Les protocoles de Retrait d'Informations Privé (RIP)

Carlos Aguilar Melchor

XLIM - UMR CNRS 6172
Université de Limoges

17 Février 2008

Définitions

Informelle

Protocole permettant à un utilisateur d'obtenir un élément d'une base de données en occultant auprès de celle-ci de quel élément il s'agit.

Un peu plus formelle

Un protocole est dit un protocole RIP si :

- il permet aux utilisateurs d'obtenir n'importe quel élément d'une base de données, quel que soit le contenu de celle-ci ;
- pour tout couples d'éléments e_1, e_2 de la base, les requêtes des utilisateurs correspondant à e_1 sont indistingables de celles correspondant à e_2.

Retrait d'Informations Privé ?

Vous voulez dire Retrait d'Informations Privées ?

Non ! C'est le retrait qui est privé.
En anglais Private Information Retrieval (PIR). Terme ambigu :

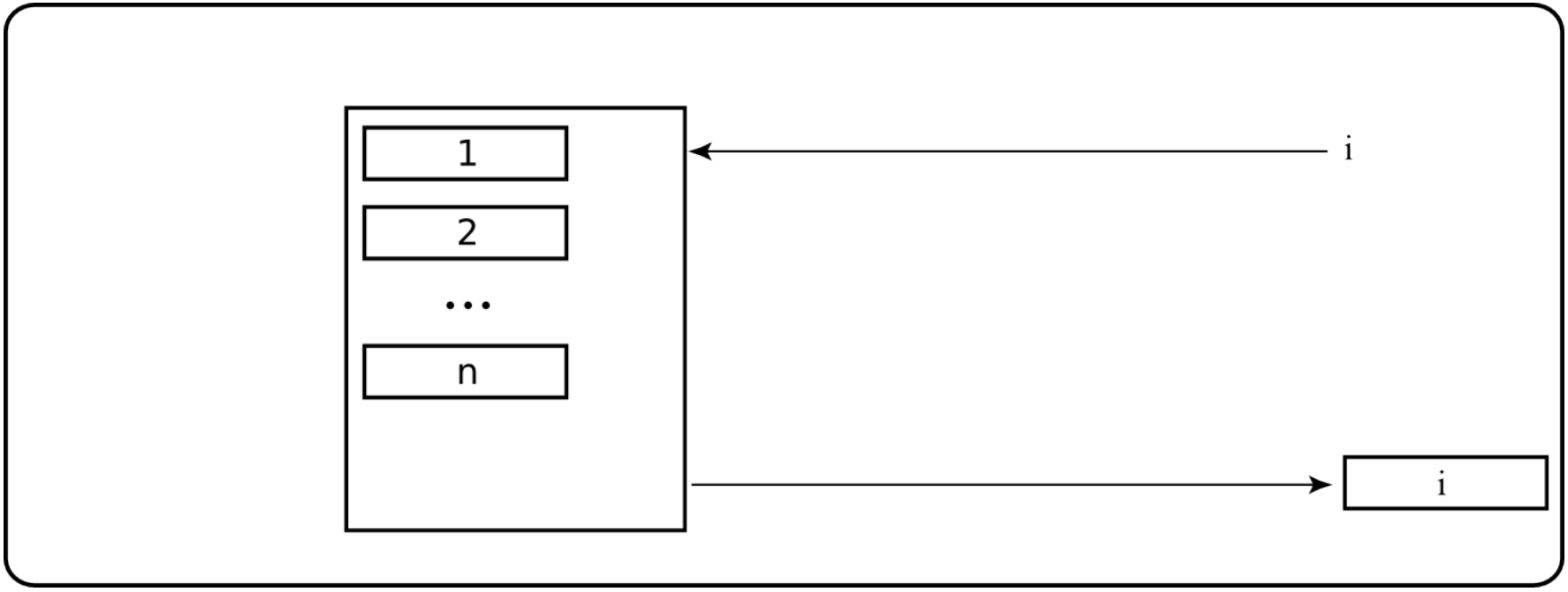
- {Private Information} Retrieval ⇒ Retrait d'Informations Privées FAUX
- Private {Information Retrieval} ⇒ Retrait d'Informations Privé

La richesse de la langue française ...

Pourquoi pas Retrait Privé d'Informations ?

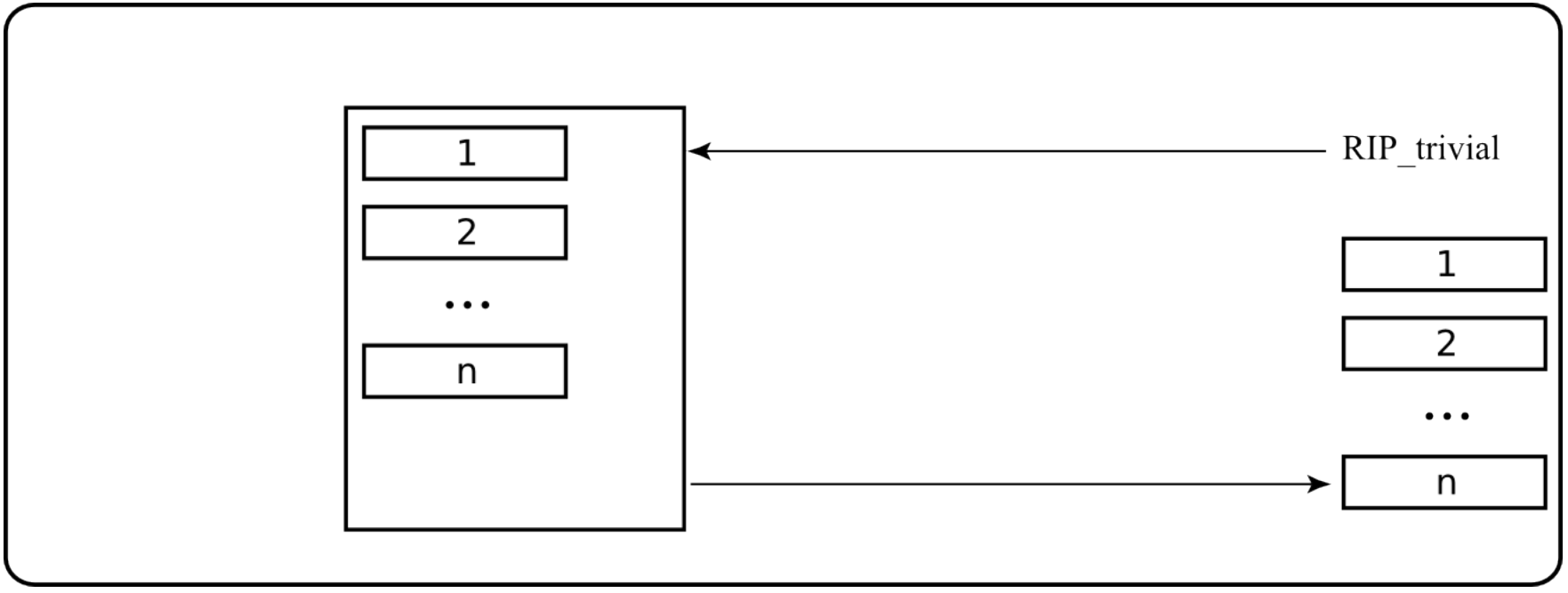
- Parce que ça peut faire penser à Retrait {Privé d'Informations} et par là à quelque chose du genre de *Zero Knowledge Retrieval* ce qui n'a rien à voir.
- Parce que RIP c'est beaucoup plus *sexy* que RPI :).

