Contribution à la Résolution Algébrique et Applications en Cryptologie

Guénaël Renault

UNIVERSITÉ PIERRE ET MARIE CURIE

HABILITATION À DIRIGER DES RECHERCHES

Spécialité **Informatique**

Contribution à la Résolution Algébrique et Applications en Cryptologie

Présentée et soutenue publiquement par

Guénaël Renault

le **jeudi 8 décembre 2016**

après avis des **rapporteurs**

M. Pierre-Alain FOUQUE	Professeur, IUF et Université Rennes 1
M. Éric SCHOST	Associate Professor, University of Waterloo
M. Émmanuel THOMÉ	Directeur de Recherche, INRIA Lorraine

devant le **jury** composé de

M. Jean-Charles FAUGÈRE	Directeur de Recherche, INRIA Paris
M. Pierre-Alain FOUQUE	Professeur, IUF et Université Rennes 1
M. Emmanuel PROUFF	Expert (HDR), Safran Identity and Security
M. Mohab SAFEY EL DIN	Professeur, IUF et UPMC
M. Éric SCHOST	Associate Professor, University of Waterloo
M. Emmanuel THOMÉ	Directeur de Recherche, INRIA Lorraine

Table des matières

CHAPITRE 1

Introduction et Synthèse

1.1 Introduction

Afin de positionner au mieux les résultats qui seront présentés dans la suite de ce document, il est important de commencer par en définir le titre.

Résolution algébrique. Lagrange, dans [94], est l'un des premiers mathématiciens à donner une définition de cette expression qui nous semble la plus adaptée. En effet, dans cet ouvrage il se propose d'étudier la résolution des équations polynomiales sous forme de radicaux. Plutôt que d'essayer de produire de nouvelles formules pour la résolution d'équation particulières, comme c'était le cas jusqu'à présent, il essaie de faire une synthèse de toutes les méthodes existantes afin d'en extraire une méthode algébrique générale. Même s'il a conscience que son objectif principal ne sera pas atteint, il montre clairement qu'il a compris l'essentiel de la démarche algorithmique permettant de décider la résolubilité par radicaux de ces équations (voir l'extrait ci-après issu de l'ouvrage [94] datant de 1770). Cette démarche algorithmique où l'ensemble des données manipulées restent symboliques et les outils employés sont essentiellement algébriques définit la résolution algébrique. Bien sûr, cette démarche fait aujourd'hui partie du *calcul formel* et peut être appliquée à d'autres problèmes que ceux de la résolution sous forme de radicaux, comme

> 109. Voilà, si je ne me trompe, les vrais principes de la résolution des équations et l'analyse la plus propre à y conduire; tout se réduit, comme on voit, à une espèce de calcul des combinaisons, par lequel on trouve *à priori* les résultats auxquels on doit s'attendre. Il serait à propos d'en faire l'application aux équations du cinquième degré et des degrés supérieurs, dont la résolution est jusqu'à présent inconnue; mais cette application demande un trop grand nombre de recherches et de combinaisons, dont le succès est encore d'ailleurs fort douteux, pour que nous puissions quant à présent nous livrer à ce travail; nous espérons cependant pouvoir y revenir dans un autre temps, et nous nous contenterons ici d'avoir posé les fondements d'une théorie qui nous paraît nouvelle et générale.

par exemple : les problèmes diophantiens ou la résolution des systèmes polynomiaux.

Un autre point important mis en avant par Lagrange est l'utilisation de structures intrinsèques à ces équations. C'est ce qu'il appelle *combinaison* et que nous appelons aujourd'hui action du groupe de Galois. Ainsi, il décrit ici un principe général de résolution où il est nécessaire d'utiliser des structures particulières au problème pour y répondre au mieux.

Depuis, nombre de mathématiciens, d'informaticiens suivent ce principe. L'objectif principal étant de développer des algorithmes efficaces en considérant toutes les particularités du problème donné plutôt que d'appliquer des stratégies trop génériques. Pour le problème étudié par Lagrange, en toute généralité il n'y a pas de solution possible, alors que si l'on considère la structure de groupe associée, on peut, dans les bons cas, résoudre le problème efficacement. Ce qui fait toute la puissance de cette approche en fait aussi sa difficulté, ne serait-ce qu'identifier les structures qui permettent de résoudre plus efficacement est parfois un problème en soi. En fonction du contexte, ces structures peuvent provenir de la définition même du problème, de sa modélisation, des quantités prenant part au calcul ou encore de la nature de la sortie. Les travaux présentés dans la suite de ce document suivent tous cet axe de recherche et ont pour certains des applications en cryptologie.

Applications en Cryptologie. La sécurité des systèmes cryptographiques asymétriques (e.g. ceux permettant l'échange à distance de clés secrètes) reposent généralement sur des problèmes algébriques. En particulier, le niveau de sécurité se mesure en fonction des coûts en temps de calcul des meilleurs algorithmes connus permettant de résoudre ces problèmes. Ainsi, ce domaine de l'informatique fournit une panoplie de problèmes qui sont souvent très structurés et donc représente un creuset pour le développement de nouveaux algorithmes *ad hoc* de résolution algébrique.

Bien que tous les résultats présentés ci-après suivent ce même axe de recherche, ils peuvent se différencier à travers les problématiques traitées. Ainsi ils peuvent être regroupés selon trois familles de problèmes généraux :

— **Problèmes galoisiens** : c'est la représentation formelle des racines d'un polynôme en une variable qui nous intéresse ici, pouvoir les manipuler symboliquement ainsi que les actions galoisiennes qui leur correspondent.
La structure naturelle que nous avons utilisée pour résoudre efficacement ces problèmes est l'action du groupe de Galois sur les racines du polynôme. Nous détaillons les résultats obtenus dans la section 1.2.

— **Problèmes diophantiens** : On s'intéressera plus particulièrement au calcul de petites racines entières d'équations diophantiennes.
Ici c'est la nature même des solutions cherchées qui guide le processus de résolution. En particulier, nous nous sommes intéressés à l'algorithme de Coppersmith qui résout ce problème en temps polynomial lorsque ces racines sont suffisamment petites. La section 1.3 présente les résultats obtenus.

— **Problèmes PoSSo** : PoSSo pour *Polynomial System Solving*, ainsi on s'intéresse ici à la résolution de systèmes polynomiaux ayant un nombre fini de solutions et à leurs applications en cryptologie.

Dans ce cadre, ce sont des problèmes de cryptanalyse qui ont motivé nos recherches. En particulier, nous nous sommes intéressés à l'utilisation des symétries apparaissant lors de la résolution du DLP sur certaines courbes elliptiques. Les résultats sont détaillés dans la section 1.4.

Dans la suite de ce chapitre, une synthèse des résultats obtenus selon ces trois familles de problèmes est présentée. Ces résultats ont tous été obtenus après 2006 et ne font pas partie de ceux obtenus durant ma thèse.

1.2 Problèmes galoisiens

Une première partie des travaux relevant de la théorie de Galois font suite à ceux réalisés au cours de ma thèse. Les travaux présentés dans cette section ont été obtenus avec la collaboration de Jean-Gabriel Kammerer, Masanari Kida, Reynald Lercier, Sébastien Orange et Kazuhiro Yokoyama.

Ici on s'intéresse à la représentation formelle des racines d'un polynôme f en une variable. Plus exactement ce problème général est défini comme suit.

Problème 1. *Soit f un polynôme en une variable de degré $n > 0$ et à coefficients dans un corps $\mathbb{K}$. Donner un algorithme permettant d'exprimer les racines de f de manière à pouvoir les manipuler formellement.*

Une première réponse à ce problème s'inscrit directement dans les travaux de Lagrange, Galois, Abel et Jordan (voir [125] pour plus de détails) portant sur la mise en place d'une méthode générale permettant la résolution sous forme de radicaux d'équations polynomiales. Nous savons aujourd'hui que cette approche ne peut donner une solution en général. Seules les équations de groupes de Galois G résoluble ont des solutions exprimables de la sorte. Du point de vue de la complexité, on sait d'après Landau et Miller [98] que dans ce cas le calcul d'une telle représentation peut se faire en temps polynomial.

De manière générale, pour représenter les racines de f formellement, nous devons passer par la représentation de son corps de décomposition. Ceci peut se faire sous la forme d'un quotient $\mathbb{K}[x_1, \ldots, x_n]/\mathcal{M}$ où $\mathcal{M}$ est appelé *idéal des relations* algébriques des racines de f. Cet idéal est défini comme le noyau du morphisme de $\mathbb{K}[x_1, \ldots, x_n]$ dans la clôture algébrique $\bar{\mathbb{K}}$ qui à x_i fait correspondre la racine α_i (on peut noter que cette définition dépend de l'ordonnancement des racines). Ainsi, dès que l'on a une base de Gröbner de l'idéal $\mathcal{M}$, il devient possible de calculer formellement avec les images des x_i dans ce quotient, chacune représentant une racine de f.

Le calcul d'une base de Gröbner de $\mathcal{M}$ peut être réalisé à l'aide d'un algorithme simple procédant par factorisations successives dans une tour d'extensions construites à partir des racines de f. D'après [97] cet algorithme a une complexité polynomiale en la taille $|G|$ du groupe de Galois de f et la taille de f (norme L_2 des coefficients de f). Cette complexité est la même lorsque l'on souhaite calculer une représentation symétrique du groupe de Galois G.

Nos travaux dans ce domaine s'intéressent à un autre algorithme que celui-ci pour calculer la base de Gröbner de $\mathcal{M}$. Ils s'appuient sur une méthode décrite par Yokoyama dans [144] et

étendue dans [131] en collaboration avec ce dernier. Cette méthode repose sur l'interpolation des polynômes prenant part à la base de Gröbner de $\mathcal{M}$ et utilise l'action du groupe de Galois G sur des approximations p-adiques des racines de f pour réaliser ce calcul.

Les résultats principaux que nous avons obtenus dans la suite de ces travaux ont permis d'exploiter au mieux cette action de groupe lors du calcul de la base de Gröbner de $\mathcal{M}$. Le gain en efficacité pratique atteint un facteur de l'ordre de plusieurs centaines. Concernant la complexité asymptotique, nous montrons que le coût combinatoire des calculs nécessaires à l'utilisation de cet action reste bornée par $|G|$ et donc ne modifie pas la complexité générale. Dans la suite nous détaillons plus précisément chacun des résultats obtenus dans cet axe de recherche.

Dans la continuité de [131], nous montrons dans [132] comment adapter l'approche modulaire à une approche multi-modulaire. Le problème se posant ici est que les approximations des racines modulo différents premiers ne coïncident pas forcément. Nous proposons une méthode de reconnaissance des bonnes associations de ces différentes images en utilisant des invariants algébriques liés aux différentes classes de conjugaison de groupe de Galois de f : les résolvantes de Lagrange.

Dans [120] nous introduisons un nouvel objet associé à chacune des classes de conjugaison d'un groupe de permutations. Cet objet est appelé *schéma de calcul* et permet, comme son nom l'indique, de calculer le plus efficacement possible une base de Gröbner de $\mathcal{M}$ en utilisant au mieux l'action du groupe de Galois. En particulier, nous introduisons des techniques permettant de remplacer certains calculs par des actions de groupe sur des polynômes. Dans [121] nous montrons comment utiliser ces techniques pour améliorer l'efficacité de l'arithmétique dans les corps de nombres définis par une tour d'extensions.

L'article [93] résout un problème d'isomorphisme entre deux familles de polynômes génériques. Un polynôme générique pour un groupe G est un polynôme à coefficients paramétrés tel que toute extension galoisienne de groupe G peut être définie comme le corps de décomposition d'une de ses instanciations. Dans cet article nous donnons explicitement un ensemble de paramètres qui, pour deux familles de tels polynômes, permet de définir le même corps de nombres. Pour obtenir un tel résultat, des calculs explicites avec les racines des polynômes génériques sont utilisés.

Les résultats présentés au-dessus sont centrés sur le calcul du corps de décomposition et ont pour objectif principal de fournir des moyens de calcul effectif avec les racines d'un polynôme. Ils n'ont pas d'application directe en cryptologie mais si l'on considère une autre représentation des racines la situation devient favorable à de telles applications. Dans l'article [90], nous utilisons l'expression sous forme de radicaux des racines de polynômes de groupe de Galois résoluble pour définir des fonctions d'encodage vers des courbes algébriques. Ce travail s'attaque au problème de définir de manière sécurisée, i.e. sans fuite d'information, des protocoles cryptographiques ayant besoin de mettre en relation des chaînes de caractères avec des points d'une courbe algébrique définie sur un corps fini.

Ces travaux galoisiens ont été présentés lors de cours ou conférences invités. En particulier pour un cours donné lors des *Journées Nationales de Calcul Formel* [125] en 2008 et pour une

conférence de vulgarisation lors d'une séance de *Mathematic Park* [126] en 2011, organisée pour le bicentenaire de la naissance d'Évariste Galois.

Au chapitre 2, un résumé des travaux [120, 90] est présenté et les articles [132] et [93] sont donnés en annexe.

1.3 Problèmes diophantiens

Les résultats obtenus ici sont issus de travaux en commun avec Guillaume Barbu, Alberto Battistello, Jingguo Bi, Jean-Sébastien Coron, Guillaume Dabosville, Jean-Charles Faugère, Christophe Giraud, Christoper Goyet, Raphaël Marinier, Phong Q. Nguyen, Soline Renner et Rina Zeitoun.

Les problèmes auxquels nous nous sommes intéressés dans ce contexte proviennent tous de la cryptologie. Plus exactement, ce sont ceux qui se modélisent à travers un système d'équations diophantiennes et dont les solutions entières fournissent le secret attendu. Le problème général, en une équation, peut se définir comme suit.

Problème 2. *Dioph(n, d, N) : Soit f polynôme de degré d en n variables à coefficients entiers et N un module entier de factorisation inconnue. On demande de trouver les solutions de l'équation*

$$f(x_1, \ldots, x_n) = 0 \mod N\,.$$

On notera Dioph(n,∞) le problème correspondant à la recherche de solutions entières à une équation diphantienne en n variables non modulaire.

En toute généralité, ce problème est difficile, il est même démontré indécidable dans sa version décisionnelle (résultat de Matiyasevich sur le 10ème problème de Hilbert). Plusieurs cryptosystèmes reposent sur la difficulté de résoudre une instance de ce problème, le plus connu étant RSA. En effet, à partir de l'exposant de chiffrement e et le module public N, retrouver le message clair correspondant à un chiffré c revient à résoudre l'équation

$$x^e - c = 0 \mod N$$

On peut alors se demander quels sont les versions faibles de ce problème dont la résolution peut se faire en temps polynomial. Dans le cadre de l'analyse de RSA, Coppersmith propose dans [35] un algorithme de complexité polynomiale pour résoudre le problème suivant.

Problème 3. *On demande de trouver les solutions entières x de Dioph($1, d, N$) qui soient plus petites, en valeur absolue, que $N^{1/d}$.*

L'impact d'un tel résultat en cryptanalyse est immédiat car il permet d'attaquer RSA lorsque l'attaquant a obtenu, par exemple, des informations partielles sur la clé privée à l'aide d'une analyse par canaux auxiliaires. L'idée de base de Coppersmith pour résoudre ce problème peut s'exprimer de la manière suivante : comme il est difficile de trouver la solution x d'une équation

Algorithme 1: Algorithme de Coppersmith

entrée: Un problème Dioph$(1, d, N)$ modélisé par f
sortie : Une petite racine entière x à ce problème, si elle existe

À partir de f on engendre un certain nombre de polynômes g_i s'annulant en x modulo N^h;
Des coefficients des g_i on déduit la base B d'un réseau euclidien L;
(*) On extrait un vecteur *suffisamment* court v du réseau L;
On construit un polynôme F dont les coefficients sont déduits de v;
La solution x doit être une racine de F vu comme un polynôme à coefficients entiers;
retourne x *si elle a été trouvée à la dernière étape*

FIGURE 1.1 – Résolution du problème 3

modulaire, essayons de trouver un polynôme en une variable qui s'annule en x sur les entiers. L'outil algorithmique permettant ce calcul est la réduction d'un réseau euclidien construit à partir de l'équation de départ. L'algorithme peut se résumer comme suit.

L'étape marquée d'un astérisque est *a priori* de complexité exponentielle. En effet, trouver trouver un vecteur court dans un réseau est connu pour être NP-difficile depuis les travaux de Ajtai. Toute la subtilité se trouve dans le *suffisamment* court. Coppersmith utilise l'algorithme LLL [104] pour réduire la base B et prend pour v le vecteur le plus court issu de cette réduction. Ce calcul se faisant en temps polynomial et il montre que si l'ensemble des g_i est bien choisi, même si v est une approximation exponentiellement mauvaise d'un vrai vecteur court de L, elle suffit à résoudre le problème 3. Plus généralement cette approche peut être adaptée pour trouver des petites racines à des problèmes Dioph(n, d, N). En particulier Coppersmith donne un algorithme dans [34] pour résoudre en temps polynomial le cas Dioph$(2, d, \infty)$. Dans les autres cas, les approches sont à ce jour toujours heuristiques.

Le résultat majeur que nous avons obtenu dans ce contexte est l'amélioration de l'algorithme 1 de Coppersmith. Nous montrons dans [15] comment la structure de la base B du réseau euclidien utilisé dans cet algorithme permet d'en améliorer son efficacité. Plus exactement, nous utilisons un calcul approché par troncature (on ne garde que les parties hautes des coordonnées des éléments de B) pour effectuer la réduction de réseau. Ce calcul permettant d'obtenir le résultat escompté puisque nous montrons que B est triangulaire à diagonale équilibrée. Aussi, nous mettons en évidence une structure cachée reliant différentes versions de la base B pour accélérer la phase énumérative de l'algorithme. Cette phase est nécessaire si l'on souhaite obtenir des solutions atteignant la borne théorique. Le gain en complexité est de l'ordre de $\log^2(N)/d^2$ et les gains pratiques sont de l'ordre de plusieurs centaines pour des modules N de tailles 1024 bits.

Dans l'article [6] nous montrons comment terminer une attaque combinée sur une implantation de RSA-CRT. Plus exactement, nous montrons comment récupérer la clé privée malgré l'application de contremesures contre les attaques par observation (de la consommation de courant par exemple) et celles par injection de fautes. Notre résultat repose sur l'hypothèse que l'attaquant est capable d'effectuer deux telles attaques au cours de la même exécution. Hypothèse réalisable en pratique aujourd'hui. L'algorithme 1 de Coppersmith est utilisé ici pour accélérer les calculs (seulement la moitié du secret doit être retrouvée grâce aux canaux auxiliaires) et ainsi

diviser par deux le temps total de l'attaque (on passe de deux heures à une heure).

Dans [38] nous attaquons le problème de la factorisation d'entier de la forme $p^r q^s$ avec p et q premiers et r et s suffisamment grand. Cet article résout un problème laissé ouvert dans [23] et répond à la question de savoir si de tels modules pouvaient être utilisés dans des versions plus efficaces de RSA. Nous montrons qu'une telle factorisation est possible en temps polynomial dès que $r \geq \log^3(p)$. Ce résultat de complexité est essentiellement asymptotique, des exposants aussi grands ne peuvent être utilisés en pratique. Il a tout de même le mérite de montrer une potentielle faiblesse de RSA pour de tels modules. Un des ingrédients permettant d'arriver à nos fins est l'algorithme de Coppersmith.

Dans l'article [66], nous nous attaquons au problème de la *factorisation implicite* introduit par May et Ritzenhofen dans [110]. Ce problème, dans le cas de deux modules, suppose que deux entiers RSA $N_1 = p_1 q_1$, $N_2 = p_2 q_2$ sont tels que p_1 et p_2 partagent une partie s de bits de poids forts en commun. L'attaquant ne connaît pas la valeur exacte de s, il a juste connaissance de l'existence de cette partie commune. L'attaque peut alors se modéliser facilement sous la forme d'un problème du type Dioph$(3, 3, \infty)$ défini par le polynôme

$$f(x, y_1, y_2) = y_1 y_2 x + y_1 N_2 - y_2 N_1$$

dont les solutions sont $(\tilde{p_1} - \tilde{p_2}, q_1, q_2)$ où $\tilde{p_i} = p_i - s$. Ainsi dès que la partie partagée s est suffisamment grande et les q_i suffisamment petits, cette solution devient une petite solution du problème et une stratégie à la Coppersmith peut être appliquée. C'est cette méthode heuristique que proposent Sarkar et Maitra dans [134]. Dans [66] nous exhibons un réseau euclidien dont un court vecteur contient une solution au problème. Ainsi nous obtenons un algorithme plus efficace pour résoudre le problème en général et qui devient déterministe dans le cas de deux modules. Par rapport à May et Ritzenhofen dans [110] où ils traitaient le cas du bloc partagé de poids faible, dans [66] nous montrons comment réaliser l'attaque pour des blocs situés n'importe où dans p_1 et p_2.

Dans l'article [64] nous considérons une hypothèse implicite au moment de la génération des aléas utilisés pour réaliser des signatures avec (EC)DSA (égalité entre certains bits de plusieurs aléas sans connaître leur valeur exacte). Nous montrons que cette hypothèse suffit à obtenir des résultats équivalents à ceux obtenus par Nguyen et Shparlinski dans [117] où l'attaquant avait la connaissance de certains bits des aléas (voir aussi les résultats récents [4, 81]).

La conférence invitée [129] reprend quelques résultats présentés ici. Au chapitre 3 une partie des travaux [15, 6] sera détaillée et les articles [66, 64] sont donnés en annexe.

1.4 Problèmes PoSSo

Les travaux présentés ici ont été réalisés avec Claude Carlet, Jean-Charles Faugère, Pierrick Gaudry, Christopher Goyet, Louise Huot, Antoine Joux, Christophe Petit, Ludovic Perret et Vanessa Vitse.

