Sujets de projet pour le cours "Algorithmique et programmation C"

Michaël Quisquater

7 novembre 2019

1 Sujets en Cryptographie

1.1 Implémentation d'un schéma à seuil basé sur le schéma de signature RSA

La signature numérique est une primitive cryptographique permettant, par exemple, de montrer son accord quant au contenu d'un message. Cette primitive est omniprésente dans le monde moderne ; elle permet notamment de valider les transactions bancaires. Un exemple d'une telle primitive est basé sur la primitive RSA. Classiquement un algorithme de signature fait intervenir une personne (unique) qui signe et une personne (quelconque) qui vérifie la signature.

Une schéma de partage de secret est une primitive cryptographique permettant de partager un secret parmi les membres d'un groupe. Chaque membre du groupe possède une information secrète appelé partage. La réunion de n'importe quel sous-ensemble de partages de taille supérieure à un seuil fixé permet la reconstitution du secret. Un des schémas les plus utilisés est basé sur l'interpolation de Lagrange et a été introduit par Shamir au début des années 80.

Un schéma de signature à seuil est un schéma qui combine les deux primitives précédentes. Il s'agit d'une signature permettant à n'importe quel sous-groupe de taille supérieure à un seuil fixé de générer une signature (classique) pouvant être vérifiée par une personne quelconque. Une solution suivante a été proposée par V. Shoup dans le cas du RSA.

Le projet se divise en différentes parties :

- Ecrire une synthèse succinte introduisant la notion de signature numérique (définition globale et description du RSA), les schémas de partage de secret (définition globale et description du schḿa de Shamir) et la signature distribuée à seuil (definition globale). Un rappel des mathématiques utilisées sera utile.
- Détailler le schéma de signature distribué proposé par Shoup (première partie de l'article).
- Implémenter en C sur ordinateur ce schéma (partage de la clé algorithme de signature de chacun, algorithme de signature de regrgoupement, algorithme de verification, etc) en utilisant la librairie des grands nombres (GMP). Une description du programme sera intégré au rapport également.
- Une seconde partie du projet consistera à implémenter en C une version partagée de l'algorithme de chiffrement El Gamal.

Le document sera écrit au moyen du traitement de texte Latex.

Références :

— Handbook of Applied Cryptography : `http://www.cacr.math.uwaterloo.ca/hac/`
— librairie GMP : `gmplib.org`
— L'article de V. Shoup : "Practical Threshold Signatures" (section 1 à 4), Advances in Cryptology - EUROCRYPT 2000. Proceedings. Lecture Notes in Computer Science, Vol. 1807, Springer-Verlag, 2000.

1.2 Implémentation d'une attaque sur un standard de signature simplifié du RSA

La signature numérique (digital signature en anglais) est une primitive cryptographique permettant entre autres de montrer son accord quant au contenu d'un message. Cette primitive est omniprésente dans le monde moderne ; elle permet notamment de valider une transaction bancaire. Un exemple d'une telle primitive est basé sur la primitive RSA (introduite en 1977 par R. Rivest, A. Shamir et L. Adleman). Pratiquement, le RSA en fonction signature est utilisé avec la méthode d'encodage PSS (probabilistic Signature Scheme) introduite par Mihir Bellare et Philip Rogaway en 1994. Le standard ISO/IEC-9796-1 a été introduit en 1991 et a également été très utilisé. Ce dernier a spécialement été conçu pour éviter les attaques dites multiplicatives.

Il y a quelques années, Coron, Naccache et Stern ont introduit une attaque permettant de mettre en défaut une version simplifiée de ce standard (ainsi que d'autres egalement). Celle-ci se base sur la propriété multiplicative du RSA et sur la notion de nombres lisses. D'autres travaux ont suivi et ont abouti à la mise en défaut de standards effectifs.

Nous proposons d'implémenter cette méthode en utilisant la librairie GMP notamment.

Le projet se divise en différentes parties :

- Ecrire une synthèse succinte introduisant la notion de signature numérique (définition générale et description du RSA). Décrire le standard (probablement ISO/IEC-9796-1) qui sera attaqué ainsi que sa version simplifiée. Un rappel des mathématiques utilisées sera utile.
- Détailler le principe de l'attaque de Coron, Naccache et Stern (voir article ci-dessous).
- Implémenter sur ordinateur cette attaque en utilisant la librairie GMP (sous C). On considérera une version simplifiée du standard pour laquelle l'attaque se monte en temps raisonnable. Une description du programme sera intégrée au rapport également.
- On pourra également appliquer la méthode à différents standards si le temps le permet. Cette partie est facultative et ne devra pas être considérée dans un premier temps.