Retrait traditionnel non-privé



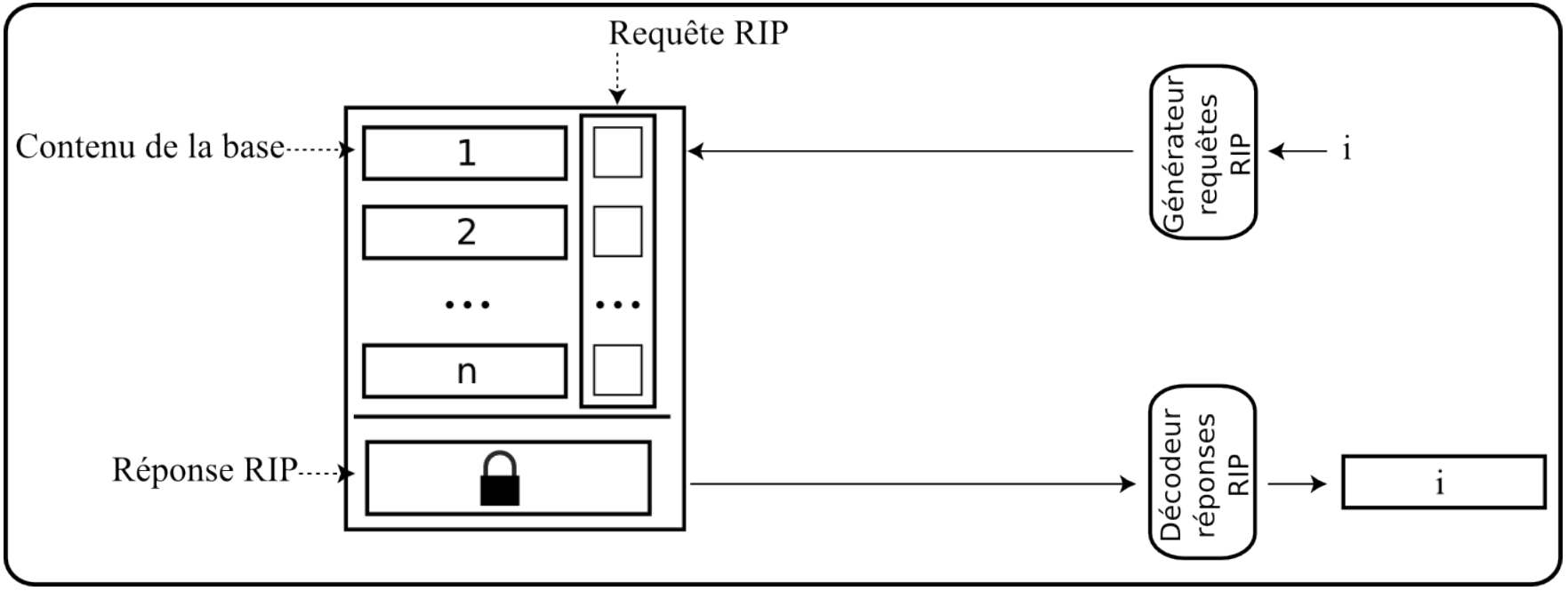
- On envoie l'index *i* de l'élément nous intéressant.
- On reçoit l'élément en question.

Retrait privé trivial



- On demande toute la base de données.
- On ne lit localement que l'élément nous intéressant.

Protocole RIP



- On génère une requête pour l'élément d'index *i*.
- La base calcule une réponse à partir de notre requête.
- On décode la réponse et on obtient l'élément souhaité.

Variantes

Le RIP symétrique (SPIR)

- L'utilisateur n'obtient aucune information sur les autres éléments.
- Il existe des transformations génériques efficaces.

L'*Oblivious Transfer*

- Axe de recherche existant depuis 1981 (Michael Rabin).
- Même conditions de sécurité que le RIP symétrique.
- Vise à réduire les coûts calculatoires pour l'utilisateur et le serveur.
- Comporte une étape où la BDD est envoyée intégralement (chiffrée).

Motivations

La base de données peut être ...

- une bibliothèque électronique, et quels livres nous lisons peut fournir des informations sur nos convictions politiques, ou certains détails de notre personnalité qu'il peut être souhaitable de garder confidentiels ;
- une base de données pharmaceutique, et certains laboratoires clients de la base peuvent désirer qu'on ne puisse pas savoir à quels principes actifs ils s'intéressent ;
- des cours d'actions, et les clients peuvent être des investisseurs ne voulant pas dévoiler quels cours ils suivent.

La solution triviale est inacceptable si la base est ...

- de taille extrêmement importante ;
- confidentielle ;
- rapidement obsolète.

Chor et al., FoCS'95

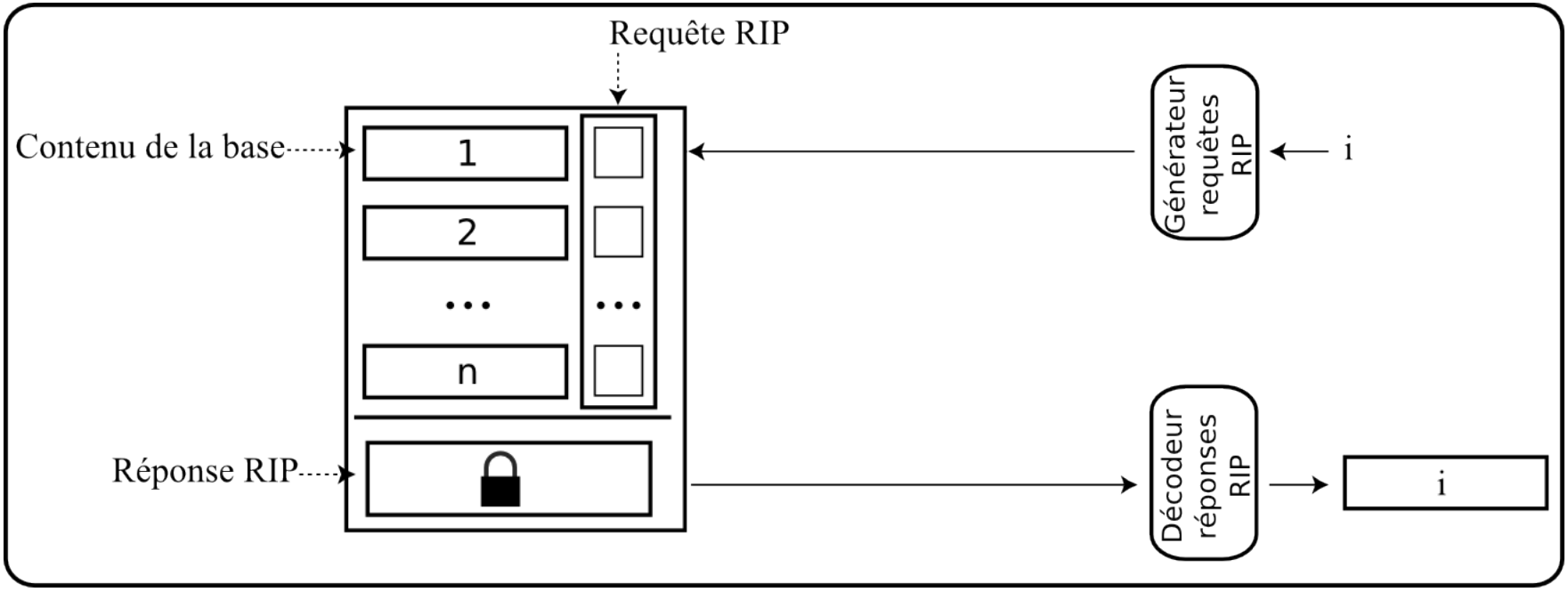
Contexte

- La base de données contient n éléments $\{e_1, \cdots, e_n\}$, chacun de ℓ bits.
- L'utilisateur veut récupérer le i-ème élément, e_i.
- Deux répliques BDD_1 et BDD_2 de la base de données peuvent être interrogées séparément.

