Chiffrement homomorphe et exécution d'algorithmes sur des données chiffrées : avancées récentes

Carlos Aguilar Melchor

PICC, XLIM, Université de Limoges
carlos.aguilar@unilim.fr

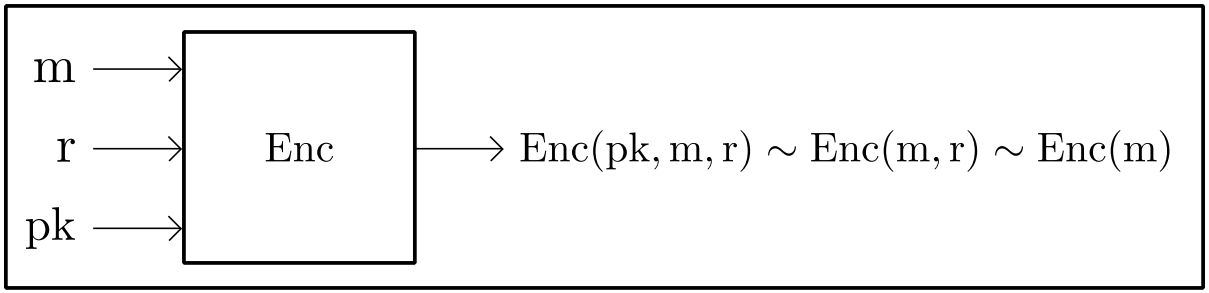
25 novembre 2011

Plan

Plan

Le chiffrement randomisé

Le chiffrement randomisé : beaucoup de chiffrés pour chaque clair



Exemple : Paillier $\mathtt{Enc}((\mathtt{N}, \mathtt{g}), \mathtt{m}, \mathtt{r}) = \mathtt{g}^{\mathtt{m}} \mathtt{r}^{\mathtt{N}} \bmod \mathtt{N}^2$

Indistingabilité pour tout couple de clairs [GM 82]

→ Les chiffrés ne donnent pas d'information sur le clair

→ On peut pas savoir si deux chiffrés correspondent au même clair

De l'impossibilité de chiffrer certaines données

Et si vous chiffriez votre boîte mail ?

Même dans le meilleur des mondes ...

- Comment filtrer le SPAM ?
- Comment rechercher des mails ?
- Comment traiter automatiquement les événements ?

Et si les algorithmes pouvaient "traverser" la couche de chiffrement ?

$$\mathbf{TriSPAM}(\texttt{message, SPAM}) \rightarrow \texttt{SPAM'}$$

$$\mathbf{TriSPAM}(\texttt{Enc(message), Enc(SPAM)}) = \texttt{Enc}(\mathbf{TriSPAM}(\texttt{message, SPAM})) \longrightarrow \texttt{Enc(SPAM')}$$

La conjecture de Rivest, Adleman et Dertouzos

On Data Banks and Privacy Homomorphisms [RAD 1978]

Il existe des systèmes de chiffrement ayant la propriété d'indistingabilité et des opérations $\mathtt{MUL}$ et $\mathtt{ADD}$ telles que:

- $\mathtt{MUL}(\mathtt{Enc}(x_1, r_1), \mathtt{Enc}(x_2, r_2), \mathtt{pk}) = \mathtt{Enc}(x_1 x_2, r_3)$
- $\mathtt{ADD}(\mathtt{Enc}(x_1, r_1), \mathtt{Enc}(x_2, r_2), \mathtt{pk}) = \mathtt{Enc}(x_1 + x_2, r_4)$

pour des sommes et produits *modulo p*.

Implications

f polynôme : $\mathtt{Enc}(x_1, r_1), \ldots, \mathtt{Enc}(x_n, r_n) \xrightarrow{f_H} \mathtt{Enc}(f(x_1, \ldots, x_n), r)$

$\mathcal{C}$ circuit : $\mathtt{Enc}(b_1, r_1), \ldots, \mathtt{Enc}(b_n, r_n) \xrightarrow{\mathcal{C}_H} \mathtt{Enc}(\mathcal{C}(b_1, \ldots, b_n), r)$.

