Les progrès algorithmiques sur le problème du logarithme discret.

E. Thomé

Inria/Caramel, CNRS, Nancy

12 novembre 2014

Plan

Introduction

La cryptographie est présente dana la vie de tous les jours.

Exigences multiples :

- Efficacité — contraintes liées au champ d'application.
- Sécurité — résistance à certains scenario d'attaque.

Sur quoi repose la sécurité ?

Ces objets « contiennent de la cryptographie ».

C'est-à-dire ? Protocoles incluant divers types de primitives :

- De la cryptographie symétrique (AES, . . .) ;
- Des fonctions de hachage (md5, SHA-1, SHA-3, . . .) ;
- De la cryptographie à clé publique (RSA, DSA, . . .).

primitives solides + implémentation parfaite → sécurité

Différents métiers

On peut s'intéresser à différentes choses :

- L'étude des protocoles cryptographiques ;
- La conception de solutions cryptographiques ;
- L'étude de leur implémentation ;
- L'étude de la sécurité des primitives sous-jacentes.

Mon travail est l'étude de la sécurité des primitives de la cryptographie à clé publique.

Objectifs antagonistes

Casser une primitive de cryptographie à clé publique = résoudre un problème mathématique.

Une métrique usuelle : la taille de la clé publique.

Lorsqu'on augmente la taille de clé :

- le problème à résoudre est plus difficile : plus de sécurité.
- les calculs sont aussi plus lourds : moins d'efficacité.

La difficulté du problème mathématique que doit résoudre l'attaquant dépend de l'algorithme employé.

Connaît-on le meilleur algorithme ?

Connaissance algorithmique utile

Pour un algorithme résolvant le problème mathématique, on a :

$$\text{temps nécessaire pour casser} = f(\text{taille de clé}).$$

La fonction f est la complexité de l'algorithme.

Connaissance algorithmique utile

Pour un algorithme résolvant le problème mathématique, on a :

$$\text{temps nécessaire pour casser} = f(\text{taille de clé}).$$

La fonction f est la complexité de l'algorithme.

- Si on connaît le meilleur algorithme :
  - On sait choisir la bonne taille de clé pour garantir une difficulté satisfaisante.
  - On peut dimensionner les standards et produits en fonction.

Connaissance algorithmique utile

Pour un algorithme résolvant le problème mathématique, on a :

$$\text{temps nécessaire pour casser} = f(\text{taille de clé}).$$

La fonction f est la complexité de l'algorithme.

- Si on connaît le meilleur algorithme :
  - On sait choisir la bonne taille de clé pour garantir une difficulté satisfaisante.
  - On peut dimensionner les standards et produits en fonction.
- Si on se trompe, et qu'un algorithme encore meilleur existe :
  - La complexité de l'attaque est moindre ;
  - La taille de clé qu'on croyait suffisante ne l'est plus.
  - Les standards et recommandations doivent être modifiés.

Amélioration des algorithmes

Il y a deux façons d'améliorer une attaque.

- Améliorer l'implémentation du meilleur algorithme connu.
  - Crucial lorsqu'on en vient à choisir la taille de clé ;
  - Pour des algorithmes complexes, il y a un fossé considérable entre une implémentation optimisée et une implémentation *proof-of-concept*.
- Améliorer la complexité de l'algorithme lui-même.
  - Remet évidemment en jeu la question de l'implémentation !

Plan

Les primitives usuelles

Les cryptosystèmes à clé publique les plus communs reposent sur des problèmes de la théorie des nombres.

- Cryptosystème RSA : factorisation d'entiers ;
- Algorithme de signature DSA : logarithme discret dans les corps finis ;
- Signature ECDSA : logarithme discret dans les courbes elliptiques.

La factorisation d'entiers

Meilleur algorithme connu pour factoriser un entier N : le crible algébrique (Number Field Sieve, NFS).

Complexité de NFS

$$\text{temps} = \exp\left((1 + o(1))(64/9)^{1/3}(\log N)^{1/3}(\log\log N)^{2/3}\right).$$

Le diable est dans les détails, notamment dans le $o(1)$.

Pour faire simple :

- Pour un entier de n bits, temps $= \exp(c\sqrt[3]{n})$.
- Une bonne implémentation gagne un (gros) facteur constant sur une implémentation plan-plan.

Complexité $\rightsquigarrow$ taille de clé

Quelle est la capacité limite de calcul d'un attaquant ?