Limitations

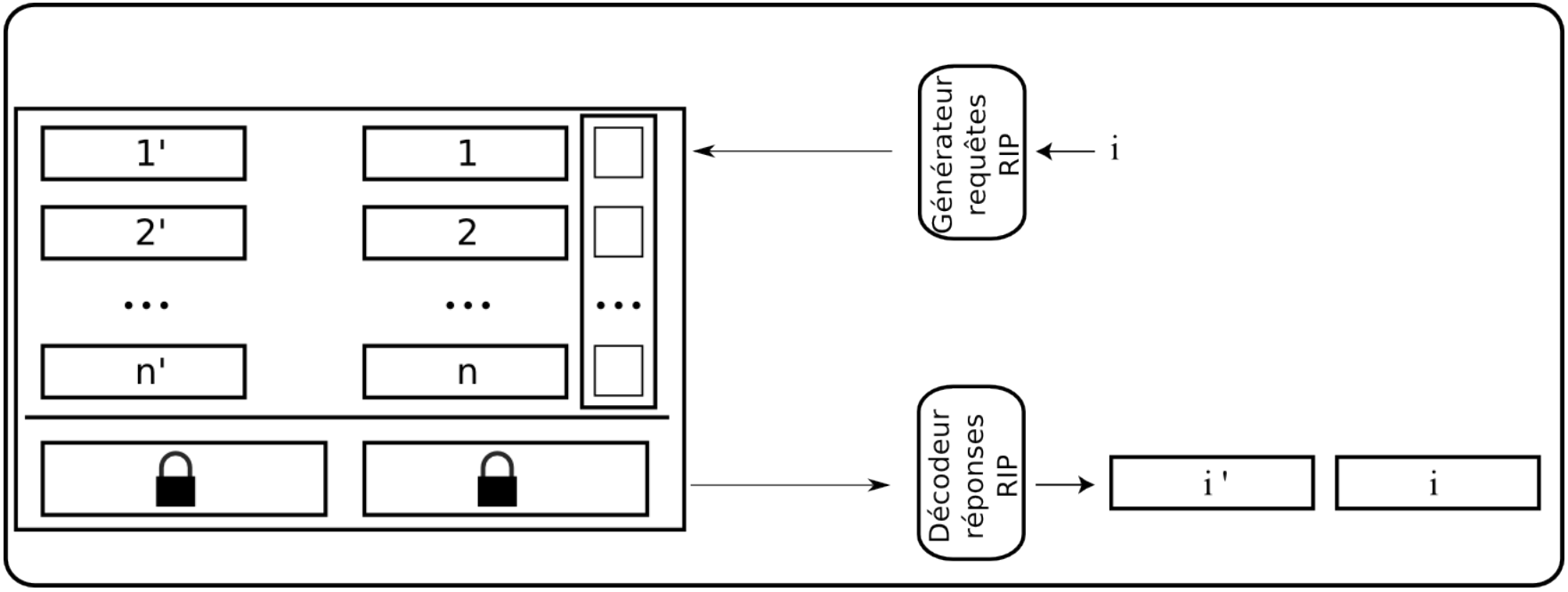
- Ils ne peuvent exister que pour des bases de données répliquées.
- Il faut que les différentes répliques soient cohérentes.
- Ils ne sont pas inconditionnellement sûrs.

Base commune



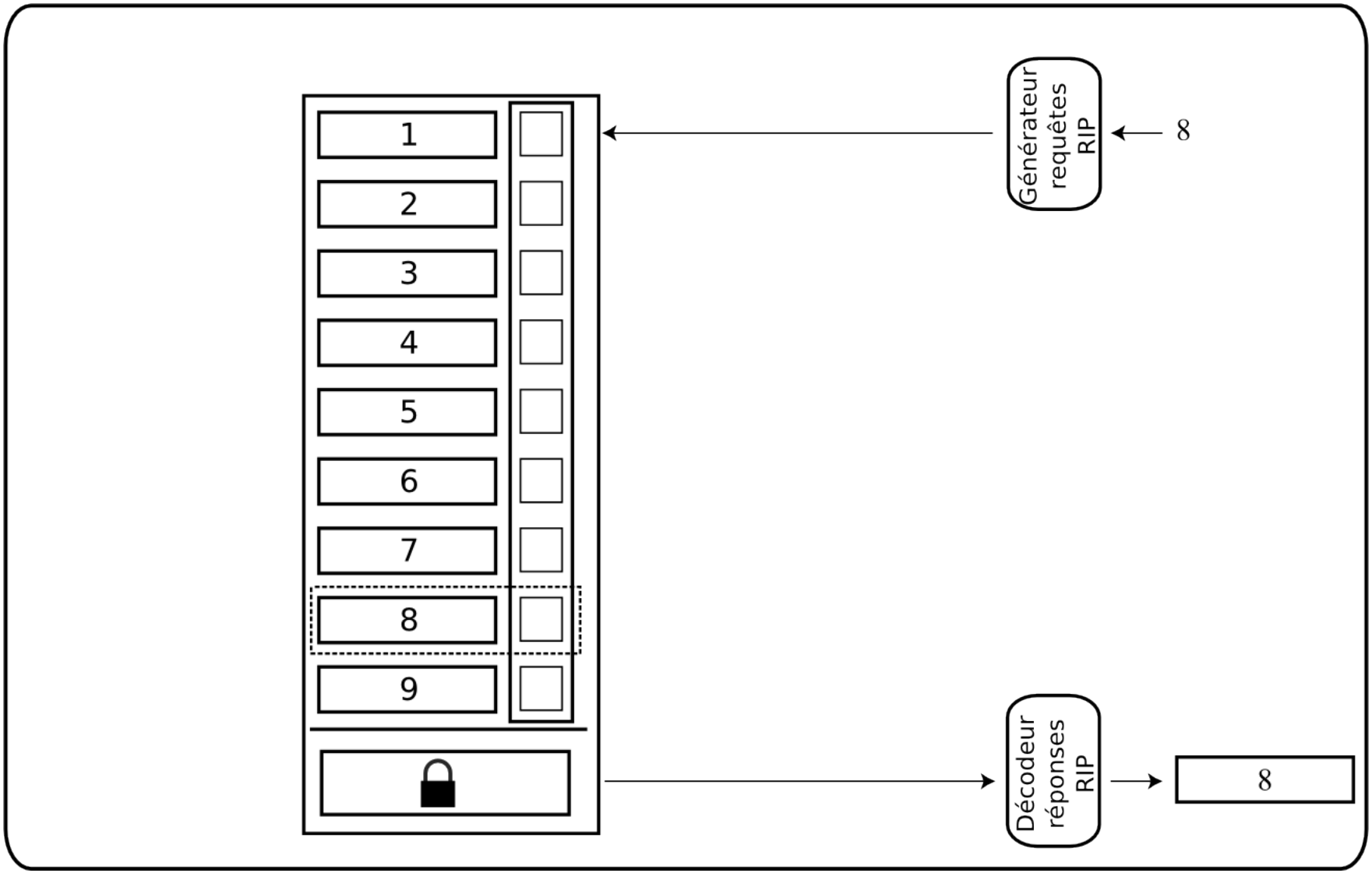
- La requête est habituellement composée de *n* éléments.
- Chaque élément de la requête est combiné à un élément de la base.
- La requête est indépendante des contenus de la base.
- La réponse peut encoder un nombre maximum de bits.