(travaillant *modulo 2*, $a + b \leftrightarrow a \oplus b$ et $ab \leftrightarrow a \wedge b$)

Applications : chiffrer des données ou des services

Chiffrer des données

- Alice stocke chez Bob $\mathtt{Enc}(\mathtt{X})$.
- Alice demande à Bob d'exécuter la fonction $\mathtt{f}$ sur ses données.
- Bob:
  - transforme $\mathtt{f}$ en $\mathtt{f}_{\mathtt{H}}$;
  - calcule $\mathtt{f}_{\mathtt{H}}(\mathtt{Enc}(\mathtt{X})) = \mathtt{Enc}(\mathtt{f}(\mathtt{X}))$;
  - envoie éventuellement le résultat à Alice.

Exemples

- Stockage dans le *cloud*
- *Outsourcing* pour hardware de confiance

Applications : chiffrer des données ou des services

Chiffrer des services

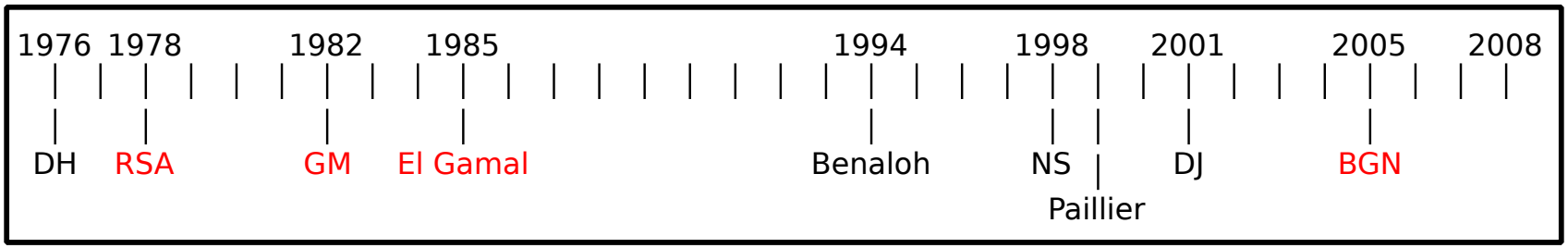
- Bob dispose d'un certain nombre de données $\mathtt{X}$.
- Alice veut que Bob évalue $\mathtt{f}$ sans dévoiler celle-ci.
- Alice envoie une version chiffrée de $\mathtt{f}$: $\mathtt{Enc}(\mathtt{f})$.
- Bob utilise une "fonction universelle" (i.e. tq $\mathtt{u}(\mathtt{f},\mathtt{X}) = \mathtt{f}(\mathtt{X})$) et :
  - transforme $\mathtt{u}$ en $\mathtt{u}_{\mathtt{H}}$;
  - calcule $\mathtt{u}_{\mathtt{H}}(\mathtt{Enc}(\mathtt{f}),\mathtt{X}) = \mathtt{Enc}(\mathtt{u}(\mathtt{f},\mathtt{X})) = \mathtt{Enc}(\mathtt{f}(\mathtt{X}))$

Exemples

- Éviter la fuite propre à une demande (en B2B comme en B2C)
- Exécution d'algorithmes sensibles sur des données sensibles

