# Cryptographie dans les bases de données

Nicolas Anciaux, SMIS project
INRIA Paris-Rocquencourt - France

Nicolas.Anciaux@inria.fr

Slides from: N. Anciaux, L. Bouganim, P. Pucheral, A. Canteaut

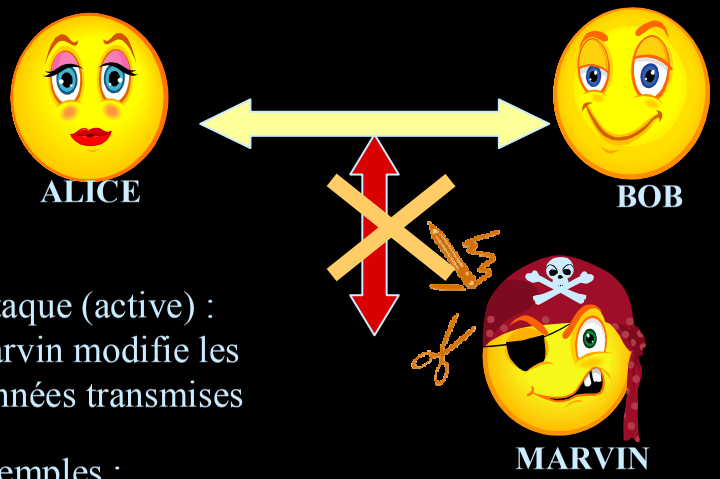
# Plan

- Outils cryptographiques
- Application du chiffrement à une base de données
- L'approche serveur
- L'approche client
- L'approche par matériel sûr (exotique…)
- Conclusion

# Cryptographie : intégrité et confidentialité

## Protection de l'intégrité:

- Hachage des messages

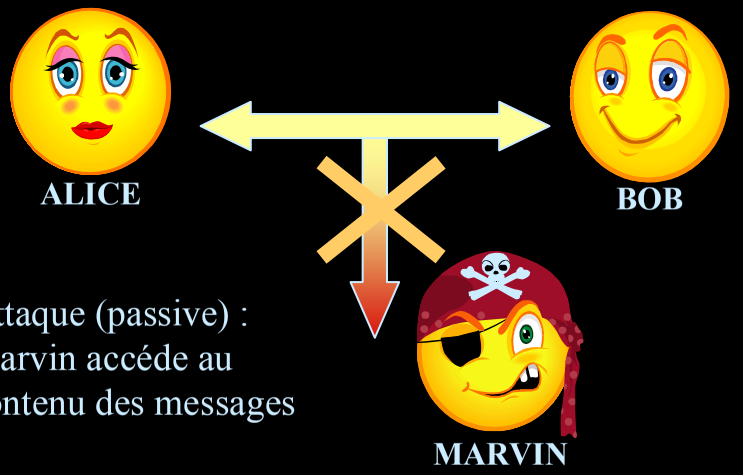



Attaque (active) :
Marvin modifie les données transmises

Exemples :
1. Vol d'identité
2. Altération
3. Forgerie
4. Rejeu
5. Repudiation
6. Destruction
7. Retard/réordonnancement

## Protection de la confidentialité:

- Chiffrement des messages




Attaque (passive) :
Marvin accéde au contenu des messages

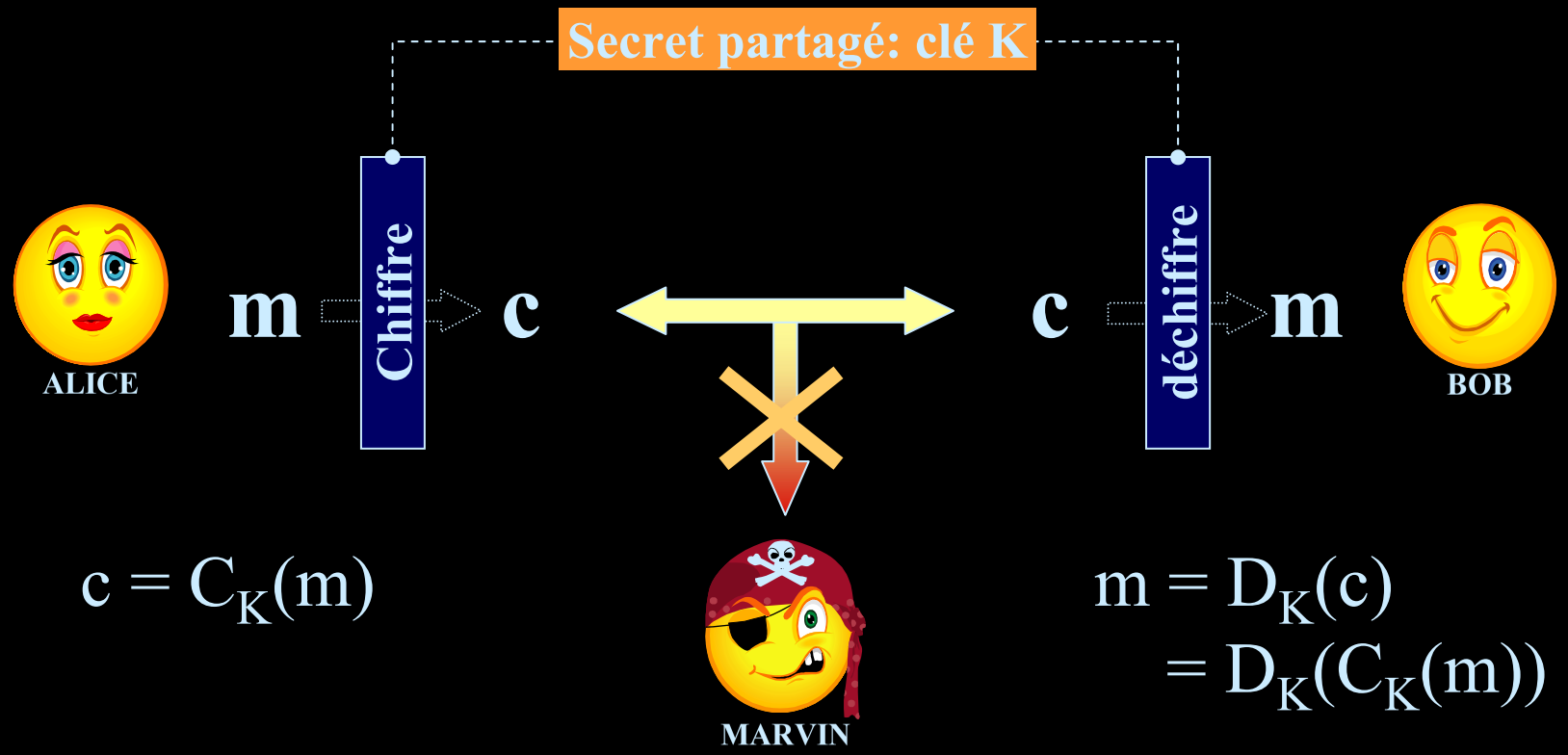
# Hachage cryptographique

- Une fonction de hachage cryptographique
  - transforme un message de taille quelconque en une empreinte de taille fixe;
  - doit être facile à calculer.
- … de plus, elle a les propriétés suivantes:
  - La fonction est difficile à inverser (à sens unique)
  - Il est difficile de trouver 2 messages de même empreinte (sans collision)
- Ainsi, toute modification du message génère une modification de l'empreinte
- Exemples d'algorithmes
  - Message Digest 5 (MD5), résultat sur 128 bits
  - Secure Hash Algorithm 1 (SHA-1), résultat sur 160 bits
- La fonction de hachage peut utiliser une clé cryptographique (authentification)

  … et/ou être combinée avec une fonction de chiffrement

# Chiffrement Symétrique (“à clé secrête”)

- Auteur et destinataire des messages partagent un **secret**
- Le secret (clé) permet de chiffrer et le déchiffrer les messages
- **La sécurité repose sur ce seul secret (tout le reste est publique)**




