Attaques algébriques sur des systèmes sous-déterminés issus de la cryptographie

Luk Bettale

équipe SALSA
LIP6, Université Paris 6 & INRIA Paris-Rocquencourt

19 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)

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.

Bases de Gröbner (1)

Définition

$G \subset \mathbb{K}[x_1, \ldots, x_n]$ est une base de Gröbner d'un idéal polynomial $\mathcal{I}$ si :

$$\forall f \in \mathcal{I}, \exists g \in G \text{ tel que } LM(g) \text{ divise } LM(f)$$

Propriété

Une base de Gröbner pour l'ordre lexicographique d'un système **zero-dimensionnel** *est de la forme :*

$$\{g_1(x_1), g_2(x_1, x_2), \ldots, g_{k_2}(x_1, x_2), \ldots, g_{k_2+1}(x_1, x_2, x_3), \ldots\}$$

Bases de Gröbner (2)

Algorithmes

- Buchberger : l'algorithme historique
- $\mathrm{F4}$: Algèbre linéaire sur des matrices
- $\mathrm{F5}$: Pas de réduction à zéro pour des séquences semi-régulières

pour $\mathrm{F5}$, complexité en $\mathcal{O}\left(\left(m \cdot \mathrm{C}_{n+d_{\mathrm{reg}}-1}^{d_{\mathrm{reg}}}\right)^{\omega}\right)$

Séquences semi-régulières

Soit $p_1, \ldots, p_m \in \mathbb{K}[x_1, \ldots, x_n]$ une séquence de polynômes homogènes de degré $d_1, \ldots, d_m$. On dit que c'est une **séquence semi-régulière** si

- $\langle p_1, \ldots, p_m \rangle \neq \mathbb{K}[x_1, \ldots, x_n]$
- quelque soit $f \in \mathbb{K}[x_1, \ldots, x_n]$ et pour tout $1 \leq i \leq m$:
  $f \cdot p_i \in \langle p_1, \ldots, p_{i-1} \rangle$ et
  $deg(f \cdot p_i) \leq d_{reg}(p_1, \ldots, p_{i-1}) \Rightarrow f \in \langle p_1, \ldots, p_{i-1} \rangle$

Cryptographie multivariée

Principe

- Utiliser des polynômes à plusieurs variables
- $F : \mathbb{K}^n \to \mathbb{K}^m$
  $(x_1, \ldots, x_n) \to (f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n))$
- Cryptographie asymétrique : $P = T \circ F \circ S$ (clé publique)
  (C^*, HFE, SFLASH,...)

Intérêts

- Pas d'algorithme en temps polynômial (problème NP-dur)...
- ...même pour les ordinateurs quantiques

Signature numérique

Vérifier l'identité de l'expéditeur d'un message.

paire de clé (sk, pk)

$$Sign_{sk} : \mathcal{A}^* \to \mathcal{A}^m$$

$$Verify_{pk} : \mathcal{A}^* \times \mathcal{A}^m \to \{\textbf{true}, \textbf{false}\}$$

On veut qu'il soit impossible de forger une signature valide.

TRMS – Présentation (1)

Tractable Rational Map Signature

- Schéma de signature (asymétrique)
- Masquer un système polynomial
- Algorithme de signature non-déterministe

Lih-Chung Wang and Yuh-Hua Hu and Feipei Lai and Chun-Yen Chou and Bo-Yin Yang.
Tractable Rational Map Signature.
PKC '05

TRMS – Présentation (2)

"Transformation Rationnelle Malléable"

Pour tous $i \in \{1, \ldots, n\}$, $p_i, q_i, f_i, g_i \in \mathbb{K}[x_1, \ldots, x_n]$ et, r_i permutation de $\mathbb{K}$ dans $\mathbb{K}$

$$
\begin{cases}
y_1 &= r_1(x_1) \\
y_2 &= r_2(x_2)\frac{p_2(x_1)}{q_2(x_1)} + \frac{f_2(x_1)}{g_2(x_1)} \\
&\vdots \\
y_k &= r_k(x_k)\frac{p_k(x_1,\ldots,x_{k-1})}{q_k(x_1,\ldots,x_{k-1})} + \frac{f_k(x_1,\ldots,x_{k-1})}{g_k(x_1,\ldots,x_{k-1})} \\
&\vdots \\
y_n &= r_n(x_n)\frac{p_n(x_1,\ldots,x_{n-1})}{q_n(x_1,\ldots,x_{n-1})} + \frac{f_n(x_1,\ldots,x_{n-1})}{g_n(x_1,\ldots,x_{n-1})}
\end{cases}
$$

TRMS – Présentation (3)

“Transformation Rationnelle Malléable”

