Cryptanalyse algébrique de fonctions de hachage

Stage M2 MPRI

Luk Bettale

équipe SALSA
LIP6, Université Paris 6 & INRIA Paris-Rocquencourt
encadré par Jean-Charles Faugère et Ludovic Perret

30 septembre 2008

Cryptanalyse algébrique

Cryptanalyse algébrique (1)

Objectif

Analyse de la sécurité – Problème fondamental en cryptologie

Démarche en 2 étapes

1. Mise en équation sous forme d'un système algébrique
2. Résolution du système (ou à défaut estimation de la difficulté)

Stratégie :
Minimiser le nombre de variables et le degré des équations

Cryptanalyse algébrique (2)

Outils puissants pour résoudre des systèmes.
$\Rightarrow$ base de Gröbner

Jean-Charles Faugère.

A new efficient algorithm for computing Gröbner bases (F4).

Journal of Pure and Applied Algebra, 139 pages 61–88, June 1999.

Jean-Charles Faugère.

A new efficient algorithm for computing Gröbner bases without reduction to zero (F5).

Proceedings of ISSAC 2002, pages 75–83. ACM Press, July 2002.

Fonctions de hachage – Définition

Fonction qui prend en entrée un message de longueur quelconque et produit en sortie une empreinte de longueur n :

$$h : \mathcal{A}^* \rightarrow \mathcal{A}^n$$

Calcul de l'empreinte d'une suite de bits : $\mathcal{A} = \{0, 1\}$

- Assurer l'intégrité d'un message
- Protection de mots de passe
- Signature électronique
- Génération pseudo-aléatoire

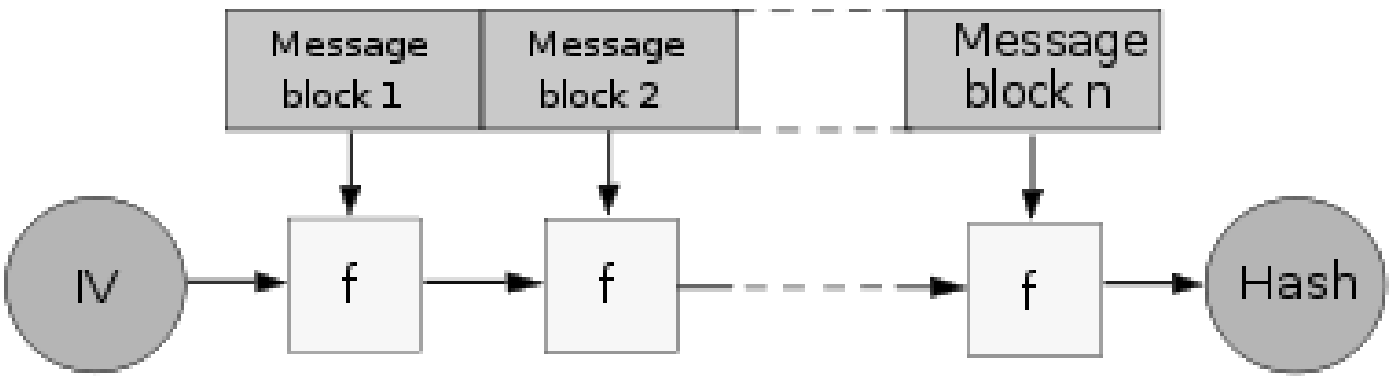
Fonctions de hachage – Sécurité

$h : \mathcal{A}^* \rightarrow \mathcal{A}^n$

- Préimage :
  Soit z une empreinte, trouver x tel que $h(x) = z$
  complexité : $O(2^n)$
- Seconde préimage :
  Soit x un message, trouver $x' \neq x$ tel que $h(x) = h(x')$
  complexité : $O(2^n)$
- Collision :
  Trouver un couple (x, x') tel que $x' \neq x$ et $h(x) = h(x')$
  complexité : $O(2^{n/2})$

SHA-1 – Présentation (1)

- Empreinte de 160 bits
- Concept similaire à MD5 [1]
- Utilise la construction de Merkle-Damgård



$$f : \mathbb{F}_2^{160} \times \mathbb{F}_2^{512} \longrightarrow \mathbb{F}_2^{160}$$

[1] Ronald Rivest en 1991

SHA-1 – Présentation (2)

