Note sur la cryptanalyse de Diffie-Hellman

Gerard Memmi, Matthieu Rambaud

Note sur la cryptanalyse de Diffie-Hellman

Discussion entre Gerard Memmi et Matthieu Rambaud

LTCI, TelecomParisTech

Résumé : La cryptographie asymétrique permet depuis 40 ans de réaliser l'exploit suivant : deux personnes qui ne connaissent pas se rencontrent dans un lieu public. Elles vont malgré tout réussir, uniquement en parlant à voix haute audible de quiconque, à tenir une conversation comprise d'elles seules.

La méthode de Diffie-Hellman permet à ces deux personnes de convenir d'un mot de passe commun, uniquement en échangeant en public. Elles peuvent en outre vérifier qu'elles parlent à la bonne personne grâce aux techniques de signature électronique. Il sera intéressant (et rassurant) de constater combien l'accélération d'opérations au cœur de la complexité de ces deux algorithmes, favorise bien plus la rapidité de l'échange que sa cryptanalyse.

Mots clés : chiffrement, cryptanalyse, complexité.

1. QUELQUES ÉLÉMENTS DE CONTEXTE

1.1. Méthodes de cryptographie de Diffie-Hellman

Diffie-Hellman est une méthode qui permet à deux personnes à distance de définir (on dit : « échanger ») un secret qu'elles seront les seules à connaître. Autrement dit, de convenir d'un mot de passe commun. Les techniques de « signature électronique » leur permettant de s'assurer mutuellement qu'elles discutent avec la bonne personne.

On appellera ***groupe*** l'ensemble des secrets possibles. On peut donc le voir comme un gros dictionnaire contenant des ***éléments*** (les mots du dictionnaire). ***Diffie-Hellman permet aux deux personnes de définir, par des échanges entendus de tous, un mot parmi ces éléments qui sera leur secret commun.***

Dans un groupe fixé, chacun des éléments est stocké sur un nombre constant de bits d'information (par exemple 128). Ce nombre, qui est le même pour tous les éléments (mais varie d'un dictionnaire à l'autre), s'appelle la ***longueur (en bits) du groupe***.

Cette longueur limite le nombre d'éléments total dans le groupe. En effet avec 128 bits d'information, on ne peut pas dépasser 2^128 éléments. Donc plus elle est grande, plus le groupe peut contenir d'éléments. Pour illustrer on peut voir un groupe comme l'ensemble des possibilités de la combinaison d'un coffre-fort, la longueur du groupe étant le nombre de molettes du cadran.

Mais il existe un paramètre de sécurité au moins aussi important que la longueur du groupe : le ***type du groupe***. Pour continuer l'analogie, les cadrans de certaines marques de coffres-forts sont plus discret au stéthoscope que d'autres. Diffie-Hellman est en pratique surtout utilisée avec deux types de groupes :

- ***les corps finis*** (le premier groupe à avoir été utilisé) ;
- et ***les courbes elliptiques***. (plus discrètes : voir le Tableau 1)

On pourra consulter le livre [Ka] pour une motivation historique des systèmes de chiffrement actuels. Et l'article d'exposition [Vi] pour une introduction aux groupes, corps finis, courbes elliptiques et à la cryptographie asymétrique en général. On pourra également lire un passage de l'article [Br] si l'on veut comprendre comment une courbe elliptique est utilisée comme un groupe [tout le texte situé entre « La méthode précédente, dite de la tangente » et « existence d'un inverse »]

Ces deux types de groupes peuvent chacun être classé en ***deux sous-catégories***, selon que les mots du groupe s'expriment :

- à l'aide de polynômes ***« petite caractéristique »***.
- ou de grands entiers ***« grande caractéristique »***.

D'où 4 sous-catégories au total, qui n'ont pas de rapport entre elles (les éléments des groupes sont des objets mathématiques différents dans chacun des cas). Dans chaque sous-catégorie on trouve une infinité de groupes possibles, de longueurs arbitrairement grandes.