On s'intéresse ici à la résolution de systèmes polynomiaux ayant un nombre fini de solutions. Plus exactement, on cherche à représenter les solutions d'un tel système pour qu'elles puissent être extraites facilement. Dans le cadre applicatif qui nous intéresse ici, on spécifie le corps de base $\mathbb{K}$ à un corps fini. Certains résultats s'étendent au corps des rationnels par exemple. Le problème général s'énonce donc ainsi.

Problème 4. *Étant donné un système de polynômes ayant un nombre fini de solutions toutes simples*

$$\mathcal{S} : \{f_1 = \cdots = f_s = 0\}$$

avec $f_1, \ldots, f_s \in \mathbb{K}[x_1, \ldots, x_n]$. *On demande de trouver* n *polynômes en une variable* $h_i \in \mathbb{K}[x_i]$ *tels que les solutions de* $\mathcal{S}$ *soient en bijection avec celles de*

$$\{x_1 - h_1(x_n) = \cdots = x_{n-1} - h_{n-1}(x_n) = h_n(x_n) = 0\}\,.$$

Clairement, lorsque l'on obtient une représentation des racines de $\mathcal{S}$ comme spécifiées dans le problème précédent, les solutions peuvent être listées facilement après la factorisation du polynômes h_n. La complexité de cette énumération est polynomiale en le degré D de h_n (à noter que les polynômes h_i ($i < n$) ont un degré borné par D puisqu'ils peuvent être réduits modulo h_n). De plus, si l'on cherche les racines dans $\mathbb{K}$ (qui est un corps fini), cette complexité (voir [76]) est de l'ordre de $\widetilde{O}(D)$ opérations arithmétiques (où la notation $\widetilde{O}$ signifie que les facteurs logarithmiques en D et polynomiaux en n sont négligés). Puisque le polynôme h_n encode toutes ces solutions, il est clair que le degré D représente aussi le nombre total des solutions du système $\mathcal{S}$.

Pour résoudre le problème 4, l'outil algorithmique de base est le calcul de bases de Gröbner. D'après les travaux de Lakshman et Lazard [96], la complexité de ce calcul est polynomiale en D. Plus exactement, l'algorithme [63] utilisé classiquement aujourd'hui pour résoudre ce problème peut être décrit comme suit :

Algorithme 2: Résolution PoSSo

entrée: Un système $\mathcal{S} \subset \mathbb{K}[x_1, \ldots, x_n]$.
sortie : Une base de Gröbner pour LEX de $\mathcal{S}$ donnant la représentation souhaitée des solutions de $\mathcal{S}$.

Calcul d'une base de Gröbner pour l'ordre DRL de $\langle \mathcal{S} \rangle$;
À partir de la base de Gröbner DRL, calculer une base de Gröbner LEX de $\langle \mathcal{S} \rangle$;
retourne *la base de Gröbner LEX pour* $\mathcal{S}$

FIGURE 1.2 – Résolution du problème 4

La première étape est un calcul de base de Gröbner, la seconde un calcul de changement d'ordre. On procède ainsi pour maîtriser au mieux les degrés des polynômes apparaissant au cours de la première étape et ainsi obtenir une meilleure complexité. Les meilleurs algorithmes pour résoudre cette étape sont dûs à Faugère [56, 57] et ont une complexité, sous une hypothèse de régularité du système donné en entrée, de l'ordre de $O(D^\omega)$. (Ici pour simplifier les études

de complexité, on suppose que le nombre de solutions D est maximal, i.e. il atteint la borne de Bézout, ce qui est une situation générique.)

La seconde étape repose essentiellement sur des calculs d'algèbre linéaire dans le quotient $\mathbb{K}[x_1, \ldots, x_n]/\langle \mathcal{S} \rangle$; calculs rendus possibles grâce à la base de Gröbner DRL calculée à l'étape précédente. L'algorithme FGLM introduit dans [63] permettant un tel changement d'ordre a une complexité de $O(nD^3)$.

Le fait que la base de Gröbner LEX fournisse la sortie attendue au problème 4 est soumis à une hypothèse particulière. Cette forme attendue est appelée *Shape Position* et on peut montrer qu'après un changement de coordonnées générique, on peut toujours se ramener à cette situation [95, 80, 11]. On peut donc supposer que la base de Gröbner LEX ainsi calculée est bien en Shape Position.

Ainsi, nous déduisons, sous des hypothèses de régularité pour le système donné en entrée et de Shape Position pour la sortie, que le problème 4 peut être résolu en temps $O(nD^3)$. En effet, c'est la seconde étape qui domine le calcul.

Dans certains contextes, il est possible d'obtenir une meilleure complexité. Par exemple, pour la recherche de racines réelles d'un système de polynômes, il est possible de les calculer en $O(12^n D^2)$ lorsqu'elles sont en nombre logarithmique en D [112]. Ou encore, lorsque la structure multiplicative du quotient est connue, alors il est possible (voir [26]) de résoudre le problème 4 en $O\left(n2^n D^{\frac{5}{2}}\right)$. Dans [55] Faugère et Mou proposent un algorithme essentiellement quadratique en D pour le changement d'ordre mais repose sur l'exploitation de la structure potentiellement creuse des matrices apparaissant en cours de calcul. Mais en toute généralité, avant le résultat [61] la meilleure complexité connue était essentiellement cubique en D.

En effet, dans [61] (voir aussi la version étendue [60]) nous proposons un nouvel algorithme de changement d'ordre permettant de résoudre efficacement la seconde étape de l'algorithme 2. Cet algorithme fournit le premier résultat de complexité sous cubique depuis l'introduction de l'algorithme FGLM [63] et ainsi ramène la résolution du problème 4 à une complexité de $\widetilde{O}(D^\omega)$ au total. L'ingrédient clé qui nous a permis de passer d'une complexité cubique à D^ω est l'adaptation de [55] en utilisant un résultat de Keller-Gehrig [92] pour le calcul rapide de produits de matrices par des vecteurs. L'utilisation d'une structure particulière des bases DRL est aussi essentielle à notre algorithme puisqu'elle permet de se passer d'une bonne part des calculs normalement nécessaires dans [63].

Les autres résultats de cette section se focalisent sur la première étape de l'algorithme 2. L'article [62] peut être vu comme un des plus marquants. Il décrit une étude complète sur l'utilisation des structures liées à des actions de groupes sur les générateurs d'un idéal. Le problème sous-jacent à cette étude est issu d'un problème intervenant dans le calcul d'indice pour la résolution du problème du logarithme discret (DLP) sur les courbes elliptiques. Les meilleurs algorithmes connus pour résoudre ce problème sont de complexité exponentielle. Nous montrons comment les points de 2-torsion permettent de gagner un facteur exponentiel par rapport aux travaux de Gaudry et Diem [78, 48, 47].

L'étude précédente est réalisée pour des corps finis de grande caractéristique, dans [65] nous étendons ces résultats au cas de la caractéristique 2. Un autre résultat important de cet article

est le record de calcul réalisé lors de ce travail. En effet, dans ce contexte, la modélisation du problème DLP passe par le calcul d'un polynôme de sommation en $n+1$ variables pour résoudre le problème de décomposition en n points. Aucun polynôme de sommation pour $n > 6$ n'avait jamais été calculé, ici nous montrons comment l'utilisation des symétries permet de calculer celui pour $n = 8$ en une quarantaine d'heures de calcul. Sachant que le précédent est calculable en 380 secondes, on se rend bien compte du caractère exponentiel de ce calcul. Les techniques utilisées pour arriver au bout de ce calcul sont issus du calcul formel et plus particulièrement de résultats de l'interpolation rapide de polynômes.

L'article [68] présente une étude préliminaire sur la résolution du DLP sur les courbes elliptiques en caractéristique 2 basée sur une version de du calcul d'indice dû à Diem [48, 47]. L'utilisation de la résolution des systèmes polynomiaux est central. Ici aucune hypothèse n'est faite sur la nature du corps de base (en particulier l'extension peut être de degré premier). Nous montrons que les systèmes polynomiaux intervenant dans cette étude possèdent une structure multi-homogène et obtenons ainsi une amélioration en complexité par rapport aux résultats de Diem. Cependant, ceci ne nous permet pas de montrer que cette approche est meilleure que les résolutions génériques du DLP. En effet, nous mettons aussi en évidence la présence de *chutes de degrés* lors de la résolution de ces systèmes. Ceci montre en particulier qu'il est difficile d'envisager un gain conséquent en efficacité sans utiliser plus de structures spécifiques au problème (comme nous l'avons fait dans [65]).

Dans [30] nous étudions la résolution des systèmes issus de l'analyse de sécurité de chiffrement par blocs. Nous étudions en particulier les structures des systèmes obtenus à l'aide d'attaques par canaux auxiliaires ; les observations physiques permettant de récupérer les poids de Hamming de secrets tout au long du calcul. Nous déduisons de cette étude un critère permettant d'évaluer la sécurité des boite-S (briques de base des chiffrements par bloc) face à des attaques combinant l'approche algébrique et des informations issues d'attaques par canaux auxiliaires.

Ces travaux furent présentés lors de conférences invitées [128, 127]. Au chapitre 4 une partie des travaux [62, 65] sera détaillée. Les articles [30] et [61] sont donnés en annexe.

1.5 Liste des publications depuis 2006

Pour faciliter l'identification des articles publiés après ma thèse (2006), ils sont listés ci-après. Ils sont aussi rappelés, avec la même numérotation, dans la bibliographie à la fin de ce document. Dans la version numérique de ce document, des liens hypertexte dans chacune de ces références permettent d'obtenir les publications correspondantes (et les slides pour certains cours ou conférences invitées).

Revues internationales avec comité de lecture

[30] Claude CARLET, Jean-Charles FAUGÈRE, Christopher GOYET et Guénaël RENAULT. "Analysis of the algebraic side channel attack". In : *J. Cryptographic Engineering* 2.1 (2012). PDF, p. 45–62.

[62] Jean-Charles FAUGÈRE, Pierrick GAUDRY, Louise HUOT et Guénaël RENAULT. "Using Symmetries in the Index Calculus for Elliptic Curves Discrete Logarithm". In : *J. Cryptology* 27.4 (2014). PDF, p. 595–635.

[93] Masanari KIDA, Guénaël RENAULT et Kazuhiro YOKOYAMA. "Quintic polynomials of Hashimoto-Tsunogai, Brumer and Kummer". In : *Int. J. Number Theory* 5.4 (2009). PDF, p. 555–571. ISSN : 1793-0421.

[121] Sébastien ORANGE, Guénaël RENAULT et Kazuhiro YOKOYAMA. "Efficient arithmetic in successive algebraic extension fields using symmetries". In : *Math. Comput. Sci.* 6.3 (2012). PDF, p. 217–233. ISSN : 1661-8270.

Actes de conférences internationales avec comité de programme

[6] Guillaume BARBU, Alberto BATTISTELLO, Guillaume DABOSVILLE, Christophe GIRAUD, Guénaël RENAULT, Soline RENNER et Rina ZEITOUN. "Combined Attack on CRT-RSA - Why Public Verification Must Not Be Public?" In : *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings*. PDF. 2013, p. 198–215.

[15] Jingguo BI, Jean-Sébastien CORON, Jean-Charles FAUGÈRE, Phong Q. NGUYEN, Guénaël RENAULT et Rina ZEITOUN. "Rounding and Chaining LLL: Finding Faster Small Roots of Univariate Polynomial Congruences". In : *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*. PDF. 2014, p. 185–202.

[38] Jean-Sébastien CORON, Jean-Charles FAUGÈRE, Guénaël RENAULT et Rina ZEITOUN. "Factoring $N = p^r q^s$ for Large r and s". In : *Topics in Cryptology - CT-RSA 2016 - The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings*. PDF. 2016, p. 448–464.

[61] Jean-Charles FAUGÈRE, Pierrick GAUDRY, Louise HUOT et Guénaël RENAULT. "Sub-cubic change of ordering for Gröbner basis: a probabilistic approach". In : *International Symposium on Symbolic and Algebraic Computation, ISSAC '14, Kobe, Japan, July 23-25, 2014*. PDF. 2014, p. 170–177.

[64] Jean-Charles FAUGÈRE, Christopher GOYET et Guénaël RENAULT. "Attacking (EC)DSA Given Only an Implicit Hint". In : *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*. PDF. 2012, p. 252–274.

[65] Jean-Charles FAUGÈRE, Louise HUOT, Antoine JOUX, Guénaël RENAULT et Vanessa VITSE. "Symmetrized Summation Polynomials: Using Small Order Torsion Points to Speed Up Elliptic Curve Index Calculus". In : *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*. PDF. 2014, p. 40–57.

[66] Jean-Charles FAUGÈRE, Raphaël MARINIER et Guénaël RENAULT. "Implicit Factoring with Shared Most Significant and Middle Bits". In : *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*. PDF. 2010, p. 70–87.

[68] Jean-Charles FAUGÈRE, Ludovic PERRET, Christophe PETIT et Guénaël RENAULT. "Improving the Complexity of Index Calculus Algorithms in Elliptic Curves over Binary Fields". In : *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*. PDF. 2012, p. 27–44.

[90] Jean-Gabriel KAMMERER, Reynald LERCIER et Guénaël RENAULT. "Encoding points on hyperelliptic curves over finite fields in deterministic polynomial time". In : *Pairing-based cryptography—Pairing 2010*. T. 6487. Lecture Notes in Comput. Sci. PDF. Springer, Berlin, 2010, p. 278–297.

[120] Sébastien ORANGE, Guénaël RENAULT et Kazuhiro YOKOYAMA. "Computation schemes for splitting fields of polynomials". In : *ISSAC 2009—Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation*. PDF. ACM, New York, 2009, p. 279–286.

[132] Guénaël RENAULT et Kazuhiro YOKOYAMA. "Multi-modular algorithm for computing the splitting field of a polynomial". In : *ISSAC 2008*. PDF. ACM, New York, 2008, p. 247–254.

Travaux récents, cours et conférences invités, posters

[9] Lucas BARTHÉLÉMY, Ninon EYROLLES, Guénaël RENAULT et Raphaël ROBLIN. "Binary Permutation Polynomial Inversion and Application to Obfuscation Techniques". In : *To appear in 2nd ACM International Workshop on Software Protection, SPRO 2016, Vienna, Austria, October 28, 2016*. PDF. 2016, p. 1–9.

[60] Jean-Charles FAUGÈRE, Pierrick GAUDRY, Louise HUOT et Guénaël RENAULT. *Polynomial Systems Solving by Fast Linear Algebra*. PDF. 2013.

[130] Guénaël RENAULT et Tristan VACCON. "On the p-adic stability of the FGLM algorithm". In : *CoRR* abs/1602.00848 (2016). PDF.

[125] Guénaël RENAULT. "Introduction à la Théorie de Galois Effective". In : *JNCF'08: Journées Nationales du Calcul Formel (online)*. PDF, Slides. Marseille, France, oct. 2008, p. 141–197.

[126] Guénaël RENAULT. *Introduction à l'Algorithmique Galoisienne*. Invited talk at Mathematik Park (Institut Henri Poincaré), Slides. 2011.

[127] Guénaël RENAULT. *On polynomial systems with structures related to the ECDLP*. Invited talk during the Conference Effective Moduli Spaces and Applications to Cryptography (Rennes, France). 2014.

[128] Guénaël RENAULT. *On Using Torsion Points in the Elliptic Curve Index Calculus*. Invited talk during the 18th Workshop On Elliptic Curve Cryptography (ECC 2014), Slides. 2014.

[129] Guénaël RENAULT. *The Heuristic Coppersmith Technique from a Computer Algebra Point of View*. Invited talk during the SIAM Conference on Applied Algebraic Geometry (Fort Collins, Colorado, USA), Slides. 2013.

[59] Jean-Charles FAUGÈRE, Pierrick GAUDRY, Louise HUOT et Guénaël RENAULT. "Fast change of ordering with exponent ω". In : *(Poster abstract) ACM Commun. Comput. Algebra* 46 (sept. 2012). PDF, p. 92–93.

CHAPTER 2

Contribution to Effective Galois Theory

In this chapter, some results on the computation of the roots of a univariate polynomial are presented. They are all related to Problem 1 introduced in Chapter 1. Our papers corresponding to the subject of this chapter are [121, 90, 120, 93, 132, 125]. In the sequel, we detail the results of [120, 90].

2.1 Introduction

In Section 2.2 we consider the problem of computing efficiently the splitting field of a univariate polynomial $f \in \mathbb{K}[x]$. A well known bootstrapping algorithm for this task consists of factoring f over $\mathbb{K}_1 = \mathbb{K}[t]/f(t)$, adjoining a root of f to $\mathbb{K}$, computing a primitive element for this field over $\mathbb{K}$, and repeating this procedure until f splits completely. This simple algorithm has a running time which is polynomial in the size of f (the L_2-norm of its coefficients) and the size of its Galois group. In general, this is the best we can do for this computation (see [97, 98]).

In the articles [132, 131, 124], we presented some techniques, obtained from the action of the Galois group, which help such a computation and we also use them when this representation is computed by interpolation. In this chapter, we give a general presentation of these techniques and we present a complexity result (see Theorem 2.10) stating that their computation do not impact the global cost of the computation. In practice, the use of these techniques provides a speed-up of a factor up to several hundreds. These results are part of the article [93].

Another way of representing roots of a univariate polynomial is, when it is possible, by radicals. In Section 2.3, we present some applications in cryptography of such a representation. The main result of this part (coming from [90]) is Theorem 2.13 stating that a family of polynomials with solvable Galois group can be used to define an encoding on an hyperelliptic curve. These encodings are deterministic and thus avoid possible leaks of secret during the computation. Such a property is of great importance in the context of embedded cryptography where an attacker can "listen" the chip during the encryption/decryption phase.

2.2 Splitting field computation

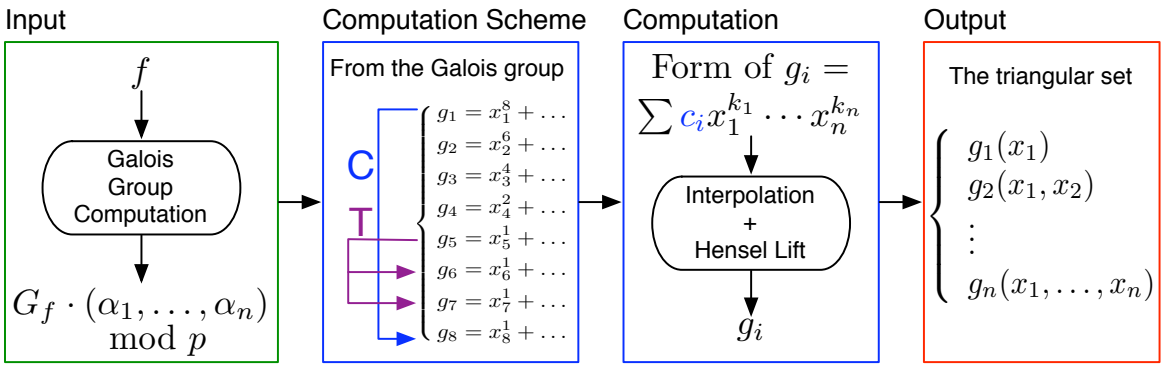
The computation of the splitting field of a polynomial f plays an important role in Galois theory and more generally in algorithmic number theory. It is the smallest field where all the roots of f lie. Providing a suitable representation of this field allows symbolic computations with all the roots of the polynomial and thus provides a solution to the problem 1.

Such a representation comes from computer algebra and more precisely from Gröbner basis theory. Let f be a univariate irreducible and separable polynomial of degree n with coefficients in a field $\mathbb{K}$ and $\alpha_1, \ldots, \alpha_n$ its roots. A natural representation of the splitting field of f is the quotient algebra

$$\mathbb{K}(\alpha_1, \ldots, \alpha_n) \simeq \mathbb{K}[x_1, \ldots, x_n]/\mathcal{M}$$

where $\mathcal{M}$ is the kernel of the surjective morphism from $\mathbb{K}[x_1, \ldots, x_n]$ to $\mathbb{K}(\alpha_1, \ldots, \alpha_n)$ which maps x_i to α_i. The ideal $\mathcal{M}$, called a **splitting ideal** of f, is zero-dimensional and maximal. Knowing a Gröbner basis of $\mathcal{M}$ allows computations in this quotient algebra by means of linear algebra operations (e.g. [42, 12]) and then symbolic operations with the roots of f.

In [132] we propose an algorithm for computing the splitting field of a monic irreducible polynomial f with coefficients in $\mathbb{K} = \mathbb{Q}$ (more generally, these methods can be applied in any global field). This algorithm takes as input the polynomial f and, as a first preparatory step, computes the action of its Galois group over p-adic approximations of its roots. The other steps are based on the relationship between the representation of the splitting field by a Gröbner basis and the action of the corresponding Galois group on this basis. The core of this new approach, called *computation scheme* (see section 2.2.1), uses the internal symmetries of the problem in order to speed up the Gröbner basis computation. This scheme is computed from the knowledge of a permutation representation of the Galois group G of f and provides a shape of the Gröbner basis of the splitting ideal of f. From this shape, the algorithm effectively computes the basis by interpolating its coefficients (see also [111, 144, 123, 103] and more generally [44] for interpolation strategies). Our general method can be depicted as the following drawing.



The efficiency of this algorithm heavily depends on this computation scheme which depends itself on the choice of the representative of the conjugacy class of G in S_n. When we choose the best representative of a given permutation group, the gain in efficiency may be very large.

Actually, this can be measured as a function of the number of coefficients we need to compute by interpolation. Without using the computation scheme, this number is bounded by $|nG|$ the size of the group times its degree. Using the computation scheme, this number can be reduced to a function $c(G)$ depending on G, for example we have the following comparison table.

