Autorité de certification distribuée pour des réseaux pair-à-pair structurés : modèle, mise en œuvre et exemples d'applications

François Lesueur

Soutenance de thèse
27 novembre 2009

Thèse préparée à Supélec dans l'équipe SSIR (EA 4039)

Sous la supervision de Ludovic Mé et Valérie Viet Triem Tong

Les réseaux pair-à-pair

Principe

- Systèmes distribués
- Pairs jouent les rôles de client et de serveur
- Agrégation des capacités des pairs

Utilisations

- Partage de fichiers
- Systèmes de fichiers distribués
- Voix sur IP
- Streaming audio/vidéo

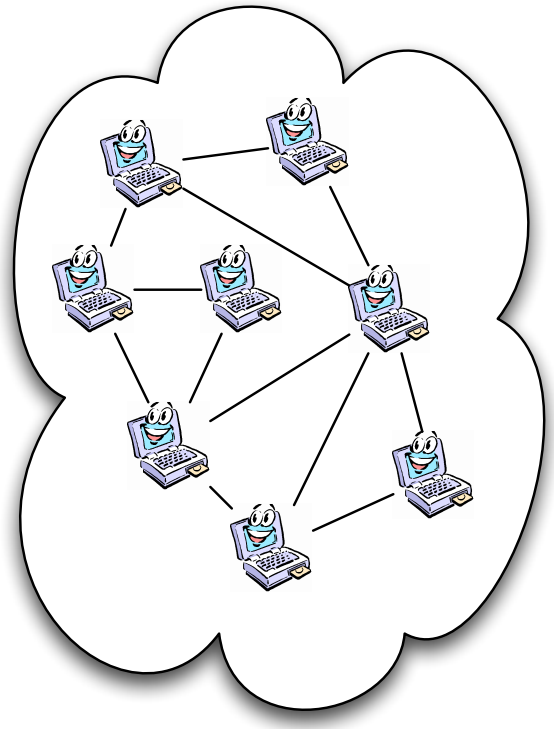
Caractéristiques des réseaux pair-à-pair

Mécanismes mis en œuvre

- Redondance des routes
- Réplication des données

Absence de centre

- Passage à l'échelle
- Haute disponibilité
- Faible coût



Impacts sur la sécurité

Sécurité et décentralisation ?

- Usuellement, les questions de sécurité sont décidées et contrôlées par une entité ponctuelle ou un consortium défini
- Nécessité de maintenir l'absence de point central

Mécanismes de sécurité sans point central

- Qui décide ?
- Qui contraint ?
- Comment ?

Contributions de cette thèse

Approche

- Autorité de certification distribuée
- Prises de décisions par un pourcentage fixé des membres
- Applications
  - à la protection contre l'attaque sybile
  - au nommage certifié des utilisateurs

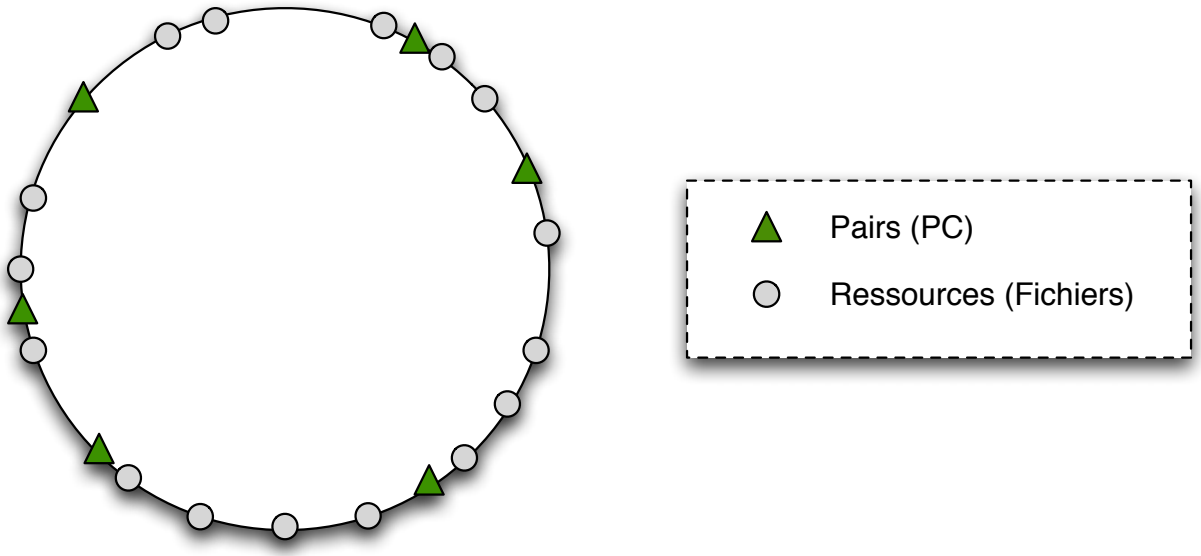
Plan

Plan

Notre cas d'étude : les DHT

Description d'une DHT (*Distributed Hash Table*)

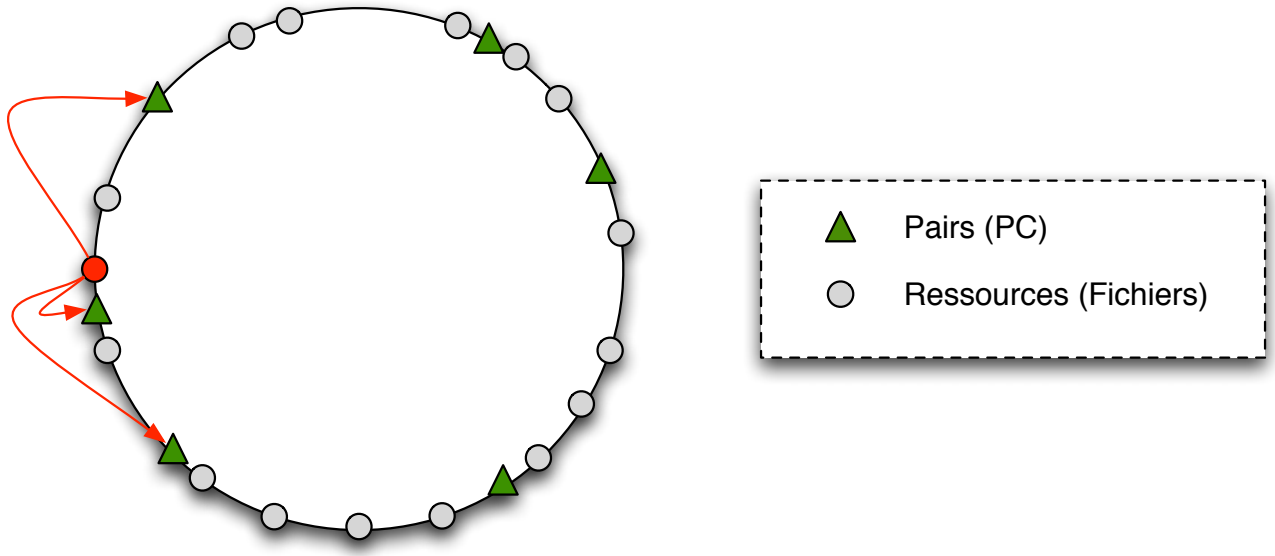
- Réseau pair-à-pair structuré
- Stockage des données avec réplication
- Insertion et obtention efficaces



Notre cas d'étude : les DHT

Description d'une DHT (*Distributed Hash Table*)

- Réseau pair-à-pair structuré
- Stockage des données avec réplication
- Insertion et obtention efficaces



Aperçu des problèmes de sécurité

Attaque sybile [Douceur, 02]