$c = C_K(m)$

$m = D_K(c)$
$= D_K(C_K(m))$

# Algorithmes et attaques

## Des algorithmes symétriques par bloc …. …. et des attaques

- DES
  - Data Encryption Standard (1976 - 1997)
  - Chiffrement par bloc de 64 bits
  - Chiffrement/déchiffrmeent = même algorithme
  - La clé fait 56 bits
  - 1997: 39 jours sur 10 000 Pentium
  - 1998: une clé DES cassée en 56h (pour 250 000 $)
  - 2007: 6.4 jours sur une machine parallèle ($10,000)
- 3DES
  - Remplace DES entre 1997 et 2001
  - Necessite 3 clés de 56 bits
  - $3DES(k_1k_2k_3, M) = DES(k_3,DES(k_2,DES(k_1,M)))$
  - La meilleure attaque connue necessite $2^{32}$ messages clairs connus, $2^{113}$ étapes, $2^{90}$ chiffrements DES, et $2^{88}$ mémoire !!
  - 3DES est sûr (actuellement)
- RIJNDAEL (AES)
  - Utilisé depuis 2001 (standard depuis 2002)
  - Chiffrement par bloc de 128 bits
  - La clé fait 128, 192 ou 256 bits
  - Rapide, necessite peu de mémoire
  - Seules des attaques canal latéral ont été réussies sur AES
  - Voir www.cryptosystem.net/aes/ pour information
  - AES est sûr (actuellement)

➔ Puis-je chiffrer avec 3DES ou AES sans problème ?

# Plan

- Outils cryptographiques
- Application du chiffrement à une base de données
- L'approche serveur
- L'approche client
- L'approche par matériel sûr (exotique…)
- Conclusion

# Application du chiffrement au contexte BD

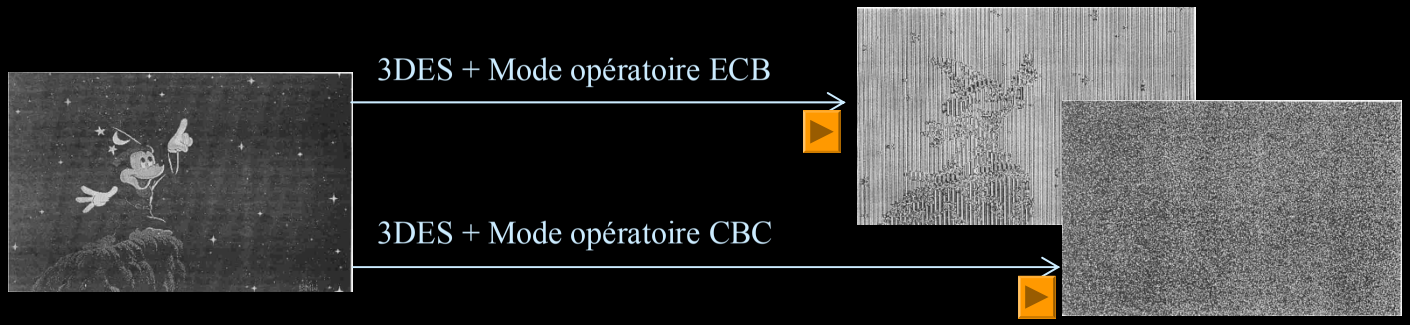
- **Si je chiffre une BD avec un algorithme sûr, le résultat est-il sûr ?**

- Les algorithmes de chiffrement résistent aux attaques suivantes:
  - Meme si l'attaquant connait le texte chiffré c, il ne peut ni trouver le message m en clair, ni trouver la clé K
  - Meme si l'attaquant connait des couples clair/chiffré (m, c), il ne peut ni trouver la clé K, ni obtenir d'autres messages en clair
- … mais pas toujours leur mise en oeuvre (mode opératoire/protocole)

- Le contexte BD a des spécificités difficiles à prendre en compte
  - Gros volume de données
  - Motifs répétés, distribution qui peuvent être connues
  - Données modifiables

# Mauvais choix du mode opératoire

- Mode ECB => attaque par analyse de fréquence
  - ECB : Motif en clair identiques => motif chiffré identique




- Mode CTR => attaque par comparaisons successives
  - CTR : m XOR m' = $D_K(m)$ XOR $D_K(m)$ => information sur le contenu des MAJ !

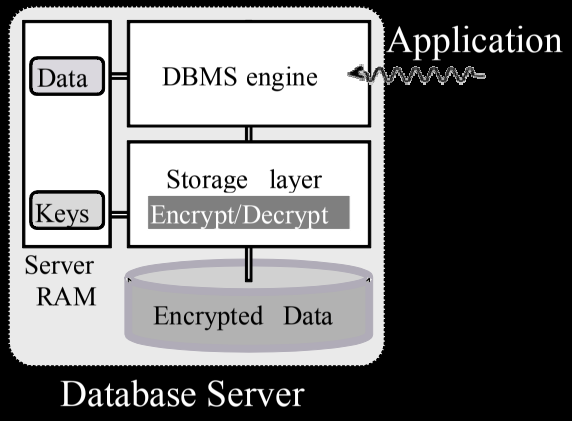
➔ **Si je chiffre une BD avec un algorithme sûr, le résultat est-il sûr ? NON…**

➔ Les spécificités du contexte BD doivent être prises en compte…

➔ Quid des "concessions" faites à la sécurité pour raisons de perf.?

# Chiffrer à quel niveau ? (1/3)

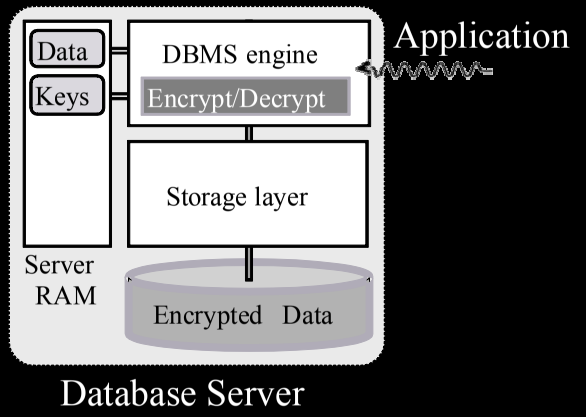
- Chiffrement niveau OS (chiffrement fichiers/stockage)




- ✚ Transparent pour le SGBD et l'application
- ━ … mais chiffrement non sélectif ➔ des limites
  - Chiffrement non lié au droits d'accès (1 clé par privilège)
  - Chiffrement partiel proscrit (=> performances?)

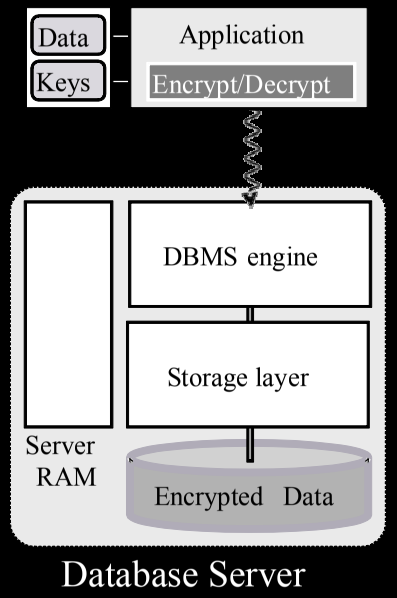
# Chiffrer à quel niveau ? (2/3)

- Chiffrement niveau SGBD
  - ✚ Chiffrement sélectif (spécifique)
    - Chiffrer selon les privilèges utilisateur
    - Chiffrer les données les plus sensibles
      - au niveau table, ligne, colonne…
      - … de façon conditionnelle (salaire >10K)
  - ✚ Transparence pour l'application
  - ━ … mais mécanismes internes SGBD à revisiter
    - Evaluation de requête + indexation sur des données chiffrées impossible
      - sauf chiffrement à propriétés particulières (préservant égalité ou l'ordre => dangereux pour la sécurité)
    - Surtout dans un contexte où le serveur n'est pas de confiance (approche client)
  - ━ … et problème de performance en perspective



