Sujets de TER 2014–2015

Janvier 2015

Sujets proposé par *Emmanuel Fleury.*

1. **Études (et implémentation) d'attaques contre SSL/TLS**
   Martial Puygrenier, Kevin Grahek, Rémi Tremblain
   Le but de ce projet est d'étudier et d'implémenter un outil permettant de réaliser les diverses attaques possibles sur les différentes versions de SSL/TLS.
   Nous étudierons les attaques suivantes et nous essayerons de toutes les implémenter dans un outil :
   – POODLE attack
   – BEAST attack
   – Heartbleed attack
   – Padding Oracle attack
   Voir : http://en.wikipedia.org/wiki/Transport_Layer_Security#Attacks_against_TLS.2FSSL
   Il faudra ensuite pouvoir mettre en place différentes versions de serveurs SSL/TLS pour essayer l'outil.
   Le choix du langage pour réaliser l'outil est laissé libre (Python, Java, C, ...).
2. **Implémentation et Analyse d'une White-box d'AES**
   Amrouche, Planchon
   La «white-box crypto» correspond à produire une implémentation obfusquée d'un algorithme de chiffrement de telle manière que certains paramètres secrets de l'algorithme (la clef, les S-box, ...) soient difficiles à retrouver même si le programme s'exécute sur une plate-forme non sûre (e.g. sur l'ordinateur du client).
   Le projet consistera à lire l'article «A Tutorial on White-box AES» de James A. Muir (http://eprint.iacr.org/2013/104.pdf), à comprendre les différentes méthodes proposées et à les implémenter dans un programme C aussi efficace que possible.

S'il reste du temps, il s'agira de voir comment «casser» ces protections.

Sujets proposés par *Jean-Marc Couveignes*.

1. **Fonctions de hachage, codes d'authentification, arithmétique rapide.**
   Sous certaines conditions, une fonction de hachage permet d'assurer l'intégrité de données : une modification de ces données invalide le certificat. On propose d'étudier cette question plus en détail. Si le temps le permet, on abordera la construction proposée par Wegman et Carter et le rôle joué par la réduction modulo un entier ou un polynôme.
   Douglas Stinson, Cryptographie, théorie et pratique, chapitre 4, Vuibert.
   J. Lawrence Carter, Mark N. Wegman, Universal classes of hash functions, Journal of Com- puter and System Sciences 18 (1979), 143–154. ISSN 0022–0000. http://cr.yp.to/bib/1979/carter.html
   Daniel J. Bernstein. "Floating-point arithmetic and message authentication." http://cr.yp.to/antiforgery/hash127-20040918.pdf
2. **Réduction de réseaux et cryptographie.**
   La réduction de réseaux est un outil puissant pour la cryptanalyse. On propose d'étudier les cryptosystèmes à base de sac-à-dos et leur attaque à l'aide de l'algorithme LLL de réduction de réseaux.
   The rise and fall of knapsack cryptosystems by Andrew Odlyzko, http://www.dtc.umn.edu/~odlyzko/doc/arch/knapsack.survey.pdf
   The Two Faces of Lattices in Cryptology by Phong Q. Nguyen and Jacques Stern Cryptography and Lattices – Proceedings of CALC '01 (March 29–30, 2001, Providence, Rhode Island, USA), J. Silverman (Ed.), vol. 2146 of Lecture Notes in Computer Science, Springer-Verlag.

Sujets proposés par *Karim Belabas*. un sujet parmi les suivants.