Un unique attaquant physique crée

- un grand nombre de pairs
- certains pairs spécifiques

⇒ Modification de ressources, filtrage des accès d'un pair

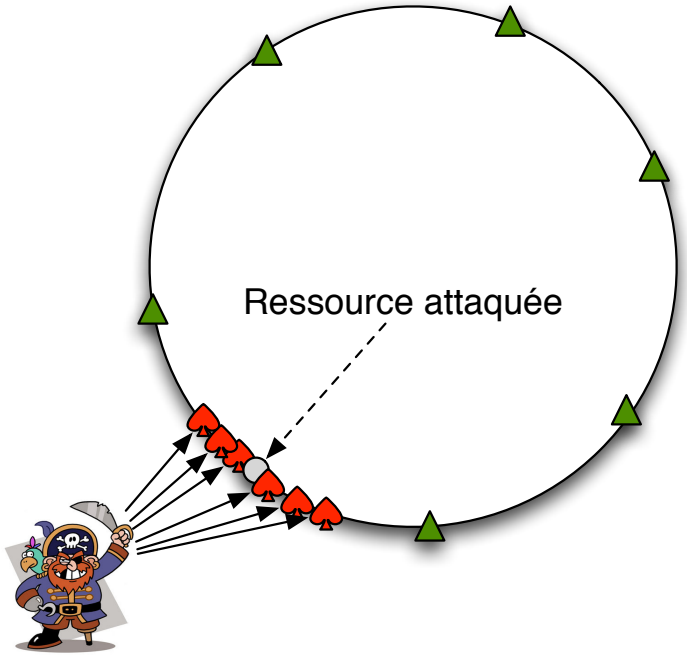
Gestion des clés cryptographiques

- Pas d'annuaire centralisé
- Obtention des clés ?

⇒ Confidentialité et intégrité des ressources

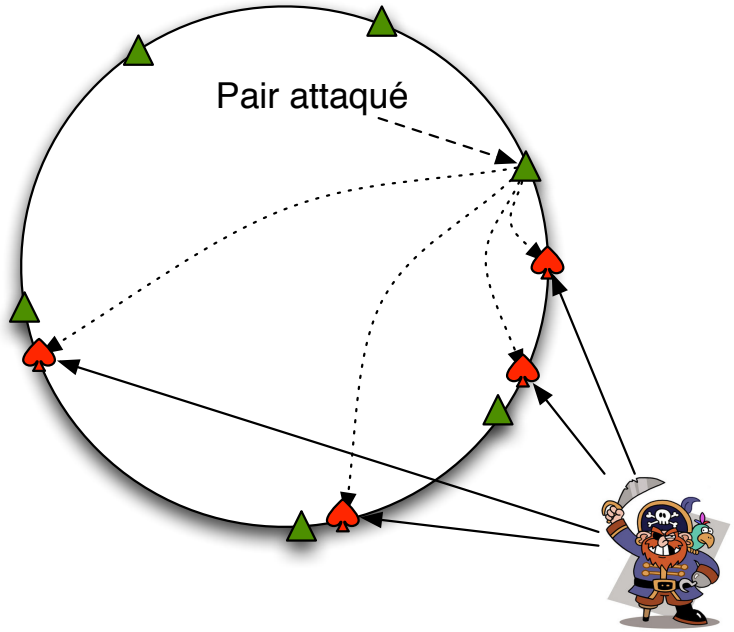
Attaque sybile

Exemple dans le cas de Kademlia



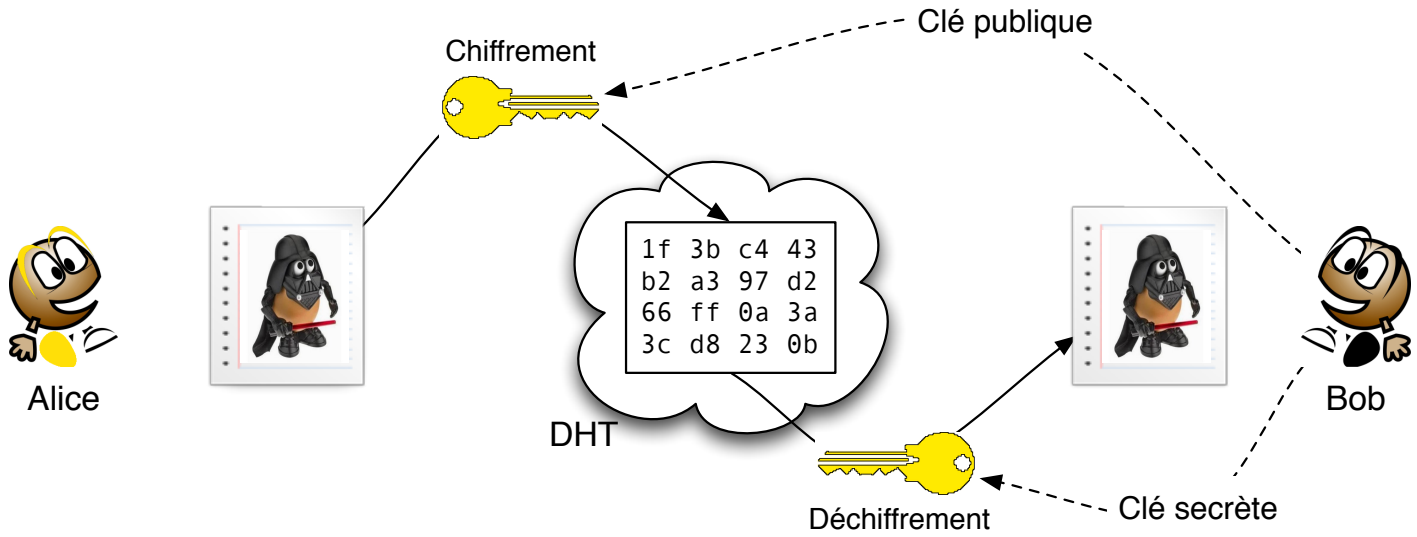
L'attaquant contrôle les accès à la ressource attaquée

Exemple dans le cas de Chord



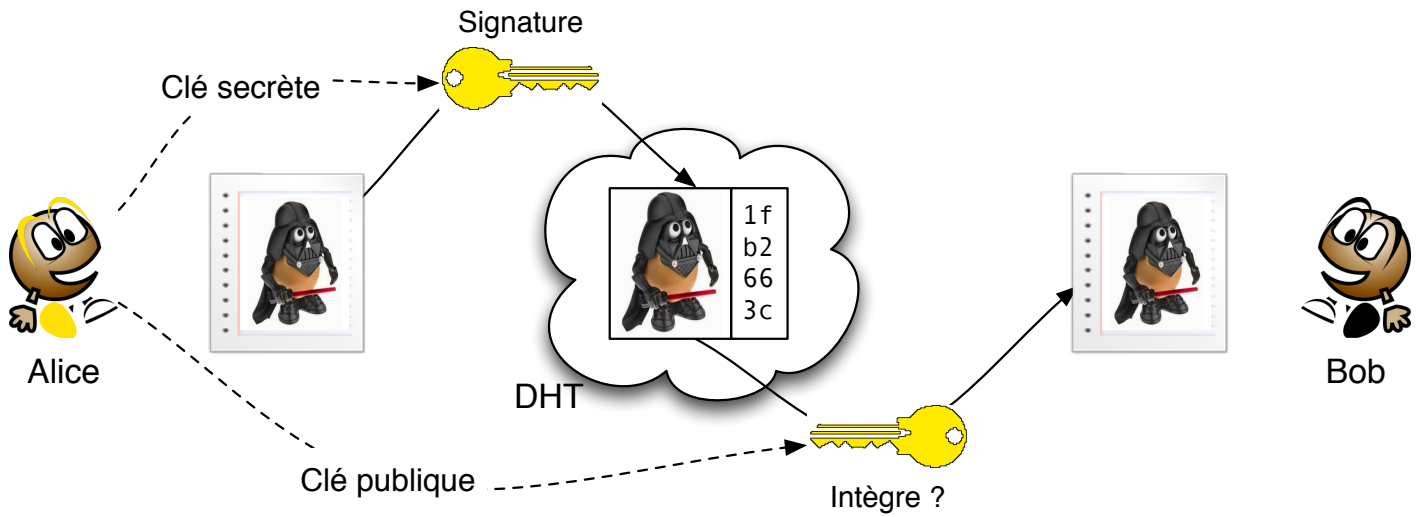
L'attaquant filtre les accès du pair attaqué

Confidentialité des ressources



Comment Alice obtient-elle la clé publique de Bob ?