Le document sera écrit au moyen du traitement de texte latex.

Références utiles :

- Handbook of Applied Cryptography Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone.
  Disponible sur http ://www.cacr.math.uwaterloo.ca/hac/
- L'article de Coron, Naccache et Stern : "On the security of RSA Padding", Crypto'99, Ed. M. Wiener, LNCS 1666, pp.1–18, 1999.
- Librairie GMP. Disponible sur http ://www.swox.com/gmp/

1.3 Implémentation de PRESENT en bit-slice

PRESENT est un algorithme de chiffrement par bloc spécialement conçu pour les environnements contraints.

Les attaques par recherche exhaustive ou par dictionnaire sont très courantes en cryptographie. Il s'agit de chiffrer ou déchiffrer un même message avec de multiples clés. Notons que cette attaque peut-être menée en parallèle, il suffit de diviser l'espace des clés en plusieurs parties et d'affecter chacune à un processeur ou ordinateur.

Une autre idée, dite du *bitslice* et qui peut être combinée à la précédente, consiste à écrire un algorithme qui calcule le chiffrement (ou le déchiffrement) d'un même pour plusieurs clé à la fois. Ce regroupement d'opérations permet souvent d'être plus efficace. Le principe est assez simple. Imaginons que l'on souhaite évaluer la fonction $f : (\mathbb{F}_2 \times \mathbb{F}_2) \to (\mathbb{F}_2 \times \mathbb{F}_2)$ pour 32 couples (x_i, y_i). Une première solution consiste à évaluer la fonction de façon itérative sur les 32 couples. Une alternative consiste à tirer parti de la représentation algébrique d'un telle fonction. Supposons que $f : (\mathbb{F}_2 \times \mathbb{F}_2) \to (\mathbb{F}_2 \times \mathbb{F}_2)$ puisse être représentée par l'expression algébrique $(x, y) \mapsto (x \oplus y, x \cdot y)$. Considérons les vecteurs $\underline{x} = (x_{31}, \cdots, x_0)$ et $\underline{y} = (y_{31}, \cdots, y_0)$. En une seule instruction il est possible de calculer le premier (resp. le second) bit de sortie de la fonction pour les 32 données, i.e. $\underline{x} \oplus \underline{y}$ (resp. $\underline{x} \& \underline{y}$). Ce principe peut évidemment être étendu à un algorithme de chiffremment/ déchiffrement complet.

Cette technique a été utilisée pour le DES dans le cadre du projet DESCHALL visant à casser une clé du DES par recherche exhaustive en utilisant de nombreaux ordinateurs du web. L'idée originale à été proposée par E. Biham. La technique a été affinée par Matthew Kwam concernant le bitslicing des Sboxes en utilisant les tables de Karnaugh.

Le projet se découpe en plusieurs parties :

- Implémenter ou obtenir une implémentation classique en C de l'algorithme afin d'avoir une base de comparaison.
- Détailler dans un rapport écrit en Latex la technique du bitslicing pour les algorithmes de chiffrement par blocs et expliquer la méthode d'optimisation des circuits combinatoires via les tables de Karnaugh.
- Implémenter en C l'algorithme de chiffrement en utilisant la technique de bitslicing (utiliser les tables de Karnaugh pour optimiser les représentations ou une autre méthode efficace). Expliquer ce code dans le rapport.
- Un façon classique de réaliser une implémentation "bitslice" consiste à écrire un code qui va générer le code "bitslice" car celui-ci est extrêment répétitif (avec des petites modifications) et très long. Cette façon de procéder est à considérer dans ce projet.
- Comparer les résultats dans le rapport.

Une piste d'extension peut consister à optimiser le code au maximum en utilisant au mieux les ressources et les représentations.