# Chiffrer à quel niveau ? (3/3)

- Chiffrement niveau application
  - ✚ Résistance aux attaques internes
  - – Aucune transparence pour l'application
    - L'application pilote de chiffrement/déchiffrement
      - Et gère les clés…
    - Le SGBD ne peut traiter les données déchiffrées
      - Au risque d'attaques internes
  - – Dégradation importante des performances
    - L'application prend à sa charge une (grande) partie de l'évaluation de requête
  - – Le client peut attaquer les droits d'accès
    - Les données et les clés sont en clair sur le client




Conclusion : chiffrement dans le SGBD ☺

# Chiffrement au niveau SGBD : approches

- Le SGBD protège l'empreinte disque contre:
  - Observation de données
  - Altération des données
  - Substitution des données
  - Rejeu d'anciennes version des données

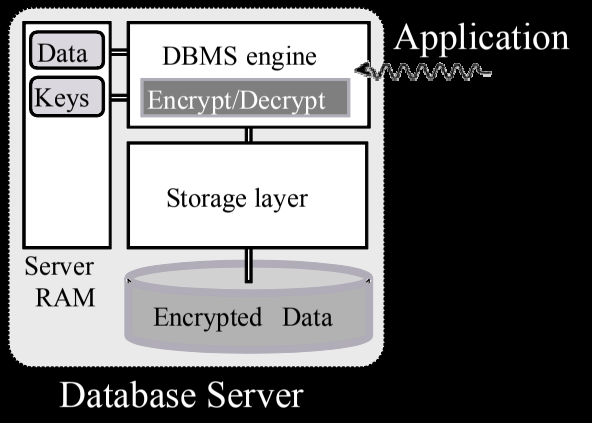
- Différentes approches:
  - Approche serveur :
    - Le serveur protège les données (Chiffrement/déchiffrement/hachage)
  - Approche client
    - Le client protège les données
  - Approches à base de matériel sûr (MS)
    - Le MS protège les données

# Plan

- Outils cryptographiques
- Application du chiffrement à une base de données
- L'approche serveur
- L'approche client
- L'approche par matériel sûr (exotique…)
- Conclusion

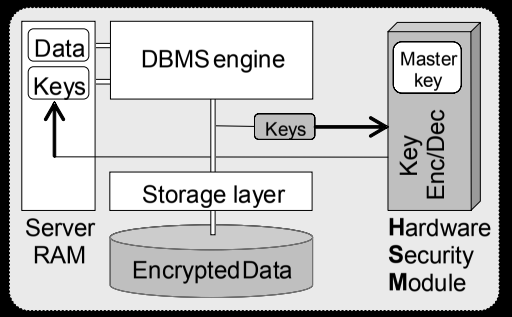
# L'approche serveur : principe

- Le serveur chiffre/déchiffre/hache les données
  - Les clés sont détenues par le serveur
  - Les données sont déchiffrées lors de l'évaluation des requêtes
- La BD est protégées sur le support de stockage persistant
  - Le niveau de protection des données n'excède pas celle celui des clés, stockées sur le serveur
  - Solution Oracle: les clés sont chiffrées par une *master key* chiffrée elle-même avec un mot de passe administrateur (*wallet*)
- Faiblesses : attaques internes
  - Attaque administrateur (le DBA a tous les droits)
    - Peut accéder aux clés/données et utiliser la base
    - Ceci sans laisser de traces

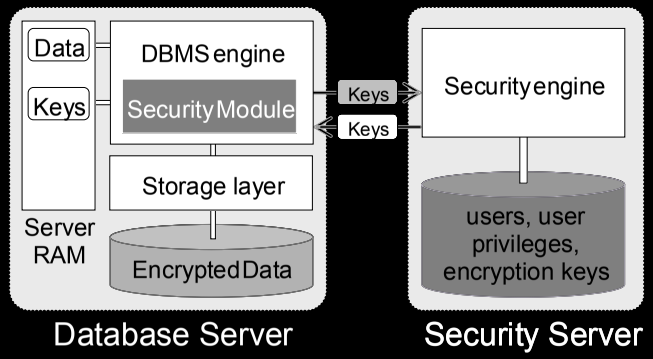


# L'approche serveur : gestion des clés

- Techniques réduisant au maximum
  - l'exposition des clés
  - Et les droits de l'administrateur
- Introduction d'un dispositif sécurisé (HSM, Hardware Security Module)
  - ✚ Les clés ne sont accessible que pendant l'exécution
  - Elles sont chiffrés avec une « master key », dans le HSM
  - ━ … mais les clés restent exposées (brièvement)
  - … et les données sont en clair lors de l'exécution




- Introduction d'un serveur de sécurité (serveur distinct)
  - Un 2ème serveur gère les clés et droits d'accès
    - 2ème Admin : Grant/revoke/create user…
  - ✚ Résiste « mieux » aux attaques internes
    - Le DBA/troyan ne peut pas observer l'empreinte de la BD
  - ━ Mais les clés/données restent en clair à l'exécution



# Approche serveur: solutions commerciales (1)

- L'exemple d'Oracle
  - DBMS Obfuscation Toolkit (8i) :
    - Solution de chiffrement niveau application
    - À base de procédures stockées (chiffrement/déchiffrement/hachage)
  - Transparent Data Encryption (TDE) :
    - Solution de chiffrement niveau SGBD
    - SQL étendu à la gestion du chiffrement
    - Master key : chiffrée par un mot de passe (admin.) ou stockée dans un HSM
    - Chiffrement niveau attribut (10g)
      - Colonnes sensibles chiffrées : dans les tablespaces (même temporaires), SGA, logs/backups…
      - Indexation des prédicats d'égalité (chiffrement NO_SALT) ➔ attaque par analyse de fréquence !
    - Chiffrement niveau Tablespace (11g)
      - Tablespaces complets chiffrés sur disque, déchiffré en SGA
      - Indexation classique (indexes fabriqués sur le clair)
- SQL server 2008 TDE : similaire à Oracle TDE / chiffrement niveau Tablespace

# Approche serveur: solutions commerciales (2)

- Protegrity Secure.Data
  - Basé sur une solution de chiffrement serveur
    - Pour tout SGBD : Oracle, SQL Server, IBM DB2 …
  - Ajout d'un serveur de sécurité
    - Secure.Manager : gestion des utilisateurs, droits et clés
    - Secure.Server : module de chiffrement intégré au noyau SGBD
  - … et 2 personnes physiques différentes …
    - Database Administrator (DBA) / Security Administrator (SA)
- IBM DB2 Data Encrytpion Expert : similaire à Protégrity

# L'approche serveur : conclusion

- L'empreinte de la BD est chiffrée
  - Attention au concession sur la sécurité pour les performance (indexation)
- Les clés sont exposées au minimum
  - HSM => exposition brève lors de l'exécution
- Les droits de l'administrateurs sont réduits au maximum
  - Serveur de sécurité => la DBA n'a plus « tous » les droits
- Mais :
  - Données en clair lors de l'exécution
  - Clés détenues ou transmises au serveur
- Limite de l'approche serveur
  - Le serveur n'est pas un site de confiance
  - Vulnérable aux attaques internes (administrateur / insider / troyan)

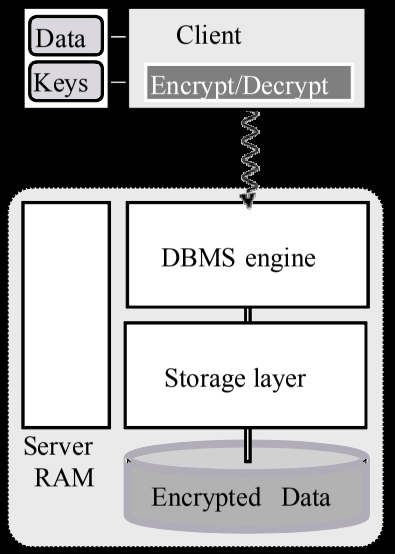
# Plan

- Outils cryptographiques
- Application du chiffrement à une base de données
- L'approche serveur
- L'approche client
- L'approche par matériel sûr (exotique…)
- Conclusion