Intégrité des ressources



Comment Bob obtient-il la clé publique d'Alice ?

Autorité de certification centralisée

Certificats

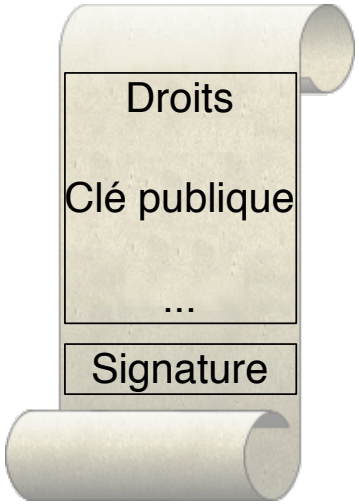
- Accès au réseau (protection sybile PAST)
- Possession d'un nom (FARSITE, SOSIMPLE)
- Permissions

Fonctionnement de l'autorité

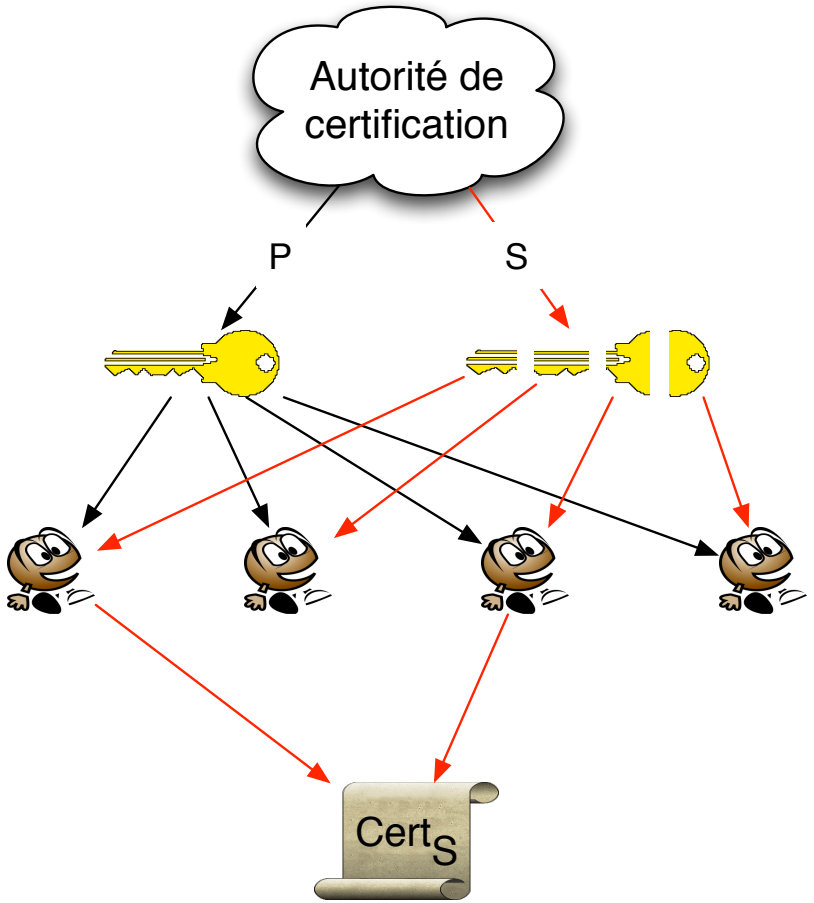
- *Décision* de la validité des requêtes
- *Signature* des certificats

Inadaptation aux réseaux pair-à-pair

- Centralisation
- Confiance



Cryptographie à seuil (t, n)



n = 4 entités

t = 2 fragments

Signature avec S

Vérification avec P

[Shamir, 79 ; Desmedt, 87]

Autorités de certification distribuées

Seuils fixes [Kong *et al.*, 01]

Les certificats sont signés par un nombre fixe de pairs
$\Rightarrow$ Pas d'adaptation à la taille du réseau

Seuils variables [Saxena *et al.*, 03]

Les certificats sont signés par un pourcentage fixe des pairs

- Communications par *broadcast*
- Utilisation d'un serveur de comptage

Notre proposition

Les certificats sont signés par un pourcentage fixe des pairs

- Communications limitées
- Pas de point central

Plan

Certification par un pourcentage des membres

Certification

Le droit d'accès, la possession d'un nom, etc., sont matérialisés par des certificats :

- Contenant la clé publique du pair
- Signés par la clé secrète du réseau

Signature du certificat

Les certificats sont signés par la coopération d'un pourcentage fixe des membres

- Communications limitées
- Pas de point central (la taille du réseau n'est pas connue)

Hypothèse : pas d'attaque sybile

Partage additif d'une clé RSA

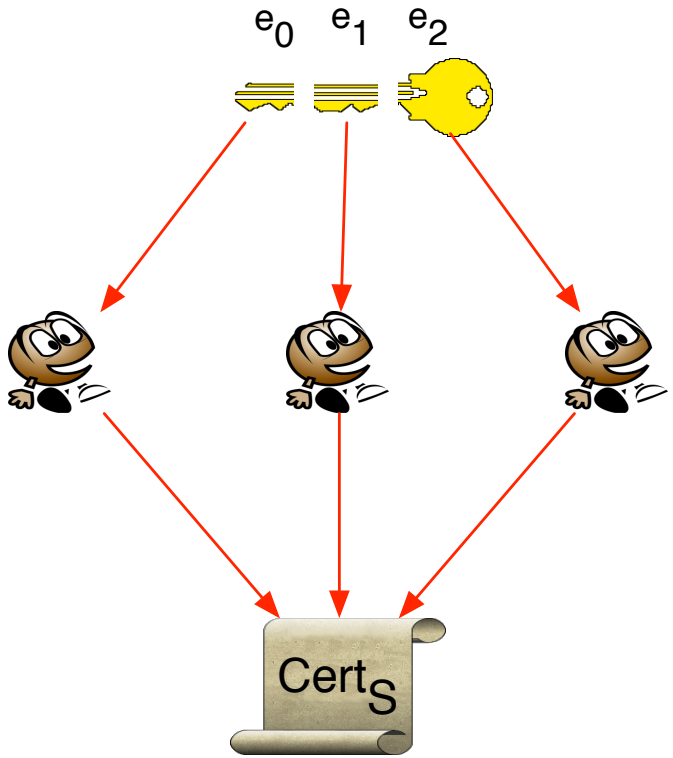
Signature RSA

Message o signé avec $S = (e, m)$:

$$h(o)^e[m]$$

Partage de la clé RSA

- $S = (e, m)$ est la clé privée du réseau
- $e_0, \ldots, e_s$ sont tels que $e = \sum e_i$
- $h(o)^e[m] = h(o)^{\sum e_i}[m] = \prod h(o)^{e_i}[m]$



[Boyd, 89 ; Frankel, 89]

Fragmentation de la clé dans le réseau

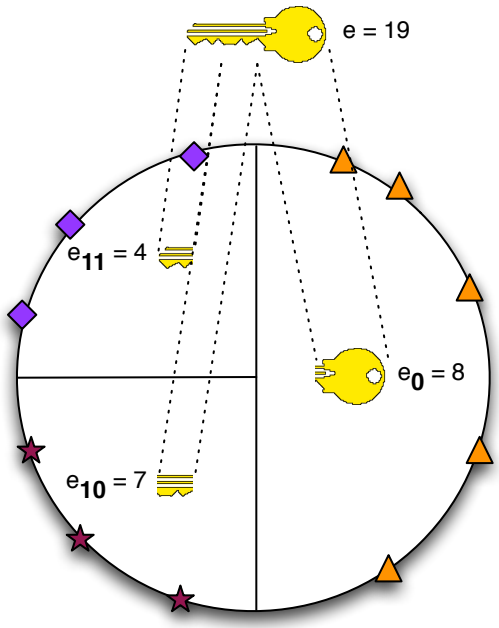
Groupes de fragmentation

- g_{min} à g_{max} membres
- Chaque groupe connaît un fragment

$\Rightarrow$ Un membre par groupe doit coopérer

Obtention d'un ratio fixe t

- t est le ratio de pairs devant coopérer
- g_{min} et g_{max} tailles limites des groupes
- $\frac{1}{g_{max}} \leq t \leq \frac{1}{g_{min}}$



