Leçons de Mathématiques d'Aujourd'hui

1er février 2001

La cryptologie est-elle soluble dans les mathématiques ?

F. Morain

POLYTECHNIQUE

http://www.lix.polytechnique.fr/Labo/Francois.Morain/

Plan

I. Introduction.

II. Le chiffrement RSA.

III. L'échange de clefs à la Diffie-Hellman.

IV. Vers des preuves de sécurité.

Planche 1

I. Introduction

Cryptographie : protéger les échanges d'information de toute malveillance.

Ex. courrier électronique, signature de contrats, carte à puces, établissement de tunnels sécurisés (IPSec), etc.

Les besoins de la correspondance électronique :

- confidentialité ;
- authentification ;
- intégrité ;
- signature.

Deux paradigmes

Cryptographie symétrique : la même clef sert à chiffrer et déchiffrer, donc doit rester secrète.

Cryptographie asymétrique : une clef pour chiffrer, une pour déchiffrer et celle-ci ne peut se déduire facilement de celle-là ; la clef de chiffrement peut être publique.

Planche 2

Cryptographie symétrique

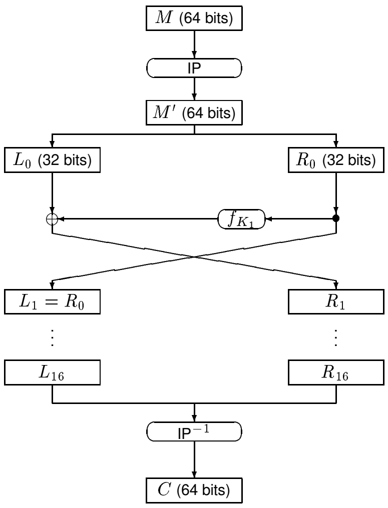


Planche 3

Cryptographie asymétrique (suite)

Prérequis : chaque utilisateur a deux algorithmes $\mathcal{C}$ et $\mathcal{D}$ tels que $\mathcal{D}(\mathcal{C}(m)) = m$ et on ne peut déduire facilement (au sens de la théorie de la complexité) $\mathcal{D}$ de $\mathcal{C}$; $\mathcal{C}$ est publié dans un annuaire.

Confidentialité :

- Alice envoie $\mathcal{C}_B(m)$;
- Bob déchiffre par $\mathcal{D}_B(\mathcal{C}_B(m)) = m$.

Signature :

- Alice envoie $\mathcal{C}_B(m)$ et $S_A(m) = \mathcal{D}_A(m)$;
- Bob déchiffre $m = \mathcal{D}_B(\mathcal{C}_B(m))$ et vérifie que $\mathcal{C}_A(S_A(m)) = m$.

Comment trouver des exemples ? prendre un problème difficile et introduire une brèche secrète permettant de chiffrer/déchiffrer.

Planche 4

Mais encore ?

Prendre un problème difficile et en faire un cryptosystème sûr.

Difficile ? non résoluble en un temps raisonnable (temps polynomial). On cherche un problème dont les instances soient (presque) toujours difficiles, pas seulement le pire cas ou bien en moyenne.

Ex. factorisation d'entiers, calcul du logarithme discret, décodage d'un code aléatoire (McEliece), problèmes dans les réseaux (Merkle, . . . , Ajtai), systèmes d'équations polynomiales (HFE, etc.), théorie des groupes (groupes de tresses ?), etc.

Sûr ? trouver un modèle de sécurité réaliste et prouver la sécurité dans ce modèle. Gardons à l'esprit cette pensée de Knudsen :

A provably secure system is probably not.

Scénarions plus élaborés : sécurité sémantique, non-malléabilité, résistance aux attaques à textes chiffrés adaptatives, etc. Attentions aux solutions non fondées (OAEP).

Planche 5

II. Le chiffrement RSA

Rivest–**S**hamir–**A**dleman, 1976.

Fabrication des clefs : p et q premiers, $p \neq q$, $N = pq$, e tq $\mathrm{pgcd}(e, \lambda(N)) = 1$, $d \equiv 1/e \bmod \lambda(N)$. La clef publique est (N, e), la clef privée est d.