Cryptanalyser l'échange de Diffie-Hellman consiste, pour une tierce personne qui ne ferait qu'écouter les échanges à voix haute (un ***attaquant passif***) , à trouver le secret.

Le tableau suivant montre l'état de la recherche sur la cryptanalyse. L'unité de mesure est le nombre d'opérations de base nécessaires, pour réaliser l'échange, et pour la cryptanalyse. Il s'agit de ***multiplications*** (soit de polynômes soit de grands entiers, suivant la sous-catégorie du groupe). En réalité les techniques de signature électronique (par exemple ECDSA), bien qu'inventées après Diffie-Hellman, reposent sur les mêmes difficultés mathématiques et les mêmes opérations de base.

Modalité	**Corps finis (petite caractéristique)**	**Corps finis (grande caractéristique)**	**Courbes elliptiques (petite et grande caractéristique)**
Échange	b	b	b
Cryptanalyse	$b^{constante}$	$\exp\left(b^{1/3}\right)$	$\exp\left(\frac{1}{2}b\right)$

Tableau 1 : Temps d'échange et de cryptanalyse, en fonction de la longueur b (en bits) du groupe (ordres de grandeur)

[Sources : pour les corps finis, [Adrian 15] et [J15]. Pour les courbes elliptiques, estimation personnelle avec l'algorithme de Pollard.]

On voit :

- que les corps finis, en petite caractéristique, peuvent être cryptanalysés très facilement. Comme l'indique la case en bas à gauche du Tableau 1, le calcul se fait en un temps (quasiment) proportionnel à une puissance de la longueur b (depuis 2013). On dit que la cryptanalyse est « quasi-polynômiale »;

- que les corps finis en grande caractéristique sont plus faciles à cryptanalyser que les courbes elliptiques. ***L'article [Adrian...15] décrit une attaque sur ces corps finis*** *(et recommande en conséquence les courbes elliptiques) ;*

- Que la cryptanalyse des courbes elliptiques est aussi difficile en petite qu'en grande caractéristique. Mais, même si les deux sont toujours recommandées par le NIST, la petite caractérisque fait peur depuis 2013 (voir l'annonce [J13]). En effet il existe une catégorie particulière de courbes elliptiques en petite caractéristique, les « courbes supersingulières » (voir le résumé dans [F12] §5.4.2), qui sont devenues facilement attaquables.

1.2. L'article [Adrian…15]

Cet article observe deux faiblesses logicielles sur internet :

- 3 % des serveurs TLS certifiés sont vulnérables à un type d'attaque qui consiste à (1) se faire passer pour le client (« man in the middle »), et (2) demander au serveur d'abaisser son niveau de sécurité ;

- et, phénomène plus général, 18 % des principaux sites HTTPS utilisent tous exactement le même groupe (un corps fini en particulier). Alors même qu'il existe beaucoup d'autres corps finis de même longueur 1024bits, qui permettraient aussi d'utiliser Diffie-Hellman. Or, un phénomène bien connu est que : ***une fois qu'un échange de Diffie-Hellman exprimé dans un corps fini fixé a été cryptanalysé, tous les autres échanges de Diffie-Hellman effectués dans ce même corps fini deviennent ensuite rapides à cryptanalyser. C'est un exemple d'effet d'« avalanche ».*** C'est pourquoi une seule cryptanalyse dans ce corps fini permettrait d'espionner ces 18 % de sites à la fois[1].

1 Plus généralement, d'après l'introduction de l'article, 10 attaques suffiraient pour cryptanalyser les échanges de « 66% des IKE VPNs, 26% des serveurs SSH, 16% des serveurs SMTP et 24% des sites HTTPS les plus fréquentés ».)

1.3. La multiplication comme base des calculs de complexité

L'unité de mesure absolue est une ***année-cœur***. C'est la quantité de calculs produite en un an par un processeur effectuant 2 Giga opérations élémentaires par seconde sur 64bits (il y a huit cœurs de ce type dans chaque puce Intel Xeon E5-2650, utilisées par les auteurs de [Adrian...15]).