# L'approche client : principe

- Chiffrement coté client
  - Pas de transmission du clair ni des clés au serveur
  - Le traitement s'effectue sur le client (pire cas)
- La BD est protégée coté serveur
  - Le serveur résiste aux attaque internes
  - Mais… dégradation très importante des performances

  ➔ Problème : déporter la majeure partie du traitement sur le serveur (données chiffrées), sans perte de sécurité
- Limites de l'approche client :
  - Gestionnaire de droit côté client
  - Données et clés en clair sur le client
  - Or le client n'est pas forcément un site de confiance
  - Donc ne convient pas à une BD partagée (BD privée uniquement)

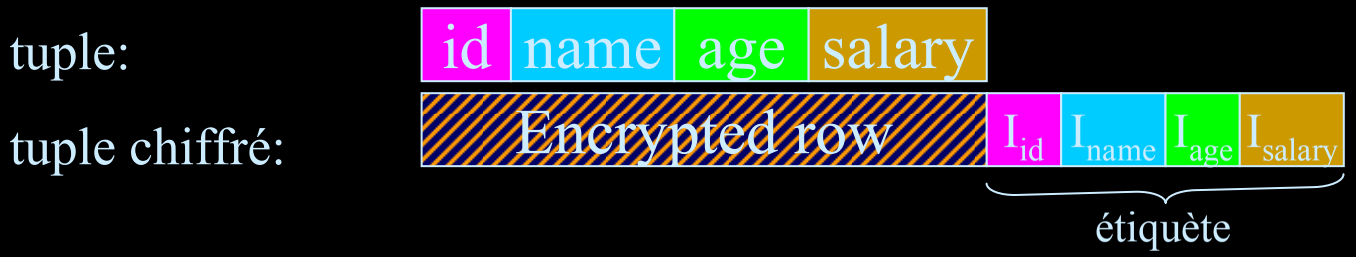


# Réponses au problème de performance

- Indexation des données
  - Indexation (index traditionnel chiffré)
    - Ex. Le client maintient et utilise (traverse) un B+-Tree [DDJ+03]
  - Etiquetage des tuples
    - Ex. Le client pose des étiquettes, pour sélectionner/joindre [HIL+02]
- Modèle de chiffrement
  - … indexer des données chiffrées ne sert à rien ☺
  - … traiter directement les données chiffrées est impossibles ☺
  - SAUF SI: on dispose de chiffrement à propriétés particulières
    - Ex. Préservant l'égalité [Ora07, BoP02], l'ordre [AKS+04], ou homomorphiques [GeZ07]
    - Compromis sécurité / performance…

# Solution par étiquetage [HIL+02]

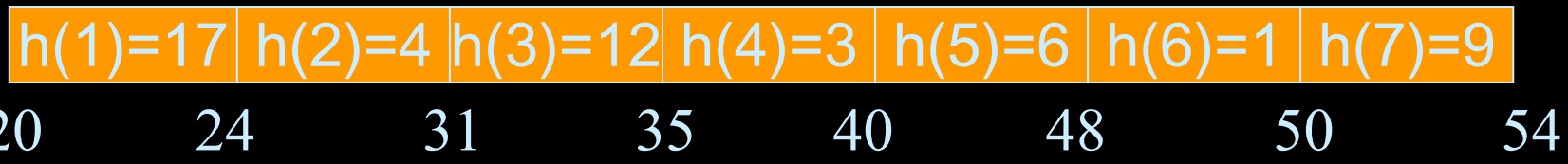
- Granule de chiffrement = tuple
- Ajout d'étiquettes d'attributs
  - Indique l'appartenance d'un attribut d'un tuple à une plage de valeurs
  - Permet des traitements approximatifs sur le serveur
    - Sélection, jointure, groupement



# Attributs numériques

- Partitionner le domaine de variation d'un attribut
  - Chaque partition contient un nombre identique de tuples

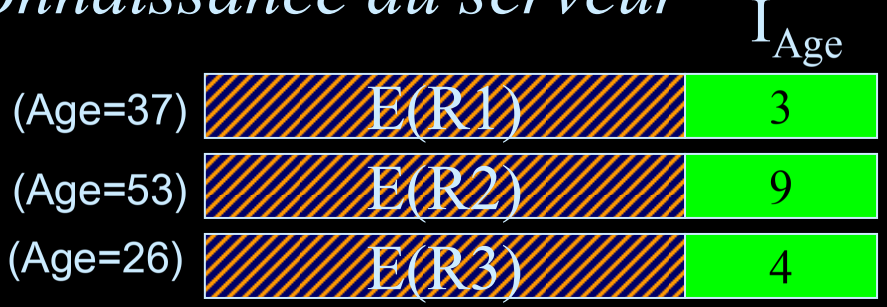
*Connaissance du client*




32<Age<40

Age=53

*Connaissance du serveur*




$I_{Age} = 12$ or $I_{Age} = 3$

$I_{Age} = 9$

# Attributs « String »

- Signatures de string (n-grams)

*Connaissance du client*

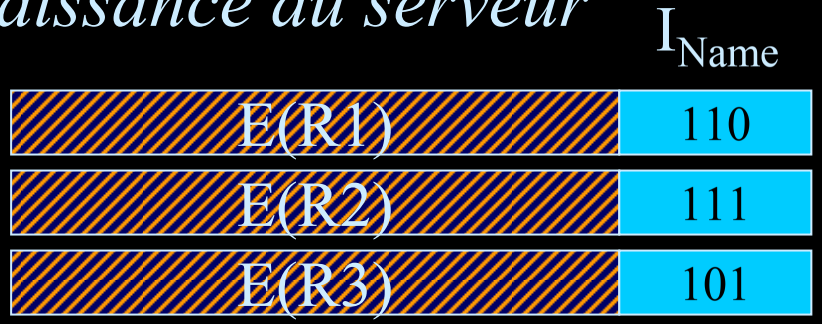
N={"g", "re", "ma"}

| string | signature |
|---|---|
| 'Greencar' | 110 |
| 'Bigrecordman' | 111 |
| 'Bigman' | 101 |