- Aujourd'hui : disons 2^{80} à 2^{100} opérations.
- Dans dix ans : attention à la loi de Moore.
- Dans k années : ordre de grandeur $2^{100+0.66k}$.

Usuellement, et quand on peut, suffisamment sûr $= 2^{128}$.

Taille de clé en fonction du temps ?

Complexité $\rightsquigarrow$ taille de clé

Quelle est la capacité limite de calcul d'un attaquant ?

- Aujourd'hui : disons 2^{80} à 2^{100} opérations.
- Dans dix ans : attention à la loi de Moore.
- Dans k années : ordre de grandeur $2^{100+0.66k}$.

Usuellement, et quand on peut, suffisamment sûr $= 2^{128}$.

Taille de clé en fonction du temps ?

$$2^{100+0.66k} = \exp(c\sqrt[3]{n}),$$

$$n = \text{fonction cubique en } k.$$

La taille de clé pour RSA est en train de devenir un problème

$n = 768$: fait ; $n = 1024$: faisable ; 2^{128} : $n \geq 3000$.

Pas que RSA !

RSA et la factorisation ne sont qu'un exemple.

Autres protocoles $\rightarrow$ autres problèmes $\rightarrow$ autres algorithmes.

On s'intéresse ici au problème du logarithme discret.

- On choisit une groupe G (donnée publique)
  - cyclique ;
  - de cardinal premier (ou presque) ;
  - dans lequel on sait calculer (vite).
- On choisit un générateur $g \in G$.
- Problème du logarithme discret (DLP) :

  Étant donné g^x, trouver x.
  (l'inverse est facile !)

Résoudre DLP ?

Dire qu'on résout DLP n'a (à peu près) pas de sens.

Il faut savoir de quoi on parle

- On fixe une famille de groupes ;
- Famille indexée par un paramètre de taille.

Résoudre DLP ?

Dire qu'on résout DLP n'a (à peu près) pas de sens.

Il faut savoir de quoi on parle

- On fixe une famille de groupes ;
- Famille indexée par un paramètre de taille.

Questions qui nous intéressent :

- DLP dans la famille des groupes de la forme $\mathbb{F}_p^\times$;
- DLP dans la famille des groupes de la forme $\mathbb{F}_{2^n}^\times$;
- DLP dans la famille des groupes de la forme $E(\mathbb{F}_p)$;
- ...

Un algorithme pour une famille : temps $= f(\text{taille})$.

DLP pour faire quoi ?

Pour quels protocoles Alice et Bob voudraient du logarithme discret ?

- Diffie-Hellman : on fait l'hypothèse que DLP est dur.
- DSA.
- El Gamal, Schnorr ($\rightarrow$ vote électronique, ...).

Quelle famille de groupes choisiraient-ils ?

- Corps finis : premiers ($\mathbb{F}_p^\times$), binaires ($\mathbb{F}_{2^n}^\times$).
- Courbes elliptiques : $E(\mathbb{F}_q)$.
- Choix du chercheur : $E(\mathbb{F}_q)$ (DLP plus dur, clés plus petites). Mais ce n'est pas tout (standards, brevets, contraintes matérielles, ...).

Couplages

Autre cas où on repose sur le DLP :

Couplages en cryptographie

Certains protocoles ($\pm$ exotiques) utilisent les couplages :

$$e : \mathbb{G} \times \mathbb{G} \to K^*, \quad (K \text{ corps fini})$$

- Identity-based encryption,
- Tripartite DH,
- Signatures courtes,
- Proxy re-encryption.

Le corps fini est ici nécessairement dans l'affaire.

Cas usuel :
- $\mathbb{G} =$ courbe elliptique sur $\mathbb{F}_q$,
- K une petite extension de $\mathbb{F}_q$.

Exigence de sécurité : DLP difficile dans $\mathbb{G}$ ET dans K.

Plan

Focus

Notre centre d'intérêt ici : $\mathbb{F}_{2^n}^{\times}$

Étudier le DLP dans $\mathbb{F}_{2^n}$ a trait à :

- la pertinence de $\mathbb{F}_{2^n}^{\times}$ comme famille de groupes en crypto ;
- la pertinence des couplages sur des courbes elliptiques en caractéristique 2.

Qui utilise la caractéristique 2 ?