L'unité de mesure utilisée dans le Tableau 2 est ***nombre de multiplications*** nécessaires pour échanger/cryptanalyser. C'est une unité de mesure relative car elle prend un temps différent selon :

- le type et la sous-catégorie (petite ou grande caractéristique) du dictionnaire ;
- la longueur du dictionnaire ;
- la vitesse de l'algorithme pour calculer une multiplication.

Mais c'est une unité de mesure quand même car lorsque ces paramètres sont fixés, la durée consommée est constante (égale à une fraction d'année-cœur). Elle permet donc de mesurer les progrès de la recherche en cryptanalyse et en défense dans une configuration donnée.

1.4. Multiplication de grands entiers (grande caractéristique)

On notera que les techniques de multiplications de grands entiers sont basées sur la multiplication de polynômes [Partons d'un nombre entier, décomposons-le en base B, on obtient un polynôme en puissances de B]. Par exemple la technique la plus connue, Schönhage-Strassen, repose directement sur la multiplication de polynômes par transformée de Fourier rapide. Le texte d'exposition [GS] explique cette technique de façon très courte et élémentaire (il s'agit pourtant de la plus grande avancée du calcul numérique des 40 dernières années). Voir aussi [CT16] et [HHL16] pour des progrès récents.

1.5. Multiplication de polynômes (petite et grande caractéristique)

Deux approches sont possibles :

- Chercher directement à diminuer le temps de calcul d'une multiplication
- Chercher à minimiser le nombre de ***multiplications élémentaires*** utilisées dans une multiplication, en espérant que cela abaissera plus radicalement le temps de calcul total.

D'autre part, dans chaque approche, les calculs typiques consistent en des multiplications sur des mots de petite longueur *b*. Chercher à accélérer ces multiplications de petites longueur *b* est donc aussi un sujet de recherche, qui peut lui aussi être traité avec deux approches :

- chercher directement à diminuer le temps de calcul d'une multiplication de petite longueur *b*
- chercher à minimiser le nombre de multiplications élémentaires utilisées dans une multiplication de petite longueur *b*

Il en résulte quatre axes de recherche :

	***b* grand**	***b* petit**
Vitesse totale	[CNH13], [HHL14]	[B07]
Multiplications élémentaires	[Ran12], [BPRS16]	[BDEZ12], [Ram15]

Tableau 2 : Quatre axes de recherche sur la multiplication des polynômes

L'approche « multiplication élémentaires » bénéficie en pratique à l'approche vitesse totale : cf. par exemple [E13] (*b* petit appliqué à *b* grand) et [ABBR15] (*b* grand appliqué à *b* grand).

2. QUELQUES QUESTIONS

2.1. Quel est le temps nécessaire pour casser une clé sur les corps finis de longueur 512, 768 bits puis 1024bits

L'article [Adrian...15] donne une estimation des temps de calcul (au §4,1) pour cryptanalyser un échange de Diffie-Hellman sur des corps finis, en grande caractéristique : 10 années-cœur pour 512bits, 37000 années-cœur pour 768bits et 45 000 000 années-cœur pour 1024bits. Mais, comme rappelé précédemment, une fois qu'un secret de Diffie-Hellman exprimé dans un corps fini fixé a été cryptanalysé, toutes les autres secrets exprimés dans le même corps fini peuvent être cryptanalysés rapidement.

[Avec le protocole RSA, ce dernier problème ne se pose pas. Mais la cryptanalyse est estimée plus courte (1 200 000 années-cœur pour 1024bits).]

À titre d'illustration, l'article estime que, pour faire une cryptanalyse sur un corps fini fixé en un an, la construction d'un ordinateur dédié aurait coûté 200M$ avec les tarifs de 2012. Cet ordinateur permettrait alors de faire les cryptanalyses suivantes, sur d'autres secrets exprimés dans le même corps fini, en environ 0,1 secondes[2].

Le temps nécessaire, pour faire une cryptanalyse de Diffie-Hellman sur des corps finis de longueur 1024 bits (et la comparaison avec RSA), peut se résumer ainsi :.