Références :

- `http://www.cs.cmu.edu/~dkindred/des/bitslice.html`
- `http://www.darkside.com.au/bitslice/`
- `http://plaintext.crypto.lo.gy/article/378/untwisted-bit-sliced-tea-time`
- `http://fr.wikipedia.org/wiki/Single_instruction_multiple_data`
- Eli Biham. A Fast New DES Implementation in Software. Booktitle, Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel.

1.4 Implémentation du schéma de Paillier et de son application au vote électronique.

En 1999, un jeune chercheur français, Pascal Paillier, a inventé un nouveau mécanisme de fonction à trappes basé sur un nouveau porblème difficile. Ce mécanisme a permis de construire un nouvel algorithme de chiffrement et de signature. Outre une très bonne efficacité, ce mécanise possède des propriétés intéressantes comme le morphisme.

Dans ce projet nous nous interesserons à une étude théorique et pratique de ce schéma ; la génération des clés, la réduction de sécurité par rapport au problème de la factorisation etc... Dans un second temps, on s'intéressera à une généralisation de ce schéma et de l'utilisation de cette généralisation au vote électronique.

Le projet consiste à écrire dans un premier temps un document (Latex) unificateur et pédagogique sur le sujet. On expliquera de façon détaillée les hypothèses et la mise en oeuvre du vote électronique en mettant en évidence certaines failles d'applications (potentielles) s'il elles existent.

Dans un second temps, on s'interessera à l'implémentation en GMP (librairie des grands nombres) de l'algorithme de Paillier et de la généralisation de celui-ci appliqué au vote électronique..

Références utiles :

- Handbook of Applied Cryptography Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone. Disponible sur http ://www.cacr.math.uwaterloo.ca/hac/
- A Generalization of Paillier's Public-Key System with Applications to Electronic Voting. I. Damgard, M. Jurik and Jesper Buus Nielsen.
- Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. P. Paillier.
- Librairie GMP. Disponible sur http ://www.swox.com/gmp/

1.5 Attaque contre le système de McEliece par information set decoding (sujet initialement proposé par Luca De Feo)

Le but de ce projet est d'implanter l ?attaque de Stern pour le système de chiffrement McEliece.

Objectifs :

- Implanter l'algorithme de Information Set Decoding.
- Implanter le systéme de McEliece.
- Implanter la cryptanalyse.

Références :

- J. Stern. *A method for finding codewords of small weight.
- A. Canteaut, F. Chabaud. A New Algorithm for Finding Minimum-Weight Words in a Linear Code.
- A. Canteaut, N. Sendrier. Cryptanalysis of the Original McEliece Cryptosystem.
- D. J. Bernstein, T. Lange, C. Peters. Attacking and defending the McEliece cryptosystem.

2 Sujets en calcul formel

2.1 "Number-Theoretic Transform" et multiplication de grands entiers

L'algorithme de Schönhage-Strassen est un algorithme de multiplication de grands entiers par transformée de Fourier rapide publié en 1971 par A. Schönhage et V. Strassen. La transformée de Fourier rapide utilise des racines de l'unité appartenant à un corps fini. Dans ce contexte , elle porte le nom de "Number-Theoretic Transform" (NTT). Il existe de nombreuses versions de la transformée de Fourier rapide. On veillera a en implémenter une qui laisse les données en place ("in place transform").

Le projet consistera à :

- Détailler dans un rapport écrit en Latex, la NTT et son implémentation "in place" ainsi que l'algorithme de Schönhage-Strassen.
- Implémenter en C la NTT "in place" (particularisé à son usage pour l'algorithme de Schönhage-Strassen) et l'algorithme de Schönhage-Strassen en utilisant la librairie GMP. Le code devra être commenté et expliqué dans le rapport.
- Un extension du projet consistera à s'intéresser à d'autres l'optimisations.

Références :

- FFT en général
  - `https://math.berkeley.edu/~strain/273.F10/duhamel.vetterli.fft.review.pdf`
  - `http://en.wikipedia.org/wiki/Discrete_Fourier_transform_%28general%29`
- NTT
  - Fast Convolution Using Fermat Number Transforms with Applications to Digital Filtering. R. Agarwal et C. Burrus.
  - `ticsp.cs.tut.fi/images/9/94/Cr1043-wien.pdf`
- Algorithme de Schönhage-Strassen
  - `http://en.wikipedia.org/wiki/Sch%C3%B6nhage%E2%80%93Strassen_algorithm`

2.2 Test de primalité polynomial : AKS

Le test de primalité AKS (Agrawal-Kayak-Saxena) est un algorithme de preuve de primalité déterministe et polynomial découvert en 2002. Cet algorithme a mis fin à une question séculaire à savoir si PRIMES est dans $\mathcal{P}$.