- Aucun standard ;
- Mais tout le monde est libre de le faire !

Le DLP dans les corps finis (FF-DLP)

On dispose dans tous les cas d'algorithmes sous-exponentiels :

La fonction L

$$L_q(\alpha, c) \overset{\text{déf}}{=} \exp\left(c(1+o(1))(\log q)^{\alpha}(\log\log q)^{1-\alpha}\right).$$

- $\alpha = 0$: polynomial en $\log q$.
- $\alpha = 1$: exponentiel en $\log q$.
- Reduire c est important, reduire α encore plus.

Situation de 2006 à 2013 : FF-DLP en $L_q(1/3, \alpha)$, mais plusieurs sous-cas.

DLP dans $\mathbb{F}_{p^n}$

La généalogie remonte à CFRAC dans les années 1970.
Algorithmes modernes : adaptations du crible algébrique (NFS).

La complexité dépend des tailles relatives de n versus $\log p$.

- DLP dans $\mathbb{F}_{p^n}^\times$ pour $\log p \ll n$: Adleman, 1994 ; Adleman, Huang, 1999 : Function Field Sieve (FFS) :

$$L_q(1/3, 1.53).$$

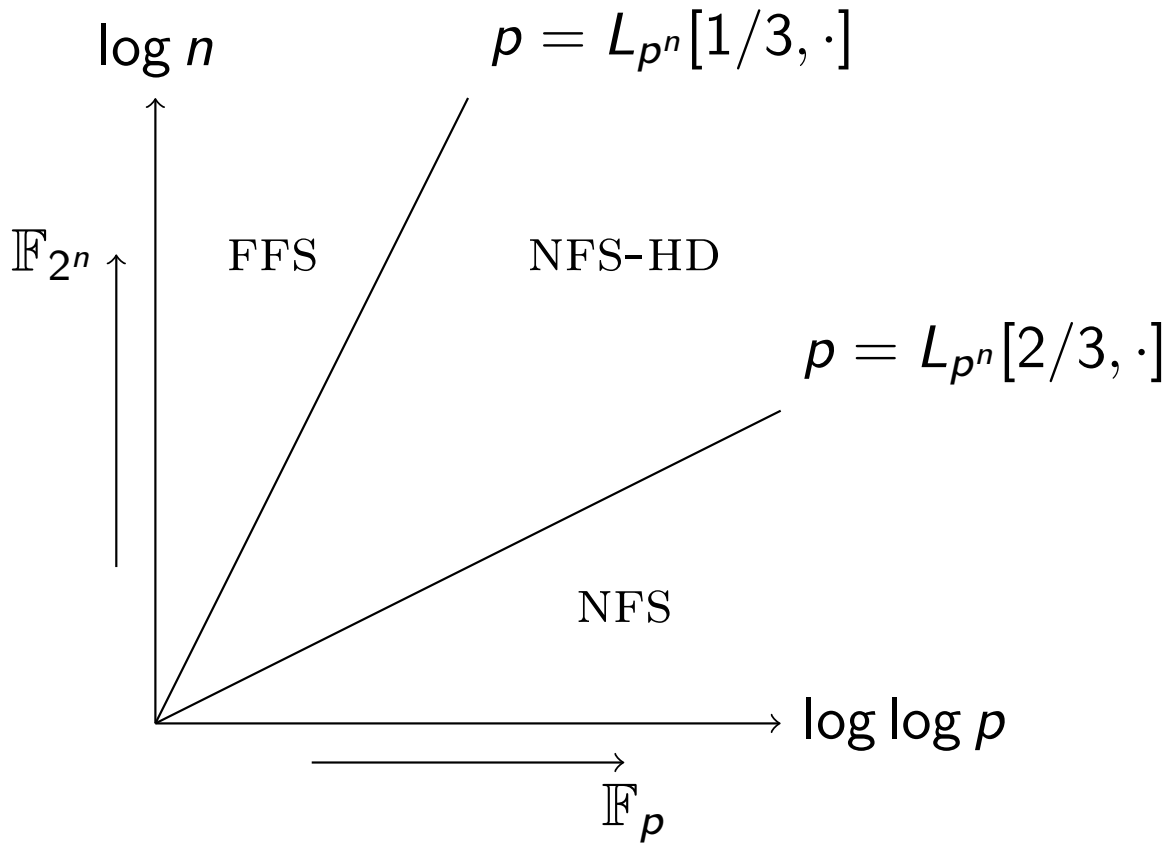
- DLP dans $\mathbb{F}_{p^n}^\times$ pour $n \ll \log p$: Schirokauer, 1999. Crible algébrique pour DLP (NFS-DL) ; plusieurs variantes ;