Group	$7T_6$	$8T_{48}$	$9T_{25}$	$9T_{29}$	$9T_{32}$	$10T_{36}$	$10T_{39}$
nG	7×2520	8×1344	9×324	9×648	9×1512	10×1920	10×3840
$c(G)$	2520	336	27	$18 + 648$	2×1512	$10 + 960$	10

As one can see, depending on the group G, this gain of efficiency may be huge. The problem is that the quantity $c(G)$ does not depends on G only but on its representative. Thus, finding this representative may have a combinatorial cost. In [131, 132] we overcome this problem by tabulating these best representatives. From an algorithmic point of view, tabulating is not really satisfying, thus we propose in [120] a method which efficiently identifies such representatives. We present this method in the sequel.

2.2.1 Computation schemes

For the natural group action Ψ of S_n on $\mathbb{Q}[x_1, \ldots, x_n]$ which permutes the x_i's by acting on the indices, the stabilizer of $\mathcal{M}$ is the permutation representation of the Galois group which will be denoted by G:

$$G = \{\sigma \in S_n \,|\, \forall g \in \mathcal{M},\, \sigma.g \in \mathcal{M}\}.$$

Proposition 2.1. *Let σ be a permutation of S_n and let $G^\sigma = \sigma G \sigma^{-1}$. The ideal $\sigma.\mathcal{M} = \{\sigma.f \,|\, f \in \mathcal{M}\}$ is the splitting ideal corresponding to the roots $\sigma.\alpha = (\alpha_{\sigma.1}, \ldots, \alpha_{\sigma.n})$ and the corresponding representation of the Galois group is G^σ (a conjugate of G).*

A first result coming from classical Galois theory (see for example [144]) shows that the basis $\mathcal{G}$ of $\mathcal{M}$ is triangular for the lexicographical order induced by $x_1 < \cdots < x_n$ (see [100]). That is, $\mathcal{G}$ is given as a set of n polynomials $\{f_1, \ldots, f_n\}$ such that f_i has a power of x_i as leading term and is separable as a polynomial in x_i. Moreover, we can deduce from G the degree d_i of the leading term of each f_i. Let E be a subset of $\{1, \ldots, n\}$, we denote by $\mathrm{Stab}_G(E)$ the pointwise stabilizer in G of E (that is the subgroup of G given by $\{\sigma \in G \mid \sigma(e) = e\, \forall e \in E\}$). We have the following classical result:

$$d_i = |\,\mathrm{Stab}_G(\{1, 2, \ldots, i-1\}|/|\,\mathrm{Stab}_G(\{1, 2, \ldots, i\})|. \tag{2.2.1}$$

Thus, the degrees d_i of elements in a Gröbner basis (not necessarily reduced but minimal) $\mathcal{G}$ of $\mathcal{M}$ can be deduced only from the stabilizer G of this ideal. This result gives a very general shape for $\mathcal{G}$. In the sequel we show how to improve this shape by inspecting the property of G.

Let i be an integer in $[\![1, n]\!]$. A sequence r of couples $[(i_1, k_1), (i_2, k_2), \ldots, (i_s, k_s)]$ with $\{i_1 < i_2 < \ldots < i_s = i\}$ a part of $\{1, \ldots, i\}$ and $k_j \leqslant d_{i_j}$ is said to be an **i-relation** if there exists a polynomial $g_i \in \mathbb{K}[x_{i_1}, \ldots, x_{i_s}]$ such that $\alpha_i^{k_i+1} + g_i(\alpha_1, \ldots \alpha_i) = 0$ with $\deg_{x_{i_j}}(g_i) \leqslant k_j$ (note that we must have $k_i = k_{i_s} = d_i - 1$ and $k_j < d_{i_j}$ for $j < s$). The polynomial g_i is called the *the tail* polynomial of this i-relation. Since the i-relation depends on the roots $\alpha_1, \ldots, \alpha_n$, the existence of a given i-relation depends only of the Galois group G, more exactly we have:

Proposition 2.2. *There exists an i-relation $[(i_1, k_1), \ldots, (i_s, k_s)]$ as soon as $\forall j \in [\![1, s]\!]$, $k_j = \frac{|\mathrm{Stab}_G(\{i_1,\ldots,i_{j-1}\})|}{|\mathrm{Stab}_G(\{i_1,\ldots,i_j\})|}$ and $\frac{|\mathrm{Stab}_G(\{i_1,\ldots,i_{s-1}\})|}{|\mathrm{Stab}_G(\{i_1,\ldots,i_s\})|} = d_i$.*

An important quantity attached to an i-relation is its **size** which corresponds to the product $k_{i_1} \times \cdots \times k_{i_s}$ and represents the maximal number of monomials of the corresponding polynomials f_i. Thus, in order to minimize the cost for computing the triangular basis $\mathcal{G}$ by *indeterminate coefficients strategy*, we need to know the best i-relation possible, that is the one with minimal size. Such an i-relation is said to be **minimal**. We now see how to avoid some of these computations by using two techniques.

The *computation scheme* can be seen as a set of techniques which help to compute $\mathcal{G}$ from G. The first technique, called **Cauchy technique**, is based on the so called *generalized Cauchy modules* (see [123]):

Definition 2.3. *Let $\mathcal{G} = \{f_1, \ldots f_n\}$ be a triangular basis of $\mathcal{M}$ and $\{i = i_1 < \cdots < i_r\}$ the orbit of i under the action of $\mathrm{Stab}_G(\{1, \ldots, i-1\})$. The d_i generalized Cauchy modules of f_i are inductively defined by $C_{i_1}(f_i) = f_i(x_{i_1})$ and for $k \geqslant 2$ the polynomial $C_{i_1,\ldots,i_k}(f_i)$ is given by the divided difference*

$$\frac{C_{i_1,\ldots,i_{k-1}}(f_i)(x_{i_k}) - C_{i_1,\ldots,i_{k-1}}(f_i)(x_{i_{k-1}})}{x_{i_k} - x_{i_{k-1}}}.$$

From these constructions, we can deduce polynomials of $\mathcal{G}$ from other ones. The following result explains this relation.

Proposition 2.4. *The Cauchy module $C_{i_1,\ldots,i_k}(f_i)$ is a polynomial of $\mathbb{K}[x_1, \ldots, x_{i_k}]$ and its leading term is $x_{i_j}^{d_i-k+1}$. Moreover, $C_{i_1,\ldots,i_k}(f_i)$ belongs to $\mathcal{M}$. In particular, if $d_i - k + 1 = d_{i_k}$ then*

$$\{f_1, \ldots, f_{i_k-1}, C_{i_1,\ldots,i_k}(f_i), f_{i_k+1}, \ldots, f_n\}$$

is a triangular basis of $\mathcal{M}$.

Cauchy, in [31, Extrait 108], already proved similar results (without the knowledge of Gröbner basis theory) when he studied the application of Ampère's ¨fonctions interpolaires¨ (what we call now Cauchy modules) for eliminating variables in symmetric funtions.

The second technique considers the natural action of G over the polynomials of $\mathcal{G}$ (permutations of the indexes of the variables) to find relations between these polynomials. These special permutations are named **transporters**.

Definition 2.5. *Let $[(i_1, k_1), \ldots, (i_s, k_s)]$ be an i-relation and $j \in [\![i+1, n]\!]$. A permutation $\sigma \in G$ is called an (i, j)-transporter if $\sigma(i) = j$ and $j = \max(\sigma(k) \mid k \in \{i_1, \ldots, i_s\})$.*

As for Cauchy technique, transporters can be used to produce, without any cost, polynomials of $\mathcal{G}$ from others taken in $\mathcal{G}$:

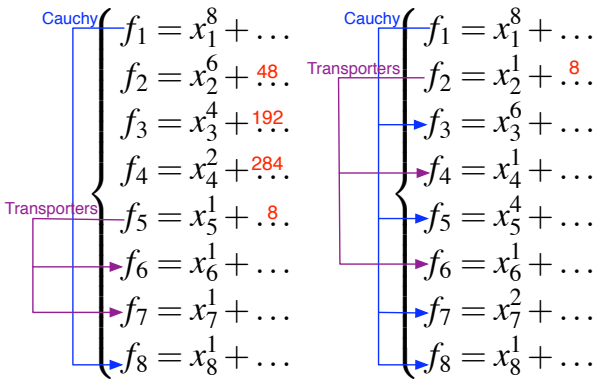
Proposition 2.6. *Let σ be an (i,j)-transporter and $g_i \in \mathbb{K}[x_{i_1},\ldots,x_{i_s}]$ the tail polynomial corresponding to f_i. If $d_i = d_j$ then $\{f_1,\ldots,f_{j-1},x_j^{d_j}+\sigma.g_i,f_{j+1},\ldots,f_n\}$ is a triangular basis of $\mathcal{M}$.*

All these techniques and the i-relations can be deduced only by inspecting the corresponding permutation group G. Thus, we can now define its corresponding computation scheme.

Definition 2.7. *The computation scheme of the permutation group G is defined by the following data:*
1. the degree d_i of the greatest variable in each polynomial in $\mathcal{G}$;
2. mathematical objects (shape) computed by Cauchy techniques and transportation ;
3. the minimal i-relation of each polynomial in $\mathcal{G}$ that can not be obtained by the preceding techniques.
The c-size of this computation scheme, denoted by $c(G)$, is defined by the number of monomials over all the i-monomials in 3.

The c-size $c(G)$ represents the total number of coefficients to compute in order to retrieve the triangular basis $\mathcal{G}$ by interpolation. The main drawback of the computation scheme is that it is dependent of the choice of the representative for the group G.

For example, let
$G_1 = \langle (8,7,6,1)(5,4,3,2),(8,1)(4,5),(5,1)\rangle$
and
$G_2 = \langle (2,1),(8,6,4,1)(7,5,3,2),(8,1)(7,2)\rangle$
be two conjugates in S_8 of the transitive permutation group $[2^4]S_4$.



The two corresponding computation schemes can be represented by the following drawings (G_1 on the left and G_2 on the right). On these drawings, the techniques are showed on the left side of the triangular bases and the integers, on right side, represents the sizes of the minimal i-relations. Thus, for G_1 we have $c(G_1) = 532$ and for G_2 we obtain only $c(G_2) = 8$.

Thus, in order to obtain the best efficiency by using computations schemes, we need a method which identifies the representative of a given group which provides a scheme with minimal c-size.

Computation of Computation Schemes

A brute force method is proposed in [123] for computing a conjugate of a permutation group with minimal computation scheme in the sense of the c-size. In this method, the number of candidates considered is equal to the index $|S_n : N_{S_n}(G)|$ which can be closed to $(n-1)!$ when G is moderate (in particular for the cyclic group of degree n, this index is equal to $\frac{(n-1)!}{\phi(n)}$). Thus the brute force method is completely useless in this case. In [120], we show how to easily construct a

computation scheme for particular types of permutation group. Also, we provide a data-structure which can be constructed for any type of permutation group and helps in finding such a minimal computation scheme. The concept of this data-structure is based on the correspondence between the sequence of pointwise stabilizer subgroups orbits of G and the sequence of factors of a polynomial f with Galois group G in the tower of subfields arising during the computation of its splitting field.

In fact, the two computations schemes techniques and minimal i-relations depend only on the structure of the different orbits of the pointwise stabilizers of G which correspond to the different factors arising during the computation of the splitting field by successive factorization.

Thus we do not need to inspect all the conjugates of a group but all the different possible sequences of orbits appearing during the process of stabilization of the group G. Then, from all these possibilities we well order the choice of the orbits to obtain the best conjugate of G in regards of its computation scheme. All these possibilities corresponds to a set of different classes of *non redundant bases* of G:

Definition 2.8. *Let G be a permutation group of degree n. A sequence $B = (b_1, \ldots, b_k)$ of different integers from $\{1, \ldots, n\}$ is called a* **regular sequence of length** *k. A regular sequence B is said to be* **non redundant** *with respect to G if*

$$G = G_B^{[1]} > G_B^{[2]} > \cdots > G_B^{[k+1]},$$

where, we denote by $G_B^{[i]}$ $(i \geqslant 2)$ the pointwise stabilizer $\mathrm{Stab}_G(\{b_1, \ldots, b_{i-1}\})$ in G. Moreover, for a non redundant regular sequence B, if $G_B^{[k+1]} = 1$, B is said to be a **non redundant base** *of G. The largest k such that there is a non redundant base of length k is called the* **depth** *of G.*

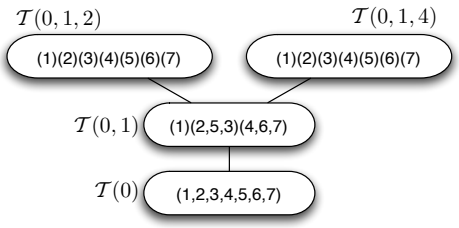
*Let $B_1 = (a_1, a_2, \ldots, a_l)$ and B_2 be two non redundant bases of G of the same length. We say that B_1 is G-***equivalent** *to B_2 if there exists $g \in G$ such that $B_2 = (g(a_1), g(a_2), \ldots, g(a_l))$*

The G-equivalence property is an equivalent relation over the set of non redundant bases. In a field theory point of view, two non redundant bases are G-equivalent iff the towers of fields defined by these bases are isomomorphic *level by level* from the ground field $\mathbb{Q}$ to the splitting field. From now on, when we will speak about classes of non redundant bases they will be always classes of G-equivalence.

A data structure attached to the permutation group G is introduced. It lets us store all the non redundant bases of G and the corresponding orbits of the natural action of its stabilizer along these different bases. From now on we say that an orbit is **non trivial** when it is not reduced to one element. It is important to recall that G is transitive since the polynomial is supposed to be irreducible.

Definition 2.9. *The* **orbit tree** *$\mathcal{T}$ of G is the recursive structure defined by*
1. The root $\mathcal{T}(0)$ of $\mathcal{T}$ is the orbit of G: $\{1, \ldots, n\}$;
2. any other node $\mathcal{T}(i_1 = 1, i_2, \ldots, i_s)$ is the set of orbits of $\mathrm{Stab}_G(\{0, i_1, \ldots, i_s\})$ where i_s is the minimal element of a non trivial orbit in the node $\mathcal{T}(i_1 = 1, i_2, \ldots, i_{s-1})$;
3. the construction is stopped as soon as the node contains only trivial orbits.
Let $\mathcal{T}(i_1, \ldots, i_s)$ be a node in $\mathcal{T}$. The **degree** *of this node is the integer defined by the index $|G : \mathrm{Stab}_G(\{0, i_1, \ldots, i_s\})|$.*

For example, if G be a copy in S_7 of the transitive group $F_{21}(7)$ generated by $\{(1,2,3,4,5,6,7), (1,2,4)(3,6,5)\}$. The drawing on the right corresponds to its orbits tree.



From the orbits tree of a given group G, we propose an algorithm in order to find the computation scheme with smallest c-size. It proceeds in the following way: For each branch in the orbit tree.

1. Maximize the use of Cauchy technique by promoting linear relations first.
2. Apply Transporter technique on the remaining polynomials
3. Compute minimal i-relations for each remaining polynomials

At the end of the process the minimal computation scheme is then deduced. By using this algorithm we obtain the following result.

Theorem 2.10. *For a given transitive permutation group G of degree n. If $|G| < 2^n$ then the computation scheme corresponding to G with minimal c-size can be computed with an algorithm of polynomial complexity in $|G|$.*

At least this computation does affect the total cost of the procedure for the splitting field computation. But this complexity does not well reflect the impact of this strategy in practice. The cost of finding the best computation scheme is negligible in comparison with the other steps of the splitting field computation. For almost all the groups G of degree at most 15 and $|G| < 10000$, the timings (done with MAGMA on an 32-bit 2.5GHz Intel processor) for the computation scheme are too small (in average < 1 second) to be really measured. Only few examples gave timings of at most 2 seconds. But, its impact on the global computation is consequent. In the following table, we show the timings (in seconds) for each step of the splitting field computation and we compare our implementation to the internal `Magma` function. In this table, $>$ (resp. $>>$) means a timing more than 600 (resp. 2000) seconds.

Group	$\|G\|$	Galois Grp.	Comp. Sch.	Interpolation	Magma
$7T_6$	2520	0.06	0.00	52.5	$>$
$8T_{32}$	96	0.16	0.00	0.72	33.5
$8T_{42}$	288	0.1	0.00	0.18	17.9
$8T_{47}$	1152	0.07	0.00	0.5	422.3
$9T_{25}$	324	0.42	0.10	4.07	106.1
$9T_{27}$	507	0.82	0.00	116.3	$>$
$9T_{31}$	1296	0.32	0.10	0.5	$>$
$9T_{31}$	1512	0.78	0.10	753.5	$>>$

As one can see, the part which takes most of the time is the interpolation step. Also, our approach using computation scheme is clearly better than the classical one used by `Magma`.

2.3 Radical representation of roots and applications

In the preceding section we describe a general method to represent the roots of a polynomial in an effective way. In some particular cases, roots of polynomials can be represented as radicals. The problem to identify exactly which polynomials verify this property can be seen as the starting point to Galois theory. Thanks to Abel and Galois, we know today that such polynomials are the ones with a solvable Galois group. In [93] we prove isomorphism between fields defined by two different C_5-generic polynomials. The core idea used in this work is the fact that the roots can be expressed as radicals, thus can be easily represented, even for a parametric polynomial. In [90] we use this fact to solve a problem related to the design of cryptographic protocols based on algebraic curves. We present a part of this work in the sequel.

Discrete Logarithm Problems based on algebraic curves on finite fields are of high interest in asymmetric cryptography. In particular they provide small size of keys needed to achieve good security. Nonetheless it is less easy to encode a message into an element of the group (in contrary to the case of DLP based on finite fields).

Let $\mathbb{F}_q$ be a finite field of odd characteristic p, and $H/\mathbb{F}_q : y^2 = f(x)$ where $\deg f = d$ be an elliptic (if $d = 3$ or 4) or hyperelliptic (if $d \geqslant 5$) curve, we consider the problem of computing points on H in deterministic polynomial time. In cryptographic applications, computing a point on a (hyper)elliptic curve is a prerequisite for encoding a message into its Jacobian group. Boneh-Franklin Identity-Based Encryption scheme [24] requires for instance to associate to any user identity a point on an elliptic curve. Having such a deterministic encoding is of high importance in the context of embedded implementation of cryptographic protocols. Non deterministic algorithms may leak information on secret keys.

For example, Atkin and Morain [5] remark that if x_0 is any element of $\mathbb{F}_q$ and $\lambda = f(x_0)$, then the point $(\lambda x_0, \lambda^{(d+1)/2})$ is on the curve $Y^2 = \lambda^d f(X/\lambda)$ (for odd integer d). But the latter can be either isomorphic to the curve or its quadratic twist, following that λ is a quadratic residue or not, and we have no way to control this in deterministic time.

In 2006, Shallue and Woestjine [136] proposed the first practical deterministic algorithm to encode points into an elliptic curve, quickly generalized by Ulas [139] to the family of hyperelliptic curves defined by $y^2 = x^n + ax + b$ or $y^2 = x^n + ax^2 + bx$. Icart [87] proposed in 2009 another deterministic encoding for elliptic curves, of complexity $O(\log^{2+o(1)} q)$, provided that the cubic root function, inverse of $x \mapsto x^3$ on $\mathbb{F}_q^*$, is a group automorphism. This encoding uses the well known radical expression of the roots of a cubic polynomial based on Cardano-Tartaglia's formulae. It parameterizes the points $(x : y : 1)$ on any elliptic curve $E : x^3 + ax + b = y^2$. Recall that any cubic polynomial has solvable Galois group and thus its roots can be represented as radicals and are easy parametrized as soon as the radicals are automorphisms.

In [90], we propose a general strategy based on polynomial with solvable Galois group in order to design encoding in (hyper)elliptic curves. Note that in parallel to this work and independently a different approach was proposed in [73]. Here we present only a part of [90], more exactly, we present how to design an encoding from the parametric (general) polynomial of De Moivre.

2.3.1 Rational and deterministic encoding from solvable polynomials

Let $f_{\underline{a}}(X)$ be a family of parameterized polynomials (where $\underline{a}$ denotes a k-tuple $(a_1, a_2, \ldots, a_k)$ of parameters) with solvable Galois group. We are interested in such parametric polynomials but also in the parametric radical expression of their roots $\chi_{\underline{a}}$. For instance $f_A(X) = X^2 + A$ in degree 2, or more interestingly $f_{A,B}(X) = X^3 + A\,X + B$ in degree 3, are such polynomials with simple radical formulae for their roots. The former verifies $\chi_A = \sqrt{-A}$ and a root of the second one is given by the well-known Cardano-Tartaglia's formulae (see [43]).

Given a parameterized family of solvable polynomials $f_{\underline{a}}(X)$, and a genus g, we now substitute a rational function $F_i(Y)$ in some variable Y for each parameter a_i in $\underline{a}$.

Let $\underline{F}(Y)$ denote the k-tuples of rational functions $(F_1(Y), F_2(Y), \ldots, F_k(Y))$. The equation $f_{\underline{F}(Y)}(X)$ now defines a plane algebraic curve C, with variables (X, Y). The genus of C increases when the degrees of $\underline{F}(Y)$ in Y increase. So if we target some fixed genus g for C, only few degrees for the numerators and denominators of $\underline{F}(Y)$ can occur. Since we can consider coefficients of these rational functions as parameters $\underline{b} = (b_1, \ldots, b_{k'})$, this yields a family of curves $C_{\underline{b}}$.