Opérations utilisées

- Opérations booléennes $(\wedge, \vee, \oplus)$
- Additions modulo 2^{32} (mots machine)

$f : \mathbb{F}_2^{160} \times \mathbb{F}_2^{512} \longrightarrow \mathbb{F}_2^{160}$

pour $i \in \{16, \ldots, 80\}$:

$m_i = (m_{i-3} \oplus m_{i-8} \oplus m_{i-14} \oplus m_{i-16})$

pour $i \in \{1, \ldots, 80\}$:

$a_i = (a_{i-1} \ll 5) + f_i(b_{i-1}, c_{i-1}, d_{i-1}) + e_{i-1} + m_i + k_i$

$b_i = a_{i-1}$

$c_i = (b_{i-1} \ll 30)$

$d_i = c_{i-1}$

$e_i = d_{i-1}$

fonction f_i :

$$\begin{aligned} \text{IF} &: (x \wedge y) \vee (\neg x \wedge z) \\ \text{XOR} &: x \oplus y \oplus z \\ \text{MAJ} &: (x \wedge y) \vee (x \wedge z) \vee (y \wedge z) \end{aligned}$$

SHA-1 – Modélisation (1)

On modélise la fonction f par un système de polynômes à coefficients dans $\mathbb{F}_2$:

- Équations linéaires $\Rightarrow$ pas d'ajout de variables
- Variables intermédiaires pour avoir des polynômes de bas degré.

SHA-1 – Modélisation (2)

Comment peut-on modéliser l'addition modulo 2^{32} avec des opérations sur des éléments dans $\mathbb{F}_2$?

$$\begin{cases} z_0 = x_0 + y_0 \\ z_1 = x_1 + y_1 + (x_0.y_0) \\ z_2 = x_2 + y_2 + (x_1.y_1) + (x_1 + y_1).(x_1 + y_1 + z_1) \\ \vdots \\ z_n = x_n + y_n + (x_{n-1}.y_{n-1}) + (x_{n-1} + y_{n-1}).(x_{n-1} + y_{n-1} + z_{n-1}) \end{cases}$$

Blandine Debraize.
Méthodes de cryptanalyse pour les schémas de chiffrement symétrique.
Thèse – Université de Versailles St-Quentin

SHA-1 – Modélisation (3)

On peut modéliser la fonction de compression de SHA-1 avec

$$160 + 512 + 32 * 4 * 80 = 10912 \text{ variables}$$

On peut réduire à 10752 en fixant les variables initiales (on ne travaille que sur un bloc de message)

- Toutes les équations sont de degré au plus 3
- Pour un bon ordre sur les monômes, le système est directement une base de Gröbner !

SHA-1 – Attaque algébrique

- Plate-forme logicielle pour d'autres attaques
- Attaque "manuelle" proposée par Sugita
- Combiner les attaques différentielles de Wang avec une approche algébrique

Makoto Sugita, Mitsuru Kawazoe, Ludovic Perret, and Hideki Imai.
Algebraic cryptanalysis of 58-round SHA-1.
In Alex Biryukov, editor, *FSE*, volume 4593 of *LNCS*, pages 349–365. Springer, 2007.

Fonctions de hachage multivariées – Présentation

Une famille de fonctions basée sur la difficulté de résoudre un système polynomial (problème NP-complet)

- Utilise aussi Merkle-Damgård
- Fonction de compression $\Leftrightarrow$ Système polynomial
- Soit $\mathbb{K} = \mathbb{F}_q$, $f : \mathbb{K}^{m+n} \rightarrow \mathbb{K}^m$
- m équations pour $m + n$ variables

Sécurité "prouvée" pour les attaques en préimage et collision.

Fonctions de hachage multivariées – Construction

Trois constructions ont été proposées par Ding et Yang :

- Polynômes *cubiques* "dense"
- Polynômes *cubiques* "creux" (ϵ le pourcentage de monômes)
- Composition de 2 systèmes quadratiques (Polynômes quartiques)

Soit $\mathbb{K} = \mathbb{F}_q$, $f : \mathbb{K}^{m+n} \longrightarrow \mathbb{K}^m$

$$\begin{cases} f_1(y_1, \ldots, y_m, x_1, \ldots, x_n) \\ \qquad\qquad \vdots \\ f_m(a_1, \ldots, a_m, x_1, \ldots, x_n) \end{cases}$$

Jintai Ding and Bo-Yin Yang.