$$L_q(1/3, 1.93).$$

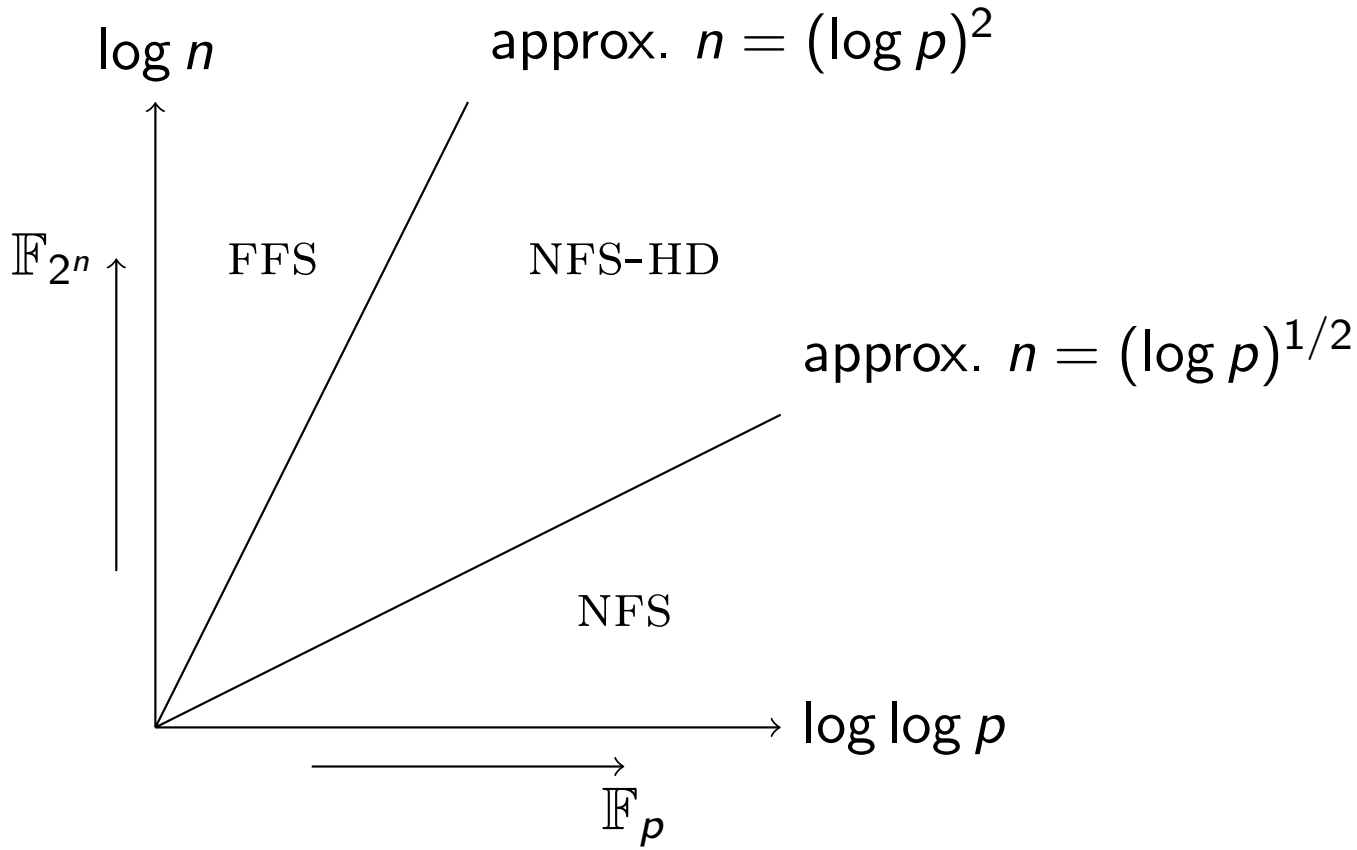
- DLP dans $\mathbb{F}_{p^n}^\times$ entre les deux : Joux *et al.*, 2006, + some recent improvements [JP13, BGGM14]. NFS-HD :

$$L_q(1/3, c) \quad \text{(pour un certain } c\text{)}.$$

Situation en 2006



Situation en 2006



À voir, à faire

Prédire le temps de calcul des algorithmes comme NFS est très difficile.

