Confidentialité et intégrité des grandes bases de données

Yanli Guo & Stéphane Jacob

Équipes-projets SMIS & SECRET,
Centre de recherche INRIA Paris-Rocquencourt

6 octobre 2009

Plan

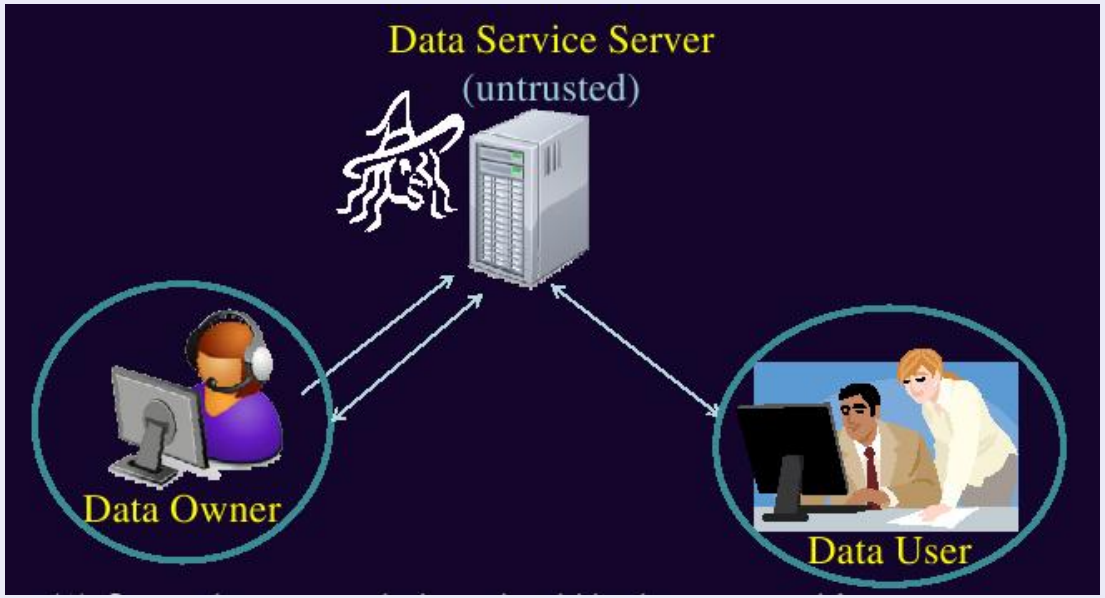
1. But et état de l'art
   - Contexte et but recherché
   - Modèles proposés
2. Étude d'un chiffrement à « déchiffrement rapide »
   - Présentation du chiffrement
   - Présentation de l'attaque

Plan

1. But et état de l'art
   - Contexte et but recherché
   - Modèles proposés
2. Étude d'un chiffrement à « déchiffrement rapide »
   - Présentation du chiffrement
   - Présentation de l'attaque

Contexte

Data As Service : données extériorisées



- Le client est sûr mais a de faibles capacités de calcul/stockage.
- Le serveur n'est pas sûr mais a une grosse capacité de calcul/stockage.

But recherché

Contraintes générales

Confidentialité : Prévenir les fuites d'information.

Intégrité : Détecter les altérations de données.

Performance : Ne pas détériorer significativement les performances.

Contraintes spécifiques aux bases de données

Le chiffrement ne doit pas « trop nuire » aux requêtes. Il doit donc respecter des problématiques de structures de base de données.

Plan

1. But et état de l'art
   - Contexte et but recherché
   - Modèles proposés
2. Étude d'un chiffrement à « déchiffrement rapide »
   - Présentation du chiffrement
   - Présentation de l'attaque

Différentes approches

Différentes parties peuvent être chiffrées :

- le contenu de la base de données,
- les index permettant les recherches dans la base,
- les relations entre les éléments de la base de données.

Exemples de chiffrements spécifiques

Chiffrement préservant l'ordre

Avantage : Requêtes simplifiées.

Problème : L'ordre est préservé.

Chiffrement préservant les préfixes

Avantage : Requêtes simplifiées.

Problème : Les préfixes sont préservés.

Exemples de chiffrements spécifiques

Chiffrement homomorphique

Premier chiffrement complètement homomorphique proposé par Craig Gentry (*Fully Homomorphic Encryption Using Ideal Lattices.* STOC 2009).

Avantage : Requêtes simplifiées.

Problème : Mise en œuvre impossible pour une sécurité raisonnable.

Exemples de chiffrements spécifiques

Chiffrement à comparaisons rapides [Ge-Zdonik 07]

Il permet de décider rapidement si deux données sont différentes. La comparaison de deux chiffrés part de l'octet de poids fort, procède octet par octet de gauche à droite et s'arrête dès qu'une différence est rencontrée ; c'est l'« Early Stopping ».
Cela est classiquement obtenu en cryptographie par des chiffrements à flot (par opposition aux chiffrements par blocs).

