UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Discrete Logarithm Cryptanalyses: Number Field Sieve and Lattice Tools for Side-Channel Attacks

THÈSE

soutenue le 25 mai 2021

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Gabrielle De Micheli

Composition du jury

Président :	Steve Kremer	Directeur de recherche, INRIA, Nancy, France
Rapporteurs :	Martin Albrecht	Professor, Royal Holloway, University of London, Royaume-Uni
	Frederik Vercauteren	Associate professor, KU Leuven, Belgique
Examinateurs :	Robert Granger	Lecturer, University of Surrey, Royaume-Uni
	Tanja Lange	Professor, Technische Universiteit Eindhoven, Pays-Bas
	Palash Sarkar	Professor, Indian Statistical Institute, Inde
Encadrants :	Pierrick Gaudry	Directeur de recherche, CNRS, Nancy, France
	Cécile Pierrot	Chargée de recherche, INRIA, Nancy, France

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Résumé en Français

La cryptographie à clé publique

La cryptographie s'intéresse au problème de l'échange de messages chiffrés, c'est-à-dire inintelligibles, que seul un récepteur légitime peut déchiffrer, donc lire. Afin d'assurer une transmission sécurisée de ces messages, une clé secrète est généralement partagée entre l'expéditeur et le destinataire. Cela pose la difficulté importante d'échanger de manière sécurisée la clé secrète mentionnée ci-dessus.

Au début des années 1970, Merkle a commencé à s'écarter de ce concept de clé partagée et ses idées publiées en 1978 [Mer78] ont été reprises dans l'article fondateur de Diffie et Hellman [DH76], *New directions in Cryptography*. Dans leur article, Diffie et Hellman formalisent la notion de cryptographie à clé publique où deux clés mathématiquement liées sont générées et utilisées : une clé publique et une clé secrète. Un message est ensuite chiffré à l'aide de la clé publique du récepteur. Ce dernier sera alors le seul capable de déchiffrer le message à l'aide de sa clé secrète correspondante.

Les cryptosystèmes à clé publique, également connus sous le nom de protocoles asymétriques, sont tous construits en tenant compte de la notion de fonction à sens unique. Cette dernière correspond à une fonction qui est facile à calculer pour toute entrée donnée mais difficile à inverser. Cette notion correspond bien aux exigences d'un protocole asymétrique. En effet, pour qu'un protocole soit sûr et efficace, le déchiffrement d'un message sans la clé secrète doit être proche de l'impossible, alors que le chiffrement d'un message et le déchiffrement avec la clé secrète doivent être faciles, c'est-à-dire réalisés uniquement avec des opérations simples.

C'est naturellement vers des problèmes mathématiques difficiles que les cryptographes se sont tournés pour trouver des primitives appropriées pour leurs protocoles. Historiquement, deux candidats ont émergé : la multiplication de deux nombres premiers et l'exponentiation modulaire. L'inverse de ces fonctions consiste à factoriser un nombre entier et à calculer un logarithme discret. La difficulté de la factorisation est au cœur du cryptosystème RSA, bien connu et déployé [RSA78]. Dans cette thèse, nous nous concentrerons sur le second candidat : l'exponentiation modulaire et son opération inverse, le calcul d'un logarithme discret.

Exponentiation modulaire et logarithme discret

L'exponentiation modulaire consiste à calculer le reste d'une division euclidienne d'un entier g élevé à une puissance x par un entier positif N, en calculant $g^x \pmod N$. Cette opération est fondamentale dans la théorie computationnelle des nombres où elle se retrouve par exemple dans le petit théorème de Fermat utilisé pour le test de primalité. L'exponentiation modulaire est également largement utilisée en cryptographie à clé publique, où les éléments de groupes tels que les groupes multiplicatifs des corps finis, $\mathbb{Z}/N\mathbb{Z}$ ou le groupe des points rationnels des courbes elliptiques, sont souvent élevés à de grandes puissances.

Pour des raisons pratiques, les opérations utilisées dans les protocoles cryptographiques doivent être faciles et efficaces à réaliser. L'attrait de l'exponentiation modulaire pour la cryptographie provient en partie de la simplicité de son calcul. En effet, le calcul d'une exponentiation modulaire peut se résumer à multiplier g par lui-même $x - 1$ fois, puis à prendre le résultat modulo N. Cependant, cette méthode serait très inefficace dans un contexte cryptographique en raison de la taille des nombres impliqués. Par conséquent, afin de construire des protocoles cryptographiques pratiques qui utilisent l'exponentiation modulaire, l'opération doit être effectuée de manière efficace.

L'efficacité des algorithmes qui calculent l'exponentiation modulaire dépend de divers paramètres tels

que le groupe considéré, la représentation de l'exposant ou le matériel utilisé. Comme l'exponentiation modulaire est présente dans de nombreux protocoles, et qu'elle est souvent l'opération la plus coûteuse du protocole, au fil des années, de nombreux algorithmes optimisés se sont accumulés pour améliorer son calcul.

La plupart des algorithmes qui tentent d'optimiser l'exponentiation modulaire visent à réduire le nombre de multiplications nécessaires. L'une des méthodes les plus anciennes et les plus simples est l'algorithme du carré et de la multiplication (Square-and-Multiply, en anglais). Cette méthode est apparue il y a plus de 2000 ans en Inde [Knu97]. L'algorithme opère bit par bit sur les bits de l'exposant et n'effectue une opération de multiplication que si ce bit de l'exposant est 1.