1. Implantation d'un algorithme de division rapide dans $\mathbb{Z}$ ou sur $\mathbb{F}_p[X]$ («Karatsuba», Newton, RMP). Éventuellement, autres algorithmes en multiprécision. [5]
2. Implantation et analyse d'algorithmes asymptotiquement rapides en algèbre linéaire sur un corps K (Strassen, Wiedemann, Lanczos...). On pourra éventuellement se limiter à $K = \mathbb{F}_2$. Référence [4] Chap. 12.
3. Implantation et analyse d'algorithmes d'interpolation (lemme Chinois) et de multi-évaluation. Applications au calcul modulaire, par exemple déterminant dans $M_n(\mathbb{Z})$ et PGCD dans $\mathbb{Q}[X]$. Référence [4] Chap. 5,6,10.
4. Implantation et analyse d'algorithmes de factorisation de polynômes sur un corps fini (Berlekamp, Frobenius itéré...).
   Foune, Diadhiou

5. Implantation et analyse du crible quadratique (QS) pour la factorisation des entiers. Référence [1,4] ci-dessus.

La programmation de ces algorithmes de (relativement) bas niveau sera plus efficace en C et l'aide d'une bibliothèque adaptée comme PARI [3] ou GMP [2], mais elle est digne d'intérêt en tout langage raisonablement riche. Suivant le langage de programmation choisi, on adaptera les buts du projet.

Références

[1] H. Cohen, *A course in computational algebraic number theory*, third ed., Springer-Verlag, 1996.

[2] T. Granlund, GMP : GNU Multi Precision library, http://swox.com/gmp/.

[3] PARI/GP, version 2.4.3, Bordeaux, 2008, http://pari.math.u-bordeaux.fr/.

[4] J. von zur Gathen & J. Gerhard, *Modern computer algebra*, Cambridge University Press, New York, 1999.

[5] P. Zimmermann, Arithmétique en précision arbitraire, http://www.loria.fr/~zimmerma/papers/RR4272.ps.gz.

Sujet proposé par *Jean-Paul Cerri*

Génération de polynômes irréductibles et factorisation des polynômes dans $K[X]$ où K est un corps fini.

Sujet proposé par *Pascal Autissier*

Le test de primalité AKS.

Caixa Lu, Dalsheimer, Giraud.

Il s'agit de comprendre et de programmer l'algorithme d'Agrawal-Kayal-Saxena, qui détermine en temps polynomial si un entier donné est premier ou composé. Il est basé sur le résultat suivant : Soient $n > 1$ et a deux entiers premiers entre eux. Alors n est premier si et seulement si on a la congruence $(X + a)^n = X^n + a \mod n$.

Sujets proposés par *Guilhem Castagnos*.

Complet.

1. **Attaques sur les primitives Elgamal et RSA**
   Alain Tran et Jennifer Fumont
   Il est bien connu que les primitives de chiffrement Elgamal et RSA, c'est à dire sans transformation préalable du message à chiffrer, ne peuvent at-

teindre les plus hauts niveaux de sécurité, par exemple ceux ou un adversaire a accès à un oracle de déchiffrement. En 2000, Boneh, Joux et Nguyen ont proposé des attaques sur ces primitives dans le cas spécifique où elles sont utilisées pour chiffrer un message court, par exemple une clef de session de cryptographie symétrique. Il s'agira d'étudier et de mettre en œuvre ces attaques.

Lien :
Boneh, Joux et Nguyen, Why textbook ElGamal and RSA encryption are insecure, http://www.ssi.gouv.fr/archive/fr/sciences/fichiers/lcr/bojong00.pdf

2. **Protocole d'interrogation anonyme de base de données**
   Baudouin Duthoit et Gaëtan Pradel
   D'ordinaire, pour récupérer un champ d'une base de données, un client envoie une requête indiquant quel élément l'intéresse, puis la base lui renvoie cet élément. Quel élément a été consulté est une information que le client peut ne pas souhaiter divulguer y compris au serveur de base de données. Pour protéger l'anonymat de la requête, des protocoles, appelés protocoles PIR (*Private Information Retrieval*) ont été proposés. Il s'agira d'étudier et de mettre en œuvre une construction de protocole PIR utilisant des systèmes de chiffrement dits homomorphes additifs, comme le système de Paillier.