- En vérité c'est une collection d'une multitude d'algorithmes ;
- Impact majeur des problèmes d'implantation ;
- Complexité à très gros grain : en fait, une myriade de raffinements.
  (NFS : 20 ans de travail absorbés dans $o(1)$).

Sujet étudié : les limites de ce qui est calculable aujourd'hui.

- Pour $\mathbb{F}_p$ à cause de ECDSA.
- Pour $\mathbb{F}_{2^p}$ à cause de son attrait pour l'implementeur.
- Pour $\mathbb{F}_{q^k}$ avec $\mathbb{F}_q$ comme précédemment, et $k < 16$ (PBC).

Vocabulaire

Notion souvent rencontrée : friabilité d'entiers ou de polynômes.

Définition : friabilité

- Un polynôme de degré $\leq n$ est k-friable si tous ses facteurs irréductibles ont degré $\leq k$.
- Un entier $\leq X$ est B-friable si ses facteurs premiers sont $\leq B$.

Probabilité de friabilité contrôlée principalement par le ratio de tailles : $u = \frac{n}{k}$, ou $u = \frac{\log X}{\log B}$.

Facile à retenir : Prob $\approx u^{-u(1+o(1))}$.

- Canfield-Erdős-Pomerance + variantes.
- Les cas limites sont difficiles.
- La th. analytique des nombres donne des résultats plus précis.

Rappel

Tester la friabilité de polynômes, tout comme leur factorisation, se fait en temps polynomial.

(c'est plus difficile pour les entiers, mais on n'en a pas besoin ici).

Plan

Dinde aux marrons

Les fêtes de Noël en 2012 ont été très productives (gastronomie ?).

- Une douzaine de records entre le 25/12/2012 et 06/2013.
- Nouveaux algorithmes.
- Travail présenté : meilleure complexité, relativement simple à présenter. (Eurocrypt 2014 BPA, avec Barbulescu, Gaudry, Joux).

On n'aborde pas les raffinements qui sont venus après :

- Göloğlu, Granger, McGuire, Zumbrägel (Crypto 2013).
- Joux (SAC 2013).
- Granger, Kleinjung, Zumbrägel (plusieurs articles en 2014).

Problèmes étudiés

L'algorithme présenté est bien adapté à $\mathbb{F}_{q^{2k}}$, pour $q \approx k$.

Problèmes étudiés

L'algorithme présenté est bien adapté à $\mathbb{F}_{q^{2k}}$, pour $q \approx k$.

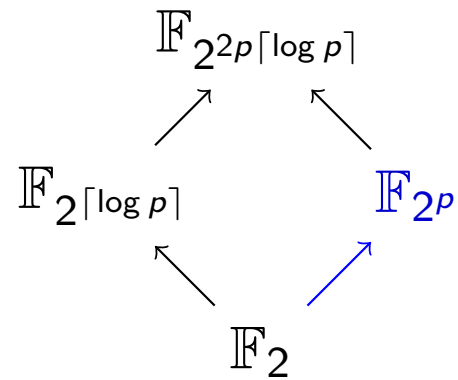
- $q \approx k$; comparaison troublante (ça ne type pas).
- Pour rentrer dans ce cadre, un corps doit posséder un sous-corps de la bonne taille.
- Cas plutôt mal adapté : $\mathbb{F}_{2^p}$ pour p premier.

Problèmes étudiés

L'algorithme présenté est bien adapté à $\mathbb{F}_{q^{2k}}$, pour $q \approx k$.

- $q \approx k$; comparaison troublante (ça ne type pas).
- Pour rentrer dans ce cadre, un corps doit posséder un sous-corps de la bonne taille.
- Cas plutôt mal adapté : $\mathbb{F}_{2^p}$ pour p premier.

On peut travailler dans $\mathbb{F}_{2^{2p\lceil \log p \rceil}}$



- Si on sait résoudre DLP dans (les sous-groupes de) $\mathbb{F}_{2^{2p\lceil \log p \rceil}}^{\times}$, alors de même pour $\mathbb{F}_{2^p}^{\times}$;
- Il se peut même qu'on obtienne ainsi une bonne complexité.