Des algorithmes d'exponentiation modulaire plus sophistiqués font appel à différentes représentations de l'exposant, comme la forme non adjacente (NAF) ou sa variante à fenêtre (wNAF). Nous renvoyons le lecteur à [Gor98] pour une étude des méthodes d'exponentiation rapide.

Bien que de nombreux efforts aient été déployés pour optimiser les algorithmes d'exponentiation modulaire, ces optimisations ont fait apparaître des vulnérabilités exploitables. En effet, les opérations effectuées dans ces algorithmes sont souvent dépendantes des valeurs binaires de l'exposant. L'exécution du code peut alors générer des fuites observables à partir desquelles des informations peuvent être déduites sur l'exposant. Le caractère spécifique des informations divulguées dépend des détails de la mise en œuvre de l'algorithme et souvent du matériel lui-même. Les attaques par canaux auxiliaires et en particulier les attaques par cache sont les principales menaces à prendre en compte lors de l'utilisation d'un algorithme d'exponentiation modulaire rapide pour un protocole.

L'opération inverse de l'exponentiation modulaire est le calcul d'un logarithme discret. L'étude des logarithmes discrets et des algorithmes associés précède leur utilisation en cryptographie. En effet, dès le 19^e siècle, les logarithmes de Zech sont utilisés pour accélérer les opérations arithmétiques dans les corps finis.

En cryptographie, le protocole Diffie-Hellman datant de la fin des années 70 a marqué un tournant dans l'étude des logarithmes discrets. L'utilisation plus récente des logarithmes discrets dans les protocoles basés sur les couplages, qui a débuté au début des années 2000, a relancé l'intérêt pour le sujet. Concrètement, un logarithme discret est défini comme suit.

Definition 27 (Logarithme discret)**.** *Étant donné un groupe cyclique fini G d'ordre n, un générateur $g \in G$ et un élément $h \in G$, le logarithme discret de h en base g est l'élément $x \in [0, n[$ tel que $g^x = h$.*

Cette définition pose le problème suivant.

Definition 28 (Le problème du logarithme discret (DLP))**.** *Étant donné un groupe cyclique fini G d'ordre n, un générateur $g \in G$, et un élément $h \in G$, trouver x tel que $g^x = h$.*

Ce problème est considéré comme difficile pour la plupart des groupes G et constitue donc un candidat prometteur pour la cryptographie à clé publique.

Question 90. *Où peut-on trouver des logarithmes discrets et des exponentiations modulaires en cryptographie ?*

Des cryptosystèmes largement déployés, tels que le protocole d'échange de clés Diffie-Hellman, le protocole de chiffrement d'ElGamal ou les protocoles de signature tels que (EC)DSA basent leur sécurité sur des hypothèse liées à la difficulté du problème du logarithme discret. Nous décrivons certains de ces protocoles.

Le protocole d'échange de clés de Diffie-Hellman [DH76]. Le protocole est assez simple. Deux entités, Alice et Bob, souhaitent communiquer et pour ce faire, elles doivent d'abord se mettre d'accord sur une clé secrète. Elles commencent par choisir un groupe G d'ordre n et un générateur g de ce groupe, qui sont maintenant des paramètres publiques. Alice choisit ensuite un élément aléatoire $a \in [0, n[$ et envoie l'élément de groupe g^a à Bob via un canal publique. De même, Bob choisit $b \in [0, n[$ et envoie g^b à

Alice. Les quantités, g, g^a, g^b sont toutes publiques. Alice et Bob peuvent maintenant tous deux calculer la quantité

$$s = (g^a)^b = (g^b)^a,$$

qui constitue le secret partagé utilisé pour les communications futures.

Un attaquant qui souhaite intercepter une conversation entre Alice et Bob doit récupérer la valeur secrète s. La récupération de g^{ab} à partir de g, g^a, g^b est connue comme le problème calculatoire de Diffie-Hellman, qui est étroitement lié au calcul des logarithmes discrets [MW96]. Le protocole d'échange de clés Diffie-Hellman est devenu une norme en 2003 (ANSI X9.42) et est utilisé dans des protocoles largement déployés tels que HTTPS, SSH/TLS.

Le protocol de chiffrement d'ElGamal [ElG85]. Le protocole de chiffrement d'ElGamal est étroitement lié à l'échange de clés Diffie-Hellman. Alice veut envoyer un message chiffré à Bob. Comme pour tout protocole de chiffrement à clé publique, Bob doit générer une paire de clés, une publique et une privée, qui sont mathématiquement liées. Pour ce faire, il choisit un groupe G d'ordre n ainsi qu'un générateur g de G. La clé privée de Bob sera un élément $b \in [0, n[$ choisi au hasard et connu seulement par Bob. La clé publique est constituée des éléments (G, g, g^b).

Afin de chiffrer un message donné m vu comme un élément de G, Alice utilisera la clé publique de Bob. Alice commence par choisir un élément aléatoire $r \in [0, n[$ et calcule la quantité $c = m \cdot (g^b)^r$. Alice calcule également $c' = g^r$. Le texte chiffré est donc composé des deux quantités (c, c') qu'Alice envoie à Bob.