Le projet consistera à :

- Détailler dans un rapport écrit en Latex, la version originale d'AKS et ses variantes (Lenstra, Bernstein). Il s'agira de mettre en évidence les similitudes et les différences entre les méthodes. Donner une synthèse sur les mathématiques nécessaires à compréhension du sujet (anneau cyclotomique essentiellement).
- Implémenter en C la version orginale de l'algorithme ainsi que les deux variantes (Lenstra, Bernstein) en utilisant la librairie GMP. Le code devra être commenté et expliqué dans le rapport.
- Un extension du projet consistera à s'intéresser à s'intéresser à d'autres l'optimisations.

Références :

- PRIMES is in P. Manindra Agrawal, Nerraj Kayal and Nitin Saxena.
- Proving Primality after Agrawal-Kayal-Saxena.

2.3 Algorithme de factorisation par la méthode des fractions continues

La sécurité du RSA est intimement liée à la difficulté de factoriser des grands nombres. De nombreuses méthodes existent et ce projet propose de s'intéresser à l'une d'entre elles, appelée méthode de factorisation basée sur les fractions continues. Le projet comportera plusieurs volets :

- Détailler dans un rapport écrit en Latex, les différentes méthodes tournant autour de la méthode de Dixon (Dixon avec ou sans fraction continue). Il s'agira de mettre en évidence les similitudes et les différences entre les méthodes. Donner une synthèse sur les fractions continues.
- Implémenter en C la version de l'algorithme qui semble la plus efficace en utilisant la librairie GMP. Le code devra être commenté et expliqué dans le rapport.
- Un extension du projet consistera à s'intéresser à l'optimisation de la partie "algèbre linéaire".

Références :

- Introduction to cryptography. J.A. Buchman.
- A course in number theory and cryptography. N. Koblitz.
- gmplib.org
- "On factoring large numbers". Lehmer and Powers.
- "A method of factoring and the factorization of F_7". M. Morrison and John Brillhart.
- "Fast Factorisation of integers". J. D. Dixon.
- Implementation of the continued fraction integer factoring algorithm. C. Pomerance and S. Wagstaff.

2.4 Calcul de petites racines d'un polynôme de bas degré à coefficients dans $\mathbb{Z}/n\mathbb{Z}$

Lorsque le RSA est utilisé en mode "broadcast" avec un exposant 3, il est connu que le théorème des restes chinois permet de retrouver le message envoyé à partir des chiffrés envoyé à 3 personnes. Une idée pour contrer cette attaque a été d'appliquer une fonction affine (différentes pour chaque envoi) au message. Comme montré par Hastad en 1985, l'attaque d'un tel protocole revient à calculer une solution d'un système de polynômes de bas degrés modulo des nombres composés relativement premiers. Hastad proposa d'utiliser l'algorithme LLL pour résoudre un tel problème. Coppersmith affina la solution par la suite.

Le but de ce projet est d'implémenter l'algorithme de Coppersmith permettant de le calcul de petite solution de polynômes de bas degré à coefficients dans $\mathbb{Z}/n\mathbb{Z}$. On utilisera pour cela la librairie Flint (série 1 qui inclut l'implémentation de LLL).

Le projet se découpe en 3 parties :

1. Expliquer la théorie de l'algorithme LLL et l'algorithme de Coppersmith pour la recherche de petite racine d'un polynôme univarié de bas degré à coefficients dans $\mathbb{Z}/n\mathbb{Z}$.
2. Expliquer l'installation et l'utilisation globale de la librairie C Flint (série 1).
3. Implémenter en C l'algorithme de Coppersmith.

Une piste d'extension est d'utiliser l'algorithme implémenté pour attaque le RSA dans certaines circonstances.