Chiffrement : $y = x^e \bmod N$.

Déchiffrement : $y^d \bmod N \equiv x$.

Signature : $S(m) = m^d \bmod N$.

Justification : Soit N un entier impair > 2. Alors N est sans facteur carré ssi $\forall\, a \in \mathbb{Z}/N\mathbb{Z}$:

$$a^{\lambda(N)+1} \equiv a \bmod N.$$

Faits :

- Fabriquer p et q est facile (probabilistiquement ou avec preuve).
- Calculer x^e se fait en temps $O(\log_2 e)$.

Planche 6

Sécurité élémentaire : factorisation d'entiers

Les différents type de méthode

Méthodes dépendant de p :

- crible : essayer tous les $p \leqslant \sqrt{N}$;
- Méthode ρ de Pollard ;
- Méthodes de type $p - 1$:

  a- calculer $g = (a^{k!} - 1, N)$ pour a premier avec N. Si $p \mid N$ et $p - 1 \mid k!$, alors $g > 1$.

  b- autres groupes : $p + 1$ (suites de Lucas) ; formes quadratiques ; ECM (courbes elliptiques, record à 54 chiffres décimaux), etc.

Méthodes générales : crible quadratique, algébrique.

Complexités :

méthode	temps de calcul
ρ	$\sqrt{p} = L_p[1, 1/2]$
ECM	$L_p[1/2, \sqrt{2}]$
QS	$L_N[1/2, c]$
NFS	$L_N[1/3, c]$

$$L_x[\alpha, c] = \exp\left(c(\log x)^{\alpha}(\log\log x)^{1-\alpha}\right)$$

$$L_x[0, c] = (\log x)^c, \quad L_x[1, c] = x^c$$

Planche 7

Les méthodes à combinaisons de congruences

Kraitchik : trouver x tq $x^2 \equiv 1 \bmod N$, $x \not\equiv \pm 1 \bmod N$.

étape 1 : trouver des couples $\{(u_i, v_i)\}_{i \in I}$ tels que

$$u_i^2 \equiv v_i \bmod N, \quad u_i^2 \neq \pm v_i.$$

étape 2 : trouver $J \subset I$,

$$\prod_{j \in J} v_j = V_J^2$$

étape 3 :

$$U_J = \prod_{j \in J} u_j, \quad U_J^2 \equiv V_J^2 \bmod N.$$

étape 4 : $x = U_J / V_J \bmod N$ est une racine carrée de 1 et avec proba $\geqslant 1/2$, elle est non triviale et factorise N.

Planche 8

Comment tester qu'un nombre est un carré ?

$$v_i = \prod_{p \in \mathbb{P}} p^{\alpha(i,p)}$$

$$Z = \prod_{j \in J} v_j = \prod_{p \in \mathbb{P}} p^{\sum_J \alpha(j,p)}$$

$$Z = \square \Leftrightarrow \forall\, p, \sum_J \alpha(j,p) \equiv 0 \bmod 2$$

$\Rightarrow$ problème d'algèbre linéaire : on doit chercher les relations de dépendance de la matrice $\mathcal{M} = (\alpha(i,p) \bmod 2)$.

Pb. $\#\mathbb{P}$ est gigantesque.

Idée : on remplace $\mathbb{P}$ par une base de nombres premiers $\mathcal{B} = \{p_1, p_2, \ldots, p_k\}$:

$$v_i = \prod_{r=1}^{k} p_r^{\alpha(i,r)}$$

$$Z = \prod_{j \in J} v_j = \prod_{r=1}^{k} p_r^{\sum_J \alpha(j,r)}$$

Planche 9

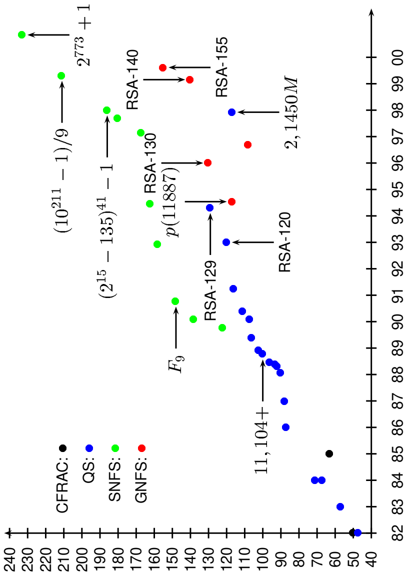


Planche 10

Quelques records

dd	qui	date	temps
100	Manasse & A. K. Lenstra	1991	7 MIPS-années
110	AKL	1992	un mois sur $5/8$ d'une MasPar 16K
120	AKL, Dodson, Denny, Manasse, Lioen, te Riele	1993	835 MIPS-années
129	Atkins, Graff, AKL, Leyland + INTERNET	1994	5000 MIPS-années
130	Dodson, Montgomery, Elkenbracht-Huizing, AKL, WWW, Fante, Leyland, Weber, Zayer	1996	500 MIPS-années
140	te Riele, Cavallar, Lioen, Montgomery, Dodson, AKL, Leyland, Murphy, Zimmermann	1999	1500 MIPS-années
155	CABAL	1999	8000 MIPS-années

Planche 11

```
RSA-155 =
10941738641570527421809707322040\
35761200373294544920599091384 2\
13147634998428893478471799725789\
12673324976257528997818337970765\
37244027146743531593354333897
=
                1026395928297411\
05772054196573991675900716567808\
03806680334193352179071130777 9
*
                1066034883801684\
54820927220360012878679207958575\
98929152227060823719306280864 3
```

Planche 12

Prospective

$$L_x[\alpha, c] = \exp\left(c(\log x)^{\alpha}(\log\log x)^{1-\alpha}\right)$$

N	$\sqrt{N}$	$L_N[1/2, 1]$	$L_N[1/3, 1]$
2^{128}	1.84×10^{19}	4.61×10^{8}	1.85×10^{5}
2^{256}	3.40×10^{38}	1.46×10^{13}	2.02×10^{7}
2^{512}	1.16×10^{77}	6.69×10^{19}	1.02×10^{10}
2^{768}	3.94×10^{115}	1.27×10^{25}	9.49×10^{11}
2^{1024}	1.34×10^{154}	4.42×10^{29}	3.82×10^{13}

$T(N)$	$N \mapsto N^2$
$\sqrt{N}$	T^2
$L_N[1/2]$	$T^{\sqrt{2}}$
$L_N[1/3]$	$T^{\sqrt[3]{2}}$

Ex. $N = 2^{512}$, $T(N) = 8000$ MIPSY,
$T(2^{1024}) = 82715$ MIPSY, mais avec une matrice $(3 \times 10^8)^2$ (faisable en 2018 (Brent) ?)

Et la loi de Joy ? les ordinateurs doublent de vitesse tous les deux ans. Dans deux ans, on peut savoir faire 2^{512+35} dans le même temps.

Planche 13

III. L'échange de clefs de Diffie-Hellman

Version idéalisée : Alice et Bob choisissent un groupe (cyclique) $G =< g >$ et échangent une valeur secrète au moyen des opérations suivantes :

$$A \xrightarrow{g^x} B$$
$$A \xleftarrow{g^y} B$$
$$A : K_{AB} = (g^y)^x = g^{xy}$$
$$B : K_{BA} = (g^x)^y = g^{xy}.$$

Rem. dans la pratique, il faut authentifier les clefs (pour contrer l'attaque par le milieu).

Problème de Diffie-Hellman : étant donnés (g, g^x, g^y), calculer g^{xy}.

Problème décisionnel : étant donnés $(g, a = g^x, b = g^y, c)$, a-t-on $c = g^{xy}$? (base du cryptosystème de Cramer-Shoup).

Logarithme discret : étant donnés (g, g^x), trouver x.

Thm. DL $\Rightarrow$ DH, réciproque presque vraie, (Maurer).

Autres algorithmes : ElGamal, Schnorr, DSA, etc.