$g_{min} = 3$, $g_{max} = 6$,
$\frac{1}{6} \leq t \leq \frac{1}{3}$

Fragmentation de la clé dans le réseau

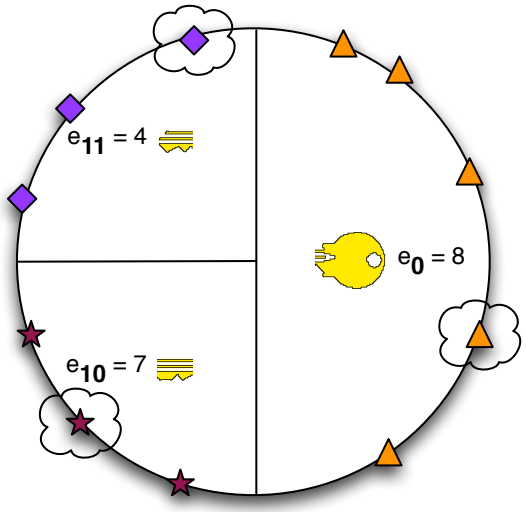
Groupes de fragmentation

- g_{min} à g_{max} membres
- Chaque groupe connaît un fragment

$\Rightarrow$ Un membre par groupe doit coopérer

Obtention d'un ratio fixe t

- t est le ratio de pairs devant coopérer
- g_{min} et g_{max} tailles limites des groupes
- $\frac{1}{g_{max}} \leq t \leq \frac{1}{g_{min}}$



$g_{min} = 3$, $g_{max} = 6$,
$\frac{1}{6} \leq t \leq \frac{1}{3}$

Fragmentation de la clé dans le réseau

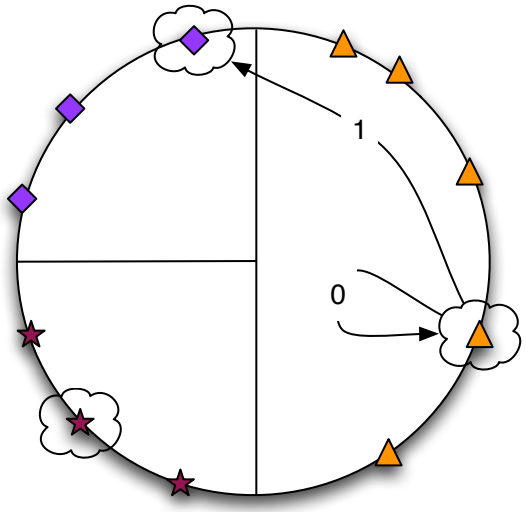
Groupes de fragmentation

- g_{min} à g_{max} membres
- Chaque groupe connaît un fragment

$\Rightarrow$ Un membre par groupe doit coopérer

Obtention d'un ratio fixe t

- t est le ratio de pairs devant coopérer
- g_{min} et g_{max} tailles limites des groupes
- $\frac{1}{g_{max}} \leq t \leq \frac{1}{g_{min}}$



$g_{min} = 3$, $g_{max} = 6$,
$\frac{1}{6} \leq t \leq \frac{1}{3}$

Fragmentation de la clé dans le réseau

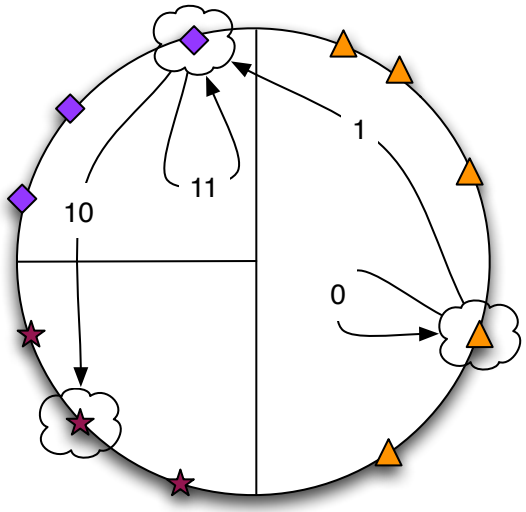
Groupes de fragmentation

- g_{min} à g_{max} membres
- Chaque groupe connaît un fragment

$\Rightarrow$ Un membre par groupe doit coopérer

Obtention d'un ratio fixe t

- t est le ratio de pairs devant coopérer
- g_{min} et g_{max} tailles limites des groupes
- $\frac{1}{g_{max}} \leq t \leq \frac{1}{g_{min}}$



$g_{min} = 3$, $g_{max} = 6$,
$\frac{1}{6} \leq t \leq \frac{1}{3}$

Opérations de maintenance

Principe

- Ratio contraint par la taille des groupes ($\frac{1}{g_{max}} < t < \frac{1}{g_{min}}$)
- Créer/Supprimer des groupes de fragmentation

Trois opérations

- Division : création de deux groupes à partir d'un
- Rafraîchissement : modification de deux fragments
- Fusion : création d'un groupe à partir de deux

Opération de division

Principe

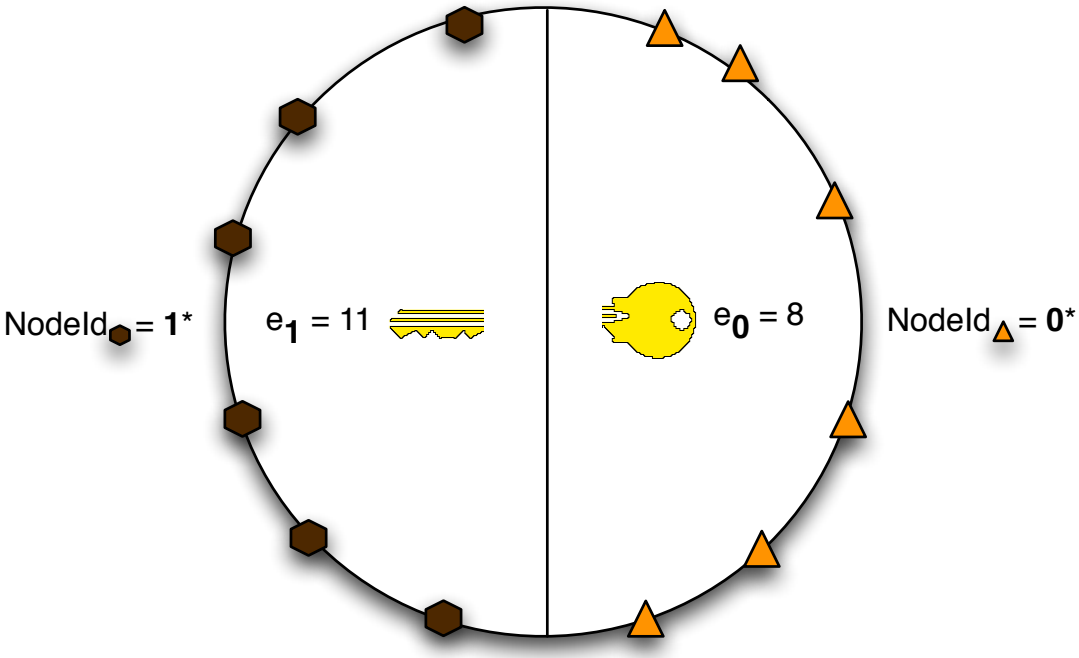
Diviser un fragment en deux quand un groupe est composé de plus de g_{max} membres

Division du fragment e_i

- Créer deux fragments e_{i0} et e_{i1} tels que $e_i = e_{i0} + e_{i1}$
- Chaque pair de e_i choisit l'un des deux nouveaux groupes
- Le fragment e_i disparaît du partage de la clé