Génération itérative des réponses

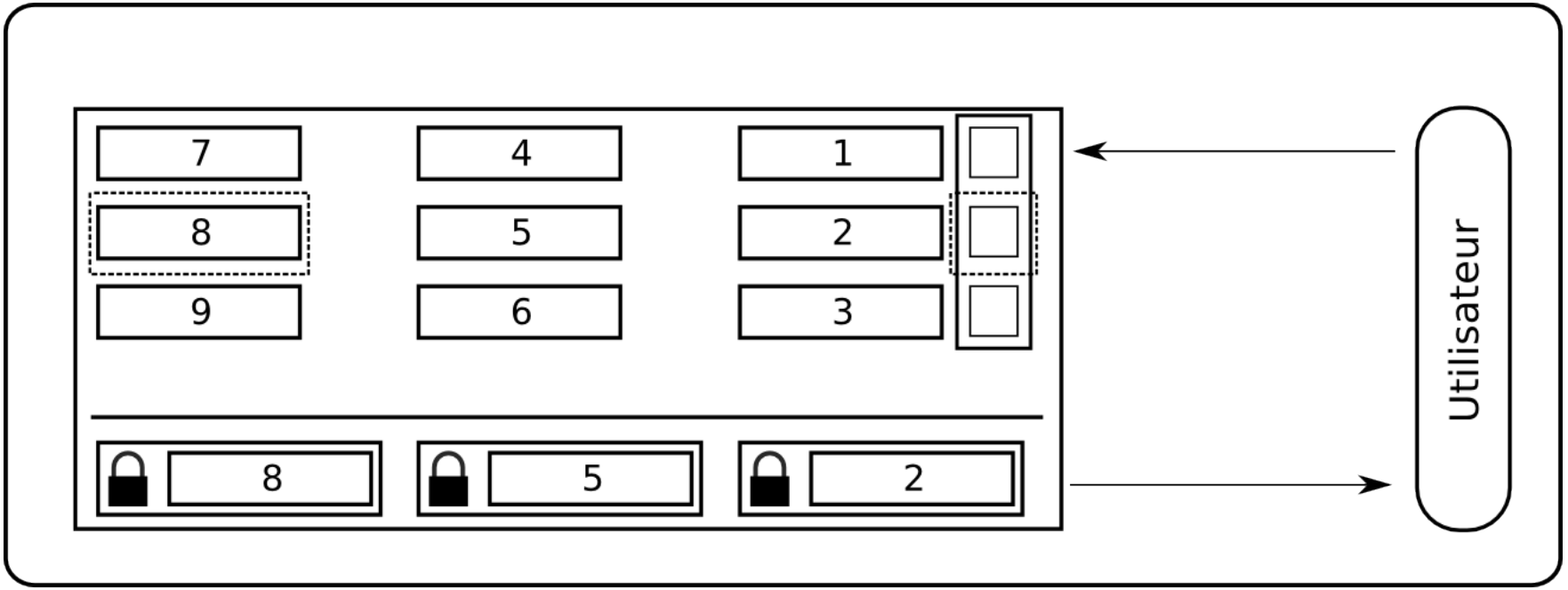


- La réponse peut encoder un nombre maximum de bits.
- La requête est indépendante des contenus de la base.
  - $\Rightarrow$ Découpage des éléments trop grands et génération itérative.

Agrégation d'éléments

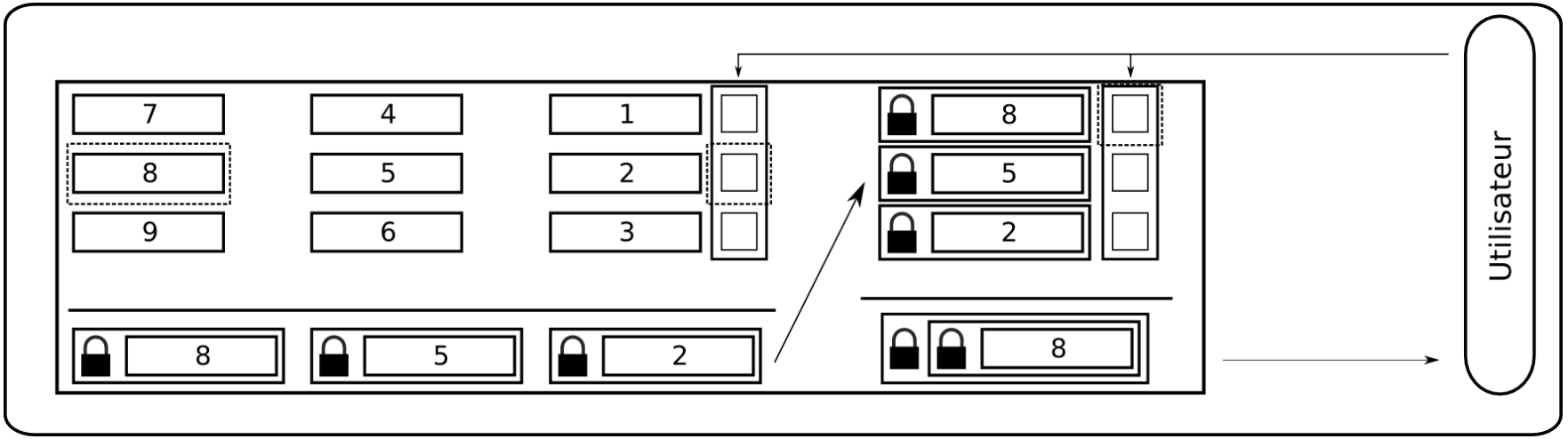


Agrégation d'éléments



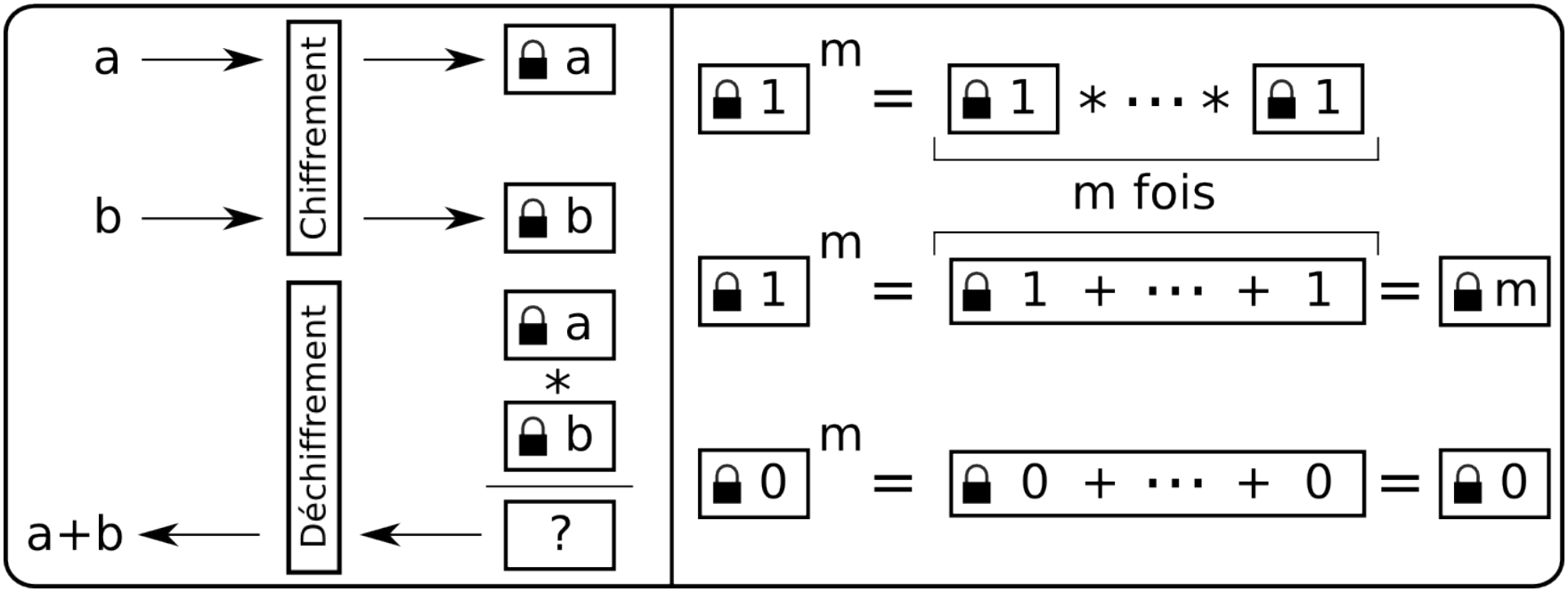
- Permet de réduire la taille de la requête d'un facteur F.
- Induit une augmentation de la taille de la réponse d'un facteur F.
- Proposé par Chor et al. pour la récupération d'un bit.
  - requête $2n$, réponse $2 \Rightarrow$ requête $2\sqrt{n}$, réponse $2\sqrt{n}$
- En pratique c'est rarement intéressant.