Plan

1978-2008 : utilisation de la théorie des nombres



30 ans de conjecture

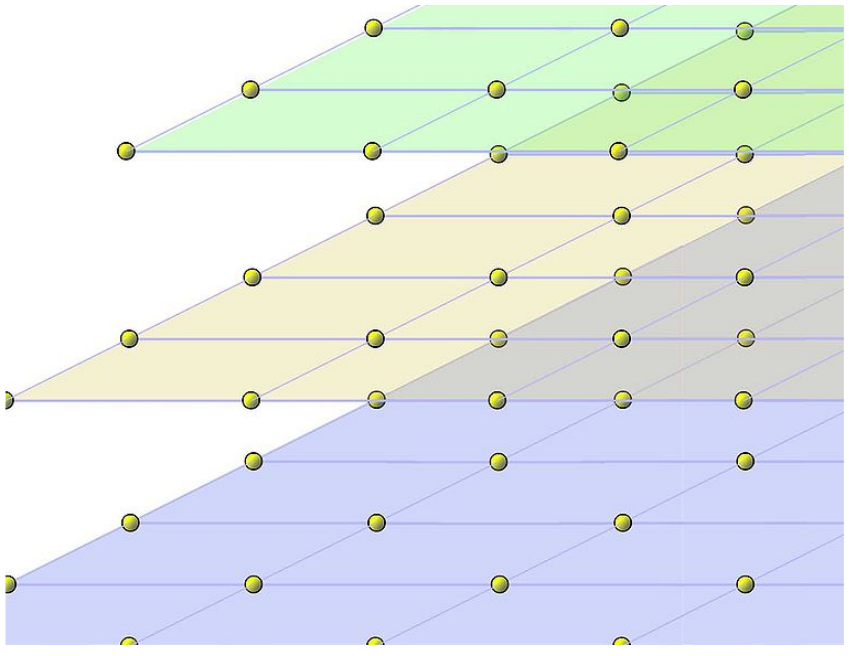
Très vite on découvre :

- un schéma tel que $\mathtt{MUL}(\mathtt{Enc}(x_1, r_1), \mathtt{Enc}(x_2, r_2), \mathtt{pk}) = \mathtt{Enc}(x_1 x_2, r_3)$ existe ;
- un schéma tel que $\mathtt{ADD}(\mathtt{Enc}(x_1, r_1), \mathtt{Enc}(x_2, r_2), \mathtt{pk}) = \mathtt{Enc}(x_1 + x_2, r_4)$ existe.

Aucun schéma de chiffrement tel que ADD et MUL existent en même temps.
La conjecture RAD78 est peut-être fausse !

Pendant ce temps ...

$$\begin{pmatrix} 1 & 1 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix} \times \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

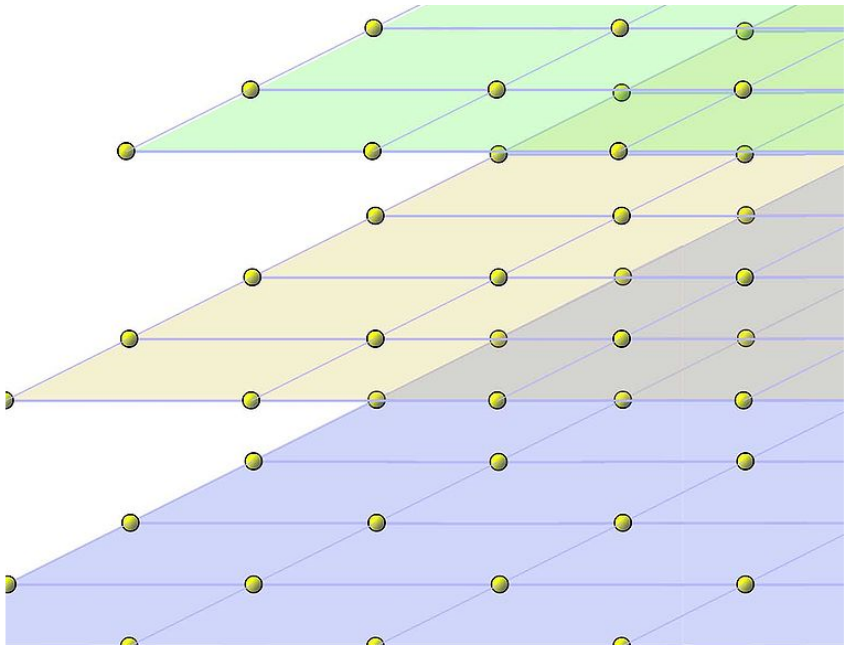


La cryptographie basée sur les réseaux euclidiens

Des preuves de sécurité extrêmement fortes
Des performances très prometteuses

Pendant ce temps ...