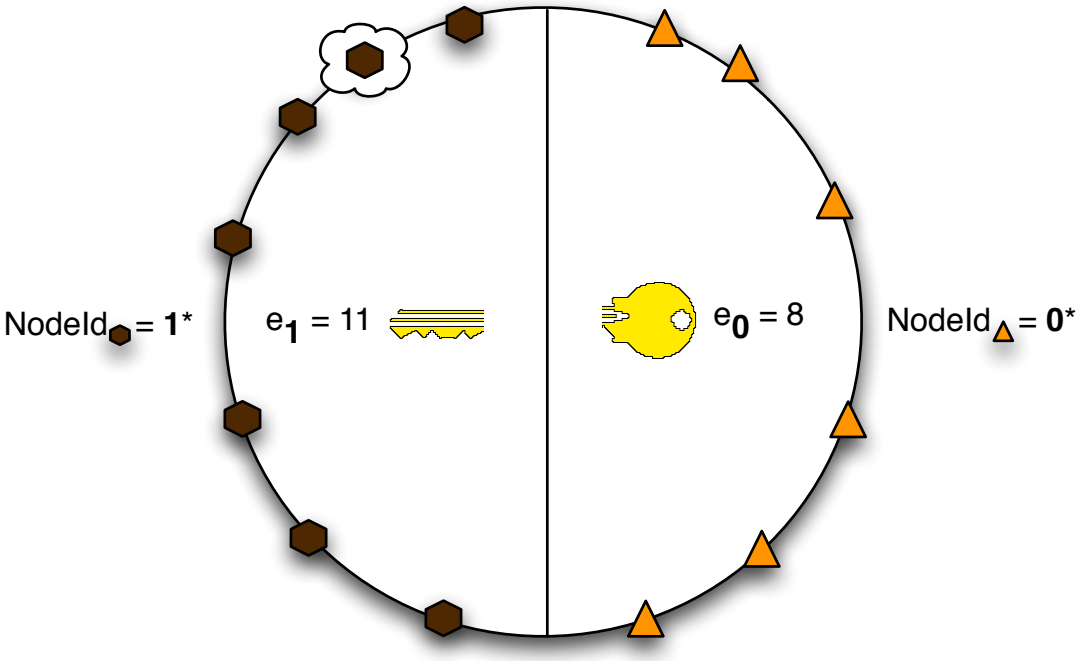
La somme des fragments vaut toujours e

Division d'un fragment



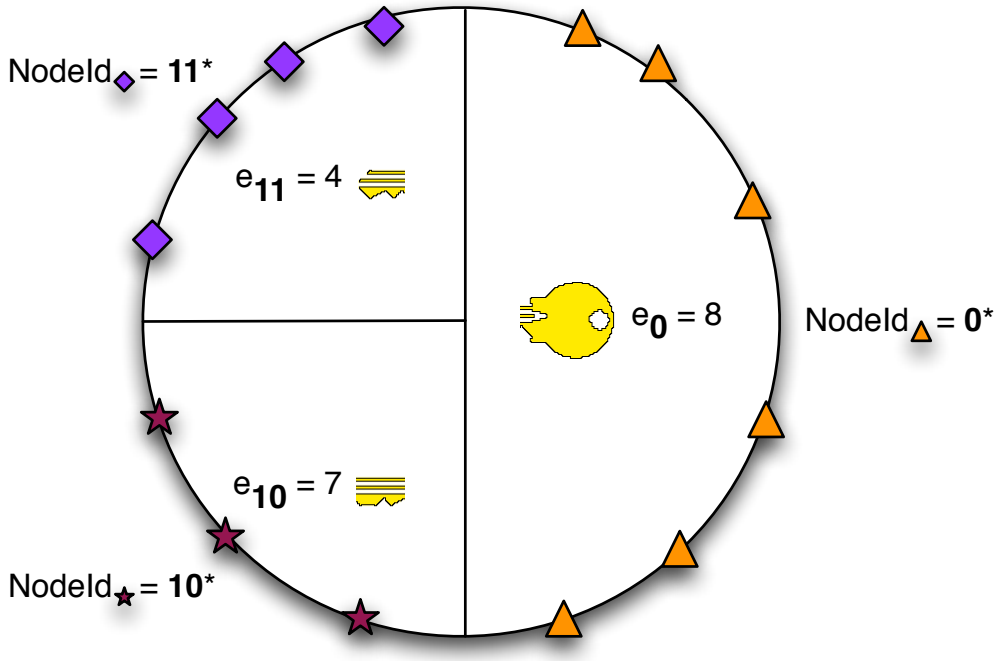
g_{min}	g_{max}	t_{min}	t_{max}	t_{eff}
3	6	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{6} = t_{min}$

Division d'un fragment



g_{min}	g_{max}	t_{min}	t_{max}	t_{eff}
3	6	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{2}{13} < t_{min}$

Division d'un fragment



g_{min}	g_{max}	t_{min}	t_{max}	t_{eff}
3	6	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{3}{13} > t_{min}$

Opération de rafraîchissement

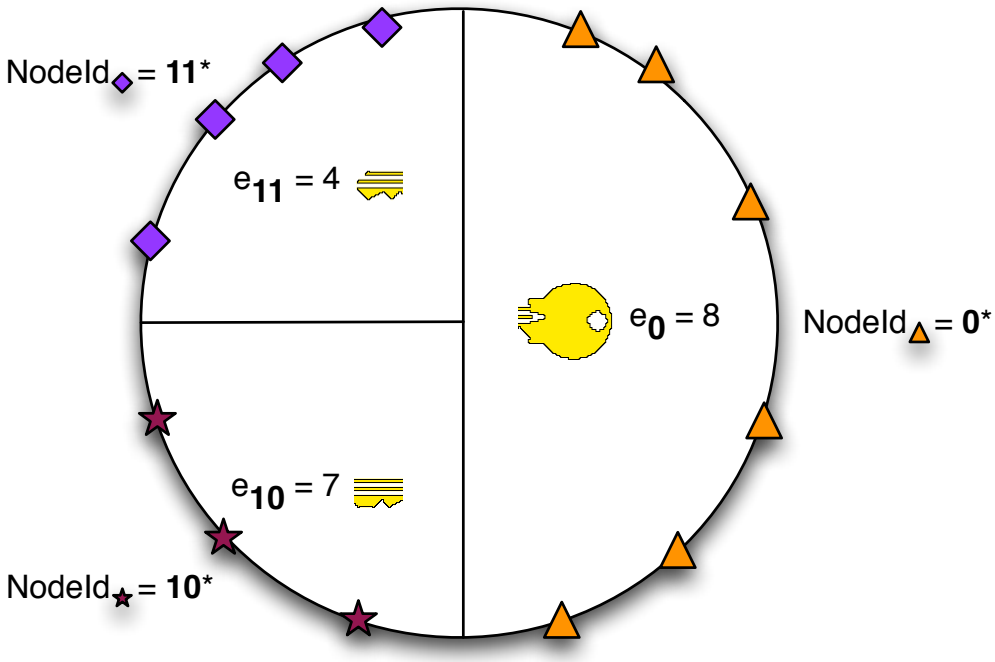
Problème

À l'issue d'une division, les deux groupes créés connaissent les deux nouveaux fragments.