Récursion



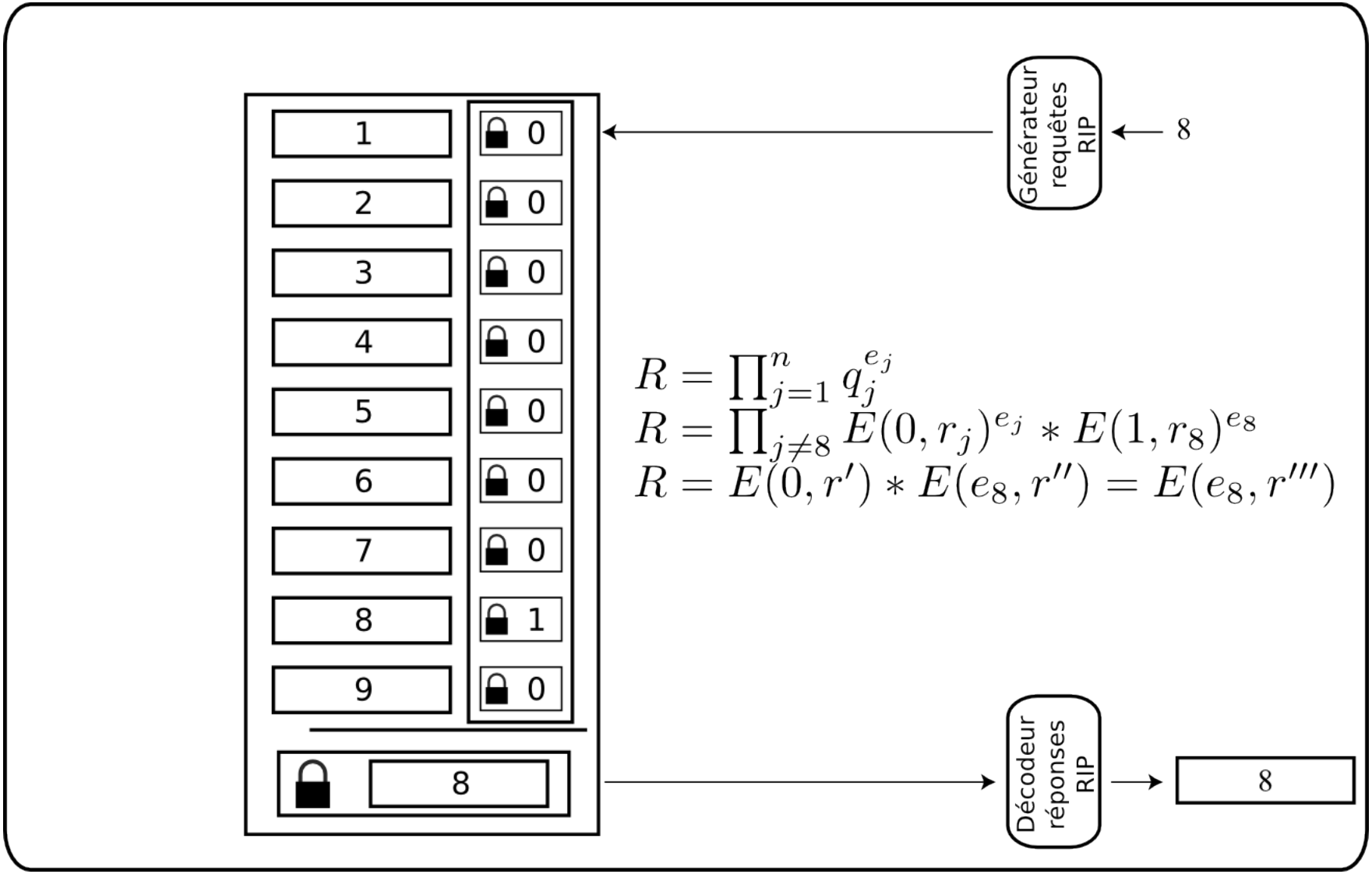
- La première requête permet d'obtenir une ligne.
- La deuxième requête opère sur la réponse générée.
  - Permet d'obtenir un élément de la ligne.
- Requête : $2\sqrt{n}$, facteur d'expansion de la réponse : $F_{RIP} * F_{RIP}$.
- Généralisation triviale $\Rightarrow$ requête : $d \times n^{1/d}$, facteur d'expansion : F_{RIP}^{d}.

Le chiffrement homomorphe



- Systèmes de chiffrement randomisés.
- Indistingabilité entre les chiffrés de 0 et de 1.
- Paillier EUROCRYPT'99, Damgård et Jurik ACISP'03.

Julien P. Stern ASIACRYPT'98



Julien P. Stern ASIACRYPT'98

Coût des communications

- Requête : n chiffrés de 2048 bits.
- Facteur d'expansion : 2.
- Récursion $\Rightarrow$ requête : $2048 \times d \times n^{1/d}$, facteur d'expansion : 2^d.

Coût calculatoire

- Facteur limitant dans les protocoles RIP : génération de la réponse.
- Coût : une multiplication modulaire de 2048 bits par bit dans la BDD.
- Traitement de la BDD à $T = 160$–490 Kbits/s.
- Utilisateur obtient l'élément l'intéressant à T/n.
- La récursion à peu d'influence sur le coût calculatoire.

Lipmaa ISC'05

Idée

- Dans le protocole de Stern on chiffre avec Paillier a chaque niveau de récursion
  - On chiffre des blocs de 1024 bits et on obtient des blocs de 2048 bits.
  - $\Rightarrow$ on multiplie par 2 la taille de la réponse à chaque fois.
- Le système de chiffrement de Damgård et Jurik permet de chiffrer en passant de $s \times 1024$ bits à $(s+1) \times 1024$ bits.
- Lors de premier niveau de récursion on chiffre les donnés par blocs de 1024 bits et on obtient des chiffrés de 2048 bits.
- Lors du deuxième niveau on chiffre les données de 2048 bits et on obtient des chiffrés de 3072 bits.
- ...

Lipmaa ISC'05

Coût des communications

- Requête : $n^{1/d}$ chiffrés de 2048 bits, $n^{1/d}$ chiffrés de 3072 bits, $\cdots$, $n^{1/d}$ chiffrés de $(d+1) \times 1024$ bits.
- Requête : $2048 \times ((d+2)(d-1)/4) \times n^{1/d}$
- Facteur d'expansion : $1+d$.

Coût calculatoire

- Les opérations au delà du premier niveau de récursion ont un coût négligeable.
- $\Rightarrow$ même coût que le protocole de Stern.

$\Rightarrow$ le protocole de Lipmaa est toujours préférable.

Les prédicats adaptatifs

Idée

- La requête ne change jamais, et est connue et commune à tout le monde.
- L'utilisateur envoie une description d'un anneau fini où un élément de la requête est particulier.
- $\Rightarrow$ seul l'élément de la BDD associé sera "absorbé".
- La sécurité repose sur le fait que la BDD est incapable de savoir quel élément est particulier dans l'anneau.

- Approche initialement proposée par Cachin Micali et Stadler à EUROCRYPT'99 sous une version peu utilisable en pratique.
- $\Rightarrow$ Aboutissement de l'approche : Gentry et Ramzan ICALP'05.

Gentry et Ramzan ICALP'05

Hypothèse de sécurité *the generalized ϕ-hiding assumption*

Soit m un module difficile à factoriser g est un élément de l'anneau $\mathbb{Z}/m\mathbb{Z}$ d'ordre π (une puissance première). Il est difficile de savoir si π (pour $\log \pi < (log\phi(m))/4$) divise $\phi(m)$ ou pas à partir du couple (m, g).