Une fois que Bob reçoit le texte chiffré, il peut déchiffrer le message m en utilisant sa clé privée b. En effet, Bob calcule $c(c'^b)^{-1} = m$.

Un attaquant qui intercepte le texte chiffré (c, c') et souhaite le déchiffrer devrait résoudre le problème calculatoire de Diffie-Hellman. Le schéma de chiffrement ElGamal est largement utilisé pour les systèmes de vote [BW14].

Le protocole de signature d'ElGamal [ElG85]. Considérons $G = (\mathbb{Z}/p\mathbb{Z})^*$ pour un nombre premier p. Alice veut envoyer un message m à Bob et en plus elle veut signer le message afin de l'authentifier. Comme précédemment, elle possède une paire de clés secrètes/publiques (a, g^a) où g est un générateur du groupe G considéré. Pour générer sa signature, Alice choisit un entier aléatoire $k \in [0, p-1]$ où p est l'ordre premier de G, et tel que k et $p-1$ sont premiers entre-eux. Elle calcule alors les deux quantités $r = g^k \pmod p$ et $s = (m - ar)k^{-1} \pmod{p-1}$. Sa signature est la paire (r, s) et elle l'envoie à Bob avec le message m.

Si Bob veut vérifier la validité du message, il vérifie la signature d'Alice en utilisant sa clé publique. D'après la génération de la signature, nous savons que $m = ar + sk \pmod{p-1}$. Puisque Bob connaît la clé publique d'Alice, g^a, nous pouvons comparer les quantités g^m et $g^{ar}g^{sk}$ modulo p.

D'autres schémas de signature tels que l'algorithme de signature numérique (DSA) de 1991 (spécifications dans FIPS 186-4 [NIS13]), une variante du schéma de signature ElGamal, et sa variante utilisant les courbes elliptiques ECDSA [JMV01], sont également des protocoles normalisés basés sur la difficulté de DLP.

Protocoles de couplage. La cryptographie basée sur les couplages est au cœur de nombreux produits de sécurité qui sont mis sur le marché et la recherche de primitives efficaces les utilisant est très active. C'est le cas notamment dans le domaine du zero-knowledge avec les applications de Zk-SNARKs aux smart contracts.

Les preuves zero-knowledge permettent à un vérificateur de certifier qu'un prouveur a connaissance d'un secret sans révéler d'informations sur le secret lui-même. Les Zk-SNARKs, Zero-knowledge Succinct Non-interactive Argument of Knowledge, sont des exemples de protocoles largement déployés dans les smart contracts qui utilisent des preuves zero-knowledge. Beaucoup de ces protocoles utilisent des couplages dans leurs constructions. L'évaluation de la sécurité de ces schémas est donc fondamentale. Concrètement, un couplage cryptographique est défini comme suit.

Definition 29 (Couplage cryptograhique)**.** *Considérons les groupes abéliens finis $\mathbb{G}_1$, $\mathbb{G}_2$, et $\mathbb{G}_T$ d'ordre n. Un couplage cryptographique est une application*

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T,$$

qui est

- *bilinéaire, c'est-à-dire que pour tout $a, b \in [0, n[$, $P \in \mathbb{G}_1$ et $Q \in \mathbb{G}_2$ on a*

$$e(aP, bQ) = e(P, Q)^{ab},$$

- *non-dégénérée, c'est-à-dire que pour tout $P \in \mathbb{G}_1$, il existe $Q \in \mathbb{G}_2$ tel que $e(P, Q) \neq 1$ et pour tout $Q \in \mathbb{G}_2$ il existe $P \in \mathbb{G}_1$ tel que $e(P, Q) \neq 1$,*
- *et calculable en temps polynomial dans la taille de l'entrée.*

La sécurité des protocoles basés sur les couplages repose sur la difficulté du problème du logarithme discret. En effet, au début des années 2000, la cryptographie basée sur les couplages a introduit de nouveaux schémas tels que le chiffrement basé sur l'identité [BF01], la signature basée sur l'identité [CC03], la signature courte [BLS01] ou les schémas de signature aveugle utilisés par exemple dans les enclaves SGX d'Intel (voir le chapitre 7) dont la sécurité est basée sur des hypothèses liées aux couplages qui deviennent fausses si le DLP est cassé. Pour construire un protocole sécurisé basé sur un couplage, on doit donc supposer que les DLP dans les trois groupes $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ sont difficiles.

Une question naturelle découle des descriptions ci-dessus des schémas et du problème du logarithme discret.

Question 91. *Quel groupe G doit-on considérer?*

L'évaluation de la difficulté de DLP est depuis longtemps un domaine de recherche actif. De nombreux algorithmes pour résoudre DLP sont apparus au fil des années. Ces algorithmes varient dans leur construction et leur complexité dépend du groupe G considéré. La cryptographie a renouvelé l'intérêt pour le problème du logarithme discret sur des groupes spécifiques. En pratique, le groupe G dans la définition du problème du logarithme discret est choisi comme étant soit le groupe multiplicatif d'un corps fini $\mathbb{F}_{p^n}$ ou le groupe des points rationnels sur une courbe elliptique $\mathcal{E}$ définie sur un corps fini.