Planche 14

Quels groupes ?

(on a juste besoin d'un semi-groupe)

- Abélien ou pas, $< g >$ l'est certainement.
- $< g >$ fini, sinon les éléments finissent par être trop grands.
- On a besoin de deux représentations de $G =< g > \rightarrow \{0, 1\}^*$: la représentation évidente est $g^x \mapsto x \mapsto (x)_2$, on a besoin d'une représentation non triviale $g^x \mapsto (z)_2$. Une présentation par générateurs et relations ne suffit pas.
- calculer g^x doit être rapide, et donc la loi de groupe aussi.

Ex. $2\mathbb{Z}, ((\mathbb{Z}/N\mathbb{Z})^*, +)$ (algorithme d'Euclide), groupe du tas de sable ; pour $G = (\mathbb{Z}/p\mathbb{Z})^*$: $g^x \mapsto (g^x)_2$.

Planche 15

Que dire de LD ?

Instance faible générique : $n = \mathrm{ord}(g)$ est friable (Pohlig-Hellman) $\Rightarrow n$ premier.

Borne supérieure pour LD : (Shanks) résoudre $g^x = a$ se fait en écrivant $x = u + v\sqrt{n}$ et $a(g^{-\sqrt{n}})^v = g^u$. La méthode ρ de Pollard a la même complexité en temps, mais $O(1)$ en espace.

Csqce. Si $n \approx 2^{200}$, LD n'est pas résoluble.

Borne inférieure pour LD : (Nechaev, Shoup) tout **algorithme générique** (i.e., qui n'exploite aucune propriété du groupe) résolvant LD (resp. DH) doit effectuer au moins $O(\sqrt{\#G})$ opérations de groupe.

Groupe de Nechaev : groupe pour lequel on pourrait prouver que le meilleur algorithme de résolution de LD est $O(\sqrt{\#G})$.

Est-ce que les groupes de Nechaev existent ? ? ?

Planche 16

Tâches algorithmiques à résoudre

- Trouver une bonne représentation des éléments.
- Être capable de tirer des éléments au hasard.
- Calculer efficacement la loi de groupe.
- Calculer $\#G$ rapidement.

Quelques groupes utilisés : groupe multiplicatif d'un corps fini, groupe de classes d'un corps de nombre, (jacobienne d'une) courbe algébrique.

Groupe	$\#G$	LD
$\mathbb{F}_q^*$	$q-1$	$L_q[1/3]$
groupe de classes	subexp	subexp
jacobienne	$g=1$: poly	$\sqrt{\#G}$
	$g=2,3,4$: poly (?)	$\sqrt{\#G}$
	$g\to\infty$: poly (?)	$L_{q^g}[1/2]$

$L_N[\alpha, c] = \exp((c+o(1))(\log N)^\alpha(\log\log N)^{1-\alpha})$.

Sécurité : 1024 bits pour $\mathbb{F}_q^*$ ou 200 bits pour les courbes.

Planche 17

Index-calculus

(Western et Miller, Pollard, Adleman, etc.)

Phase 1 : on cherche des relations du type

$$g^b \equiv \prod_i p_i^{\alpha_i} \bmod p$$

$$b \equiv \sum_i a_i \log_g p_i \bmod (p-1).$$

Avec k relations, on résout le système linéaire qui nous donne les $\log_g p_i$.

Phase 2 : on cherche b tq

$$g^x g^b \equiv \prod_i p_i^{\beta_i} \bmod p$$

$$x + b \equiv \sum_i \beta_i \log_g p_i \bmod (p-1).$$

Meilleur algorithme : $L_p[1/3]$ (Gordon, Schirokauer).

Record : Joux & Lercier (19 janvier 2001), p avec 110 chiffres décimaux.

Planche 18

Utilisation des courbes algébriques

Soit C une courbe projective lisse plane de genre g d'équation $F(X,Y)=0$ à coefficients dans $\mathbb{K}$, $\mathrm{char}(\mathbb{K})=p$, $\#\mathbb{K}=q=p^n$.