Références :

— J. Hastad. On Using RSA with Low Exponent in a Public Ket Network. LNCS. Crypto'85.
— D. Coppersmith. Small Solutions to Polynomials Equations, and Low Exponent RSA Vulnerabilities. Journal of Cryptology. 1997.
— D. Coppersmith. Finding Small Solutions to Small Degree Polynomials. CaLC 2001. LNCS.
— Chapitre sur LLL de "Modern Computer Algebra" de Joachim von zur Gathen an Jürgen Gerhard.
— `http://en.wikipedia.org/wiki/Coppersmith's_Attack`
— `www.flintlib.org/`

2.5 Système modulaire de représentation (RNS) et approche de Montgomery

Le système modulaire de représentation (RNS) est une façon particulière de représenter les nombres qui permet, dans certaines conditions, d'accélérer les opérations (addition, multiplication) et de les réaliser en parallèle. Ce système de représentation repose sur le théorème des restes chinois. Il est beaucoup étudié en cryptographie étant donné la taille des nombres considérés.

La représentation de Montgomery permet d'effectuer des multiplications modulo sans "division trial". Cette représentation est couramment utilisée lorsque des exposentiations rapides de grands nombres sont réalisées. Il a été montré que cette représentation est particulièrement compatible avec la représentation RNS.

Le projet consistera à :

- Détailler dans un rapport écrit en Latex, le principe de la repésentation RNS (ainsi que la façon de réaliser les opérations courantes) ainsi que la l'approche de Montgomery. La combinaison des deux sera également détaillée.
- Implémenter en C ces différentes représentations et opérations en utilisant la librairie GMP ainsi que la combinaison des deux méthodes. Le code devra être commenté et expliqué dans le rapport.
- On comparera l'efficacité de ces opéraions avec les opérations déjà implémentées dans GMP pour différentes tailles de nombres. On expérimentera différentes bases pour RNS.
- Une extension du projet consistera à s'intéresser à l'application de cette approche aux courbes elliptiques.

Références :

- `http://en.wikipedia.org/wiki/Residue_number_system`
- `http://hal-lirmm.ccsd.cnrs.fr/lirmm-00106470/en/`
- `arith.cs.ucla.edu/publications/Residue-SPIE06.pdf`

2.6 Construction de bases normales sur les corps finis

Les bases normales permettent de calculer efficacement dans les corps finis, notamment car l'automorphisme de Frobenius appliqué à un élément correspond à une rotation des ces coefficients.

Le projet consistera à :

- Etudier les contructions de bases normales (méthodes probabilistes et déterministes) et produire un document pédagogique en Latex justifiant théoriquement les les algorithmes. Une distinction entre la caractéristique paire et impaire est peut-être pertinente.
- Implémenter en C au moyen de la bibliothèque GMP les algorithmes considérés. Le code devra être commenté et expliqué dans le rapport.
- Un extension du projet consistera à s'intéresser aux bases normales optimales.

Références :

- S. Gao. Normal bases over finite fields.

2.7 Tests de primalité probabilistes

Les tests de primalités sont des algorithmes utilisés pour déterminer si un nombre est premier. Il existe deux types d'algorithmes ; les déterministes et les probabilistes. Alors que les tests déterministes prouvent la primalité de nombres sans commettre d'erreur, les tests probabilistes peuvent prétendre qu'un nombre premier alors qu'il est composé. La probabilité est généralement bornée théoriquement.

Le projet consistera à :

- Détailler dans un rapport écrit en Latex, le principe des tests de primalité :
  - de Fermat,
  - de Miller-Rabin,
  - de Solovay-Strassen.

  On s'intéressera en particulier aux liens éventuels entre ces algorithmes, aux avantages des uns et des autres et aux preuves concernant la probabilité de succès ou d'erreur.
- Implémenter en C ces différents algorithmes et opérations en utilisant la librairie GMP. Le code devra être commenté et expliqué dans le rapport.
- On comparera l'efficacité de ces algorithmes.
- Une extension du projet consistera à s'intéresser au test de primalité de Frobenius et de Baillies-PSW.

Références :

- The Solovay-Strassen Test. Keith Conrad.
- Primalty Testing. Brinch Hansen.
- Proabilistic Algorithm for testing Primality. Michael O. Rabin.

2.8 Calcul d'indice pour le calcul de logarithmes discrets sur les corps finis

La sécurité d'une importante classe d'algorithmes cryptographiques repose sur la difficulté à calculer le logarithme discret sur certains corps finis. Différentes méthodes ont été proposées et parmi celles-ci "le calcul d'indice".

