Crypto Avancée

A. Bonnecaze

Institut de Mathématiques de Luminy (IML)

ESIL, 1er Semestre 2010

Contenu

- 6 séances
- 5 séances de TP-projet (pas d'exam ! ! ! !)
- Mais rendre le projet à la dernière séance
- Buts du projet
  1. Utiliser des grands nombres
  2. Travailler avec des courbes elliptiques
  3. Travailler dans des corps
  4. Etre capable de programmer des primitives cryptographiques

Rappels de crypto

- Applications cryptographiques
- Problèmes de Diffie-Hellman
- Echange de clés
- Chiffrement de Massey-Omura
- Chiffrement ElGamal

Introduction : Application cryptographiques

1. Chiffrement
   - Symétrique VS asymétrique
   - Chiffrement basé sur l'identité
   - Chiffrement avec recherche de mots clés
2. Authentification/Signatures
   - Signature courte
   - Signature en aveugle
   - Multisignature
   - Signature agrégée
   - Signature en anneau
   - Signature de groupe
3. Echange de clés
4. Fonction de hachage
5. Cryptographie à seuil

Problèmes de Diffie-Hellman

G_1 un groupe additif d'ordre q. P un générateur de G_1.

Données : (P, aP, bP), avec $a, b \in \mathbb{Z}_q^*$.
Sortie : abP.

La probabilité qu'un algorithme $\mathcal{A}$, à valeur $0/1$, probabiliste, résolve CDH en temps polynomial est définie par :

$$Succ_{\mathcal{A},G_1}^{CDH} = Prob[\mathcal{A}(P, aP, bP, abP) = 1 : a, b \in_R \mathbb{Z}_q^*].$$

Hypothèse CDH : Pour tout algorithme $\mathcal{A}$ à valeur $0/1$ et probabiliste, $Succ_{\mathcal{A},G_1}^{CDH}$ est négligeable.

Problème de Diffie-Hellman décisionnaire (DDH)

Données : (P, aP, bP, cP), avec $a, b, c \in \mathbb{Z}_q^*$.
Sortie : Oui si $c = ab \mod q$, Non sinon.

Le problème DDH est facile dans G_1. (voir attaque MOV)

La probabilité qu'un algorithme $\mathcal{A}$, à valeur $0/1$, probabiliste, résolve DDH en temps polynomial est définie par :

$$Succ_{\mathcal{A},G_1}^{DDH} = [Prob[\mathcal{A}(P, aP, bP, cP) = 1] - Prob[(P, aP, bP, abP) = 1] : a, b, c \in_R$$

Hypothèse DDH : Pour tout algorithme $\mathcal{A}$ à valeur $0/1$ et probabiliste, $Succ_{\mathcal{A},G_1}^{DDH}$ est négligeable.

Problème de Diffie-Hellman *weak* (W-DH)

Données : (P, Q, aP), avec $a \in \mathbb{Z}_q^*$.
Sortie : aQ.

Le problème W-DH n'est pas plus difficile que CDH.
Il existe bien d'autres versions de problèmes de Diffie-Hellman ...

Echange de clés Diffie-Hellman

- S'appuie sur le CDH
- Man in the middle !
- Généralisation ?

Chiffrement de Massey-Omura

- S'appuie sur le CDH
- Alice et Bob veulent communiquer sans avoir de clé privée.
  1. Ils se mettent d'accord sur les paramètre suivants (publics) : E sur $\mathbb{F}_q$ tel que le DLP est difficile dans $E(\mathbb{F}_q)$. $N = \#E(\mathbb{F}_q)$.
  2. Alice transforme son message en un point $M \in E(\mathbb{F}_q)$.
  3. Alice choisit un entier (secret) $m_a \in \mathbb{Z}_N^*$, calcule $M_1 = m_a M$, et envoie M_1 à Bob.
  4. Bob choisit un entier (secret) $m_b \in \mathbb{Z}_N^*$, calcule $M_2 = m_b M_1$, et envoie M_2 à Alice.
  5. Alice calcule $m_a^{-1} \in \mathbb{Z}_N^*$, puis $M_3 = m_a^{-1} M_2$ et envoie M_3 à Bob.
  6. Bob calcule $m_b^{-1} \in \mathbb{Z}_N^*$, puis $M_4 = m_b^{-1} M_3$ qui est le message M envoyé par Alice.

Chiffrement de ElGamal

Bob choisit un point P de E (l'ordre de P étant un grand nombre premier).
Il choisit un secret s et calcule $Q = sP$.
La clé publique de Bob est $E, \mathbb{F}_q, P$ et Q
La clé privée de Bob est s.
Afin d'envoyer un message à Bob, Alice doit

1. télécharger la clé publique de Bob.
2. Traduire son message en un point $M \in E(\mathbb{F}_q)$.
3. Choisir au hasard un entier k et calculer $M_1 = kP$.
4. Calculer $M_2 = M + kQ$.
5. Envoyer M_1 et M_2 à Bob.

1. Bob déchiffre en calculant $M = M_2 - sM_1$.
2. Si Eve sait résoudre le DLP, elle peut décrypter le message

TP/Projet : sujet 1

Le TP de cinq séances de deux heures consiste à programmer en C des primitives cryptographiques basées sur des courbes elliptiques.

- Faire connaissance avec la librairie GMP
- Programmer l'opposé d'un point, la loi d'addition, la loi de doublement, le multiple d'un point, pour des courbes de Weierstrass
- Voir DB de ARCANA pour trouver des courbes, prendre par exemple E256-001.gp
- Ecrire un programme donnant l'ordre d'un point
- Programmer le protocole Diffie-Hellman
- Programmer le chiffrement de Massey-Omura

TP/Projet : sujet 2

Soit la courbe d'équation
$y^2 + xy = x^3 + 1$
définie sur $F256 := F_2[x]/(P8)$
avec $P8 := x^8 + x^4 + x^3 + x^2 + 1$;
un polynôme primitif irréductible.

- Programmer l'opposé d'un point, la loi d'addition, la loi de doublement, le multiple d'un point
- Ecrire un programme donnant l'ordre d'un point
- Echange de clef : Alice et Bob veulent partager une clef secrète de 16 bits. Programmer un protocole permettant d'obtenir une telle clef.

Magma

En magma (http ://magma.maths.usyd.edu.au/magma/) la courbe se programme de la manière suivante (voir calculator et online help) :

```
PR<x>:=PolynomialRing(GF(2));
E:=EllipticCurve(x^3 + 1,x);
P8:=x^8 + x^4 + x^3 + x^2 + 1;
F256<a>:=ext<GF(2)|P8>;
E256:=BaseChange(E,F256);
Points(E256,a); // points ayant pour premier coef a
// [ (a : a^29 : 1), (a : a^194 : 1) ]
P1:=[a,a^29,1];
IsPoint(E256,P1);
//true (a : a^29 : 1)
P1:=E256!P1;
Order(P1); //48
```