Déf. $C(\mathbb{K}) = \{P=(x,y)\in\mathbb{K}^2, F(x,y)=0\}\cup\{$points à l'$\infty\}$.

Thm. Quand $g\in\{0,1\}$, il y a une loi de groupe sur $C(\mathbb{K})$, sinon, il faut aller dans la jacobienne (groupe des *diviseurs*, i.e., sommes formelles de points).

Thm. (Weil) Soit $N_r = \#C(\mathbb{K}_r)$ où $[\mathbb{K}_r : \mathbb{K}] = r$:

$$Z(T) = \exp\left(\sum_{r\geqslant 1} N_r T^r/r\right) = \frac{L(T)}{(1-T)(1-qT)}$$

Thm. $\#\mathrm{Jac}(C) = L(1)$.

Coro. $|\#C-(q+1)|\leqslant 2g\sqrt{q}$;
$(\sqrt{q}-1)^{2g}\leqslant \#\mathrm{Jac}(C)\leqslant(\sqrt{q}+1)^{2g}$.

Intérêt : grande variété d'objets de cardinalité $\approx q^g$; familles paramétrées faciles à construire.

Planche 19

Les courbes utilisées dans la pratique

Le cas le plus utilisé : courbes hyperelliptiques $Y^2 = X^{2g+1}+\cdots$ (Miller, Koblitz pour $g=1$, Koblitz pour $g>1$, etc.) :

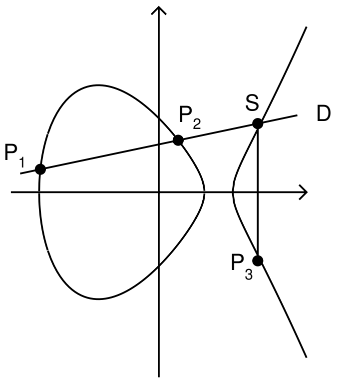
- $g=1$: points de E ; corde-et-tangente ;
- $g>1$: représentation de Mumford $(<u(z),v(z)>$ où u et $v\in\mathbb{F}_q[z]$ et $\deg(v)<\deg(u)\leqslant g)$; calcul de la loi de groupe par l'algorithme de Cantor.

Autres courbes :

- superelliptiques $Y^n = f(X)$ (Galbraith, Paulus, Smart), C_{ab} (Arita), e.g. $Y^3+5X^2Y-14XY+X^4-123=0$: un diviseur est représenté par un élément du groupe de classes du corps de fonction (HNF) ; loi de groupe par multiplication d'idéaux.
- Courbes modulaires (Frey, Müller).

Planche 20

Loi de groupe pour $g = 1$



$$P_3 = P_1 \oplus P_2$$
$$[k]P = \underbrace{P \oplus \cdots \oplus P}_{k \text{ times}}$$