Performances

- Requête : 2048 bits.
- Facteur d'expansion : 4.
- Coût calculatoire : une multiplication modulaire de 1024 bits par bit dans la BDD. $\Rightarrow T =$ 490–1500 Kbits/s

Pre-calcul

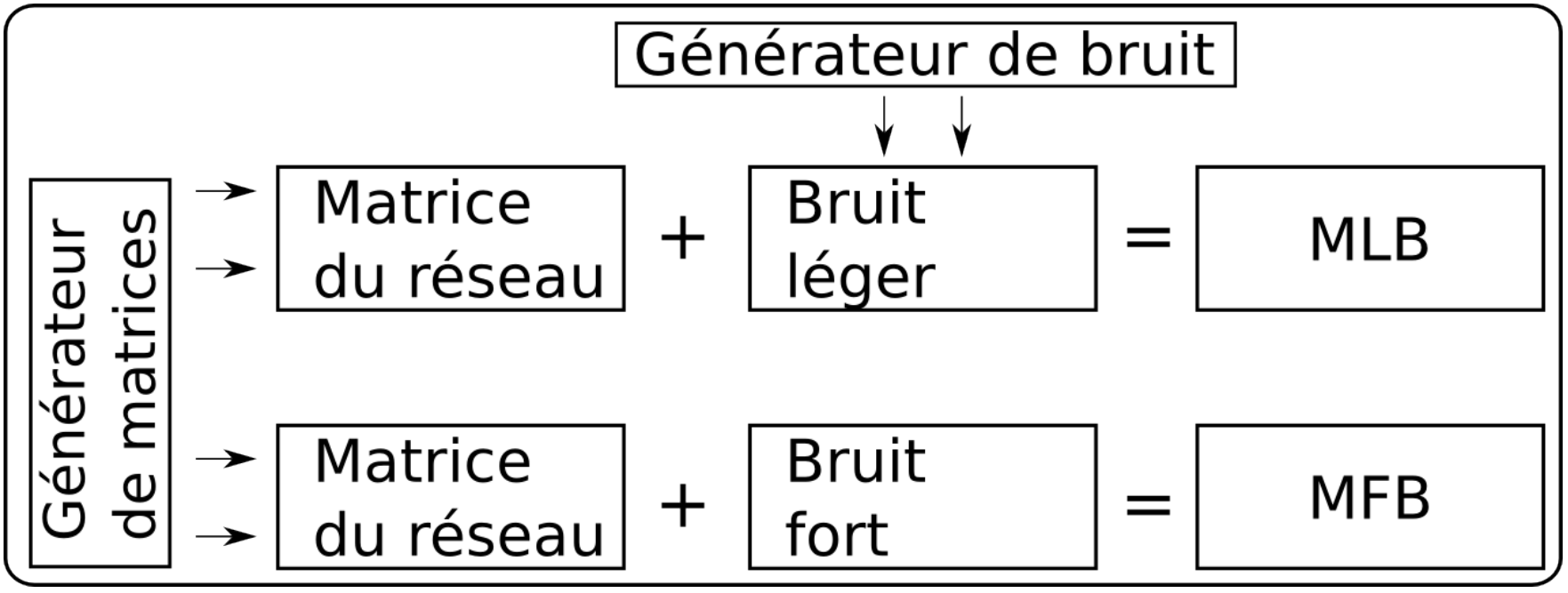
- Phase de pre-calcul nécessaire.
- Coût en $O(\ell \times n^2)$. Deux heures pour 1000 fichiers de 2 Mo.
- Doit être relancée quand la base est actualisée.

Les protocoles basés sur le bruit

Travaux

- Kiayias et Yung ICALP'01 : basé sur le *noisy polynomial reconstruction problem*.
  - Cassé par Bleichenbacher et al. (ICALP'03) puis par Coppersmith et Sudan (STOC'03).
- Gasarch et Yerukhimovich 2006 : basé sur le cryptosystème de Regev.
  - Donne lieu à un facteur d'expansion de l'ordre de 3×10^5.
  - Pas plus rapide que les protocoles basés sur le chiffrement homomorphe ou les prédicats adaptatifs.
- Aguilar et Gaborit WEWoRC'07 : basé sur des réseaux perturbés
  - Problèmes nouveaux, et donne lieu à des requêtes de grande taille.
  - Très rapide.

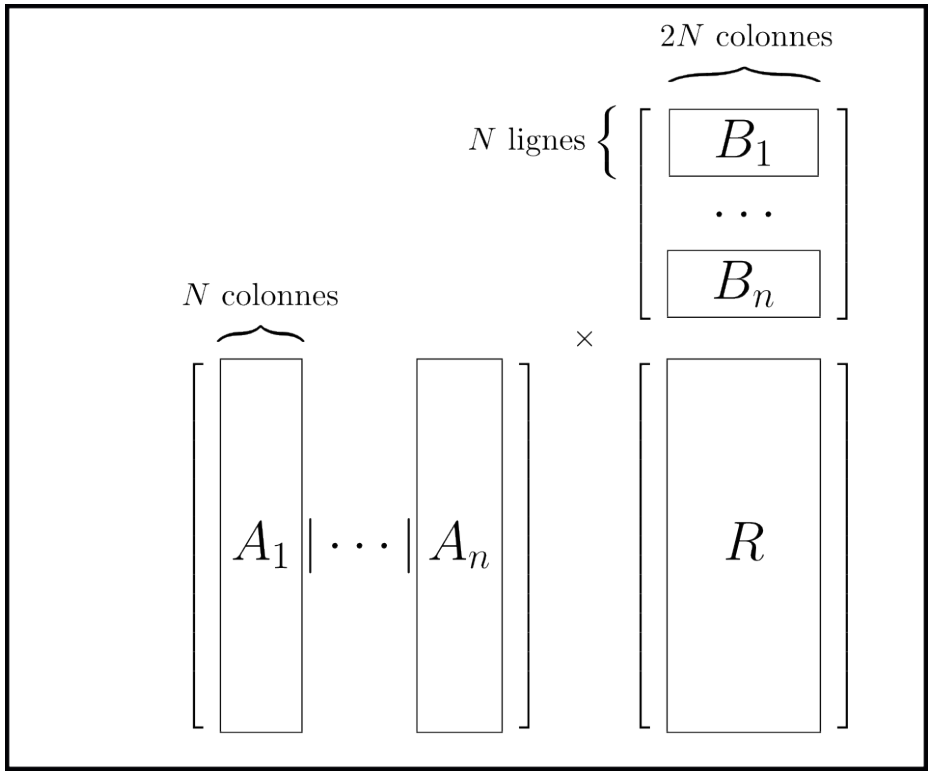
Aguilar et Gaborit WeWORC'07



Hypothèse de sécurité : *the differential hidden lattice problem*

Il est difficile de distinguer les bases légèrement perturbées des bases fortement perturbées d'un réseau dont on ne connaît pas une base non perturbée.

Aguilar et Gaborit WeWORC'07



Génération de la réponse

Chaque scalaire de A multiplie une ligne d'une matrice de la requête.
$\Rightarrow$ coût par bit : une centaine de petites sommes.

Aguilar et Gaborit WeWORC'07

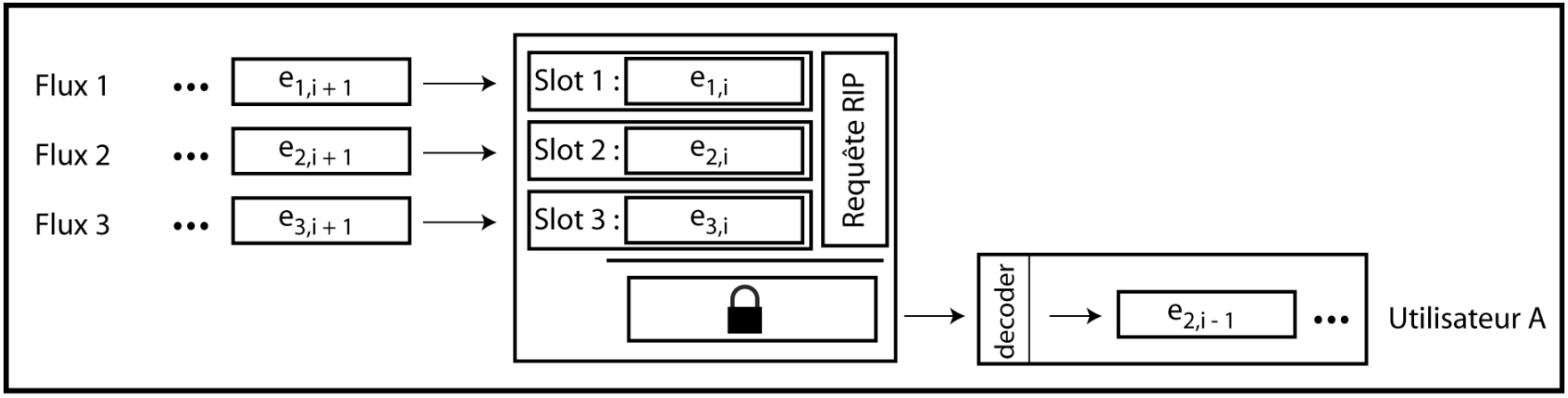
Performances en communication

- Requête : $300 \times 10^3 \times d \times n^{1/d}$.
- Facteur d'expansion : 6^d.
- Requêtes environ 100 fois plus grandes que pour Lipmaa.
- Version soumise permettant de réduire ce facteur ...