La cryptographie basée sur les couplages illustre la nécessité de considérer le problème de logarithme discret à la fois sur les corps finis et sur les courbes elliptiques. En effet, les groupes considérés pour un couplage cryptographique sont généralement $\mathbb{G}_1$, un sous-groupe de $\mathcal{E}(\mathbb{F}_p)$, le groupe des points d'une courbe elliptique $\mathcal{E}$ définie sur le corps premier $\mathbb{F}_p$, $\mathbb{G}_2$, un autre sous-groupe de $\mathcal{E}(\mathbb{F}_{p^n})$ où l'on considère une extension de corps et $\mathbb{G}_T$ un sous-groupe multiplicatif de ce même corps fini $\mathbb{F}_{p^n}$.

Il est intéressant de noter que le groupe de points rationnels sur une courbe elliptique $\mathcal{E}$ définie sur un corps fini ne fournit aucune représentation utile pour accélérer le calcul d'un logarithme discret. Par conséquent, les meilleurs algorithmes connus pour résoudre DLP sur ce groupe sont les algorithmes génériques avec une complexité en racine carrée.

D'autre part, la prise en compte du problème du logarithme discret dans les corps finis $\mathbb{F}_{p^n}$ a permis d'améliorer considérablement l'efficacité des algorithmes qui le résolvent. La nature et la complexité de ces algorithmes dépendent des caractéristiques du corps fini et plus précisément de la relation entre la caractéristique p et le degré d'extension n. Les algorithmes les plus efficaces pour résoudre DLP dans les corps finis proviennent de la famille des algorithmes de calcul d'indices. Parmi ces algorithmes, nous avons le Function Field Sieve (FFS), le crible algébrique (Number Field Sieve en anglais, NFS) et ses nombreuses variantes, ainsi que les algorithmes plus récents en temps Quasi-Polynomial (QP). Une vue d'ensemble de tous ces algorithmes fait l'objet du Chapitre 1.

En 1994, Shor a introduit un algorithme quantique en temps polynomial pour calculer les logarithmes discrets. Cela implique qu'aucun protocole s'appuyant sur la difficulté de DLP ne serait sûr en présence

d'ordinateurs quantiques, quel que soit le groupe considéré. Cependant, à l'heure actuelle, il n'existe pas d'ordinateurs quantiques capables d'effectuer des calculs à grande échelle, bien que des progrès impressionnants aient été réalisés ces dernières années (voir [AAB+19] pour une machine récente de 53 qubits). Par conséquent, nous limiterons cette thèse à la configuration classique.

Un autre candidat: les réseaux euclidiens

Au cours des dernières décennies, motivés par la menace imminente des ordinateurs quantiques accessibles, de nouveaux candidats prometteurs sont apparus pour construire des protocoles à clé publique : réseaux, isogénies, codes correcteurs d'erreurs, polynômes multivariés et fonctions de hachage. Nous nous concentrons ici sur les réseaux.

L'étude des réseaux en mathématiques a commencé dès le 18^e siècle. Des mathématiciens tels que Gauss, Lagrange et Minkowski ont étudié les réseaux dans le contexte de la géométrie des nombres et de la géométrie convexe, plus particulièrement pour la théorie de la réduction des formes quadratiques qui a ensuite conduit au célèbre algorithme de Gauss. Il faut attendre les années 1980 pour que les réseaux soient étudiés d'un point de vue informatique et utilisés dans des domaines plus proches de l'informatique tels que l'optimisation combinatoire et la cryptographie.

En cryptographie, les réseaux sont utilisés pour la première fois pour casser des cryptosystèmes. Des algorithmes tels que l'algorithme de Lenstra, Lenstra, Lovász (LLL) [LLL82] sont développés pour donner des solutions approximatives aux problèmes de réseaux difficiles et sont largement utilisés pour la cryptanalyse. En 1982, Shamir [Sha82] a utilisé l'algorithme LLL pour casser le système de chiffrement de Merkle-Hellman. En 1996, Coppersmith [Cop96b, Cop96a] a proposé une méthode permettant de factoriser un entier n lorsque certains bits des facteurs de n sont connus, en utilisant des algorithmes de réduction de base de réseaux tels que LLL, affectant ainsi la sécurité du cryptosystème RSA.

L'utilisation de réseaux pour la conception de protocoles cryptographiques n'a commencé qu'en 1996 avec les travaux d'Ajtai [Ajt96]. Les premiers cryptosystèmes à utiliser des réseaux comme blocs de construction sont les cryptosystèmes Ajtai-Dwork [AD97] et NTRU [HPS98] à la fin des années 90. Aujourd'hui, la cryptographie basée sur les réseaux est un domaine de recherche important et de nombreux protocoles basés sur les réseaux sont candidats à la compétition post-quantique du NIST. Dans cette thèse, nous ne considérerons pas les protocoles construits sur la difficulté des problèmes liés aux réseaux. Cependant, nous utiliserons les techniques de réseaux dans deux configurations différentes :

- nous utilisons des algorithmes de réduction de réseaux pour produire des polynômes à petits coefficients, voir Chapitre 3 ou pour trouver les petites racines de polynômes modulaires, voir Partie III.
- nous utilisons des algorithmes d'énumération dand les réseaux, en particulier une adaptation de l'algorithme de Schnorr-Euchner, pour accélérer la recherche de relations algébriques dans le contexte des calculs de logarithmes discrets, voir le chapitre 4.