Planche 21

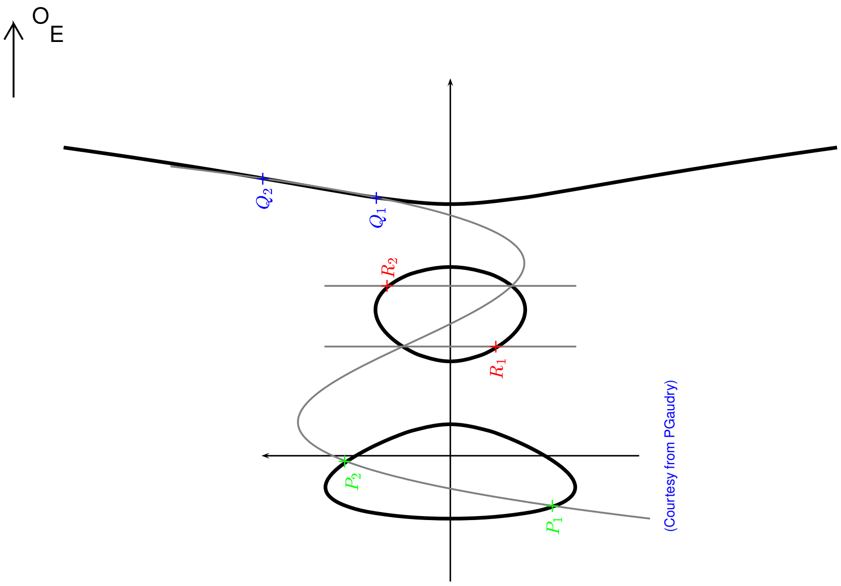


Planche 22

Calcul de la cardinalité

- $g = 1$: algorithme de Schoof-Elkies-Atkin (SEA) efficace pour p grand; algorithme de Satoh (1999) pour p petit $p \geqslant 5$, extension par Fouquet, Gaudry, Harley pour $p = 2$ (5 minutes sur une Alpha 750 MHz pour construire une bonne courbe aléatoire sur $\mathbb{F}_{2^{233}}$); des progrès sont à attendre.
- $g = 2$: Gaudry & Harley (ANTS IV) miment SEA autant que possible (utilisation de polynômes de division, Cartier-Manin). Le plus grand exemple traité est : $p = 10^{19} + 51$;

  $$y^2 = x^5 + 3141592653589793238\,x^4 + \cdots$$

  $$\#\mathrm{Jac} = 99999999982871020671452277000281660080,$$

  > 50 jours sur une Alpha 500 MHz.
- $g > 2$: Stein-Teske (1999) : $q = 10^8 + 7$, $g = 3$, $\#\mathrm{Jac} \approx 10^{24}$ en 10 heures sur un Pentium Pro/200 avec SIMATH.

Planche 23

Instances faibles

- Réduction $\mathrm{Jac}(C)/\mathbb{F}_q \hookrightarrow \mathbb{F}_{q^k}$ avec k petit :
  - Courbes supersingulières : MOV (Menezes, Okamoto, Vanstone avec le couplage de Weil); Frey & Rück (avec le couplage de Tate); Galbraith.
  - autres cas : courbes elliptiques tq $\#E = p - 1$ avec le couplage de Tate.
- LD dans des sous-groupes d'ordre p^e de $\mathrm{Jac}(C)/\mathbb{F}_{p^r}$ peut être résolu en temps polynomial : $g = 1$ (courbes anomales, $\#E = p$), par Satoh-Araki, Semaev, Smart; $g > 1$ par Rück.
- Weil-Koblitz : courbes sur $\mathbb{F}_q$ pour q petit et utilisées dans $\mathrm{Jac}(C)/\mathbb{F}_{q^k}$. ECLD est un peu plus facile.

Planche 24

LD sur les courbes elliptiques

Meilleure attaque connue (pour le moment) : $O(\sqrt{n})$ sur des courbes ordinaires ; à sécurité égale, on diminue la taille des clefs : 1024 de DH classique équivalent à 200 bits de courbes elliptiques.

Records de log discret : R. Harley *et al.* le 13/04/00 (09/99).

- ECC2K-108 (ECC2-97) proposé par Certicom (`http://www.certicom.com/chal`).
- 1300 (195) volontaires de 40 (20) pays avec quatre mois (40 jours) de calcul sur 9500 (740) ordinateurs.
- 200000 (16,000) MIPSY ou 500 ans de PC à 450 MHz, soit 25 fois plus que RSA-155 !

Planche 25

Résolution de LD quand $g > 1$

- Algorithme ADH dû à Adleman, DeMarrais, Huang (ANTS I) : $L_{p^{2g+1}}[1/2, c]$ avec $c \leqslant 2.181$ si $\log p \leqslant (2g+1)^{0.98}$ (heuristique en utilisant un théorème de Lovorn sur les polynômes friables); SNF.
- Flassenberg & Paulus : expériences avec $Y^2 = X^{2g+1} + 2X + 1$ en utilisant des cribles, plus rapide que Shanks pour $g \geqslant 6$.
- $Y^2 = X^{2g+2} + \cdots$ (Müller-Stein-Thiel) : prouvent $L_{p^{2g+2}}[1/2, 1.44]$.
- Extensions, analysise rigoureuse et optimisations par Enge : $L_{q^g}[1/2, c(\theta)]$ si $\theta \log q \leqslant g$, avec $\lim_{\theta \to 0} c(\theta) = +\infty$; SNF plus facile ; $c = \sqrt{2}$ par Enge et Gaudry.
- Variante de Gaudry : rapide, pratique, implantée, utilisée dans l'étape finale de la descente de Weil qui casse certains LD sur les courbes elliptiques (Frey; Gaudry, Hess, Smart; Menezes & Wu; Smart). Marche aussi pour les courbes C_{ab} (Arita), et sans doute sur des courbes plus générales. Les courbes de genre > 3 sont en danger !