Le projet consistera à :

- Etudier l'algorithme du calcul d'indice sur un corps premier $\mathbb{Z}/p\mathbb{Z}$, produire un document pédagogique en Latex justifiant théoriquement l'algorithme.
- Implémenter l'algorithme en C au moyen de la bibliothèque GMP. Le code devra être commenté et expliqué dans le rapport.
- Un extension du projet consistera à s'intéresser au calcul d'indice sur les extensions de corps. Une autre direction peut être l'optimisation de la résolution du système linéaire.

Références :

- http ://howelljs.people.cofc.edu/math/msthesis.pdf

2.9 Division de polynômes et inversion dans les corps finis (sujet initialement proposé par Luca De Feo)

Le but de ce projet est d'implanter des méthodes asymptotiquement rapides d'inversion de séries et de division de polynômes à coefficients dans $\mathbb{F}_p$. Au dessus de ces routines, il sera possible de construire une bibliothèque de calculs dans les corps finis. L'utilisation de la bibliothèque Flint pour la multiplication de polynômes à coefficients dans $\mathbb{F}_p$ est conseillée.

Objectifs :

- Implanter l'itération de Newton pour l'inversion de séries.
- Implanter l'algorithme de Newton-Raphson pour la division de polynômes.
- Implanter le pgcd de polynômes (demi-pgcd et autres).
- Implanter les fonctions de base pour calculer dans $\mathbb{F}_{p^n}$. Comparer avec les performances d'une bibliothèque de calcul dans les corps finis, comme par exemple Flint.

Prérequis :

- Connaissances de base sur les corps finis.

Références :

- A. Bostan, F. Chyzak, M. Giusti, R. Lebreton, G. Lecerf, B. Salvy, É. Schost. Algorithmes Efficaces en Calcul Formel, chapitre 4 (https ://hal.inria.fr/hal-01431717/document).
- `https://en.wikipedia.org/wiki/Division_algorithm#Newton.E2.80.93Raphson_division`

2.10 Algorithme d'Euclide rapide (sujet initialement proposé par Luca De Feo)

Le but de ce projet est d'implanter différents algorithmes de pgcd étendu au moyen de la bibliothèque GMP.
Objectifs :

- Implanter les différents algorithme présents dans l'article de Stehlé et Zimmermann. En particulier, la variante de Stehlé et Zimmermann.
- Comparer les performances des ces différentes implémentations. Comparer les performances avec le pgcd de GMP.

Référence :

- D. Stehlé, P. Zimmermann. A Binary Recursive Gcd Algorithm.

2.11 Algorithmes rapides d'interpolation/évaluation (sujet initialement proposé par Luca De Feo)

Le but de ce projet est d ?implanter les algorithmes asymptotiquement rapides d'évaluation multipoint, d'interpolation, et leurs généralisations (restes chinois, multiplicités). Utilisation de la bibliothèque Flint pour la multiplication de polynômes.
Objectifs :

- Implanter l'algorithme d'évaluation multi-point.
- Implanter l'algorithme d'interpolation.
- Implanter les généralisations.

Références :

- J. von zur Gathen, J. Gerhard. Modern computer Algebra.

2.12 Factorisation de polynômes (sujet initialement proposé par Luca De Feo)

Le but de ce projet est d'implanter l'algorithme de factorisation de polynômes sur les corps finis de Cantor-Zassenhaus. Il est conseillé d'utiliser la bibliothèque Flint pour les corps finis.
Objectifs :

— Implanter l'algorithme de Cantor-Zassenhaus pour les corps premiers.

— Implanter l'algorithme de Cantor-Zassenhaus pour les corps finis quelconques.

Références :

— D. G. Cantor, H. Zassenhaus. A new algorithm for factoring polynomials over finite fields.

— J. von zur Gathen, J. Gerhard. Modern computer Algebra.

2.13 Décodage de codes de Reed-Solomon par l'algorithme de Gao (sujet initialement proposé par Luca De Feo)

Le but de ce projet est d'implanter l'algorithme de Gao pour le décodage des codes de Reed-Solomon.