Less easily, it remains then to determine among these $\underline{F}(Y)$ the ones which yield roots $\chi_{\underline{F}(Y)}$ which can be computed in deterministic time. The easiest case is probably when no square root occurs in the computation of $\chi_{\underline{a}}$, since then any choice for $\underline{F}(Y)$ will work, at the expense of some constraint on the finite field. But this is usually not the case, and we might try instead to link these square roots to some algebraic parameterization of an auxiliary algebraic curve

In some case (typically hyperelliptic curves), it is useful to derive from the equation for $C_{\underline{b}}$ a minimal model (typically of the form $y^2 = g_{\underline{b}}(x)$). In order to still have a deterministic encoding with the minimal model, we need explicit birational maps $x = \Lambda_{\underline{b}}(X, Y)$, $y = \Omega_{\underline{b}}(X, Y)$ too. All in all, we obtain an encoding for a minimal model $g_{\underline{b}}$:

— Fix some Y as a (non-rational) function of some parameter t so that all the square roots appearing in the expression of $\chi_{\underline{F}(Y)}$ are well defined ;
— Compute $X = \chi_{\underline{F}(Y)}$;
— Compute $x = \Lambda_{\underline{b}}(X, Y)$ and $y = \Omega_{\underline{b}}(X, Y)$.

Once we will have found an encoding, it is important for cryptographic applications to study the cardinality of the subset of the curve that we parameterize. This ensures that we obtain convenient weak encodings for hashing into curves primitives (see [28]).

In the case of the De Moivre's family, we are able to deduce from the encoding formulae, a polynomial relation $P_{\underline{b}}(Y, t)$ between any Y of a point of the image and its preimages. Then the number of possible preimages is at most the t-degree of $P_{\underline{b}}(Y, t)$.

We also need to know *in advance* which values of $\mathbb{F}_q$ cannot be encoded using such functions, in order to deterministically handle such cases. In the De Moivre case, this set is small.

2.3.2 De Moivre's polynomials

This well-known family of quintic polynomials was first introduced by De Moivre in 1706 for the study of trigonometric equalities and its study from a Galoisian point of view was done by Borger in [25]. This definition can be easily generalized for any odd degree.

Definition 2.11 (De Moivre's polynomials)**.** *Let $\mathbb{K}$ be a field and d be an odd integer coprime with* $\operatorname{char} \mathbb{K}$*. The family of De Moivre's polynomials $p_{a,b}(x) \in \mathbb{K}[x]$ of degree d is defined for $a, b \in \mathbb{K}$ by*

$$p_{a,b}(x) = x^d + dax^{d-2} + 2da^2x^{d-4} + 3da^3x^{d-6} + \cdots + 2da^{(d-1)/2-1}x^3 + da^{(d-1)/2}x + b\,.$$

Examples. De Moivre's polynomials of degree 5 are $x^5 + 5ax^3 + 5a^2x + b$. De Moivre's polynomials of degree 13 are $x^{13} + 13ax^{11} + 26a^2x^9 + 39a^3x^7 + 39a^4x^5 + 26a^5x^3 + 13a^6x + b$.

Borger proved in [25] that De Moivre's polynomials of degree 5 are solvable by radical, the same is true for De Moivre's polynomials of any degree. Actually, by using a variable substitution $x = \gamma - a/\gamma$, then γ^d is a root of the polynomial $q_{a,b}(\theta)$ and we obtain the following result.

Lemma 2.12 (Resolution of De Moivre's polynomials)**.** *Let $p_{a,b}$ be a De Moivre's polynomial of degree d, let θ_0 and θ_1 be the roots of $q_{a,b}(\theta) = \theta^2 + b\theta - a^d$, then the roots of $p_{a,b}$ are*

$$(\omega_k \theta_0^{1/d} + \omega_k^{d-1}\theta_1^{1/d})_{0 \leqslant k < d}$$

where $(\omega_k)_{0 \leqslant k < d}$ are the d-th roots of unity.

Hence De Moivre's polynomials can be used to define a family of deterministically parameterized hyperelliptic curves for any genus. This encoding is described in the Algorithm 3.

Algorithm 3: DeMoivreEncode

input : A curve $H : p_{a,b}(x) - y^2 = 0$ and $t \in \mathbb{F}_q^* \setminus \mathcal{S}$.
output: A point $(x_t : y_t : 1)$ on H

if $a = 0$ **then**
  return $\left((t^2 - b)^{1/d \bmod q-1} : t : 1\right)$
$\delta := -(3a^d + b^2 + t^4)/6t - 2\,b^3/27 - a^d b/3 - t^6/27$; $A := \delta^{1/3 \bmod q-1} + t^2/3$;
$Y := tA - (3a^d + b^2 + t^4)/(6t)$;
$\alpha := 3a^d/(-3A + b)$;
$y_t := -3Y/(-3A + b)$; $x_t := \alpha^{1/d \bmod q-1} + (-a^d/\alpha)^{1/d \bmod q-1}$;
return $(x_t : y_t : 1)$

Figure 2.1 – Encoding on De Moivre's curves

And we obtain our main result on using De Moivre polynomial in cryptography.

Theorem 2.13. *Let $\mathbb{F}_q$ be the finite field with q elements. Suppose q odd and $q \equiv 2 \mod 3$ and d coprime with $q - 1$. Let $H_{a,b}/\mathbb{F}_q : y^2 = p_{a,b}(x)$ be the hyperelliptic curve where $p_{a,b}$ is a De Moivre polynomial defined over $\mathbb{F}_q$ with non-zero discriminant.*

Algorithm 3 computes a deterministic encoding $e_{a,b} : \mathbb{F}_q^* \setminus \mathcal{S} \to H_{a,b}$, where $\mathcal{S}$ is a subset of $\mathbb{F}_q$ of size at most 7, in time $O(\log^{2+o(1)} q)$.

Conversely, given a point on H we study how many elements in $\mathbb{F}_q$ yield this point.

Theorem 2.14. *Given a point $(x : y : 1) \in H_{a,b}(\mathbb{F}_q)$, we can compute the solutions s of the equation $e_{a,b}(s) = (x : y : 1)$ in time $O(\log^{2+o(1)} q)$. There are at most 8 solutions to this equation.*

The case of characteristic 2 is very similar. De Moivre's polynomials are solvable using the same auxiliary polynomial. A dimension 1 family of genus 2 curves is given by $p_{a,b}(x) = y + y^2$ which are also $p_{a,b+y+y^2}(x) = 0$ (see [90] for details).

CHAPTER 3

Contribution to Coppersmith's Algorithm

In this Chapter, we present some results on solving modular univariate equations by using Coppersmith's algorithm. Our papers corresponding to this subject are [38, 15, 6, 64, 66]. In the sequel, we use parts of [6, 15] to present our main result on the Coppersmith's algorithm and an application of it.

3.1 Introduction

The problem of finding integer roots of a Diophantine equation is known to be difficult in general. Its decisionial form (the tenth Hilbert's problem) is proven to be undecidable in general. Even the smaller interesting case, deciding if a two-variable quadratic has integer roots, is proven to be NP-complete by Adleman and Manders [108]. A straightforward application of solving Diophantine equation is related to the problem of factoring integers. For a given integer N, finding the non trivial solutions of the equation

$$xy - N = 0$$

provides a factorization of N. In particular, the security of the cryptosystem RSA, depends on the possibility to solve this equation. Fortunately no such an efficient resolution is known for the moment.

At EUROCRYPT '96, Coppersmith [35, 34, 37] showed how to find efficiently all small roots of polynomial equations (modulo an integer, or over the integers). The simplest (and perhaps most popular) result is the following: Given an integer N of unknown factorization and a monic polynomial $f(x) \in \mathbb{Z}[x]$ of degree δ, Coppersmith's lattice-based algorithm finds all integers $x_0 \in \mathbb{Z}$ such that $f(x_0) \equiv 0 (\mathrm{mod} N)$ and $|x_0| \leq N^{1/\delta}$ in time polynomial in $\log N$ and δ. This has many applications in public-key cryptanalysis (*e.g.* attacking special cases of RSA and factoring with a hint), but also in a few security proofs (such as in RSA-OAEP [138]). Accordingly, Coppersmith's seminal work has been followed up by dozens of articles (see May's survey [109] for references), which introduced new variants, generalizations, simplifications and applications. In section 3.3 we present such an application from [6] of Coppersmith's techniques which allows a gain in efficiency for a side channel attack on RSA.

All these small-root algorithms are based on the same idea of finding new polynomial equations using lattice basis reduction: it reduces the problem of finding small roots to finding LLL-short vectors in a lattice. This can theoretically be done in polynomial time using the LLL algorithm [104], but is by no means trivial in practice: the asymptotic running time is a high-degree polynomial, because the lattice is huge. More precisely, we provide in [15] a tight analysis of the Coppersmith's algorithm: asymptotically, one may take a matrix of dimension $O(\log N)$, and bit-size $O((\log^2 N)/\delta)$, resulting in a complexity upper bound $O((\log^9 N)/\delta^2)$ using the Nguyen Stehlé L^2 algorithm [115] as the reduction algorithm. In typical applications, δ is small ≤ 9 but $\log N$ is the bit-size of an RSA modulus, *i.e.* at least 1024 bits, which makes the theoretical running time daunting: $\log^9 N$ is already at least 2^{90}.

The bottleneck of all Coppersmith-type small-root algorithms is the LLL reduction. Despite considerable attention, no significant improvement on the running time has been found, except that LLL algorithms have improved since [37], with the appearance of L^2 [115] and $\tilde{L}^1$ [119]. And this issue is reflected in experiments (see [39]): in practice, one settles for sub-optimal parameters, which means that one can only find small roots up to a bound lower than the asymptotic bound. To illustrate this point, the celebrated Boneh-Durfee attack [22] on RSA with short secret exponent has the theoretical bound $d \leq N^{1-1/\sqrt{2}} \approx N^{0.292}$, but the largest d in the Boneh-Durfee experiments is only $d \approx N^{0.280}$ with a 1000-bit N, and much less for larger N, *e.g.* $d \approx N^{0.265}$ for 4000-bit N.

In section 3.4 we present two speedups from [15] over Coppersmith's algorithm 1, which can be combined in practice.

The first speedup is provable, if one uses L^2 [115], the total bit-complexity is upper bounded by $O(\log^7 N)$, which gives a speedup of $\Theta((\log^2 N)/\delta^2)$ which is quadratic in the bit-size of the small-root bound $N^{1/\delta}$. This speedup comes from combining LLL reduction with rounding: instead of LLL-reducing directly a matrix with huge entries, we suitably round the coefficients before LLL reduction to make them much smaller, and show that the LLL output allows to derive sufficiently short vectors in the original lattice. In order to use this rounding strategy, we show that matrices used in the Coppersmith's algorithm have a specific structure which allow such a method. More precisely, the matrices are triangular whose diagonal entries are reasonably balanced, which can be exploited.

Our second speedup is heuristic and applies whenever one wants to enlarge the root size X of Coppersmith's algorithm by exhaustive search: it is well-known that any root size X can be extended to mX by applying m times the algorithm on "shifted" polynomials. This enlargement is necessary when one wants to go beyond Coppersmith's bound $N^{1/\delta}$, but it is also useful to optimize the running time below $N^{1/\delta}$. In this setting, one applies Coppersmith's algorithm with the same modulus N but different shifted polynomials: $f_t(x) = f(X \cdot t + x)$ for varying t, where $0 \leqslant t < N^{1/\delta}/X$. We show that this creates hidden relationships between the matrices to be LLL reduced, which can be exploited in practice.

Rounding has been used in lattice reduction before, for instance, Buchmann [29] used rounding to rigorously estimate when a computation with real lattices can be alternatively performed using integer bases and the $\tilde{L}^1$ [119] algorithm is also based on rounding. Chaining has also been used in lattice reduction before, *e.g.* in the MIMO context [114]. However, it seems that none of the previous works identified the special structures of Coppersmith matrices which we exploit.

It is important to note that the complexity result we present for the rounding strategy on the Coppersmith matrix could be deduced from the analysis of the seminal algorithm by using results from [45] and by applying $\tilde{L}^1$ [119] for the lattice reduction. But, in the sequel we present an adaptation of the Coppersmith's algorithm that can be easily implemented in computer algebra system. Since, to the best of our knowledge, there is no public implementation of $\tilde{L}^1$, we focus on the use of L^2.

3.2 Coppersmith's method for finding small roots

Coppersmith in [35, 37] proposes an algorithm to find efficiently all small roots of univariate polynomial equation modulo an integer N of unknown factorization. Let f be a monic univariate polynomial of degree δ with coefficients in $\{0, \ldots, N-1\}$. In order to simplify the exposition, we assume in all the sequel that the degree remains small, in particular:

$$\delta + 1 < \log(N)/2\,.$$

The main theorem of Coppersmith is stated as follows.

Theorem 3.1 (Coppersmith [35, 37])**.** *There is an algorithm (Algorithm 4) which outputs all integers $x_0 \in \mathbb{Z}$ such that $f(x_0) \equiv 0 \ (mod\ N)$ and $|x_0| \leq N^{1/\delta}$ in time polynomial in $\log N$ and δ.*

It is also important to note that Coppersmith's algorithm (Algorithm 4) does not directly achieve the bound $N^{1/\delta}$: indeed, it finds efficiently all roots up to some bound X ($< N^{1/\delta}$) depending on an integer parameter $h \geq 2$, chosen asymptotically to be $h = O((\log N)/\delta)$. When h is sufficiently large, then X becomes sufficiently close to $N^{1/\delta}$ so that one can find all roots up to $N^{1/\delta}$. However, it is well-known that the bound $X = N^{1/\delta}$ should not be reached by taking such a large h. Instead, it is faster to use a smaller h, and perform exhaustive search on the most significant bits of the solutions, as depicted in Algorithm 4.

We now explain Coppersmith's algorithm (Algorithm 4). The core idea consists in reducing the problem to solving univariate polynomial equations over the integers, by transforming modular roots into integral roots. More precisely, it constructs a polynomial $g(x) \in \mathbb{Z}[x]$ such that: if $x_0 \in \mathbb{Z}$ is such that $f(x_0) \equiv 0 \pmod N$ and $|x_0| \leq X$, then $g(x_0) = 0$ and can be solved easily over $\mathbb{Z}$. To do so, it uses the following elementary criterion:

Lemma 3.1 (Howgrave-Graham [86])**.** *Let $g(x) \in \mathbb{Z}[x]$ be a polynomial with at most n non-zero coefficients. Let M be an integer ≥ 1. Assume that $\|g(xX)\| < \frac{M}{\sqrt{n}}$ for some $X \in \mathbb{R}$. If $x_0 \in \mathbb{Z}$ is such that $g(x_0) \equiv 0 \ (mod\ M)$ and $|x_0| \leq X$, then $g(x_0) = 0$.*

Lemma 3.1 will be used with $M = N^{h-1}$ and $g(x)$ found by lattice reduction. Let $h \geq 2$ be an integer and define the following family of $n = h\delta$ polynomials:

$$g_{i,j}(x) = (x)^j N^{h-1-i} f^i(x) \quad 0 \leq i < h, 0 \leq j < \delta \tag{3.2.1}$$

Algorithm 4: Coppersmith's Method

input : Two integers $N \geq 1$ and $h \geq 2$, a univariate degree-δ monic polynomial $f(x) \in \mathbb{Z}[x]$ with coefficients in $\{0, \ldots, N-1\}$ and $2 < \delta + 1 < (\log N)/2$.

output: All $x_0 \in \mathbb{Z}$ s.t. $|x_0| \leq N^{1/\delta}$ and $f(x_0) \equiv 0 \mod N$.

1: Let $n = h\delta$, X the bound given in (3.2.3), and $t = 0$.
2: **while** $Xt < N^{1/\delta}$ **do**
3: $f_t(x) = f(Xt + x) \in \mathbb{Z}[x]$.
4: Build the $n \times n$ lower-triangular matrix B whose rows are the $g_{i,j}(xX)$'s defined by (3.2.1).
5: Run the L^2 algorithm [115] on the matrix B.
6: The first vector of the reduced basis corresponds to a polynomial of the form $v(xX)$ for some $v(x) \in \mathbb{Z}[x]$.
7: Compute all the roots x_0' of the polynomial $v(x) \in \mathbb{Z}[x]$ over $\mathbb{Z}$.
8: Output $x_0 = Xt + x_0'$ for each root x_0' which satisfies $f_t(x_0') \equiv 0 \,(\mathrm{mod}\, N)$ and $|x_0'| \leq X$.
9: $t \leftarrow t + 1$.
10: **end while**

These n polynomials satisfy: if $f(x_0) \equiv 0 (\mathrm{mod} N)$ for some $x_0 \in \mathbb{Z}$, then $g_{i,j}(x_0) \equiv 0 (\mathrm{mod} N^{h-1})$. In order to apply Lemma 3.1 for a bound $X \geq 1$ to be determined later, Coppersmith's algorithm constructs the n-dimensional lattice L spanned by the rows of the $n \times n$ matrix B formed by the n coefficient vectors of $g_{i,j}(xX)$, where the polynomials are ordered by increasing degree (*e.g.* in the order $(i,j) = (0,0), (0,1), \cdots, (0,\delta-1), (1,0), \cdots (h-1, \delta-1)$) and the coefficients are ordered by increasing monomial degree: the first coefficient is thus the constant term of the polynomial. The matrix B is lower triangular, and its n diagonal entries are:

$$\left(N^{h-1}, N^{h-1}X, \ldots, N^{h-1}X^{\delta-1}, \ldots, N^0X^{\delta h-\delta}, \ldots, N^0X^{\delta h-2}, N^0X^{\delta h-1}\right), \tag{3.2.2}$$

because $f(x)$ is monic. In other words, the exponent of X increases by one at each row, while the exponent of N decreases by one every δ rows. It follows that $\mathrm{vol}(L) = \det(B) = N^{\frac{1}{2}n(h-1)}X^{\frac{1}{2}n(n-1)}$. Algorithm 4 applies the LLL algorithm to the matrix B, which provides a non-zero polynomial $v(x) \in \mathbb{Z}[x]$ such that $\|v(xX)\| \leq 2^{\frac{n-1}{4}}\mathrm{vol}(L)^{\frac{1}{n}} = 2^{\frac{n-1}{4}}N^{\frac{h-1}{2}}X^{\frac{n-1}{2}}$. It follows that the polynomial $v(x)$ satisfies Lemma 3.1 with $M = N^{h-1}$ and $g(x) = v(x)$ if

$$X \leq \frac{1}{\sqrt{2}}N^{\frac{h-1}{n-1}}(n+1)^{-\frac{1}{n-1}}. \tag{3.2.3}$$

The dimension of B is $n = h\delta$, and the entries of the matrix B have bit-size $O(h \log N)$, therefore the running time of L^2 in Step 5 without fast integer arithmetic is $O(\delta^6 h^7 \log N + \delta^5 h^7 \log^2 N)$, which is $O(\delta^5 h^7 \log^2 N)$ because $\delta + 1 < (\log N)/2$. We obtain the following concrete version of Theorem 3.1:

Corollary 3.2. *Algorithm 4 of Theorem 3.1 with* $h = \lfloor \log N/\delta \rfloor$ *and* $X = \lfloor 2^{-1/2}N^{\frac{h-1}{n-1}}(n+1)^{-\frac{1}{n-1}} \rfloor$ *runs in time* $O((\log^9 N)/\delta^2)$ *without fast integer arithmetic using* L^2 *in Step 5.*

In the next section, we present an application of Coppersmith's algorithm in the context of side channel attacks on RSA.

3.3 Application to combined attack on RSA-CRT

In embedded systems, most RSA implementations use the Chinese Remainder Theorem (CRT) which yields an expected speed-up factor of four [41] on signature or decryption. Following the CRT-RSA algorithm, the signature generation is composed of two exponentiations $S_p = m^{d_p} \bmod p$ and $S_q = m^{d_q} \bmod q$, where $d_p = d \bmod p - 1$ and $d_q = d \bmod q - 1$. The signature is then obtained by recombining S_p and S_q, which is usually done by using Garner's formula [75]:

$$S = CRT(S_p, S_q) = S_q + q(i_q(S_p - S_q) \bmod p) \ , \tag{3.3.1}$$

where $i_q = q^{-1} \bmod p$.

Several countermeasures have been developed to protect CRT-RSA embedded implementations against both *Side Channel Attacks* (SCA) and *Fault Injection* (FI). In the framework of this work, we consider an algorithm protected:

— against SCA by using message and exponent blinding as suggested in [143], a regular exponentiation algorithm such as the Square Always [32] and a mask refreshing method along the exponentiation such as the one presented in [50]. Moreover, the blinding is kept all along the CRT-recombination.

— against FI by verifying the signature using the public exponent e [21]. In addition, we also use the approach presented in [49] which mainly consists in checking the result of the verification twice to counteract double FI attacks.

Figure 3.1 depicts the main steps of such an implementation where the k_i's are random values (typically of 64 bits) generated at each execution of the algorithm and S'_p, S'_q and S' represent the blinded version of S_p, S_q and S respectively.

It is well known that injecting a fault during the signature computation $\bmod p$ (or $\bmod q$) uniquely leads to a faulty signature that allows the attacker to recover the private key. However the verification with the public exponent detects such a disturbance and the faulty signature is never revealed to the attacker. Moreover, at first glance, it seems impossible to perform an SCA during the signature process due to the blinding countermeasure. However by observing Figure 3.1, one may note that the faulty signature $\widetilde{S}$ remains blinded until the end of exponentiation with e modulo N. Therefore if we can express $\widetilde{S}^e \bmod N$ in terms of the message m and of the private key then we can perform an SCA on this value. This is the main idea of our *Combined Attack* (CA) as described in the following.

If a fault ε is induced in m such that the faulty message $\widetilde{m}$ is equal to $m + \varepsilon$ at the very beginning of the computation of S_p then the faulty signature $\widetilde{S}$ verifies

$$\widetilde{S}^e = m + \varepsilon q i_q \bmod N \ . \tag{3.3.2}$$

Thanks to this relation, we deduce a CA: Firstly, the attacker asks the embedded device to sign several messages m_i through a CRT-RSA implemented as in Figure 3.1. For each signature,

the computation of S_q is performed correctly and a constant additive error ε is injected on the message m_i at the beginning of each S_p computation. Then during each signature verification, the attacker monitors the corresponding side-channel leakage $\mathcal{L}_i$ which represents the manipulation of $\widetilde{S}_i^e \bmod N$.