Planche 26

Un exemple proposé par Buhler et Koblitz

Magma 2.3 sur Pentium II 450 MHz, 128 Mo.

corps	$\mathbb{F}_{5026243}$
équation	$y^2 + y = x^{13}$
genre	6
auto	13
$\#\mathrm{Jac}$	$\approx 10^{40}$
base	$193,485$
matrice	$165,778 \times 165,779$

construct. base	1638 s
relations	6 jours
SGE	1218
Lanczos	9 jours

Planche 27

IV. Vers des preuves de sécurité

RSA : casser RSA $\Leftarrow$ factoriser N ; $\Rightarrow$ est sans doute fausse : Boneh et Venkatesan ont montré que casser LE-RSA ne peut être équivalent à factoriser N (cohérent avec les attaques connues à la Coppersmith's attacks).

Rem. Variante de Rabin ($x \mapsto x^2 \bmod N$) pour laquelle la difficulté est équivalente à celle de la factorisation ; mais Rabin est très sensible à une attaque à texte chiffré choisi.

DH : casser DH $\Leftarrow$ résoudre le logarithme discret; $\Rightarrow$ presque vraie (Maurer).

Principe : on veut résoudre $a = g^x$ dans G de cardinal p premier. On suppose avoir un oracle :

$$\mathcal{O}(g^\alpha, g^\beta) = g^{\alpha\beta}.$$

Prop. On sait calculer g^{α^e} pour tout entier e, et plus généralement $g^{Q(\alpha)}$ pour toute fraction rationnelle $Q(X)$.

Planche 28

Soit $E/\mathbb{F}_p$ d'ordre friable, générateur $P = (u, v)$.

Prop. Étant donnés (g^{u_1}, g^{v_1}) et (g^{u_2}, g^{v_2}), avec (u_i, v_i) sur E, on peut calculer le couple (g^{u_3}, g^{v_3}) où $(u_3, v_3) = (u_1, v_1) + (u_2, v_2)$ sur E.

Hyp. : x est l'abscisse d'un point sur la courbe E.

On résout

$$g^{x^3+ax+b} = g^{y^2}$$

avec l'oracle. (Si $p \equiv 3 \bmod 4$, $\sqrt{y} = y^{(p+1)/4}$ qu'on peut calculer par l'oracle.)

E cyclique $\Rightarrow (x, y) = [k]P$. On utilise Pohlig-Hellman/Shanks pour trouver k avec l'oracle, d'où x.

Prop. On n'est pas capable de prouver l'existence de E, mais on peut généraliser à une jacobienne.

Rem. Généralisé par Boneh et Lipton.

Planche 29

Pourquoi RSA-NAIF doit être proscrit

RSA-NAIF n'est pas sémantiquement sûr : il est facile de distinguer RSA-NAIF("oui") et RSA-NAIF("non").

RSA-NAIF est malléable : $x^e \times y^e = (xy)^e$. Ex. : si $M < 2^m$ et $M = M_1 M_2$, $M_i < 2^{m/2}$, alors

$$M_1^e M_2^e \equiv C \bmod N \iff C/M_2^e \equiv M_1^e \bmod N.$$

RSA-NAIF ne résiste pas à une attaque à textes chiffrés choisis :

- Charlie intercepte $C = M^e \bmod N$;
- Charlie choisit r au hasard et fait signer $y = r^e C$ par Alice ;
- Alice envoie à Charlie la valeur $y^d = r^{ed} C^d = r C^d$ et celui-ci en déduit $C^d = M$.

