Exponentiation modulaire pour la cryptographie à clé publique

Gabrielle De Micheli [1]

Gabrielle De Micheli a soutenu sa thèse [2] le 25 mai 2021 opérée au sein de l'université de Lorraine sous la direction de Pierrick Gaudry et de Cécile Pierrot. Vous trouverez ci-dessous un résumé de cette thèse.

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 [4] ont été reprises dans l'article fondateur de Diffie et Hellman [2], *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 la clé secrète correspondante.

Les cryptosystèmes à clé publique, également connus sous le nom de protocoles asymétriques, sont tous construits en utilisant la notion de fonction trappe. *Cette*

1. gdemicheli@eng.ucsd.edu.

2. https://www.theses.fr/2021LORR0104.

DOI:10.48556/SIF.1024.19.75

terminologie désigne une fonction qui est facile à évaluer pour toute entrée mais dont il est difficile de calculer l'antécédent d'une valeur donnée. 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. Inverser ces fonctions demande de savoir, respectivement, factoriser un entier ou calculer un logarithme discret. La difficulté de la factorisation est au cœur du cryptosystème RSA, bien connu et déployé [5]. Dans ce travail, nous nous sommes concentré 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. 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.

Bien que de nombreux efforts aient été déployés pour optimiser les algorithmes pour l'exponentiation modulaire, ces optimisations ont fait apparaître des vulnérabilités exploitables. 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.

Définition 1 (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.

Définition 2 (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. 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. La sécurité des protocoles basés sur les couplages repose également sur la difficulté du problème du logarithme discret.

Contributions

L'objectif de ce travail est de répondre à la question suivante.

Question 1. *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 utilisés dans les protocoles cryptographiques considérés.

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

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. Cela dépend fortement du groupe considéré. Dans ce travail, nous nous concentrerons sur l'estimation de la difficulté de DLP pour des corps finis spécifiques, situés à un cas frontière. L'estimation de la difficulté du DLP pour ces corps finis 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, des applications bilinéaires non dégénérées. Ces dernières envoient une paire d'éléments, généralement des points sur une courbe elliptique bien définie vers un élément d'un corps finis. 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 à la frontière entre les corps finis dits de petite caractéristique et ceux de caractéristique moyenne. La complexité asymptotique de ces algorithmes à ce cas frontière était, jusqu'à ce travail, inexistante dans la littérature. Grâce à cette analyse, nous avons finalement pu fournir des informations supplémentaires sur les paramètres de sécurité des protocoles basés sur les couplages.

Une autre façon d'obtenir de meilleures estimations de sécurité consiste à réaliser des expériences à grande échelle avec des variantes de l'algorithme Number Field Sieve [3]. 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) [1]. Ce dernier exploite la structure algébrique des tours de corps de nombres. Malgré le fait qu'en théorie la variante est plus que prometteuse, aucune implémentation et donc aucun calcul record n'avait été fait en utilisant TNFS, jusqu'à ce travail.

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.

Nous surmontons cette difficulté en considérant un algorithme d'énumération sur les réseaux que nous adaptons à ce contexte spécifique. 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.

Notre analyse sur la sécurité des protocoles basée sur les couplages ainsi que le calcul record avec TNFS 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.

Nous présentons dans ce travail un aperçu des techniques connues pour récupérer des clés secrètes à partir d'informations partielles ainsi que deux cas pratiques. Nous étudions d'une part 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 nous concentrons également 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. 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.

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.

Références

[1] R. Barbulescu, P. Gaudry, and T. Kleinjung. The tower number field sieve. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 31–55. Springer, 2015.

[2] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theor.*, 22(6) :644–654, Sept. 2006.

[3] A. K. Lenstra, H. W. Lenstra, M. S. Manasse, and J. M. Pollard. The number field sieve. In *The development of the number field sieve*, pages 11–42. Springer, 1993.

[4] R. C. Merkle. Secure Communications over Insecure Channels. *Commun. ACM*, 21(4) :294–299, Apr. 1978.

[5] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM*, 21(2) :120–126, Feb. 1978.