From relation (3.3.2), we know that there exists a sensitive value σ satisfying the relation $\widetilde{S}_i^e \bmod N = m_i + \sigma$. Therefore, the attacker will perform a Correlation Power Attack to recover this sensitive value by computing the Pearson correlation between $m_i + \sigma$ and $\mathcal{L}_i$ for all the possible values of σ. It follows that σ will be equal either to $\varepsilon q i_q \bmod N$ or to $\varepsilon q i_q \bmod N - N$. W.l.o.g. one can assume that $\sigma = \varepsilon q i_q \bmod N$

It is not possible to perform a statistical attack targeting the full value of σ at once due to its large size (i.e. $\lceil \log_2(N) \rceil$ bits). However, one can attack each subpart of this value, for instance by attacking byte per byte starting with the least significant one in order to be able to propagate easily the carry. Moreover, the remaining part of σ can be retrieve by using Coppersmith's algorithm. Actually, suppose we are given the t least significant bits (LSB) of the secret $\sigma = \varepsilon q i_q \bmod N$ (obtained after the CA) then we have:

$$\varepsilon q i_q = 2^t x_0 + k \bmod N \ ,$$

where t and k are known values, and x_0 is the $\lceil \log_2(N) - t \rceil$-bit unknown integer that is to be recovered.

The unknown secret part x_0 is solution of the polynomial $P_\varepsilon(x)$:

$$P_\varepsilon(x) = x^2 + c\,(2^{t+1}k - 2^t\varepsilon)\,x + c\,(k^2 - k\varepsilon) \equiv 0 \bmod N$$

where $c = (2^{2t})^{-1} \bmod N$, k, t, N are known, and ε is the induced fault.

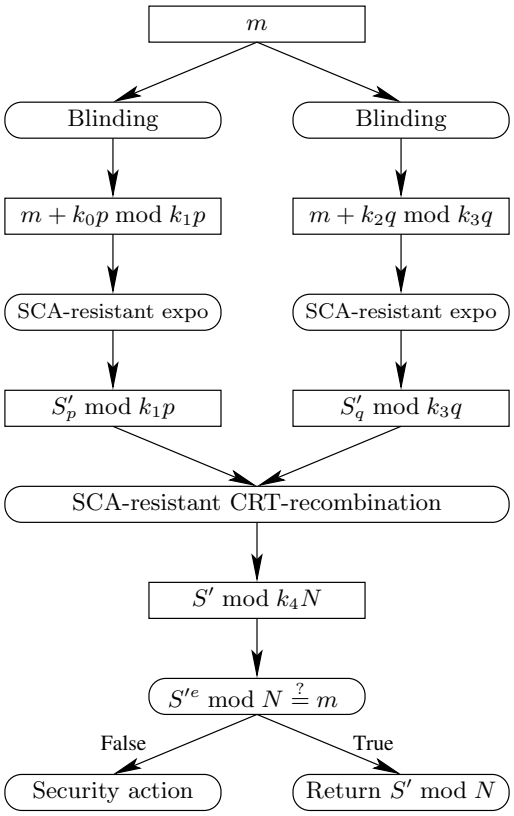


Figure 3.1 – CRT-RSA implementation secure against SCA and FI

Assume ε is known or sufficiently small to be exhaustively instanciated. Then by applying Theorem 3.1 we can compute x_0 in time polynomial in $\log_2(N)$ as soon as half (polynomial P_ε is of degree 2) of its bits are known. In practice, for 1024-bit RSA modules, the Coppersmith's algorithm takes only few seconds of time and allows to divide by 2 the total time of the attack (from 2 hours to 1 hour in total).

3.4 Rounding and chaining for Coppersmith's algorithm

The bottleneck of Coppersmith's algorithm is the LLL reduction of the matrix B, whose dimension is $n = h\delta$, and whose entries have bit-size $O(h \log N)$. Asymptotically, we have

$h = O(\log N/\delta)$ so the dimension is $O(\log N)$ and the bit-size is $O((\log^2 N)/\delta)$. We will modify Coppersmith's algorithm in such a way that we only need to LLL-reduce a matrix of the same dimension but with much smaller entries, namely bit-length $O(\log N)$.

To explain the intuition behind our method, let us first take a closer look at the matrix B and uncover some of its special properties:

Lemma 3.3. *Let $X \leq N^{1/\delta}$. The maximal diagonal coefficient of matrix B defined in Step. 4 of Algorithm 4 is $N^{h-1}X^{\delta-1} < N^h$, the minimal diagonal coefficient is $X^{h\delta-\delta} \leq N^{h-1}$, and $\frac{N^{h-1}X^{\delta-1}}{X^{h\delta-\delta}} \geq N^{1-1/\delta}$ if $h \geq 2$. Furthermore, if $X \geq \Omega(N^{\frac{h-1}{n-1}})$, $h \geq 2$ and $h\delta = O(\log N)$ then we have:*

$$X^{h\delta-\delta} \geq N^{h-O(1)}. \tag{3.4.1}$$

This implies that the diagonal coefficients of B are somewhat balanced: the matrix B is not far from being reduced. In fact, the first row of B has norm N^{h-1} which is extremely close to the bound $N^{h-1}/\sqrt{n}$ required by Lemma 3.1: intuitively, this means that it should not be too difficult to find a lattice vector shorter than $N^{h-1}/\sqrt{n}$.

To take advantage of the structure of B, we first size-reduce B to make sure that the subdiagonal coefficients are smaller than the diagonal coefficients. Then we round the entries of B so that the smallest diagonal coefficient becomes $\lfloor c \rfloor$ where $c > 1$ is a parameter. More precisely, we create a new $n \times n$ triangular matrix $\tilde{B} = (\tilde{b}_{i,j})$ defined by:

$$\tilde{B} = \left\lfloor cB/X^{h\delta-\delta} \right\rfloor \tag{3.4.2}$$

By Lemma 3.3, we have:

$$b_{i,i} \geq X^{h\delta-\delta} \quad \text{and} \quad \tilde{b}_{i,i} \geq \lfloor c \rfloor . \tag{3.4.3}$$

We LLL-reduce the rounded matrix $\tilde{B}$ instead of B: let $\tilde{\mathbf{v}} = \mathbf{x}\tilde{B}$ be the first vector of the reduced basis obtained. If we applied to B the unimodular transformation that LLL-reduces $\tilde{B}$, we may not even obtain an LLL-reduced basis in general. However, because of the special structure of B, it turns out that $\mathbf{v} = \mathbf{x}B$ is still a short non-zero vector of L. From this structure we obtain our first main result on speeding up Coppersmith's algorithm by taking $c = (3/2)^n$.

Theorem 3.2. *The rounding strategy in the Coppersmith's algorithm provides a new algorithm which outputs all integers $x_0 \in \mathbb{Z}$ such that $f(x_0) \equiv 0 \ (mod\ N)$ and $|x_0| \leq N^{1/\delta}$ in time $O(\log^7 N)$ without fast integer arithmetic using the L^2 algorithm.*

The chaining strategy is based on an hidden relation between the matrices reduced during the exhaustive search. More precisely, in the Coppersmith's algorithm in order to find all solutions which are close to the bound $N^{1/\delta}$, one should not use a very large lattice dimension (*i.e.* $n = O(\log N)$). Instead, it is better to use a lattice of reasonable dimension and to perform exhaustive search on the most significant bits of x until finding all solutions. Namely, we consider polynomials $f_t(x) = f(X \cdot t + x)$ where $0 \leqslant t < \frac{N^{1/\delta}}{X}$ and $X = \lfloor 2^{\frac{-1}{2}} N^{\frac{h-1}{n-1}} (n+1)^{-\frac{1}{n-1}} \rfloor$. Thus, an initial solution x_0 that can be written $x_0 = X \cdot t_0 + x_0'$ is obtained by finding the solution

x'_0 of the polynomial f_{t_0}. In this case, this solution satisfies $|x'_0| < X$ and it has a correct size for LLL to find it using a lattice of dimension n. For each polynomial f_t, one runs LLL on a certain matrix. We put in evidence a connection between the lattice used for the case $t = i$ and the next lattice used for $t = i + 1$. This connection is based on the well-known *Pascal matrix* $P = (p_{s,t})$ defined as the $n \times n$ lower-triangular matrix whose non-zero coefficients are the binomials: $p_{s,t} = \binom{s}{t}$ for $0 \leq t \leq s \leq n - 1$. More precisely, if B_i^R denotes the LLL-reduced matrix used for solving f_t for $t = i$ and P the Pascal matrix. The matrix

$$B_{i+1} = B_i^R \cdot P$$

spans the same lattice used for solving the case $t = i + 1$. This matrix consists of vectors $\mathbf{b_{i+1},}_j$ whose norms are close to vector norms of the LLL-reduced matrix B_i^R. Namely, for all $1 \leqslant j \leqslant n$ we have:

$$||\mathbf{b_{i+1},}_j|| < \sqrt{n} \cdot 2^{n-1} \cdot ||\mathbf{b_i^R}_{,j}|| \,.$$

In particular, for the case $i = t_0$ the first vector of B_{i+1} has a norm bounded by $2^{n-1} \cdot N^{h-1}$.

This result shows us that vectors of B_{i+1} are relatively close to the ones in the LLL-reduced matrix B_i^R. Thus, we intuitively expect the LLL-reduction of B_{i+1} to be less costly than the one of the original Coppersmith's matrix. However, our bounds are too weak to rigorously prove this. Yet, one can use this property iteratively to elaborate a new method which *chains* all LLL reductions as follows. First, one LLL-reduces B_0 for the case $t = 0$. This gives a reduced matrix B_0^R. Then, one iterates this process by performing LLL reduction on $B_{i+1} = B_i^R \cdot P$ (for $i \geq 0$) to obtain B_{i+1}^R and so forth until all solutions are found (each time by solving the polynomial corresponding to the first vector of B_i^R).

These two techniques *rounding* and *chaining* can be combined to give a new version of the Coppersmith's algorithm. Unfortunately, the complexity analysis can not be prove for the moment (due to the lack of tight approximation of the vectors computed during the chaining process). Even under some assumptions on these vectors, the asymptotic complexity analysis does not provide a better than the one of Theorem 3.2. But the impact in practice in huge, as one can see on the following table. Here we compare different possibilities for the lattice reduction

Upper bound for $\mathbf{x_0}$	2^{492}	2^{496}	2^{500}	2^{503}	$\mathbf{2^{504}}$	2^{505}	...	2^{512}
Lattice Dimension	29	35	51	71	**77**	87	...	NA
Original LLL (sec.)	10.6	35.2	355	2338	**4432**	11426	...	NA
Rounding LLL (sec.)	1.6	3.5	18.8	94	**150**	436	...	NA
Rounding + Chaining (sec.)	0.04	0.12	1.4	9.9	**15.1**	46.5	...	NA

during Coppersmith's algorithm. The RSA modulus is a 1024-bit integer and we consider a degree 2 polynomial. The most interesting column corresponds to dimension 77 lattice. Actually, this is the dimension where it is more efficient to begin the exhaustive search than continue to reduce larger lattices. In this case, our techniques allow a speed-up of factor almost 700 over the standard approach of Coppersmith's algorithm.

CHAPTER 4

Contribution to Polynomial System Solving

In this chapter some results on the resolution of polynomial systems coming from the modeling of cryptography problems are presented. Our papers related to this subject are [62, 65, 61, 68, 30]. In the sequel, we present results related to elliptic curve cryptography from [62, 65].

4.1 Introduction

The security of these cryptosystems is based on the difficulty to solve the *elliptic curves discrete logarithm problem (ECDLP)*: let E be an elliptic curve defined over a finite field $\mathbb{K}$. The set of its rational points forms a commutative group, $E(\mathbb{K})$. Given two points P and Q of $E(\mathbb{K})$, the ECDLP is to find, if it exists, an integer x such that $Q = [x]P$. The notation $[x]P$ denotes, the sum of P with itself x times.

In 2004, Gaudry [78], based on a seminal work by Semaev [135], proposes an index calculus attack to solve the ECDLP defined over a non prime finite field $\mathbb{K} = \mathbb{F}_{q^n}$ where $n > 1$. Recall that in this case the order of $E(\mathbb{K})$ is $\approx q^n$ and thus generic algorithm for solving the ECDLP have a complexity bounded by $O(q^{\frac{n}{2}})$. Diem, independently, follows the same approach in [48, 47] and provides a more general framework for this attack. For large genus curves, such an approach provides subexponential algorithms (e.g. [1, 77, 79, 52]), but in the case of elliptic curves, these results provide algorithms of exponential complexity which can, for certain parameters, be better than the generic Pollard's rho.

The attack based on the index calculus proceeds by sieving and linear algebra on a well chosen factor base (recall that the base field is not prime $\mathbb{K} = \mathbb{F}_{q^n}$):

1. **Factor base identification**: $\mathcal{F} = \{(x, y) \in E(\mathbb{F}_{q^n}) \mid x \in \mathbb{F}_q\}$.
2. **Sieving** Find $\#\mathcal{F} + 1$ relations of the form: $[a_j]P \oplus [b_j]Q = P_1 \oplus \cdots \oplus P_n$, where $P_1, \cdots, P_n \in \mathcal{F}$ and a_j and b_j are randomly picked up in $\mathbb{Z}$.
3. **Linear resolution**: Find λ_j such that $\bigoplus_j [\lambda_j a_j]P \oplus [\lambda_j b_j]Q = 0$ and deduce x.

For a fixed extension degree n, Gaudry proves that the complexity of this approach is $O(q^{2-\frac{2}{n}})$. It is thus faster than Pollard rho method in $O(q^{\frac{n}{2}})$ for $n \geq 3$. However, this complexity hides an exponential dependency in n in step 2 due to the resolution of polynomial systems. In particular, if one finds a way to reduce its complexity to subexponential, the index calculus will become of

same complexity. Thus studying this algebraic resolution is important for the evaluation of the ECDLP security. In the sequel, we focus of this study, we show how to gain a factor exponential in n when some symmetries are present in $E(\mathbb{K})$. Fortunately for the security of the ECDLP, this gain is not sufficient to reduce the complexity to subexponential.

To solve a system of degree at most d polynomials of n variables and defined over a finite field, through past experience, it is straightforward to choose the Gröbner based algorithm 2. As explained in chapter 1, in order to well control the complexity, this resolution requires two steps. First, by using efficient algorithms to compute Gröbner basis such as F_4 [56] or F_5 [57], a DRL Gröbner basis of the system is computed. Then, by using a change of ordering algorithm such as FGLM [67, 63], a LEX Gröbner basis is computed, from which one can read off the solutions of the system. The complexities (arithmetic in $\mathbb{F}_q$) of these two steps are $O(ne^{\omega n}d^{\omega n})$ (F_4, F_5) and $O(nD^3)$ (FGLM) respectively, where $2 \leq \omega < 3$ is the linear algebra constant and D is the number of solutions of this system. When the Bézout bound is reached i.e. $D = d^n$, which is generically the case and in particular in this context, it is clear that the change of ordering step dominates. In order to gain in efficiency we propose in [61, 60] a new algorithm of which has, under some assumptions, a complexity bounded by $O(nD^\omega)$. Thus, the step 1 is now the dominant one.

As said above, the direct relation between polynomial system solving and the ECDLP is in the step 2 of the index calculus attack. Natural symmetries appear in the definition of the problem to solve (e.g. different indexations of the sum does not change the result). This particular structure is already considered in the works of Semaev, Gaudry and Diem. In [62] we study the case where the underlying curve has, by choice, some particular intrinsic symmetries. These cases appear naturally when such curves are chosen in order to increase the efficiency of the arithmetic of their group of rational points (e.g. [14, 51, 13]). In the sequel, we show how to exploit such a structure to gain an exponential factor in the complexity of the polynomial system resolution (see theorems 4.3 and 4.4) and thus in the total complexity for solving the ECDLP. Although the general complexity is still exponential, this gain allows to perform computations which were not possible before.

4.2 Point decomposition problem and polynomial system solving

Until section 4.5, we focus on the case where the field $\mathbb{F}_{q^n}$ has a large characteristic q.

4.2.1 Problematic

The step 2 of the index calculus algorithm is related to solving many instances of the *Point Decomposition Problem*, defined as follows.

Problem. ***PDP***$(R, \mathcal{F})$*: Given a point* R *on an elliptic curve* $E(\mathbb{F}_{q^n})$ *and a factor base* $\mathcal{F} \subset$

$E(\mathbb{F}_{q^n})$, find, if they exist, $P_1, \ldots, P_n$ in $\mathcal{F}$ (the factor base), such that

$$R = P_1 \oplus \cdots \oplus P_n \,.$$

To model this problem, one can use the algebraic representation of the group law on the curve to obtain a polynomial system $\mathcal{T} = \{g_1, \ldots, g_s\}$:

$$\mathcal{T}: \begin{cases} (x_i, y_i) \in E \\ (x_1, y_1) \oplus (x_2, y_2) \oplus \cdots \oplus (x_n, y_n) = (R_x, R_y) \end{cases}$$

From the definition of the factor base $\mathcal{F} = \{(x, y) \in E(\mathbb{F}_{q^n}) \mid x \in \mathbb{F}_q\}$, the solutions (x_i, y_i) have to be found in $\mathbb{F}_q \times \mathbb{F}_{q^n}$. To avoid the impact of the size of the y_i coordinates, Semaev in [135], proposes to project this system on the x_i coordinates. In this way, he obtains a multivariate polynomial $f_{n+1}(x_1, \ldots, x_n)$, called *Summation Polynomial* which characterizes the solutions of **PDP**$(R, \mathcal{F})$. By applying a scalar restriction on this polynomial (we consider $\mathbb{F}_{q^n}$ as an extension $\mathbb{F}_q(\omega)$ of degree n over $\mathbb{F}_q$):

$$f_{n+1}(x_1, \ldots, x_n, R_x) = 0 = \sum_{i=0}^{n-1} \psi_i(x_1, \ldots, x_n) \cdot \omega^i \,.$$

We thus obtain a polynomial system

$$\mathcal{S}: \ \{\psi_0, \ldots, \psi_{n-1}\} \subset \mathbb{F}_q[x_1, \ldots, x_n]\}$$

of n equations with n variables and of maximal degree $2^{n(n-1)}$. Solving this system in $\mathbb{F}_q$ thus provides solutions of the **PDP**$(R, \mathcal{F})$. This system can be seen as the starting point of our work and we reduce it using internal symmetries as we present in sequel.

4.2.2 On using symmetries

The system $\mathcal{S}$ is also considered by Gaudry in [78], in particular, he notes that it can be rewritten in terms of the elementary symmetric polynomials $e_1, \ldots, e_n$ in $x_1, \ldots, x_n$. Actually, the symmetric group acts naturally in the definition of the PDP: the ordering of the P_i's does not change their sum. Thus Semaev proves that the full symmetric group acts naturally on the summation polynomial. Hence the polynomial system $\mathcal{S}$ corresponding to the PDP. Such a use of the full symmetric group is a classical way to reduce the number of solutions by a factor $n!$. After a change of variables the PDP becomes modeled by:

$$f_{n+1}(e_1, \ldots, e_n, R_x) = 0_E = \sum_{i=0}^{n-1} \varphi_i(e_1, \ldots, e_n) \cdot \omega^i$$

and for the polynomial system model:

$$\mathcal{S}_{\mathfrak{S}_n}: \ \{\varphi_0, \ldots, \varphi_{n-1}\} \subset \mathbb{F}_q[e_1, \ldots, e_n]\}$$

The system $\mathcal{S}_{\mathfrak{S}_n}$ has n equations with n variables and are of maximal degree $2^{(n-1)}$ (remark the reduction in comparison with the system $\mathcal{S}$.

In the case of the Pollard rho method, it is well-known that if there is a small rational subgroup in the support of the DLP, the Pohlig-Hellman reduction allows to speeds-up the computation by a factor of roughly the square root of the order of this subgroup. It is also the case if there is an explicit automorphism of small order. For index calculus in general, it is far less easy to make use of such an additional structure. See for instance the article by Couveignes and Lercier [40], where, in the context of DLP over finite fields, a factor base is chosen especially to fit this need.

On the other side, many particular families of curves with small torsion subgroups has been proposed for their arithmetical efficiency. This is in particular the case for the Twisted Edwards, twisted Jacobi intersection curves and universal Edwards model of elliptic curves which all include a rational 2-torsion point. From the preceding remarks, if one wants to use this subgroup to help to solve the ECDLP by index calculus attack, one can try to translate it as an action over the solutions of the PDP.

Suppose that we have a solution $(P_1, P_2, \ldots, P_n)$ to the PDP, and denote by T_2 a 2-torsion point. Thus for all $k \in \{1, \ldots, \lfloor n/2 \rfloor\}$ we have $P_1 \oplus \ldots \oplus P_n \oplus [2k]T_2 = R$. Therefore, from one decomposition of R one can deduce 2^{n-1} decompositions of R by adding an even number of times a 2-torsion point :

$$\begin{aligned} R &= P_1 \oplus \cdots \oplus P_n \\ &= (P_1 \oplus T_2) \oplus (P_2 \oplus T_2) \oplus P_3 \oplus \cdots \oplus P_n \\ &= (P_1 \oplus T_2) \oplus P_2 \oplus (P_3 \oplus T_2) \oplus P_4 \oplus \cdots \oplus P_n \\ &\vdots \end{aligned}$$

In general, these decompositions do not correspond to solutions of the PDP, since $(P_i + T_2)$ is not always in the factor base $\mathcal{F}$. But in the case of Twisted Edwards, twisted Jacobi intersection curves and universal Edwards model of elliptic curves, we prove that it is possible to modify the definition of the curve and the factor base so that the action of a 2-torsion leaves invariant the set of solutions of the PDP. Actually, the action of the 2-torsion point T_2 in this context is very simple:

$$(x, y) \oplus T_2 = (-x, -y)\,.$$

Thus, this lets us hope to deduce an action on the polynomial system and a gain in efficiency for its resolution.