Problèmes étudiés

L'algorithme présenté est bien adapté à $\mathbb{F}_{q^{2k}}$, pour $q \approx k$.

- $q \approx k$; comparaison troublante (ça ne type pas).
- Pour rentrer dans ce cadre, un corps doit posséder un sous-corps de la bonne taille.
- Cas plutôt mal adapté : $\mathbb{F}_{2^p}$ pour p premier.
- Cas plus favorable : les petites extensions de $\mathbb{F}_{2^p}$, comme dans le contexte des couplages.
  Exemple : couplages sur une courbe définie sur $\mathbb{F}_{2^{239}}$, à valeurs dans $\mathbb{F}_{2^{239*4}}$.
  - $\mathbb{F}_{2^{239*4}}$ a comme sous-corps $\mathbb{F}_{2^{239}}$, mais indice trop petit.
  - $\mathbb{F}_{2^{239*16}}$ est bien adapté au nouvel algorithme,
    Le "*4" inhérent a fait une partie du travail pour nous.

Notations

Soit $K = \mathbb{F}_{q^{2k}}$ avec $k \lessapprox q$.

On représente $\mathbb{F}_{q^2}$ comme on veut.

Les éléments de K sont des polynômes sur $\mathbb{F}_{q^2}$
(modulo un polynôme irréductible définissant $\mathbb{F}_{q^{2k}}$).

Construction du polynôme de définition :

- Soit $h_0, h_1 \in \mathbb{F}_{q^2}[x]$ de petit degré (2 est normalement ok).
- Soit $\Phi(x) = h_1(x)x^q - h_0(x)$.
- Jusqu'à ce que $\Phi(x)$ ait un facteur irréd. $f(x)$ de degré k.

Heuristique 1

En prenant h_0, h_1 de degré borné par une constante suffit à trouver un f idoine. Soit δ cette constante.

Stratégie

DLP dans $\mathbb{F}_{q^{2k}}$:

- Entrée : $P(x) \in \mathbb{F}_{q^{2k}}$, degré $D < k$, coefficients dans $\mathbb{F}_{q^2}$.
- Objectif : $\log P(x)$, modulo $\ell \mid q-1$.

Étape intermédiaire appelé un pas de descente

Récrire P comme produit de polynômes plus en plus petits.

- Récrire P (par ex. $D = \deg P = 1000$) comme produit de polynômes de degré 500 ;
- Récrire chacun comme produit de polynômes de degré 250 ;
- Récrire chacun comme produit de polynômes de degré 125 . . .
- Idem pour relier ensemble tous les polynômes de petit degré.

Pas comme avant

Une telle stratégie n'est pas extraordinairement innovante :

- Encore faut-il disposer d'un algorithme pour faire un tel pas de descente ;
  (auparavant on ne descendait pas aussi vite)
- Il est nécessaire de contrôler combien de polynômes apparaissent dans chaque récriture ;
- Veiller à la complexité de chaque pas de descente.

Point de départ

Qu'obtient-on avec $h_1(x)x^q - h_0(x) \equiv 0 \mod f(x)$?

$$x^q - x = \prod_{\alpha \in \mathbb{F}_q} (x - \alpha),$$

$$X^q Y - X Y^q = \prod_{(\alpha:\beta) \in \mathbb{P}^1(\mathbb{F}_q)} (\beta X - \alpha Y),$$

$$h_0(x) - x h_1(x) \equiv h_1(x) \prod_{\alpha \in \mathbb{F}_q} (x - \alpha).$$

- à droite : polynômes de degré 1 ;
- à gauche : petit degré, donc se factorise en petits polynômes.

Ça ne fait qu'une seule relation, hélas.

Mieux ?

On repart de la même relation :

$$X^q Y - X Y^q = \prod_{(\alpha:\beta)\in\mathbb{P}^1(\mathbb{F}_q)} (\beta X - \alpha Y).$$

Substitutions : $(X, Y) \leftarrow (aP + b, cP + d)$ pour $a, b, c, d \in \mathbb{F}_{q^2}$.

Mieux ?