Des informations préliminaires sur les réseaux et les algorithmes associés sont donc données dans le Chapitre 2.

Remark 26. *Dans cette thèse, nous parlerons de crible pour les réseaux et de cible algébrique. Ces deux notions ne désignent pas la même chose ! Le crible algébrique fait généralement référence à une étape de NFS, tandis que le crible pour les réseaux, tel que le Gauss Sieve, fait référence à un algorithme concernant les réseaux qui trouve des vecteurs courts. Le crible pour les réseaux peut également être trouvé dans le contexte de NFS. Nous précisons quel crible est considéré lorsque le contexte n'est pas clair.*

Contributions

L'objectif de cette thèse est de répondre à la question suivante.

> **Question 92.** *Comment évaluer la sécurité des protocoles dans lesquels une exponentiation modulaire impliquant un secret est effectuée ?*

La réponse à cette question se divise en deux points.

1. La résolution du problème du logarithme discret donne un accès direct à l'exposant, donc au secret. Ainsi, nous voulons estimer la difficulté de DLP dans les groupes considérés par les protocoles.

2. L'étude des vulnérabilités d'implémentation pendant l'exponentiation rapide peut également conduire à l'exposant secret. Ainsi, nous voulons également évaluer et étudier les attaques rendues possibles grâce aux informations fuitées par des canaux auxiliaires.

Ces deux points vont façonner la structure de cette thèse.

Estimation de la difficulté de DLP dans les corps finis

Une façon d'estimer la sécurité des protocoles basés sur la difficulté du problème du logarithme discret est d'étudier directement la complexité des algorithmes qui résolvent ce dernier. Comme nous l'avons mentionné plus haut, cela dépend fortement du groupe considéré. Dans cette thèse, nous nous concentrerons sur l'estimation de la difficulté de DLP pour des corps finis spécifiques.

Question 93. *Sur quels corps finis nous concentrons-nous et pourquoi ?*

Les corps finis $\mathbb{F}_{p^n}$ sont généralement séparés en trois familles appelées petite, moyenne et grande caractéristique, en fonction de la relation entre la caractéristique p et le degré d'extension n du corps fini. À ce jour, les algorithmes connus les plus rapides pour résoudre DLP sont les algorithmes en temps quasi-polynomial pour les corps finis de petite caractéristique [BGJT14, KW19].

Cependant, dans cette thèse, nous nous intéresserons aux corps finis compris entre la frontière entre petite et moyenne caractéristique et la grande caractéristique. En effet, comme ces familles ne fournissent pas l'algorithme connu le plus rapide pour résoudre DLP, elles concernent la plupart des corps finis utilisés en pratique, par exemple dans les protocoles basés sur les couplage. Par conséquent, l'estimation de la difficulté du DLP pour ces familles a un impact significatif sur notre compréhension de la sécurité des protocoles largement déployés.

Notre première motivation concerne la sécurité des protocoles basés sur les couplages. Si nous voulons qu'un couplage soit sûr, nous voulons équilibrer la complexité de l'algorithme en racine carrée qui calcule les logarithmes discrets dans le sous-groupe pertinent de la courbe elliptique considérée, et la complexité de l'algorithme le plus rapide qui résout DLP dans le corps fini. Ceci nous a amené à étudier les algorithmes de la famille du calcul d'indice mentionnés ci-dessus à la frontière entre les corps finis de petite caractéristique et ceux de caractéristique moyenne. La complexité asymptotique de ces algorithmes à ce cas frontière était, jusqu'à cette thèse, inexistante dans la littérature. Cette étude nous a également permis de fournir des points d'intersection précis entre ces nombreuses complexités. Ce sera l'objet du Chapitre 3, illustré par la figure 1. Grâce à cette analyse, nous avons finalement pu fournir des informations supplémentaires sur les paramètres de sécurité des protocoles basés sur les couplages. Plus précisément, ce chapitre répond à la question suivante.

Question 94. *Asymptotiquement, quel corps fini $\mathbb{F}_{p^n}$ devrait être considéré afin d'obtenir le plus haut niveau de sécurité lors de la construction d'un couplage ?*

Dans ce chapitre, nous donnons des valeurs optimales pour la caractéristique p et le degré d'extension n, en prenant également en compte la valeur dite ρ des constructions de couplages. Fait surprenant, nous avons pu distinguer quelques caractéristiques spéciales qui sont asymptotiquement aussi sûres que les caractéristiques de même taille mais sans forme spéciale. L'article suivant résume nos résultats.

1. **Asymptotic complexities of discrete logarithm algorithms in pairing-relevant finite fields**, avec Pierrick Gaudry et Cécile Pierrot, publié dans les actes de la conférence Crypto 2020.