Fonctionnement

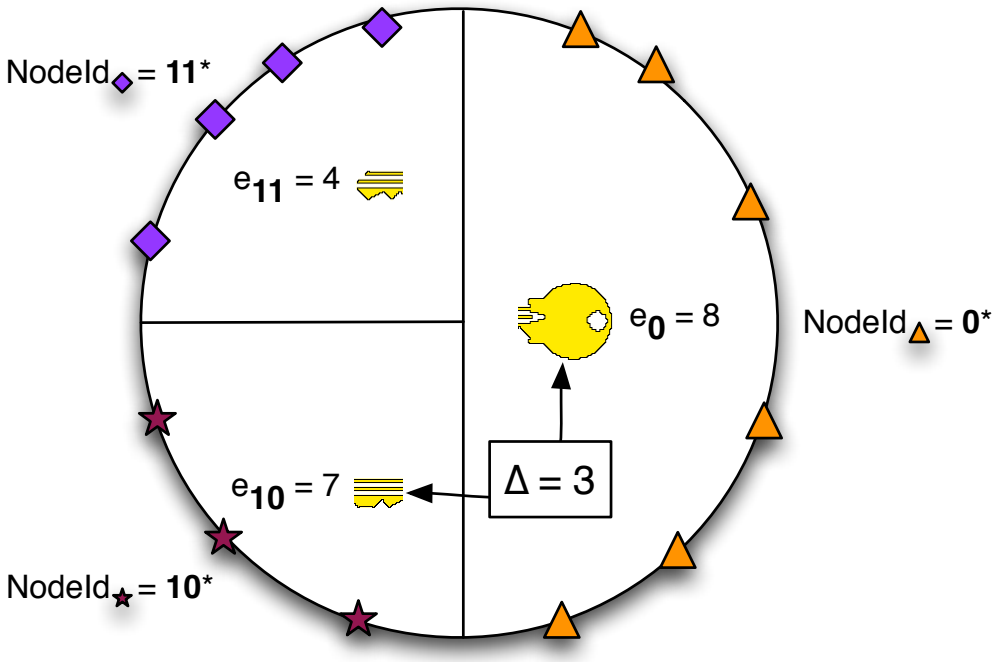
- Demande de rafraîchissement à un autre groupe quelconque
- Échange d'une valeur Δ aléatoire et secrète
- Soustraction/Addition de cette valeur au fragment

Rafraîchissement de deux fragments



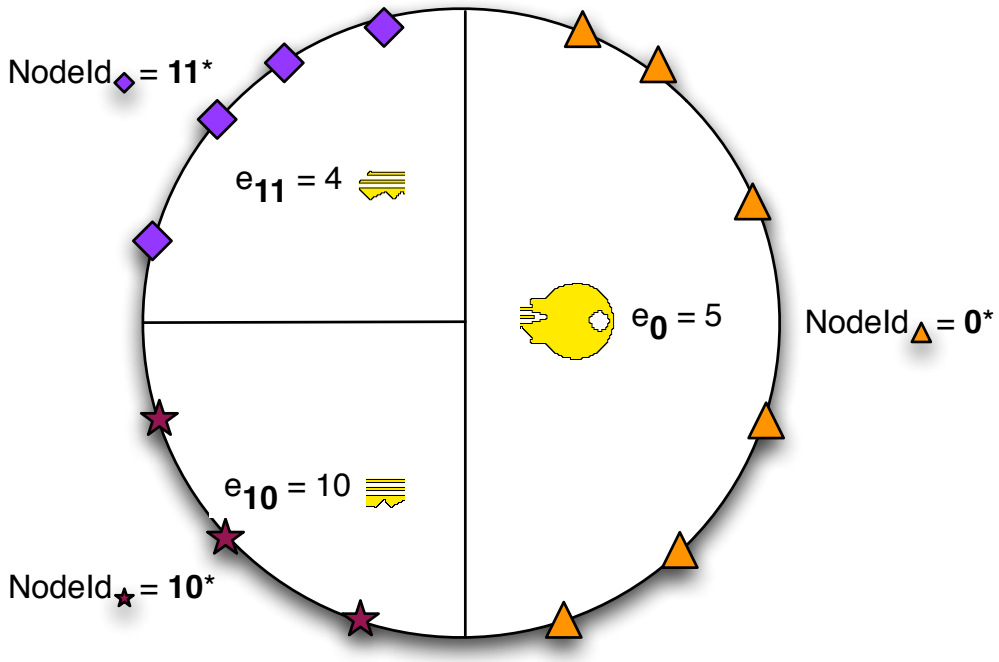
Les pairs de e_{11} connaissent e_{10}

Rafraîchissement de deux fragments



Δ est échangé entre e_{10} et e_0

Rafraîchissement de deux fragments



Les pairs de e_{11} ne connaissent plus e_{10}

Opération de fusion

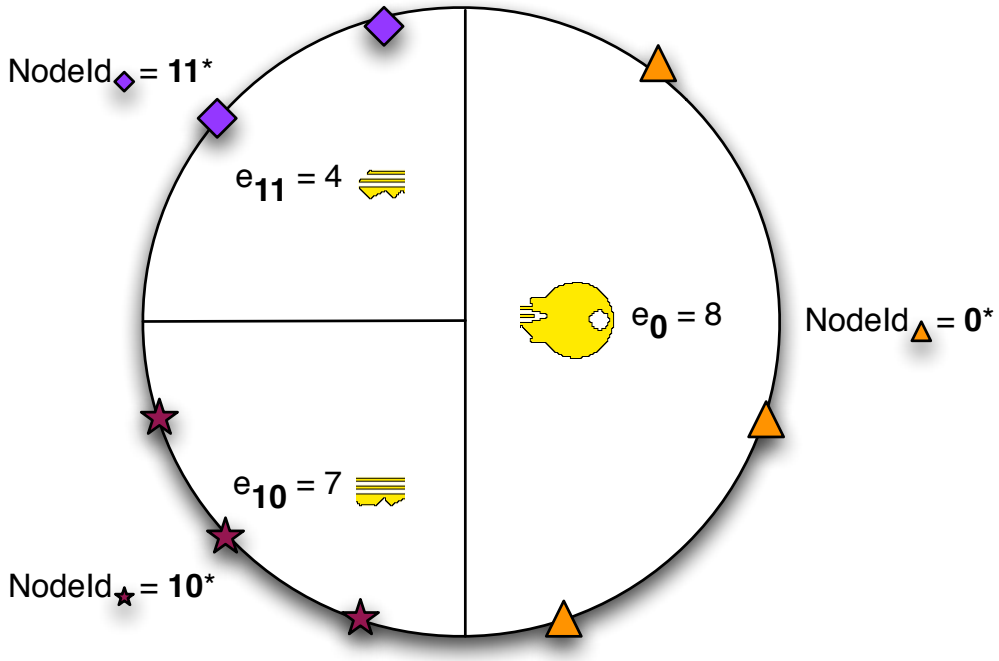
Principe

Réunir 2 groupes de fragmentation quand l'un des deux possède moins de g_{min} membres

Fonctionnement

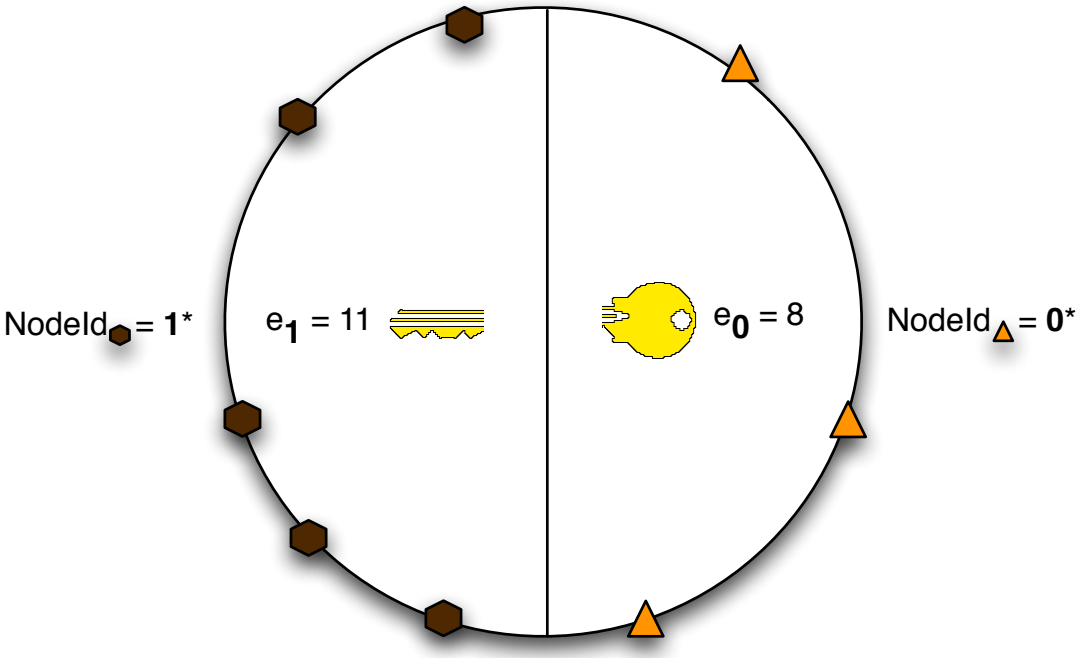
- Demande de fusion au groupe adjacent
- Calcul du nouveau fragment somme
- Entrée des membres des deux groupes dans le nouveau groupe