Solution : ne chiffrer/signer que des messages formattés, par exemple $00 \cdots 00||M||11 \cdots 11$ ou bien $M||\mathcal{H}(M)$.

Attaques : Chaum *et al.*, M. Girault, J.-F. Misarsky ; Bleichenbacher ; Coron, Naccache, Stern, etc.

Planche 30

Module commun : (Simmons) N commun à tous les utilisateurs : si M est envoyé à deux utilisateurs avec $(e_1, e_2) = 1$, alors on trouve $ue_1 + ve_2 = 1$ et

$$(M^{e_1})^u (M^{e_2})^v \equiv M \bmod N.$$

Exposant commun : $C_i = M^3 \bmod N_i$ pour $i = 1, 2, 3$; on construit $C = M^3 \bmod N_1 N_2 N_3$; comme $M < N_i$, on en déduit $C = M^3$, d'où M. Généralisation à des polynômes $g_i(M)$ par J. Håstad.

Attaques sur la datation : si on envoie $M^e \bmod N$ et $(M + c)^e \bmod N$ avec c petit, on peut retrouver M.

Ex. (Franklin-Reiter) $C_1 \equiv M^3 \bmod N$, $C_2 \equiv (M + 1)^3 \bmod N$; alors :

$$\begin{cases} C_2 + 2C_1 - 1 &= 3M^3 + 3M^2 + 3M \\ C_2 - C_1 + 2 &= 3M^2 + 3M + 3 \end{cases}$$

d'où $M = (C_2 + 2C_1 - 1)/(C_2 - C_1 + 2) \bmod N$.

Thm. (Coppersmith) si $f(X)$ est de degré d, on peut trouver les solutions $< N^{1/d-\varepsilon}$ de $f(X) \equiv 0 \bmod N$ en temps polynomial en $\min(1/\varepsilon, \log N)$.

Planche 31

OAEP et consorts

Idée : prendre un système faiblement résistant et le transformer en un système résistant aux attaques à textes chiffrés choisis adaptatives.

OAEP : (Bellare & Rogaway)

Entrées :

- Algorithme public f, algorithme privé g opérant sur des chaînes $\in \{0,1\}^k$; $k_0 + k_1 < k$;
- Deux fonctions de hachage : $G : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$, $H : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$.
- L'algorithme chiffre $M \in \{0,1\}^n$, avec $n = k - k_0 - k_1$.

Chiffrement : choisir $r \in_R \{0,1\}^{k_0}$, puis calculer

$$\begin{aligned} s &= G(r) \oplus (M||0^{k_1}) \in \{0,1\}^{n+k_1} \qquad (*) \\ t &= H(s) \oplus r \in \{0,1\}^{k_0} \\ w &= s||t \in \{0,1\}^k \\ C &= f(w) \end{aligned}$$

Planche 32

Déchiffrement :

$$\begin{aligned} w &= g(C) \\ s &= w[0..n+k_1-1] \\ t &= w[n+k_1..k] \\ r &= H(s) \oplus t \\ z &= G(r) \oplus s \\ x &= z[0..n-1] \\ c &= z[n..n+k_1-1] \end{aligned}$$

Si $c = 0^{k_1}$, alors $M = x$, sinon rejeter C et ne pas retourner x.

Rem. Shoup a découvert une faille dans la preuve et a proposé OAEP+, qui remplace $(*)$ par :

$$s = (G(r) \oplus M)||H'(r||M).$$

Rem. RSA-OAEP est quand même sûr (Fujisaki, Okamoto, Pointcheval et Stern).

Planche 33

Conclusions

- Les preuves de sécurité vont devenir de plus en plus fréquentes et justes, détruisant toujours plus de systèmes.
- Nous ne disposons pas de beaucoup de problèmes difficiles . . .
- . . . pas plus que de nombreuses instances difficiles.
- Peu de groupes disponibles pour DH : nous devons regarder des objets mathématiques plus complexes, mais y en a-t-il beaucoup d'utilisables ?

Planche 34