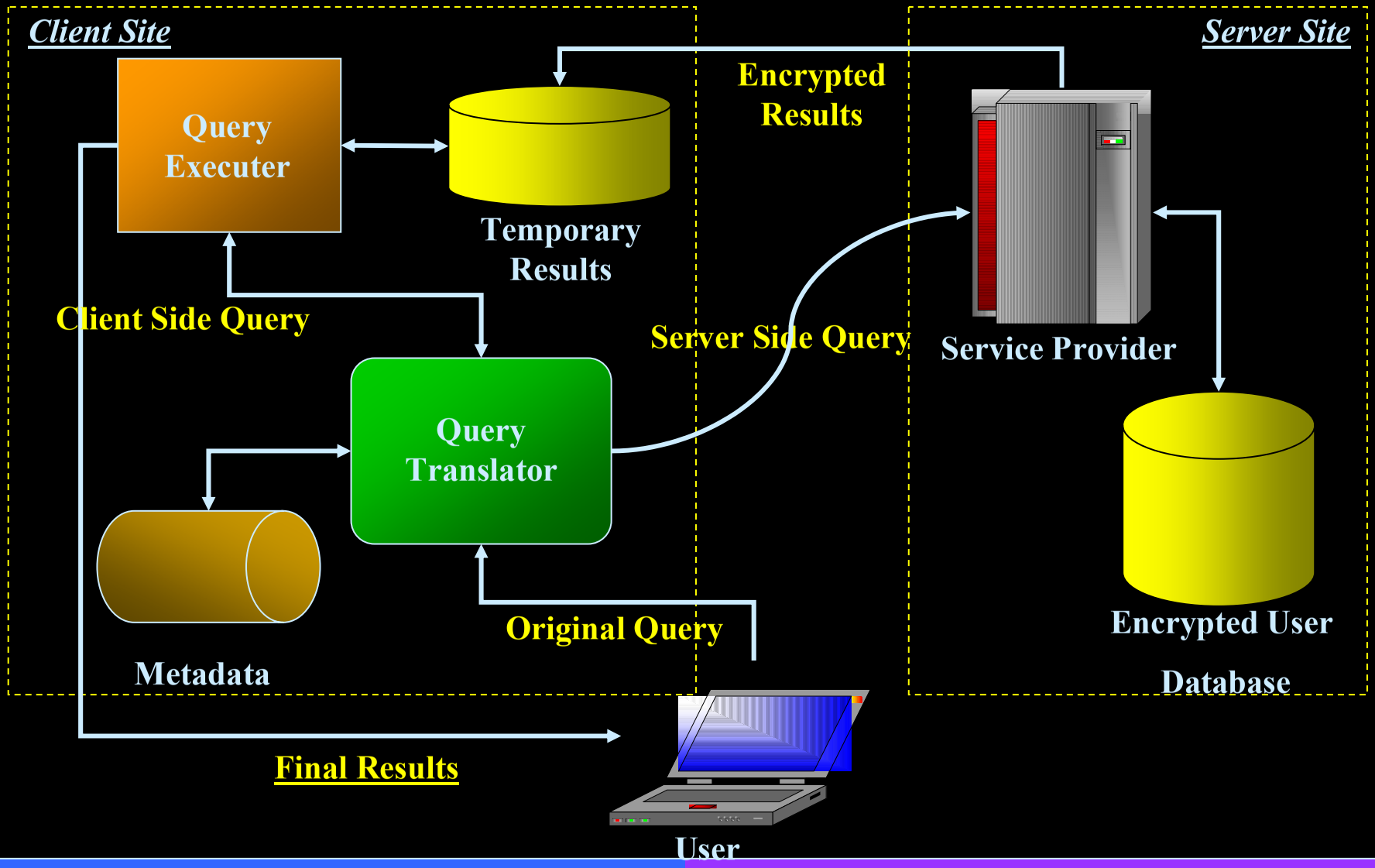
name LIKE '%green%'

$I_{Name}$ in (110, 111)