	Première attaque	**Attaques suivantes**
Diffie-Hellmann	45 000 000	0,1
RSA	1 200 000	1 200 000

Tableau 3 : Durée des attaques (en années-cœur) sur un corps fini fixé de longueur 1024 bits

2.2. Enjeu d'une accélération de la multiplication, à la fois pour les émetteurs et les attaquants.

On reprend l'exemple ci-dessus de [Adrian...15] relatif à l'échange Diffie-Hellman sur des « corps finis de grande caractéristique ».

- Premièrement, lorsque l'on augmente la longueur des mots, le temps de cryptanalyse d'un secret croît beaucoup plus vite que le temps d'échange. ***Imaginons que l'on passe de groupes de longueur 512bits, à des groupes de longueur 1024bits :***

2.3. *Le temps d'échange est multiplié par 8*

Cette estimation repose sur deux hypothèses :

- que les multiplications sont prédominantes. Le temps nécessaire pour celles-ci croît en pratique selon le carré la longueur (en bits) du groupe. Donc quand la longueur est x2, le temps est x4 ;

- que l'on s'autorise réellement autant de secrets que de mots dans le nouveau groupe[3] , dont la longueur est deux fois plus grande. Le nombre de multiplications nécessaires pour l'échange est alors multiplié par 2 (algorithme d'exponentiation rapide).

2.4. *Le temps de cryptanalyse est multiplié par 4 500 000* (de 10 années-cœur, à 45 000 000 années-cœur).

Faisons maintenant l'hypothèse que l'on réussisse, dans le futur, à diviser par 8 le temps de calcul nécessaire pour une multiplication sur une même machine. Les durées nécessaires pour l'échange et pour la cryptanalyse seraient divisées approximativement par 8[4]. Fixons à 1 unité le temps d'échange initial sur 512 bits. On peut alors résumer (tableau 4) l'évolution des temps de calcul :

2 Source : Table 2 de l'article, colonne Descent, ligne DH-1024, diviser 30 days par les 300 000 cores du Titan supercomputer (paragraphe « costs in hardware »). Puis diviser par 80 (suivant l'hypothèse « if we optimistically... » dans le cas où un matériel dédié serait fabriqué)

3 C'est à dire que l'on s'autorise des exposants aléatoires secrets plus grands que dans un groupe de 512 bits.

4 En pratique le temps pour attaquer Diffie-Hellman sur les corps finis ne diminuerait sans doute pas, car les méthodes d'attaque reposent essentiellement sur l'inversion de matrices creuses.

	Avant amélioration		Après amélioration
	512 bits	1024 bits	1024 bits
Temps d'émission (unité arbitraire)	1	8	1
Temps d'attaque (en années cœur)	10 ans	45 000 000 ans	5 000 000 ans

Tableau 4 : Bilan de l'exemple : impact de l'accélération de la multiplication sur l'échange et la cryptanalyse

Supposons donc que les interlocuteurs décident de passer de 512bits à 1024bits : le temps d'échange de Diffie-Hellman reste identique à celui sur 512bits avant l'accélération. Le temps de cryptanalyse, aura lui, au total, fortement augmenté (il est passé de 10 à 5 000 000 ans).

3. CONCLUSION

On vient de voir sur le tableau précédent qu'une petite amélioration des temps de calcul permet aux interlocuteurs d'ajouter, à temps d'échange égal, des « molettes supplémentaires sur le coffre-fort » (c'est à dire d'utiliser des groupes de plus grande longueur). Et que chaque ajout démultiplie les combinaisons possibles (le nombre d'éléments du groupe), et donc rend la tâche des attaquants exponentiellement plus difficile.

Cependant, la plupart des attaques se basent en fait sur un manque de rigueur dans les étapes qui précèdent la discussion proprement dite. Par exemple, un pirate pourrait faire en sorte que les deux personnes qui échangent choisissent une molette peu discrète au stéthoscope (ce qui peut arriver même sur des groupes de grande longueur). Il reste donc de la responsabilité du client de vérifier la solidité de la courbe elliptique que le serveur lui propose pour faire l'échange (voir les recommandation [An] p21).