Une autre façon d'obtenir de meilleures estimations de sécurité consiste à réaliser des expériences à grande échelle avec des variantes du Number Field Sieve. En effet, l'algorithme Number Field Sieve a donné lieu à de nombreuses variantes, chacune tentant de réduire la complexité asymptotique de l'algorithme original. L'une de ces variantes est le Tower Number Field Sieve (TNFS). Ce dernier exploite la structure algébrique des tours de corps de nombres. Malgré le fait qu'en théorie la variante est plus

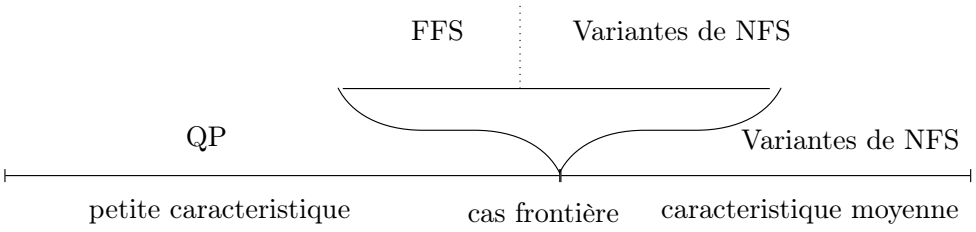

Figure 1: Représentation des corps finis et algorithmes associés étudiés dans le Chapitre 3.

que prometteuse, aucune implémentation et donc aucun calcul record n'avait été fait en utilisant TNFS, jusqu'à cette thèse.

Un obstacle majeur à une mise en œuvre efficace de TNFS est la collection de relations algébriques où des équations entre de petits éléments de corps de nombres doivent être trouvées. Le cas de TNFS est plus complexe que celui de NFS car cette collecte de relations se produit en dimension supérieure à 2. Cela nécessite la construction de nouveaux algorithmes de crible qui restent efficaces lorsque la dimension augmente. Dans le Chapitre 4, nous surmontons cette difficulté en considérant un algorithme d'énumération sur les réseaux que nous adaptons à ce contexte spécifique. Nous considérons également une nouvelle zone de crible, une sphère de haute dimension, alors que les algorithmes de crible précédents (pour les dimensions 2 et 3) considéraient un hyper-parallélépipède.

Cela nous a permis d'effectuer le premier calcul record d'un logarithme discret avec TNFS dans un corps fini de 521 bits $\mathbb{F}_{p^6}$. Le corps fini cible $\mathbb{F}_{p^6}$ choisi est de la même forme que les corps finis utilisés dans les récentes preuves zéro knowledege de certaines blockchains. Ce calcul record a été annoncé en février 2021.

2. **Discrete logarithm in GF(p^6) with Tower NFS** avec Pierrick Gaudry et Cécile Pierrot, annoncé dans liste de diffusion de Théorie des Nombres. Article correspondant en cours de soumission.

Les détails de l'implémentation et du calcul sont donnés dans le Chapitre 5. Comme on peut le voir dans le Tableau 1, notre algorithme est beaucoup plus rapide que les algorithmes de crible en dimension supérieur à deux existants malgré la plus grande dimension et le plus grand corps fini.

Paramètres	[GGMT17]	[MR21]	Cette thèse
Algorithme	NFS	NFS	TNFS
Taille du corps fini (bits)	422	423	521
Dimension de crible	3	3	6
Temps de crible	201,600	69,120	**23,300**

Table 1: Comparison de l'étape de collecte de relations en heures de calcul sur un coeur avec [GGMT17] et [MR21] pour $\mathbb{F}_{p^6}$.

Ces deux travaux contribuent à estimer la difficulté du DLP dans les corps finis en étudiant les complexités asymptotiques des algorithmes pertinents et en fournissant un calcul record avec TNFS. Les considérations faites sur la sécurité des couplages devraient compléter les estimations pratiques trouvées dans la littérature et, espérons-le, orienter les cryptanalystes vers les bons choix de paramètres. Les performances pratiques de TNFS avec notre nouvel algorithme de crible sont prometteuses et indiquent que des corps finis plus grands pourraient être atteints en un temps raisonnable. En général, les calculs records fournissent des indications supplémentaires sur l'écart entre les tailles de clés recommandées pour les protocoles basés sur DLP et ce qui est faisable sur le plan informatique.

Exploitation des vulnérabilités de l'exponentiation rapide

La sécurité des protocoles déployés ne dépend pas seulement de la difficulté du problème mathématique sous-jacent, mais aussi de l'implémentation des algorithmes concernés.

Les implémentations vulnérables de l'exponentiation modulaire rapide ont souvent été la cible d'attaques par canaux auxiliaires où des informations secrètes sont récupérées en créant des liens observables entre les différentes unités d'exécution du CPU. En particulier, les attaques temporelles exploitent les variations du temps d'exécution qui sont courantes dans les algorithmes d'exponentiation modulaire.

Dans le Chapitre 6, nous présentons un aperçu des techniques connues pour récupérer des clés secrètes à partir d'informations partielles. La fuite d'information, généralement un certain nombre de bits d'un élément secret du protocole, est illustrée dans la Figure 2.

Figure 2: Exemple de representation de l'information partielle fuitée par une attaque par canaux auxiliaires.

De nombreuses techniques de récupération de clés secrètes à partir d'informations partielles existent en fonction de la nature de l'information récupérée par l'attaque par canaux auxiliaires et des spécificités de l'algorithme utilisé. Ce chapitre présente les techniques les plus utiles ainsi qu'une classification complète de ce qui est connu pour être efficace pour les scénarios les plus couramment rencontrés dans la pratique. Nous nous concentrons sur les algorithmes largement utilisés qui sont les cibles les plus populaires des attaques, à savoir RSA, (EC)DSA et Diffie-Hellman (ainsi que sa variante avec une courbe elliptique). Nos résultats figurent dans l'article suivant.