*Connaissance du serveur*

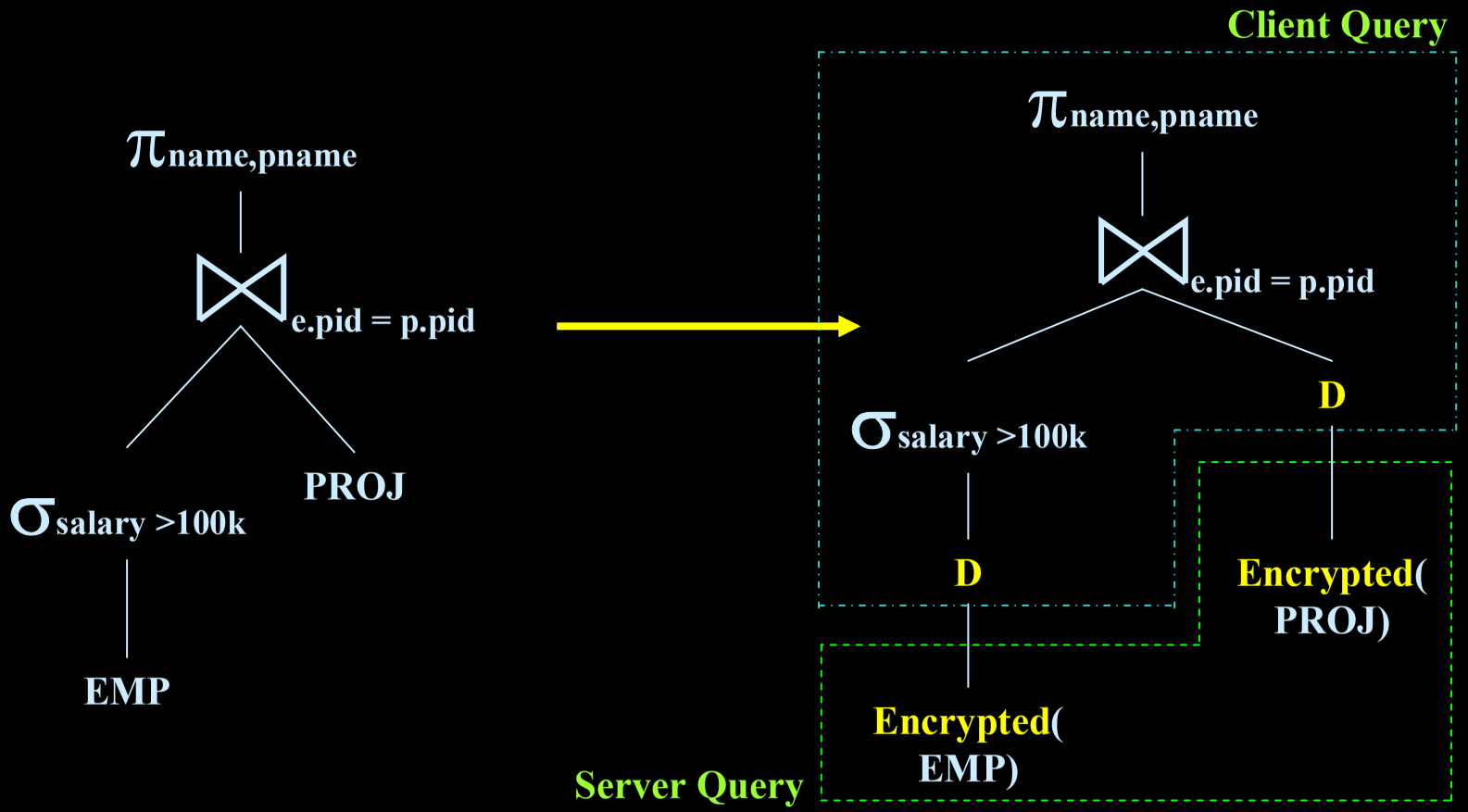


# Architecture

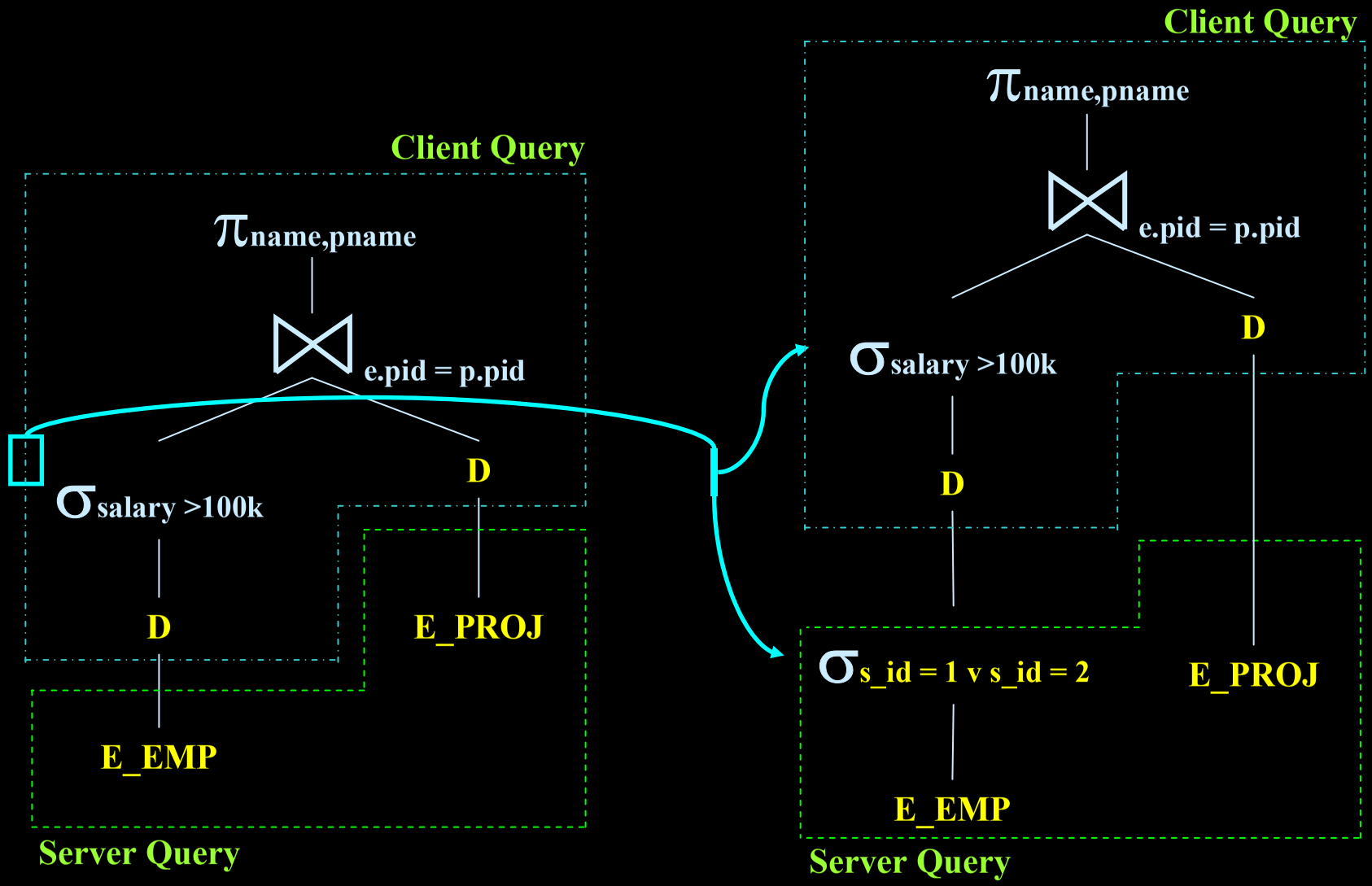


# Décomposition de requête (1)

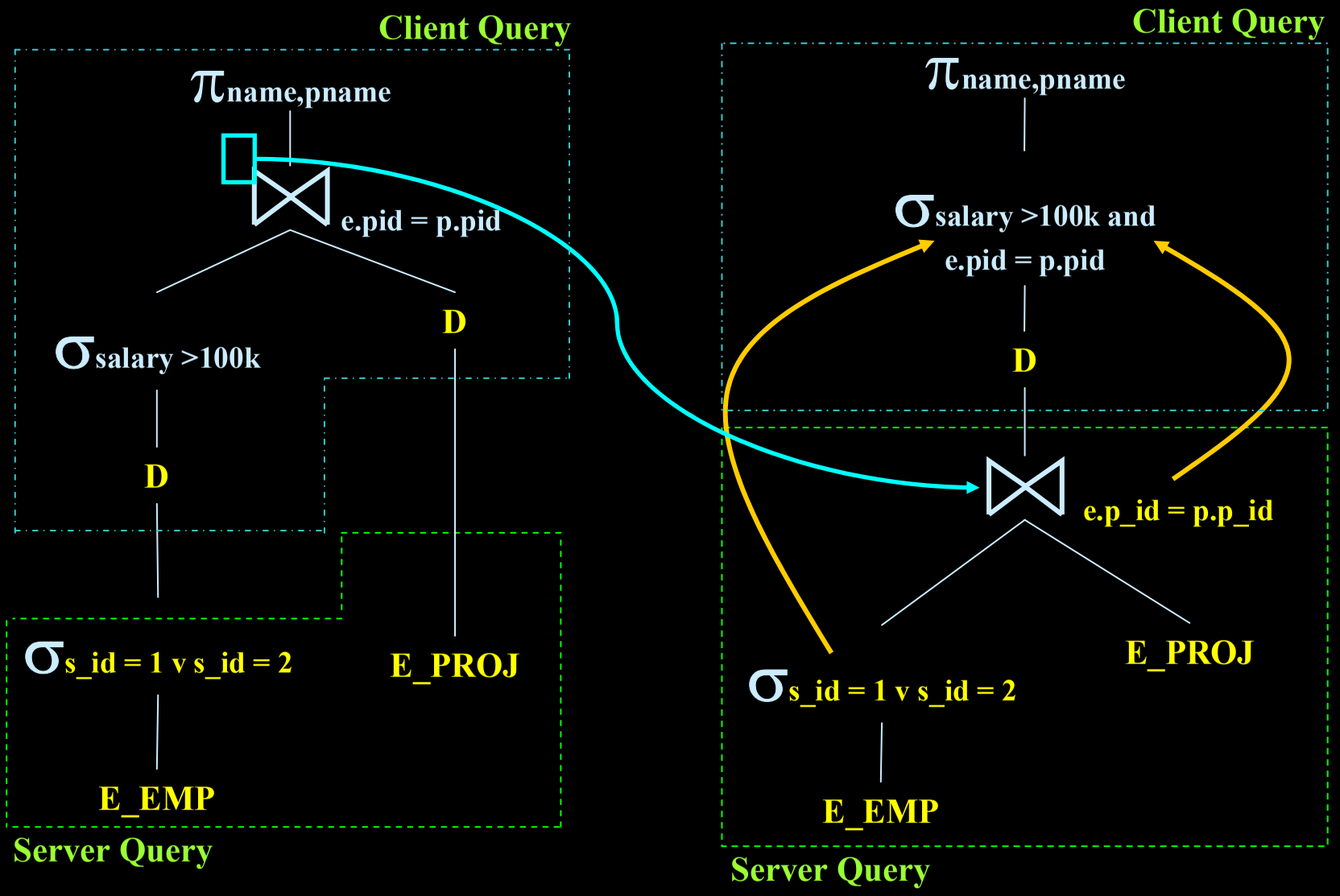
Q: SELECT name, pname FROM emp, proj
WHERE emp.pid=proj.pid AND salary > 100k



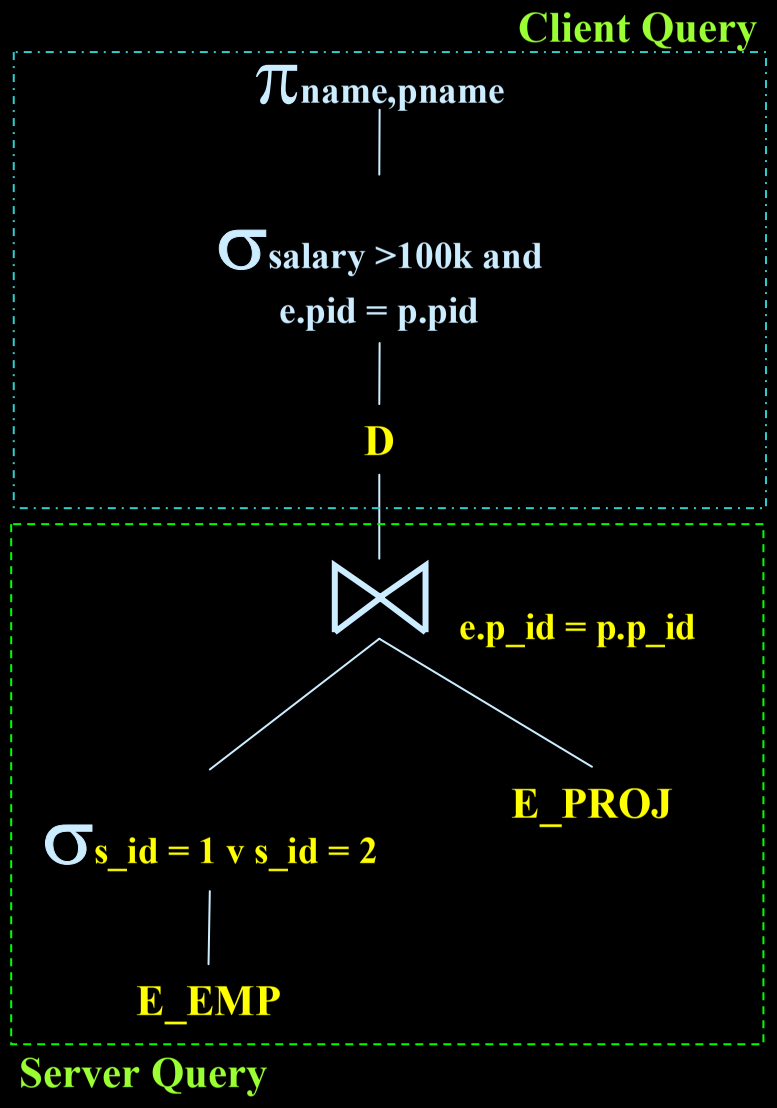
# Décomposition de requête (2)