Inversion de manière séquentielle

$$
\begin{cases}
x_1 &= r_1^{-1}(y_1) \\
x_2 &= r_2^{-1}((y_2 - \frac{f_2(x_1)}{g_2(x_1)})\frac{q_2(x_1)}{p_2(x_1)}) \\
&\vdots \\
x_k &= r_k^{-1}((y_k - \frac{f_k(k_1,\ldots,k_{k-1})}{g_k(k_1,\ldots,k_{k-1})})\frac{q_k(k_1,\ldots,k_{k-1})}{p_k(k_1,\ldots,k_{k-1})}) \\
&\vdots \\
x_n &= r_n^{-1}((y_n - \frac{f_n(n_1,\ldots,n_{n-1})}{g_n(n_1,\ldots,n_{n-1})})\frac{q_n(n_1,\ldots,n_{n-1})}{p_n(n_1,\ldots,n_{n-1})})
\end{cases}
$$

TRMS – Présentation (4)

$n = m + r$

$\varphi_1 : \mathbb{K}^n \to \mathbb{K}^n$

$\varphi_2 : \mathbb{K}^n \to \mathbb{K}^m$

$\varphi_3 : \mathbb{K}^m \to \mathbb{K}^m$

paramètres :

$\mathbb{K} = \mathbb{F}_{2^8}, n = 28, m = 20$

avec φ_1, φ_3 linéaires, et $\varphi_2 = \pi \circ \widetilde{\varphi_2} \circ i$ où $\widetilde{\varphi_2}$ une TRM

secret	$S : \mathbb{K}^m \to \mathbb{K}^n,$	$S = \varphi_3^{-1} \circ \varphi_2^{-1} \circ \varphi_1^{-1}$
publique	$V : \mathbb{K}^n \to \mathbb{K}^m,$	$V = \varphi_3 \circ \varphi_2 \circ \varphi_1$

Vérifier une signature

$$\begin{cases} f_1(x_1, \ldots, x_n) \\ \vdots \\ f_m(x_1, \ldots, x_n) \end{cases}$$

$deg(f_i) = 2, i \in \{1, \ldots, m\}$

TRMS – Attaque (1)

Forger une signature

Trouver $(x_1, \ldots, x_n)$ tel que

$$\begin{cases} f_1(x_1, \ldots, x_n) = z_1 \\ \vdots \\ f_m(x_1, \ldots, x_n) = z_m \end{cases}$$

Problèmes

- $\mathbb{K}$ est un "grand" corps.
- Le système sous-déterminé va avoir au moins $\#\mathbb{K}^r$ solutions **valides**.

TRMS – Attaque (2)

Solution

- On n'ajoute pas les équations de corps.
- On peut fixer r variables.

Forger une signature

Trouver $(x_1, \ldots, x_m)$ tel que

$$\begin{cases} f_1(x_1, \ldots, x_m) = z_1 \\ \vdots \\ f_m(x_1, \ldots, x_m) = z_m \end{cases}$$

$$\begin{aligned} \textit{Macaulay} : &\quad d_{reg} = m + 1 \\ \textit{Bézout} : &\quad \#Var = 2^m \end{aligned}$$

TRMS – Attaque (3)

Approche hybride

- On spécialise en plus 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
$\mathcal{O}\left((\#\mathbb{K})^k\left((n-k)\cdot \mathrm{C}^{d_{\mathrm{reg}}}_{n+d_{\mathrm{reg}}-1}\right)^{\omega}\right)$


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*, 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)*.

TRMS – Résultats

Forger une signature

$\#\mathbb{K}$	m	$m-k$	k	T_{F_5}	Mem	$\mathrm{Nop}_{\mathrm{F}_5}$	N
2^8	20	18	2	51h	41.940 Go	2^{41}	2^{57}
		17	3	2h45min.	4.402 Go	2^{37}	2^{61}
		16	4	626 sec.	912 Mo	2^{34}	2^{66}
		15	5	46 sec.	368 Mo.	2^{30}	2^{70}

FIG.: Temps de résolution et complexité

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 : $O(2^{n/2})$

Fonctions de hachage multivariées – Présentation

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

- Utilise Merkle-Damgård
- Fonction de compression $\Leftrightarrow$ Système polynomial
- Soit $\mathbb{K} = \mathbb{F}_{2^k}$, $f : \mathbb{K}^{m+n} \longrightarrow \mathbb{K}^m$
- m équations pour $m + n$ variables

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 $\mathbb{K} = \mathbb{F}_{2^k}$, $f : \mathbb{K}^{m+n} \longrightarrow \mathbb{K}^m$

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

Jintai Ding and Bo-Yin Yang.
Multivariates polynomials for hashing.
INSCRYPT 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 en plus 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

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	$\approx$ 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 – Contexte

- 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 – Attaque en collision

Attaque en collision

1. choisir une différence δ de poids de Hamming $w(\delta)$ fixé
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
Construction creuse

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

Conclusion

Perspectives

- Analyse systématique d'autres schémas
- Complexité des systèmes creux