$$\begin{pmatrix} 1 & 1 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix} \times \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

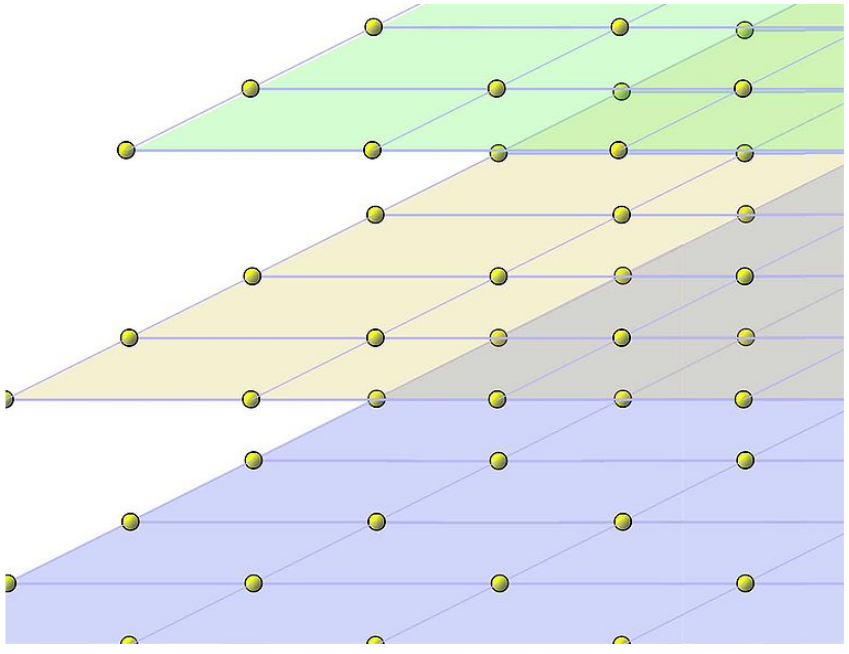


Tailles habituelles

Vecteurs de 100 ou 1000 colonnes
Scalaires modulo un entier, par exemple de 32 bits

Pendant ce temps ...

$$\begin{pmatrix} 1 & 1 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix} \times \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$



Cas classique

Chiffré : Vecteur proche d'un point du réseau choisi au hasard
Clair : Correction pour revenir au réseau (vecteur plus petit)

2008 : année charnière

Contexte

Les réseaux : sommes plus simples mais fortement limitées
$\Rightarrow$ Premiers schémas tels que `ADD` et `MUL` existent

LFHE : eprints de l'IACR [AMGH08]

+ Basé sur des problèmes standards des réseaux
– Paramètres dépendent des opérations : log(#S) exp(#P)

FHE : séminaire d'IBM Watson [G08]

+ FHE Paramètres indépendants du nombre d'opérations
– Basé sur des problèmes nouveaux : BDD, SSSP, circular security

2008 : année charnière

Contexte

Les réseaux : sommes plus simples mais fortement limitées
$\Rightarrow$ Premiers schémas tels que `ADD` et `MUL` existent

LFHE : eprints de l'IACR [AMGH08]

+ Basé sur des problèmes standards des réseaux
– Paramètres dépendent des opérations : log(#S) ~~exp(#P)~~ poly(#P)

FHE : séminaire d'IBM Watson [G08]

+ FHE Paramètres indépendants du nombre d'opérations
– Basé sur des problèmes nouveaux : BDD, SSSP, circular security

2008 : année charnière

Contexte

Les réseaux : sommes plus simples mais fortement limitées
$\Rightarrow$ Premiers schémas tels que `ADD` et `MUL` existent

LFHE : eprints de l'IACR [AMGH08] $\rightarrow$ CRYPTO'10

+ Basé sur des problèmes standards des réseaux
– Paramètres dépendent des opérations : log(#S) ~~exp(#P)~~ poly(#P)