Coût calculatoire

Protocole	Vitesse de traitement		
	matériel à 150$	matériel à 800$	matériel à 1100$
Lipmaa	160 Kbits/s	280 Kbits/s	490 Kbits/s
Gentry et Ramzan	490 Kbits/s	890 Kbits/s	1500 Kbit/s
Aguilar et Gaborit (CPU)	33 Mbits/s	130 Mbits/s	230 Mbits/s
Aguilar et Gaborit (GPU)	270 Mbits/s	1.2 Gbits/s	2 Gbits/s
Protocole trivial	Bande passante de l'utilisateur		

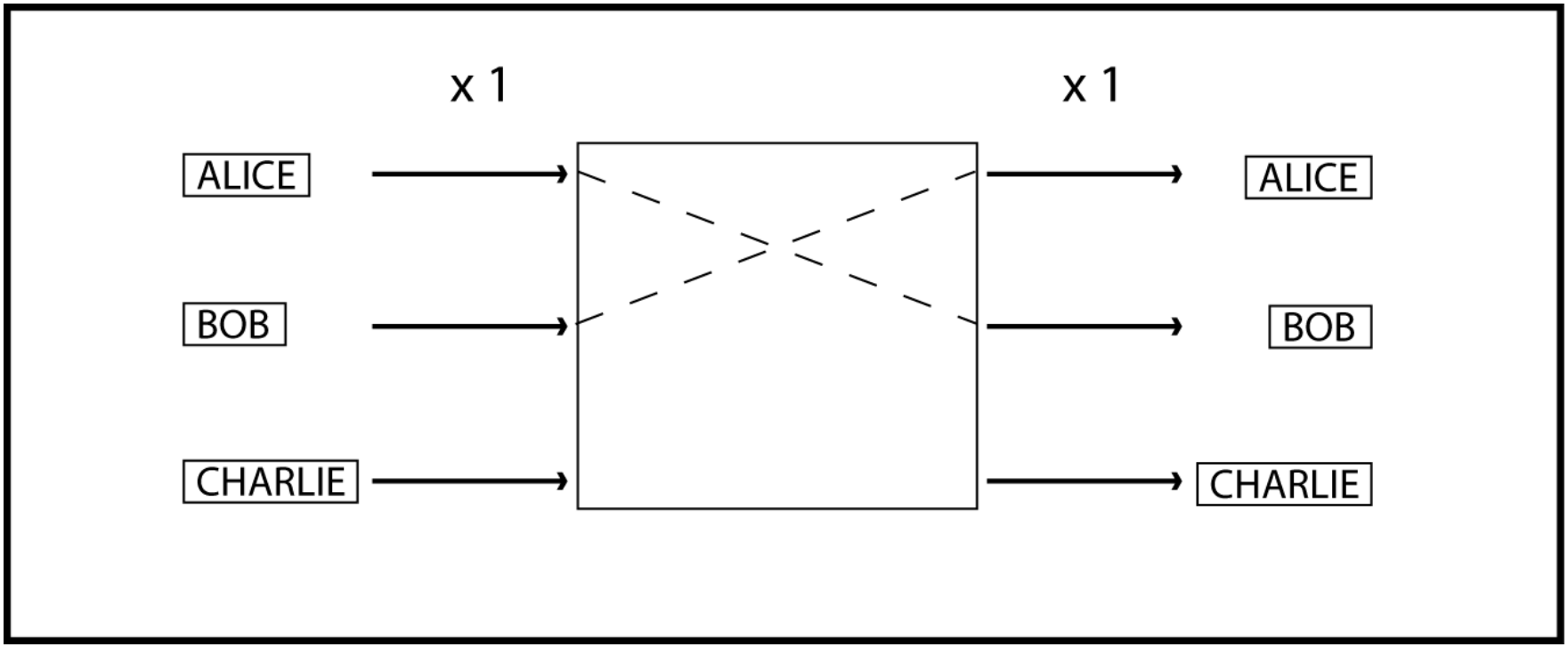
Vidéo à la demande



Contexte

- Un service de vidéo à la demande propose vingt films payants.
- Pour pouvoir voir le film en direct il est nécessaire pour l'utilisateur d'avoir un flux de 500 Kbits/s.
- Débit par utilisateur $T/(20 * N_u)$.
- $T = 2$ Gbits/s $\Rightarrow$ 200 utilisateurs.
- Requête : $300 * 10^3 * 20 = 6$ Mbits.

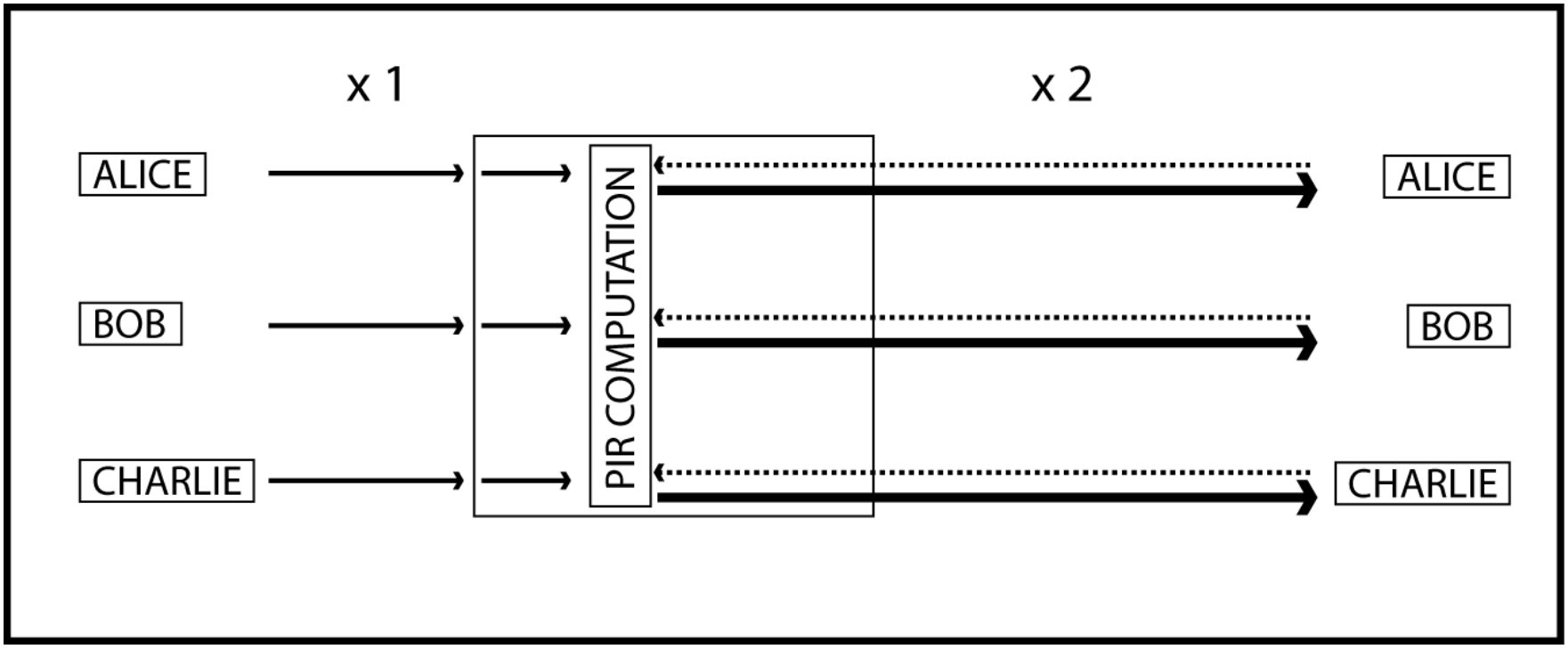
Serveur de VoIP



Performances

- Aguilar et Deswarte NCA'05, NCA'06, ACSAC'07.
- Permet de communiquer anonymement à des centaines d'utilisateurs avec les protocoles classiques.
- $x100$ si on réduit la taille des requêtes dans notre protocole RIP.