En poussant le degré de paranoïa plus loin, on pourrait imaginer que même la poignée de courbes elliptiques recommandées par les agences gouvernementales, possède elle aussi des failles introduites délibérément (comme ce fut le cas de DES dans les années 90). Tout comme pour Diffie-Hellman dans les corps finis, la solution idéale serait donc de générer une courbe elliptique au hasard à chaque échange (voir par exemple §5.2 de [ECDSA] pour des recommandations).

Cette solution, qui n'est pas disponible dans la plupart des matériels actuels, posséderait l'intérêt supplémentaire que, même si une courbe elliptique en particulier se révélait être cryptanalysée facilement, cela éviterait de briser la confidentialité des autres échanges.

Ce qui amène des questions à plus long terme. En effet si les techniques de cryptographie asymétriques semblent robustes depuis 40 ans (Diffie-Hellman, RSA, El-Gamal), il n'en va pas de même de la cryptographie symétrique. Par exemple en moins de 30 ans on a vu apparaître, puis disparaître, les standards DES, MD5 et SHA-1 : voir l'exposé [C-M]. La recherche théorique a heureusement permis d'anticiper ces faiblesses une quinzaine d'années en avance, puis de proposer des solutions. Charge ensuite aux utilisateurs de profiter de ce délai pour changer leur protection (ce que les navigateurs internet n'ont malheureusement pas fait à temps, comme on vient de le voir sur SHA-1 [SHA]).

Dans un objectif de protection des données à plus long terme, la recherche actuelle étudie des techniques résistantes aux attaques quantiques. La cryptographie à base de codes correcteurs fera par exemple partie des techniques proposées au NIST fin 2017.

D'ici-là il existe aujourd'hui d'autres façons de rendre la tâche plus difficile aux attaquants, sans chercher à être exhaustif, voici trois approches à prendre en considération :

- « Trusted computing ». Consiste à utiliser un matériel dédié à l'exécution de tâches sensibles (vérification de signatures électroniques, boot, traçage des accès à un fichier partagé, etc.). Voir les articles sur [TCM] pour des exemples de recherche sur divers types de supports. Le consortium d'industriels [TC] propose ces solutions sur des puces.
- « Partage de secret ». Un secret est fragmenté puis réparti entre plusieurs personnes, de sorte que chacune d'elles (et même un petit groupe) n'a aucune information sur le secret total ; [Voir par exemple

L'exemple présenté conserve cependant l'hypothèse d'une division par 8 du temps d'attaque. Parce que c'est ce qui donne une approche prudente du résultat et aussi parce que c'est une hypothèse pertinente dans l'exemple d'une attaque sur les courbes elliptiques (la seule attaque connue sur ces dernières utilise essentiellement la multiplication).

[GMRW13] pour un protocole où, même en supposant que le serveur qui centralise les calculs serait malhonnête, il ne peut retrouver le secret].

- « Blockchain ». Consiste à faire signer un document par plusieurs personnes (avec une copie datée à chaque modification, etc.). Voir [D15] pour une introduction. En pratique, la certification de grandes blockchains comme le Bitcoin demande une débauche de ressources, et ne peut donc plus être décentralisée auprès de particuliers. La recherche actuelle s'intéresse à d'autres méthodes plus économes : voir par exemple l'article [SpM].

Ces protections supplémentaires ne peuvent, elles non plus, être contournées en se faisant passer pour une personne de confiance. Et c'est une condition indispensable. Car comme le rappelle le principe de Kherkoff (1883), lorsque l'on souhaite protéger des données, le pirate est réputé connaître le système de l'intérieur.

4. RÉFÉRENCES

[Adrian…15] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, B. Vandersloot, E. Wustrow, S. Zanella-Béleguin, and P. Zimmermann "Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice" CCS'15, ACM, Denver, Colorado, October 2015.