   Liens :
   Rafail Ostrovsky, William E. Skeith III, A Survey of Single-Database PIR : Techniques and Applications, http://eprint.iacr.org/2007/059.pdf
   Pascal Paillier, Public Key Cryptosystems based on Composite Degree Residue Classes, http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.112.4035

3. **Algorithme ρ de Pollard pour le logarithme discret**
   Les systèmes cryptographiques basés sur le problème du logarithme discret tels que le chiffrement Elgamal ou la signature DSA peuvent utiliser un sous groupe multiplicatif d'ordre q premier d'un corps fini $\mathbf{F}_p$ avec p premier. Dans un tel cadre, la complexité de l'algorithme le plus performant pour le problème du logarithme discret, le calcul d'index, dépend de la taille de p. Pour l'algorithme ρ de Pollard, c'est la taille de q qui compte. Il s'agira d'étudier et d'implanter cet algorithme dans ce cadre. On pourra notamment s'intéresser à des méthodes récentes pour accélérer la méthode de Pollard.

Liens :

Jung Hee Cheon, Jin Hong, and Minkyu Kim, Speeding Up the Pollard Rho Method on Prime Fields, http://www.math.snu.ac.kr/~jhcheon/publications/2008/TTDLP_A08_CheonHongKim.pdf

4. **Générateur pseudoaléatoire de type FCSR filtré**
   Il s'agira d'étudier et d'implanter un générateur pseudoaléatoire, basé sur un registre FCSR (*Feedback with Carry Shift Register*) associé à une fonction de filtrage, proposé par Arnault, Berger et autres, ainsi qu'une attaque dévastatrice sur ce schéma due à Hell et Johansson.

   Liens :

   La soumission de l'algorithme F-FCSR au projet eSTREAM : http://www.ecrypt.eu.org/stream/ffcsrpf.html

   Hell, Johansson, Breaking the F-FCSR-H Stream Cipher in Real Time, https://www.iacr.org/archive/asiacrypt2008/53500563/53500563.pdf

Sujet proposé par *Christine Bachoc*.

L'attaque de Sidelnikov et Shestakov sur le cryptosystème de Niederreiter basé sur les codes de Reed-Solomon généralisés.

Pauline Bert, Thomas Clédel, Aurélie Phesso

Le cryptosystème de Niederreiter est un système à clé publique basé sur la difficulté du décodage par syndrome d'un code linéaire. La clé publique est une matrice de parité H d'un code linéaire sur $\mathbb{F}_q$, et la clé privée est une paire de matrices (K, P) telles que K est inversible, P est une matrice de permutations, et $H' = KHP$ est une matrice de parité d'un code pour lequel il existe un algorithme efficace de décodage par syndrome. Dans le cas des codes de Reed-Solomon généralisés, Sidelnikov et Shestakov ont donné un algorithme polynomial qui permet de reconstituter (K, P) et donc H' à partir de H. Le but du projet est de comprendre est d'implémenter cet algorithme à partir de l'article original :

V.M. Sidelnikov and S.O. Shestakov On insecurity of cryptosystems based on generalized Reed-Solomon codes Discrete Math. Appl. Vol 2 (4) (1992).

Sujets proposés par *Gilles Zémor*.

1. **Décodage par ensemble d'information**
   Il s'agit d'une méthode générique de décodage d'un code linéaire qui peut être assez efficace pour des longueurs qui restent raisonnables. Le principe est de «deviner» un ensemble de k coordonnées sans erreur. Si on y arrive, il suffit de réencoder pour trouver le mot de code correct et localiser les erreurs. Bien sûr, il faut essayer plusieurs fois avant de trouver un tel

ensemble d'information, mais un calcul simple montre que si $k = n/2$, le nombre moyen de tentatives pour corriger t erreurs est 2^t, ce qui reste praticable pour t petit. Il s'agira d'implémenter ce décodage et de faire quelques expériences avec.

2. **Les codes polaires**

   Il s'agit de s'initier à la stratégie de codage correcteur d'Arikan introduite en 2009, et qui, soixante ans après Shannon, permet enfin d'atteindre la capacité du canal binaire symétrique (et de nombreux autres canaux) avec une complexité praticable. On étudiera les codes polaires dans le cas simple du canal à effacements et pratiquera une implémentation.