Fusion de deux fragments



g_{min}	g_{max}	t_{min}	t_{max}	t_{eff}
3	6	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{3}{8} > t_{max}$

Fusion de deux fragments



g_{min}	g_{max}	t_{min}	t_{max}	t_{eff}
3	6	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{2}{8} < t_{max}$

Problème de la maintenance

Problème

L'implémentation de ces opérations nécessiterait

- Une synchronisation entre les opérations initiées
- Une synchronisation entre les pairs de chaque groupe
- Des consensus en présence de pairs byzantins

$\Rightarrow$ Problème difficile en fonction du nombre de pairs

Principe des arbres de fragmentation

Proposition

Les *arbres de fragmentation* permettent la maintenance

- Sans consensus
- Sans synchronisation

Contraintes

- Choix locaux et prédéterminés
- Confidentialité des fragments
- Opérations simultanées

Description des arbres de fragmentation

Définition

- Arbres binaires
- $e_i = e_{i0} + e_{i1}$
- Définissent les fragments à utiliser après division

Construction

- Initialement : $e_{i0} = RNG_{h(e_i)}$, $e_{i1} = e_i - e_{i0}$
- Les rafraîchissements modifient les feuilles

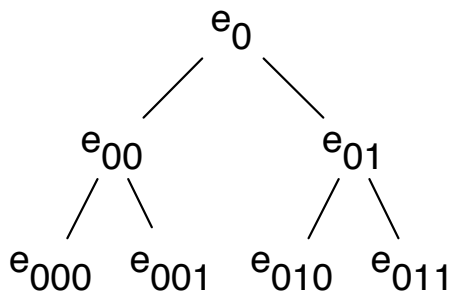
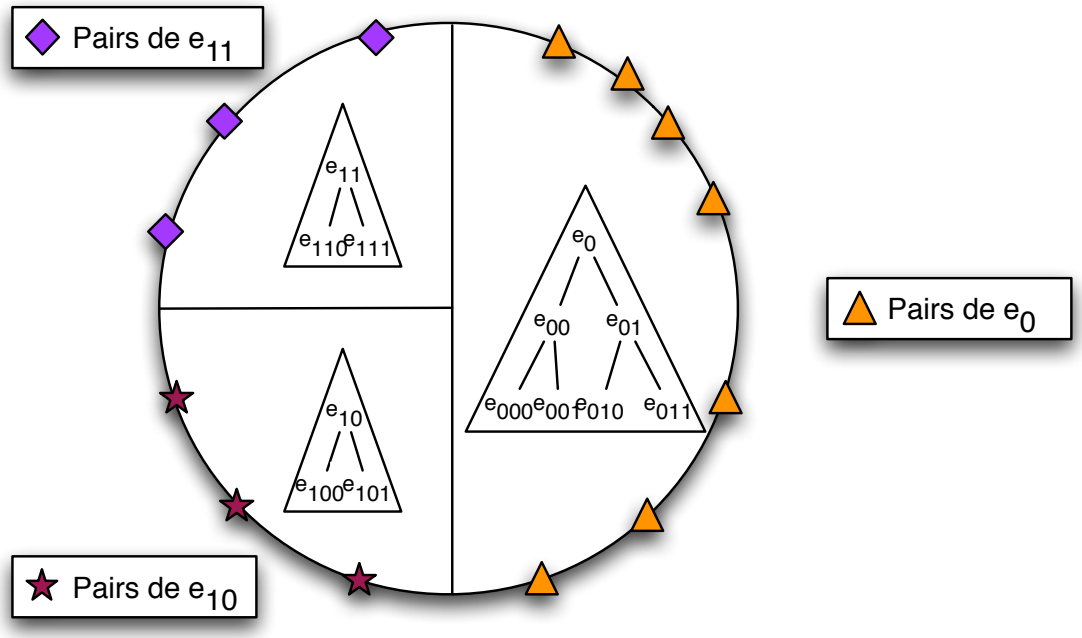
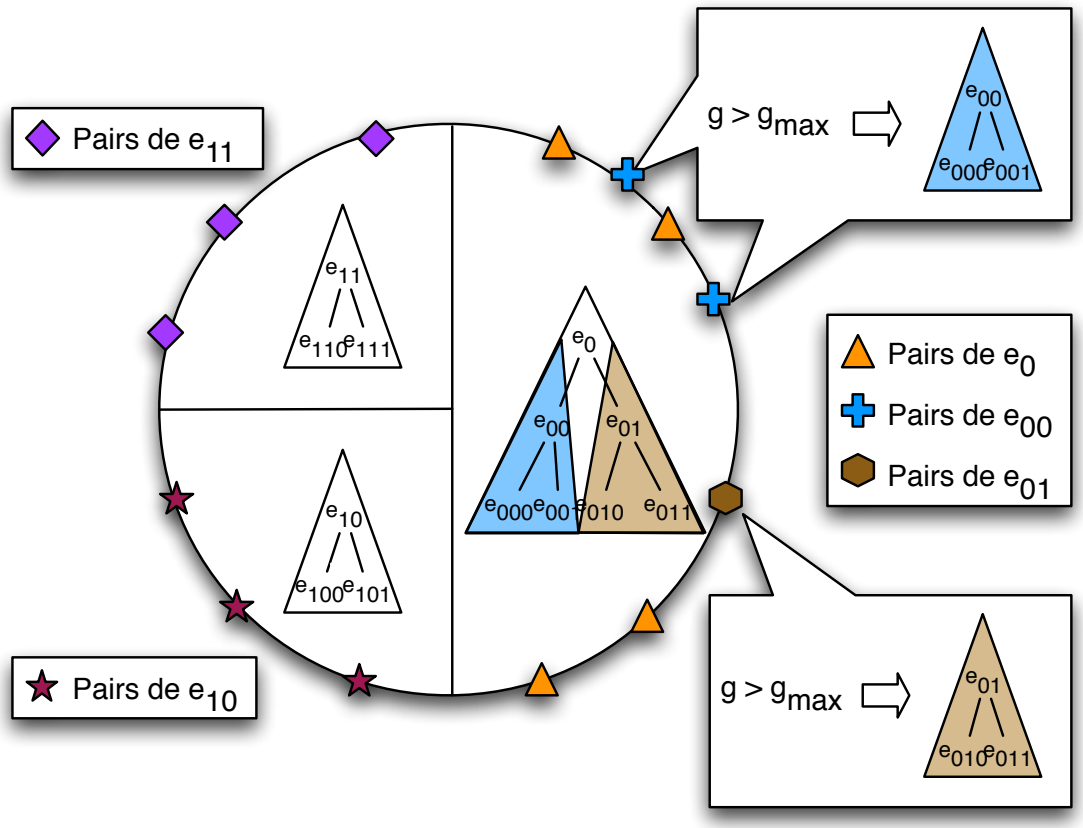