Objectifs :

- Implanter l'arithmétique des polynômes à coefficients dans $\mathbb{F}_p$ (addition, multiplication, division euclidéenne, xgcd, interpolation)
- Implanter l'algorithme de Gao pour des Reed-Solomon sur un corps premier
- Revenir sur l'arithmétique de $\mathbb{F}_p[X]$ en implantant la FFT pour certains premiers p spéciaux, comme décrit dans le papier.
- Faire des tests de performances.

Références

- S. Gao. A new algorithm for decoding Reed-Solomon codes
- J. von zur Gathen, J. Gerhard. Modern computer Algebra.
- A. Bostan, F. Chyzak, M. Giusti, R. Lebreton, G. Lecerf, B. Salvy, É. Schost. Algorithmes Efficaces en Calcul Formel.
- S. Fedorenko. A simple algorithm for decoding Reed-Solomon codes and its relation to the Welch-Berlekamp algorithm.

2.14 Modèles de courbes elliptiques (sujet initialement proposé par Luca De Feo)

Le but de ce projet est d'implanter différents modèles de courbes elliptiques et de comparer leurs performances.

Résumé : Les courbes elliptiques sont traditionnellement présentées comme les lieux d'annulation d'une équation courte de Weierstrass projective $Y^2Z = X^3 + aXZ^2 + bZ3$. Cette représentation a une importance historique, et joue un rôle central dans la théorie des courbes elliptiques sur $\mathbb{C}$, ce qui suffit à justifier son utilisation presque universelle.

Cependant, lorsque on a pour objectif de faire une implantation cryptographique la plus rapide possible, d'autres représentations des courbes elliptiques se révèlent plus efficaces. C'est le cas des modèles d'Edwards, de Jacobi, de Huff, ou encore du modèle Hessian.

En plus du choix du modèle, l'implantation de la multiplication scalaire est sensible à d'autres paramètres, comme la représentation du plan projectif, ou encore l'algorithme de double-and-add choisi.

Ce projet a pour but d'implanter différentes représentations de courbes elliptiques sur des corps premiers, et de comparer leurs performances. Il est conseillé d'utiliser la bibliothèque GMP pour la représentation des grands entiers modulaires.

En bonus, on pourra s'intéresser à l'implantation de la courbe Ed25519 à partir des primitives pour les entiers en virgule flottante, sans avoir recours à une bibliothèque de grands entiers.

Objectifs :

- Implanter le modèle de Weierstrass projectif, affine et Jacobian.
- Implanter le modèle d ?Edwards projectif et inversé.
- Implanter le modèle des quartiques de Jacobi projectif.
- Implanter le modèle des intersections de Jacobi projectif.
- Implanter le modèle Hessien projectif.
- Implanter le modèle de Montgomery et la multiplication par échelle.
- Implanter les formules de conversion entre les différents modèles.
- Comparer les performances des implantations.
- Essayer d ?approcher le mieux possible le record d ?implantation de la courbe Ed25519 en utilisant de l ?arithmétique flottante.

Prérequis :

Notions de base sur les courbes elliptiques

Références :

- The Explicit-Formulas Database `http://www.hyperelliptic.org/EFD/index.html`.
- D. J. Bernstein, T. Lange. Analysis and optimization of elliptic-curve single-scalar multiplication.
- D. J. Bernstein, P. Birkner, M. Joye, T. Lange, C. Peters. Twisted Edwards Curves.
- P. L. Montgomery. Speeding the Pollard and Elliptic Curve Methods of Factorization.
- D. J. Bernstein. Curve25519 : new Diffie-Hellman speed records.

3 GLV

L'addition efficace sur les courbes elliptiques peut tirer parti de l'évaluation particulièrement rapide d'endomorphismes.

Le projet consistera :

- à détailler dans un document en Latex la méthode GLV, à expliquer pourquoi elle fonctionne et préciser le gain d'efficacité.
- Implémenter en C ces différents algorithmes et opérations en utilisant la librairie GMP. Le code devra être commenté et expliqué dans le rapport.
- On comparera l'efficacité de ces algorithmes (avec et sans l'accélération). Les résultats seront présentés dans le rapport.
- Une extension du projet consistera à s'intéresser à des extensions de la méthode.

Référence :

- `http://antoanthongtin.vn/Portals/0/UploadImages/kiennt2/KyYeu/DuLieuNuocNgoai/6.Advances%20in%20cryptology-Crypto%202001-LNCS%202136.9/21390190.pdf`