On repart de la même relation :

$$X^q Y - XY^q = \prod_{(\alpha:\beta)\in\mathbb{P}^1(\mathbb{F}_q)} (\beta X - \alpha Y).$$

Substitutions : $(X, Y) \leftarrow (aP + b, cP + d)$ pour $a, b, c, d \in \mathbb{F}_{q^2}$.

$$(a^q P(x)^q + b^q)(cP + d) - (aP + b)(c^q P(x)^q + d^q) \equiv$$
$$\prod_{(\alpha:\beta)\in\mathbb{P}^1(\mathbb{F}_q)} (\beta(aP + b) - \alpha(cP + d)).$$

Mieux ?

On repart de la même relation :

$$X^q Y - X Y^q = \prod_{(\alpha:\beta)\in\mathbb{P}^1(\mathbb{F}_q)} (\beta X - \alpha Y).$$

Substitutions : $(X, Y) \leftarrow (aP + b, cP + d)$ pour $a, b, c, d \in \mathbb{F}_{q^2}$.

$$(a^q P(x)^q + b^q)(cP + d) - (aP + b)(c^q P(x)^q + d^q) \equiv$$
$$\prod_{(\alpha:\beta)\in\mathbb{P}^1(\mathbb{F}_q)} (\beta(aP + b) - \alpha(cP + d)).$$

On peut récrire $P(x)^q$ $= (a_0 + a_1 x + \cdots + a_D x^D)^q,$

$$= a_0^q + a_1^q x^q + \cdots + a_D^q x^{qD},$$
$$\equiv a_0^q + a_1^q (\frac{h_0}{h_1}) \cdots + a_D^q (\frac{h_0}{h_1})^D,$$
$$\equiv \tilde{P}(\frac{h_0}{h_1}) \text{ pour un } \tilde{P} \text{ du même degré.}$$

De nouvelles relations

$$h_1^D \left((a^q \tilde{P}(h_0/h_1) + b^q)(cP + d) - (aP + b)(c^q \tilde{P}(h_0/h_1) + d^q) \right)$$
$$\equiv h_1^D \prod_{(\alpha:\beta)\in\mathbb{P}^1(\mathbb{F}_q)} (\beta(aP + b) - \alpha(cP + d)).$$

- À droite : on a des polynômes de degré $D = \deg P$.
- À gauche : degré au plus $(1 + \delta)D$.
  La probabilité de $D/2$-friabilité est minorée par une constante.

Chaque nouveau quadruplet (a, b, c, d) = une nouvelle chance d'être friable à gauche.

- Combien peut-on écrire de relations avec le LHS friable ?
- On souhaiterait n'avoir que P à droite, pas ses frères et sœurs.

Combien de relations

Pour un P donné, combien obtient-on de relations ?

- Nombre de choix non équivalents pour (a, b, c, d) : $q^3 + q$.
  (lié aux classes de $\mathrm{PGL}(2, q^2)$ modulo $\mathrm{PGL}(2, q)$)
- La probabilité étant minorée, on a $\Theta(q^3)$ relations.

Peut-on avoir uniquement P à droite ?

- Chaque relation implique q polynômes parmi $\left\{P + \gamma,\ \gamma \in \mathbb{F}_{q^2}\right\}$;
- On peut multiplier, diviser, les relations entre elles pour agir sur le terme de droite ;
- Problème d'algèbre linéaire à q^2 équations et q^3 inconnues.

Combien de relations

Pour un P donné, combien obtient-on de relations ?

- Nombre de choix non équivalents pour (a, b, c, d) : $q^3 + q$. (lié aux classes de $\mathrm{PGL}(2, q^2)$ modulo $\mathrm{PGL}(2, q)$)
- La probabilité étant minorée, on a $\Theta(q^3)$ relations.

Peut-on avoir uniquement P à droite ?

- Chaque relation implique q polynômes parmi $\left\{P + \gamma,\ \gamma \in \mathbb{F}_{q^2}\right\}$;
- On peut multiplier, diviser, les relations entre elles pour agir sur le terme de droite ;
- Problème d'algèbre linéaire à q^2 équations et q^3 inconnues.

Un pas de descente

En temps $O(q^5)$, on peut récrire P comme produit d'au plus $q^2 D$ polynômes de degré au plus $\lceil D/2 \rceil$.

L'arbre de descente

Chaque nœud de l'arbre de descente est une application du pas de descente. L'arité est donc $q^2 D$.

étage	deg P_i	largeur
0	k	1
1	$k/2$	$q^2 k$
2	$k/4$	$q^2 k \cdot q^{2\frac{k}{2}}$
3	$k/8$	$q^2 k \cdot q^{2\frac{k}{2}} \cdot q^{2\frac{k}{4}}$
$\vdots$	$\vdots$	$\vdots$
$\log k$	1	$\leq q^{2\log k} k^{\log k}$

Nombre total de nœuds $= q^{O(\log k)}$.