4.2.3 From geometric symmetries to algebraic structures

In order to apply a change of variables as in the case of the symmetric group, we need more than just an action on the solutions of the problem we want to solve. One needs to translate

the geometric action as an algebraic one. The 2-torsion action on the solutions of the PDP combined with the full symmetric group can be seen as the group which leave invariant an n-demihypercube. Thanks to the work of Weyl and Coxeter, such an action can be represented as a linear group D_n which is isomorphic to $(\mathbb{Z}/2\mathbb{Z})^{n-1} \rtimes \mathfrak{S}_n$ (see [91]). This linear group acts naturally on a polynomial ring in n variable by permuting the variables and changing an even number of signs of them.

To proceed to a change of variables, we now need an algebraic framework. In the case of the full symmetric group, the change of variables is known since Newton. The study of the general case starts with the work of Hilbert. In particular, the problem to know if the ring of invariant polynomials under the action of a group can be represented as a polynomial ring of invariant was solved more recently [137]. We have the following central result of invariant theory:

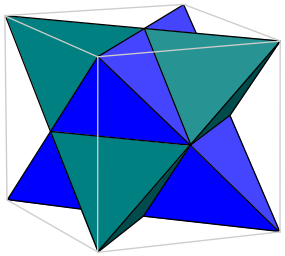


Figure 4.1 – A 3-demihypercube is a tetrahedron

Theorem 4.1 (Shephard, Todd)**.** *Let G be a finite linear group such that its order is not a multiple of char$(\mathbb{K})$. There exist algebraically independent invariants $\theta_1, \ldots, \theta_n$ such that the invariant polynomial ring $\mathbb{K}[x_1, \ldots, x_n]^G$ is equal to the algebra $\mathbb{K}[\theta_1, \ldots, \theta_n]$ if and only if G is a pseudo-reflection group.*

The θ_i's defined in Theorem 4.1 are called *primary invariant*. A direct consequence of Theorem 4.1 is the existence of an isomorphism Ω_G between $\mathbb{K}[x_1, \ldots, x_n]^G$ and $\mathbb{K}[y_1, \ldots, y_n]$ where $y_1, \ldots, y_n$ are new variables.

Definition 4.1. *Let G be a pseudo-reflective group and $\theta_1, \ldots, \theta_n \in \mathbb{K}[x_1, \ldots, x_n]^G$ be the primary invariants of G. We denote by Ω_G the ring isomorphism from $\mathbb{K}[x_1, \ldots, x_n]^G$ to $\mathbb{K}[y_1, \ldots, y_n]$ corresponding to the change of coordinates by the θ_i's and defined by*

$$\begin{array}{rcl} \Omega_G^{-1} : \mathbb{K}[y_1, \ldots, y_n] & \longrightarrow & \mathbb{K}[x_1, \ldots, x_n]^G \\ f & \longmapsto & f(\theta_1, \ldots, \theta_n)\,. \end{array}$$

Since the group D_n is a reflection group (it leaves invariant a polytope) Theorem 4.1 can be applied in our case. In particular, it is possible to apply a change of variables with the following invariants which are known to be algebraically independent (see [91]):

$$s_i = \sum_{1 \le j_1 < \ldots < j_i \le n} \prod_{k=1}^{i} x_{j_k}^2 \quad i = 1, \ldots, n-1$$

$$s_n = e_n = \prod_{k=1}^{n} x_k$$

These polynomials are thus *primary invariants* for D_n.

We prove that (modulo some adaptations of the models), the summation polynomials corresponding to twisted Edwards, twisted Jacobi intersection curves and universal Edwards model of elliptic curves are all invariant under the action of the linear group representing D_n. Since this group action on an n-variate polynomials corresponds to any permutation of the variables (the $\mathfrak{S}_n$ part) and an even number of changes of sign (the $(\mathbb{Z}/2\mathbb{Z})^{n-1}$), one can check that the polynomials s_i are well invariant.

From the invariance of the summation polynomial, we deduce, as in the case of the symmetric group, a polynomial system modeling the PDP where each polynomial is invariant under D_n. Then one can apply a change of variables Ω_{D_n} as given in Theorem 4.1:

$$\mathcal{S}_{D_n} = \{\varphi_1, \ldots, \varphi_n\} \subset \mathbb{F}_q[s_1, \ldots, s_{n-1}, e_n]$$

This new system seems *smaller* than the one obtained after the change of variables using the elementary symmetric polynomials only. Actually, this fact can be read from the degree of the polynomials. In order to compare this quantity well, we need to define some graduation. Such graduation will be used to evaluate the complexity of the resolution using these symmetries.

4.2.4 Size estimation

We recall that a graduation $\deg_w$ on the monomials of $\mathbb{K}[x_1, \ldots, x_n]$ is defined from a given sequence of weights $w = (w_1, \ldots, w_n)$ in the following way:

$$\deg_w(x_1^{\alpha_1} \cdots x_n^{\alpha_n}) = \sum_{i=1}^{n} w_i \alpha_i.$$

It is worth noticing that the usual degree corresponds to $\deg_w$ with weights $(1, \ldots, 1)$. In order to keep the standard notation, we use deg in this case and call weighted degree for any other graduation (i.e when $w \neq (1, \ldots, 1)$). In this general context, a polynomial is said to be *homogeneous* if all its monomials have the same graduation. It is important to note that the homogeneity of a polynomial depends on the graduation.

Since the group D_n contains the full symmetric group, the primary invariants $s_1, \ldots, s_{n-1}, e_n \in \mathbb{K}[x_1, \ldots, x_n]$ of D_n are thus invariant under the action of $\mathfrak{S}_n$. In particular, these polynomials can be rewritten in terms of the elementary symmetric functions. Let ρ_i denotes an expression of s_i in $\mathbb{K}[e_1, \ldots, e_n]$ one can deduce:

$$\begin{cases} \rho_i = e_i^2 + 2\sum_{j=1}^{i-1}(-1)^j e_{i-j}e_{i+j} + 2(-1)^i e_{2i} & \text{if } \; i \leq \lfloor n/2 \rfloor \\ \rho_i = e_i^2 + 2\sum_{j=1}^{n-i}(-1)^j e_{i-j}e_{i+j} & \text{if } \; \lfloor n/2 \rfloor < i < n \\ \rho_n = e_n \end{cases} .$$

From this representation of the primary invariants of D_n, we deduce a weighted degree which preserves the grading between the two rings $\mathbb{K}[e_1, \ldots, e_n]$ and $\mathbb{K}[s_1, \ldots, s_{n-1}, e_n]$, more exactly we have:

Lemma 4.2. *For all* $f \in \mathbb{K}[x_1, \ldots, x_n]^{D_n} \subset \mathbb{K}[x_1, \ldots, x_n]^{\mathfrak{S}_n}$*, if* $\mathbb{K}[s_1, \ldots, s_{n-1}, e_n]$ *is equipped with the graduation* $\deg_w$ *with weights* $w = (2, \ldots, 2, 1)$ *then*

$$\deg_w (\Omega_{D_n}(f)) = \deg (\Omega_{\mathfrak{S}_n}(f)) \ .$$

Thanks to this Lemma, one can see that the system $\mathcal{S}_{D_n}$ is smaller than $\mathcal{S}_{\mathfrak{S}_n}$ in terms of degree since we need some weights on the variables in order to obtain the sane graduation.

Now the question is to know how to use the structures of these algebraic systems in order to gain in efficiency during their resolution.

4.3 Solving polynomial systems with symmetries

As recalled in the chapter 1, solving a 0-dimensional polynomial proceeds in two steps. First a Gröbner basis for the degree reverse lexicographical ordering (DRL) is computed. Then, from this basis, a lexicographical Gröbner basis is computed by using a change of ordering algorithm [55, 63, 59].

For the the first step, we consider the algoritms F_4 or F_5 [56, 57], their complexity depends on the form of the polynomials in the system given as input. In particular, there are well estimate when the system is *homogeneous and regular*.

Definition 4.3 (Regular systems)**.** *Let* $F = (f_1, \ldots, f_s) \in (\mathbb{K}[x_1, \ldots, x_n])^s$ *be a sequence of* $s \leq n$ *non-zero homogeneous polynomials for a fixed graduation* $\deg_w$*. The sequence* F *is said to be* regular *if for all* $i \in \{1, \ldots, s-1\}$*, the polynomial* f_{i+1} *is not a zero divisor in the quotient ring* $\mathbb{K}[x_1, \ldots, x_n]/\langle f_1, \ldots, f_i \rangle$*. A homogeneous polynomial system* $\{f_1, \ldots, f_s\}$ *is said to be regular if the sequence* $(f_1, \ldots, f_s)$ *is regular.*

Here we consider only system with the same number of variables as of equations ($s = n$). When such a system is regular then the ideal that it generates is zero-dimensional. For homogeneous regular systems, the complexity of computing a graded reverse lexicographical Gröbner basis can be bounded by the complexity of computing the reduced row echelon form of a particular matrix: the Macaulay matrix.

Definition 4.4 (Macaulay matrix)**.** *Let* $\{f_1, \ldots, f_n\}$ *be a set of homogeneous polynomials of* $\mathbb{K}[x_1, \ldots, x_n]$ *and* $>$ *be a graded monomial ordering for a fixed graduation* $\deg_w$*. The Macaulay matrix in graduation* d*, denoted Mac*(d)*, is the matrix whose rows contain the coefficients of the polynomials* tf_j *for* $j = 1, \ldots, n$ *and all monomials* t *of* $\mathbb{K}[x_1, \ldots, x_n]$ *such that* $\deg_w(tf_j) = d$*. Each column of the matrix corresponds to a monomial of* $\mathbb{K}[x_1, \ldots, x_n]$ *of graduation* d*. The columns are arranged in descending order w.r.t. the monomial ordering* $>$*.*

The size of this matrix depends on a certain graduation d_{reg} (see [8]) called the *degree of regularity* of the system.

Definition 4.5 (Degree of regularity)**.** *Let $\mathcal{I}$ be a zero dimensional ideal in the polynomial ring $\mathbb{K}[x_1, \ldots, x_n]$ equipped with a graded monomial ordering for a fixed graduation $\deg_w$. We assume that the ideal $\mathcal{I}$ is generated by a sequence of homogeneous polynomials $(f_1, \ldots, f_n)$. Let LT($\mathcal{I}$) be the leading term ideal of $\mathcal{I}$, also called initial ideal, which is the ideal of $\mathbb{K}[x_1, \ldots, x_n]$ generated by the leading terms LT(f) of the elements f in $\mathcal{I}$. The degree of regularity of $\mathcal{I}$, denoted d_{reg}, is defined as the minimal graduation d such that the set $M(d)$ of monomials $m \in \mathbb{K}[x_1, \ldots, x_n]$ of graduation $\deg_w(m)$ greater or equal to d verifies*

$$M(d) \subset \mathit{LT}(\mathcal{I}) .$$

For regular systems, the Macaulay bound gives a bound on d_{reg} when the graduation is the usual degree (see [99]). A generalization of this result expressed in terms of a weighted degree is given in [58]. This result is of high importance in our work since it allows to well explain the gain in efficiency by using the intrinsic symmetries of the PDP.

Theorem 4.2 ([99][58])**.** *Let $F = (f_1, \ldots, f_n)$ be a regular sequence of non-zero homogeneous polynomials of $\mathbb{K}[x_1, \ldots, x_n]$ equipped with a graded monomial ordering for a fixed graduation $\deg_w$. By denoting d_i the graduation $\deg_w(f_i)$ we have the following bound*

$$d_{reg} \leq \max_{i=1,\ldots,n} \{w_i\} + \sum_{i=1}^{n} (d_i - w_i) .$$

The size of the Macaulay matrix in graduation d, is then deduce from the number of monomials in n variables of graduation d (see [58] for more details on these results). Hence, for homogeneous regular systems, a bound on the arithmetic complexity of F_4 or F_5 algorithms for computing a DRL Gröbner basis with fixed weights $(w_1, \ldots, w_n)$ can be deduced. In particular, when one of the w_i is equal to one, we have the following bound:

$$O\left(\frac{n}{\Delta^\omega}\binom{d_{\text{reg}} + S_n}{n}^\omega\right) \tag{4.3.1}$$

where $\Delta = \prod_{i=1}^{n} w_i$, S_n is defined by $S_1 = 0$ and $S_i = S_{i-1} + w_i \frac{\text{Gcd}_{j=1,\ldots,i-1}\{w_j\}}{\text{Gcd}_{j=1,\ldots,i}\{w_j\}}$ for $i \geq 2$ and $2 \leq \omega < 3$ is the linear algebra constant.

Since one of the w_i's is equal to one, their indexations can be chosen so that $S_n < \sum_{i=1}^{n} w_i$. Let d be a bound on the degree of the f_i, from Theorem 4.2, we have $d_{reg} \leq nd + w_{max} - \sum_{i=1}^{n} w_i$ with w_{max} the maximum over the weights w_i. The bound 4.3.1 then verifies

$$O\left(\frac{n}{\Delta^\omega}\binom{d_{\text{reg}} + S_n}{n}^\omega\right) = O\left(\frac{n}{\Delta^\omega}\binom{nd + w_{max}}{n}^\omega\right) \tag{4.3.2}$$

Hence, since w_{max} is fixed, when $d \to \infty$ and $n \to \infty$, using Stirling approximation, this asymptotic complexity can be bounded by

$$O\left(\frac{nd^{n\omega}e^{n\omega}}{\Delta^\omega}\right) \tag{4.3.3}$$

This complexity bound for the computation of the DRL Gröbner basis will be used in the sequel for the estimation of the gain obtained thanks to the use of the action of the 2-torsion point.

The problem here, as in many other applications, the polynomial systems modeling the PDP are not homogeneous. By consequence one needs to relate the complexity of solving an affine polynomial system to the complexity of solving a particular homogeneous system. For this purpose, it is usual to consider the *homogeneous component of highest graduation* as specified in the next definition.

Definition 4.6 (Affine regular systems)**.** *Let $F = (f_1, \ldots, f_n)$ be a sequence of non-zero affine polynomials of $\mathbb{K}[x_1, \ldots, x_n]$. We denote by $f_i^{(h)}$ the homogeneous component of highest graduation of f_i. The sequence F is said to be regular if the sequence of homogeneous polynomials $F^{(h)} = (f_1^{(h)}, \ldots, f_n^{(h)})$ is regular. An affine polynomial system is said to be regular if it is defined by an affine regular sequence.*

Let $F = \{f_1, \ldots, f_n\} \subset \mathbb{K}[x_1, \ldots, x_n]$ equipped with a fixed graduation $\deg_w$. Assume that F is an affine regular system as specified in the preceding definition. Let $G = \{g_1, \ldots, g_n\} \subset \mathbb{K}[x_1, \ldots, x_n, h]$ be the set of the homogenization of the elements in F. By equipping the polynomial ring $\mathbb{K}[x_1, \ldots, x_n, h]$ with the graduation $\deg_{w'}$ where $w'_{n+1} = 1$ and $w'_i = w_i$ for $i = 1, \ldots, n$, the complexity of computing the graded reverse lexicographical Gröbner basis of $\langle F \rangle$ can be bounded by the complexity of computing the graded reverse lexicographical Gröbner basis of $\langle G \rangle$. By consequence, for affine regular systems in $\mathbb{K}[x_1, \ldots, x_n]$, the complexity of computing a graded reverse lexicographical Gröbner basis can be bounded by the formula 4.3.3 after replacing n by $n + 1$ and setting $w_{n+1} = 1$.

When the system is not regular, the complexity of algorithms F_4 and F_5 is much more difficult to handle. Indeed, for affine non regular systems, some polynomials of graduation d in the ideal can be obtained by combination of polynomials of higher graduation *i.e.*:

$$f = \sum_{i=1}^{n} h_i f_i \text{ and } \exists i \in \{1, \ldots, n\} \text{ such that } \deg_w(h_i f_i) > \deg_w(f)\,. \tag{4.3.4}$$

As this phenomenon (called *degree fall*) is difficult to anticipate, the complexity of F_4 or F_5 is very hard to estimate and there is no general tight bound. Thus our estimation of the complexity to solve the PDP is given under the following hypothesis.

Hypothesis 1. *Polynomial systems arising from a Weil descent on summation polynomial on which we apply the change of coordinates corresponding to the action of the symmetric group are regular.*

This hypothesis was verified in every experiments we did. Thus we consider it as an *heuristic* in our case. But, the system we consider now is the one coming from the action of the 2-torsion point in addition of the full symmetric group. We prove that this regularity is kept:

Proposition 4.7. *Let* $(f_1, \ldots, f_n) \in \left(\mathbb{K}[x_1, \ldots, x_n]^{D_n}\right)^n \subset \left(\mathbb{K}[x_1, \ldots, x_n]^{\mathfrak{S}_n}\right)^n$ *be a sequence of polynomials such that* $(\Omega_{\mathfrak{S}_n}(f_1), \ldots, \Omega_{\mathfrak{S}_n}(f_n)) \in (\mathbb{K}[e_1, \ldots, e_n])^n$ *is a regular sequence for the usual graduation* $deg = deg_w$ *with* $w = (1, \ldots, 1)$.

If $\mathbb{K}[s_1, \ldots, s_{n-1}, e_n]$ *is equipped with a weighted degree* deg_w *of weights* $w = (2, \ldots, 2, 1)$ *then* $(\Omega_{D_n}(f_1), \ldots, \Omega_{D_n}(f_n)) \in (\mathbb{K}[s_1, \ldots, s_{n-1}, e_n])^n$ *is a regular sequence.*

We now have all the ingredients to estimate the complexity of the computation of the DRL Gröbner basis, it remains to estimate the second step, the change of ordering.

The classical algorithm of change of ordering for Gröbner basis is FGLM [63]. Its complexity is in $O(nD^3)$ arithmetic operations (recall that D is the number of solutions of the systems) but we want to use the algorithm with better bound proven in [61]:

$$O(n \log\log D \log^2(D) D + \log(D) D^\omega) \tag{4.3.5}$$

This is possible only when the situation is sufficiently generic. Thus, we need the following hypothesis in order to apply this result:

Hypothesis 2. *The LEX Gröbner basis of* $\langle \mathcal{S}_G \rangle$ *is in* Shape Position *and the matrix representing the multiplication by the smallest variable in the quotient* $\mathbb{K}[x_1, \ldots, x_n]/\langle \mathcal{S}_G \rangle$ *can be read from the DRL Gröbner basis of* $\langle \mathcal{S}_G \rangle$.

Depending on the expected computation, the group G in this hypothesis is instantiated to $\mathfrak{S}_n$ or D_n.

This hypothesis states that the smallest variable x_n (for the LEX ordering) is separating for the algebraic set of solutions of the corresponding ideal. This means that any solution in the variety is uniquely represented by the value of its x_n coordinate. Moreover, this hypothesis also states that the multiplication by this smallest variable can be retrieved by reading the DRL Gröbner basis. This situation seems very particular, but we prove in [61] that is actually generic. We check it on systems coming from the PDP problems (when G is instantiated to $\mathfrak{S}_n$ or D_n) and has been be always valid. We thus consider it as a heuristic.

The impact on the change of ordering when one uses the symmetries is related to the decreasing of the number of solutions. More exactly we have the following result.

Proposition 4.8. *Let* G *be a pseudo reflection group. Let* $\langle \mathcal{S} \rangle$ *be a* 0*-dim ideal generated by pointwise* G*-invariant polynomials. Applying the change of coordinates associated to* G *divides the degree of the ideal by* $(\#G)$.

It is now possible to present the main result of our work on solving the PDP using symmetries.

4.4 Gain of efficiency in the resolution of the PDP with 2-torsion action

Thanks to the results presented in section 4.3 and the structures of the systems $\mathcal{S}_{\mathfrak{S}_n}$ and $\mathcal{S}_{D_n}$, we can now deduce complexity results on solving the PDP. Here we consider a system $\mathcal{S}_{\mathfrak{S}_n}$ coming from a general elliptic curve where, in particular, the action of a 2-torsion is not present. We compare the complexity of solving the PDP for this case with the system $\mathcal{S}_{D_n}$ corresponding to an elliptic curve with a 2-torsion point (twisted Edwards, twisted Jacobi intersection curves and universal Edwards models).

For the first step of the algorithm 2, the complexity of computing a DRL Gröbner basis from the polynomial system $\mathcal{S}_{\mathfrak{S}_n}$ is given by (in this case, the weights are $(1,\dots,1)$):

$$\mathcal{S}_{\mathfrak{S}_n} \text{ case: } O\left(n\binom{n2^{n-1}+1}{n}^{\omega}\right) = O\left(ne^{\omega n}2^{\omega n(n-1)}\right)$$

arithmetic operations in $\mathbb{F}_q$. In the case where the symmetries take into account the 2-torsion, we obtain (here the weights are $(2,\dots,2,1)$ and thus the complexity is divided by $2^{(n-1)\omega}$, see relation 4.3.3):

$$\mathcal{S}_{D_n} \text{ case: } O\left(ne^{\omega n}\left(\frac{2^{n(n-1)}}{2^{n-1}}\right)^{\omega}\right) = O\left(ne^{\omega n}2^{\omega(n-1)^2}\right)$$

All these results are obtain under the assumption that the hypothesis 1 is verified.