Stockage de la base de données

id	nom	dépenses de santé
1	Dupont	0
2	Dupond	42000
...	...	...

$\longrightarrow$

id	nom
1	Dupont
2	Dupond
...	...

id	dépenses de santé
1	0
2	42000
...	...

Chiffrement avec $(s_t, t \geq 0)_{\{K,\ \text{numéro de la page}\}}$

id	dépenses de santé
1	$0 \oplus s_0$
2	$42000 \oplus s_1$
...	...

Plan

1 But et état de l'art
- Contexte et but recherché
- Modèles proposés

2 Étude d'un chiffrement à « déchiffrement rapide »
- Présentation du chiffrement
- Présentation de l'attaque

Étude d'un chiffrement à « déchiffrement rapide »

T. Ge and S. Zdonik.
« Fast, secure encryption for indexing in a column-oriented DBMS ».
In *International Conference on Data Engineering - ICDE 2007*, pages 676–685. IEEE, 2007.

Modèle idéal de chiffrement

Schéma de chiffrement : r-FCE

Clair : suite d'octets m_i, divisée en pages de p octets.
Chiffré : suite d'octets c_i, divisée en pages de p octets.

La clef secrète K de k bits est commune à toute la base. À chaque page de clair est associée une permutation P aléatoire de $\{0, \ldots, p-1\}$.
Pour chiffrer une page :

Pour i de 0 à $p-1$:

1. $d_i = P(i) \bmod k$
2. $v_i = K_{\{d_i \to d_i+7\}}$ est l'octet de clef commençant au **bit** d_i :
$K_0, K_1, \ldots, \underbrace{K_{d_i}, \ldots, K_{d_i+7}}_{v_i = K_{\{d_i \to d_i+7\}}}, \ldots, K_{k-1}$
3. $c_i = m_i \oplus v_i$

Chiffrement de la base de données

Schéma de chiffrement : FCE

On remplace ici la permutation de $\{0, \ldots, p-1\}$ par :

$$P(x) = ax^3 + bx^2 + cx + d \bmod p \text{ avec } a, b, c, d \in [0, p-1],$$

où a, b, c et d sont dérivés de K et du numéro de page j par un chiffrement par bloc classique, par exemple $\text{AES}_{K_{\{128j \to 128j+127\}}}(j)$.

Paramètres :

- taille de clef $k = 2^\kappa = 2^{15}$ bits,
- taille de page $p = 2^{\kappa+1} = 2^{16}$ octets,
- taille de a, b, c, d : 64 bits par page de 64Ko.

Chiffrement de la base de données

Schéma de chiffrement : FCE

Clef K de 2^κ bits, commune à toute la base.

Chiffrement d'une page :

Pour chaque page j de clair :

1. $(a, b, c, d) \leftarrow \text{AES}_{K_{\{128j \to 128j+127\}}}(j)$ avec $a, b, c, d \in [0, 2^{\kappa+1} - 1]$
2. $P(x) = ax^3 + bx^2 + cx + d \bmod 2^{\kappa+1}$

Pour i de 0 à $2^{\kappa+1} - 1$:

1. $d_i = P(i) \bmod 2^\kappa$
2. $c_i = m_i \oplus K_{\{d_i \to d_i+7\}}$

Plan

1 But et état de l'art
- Contexte et but recherché
- Modèles proposés

2 Étude d'un chiffrement à « déchiffrement rapide »
- Présentation du chiffrement
- Présentation de l'attaque

Attaque

Attaque à clair connu

- Entrée : une demi-page de clair (2^κ octets m_i) et son chiffré (2^κ octets c_i).
- Résultat : la clef K (et donc aussi les permutations de toutes les pages).

Attaque

Étape 1 : récupération de la suite chiffrante

$$\forall i \in \{0, \ldots, 2^\kappa - 1\}\,,\ v_i = m_i \oplus c_i$$

Coût : 2^κ XORs.

On connaît donc l'ensemble des octets de K commençant au bit $P(0), P(1), \ldots, P(2^\kappa - 1)$ avec $P(x) = ax^3 + bx^2 + cx + d \bmod 2^{\kappa+1}$, mais pas leur agencement.

Clefs équivalentes.

$\forall\ d'$, les clefs $\begin{cases} K' = K \ggg d' \\ P'(x) = ax^3 + bx^2 + cx + (d - d') \end{cases}$ sont équivalentes.

Donc, on va chercher $\widetilde{K} = K \ggg d$.

Attaque

Objectif : trouver (a, b, c) à partir des $\widetilde{K}_{\{\widetilde{P}(i)\to\widetilde{P}(i)+7\}} = K_{\{P(i)\to P(i)+7\}}$ où $\widetilde{P}(i) = ai^3 + bi^2 + ci \bmod 2^{\kappa+1}$.

