffrement : $M = C^d \mod (n)$

- 1 Introduction
- 2 Environnement de programmation
- 3 Les algorithmes principaux
- 4 Conclusion**

- Beaucoup de connaissances acquises ou approfondies :
 - Notions et principes de cryptographie
 - C++
 - GMP
 - \LaTeX
 - Gestion de projet de recherche
- La bibliothèque a été créée et est fonctionnelle
- Beaucoup d'exemples pour utiliser la bibliothèque et comprendre les algorithmes de cryptographie en jeu