Multivariates polynomials for hashing.

Cryptology ePrint Archive, Report 2007/137, 2007.

Fonctions de hachage multivariées – Attaque (1)

Attaque en collision

1. choisir une différence δ aléatoirement
2. construire le système $f' = f(y, x + \delta) - f(y, x) = 0$, et fixer les valeurs de y aux valeurs initiales.
3. calculer les solutions de f'
4. si on trouve une solution, on a trouvé une collision, sinon revenir à l'étape 1

Fonctions de hachage multivariées – Attaque (2)

$(a_1, \ldots, a_m) \in \mathbb{K}^m$,
$(\delta_1, \ldots, \delta_n), (x_1, \ldots, x_n) \in \mathbb{K}^n$

On doit résoudre le système :

$$\begin{cases} f_1(a_1, \ldots, a_m, x_1, \ldots, x_n) - f_1(a_1, \ldots, a_m, x_1 + \delta_1, \ldots, x_n + \delta_n) = 0 \\ \vdots \\ f_m(a_1, \ldots, a_m, x_1, \ldots, x_n) - f_m(a_1, \ldots, a_m, x_1 + \delta_1, \ldots, x_n + \delta_n) = 0 \end{cases}$$

On fait diminuer de 1 le degré total des polynômes du système
$\Rightarrow$ On obtient un système plus facile à résoudre.

Fonctions de hachage multivariées – Attaque (3)

Approche hybride

- On spécialise k variables du système
- On résout des systèmes surdéterminés (plus facile)
- On doit calculer au total $\#\mathbb{K}^k$ bases de Gröbner

Les systèmes générés se comportent comme des *séquences semi-régulières*
$\Rightarrow$ complexité théorique de l'approche

Luk Bettale, Jean-Charles Faugère, and Ludovic Perret.
Cryptanalysis of the TRMS signature scheme of PKC'05.
In *Progress in Cryptology – AFRICACRYPT 2008*, volume 5023 of *LNCS*, pages 143–155. Springer, 2008.

Magali Bardet, Jean-Charles Faugère, and Bruno Salvy.
On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations.
In *Proc. International Conference on Polynomial System Solving (ICPSS)*, pages 71–75, 2004.

Fonctions de hachage multivariées
Construction dense

Attaque en collision

$\#\mathbb{K}$	n	$n-k$	k	T_{F_5}	$\mathrm{Nop}_{\mathrm{F}_5}$	N	$\mathrm{N}_{\mathrm{gen}}$
2^{16}	16	15	1	≈ 1 h.	$2^{36.9}$	$2^{52.9}$	2^{128}
		14	2	126 s.	$2^{32.3}$	$2^{64.3}$	
2^{8}	20	18	2	51 h.	2^{41}	2^{57}	2^{80}
		17	3	2h45min.	2^{37}	2^{61}	
		16	4	643.1 s.	2^{34}	2^{66}	

FIG.: Temps de résolution et complexité

Fonctions de hachage multivariées
Construction creuse (1)

- Les systèmes creux semblent plus faciles à résoudre
- Moins de variables à fixer
- Stratégie $\Rightarrow$ Choisir δ de faible poids de Hamming

Fonctions de hachage multivariées
Construction creuse (2)

Attaque en collision

paramètres	$w(\delta)$	temps min/max		prob
$q = 2^8, n = 20, \epsilon = 0.2\%$	4	0.5 s.	48 h.	1/4
$q = 2^{16}, n = 16, \epsilon = 0.2\%$	5	0.1 s.	311.9 s.	1/3
$q = 2^8, n = 32, \epsilon = 0.1\%$	2	0.4 s.	690.3 s.	1/15

FIG.: Temps de résolution et probabilité


Luk Bettale, Jean-Charles Faugère, and Ludovic Perret.
Cryptanalysis of multivariate polynomials for hashing.
submitted to INSCRYPT 2008, 2008.

Conclusion

Ce qui a été fait :

- Plate-forme logicielle pour attaques algébriques sur SHA-1 (Magma)
- Analyse de sécurité de fonctions de hachage multivariées

Ce qui reste à faire :

- Analyse algébrique systèmatique d'autres fonctions de hachage
- Complexité des systèmes creux