Principe : chercher (α, β, γ) antécédents de $(1, 2, 3)$ par $\widetilde{P}$. Pour chaque triplet (α, β, γ) possible, on cherche, dans l'anneau $\mathbb{Z}/2^\kappa\mathbb{Z}$, une solution du système de Vandermonde suivant :

$$\begin{pmatrix} \alpha^3 & \alpha^2 & \alpha \\ \beta^3 & \beta^2 & \beta \\ \gamma^3 & \gamma^2 & \gamma \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + \begin{pmatrix} \lambda_1 2^\kappa \\ \lambda_2 2^\kappa \\ \lambda_3 2^\kappa \end{pmatrix}$$

Remarque

- α et γ sont nécessairement impairs.
- Sur les 2^{43} systèmes possibles avec $(\alpha, \beta, \gamma) \in \{0, \ldots, 2^\kappa - 1\}^3$ et α et γ impairs, environ 2^{27} ont une solution dans l'anneau $\mathbb{Z}/2^{15}\mathbb{Z}$.

Attaque

Étape 2 : réduction du nombre de systèmes à considérer

Si $\widetilde{P}$ est la permutation recherchée, v_0 donne de l'information sur les α tels que $\widetilde{P}(\alpha) = 1$:

$$v_0 = \widetilde{K}_0, \underbrace{\widetilde{K}_1, \ldots, \widetilde{K}_7}$$

$$\overbrace{\widetilde{K}_{\widetilde{P}(\alpha)}, \ldots, \widetilde{K}_{\widetilde{P}(\alpha)+6}}, \widetilde{K}_{\widetilde{P}(\alpha)+7}$$

$$\mathcal{E}_1 = \left\{\alpha \text{ impair} | 0 \leq \alpha \leq 2^\kappa - 1 \text{ et } \left(\widetilde{K}_{\{\widetilde{P}(\alpha) \to \widetilde{P}(\alpha)+6\}}\right) = \left(\widetilde{K}_{\{1 \to 7\}}\right)\right\}$$

$$\mathcal{E}_2(x) = \left\{\beta | 0 \leq \beta \leq 2^\kappa - 1 \text{ et } \left(\widetilde{K}_{\{\widetilde{P}(\beta) \to \widetilde{P}(\beta)+6\}}\right) = \left(\widetilde{K}_{\{2 \to 7\}}, x\right)\right\}$$

$$\mathcal{E}_3(x, y) = \left\{\gamma \text{ impair} | 0 \leq \gamma \leq 2^\kappa - 1 \text{ et } \left(\widetilde{K}_{\{\widetilde{P}(\gamma) \to \widetilde{P}(\gamma)+6\}}\right) = \left(\widetilde{K}_{\{3 \to 7\}}, x, y\right)\right\}$$

Attaque

On a : $|\mathcal{E}_1| \sim 2^7$, $|\mathcal{E}_2(x)| \sim 2^7$ et $|\mathcal{E}_3(x,y)| \sim 2^5$.

Le coût du calcul de ces ensembles est de 2^κ masques et comparaisons.

Attaque

Étape 3 : sélectionner des (a, b, c) qui peuvent convenir

Pour tout $\alpha \in \mathcal{E}_1$
  $x_\alpha \leftarrow$ dernier bit de v_α
  Pour tout $\beta \in \mathcal{E}_2(x_\alpha)$
    $y_\beta \leftarrow$ dernier bit de v_β
    Pour tout $\gamma \in \mathcal{E}_3(x_\alpha, y_\beta)$
      Si le système a une solution (a, b, c) dans $\mathbb{Z}/2^\kappa\mathbb{Z}$
        $\mathcal{L} \leftarrow \mathcal{L} \cup \{(a, b, c)\}$.

Coût de la création de $\mathcal{L}$: 2^{19} résolutions de systèmes 3×3 qui se résument au pire à 23 multiplications.
Taille de $\mathcal{L} \sim 8$ triplets (a, b, c).

Attaque

Étape 4 : réduction maximale de $\mathcal{L}$

Pour chaque $(a, b, c) \in \mathcal{L}$, on construit le $\widetilde{K}$ correspondant.

Pour $d \in \{0, \ldots, 2^\kappa - 1\}$

$K \leftarrow \left(\widetilde{K} \ggg d\right)$

On calcule le polynôme dérivé de la clef K par AES et on le compare à $\widetilde{P}$.

Coût : 2^κ AES.

Attaque

Complexité

Le coût de l'attaque est donc, en moyenne :

$$2^{15} \text{ AES} + 2^{23} \text{ multiplications sur } 16 \text{ bits.}$$

Avec seulement une demi-page de clair/chiffré, on est donc capable de retrouver la clef K et l'ensemble des permutations en très peu de temps.

Conclusion

Il reste de nombreux problèmes du point de vue cryptographique pour le chiffrement des bases de données.

- Comment utiliser correctement un chiffrement à flot dans ce contexte, quelle IV utiliser ?
- Comment chiffrer en gardant une structure permettant une recherche efficace ?
- Comment protéger l'intégrité (CCM...) ?