3. **Recovering cryptographic keys from partial information, by example**, avec Nadia Heninger. Mis en ligne sur Eprint:Report 2020/1506.

Les techniques présentées dans le Chapitre 6 ont souvent conduit à des attaques réelles sur des protocoles déployés. Dans cette thèse, nous nous concentrons sur deux de ces techniques qui reposent sur des constructions de réseaux : le Hidden Number Problem et le Extended Hidden Number Problem.

Dans le Chapitre 7, nous étudions la sécurité de l'implémentation d'Intel du protocole EPID (Extended Privacy ID), un protocole d'authentification et d'attestation à distance. Nous identifions une faiblesse d'implémentation qui fait fuiter des informations via un canal auxiliaire du cache. Cette fuite d'information nous permet de monter une approche basée sur les réseaux pour résoudre le Hidden Number Problem, que nous adaptons à la preuve zero-knowledge du protocole EPID, étendant ainsi les attaques antérieures sur les systèmes de signature. Ce travail montre qu'un fournisseur d'attestation malveillant peut utiliser l'information divulguée pour briser les garanties de non-liaison d'EPID. Nous fournissons également des preuves expérimentales que l'attaque par réseaux peut toujours réussir même lorsqu'un petit nombre de traces erronées est inclus. Ces résultats sont présentés dans l'article suivant.

4. **CacheQuote: Efficiently Recovering Long-term Secrets of SGX EPID via Cache Attacks**, avec Fergus Dall, Thomas Eisenbarth, Daniel Genkin, Nadia Heninger, Ahmad Moghimi et Yuval Yarom, publié dans le journal IACR Transactions on Cryptographic Hardware and Embedded Systems,Volume 2, 2018.

Nous nous concentrons enfin sur la sécurité du protocole ECDSA lorsque le nonce k utilisé dans l'algorithme de signature comme exposant modulaire est exprimé sous la forme wNAF. Dans le Chapitre 8, nous réétudions la construction du réseau utilisé dans le Extended Hidden Number Problem (EHNP). Nous trouvons la clé secrète avec seulement 3 signatures, atteignant ainsi une limite théorique connue, alors que les meilleures méthodes précédentes nécessitaient au moins 4 signatures en pratique. Étant donné un modèle de fuite spécifique, notre attaque est plus efficace que les attaques précédentes et, dans la plupart des cas, a une meilleure probabilité de succès. Nous fournissons également une première analyse de la résistance aux erreurs de EHNP. Ce travail est décrit dans l'article suivant.

5. **A Tale of Three Signatures: Practical Attack of ECDSA with wNAF**, avec Cécile Pierrot et Rémi Piau, publié dans les actes de la conférence Africacrypt 2020.

En considérant des cibles réelles telles que EPID dans l'architecture d'Intel et l'algorithme ECDSA largement déployé, nous montrons tout au long de ces travaux que même si les bons paramètres sont pris en compte pour que le problème du logarithme discret reste suffisamment difficile à résoudre à des fins cryptographiques, les attaques peuvent provenir d'implémentations vulnérables de l'exponentiation modulaire. Afin d'évaluer réellement la sécurité des protocoles à clé publique déployés, il faut donc considérer simultanément les menaces provenant de la primitive mathématique elle-même et de l'implémentation des algorithmes.

Les contributions et l'organisation de la thèse sont résumées dans la Figure 3.

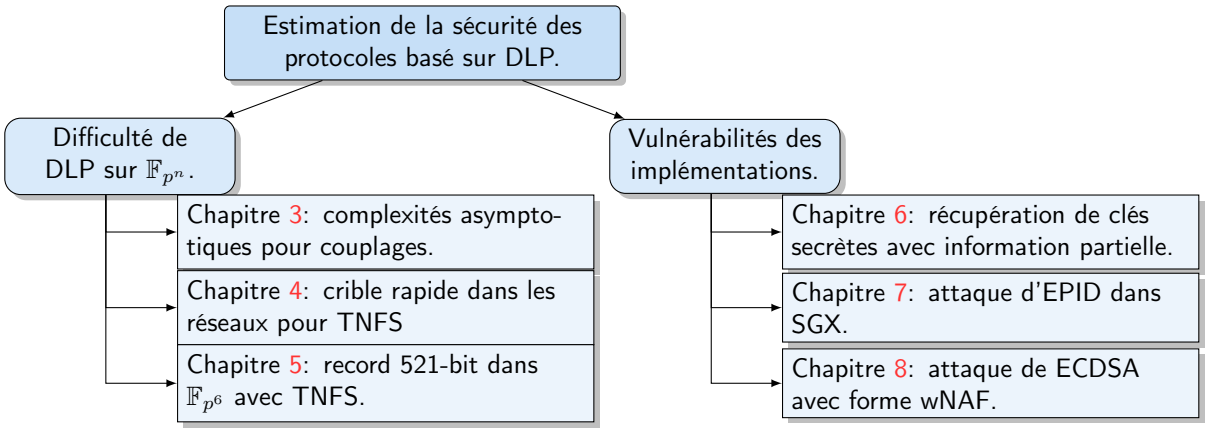

Figure 3: Organisation des contributions de la thèse.