# Décomposition de requête (3)



# Décomposition de requête (4)




Q: SELECT name, pname
FROM emp, proj
WHERE emp.pid=proj.pid AND
salary > 100k

$Q^S$: SELECT e_emp.etuple,
e_proj.etuple
FROM e_emp, e_proj
WHERE e.p_id=p.p_id AND
s_id = 1 OR s_id = 2

$Q^C$: SELECT name, pname
FROM temp
WHERE emp.pid=proj.pid AND
salary > 100k

# Plan

- Outils cryptographiques
- Application du chiffrement à une base de données
- L'approche serveur
- L'approche client
- L'approche par matériel sûr
- Conclusion

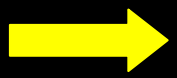
# L'approche à base de matériel sûr (MS)

- MS sur le client
  - Résiste aux attaques sur le client
    - Gestionnaire de droits, clés, données en clair
- Des instances
  - Chip sécurisé embarqué dans une clé USB
  - Carte à puce + serveur
  - Carte SIM + téléphone
- Problématique
  - Traitement BD embarqué – chip extrêmement contraint (RAM)
  - Ex. C-SDA [BoP02]

# Chiffrement de la BD [BoP02]

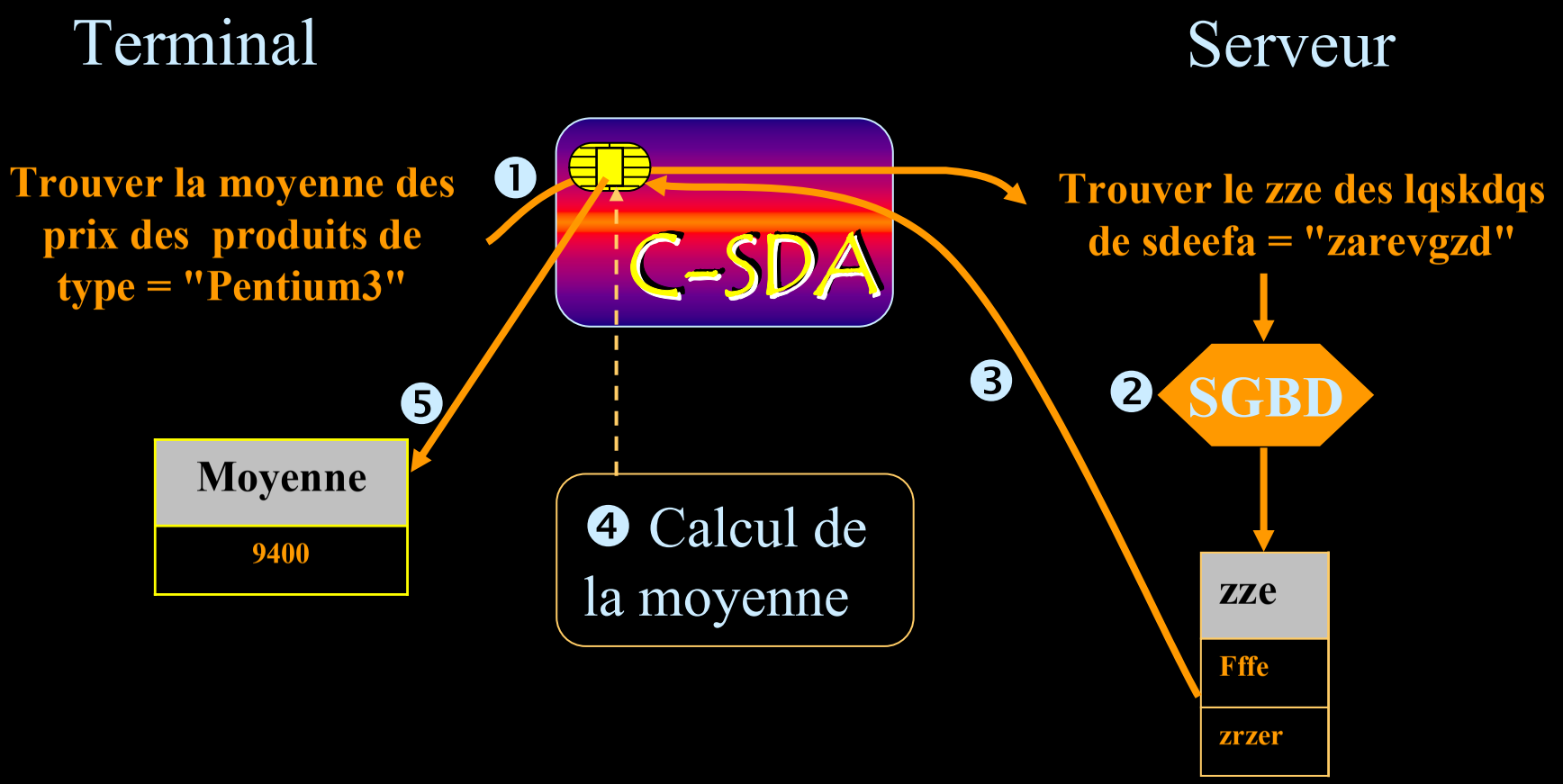
- Approche pionnière (avec des défauts!)
- Données et méta-données chiffrées
  - Hypothèse initiale : chiffrement de granule attribut, préservant l'égalité
- Seuls les clients sont habilités à Créer/Détruire/Modifier des données
- Chiffrement partiel possible

| ref | nom | type | prix |
|---|---|---|---|
| d300 | Dell | Pentium3 | 9800 |
| I260 | IBM | Pentium2 | 6400 |
| ... | ... | ... | ... |



| sdz | azds | sdeefa | zze |
|---|---|---|---|
| zszd | dedef | zarevgzd | Fffe |
| df'g | Sde | iukèefsa | dgss |
| ... | ... | ... | ... |

# Traitement d'une requête complexe




**La partie des requêtes non évaluable sur les données chiffrées est évaluée sur la carte (prédicats <, >, fonctions de calcul, etc..)**

# Conclusion

- Approche serveur
  - Solution des grands éditeurs
  - Non résistant aux attaque internes
  - Problème de résistance aux attaques
    - Indexation des colonnes chiffrées => chiffrement préservant l'ordre
  - Problèmes de perfs
- Approche client
  - Les problèmes sont décuplés
  - Résiste aux attaque internes
  - Ne résiste pas aux attaques du client => BD privées
  - Beaucoup de problèmes de recherche
    - Indexation, étiquetage, modèle de chiffrement
    - Evaluation de requêtes complexes, mises à jour
- Approche par matériel sûr : pour l'instant confiné à la recherche