FHE : séminaire d'IBM Watson [G08] $\rightarrow$ STOC'09

+ FHE Paramètres indépendants du nombre d'opérations
– Basé sur des problèmes nouveaux : BDD, SSSP, circular security

2009-11 : Multiplication des propositions

Propositions

Auteurs	Conférence	LFHE (simplification)	Coût LFHE	FHE ?
Smart Vercauteren	PKC'10	SPIP, PCP	$poly(d)$	SSSP, Circ.
van Djik et al.	EUROCRYPT'10	approx-GCD	$poly(d)$	SSSP, Circ.
Gentry et al.	EUROCRYPT'10	general-SIVP	$poly(d)$	
Gentry	CRYPTO'10	ideal-SIVP	$poly(d)$	SSSP, Circ.
Stehlé Steinfeld	ASIACRYPT'10	BDD	$poly(d)$	SSSP, Circ.
Coron et al.	CRYPTO'11	approx-GCD	$poly(d)$	SSSP, Circ.
Brakerski Vaik.	CRYPTO'11	ideal-SIVP	$poly(d)$	SSSP, Circ(?)
Gentry Halevi	FOCS'11	variable	$poly(d)$	Circ.
Brakerski Vaik.	FOCS'11	general-SIVP	$poly(d)$	Circ.
Brakerski et al.	ITCS'12	general/ideal-SIVP	$log^3(d)$!!!	Circ.

Plan

Évolution des tailles et des coûts

Théorie (Pbs standards : TRÈS SIMPLIFIÉ !!!)

	2008	2009	2010	2011	2012 ...
LFHE	2^d	d^5	d^3	d^3	$log^3(d)$
FHE	λ^7		$\lambda^{3.5}$	λ^2	$\lambda \vert polylog(\lambda)$

Pratique (ESTIMATION PERSONNELLE PEU FIABLE !!)

	2011	2012-13
LFHE	$T = 4500 \cdot d^3$, $Exp = d^2$	$T = 4500 \cdot log^3(d)$, $Exp = log^2(d)$
FHE	$T = 18 \cdot 10^9$, $Temps = 31$ min	$T = 8 \cdot 10^6$, $Temps = 10$ ms

Retour sur les applications : réinventer la crypto

Clé publique $\times 1000$? Facteur d'expansion $\times 100$?

Peu d'applications peuvent accepter de tels coûts ...

Le retour de la cryptographie hybride ...

Notre boîte mail est chiffrée par AES (clé : 128 bits, exp 1).
Peut-on stocker une clé publique et un chiffré homormophe ? OUI
Avant de lancer le calcul $\mathtt{f}_{\mathtt{H}}$ sur $\mathtt{Enc}_{\mathtt{AES}}(\mathtt{X})$ on fait :

$$\mathtt{Dec}_{\mathtt{AES},\mathtt{H}}(\mathtt{Enc}_{\mathtt{AES}}(\mathtt{X}), \mathtt{Enc}_{\mathtt{H}}(\mathtt{aesKey})) = \mathtt{Enc}_{\mathtt{H}}(\mathtt{Dec}_{\mathtt{AES}}(\mathtt{Enc}_{\mathtt{AES}}(\mathtt{X}), \mathtt{aesKey})).$$

On obtient donc $\mathtt{Enc}_{\mathtt{H}}(\mathtt{X})$!!!!!

L'expansion n'impacte pas sur le stockage ou les transmissions.

Peut-on admettre un coût *x*100 en mémoire et *x*1000 en calcul?

Perspectives

Conclusions

- Stockage et transmission sans facteur d'expansion
- Clés faciles à stocker
- Coûts en mémoire et calcul en $\times 100$ et $\times 1000$ (EN ÉTANT OPTIMISTE)
- Ne pas comparer aux autres clés (symétriques ou publiques) !

Perspectives

- Utilisation de hardware spécialisé (GPGPU) ?
- Encore des bonds en avant ? Possible mais pas indéfiniment ...

Questions

Merci de votre attention. Des questions ?