For the second step, the change of ordering, we assume the hypothesis 2 verified. In the case of $\mathcal{S}_{\mathfrak{S}_n}$ the degree D is obtained from the Bézout's bound $2^{(n-1)n}$ and the complexity result follows.

$$\mathcal{S}_{\mathfrak{S}_n} \text{ case: } O\left(n^2 2^{\omega n(n-1)}\right)$$

and in the case where the 2-torsion action is taken into account, thanks to proposition 4.8, the degree of the ideal $\langle \mathcal{S}_{D_n} \rangle$ can be bounded by $2^{(n-1)n}/2^{n-1}$. Thus the change of ordering step has a cost bounded by

$$\mathcal{S}_{D_n} \text{ case: } O\left(n^2 \cdot 2^{\omega(n-1)^2}\right) .$$

Finally one can see that the step of DRL Gröbner basis computation is dominant and we obtain our first main result:

Theorem 4.3. *In the cases of twisted Edwards, twisted Jacobi intersection curves or universal Edwards model of elliptic curves. Under the hypotheses 1 and 2, the Point Decomposition Problem can be solved in time*

$$O\left(ne^{\omega n}2^{\omega(n-1)^2}\right)$$

where $2 \leq \omega < 3$ is the linear algebra constant. In particular, we obtain a gain in efficiency of about $2^{\omega(n-1)}$ over the case where the elliptic curve is general.

Again, it is important to note that these two hypotheses were verified in practice and can be considered as heuristics. We now present some experimental data showing that this gain is not only theoretical. The following table gives some timings corresponding to the comparison of solving the PDP when the 2-torsion is used (lines with D_n in front) or not (lines with $\mathfrak{S}_n$). For $n = 4$, all the computations were done with MAGMA, for $n = 5$ only `FGb` succeeded. This is already a result in itself since it is the first time that a PDP was solved for $n = 5$, which prove the impact of the use of the 2-torsion. In the following table we present the timings for a fixed 16-bit prime q and $n = 4$ or 5.

n		Step 1	Step 2	Total
		Time (s)	Time (s)	Time (s)
4	$\mathfrak{S}_n$	5	423	428
MAGMA	D_n	1	2	3
5	$\mathfrak{S}_n$	> 2 days		
fgb	D_n	567	2165	2732

We can observe that taking into account the symmetries dramatically decreases the computing time of the PDP resolution by a factor of about 400. This is consistent with the theoretical expected gain. Actually, for the computations done with MAGMA the gain is closer to $2^{3(n-1)}$ since the change of ordering is still FGLM and thus its cubic complexity is dominant in the total computation.

4.5 Symmetries in characteristic 2

In this section, the field $\mathbb{K}$ is a binary extension of non prime degree $\mathbb{F}_{2^{kn}}$.

The theorem 4.1 cannot be applied in the characteristic 2 for the group D_n (its order is $n!2^{n-1}$). But elliptic curves in characteristic 2 always contains a 2-torsion point, thus a natural question is to know if it is possible to adapt our approach in characteristic 2 without using the Shepard Todd theorem. Since this question is about invariance in ring, we go back to Galois theory.

An elliptic curve E defined over $\mathbb{K} = \mathbb{F}_{2^{kn}}$ with $j(E) \neq 0$ is defined by the equation $y^2+xy = x^3 + ax^2 + b$. Let γ be such that $b = \gamma^4$, the 2-torsion point $T_2 = (0, \gamma^2)$ in E has the following action in E:

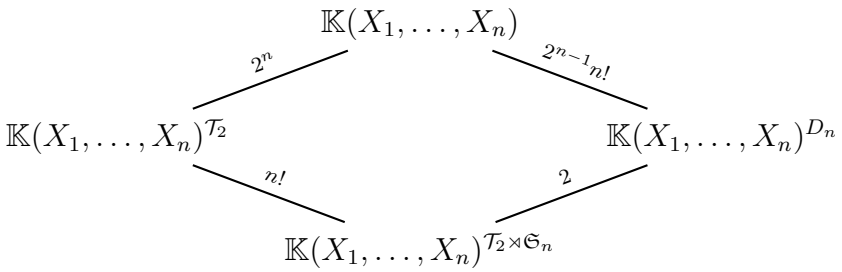
$$\forall P \in E : \quad P \oplus T_2 = (\frac{\gamma^2}{x(P)}, y(P))$$

The situation is not as easy as in the case of the curves we studied in large characteristic. We thus change the coordinates in order to obtain a simpler action. By applying the transformation $x \mapsto \frac{\gamma}{x+\gamma} + \lambda$, the 2-torsion point becomes $T_2 = (1 + \lambda, \gamma^2)$ and its action on E:

$$\forall P \in E : \quad P \oplus T_2 = (x(P) + 1, y(P))$$

This action seems easier to handle but not representable as a linear group. We thus use Galois theory and consider field extensions. Let $\mathcal{T}_2$ denote the full action $(X_i \rightarrow X_i + 1))$ of the 2-torsion point on an any number of coordinates. Again the group D_n represents the action of the

full symmetric group on the coordinates and the action of the 2-torsion point on an even number of coordinates. We have the following normal extensions



from which we obtain for a set of generators:

$\mathbb{K}(X_1,\ldots,X_n)^{\mathcal{T}_2} = \mathbb{K}(X_1^2+X_1,\ldots,X_n^2+X_n)$
$\mathbb{K}(X_1,\ldots,X_n)^{\mathcal{T}_2 \rtimes \mathfrak{S}_n} = \mathbb{K}(s_1,\ldots,s_n)$,
$\mathbb{K}(X_1,\ldots,X_n)^{D_n} = \mathbb{K}(e_1,s_2,\ldots,s_n)$

where $s_i = e_i(X_1^2+X_1,\ldots,X_n^2+X_n)$ (the polynomials e_i are still the elementary symmetric function, in particular $e_1 = X_1+\cdots+X_n$). We thus obtain the following result which extends the results we obtain in large characteristic

$$\mathbb{K}[X_1,\ldots,X_n]^{D_n} = \mathbb{K}[e_1,s_2,\ldots,s_n]\,.$$

The summation polynomial f_{n+1} computed from the curve E (after the application of the change of coordinates) verifies

$$f_{n+1}(x_1,\ldots,x_n,x_R) \in \mathbb{F}_{2^{kn}}[x_1,\ldots,x_n]^{D_n} = \mathbb{F}_{2^{kn}}[e_1,s_2\ldots,s_n]\,.$$

This gives a situation similar to the large characteristic. Actually, we prove that this polynomial is ever more structured in this case:

$$f_{n+1}(x_1,\ldots,x_n,x_R) \in \mathbb{F}_{2^{kn}}[e_1^2,s_2^2\ldots,s_{n-1}^2,s_n]\,.$$

From this result, we deduce a structured polynomial system $\mathcal{S}_{D_n}$ modeling the PDP in characteristic 2. By applying the same approach described above for a large characteristic, we obtain the following second main result.

Theorem 4.4. *Under the same hypothesis as in large characteristic, the PDP in characteristic* 2 *can be solved with a gain of*

$$2^{\omega 2(n-1)}$$

by using the action of the 2*-torsion point.*

Whether in theory or in practice using the action of the two torsion of binary elliptic curves allows to significantly improve the resolution of the *Point Decomposition Problem*. For example, for $n=5$ and a 16-bit base field, the PDP was intractable while one can now solve it in approximately 5 minutes using MAGMA. Note that by using FGb, timings are even much faster and the timings with MAGMA could be improved with an efficient implementation of change of ordering

algorithm [61] for *Shape Position* ideals. For instance for $k = 31$ and $n = 5$ solving the PDP can be achieved in approximately 10 seconds with FGb on one core of a 1.70GHz Intel® i7-4650U CPU. This result let us hope that an attack based on the index calculus would be possible in the near future for the ECDLP of the IPSEC Oakley key determination "Well Know Group 3" curve [88] defined over the field $\mathbb{F}_{2^{31\times 5}}$ or at least a very close example.

Bibliographie

[1] Leonard M ADLEMAN, Jonathan DEMARRAIS et Ming-Deh HUANG. “A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields”. In : *International Algorithmic Number Theory Symposium*. Springer. 1994, p. 28–40.

[2] David ADRIAN, Karthikeyan BHARGAVAN, Zakir DURUMERIC, Pierrick GAUDRY, Matthew GREEN, J. Alex HALDERMAN, Nadia HENINGER, Drew SPRINGALL, Emmanuel THOMÉ, Luke VALENTA, Benjamin VANDERSLOOT, Eric WUSTROW, Santiago Zanella BÉGUELIN et Paul ZIMMERMANN. “Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice”. In : *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*. Sous la dir. d’Indrajit RAY, Ninghui LI et Christopher KRUEGEL. ACM, 2015, p. 5–17. ISBN : 978-1-4503-3832-5.

[3] Martin R. ALBRECHT, Shi BAI et Léo DUCAS. “A Subfield Lattice Attack on Overstretched NTRU Assumptions - Cryptanalysis of Some FHE and Graded Encoding Schemes”. In : *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*. 2016, p. 153–178.

[4] Diego F. ARANHA, Pierre-Alain FOUQUE, Benoît GÉRARD, Jean-Gabriel KAMMERER, Mehdi TIBOUCHI et Jean-Christophe ZAPALOWICZ. “GLV/GLS Decomposition, Power Analysis, and Attacks on ECDSA Signatures with Single-Bit Nonce Bias”. In : *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*. Sous la dir. de Palash SARKAR et Tetsu IWATA. T. 8873. Lecture Notes in Computer Science. Springer, 2014, p. 262–281. ISBN : 978-3-662-45610-1.

[5] A. O. L. ATKIN et F. MORAIN. “Elliptic curves and primality proving”. In : *Mathematics of Computation* 61.203 (juil. 1993), p. 29–68.

[6] Guillaume BARBU, Alberto BATTISTELLO, Guillaume DABOSVILLE, Christophe GIRAUD, Guénaël RENAULT, Soline RENNER et Rina ZEITOUN. “Combined Attack on CRT-RSA - Why Public Verification Must Not Be Public?” In : *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key*

Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings. PDF. 2013, p. 198–215.

[7] Razvan BARBULESCU, Pierrick GAUDRY, Antoine JOUX et Emmanuel THOMÉ. "A Heuristic Quasi-Polynomial Algorithm for Discrete Logarithm in Finite Fields of Small Characteristic". In : *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*. 2014, p. 1–16.

[8] Magali BARDET, Jean-Charles FAUGÈRE et Bruno SALVY. "On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations". In : *International Conference on Polynomial System Solving - ICPSS*. Paris, France, nov. 2004, p. 71–75.

[9] Lucas BARTHÉLÉMY, Ninon EYROLLES, Guénaël RENAULT et Raphaël ROBLIN. "Binary Permutation Polynomial Inversion and Application to Obfuscation Techniques". In : *To appear in 2nd ACM International Workshop on Software Protection, SPRO 2016, Vienna, Austria, October 28, 2016*. PDF. 2016, p. 1–9.

[10] Aurélie BAUER et Antoine JOUX. "Toward a Rigorous Variation of Coppersmith's Algorithm on Three Variables". In : *Advances in Cryptology - EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings*. Sous la dir. de Moni NAOR. T. 4515. Lecture Notes in Computer Science. Springer, 2007, p. 361–378. ISBN : 978-3-540-72539-8.

[11] Eberhard BECKER, Teo MORA, Maria Grazia MARINARI et Carlo TRAVERSO. "The shape of the Shape Lemma". In : *Proceedings of the international symposium on Symbolic and algebraic computation*. ISSAC '94. Oxford, United Kingdom : ACM, 1994, p. 129–133. ISBN : 0-89791-638-7.

[12] T. BECKER et V. WEISPFENNING. *Gröbner bases*. T. 141. Graduate Texts in Mathematics. A computational approach to commutative algebra, In cooperation with Heinz Kredel. New York : Springer-Verlag, 1993, p. xxii+574. ISBN : 0-387-97971-9.

[13] Daniel J. BERNSTEIN, Peter BIRKNER, Marc JOYE, Tanja LANGE et Chistiane PETERS. "Twisted Edwards Curves". In : *Proceedings of the Cryptology in Africa 1st international conference on Progress in cryptology*. AFRICACRYPT'08. Casablanca, Morocco : Springer-Verlag, 2008, p. 389–405.

[14] Daniel J. BERNSTEIN et Tanja LANGE. "Faster Addition and Doubling on Elliptic Curves". In : *Advances in Cryptology : ASIACRYPT 2007*. T. 4833. Lecture Notes in Computer Science. Springer, 2007, p. 29–50.

[15] Jingguo BI, Jean-Sébastien CORON, Jean-Charles FAUGÈRE, Phong Q. NGUYEN, Guénaël RENAULT et Rina ZEITOUN. "Rounding and Chaining LLL: Finding Faster Small Roots of Univariate Polynomial Congruences". In : *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*. PDF. 2014, p. 185–202.

[16] Jean-François BIASSE et Claus FIEKER. “A polynomial time algorithm for computing the HNF of a module over the integers of a number field”. In : *International Symposium on Symbolic and Algebraic Computation, ISSAC'12, Grenoble, France - July 22 - 25, 2012*. Sous la dir. de Joris van der HOEVEN et Mark van HOEIJ. ACM, 2012, p. 75–82. ISBN : 978-1-4503-1269-1.

[17] Jean-François BIASSE et Fang SONG. “Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields”. In : *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*. Sous la dir. de Robert KRAUTHGAMER. SIAM, 2016, p. 893–902. ISBN : 978-1-61197-433-1.

[18] Fabrizio BIONDI, Sébastien JOSSE et Axel LEGAY. *Comparative Evaluation of the Effectiveness of Constraint Solvers against Opaque Conditionals*. Poster presented during the 36th IEEE Symposium on Security and Privacy. `http://www.ieee-security.org/TC/SP2015/program-posters.html`. 2015.

[19] Fabrizio BIONDI, Sébastien JOSSE, Axel LEGAY et Thomas SIRVENT. “Effectiveness of Synthesis in Concolic Deobfuscation”. working paper or preprint. Déc. 2015.

[20] Dereje Kifle BOKU, Wolfram DECKER, Claus FIEKER et Andreas STEENPASS. “Gröbner bases over algebraic number fields”. In : *Proceedings of the 2015 International Workshop on Parallel Symbolic Computation, PASCO 2015, Bath, United Kingdom, July 10-12, 2015*. 2015, p. 16–24.

[21] D. BONEH, R.A. DEMILLO et R.J. LIPTON. “On the Importance of Checking Cryptographic Protocols for Faults”. In : *EUROCRYPT*. 1997, p. 37–51.

[22] Dan BONEH et Glenn DURFEE. “Cryptanalysis of RSA with private key d less than $N^{0.292}$”. In : *IEEE Transactions on Information Theory* 46.4 (2000), p. 1339.

[23] Dan BONEH, Glenn DURFEE et Nick HOWGRAVE-GRAHAM. “Factoring N = $p^r q$ for Large r”. In : *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*. 1999, p. 326–337.

[24] Dan BONEH et Matthew K. FRANKLIN. “Identity-Based Encryption from the Weil Pairing”. In : *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*. 2001, p. 213–229.

[25] R. L. BORGER. “On De Moivre's Quintic”. In : *The American Mathematical Monthly* 15.10 (1908), p. 171–174. ISSN : 00029890.

[26] Alin BOSTAN, Bruno SALVY et Éric SCHOST. “Fast Algorithms for Zero-Dimensional Polynomial Systems Using Duality”. In : *Applicable Algebra in Engineering, Communication and Computing* 14.4 (2003), p. 239–272.

[27] Charles BOUILLAGUET, Chen-Mou CHENG, Tung CHOU, Ruben NIEDERHAGEN et Bo-Yin YANG. “Fast Exhaustive Search for Quadratic Systems in $\mathbb{F}_2$ on FPGAs”. In : *Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*. 2013, p. 205–222.

[28] Eric BRIER, Jean-Sébastien CORON, Thomas ICART, David MADORE, Hugues RANDRIAM et Mehdi TIBOUCHI. “Efficient Indifferentiable Hashing into Ordinary Elliptic Curves”. In : *CRYPTO*. `http://eprint.iacr.org/2009/340/`. 2010, p. 237–254.

[29] Johannes BUCHMANN. “Reducing lattice bases by means of approximations”. In : *Algorithmic Number Theory – Proc. ANTS-I*. T. 877. Lecture Notes in Computer Science. Springer, 1994, p. 160–168.

[30] Claude CARLET, Jean-Charles FAUGÈRE, Christopher GOYET et Guénaël RENAULT. “Analysis of the algebraic side channel attack”. In : *J. Cryptographic Engineering* 2.1 (2012). PDF, p. 45–62.

[31] A. CAUCHY. “Usage des fonctions interpolaires dans la détermination des fonctions symétriques des racines d’une équation algébrique donnée”. In : *Oeuvres* **5** (1840), 473 Extrait 108.

[32] Christophe CLAVIER, Benoit FEIX, Georges GAGNEROT, Mylène ROUSSELLET et Vincent VERNEUIL. “Square Always Exponentiation”. In : *INDOCRYPT*. 2011, p. 40–57.

[33] Christian COLLBERG. *Engineering Code Obfuscation*. Invited talk at EUROCRYPT’16. 2016.

[34] Don COPPERSMITH. “Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known”. In : *Advances in Cryptology - Proc. EUROCRYPT ’96*. T. 1070. Lecture Notes in Computer Science. Springer, 1996, p. 178–189.

[35] Don COPPERSMITH. “Finding a Small Root of a Univariate Modular Equation”. In : *Advances in Cryptology - Proc. EUROCRYPT ’96*. T. 1070. Lecture Notes in Computer Science. Springer, 1996, p. 155–165.

[36] Don COPPERSMITH. “Finding Small Solutions to Small Degree Polynomials”. In : *Cryptography and Lattices, International Conference, CaLC 2001, Providence, RI, USA, March 29-30, 2001, Revised Papers*. Sous la dir. de Joseph H. SILVERMAN. T. 2146. Lecture Notes in Computer Science. Springer, 2001, p. 20–31. ISBN : 3-540-42488-1.

[37] Don COPPERSMITH. “Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities”. In : *J. Cryptology* 10.4 (1997). Journal version of [35, 34], p. 233–260.

[38] Jean-Sébastien CORON, Jean-Charles FAUGÈRE, Guénaël RENAULT et Rina ZEITOUN. “Factoring $N = p^r q^s$ for Large r and s”. In : *Topics in Cryptology - CT-RSA 2016 - The Cryptographers’ Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings*. PDF. 2016, p. 448–464.

[39] Christophe COUPÉ, Phong Q. NGUYEN et Jacques STERN. "The Effectiveness of Lattice Attacks Against Low-Exponent RSA". In : *Public Key Cryptography – Proc. PKC '99*. T. 1560. Lecture Notes in Computer Science. Springer, 1999, p. 204–218.

[40] Jean-Marc COUVEIGNES et Reynald LERCIER. "Galois invariant smoothness basis". In : *Series on Number Theory and Its Applications* 5 (mai 2008). World Scientific, p. 142–167.

[41] C. COUVREUR et J.-J. QUISQUATER. "Fast Decipherment Algorithm for RSA Public-Key Cryptosystem". In : *Electronics Letters* 18.21 (1982), p. 905–907.

[42] D. COX, J. LITTLE et D. O'SHEA. *Ideals, varieties, and algorithms*. Second. Undergraduate Texts in Mathematics. An introduction to computational algebraic geometry and commutative algebra. New York : Springer-Verlag, 1997, p. xiv+536. ISBN : 0-387-94680-2.

[43] David A. COX. *Galois theory*. Pure and Applied Mathematics (New York). Hoboken, NJ : Wiley-Interscience [John Wiley & Sons], 2004, p. xx+559. ISBN : 0-471-43419-1.

[44] Xavier DAHAN et Éric SCHOST. "Sharp estimates for triangular sets". In : *Symbolic and Algebraic Computation, International Symposium ISSAC 2004, Santander, Spain, July 4-7, 2004, Proceedings*. 2004, p. 103–110.

[45] Hervé DAUDÉ et Brigitte VALLÉE. "An Upper Bound on the Average Number of Iterations of the LLL Algorithm". In : *Theor. Comput. Sci.* 123.1 (1994), p. 95–115.

[46] Jérémie DETREY, Guillaume HANROT, Xavier PUJOL et Damien STEHLÉ. "Accelerating Lattice Reduction with FPGAs". In : *Progress in Cryptology - LATINCRYPT 2010, First International Conference on Cryptology and Information Security in Latin America, Puebla, Mexico, August 8-11, 2010, Proceedings*. 2010, p. 124–143.

[47] C. DIEM. "On the discrete logarithm problem in class groups of curves". In : *Math. Comp* 80 (2011), p. 443–475.

[48] C. DIEM. "On the discrete logarithm problem in elliptic curves". In : *Compositio Mathematica* 147 (2011), p. 75–104.

[49] Emmanuelle DOTTAX, Christophe GIRAUD, Matthieu RIVAIN et Yannick SIERRA. "On Second-Order Fault Analysis Resistance for CRT-RSA Implementations". In : *WISTP*. 2009, p. 68–83.

[50] Vincent DUPAQUIS et Alexandre VENELLI. "Redundant Modular Reduction Algorithms". In : *CARDIS, 2011*. 2011, p. 102–114.

[51] Harold M. EDWARDS. "A Normal Form for Elliptic Curves". In : *Bulletin of the American Mathematical Society*. T. 44. Juil. 2007, p. 393–422.

[52] Andreas ENGE, Pierrick GAUDRY et Emmanuel THOMÉ. "An $L(1/3)$ Discrete Logarithm Algorithm for Low Degree Curves". In : *J. Cryptology* 24.1 (2011), p. 24–41.

[53] Thomas ESPITAU, Pierre-Alain FOUQUE, Alexandre GÉLIN et Paul KIRCHNER. "Computing generator in cyclotomic integer rings". In : *IACR Cryptology ePrint Archive* 2016 (2016), p. 957.

[54] Reza Rezaeian FARASHAHI, Hongfeng WU et Changan ZHAO. “Efficient Arithmetic on Elliptic Curves over Fields of Characteristic Three”. In : *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*. Sous la dir. de Lars R. KNUDSEN et Huapeng WU. T. 7707. Lecture Notes in Computer Science. Springer, 2012, p. 135–148. ISBN : 978-3-642-35998-9.