# Panorama et perspectives : chiffrement

| Domaine | Problème de recherche | Bibliographie (en annexe) |
|---|---|---|
| Chiffrement de BD<br>(approche client, avec materiel sûr) | Modèle d'indexation pour BD chiffrée:<br>indexes chiffrés et étiquetage, execution distribuée client/serveur… | BD relationnelle : [1_xx]<br>BD XML : [2_xx]<br>Avec MS : [3_xx] |
| | Modèle de chiffrement pour BD: préservant égalité/ordre/homomorphisme, indexation sur les données chiffrées, execution centralisée, compromis sécurité/performances… | [4_xx] |
| Recherche d'information et sécurité<br>(email, moteur de recherche, etc.) | Recherche dans des données chiffrées (SED, Search Encrypted Data), indexation par mots clés, chiffrement symétrique/asymétrique… | symétrique: [5_xx]<br>asymétrique: [6_xx] |
| Système de gestion de fichiers | Autentification, gestion de clés, partage sécurisé de fichiers | [7_xx] |
| | Chiffrement transparent | [8_xx] |

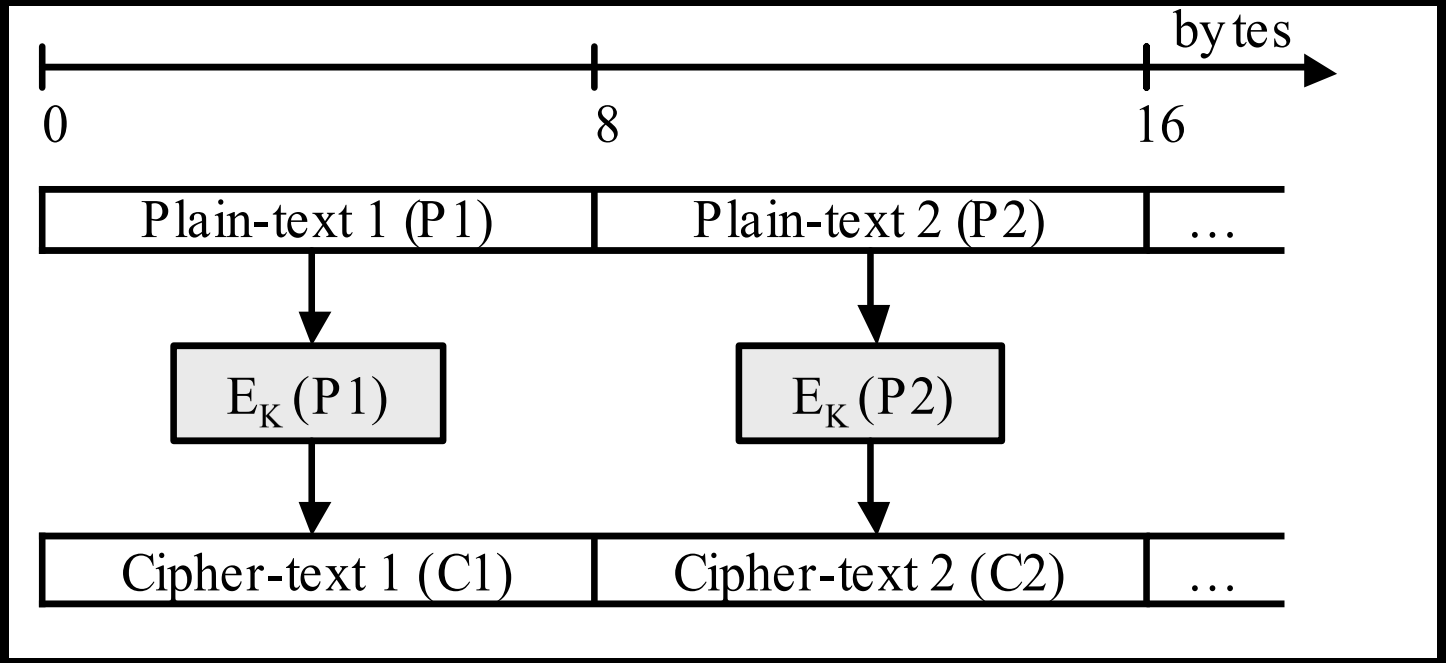
# Bibliographie (partielle)


- [DDJ+03] E. Damiani, S. De Capitani Vimercati, S. Jajodia, S. Paraboschi, P. Samarati, "Balancing Confidentiality and Efficiency in Untrusted Relational DBMSs", ACM CCS, 2003.
- [HIL+2] H. Hacigumus, B. Iyer, C. Li, S. Mehrotra, "Executing SQL over Encrypted Data in the Database-Service-Provider Model", SIGMOD, 2002.
- [GeZ07] Tingjian Ge and Stan Zdonik, "Answering aggregation queries in a secure system model", VLDB, 2007.
- [Ora07] Oracle Advanced Security, Technical White Paper, Oracle Corp. 2007.
- [AKS+04] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu, "Order preserving encryption for numeric data", SIGMOD 2004
- [BoP02] Luc Bouganim and Philippe Pucheral, "Chip-secured data access: condential data on untrusted servers", VLDB 2002


- …voir le complément en annexe

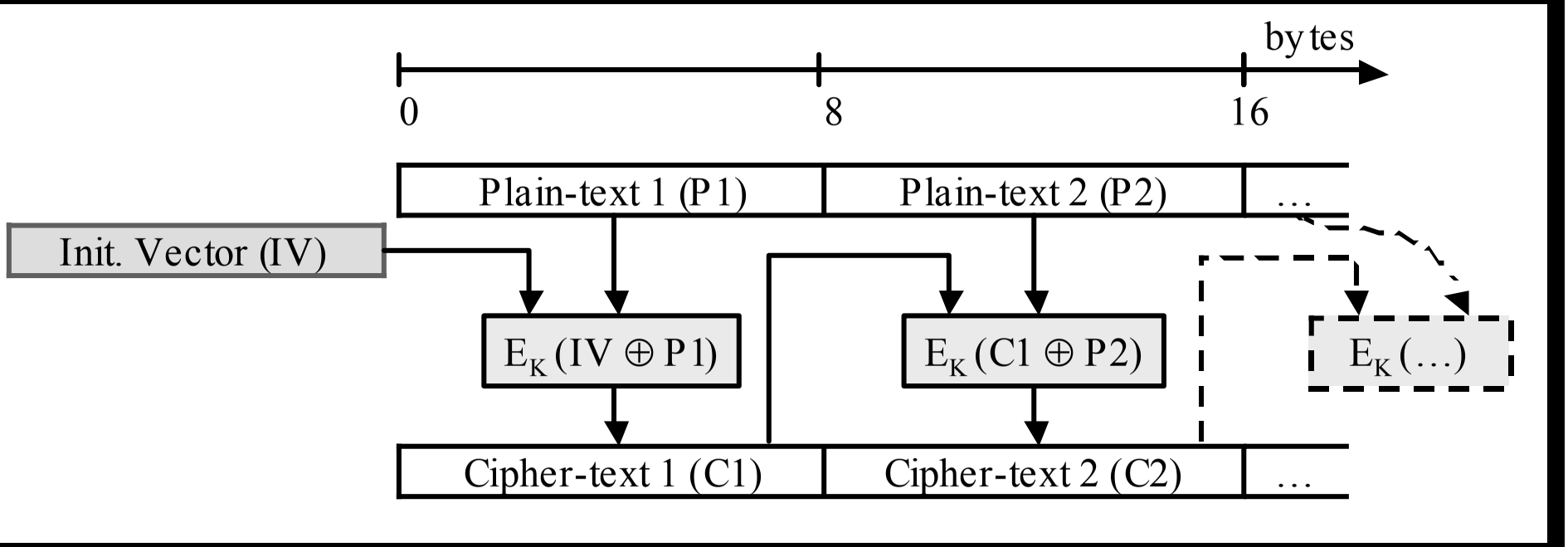
# Mode opératoire ECB (Electronic Code Book)

- Les blocs sont chiffrés indépendamment



# Mode opératoire CBC (Cipher Block Chaining)

- les blocs chiffrés intégrent la valeur des précédents



# Mode opératoire CTR

- Résultat du XOR entre le clair et un mot aléatoire