Chaque nœud induit un coût polynomial en q.

Tout en bas

Au niveau $O(\log k)$, le degré est en dessous d'une constante (2).

En continuant avec le pas de descente, on obtient un système surdéterminé reliant les logarithmes de tous les petits polynômes.

- Suivre l'arbre de descente pour réduire le défi d'origine au calcul de nombreux logarithmes de polynômes de degré 2.
- Continuer «en boucle» pour «descendre» de deg ≤ 2 à deg ≤ 2 ; résoudre le système linéaire.
- Substituer pour trouver la solution voulue.

Complexité totale

$q^{O(\log k)}$ nœuds, $O(q^5)$ par nœud $\rightsquigarrow$ $\boxed{\text{Complexité } q^{O(\log k)}}$.

Résultat principal

Résultat principal

Soit K un corps fini de la forme $\mathbb{F}_{q^{2k}}$. Un logarithme discret dans K peut être calculé en temps

$$\max(q,k)^{O(\log k)}.$$

- On a traité le cas $k \approx q$.
- Pour $q < k$, on compose avec le corps $\mathbb{F}_{q^c}$ avec $c = \left\lceil \frac{\log k}{\log q} \right\rceil$ (d'où le max dans la formule).

Exemple pour $K = \mathbb{F}_{2^p}$:

- On prend $q \approx k \approx p$.
- Coût $p^{O(\log p)} = \exp(O((\log p)^2))$.
- C'est quasi-polynomial.

Plan

Complexité quasi-polynomiale : conséquences

Avec un algorithme de complexité $\exp(O(\log p)^2)$ pour $\mathbb{F}_{2^p}^\times$, comment évolue la taille de clé avec le temps ?

$$2^{100+0.66k} = \exp(O(\log p)^2),$$
$$p = \exp O(\sqrt{k}).$$

Deux questions :

- quelle portée pratique aujourd'hui ?
- quelle portée pratique demain ?

Complexité quasi-polynomiale : conséquences

Avec un algorithme de complexité $\exp(O(\log p)^2)$ pour $\mathbb{F}_{2^p}^\times$, comment évolue la taille de clé avec le temps ?

$$2^{100+0.66k} = \exp(O(\log p)^2),$$
$$p = \exp O(\sqrt{k}).$$

Deux questions :

- quelle portée pratique aujourd'hui ?
- quelle portée pratique demain ?
  clairement, une telle attaque disqualifie $\mathbb{F}_{2^p}^\times$ pour la crypto.

Portée pratique aujourd'hui

Cet algorithme et ses améliorations battent-ils des records ?

- OUI pour $\mathbb{F}_{2^p}$: Kleinjung, 2014 : DLP dans $\mathbb{F}_{2^{1279}}$ vu comme sous-corps de $\mathbb{F}_{2^{10232}}$.
- Deux fois OUI pour $\mathbb{F}_{2^{kp}}$ dans le contexte des couplages : Granger, Kleinjung, Zumbrägel, Crypto 2014 : $\mathbb{F}_{2^{4\cdot 1223}}$, $\mathbb{F}_{2^{12\cdot 367}}$.

Dans les deux cas, une importante machinerie additionnelle est nécessaire.

Conclusion

On peut désormais totalement oublier dans le contexte de la cryptographie :

- DLP dans $\mathbb{F}_{2^k}^{\times}$.
- Les couplages sur les courbes définies sur $\mathbb{F}_{2^k}$.
- Idem plus généralement en petite caractéristique.

Extensions ?

L'algorithme est très sensible au degré de x^q :

- Extension en grande caractéristique ? Fantaisiste.
- Extension à la factorisation d'entiers ? Même obstacle.
- Analogie : factorisation d'entiers versus polynômes.

Depuis la découverte de l'algorithme, ça bouge ! Améliorations pratiques notamment.

- Pourquoi pas des extensions de l'approche ;
- Là où il y a davantage de polynômes, possibilités accrues ;
- DLP dans $E(\mathbb{F}_{2^n})$ pourrait être une cible.

Cryptographie pour la décennie à venir : vraisemblablement $E(\mathbb{F}_p)$.