[55] J.-C. FAUGÈRE et C. MOU. “Fast Algorithm for Change of Ordering of Zero-dimensional Gröbner Bases with Sparse Multiplication Matrices”. In : *ISSAC ’11: Proceedings of the 2011 international symposium on Symbolic and algebraic computation*. ISSAC ’11. San Jose, USA : ACM, 2011, p. 1–8.

[56] Jean-Charles FAUGÈRE. “A New Efficient Algorithm for Computing Gröbner Bases (F4).” In : *Journal of Pure and Applied Algebra* 139.1–3 (juin 1999), p. 61–88. ISSN : 0022-4049.

[57] Jean-Charles FAUGÈRE. “A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5)”. In : *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*. ISSAC ’02. Lille, France : ACM, 2002, p. 75–83. ISBN : 1-58113-484-3.

[58] Jean-Charles FAUGÈRE, Mohab Safey El DIN et Thibaut VERRON. “On the complexity of computing Gröbner bases for weighted homogeneous systems”. In : *J. Symb. Comput.* 76 (2016), p. 107–141.

[59] Jean-Charles FAUGÈRE, Pierrick GAUDRY, Louise HUOT et Guénaël RENAULT. “Fast change of ordering with exponent ω”. In : *(Poster abstract) ACM Commun. Comput. Algebra* 46 (sept. 2012). PDF, p. 92–93.

[60] Jean-Charles FAUGÈRE, Pierrick GAUDRY, Louise HUOT et Guénaël RENAULT. *Polynomial Systems Solving by Fast Linear Algebra*. PDF. 2013.

[61] Jean-Charles FAUGÈRE, Pierrick GAUDRY, Louise HUOT et Guénaël RENAULT. “Subcubic change of ordering for Gröbner basis: a probabilistic approach”. In : *International Symposium on Symbolic and Algebraic Computation, ISSAC ’14, Kobe, Japan, July 23-25, 2014*. PDF. 2014, p. 170–177.

[62] Jean-Charles FAUGÈRE, Pierrick GAUDRY, Louise HUOT et Guénaël RENAULT. “Using Symmetries in the Index Calculus for Elliptic Curves Discrete Logarithm”. In : *J. Cryptology* 27.4 (2014). PDF, p. 595–635.

[63] Jean-Charles FAUGÈRE, Patrizia GIANNI, Daniel LAZARD et Teo MORA. “Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering”. In : *Journal of Symbolic Computation* 16.4 (1993), p. 329–344. ISSN : 0747-7171.

[64] Jean-Charles FAUGÈRE, Christopher GOYET et Guénaël RENAULT. “Attacking (EC)DSA Given Only an Implicit Hint”. In : *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*. PDF. 2012, p. 252–274.

[65] Jean-Charles FAUGÈRE, Louise HUOT, Antoine JOUX, Guénaël RENAULT et Vanessa VITSE. “Symmetrized Summation Polynomials: Using Small Order Torsion Points to Speed Up Elliptic Curve Index Calculus”. In : *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*. PDF. 2014, p. 40–57.

[66] Jean-Charles FAUGÈRE, Raphaël MARINIER et Guénaël RENAULT. “Implicit Factoring with Shared Most Significant and Middle Bits”. In : *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*. PDF. 2010, p. 70–87.

[67] Jean-Charles FAUGÈRE et Chenqi MOU. “Fast Algorithm for Change of Ordering of Zero-dimensional Gröbner Bases with Sparse Multiplication Matrices”. In : *ISSAC ’11: Proceedings of the 2011 international symposium on Symbolic and algebraic computation*. ISSAC ’11. San Jose, USA : ACM, 2011, p. 1–8.

[68] Jean-Charles FAUGÈRE, Ludovic PERRET, Christophe PETIT et Guénaël RENAULT. “Improving the Complexity of Index Calculus Algorithms in Elliptic Curves over Binary Fields”. In : *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*. PDF. 2012, p. 27–44.

[69] Jean-Charles FAUGÈRE, Ludovic PERRET et Frédéric de PORTZAMPARC. “Algebraic Attack against Variants of McEliece with Goppa Polynomial of a Special Form”. In : *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*. 2014, p. 21–41.

[70] Jean-Charles FAUGÈRE, Pierre-Jean SPAENLEHAUER et Jules SVARTZ. “Sparse Gröbner bases: the unmixed case”. In : *International Symposium on Symbolic and Algebraic Computation, ISSAC ’14, Kobe, Japan, July 23-25, 2014*. Sous la dir. de Katsusuke NABESHIMA, Kosaku NAGASAKA, Franz WINKLER et Ágnes SZÁNTÓ. ACM, 2014, p. 178–185. ISBN : 978-1-4503-2501-1.

[71] Jean-Charles FAUGÈRE et Jules SVARTZ. “Gröbner bases of ideals invariant under a commutative group: the non-modular case”. In : *International Symposium on Symbolic and Algebraic Computation, ISSAC’13, Boston, MA, USA, June 26-29, 2013*. 2013, p. 347–354.

[72] Claus FIEKER et Jürgen KLÜNERS. “Computation of Galois groups of rational polynomials”. In : *LMS Journal of Computation and Mathematics* 17.1 (jan. 2014), p. 141–158.

[73] Pierre-Alain FOUQUE et Mehdi TIBOUCHI. “Deterministic Encoding and Hashing to Odd Hyperelliptic Curves”. In : *Pairing-Based Cryptography - Pairing 2010 - 4th International Conference, Yamanaka Hot Spring, Japan, December 2010. Proceedings*. Sous la dir. de Marc JOYE, Atsuko MIYAJI et Akira OTSUKA. T. 6487. Lecture Notes in Computer Science. Springer, 2010, p. 265–277. ISBN : 978-3-642-17454-4.

[74] Joshua FRIED, Pierrick GAUDRY, Nadia HENINGER et Emmanuel THOMÉ. “A kilobit hidden SNFS discrete logarithm computation”. In : *IACR Cryptology ePrint Archive* 2016 (2016), p. 961.

[75] H. GARNER. “The Residue Number System”. In : *IRE Transactions on Electronic Computers* 8.6 (juin 1959), p. 140–147.

[76] Joachim von zur GATHEN et Jürgen GERHARD. *Modern computer algebra (2nd ed.)* Cambridge University Press, 2003, p. I–XIII, 1–785. ISBN : 978-0-521-82646-4.

[77] Pierrick GAUDRY. “An Algorithm for Solving the Discrete Log Problem on Hyperelliptic Curves”. In : *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*. Sous la dir. de Bart PRENEEL. T. 1807. Lecture Notes in Computer Science. Springer, 2000, p. 19–34. ISBN : 3-540-67517-5.

[78] Pierrick GAUDRY. “Index Calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem”. In : *Journal of Symbolic Computation* 44.12 (2009), p. 1690–1702.

[79] Pierrick GAUDRY, Emmanuel THOMÉ, Nicolas THÉRIAULT et Claus DIEM. “A double large prime variation for small genus hyperelliptic index calculus”. In : *Math. Comput.* 76.257 (2007), p. 475–492.

[80] Patrizia GIANNI et Teo MORA. “Algebraic Solution of Systems of Polynomial Equations using Gröbner Bases”. In : *Applied Algebra, Algebraic Algorithms and Error Correcting Codes, Proceedings of AAECC-5, volume 356 of LNCS*. Springer, 1989, p. 247–257.

[81] Dahmun GOUDARZI, Matthieu RIVAIN et Damien VERGNAUD. “Lattice Attacks against Elliptic-Curve Signatures with Blinded Scalar Multiplication”. In : *Selected Areas in Cryptography - SAC 2016*. Sous la dir. de Roberto AVANZI et Howard HEYS. Selected Areas in Cryptography - SAC 2016. St. John’s, Canada : Springer, août 2016.

[82] Tim GÜNEYSU, Timo KASPER, Martin NOVOTNÝ, Christof PAAR et Andy RUPP. “Cryptanalysis with COPACOBANA”. In : *IEEE Trans. Computers* 57.11 (2008), p. 1498–1513.

[83] P.K. GUPTA. “Xeon+FPGA Platform for the Data Center (presentation only)”. In : *The Fourth Workshop on the Intersections of Computer Architecture and Reconfigurable Logic (CARL 2015)*. 2015.

[84] Mark van HOEIJ, Jürgen KLÜNERS et Andrew NOVOCIN. “Generating subfields”. In : *J. Symb. Comput.* 52 (2013), p. 17–34.

[85] Joris van der HOEVEN et Mark van HOEIJ, éds. *International Symposium on Symbolic and Algebraic Computation, ISSAC’12, Grenoble, France - July 22 - 25, 2012*. ACM, 2012. ISBN : 978-1-4503-1269-1.

[86] Nick HOWGRAVE-GRAHAM. “Finding Small Roots of Univariate Modular Equations Revisited”. In : *Cryptography and Coding – Proc. IMA ’97*. T. 1355. Lecture Notes in Computer Science. Springer, 1997, p. 131–142.

[87] T. ICART. “How to Hash into Elliptic Curves”. In : *CRYPTO*. 2009, p. 303–316.

[88] IETF. *The Oakley key determination protocol, IETF RFC 2412*. 1998.

[89] Charanjit S. JUTLA. “On Finding Small Solutions of Modular Multivariate Polynomial Equations”. In : *Advances in Cryptology - EUROCRYPT ’98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*. Sous la dir. de Kaisa NYBERG. T. 1403. Lecture Notes in Computer Science. Springer, 1998, p. 158–170. ISBN : 3-540-64518-7.

[90] Jean-Gabriel KAMMERER, Reynald LERCIER et Guénaël RENAULT. “Encoding points on hyperelliptic curves over finite fields in deterministic polynomial time”. In : *Pairing-based cryptography—Pairing 2010*. T. 6487. Lecture Notes in Comput. Sci. PDF. Springer, Berlin, 2010, p. 278–297.

[91] Richard KANE. *Reflection Groups and Invariant Theory*. Springer, 2001.

[92] Walter KELLER-GEHRIG. “Fast Algorithms for the Characteristic Polynomial”. In : *Theor. Comput. Sci.* 36 (2-3 juin 1985), p. 309–317. ISSN : 0304-3975.

[93] Masanari KIDA, Guénaël RENAULT et Kazuhiro YOKOYAMA. “Quintic polynomials of Hashimoto-Tsunogai, Brumer and Kummer”. In : *Int. J. Number Theory* 5.4 (2009). PDF, p. 555–571. ISSN : 1793-0421.

[94] J.-L. LAGRANGE. “Réflexions sur la résolution algébrique des équations”. In : *Nouveaux Mémoires de l’Académie Royale des Sciences et Belles-Lettres de Berlin* 3 (1770-1771), p. 205–421.

[95] Yagati N. LAKSHMAN. “On the Complexity of Computing a Gröbner Basis for the Radical of a Zero Dimensional Ideal”. In : *Proceedings of the twenty-second annual ACM symposium on Theory of computing*. STOC ’90. Baltimore, Maryland, United States : ACM, 1990, p. 555–563. ISBN : 0-89791-361-2.

[96] Yagati N. LAKSHMAN et Daniel LAZARD. “On the Complexity of Zero-Dimensional Algebraic Systems”. In : *Effective methods in algebraic geometry*. T. 94. Birkhauser. 1991, p. 217.

[97] Susan LANDAU. “Polynomial Time Algorithms for Galois Groups”. In : *EUROSAM 84, International Symposium on Symbolic and Algebraic Computation, Cambridge, England, July 9-11, 1984, Proceedings*. Computation. 1984, p. 225–236.

[98] Susan LANDAU et Gary L. MILLER. “Solvability by Radicals is in Polynomial Time”. In : *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*. 1983, p. 140–151.

[99] D. LAZARD. “Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations”. In : *Computer Algebra*. Sous la dir. de J. van HULZEN. T. 162. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 1983, p. 146–156. ISBN : 978-3-540-12868-7.

[100] D. LAZARD. “Solving zero-dimensional algebraic systems”. In : *J. Symbolic Comput.* 13.2 (1992), p. 117–131. ISSN : 0747-7171.

[101] Daniel LAZARD et Annick VALIBOUZE. “Computing subfields: Reverse of the primitive element problem”. In : *In Computational algebraic geometry*. Birkhäuser, 1993, p. 163–176.

[102] Romain LEBRETON et Éric SCHOST. “Algorithms for the universal decomposition algebra”. In : *International Symposium on Symbolic and Algebraic Computation, ISSAC'12, Grenoble, France - July 22 - 25, 2012*. Sous la dir. de Joris van der HOEVEN et Mark van HOEIJ. ACM, 2012, p. 234–241. ISBN : 978-1-4503-1269-1.

[103] M. LEDERER. “Explicit constructions in splitting fields of polynomials”. In : *Riv. Mat. Univ. Parma (7)* 3* (2004), p. 233–244. ISSN : 0035-6298.

[104] A. K. LENSTRA, H. W. LENSTRA Jr. et L. LOVÁSZ. “Factoring polynomials with rational coefficients”. In : *Mathematische Ann.* 261 (1982), p. 513–534.

[105] H. W. LENSTRA et A. SILVERBERG. “Lattices with Symmetry”. In : *IACR Cryptology ePrint Archive* 2014 (2014), p. 1026.

[106] H. W. LENSTRA et A. SILVERBERG. “Revisiting the Gentry-Szydlo Algorithm”. In : *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*. 2014, p. 280–296.

[107] J. LV, P. KALLA et F. ENESCU. “Efficient Gröbner Basis Reductions for Formal Verification of Galois Field Arithmetic Circuits”. In : *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 32.9 (sept. 2013), p. 1409–1420. ISSN : 0278-0070.

[108] Kenneth L. MANDERS et Leonard ADLEMAN. “NP-Complete decision problems for binary quadratics”. In : *Journal of Computer and System Sciences* 16.2 (1978), p. 168–184. ISSN : 0022-0000.

[109] A. MAY. “Using LLL-Reduction for Solving RSA and Factorization Problems: A Survey”. In : In [116]. 2010.

[110] Alexander MAY et Maike RITZENHOFEN. “Implicit Factoring: On Polynomial Time Factoring Given Only an Implicit Hint”. In : *Public Key Cryptography - PKC 2009, 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings*. 2009, p. 1–14.

[111] J. MCKAY et R. STAUDUHAR. “Finding relations among the roots of an irreducible polynomial”. In : *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation (Kihei, HI)*. New York : ACM, 1997, 75–77 (electronic).

[112] Bernard MOURRAIN et Victor Y PAN. “Asymptotic Acceleration of Solving Multivariate Polynomial Systems of Equations”. In : *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM. 1998, p. 488–496.

[113] Ginger MYLES et Christian S. COLLBERG. “Software watermarking via opaque predicates: Implementation, analysis, and attacks”. In : *Electronic Commerce Research* 6.2 (2006), p. 155–171.

[114] H. NAJAFI, M.E.D. JAFARI et M.-O. DAMEN. “On Adaptive Lattice Reduction over Correlated Fading Channels”. In : *Communications, IEEE Transactions on* 59.5 (2011), p. 1224–1227.

[115] P. Q. NGUYEN et D. STEHLÉ. “An LLL Algorithm with Quadratic Complexity”. In : *SIAM J. of Computing* 39.3 (2009), p. 874–903.

[116] P. Q. NGUYEN et B. VALLÉE, éds. *The LLL Algorithm: Survey and Applications*. Information Security and Cryptography. Springer, 2010.

[117] Phong Q. NGUYEN et Igor E. SHPARLINSKI. “The Insecurity of the Digital Signature Algorithm with Partially Known Nonces”. In : *J. Cryptology* 15.3 (2002), p. 151–176.

[118] NIST. *Post-Quantum Crypto Standardization*. `http://csrc.nist.gov/groups/ST/post-quantum-crypto/index.html`. 2016.

[119] Andrew NOVOCIN, Damien STEHLÉ et Gilles VILLARD. “An LLL-reduction algorithm with quasi-linear time complexity: extended abstract”. In : *Proc. STOC ’11*. ACM, 2011, p. 403–412.

[120] Sébastien ORANGE, Guénaël RENAULT et Kazuhiro YOKOYAMA. “Computation schemes for splitting fields of polynomials”. In : *ISSAC 2009—Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation*. PDF. ACM, New York, 2009, p. 279–286.

[121] Sébastien ORANGE, Guénaël RENAULT et Kazuhiro YOKOYAMA. “Efficient arithmetic in successive algebraic extension fields using symmetries”. In : *Math. Comput. Sci.* 6.3 (2012). PDF, p. 217–233. ISSN : 1661-8270.

[122] Tim PRUSS, Priyank KALLA et Florian ENESCU. *Word-Level Abstraction from Bit-Level Circuits using Gröbner Bases*. Paper presented during the International Workshop on Logic and Synthesis. `http://www.ece.utah.edu/~pruss/downloads/abstractIWLS13.pdf`. 2013.

[123] G. RENAULT et Yokoyama K. “A Modular Method for Computing the Splitting Field of a Polynomial.” In : *ANTS*. 2006, p. 124–140.

[124] Guénaël RENAULT. “Computation of the splitting field of a dihedral polynomial”. In : *ISSAC 2006*. ACM, New York, 2006, p. 290–297.

[125] Guénaël RENAULT. “Introduction à la Théorie de Galois Effective”. In : *JNCF’08: Journées Nationales du Calcul Formel (online)*. PDF, Slides. Marseille, France, oct. 2008, p. 141–197.

[126] Guénaël RENAULT. *Introduction à l’Algorithmique Galoisienne*. Invited talk at Mathematik Park (Institut Henri Poincaré), Slides. 2011.

[127] Guénaël RENAULT. *On polynomial systems with structures related to the ECDLP*. Invited talk during the Conference Effective Moduli Spaces and Applications to Cryptography (Rennes, France). 2014.

[128] Guénaël RENAULT. *On Using Torsion Points in the Elliptic Curve Index Calculus*. Invited talk during the 18th Workshop On Elliptic Curve Cryptography (ECC 2014), Slides. 2014.

[129] Guénaël RENAULT. *The Heuristic Coppersmith Technique from a Computer Algebra Point of View*. Invited talk during the SIAM Conference on Applied Algebraic Geometry (Fort Collins, Colorado, USA), Slides. 2013.

[130] Guénaël RENAULT et Tristan VACCON. "On the p-adic stability of the FGLM algorithm". In : *CoRR* abs/1602.00848 (2016). PDF.

[131] Guénaël RENAULT et Kazuhiro YOKOYAMA. "A modular method for computing the splitting field of a polynomial". In : *Algorithmic number theory*. T. 4076. Lecture Notes in Comput. Sci. Springer, Berlin, 2006, p. 124–140.

[132] Guénaël RENAULT et Kazuhiro YOKOYAMA. "Multi-modular algorithm for computing the splitting field of a polynomial". In : *ISSAC 2008*. PDF. ACM, New York, 2008, p. 247–254.

[133] Maike RITZENHOFEN. "On efficiently calculating small solutions of systems of polynomial equations: lattice-based methods and applications to cryptography". Thèse de doct. Ruhr University Bochum, 2010.

[134] Santanu SARKAR et Subhamoy MAITRA. "Further results on implicit factoring in polynomial time". In : *Adv. in Math. of Comm.* 3.2 (2009), p. 205–217.

[135] I. SEMAEV. *Summation Polynomials and the Discrete Logarithm Problem on Elliptic Curves*. Cryptology ePrint Archive, Report 2004/031. `http://eprint.iacr.org/`. 2004.

[136] A. SHALLUE et C. van de WOESTIJNE. "Construction of Rational Points on Elliptic Curves over Finite Fields". In : *ANTS*. 2006, p. 510–524.

[137] Geoffrey C. SHEPHARD et John A. TODD. "Finite unitary reflection groups". In : *Canadian J. Math.* 6 (1954), p. 274–304.

[138] Victor SHOUP. "OAEP Reconsidered". In : *J. Cryptology* 15.4 (2002), p. 223–249.

[139] M. ULAS. "Rational points on certain hyperelliptic curves over finite fields". In : *Bull. Polish Acad. Sci. Math.* 55 (2007), p. 97–104.

[140] Tristan VACCON. "Matrix-F5 algorithms over finite-precision complete discrete valuation fields". In : *International Symposium on Symbolic and Algebraic Computation, ISSAC '14, Kobe, Japan, July 23-25, 2014*. 2014, p. 397–404.

[141] Erich WENGER et Paul WOLFGER. "Harder, better, faster, stronger: elliptic curve discrete logarithm computations on FPGAs". In : *J. Cryptographic Engineering* 6.4 (2016), p. 287–297.

[142] Gareth Andrew WHITE. "Algorithms for Galois Group Computations over Multivariate Function Fields". In : *Bulletin of the Australian Mathematical Society* 94.1 (août 2016), p. 169–170.

[143] Marc WITTEMAN, Jasper van WOUDENBERG et Federico MENARINI. “Defeating RSA Multiply-Always and Message Blinding Countermeasures”. In : *CTRSA*. 2011, p. 77–88.

[144] K. YOKOYAMA. “A modular method for computing the Galois groups of polynomials”. In : *J. Pure Appl. Algebra* 117/118 (1997). Algorithms for algebra (Eindhoven, 1996), p. 617–636. ISSN : 0022-4049.

[145] Yongxin ZHOU, Alec MAIN, Yuan X. GU et Harold JOHNSON. “Information Hiding in Software with Mixed Boolean-Arithmetic Transforms”. In : *Information Security Applications: 8th International Workshop, WISA 2007, Jeju Island, Korea, August 27-29, 2007, Revised Selected Papers*. Sous la dir. de Sehun KIM, Moti YUNG et Hyung-Woo LEE. Berlin, Heidelberg : Springer Berlin Heidelberg, 2007, p. 61–75. ISBN : 978-3-540-77535-5.