Autre contribution

Overstretched NTRU est une variante de NTRU avec un grand modulo. De récentes attaques de sous-corps et de sous-anneaux de réseaux ont brisé les paramètres suggérés pour plusieurs schémas. Il existe un certain nombre d'affirmations contradictoires dans la littérature sur l'attaque qui présente les meilleures performances. Ces affirmations sont généralement basées sur des expériences plutôt que sur des analyses.

Dans ce travail, nous soutenons que les comparaisons devraient se concentrer sur la dimension du réseau utilisé dans l'attaque. Nous donnons des preuves, à la fois analytiques et expérimentales, que l'attaque par sous-anneaux trouve des vecteurs plus courts et devrait donc réussir avec un réseau de plus petite dimension que l'attaque par sous-corps pour les mêmes paramètres du problème, et également réussir avec un modulo plus petit lorsque la dimension du réseau est fixée.

Comme la thématique de ce travail est en dehors du thème principal de cette thèse, nous ne l'incluons pas dans le manuscrit. L'article suivant résume ces résultats.

6. **Characterizing overstretched NTRU attacks**, avec Nadia Heninger et Barak Shani. Présenté à Mathcrypt 2018, publié dans Journal of Mathematical Cryptology, Volume 14, no. 1, 2020.

Abstract

Public-key cryptosystems are constructed using one-way functions which ensure both the security and the efficiency of the schemes. One of the two main candidates originally considered to construct public-key cryptosystems is modular exponentiation with its hard inverse operation, computing discrete logarithms. In this thesis, we study the security of protocols that make use of modular exponentiation where the exponent is a secret of the protocol. In order to assess the security of such protocols, one can either estimate the hardness of directly solving the discrete logarithm problem (DLP) in the groups considered by the protocols or look at implementation vulnerabilities coming from fast exponentiation algorithms.

One way of estimating the security of protocols based on the hardness of the discrete logarithm problem is to directly study the complexity of the algorithms that solve the latter. In this thesis, we first study the asymptotic complexity of algorithms that solve DLP over finite fields $\mathbb{F}_{p^n}$ precisely of the form where pairings take their values. These algorithms come from the index-calculus family from which the Number Field Sieve (NFS) is an example. This study allows us to draw conclusions on the security of pairing-based protocols. We also propose a first implementation of the variant Tower Number Field Sieve (TNFS) of NFS, which has better asymptotic complexity, along with a record computation of a discrete logarithm in a 521-bit finite field with TNFS. This variant had never been implemented before due to the difficulty of sieving in higher dimensions, *i.e.,* dimensions greater than two.

Finally, the security of deployed protocols not only relies on the hardness of the underlying mathematical problem but also on the implementation of the algorithms involved. Many fast modular exponentiation algorithms have piled up over the years and some implementations have brought forth vulnerabilities that are exploitable by side-channel attacks, in particular cache attacks. The second aspect of this thesis thus considers key recover methods when partial information is recovered from a side channel.

Résumé

Les cryptosystèmes dits à clé publique sont construits à l'aide de fonctions à sens unique qui assurent à la fois la sécurité et l'efficacité des cryptosystèmes. L'un des deux principaux candidats envisagés à l'origine pour construire de tels cryptosystèmes est l'exponentiation modulaire avec son opération inverse, le calcul de logarithmes discrets. Dans cette thèse, nous étudions la sécurité de protocoles qui utilisent des exponentiations modulaires où l'exposant est un secret du protocole. Afin d'évaluer la sécurité de tels protocoles, on peut d'une part estimer la difficulté de résoudre directement le problème du logarithme discret (DLP) dans les groupes considérés par les protocoles, ou examiner les vulnérabilités issues de l'implémentation des algorithmes d'exponentiation rapide.

Une première façon d'estimer la sécurité des protocoles basés sur la difficulté du problème du logarithme discret est d'étudier directement la complexité des algorithmes qui résolvent ce dernier. Dans cette thèse, nous étudions la complexité asymptotique des algorithmes qui résolvent le DLP sur des corps finis $\mathbb{F}_{p^n}$ précisément de la forme où les couplages prennent leurs valeurs.

Nous proposons également une première implémentation et un calcul record d'un logarithme discret dans un corps fini de 521 bits en utilisant l'algorithme Tower Number Field Sieve, une variante de NFS dont la complexité asymptotique est meilleure. Cette variante n'avait jamais été implémentée auparavant en raison de la difficulté du crible algébrique dans des dimensions supérieures à deux.

Enfin, la sécurité des protocoles déployés ne repose pas seulement sur la difficulté du problème mathématique sous-jacent, mais aussi sur l'implémentation des algorithmes considérés. De nombreux algorithmes d'exponentiation modulaire rapide se sont accumulés au fil des ans et certaines implémentations ont fait apparaître des vulnérabilités exploitables par des attaques par canaux auxiliaires. Un second aspect de cette thèse considère donc les principales méthodes pour reconstituer une clé secrète lorsque des informations partielles sont récupérées à partir d'un canal auxiliaire.