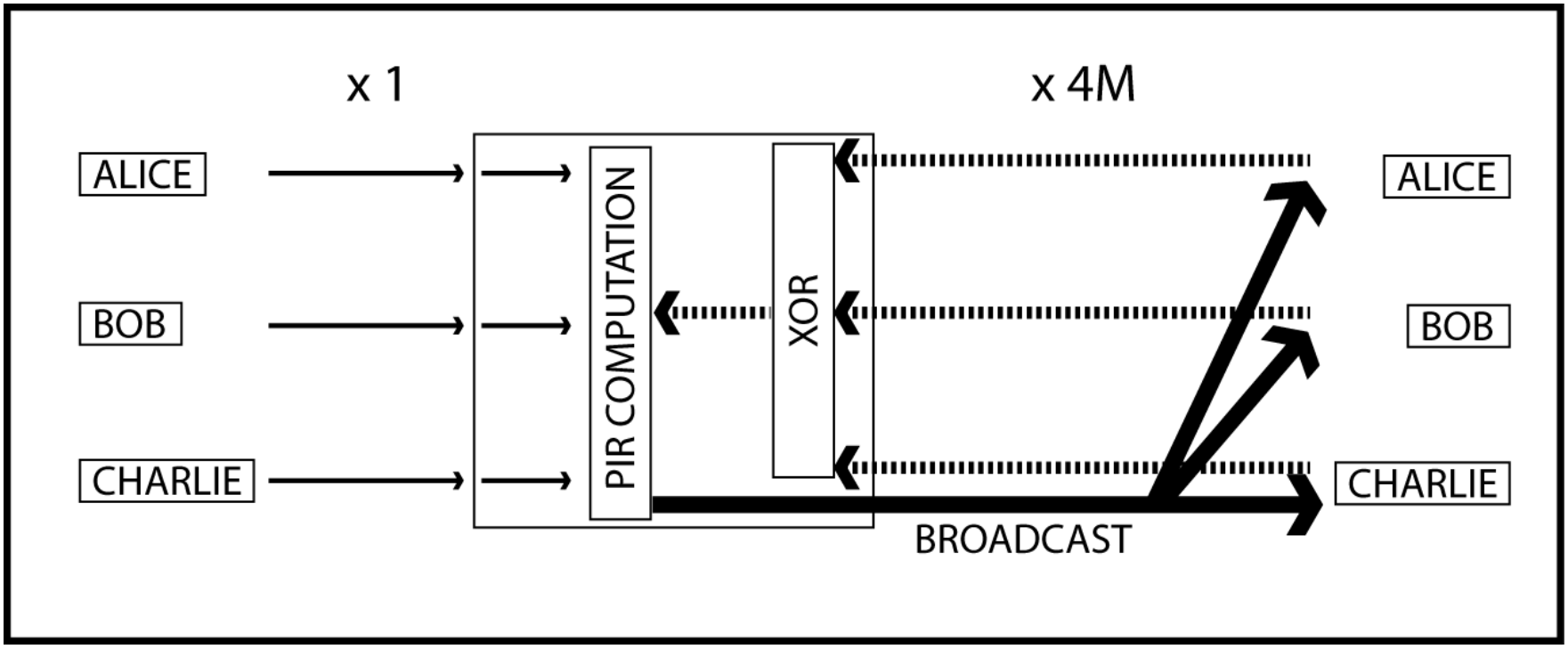
Serveur de VoIP



Performances

- Aguilar et Deswarte NCA'05, NCA'06, ACSAC'07.
- Permet de communiquer anonymement à des centaines d'utilisateurs avec les protocoles classiques.
- $x100$ si on réduit la taille des requêtes dans notre protocole RIP.

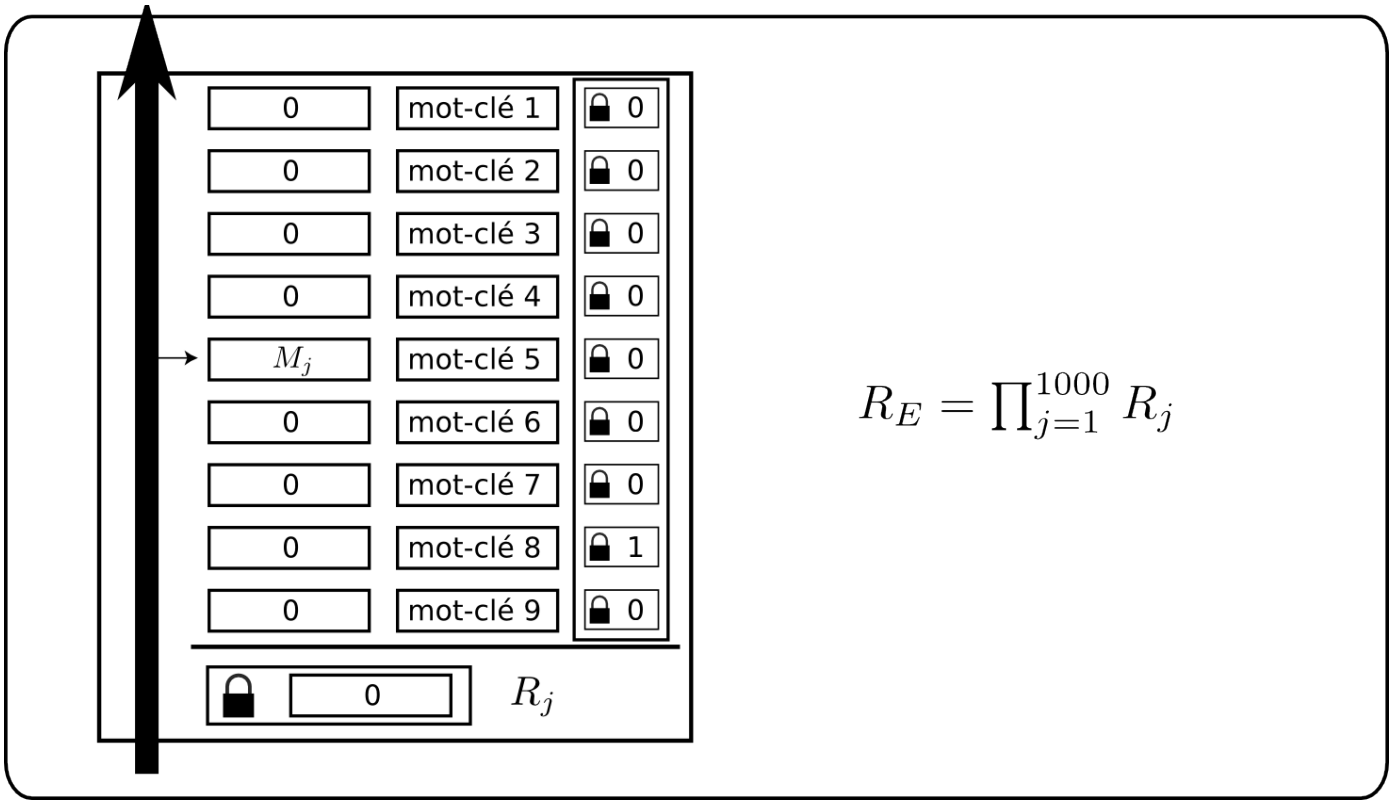
Serveur de VoIP



Performances

- Aguilar et Deswarte NCA'05, NCA'06, ACSAC'07.
- Permet de communiquer anonymement à des centaines d'utilisateurs avec les protocoles classiques.
- $x100$ si on réduit la taille des requêtes dans notre protocole RIP.

Ostrovsky et Skeith III CRYPTO'05



Performances

- Les éléments à zéro engendrent pas un coût calculatoire.
- $\Rightarrow T_L = T/C$, C nombre moyen de matchings par message.
- Requêtes grandes mais changent peu souvent.

Conclusion

- Des applications pratiques existent.
- Il faut se concentrer sur le coût calculatoire.
- L'algèbre linéaire permet d'avoir des protocoles très intéressants.

Questions

Merci de votre attention. Des questions ?

Bibliographie

http ://tinyurl.com/2wdlbx
http ://freehaven.net/anonbib