[ABBR15] K. Atighehchi, S. Ballet, A. Bonnecaze, R. Rolland. On Chudnovsky-Based Arithmetic Algorithms in Finite Fields (preprint)

[An] Anssi, http://www.ssi.gouv.fr/uploads/2016/09/guide_tls_v1.1.pdf

[BDEZ12] R. Barbulescu, J. Detrey, N. Estibals & P. Zimmermann. "Finding optimal formulae for bilinear maps." WAIFI 2012

[B07] M. Bodrato, Towards Optimal Toom-Cook Multiplication for Univariate and Multivariate Polynomials in Characteristic 2 and 0. WAIFI 2007.

[BPRS16] S. Ballet, J. Pieltant, M. Rambaud, J. Sijsling. On some bounds for symmetric tensor rank of multiplication in finite fields. AGCT 2015 (à paraître)

[Br] F. Brunault, Le rang des courbes elliptiques. Images des mathématiques (http://images.math.cnrs.fr/Le-rang-des-courbes-elliptiques.html?lang=fr) 2012

[C-M] A. Canteau, M. Minier. De l'espérance de vie d'un algorithme symétrique (ou l'AES dix ans après). MISC, Hors-Série 5 (copies disponibles en ligne), 2012.

[CNH13] M. Cenk, C. Negre, A. Hasan. Improved Three-Way Split Formulas for Binary Polynomial and Toeplitz Matrix Vector Products. IEEE Trans. Computers 2013

[CT16] S. Covanov et E. Thomé. Fast arithmetic for faster integer multiplication (preprint)

[D15] J. P Delahaye, Les blockchains, clefs d'un nouveau monde. Pour la Science, Mars 2015, http://www.lifl.fr/~jdelahay/pls/2015/256.pdf

[E13] N. Estibals. Algorithmes et arithmétique pour l'implémentation de couplages cryptographiques. Thèse, Université de Lorraine. 2013

[ECDSA] The Elliptic Curve Digital Signature Algorithm (ECDSA). Johnson, Menezes, Vanstone. 2001

[F12] J.-P. Flori, Fonctions booléennes, courbes algébriques et multiplication complexe. Thèse, Télécom ParisTech 2012.

[GMRW13] S. Dov Gordon and Tal Malkin and Mike Rosulek and Hoeteck Wee, Multi-Party Computation of Polynomials and Branching Programs without Simultaneous Interaction. EUROCRYPT 2013.

[GS] X. Gourdon – Claude Sebah, FFT Based multiplication of large numbers. Disponible sur http://numbers.computation.free.fr/Constants/constants.html

[HHL14] D. Harvey, J. van der Hoeven, G. Lecerf. Faster polynomial multiplication over Finite Fields (preprint)

[HHL16] D. Harvey, J. van der Hoeven, G. Lecerf. Even faster integer multiplication. J. Complexity, 2016

[J15] A. Joux. Gazette de la SMF, Avril 2015.

[J13] A. Joux. https://listserv.nodak.edu/cgi-bin/wa.exe?A2=NMBRTHRY;49bb494e.1305

[Ka] D. Kahn, « the Code Breakers" the Comprehensive History of Secret Communication from Ancient Times to the Internet. Scribner, 1996.

[Ran12] H. Randriambololona. Bilinear complexity of algebras and the Chudnovsky-Chudnovsky interpolation method. Journal of Complexity

[Ram15] M. Rambaud, Finding optimal Chudnovsky-Chudnovsky multiplication algorithms. WAIFI 2014

[SHA] M. Stevens, Bursztein, Karpman, Albertini, Markov. The first collision for full SHA-1, https://shattered.io/static/shattered.pdf , 2017

[SpM] Sunoo Park and Krzysztof Pietrzak and Albert Kwon and Joël Alwen and Georg Fuchsbauer and Peter Gaži. SpaceMint: A Cryptocurrency Based on Proofs of Space. Preprint 2016

[TC] https://trustedcomputinggroup.org/

[TCM] Trusted Computing, page du groupe de recherche du MIT : http://projects.csail.mit.edu/tc/

[Vi] V. Vitse, Des corps, des courbes, des couplages, des messages codés… et des logarithmes discrets. Images des mathématiques (http://images.math.cnrs.fr/Des-corps-des-courbes-des-2407?lang=fr) 2013