Illustration des arbres de fragmentation



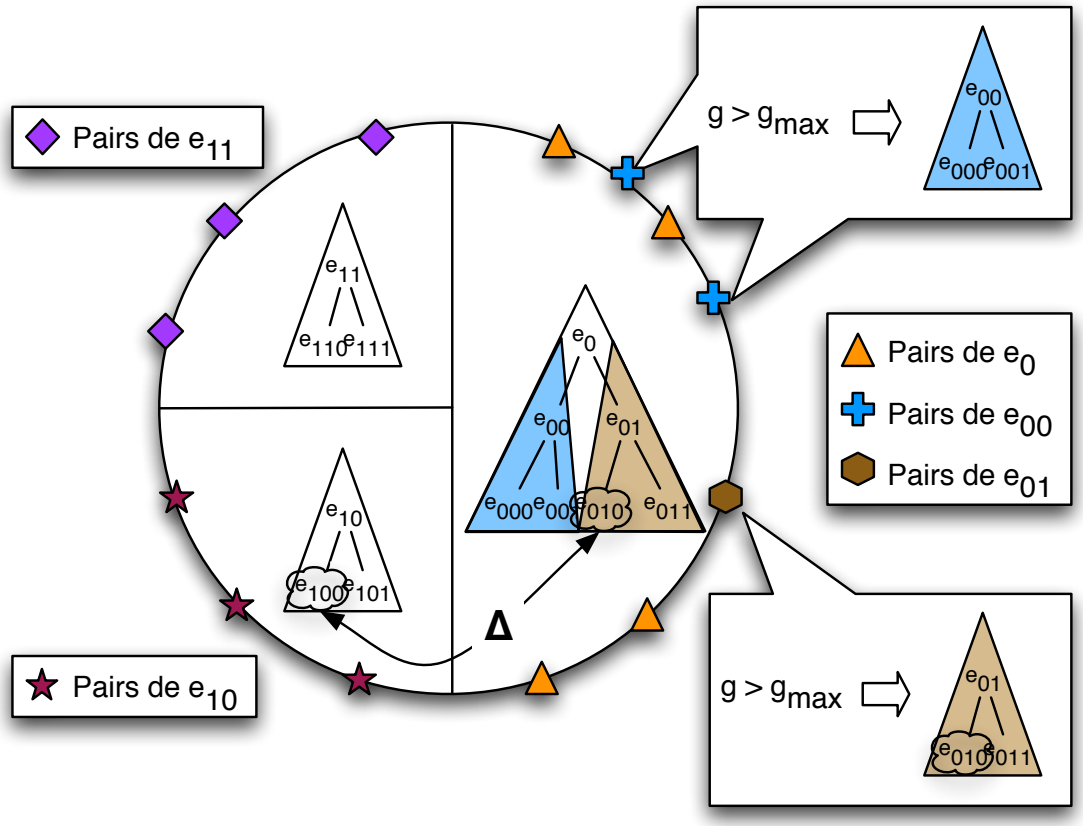
Le groupe e_0 doit se diviser

Illustration des arbres de fragmentation



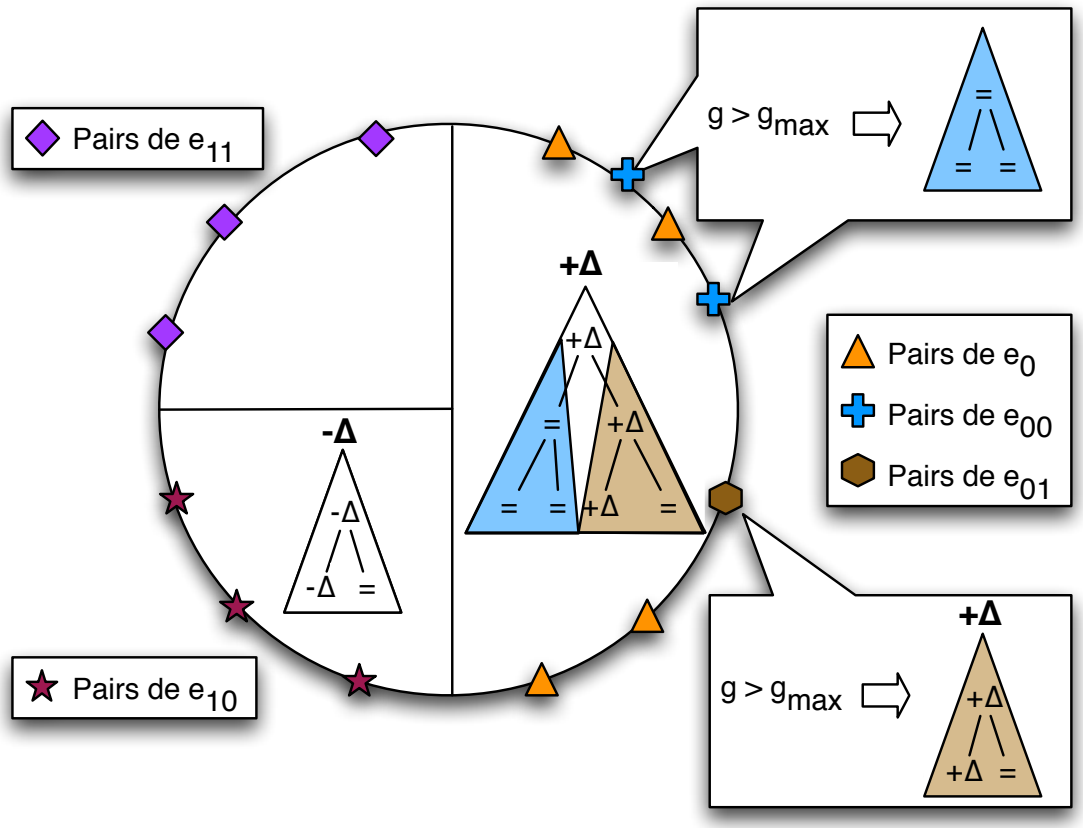
Les pairs de e_0 divisent le groupe progressivement

Illustration des arbres de fragmentation



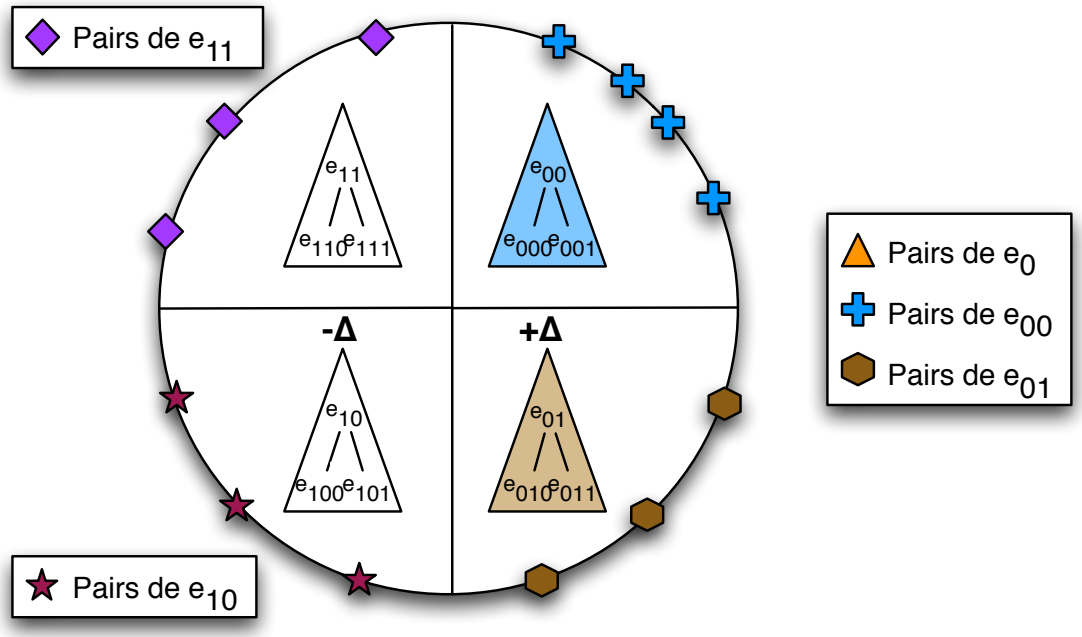
Un rafraîchissement est initié entre e_{100} et e_{010}

Illustration des arbres de fragmentation



Les fragments e_0, e_{01} et e_{10} changent ; pas e_{00}

Illustration des arbres de fragmentation



La division se termine de manière cohérente

Propriétés prouvées

Propriétés

- Confidentialité des fragments : $\forall e_i \in E, \forall p \in P \backslash \mathring{e}_i, p \not\triangleleft e_i$
- Cohérence du partage : $\forall e_i \in E, \forall p \in \mathring{e}_i, p \triangleleft e_i$
- Intégrité du partage : $\sum_{e_i \in E} e_i = e$

Notations :

- P est l'ensemble des pairs
- E est l'ensemble des fragments courants
- $\mathring{e}_i$ est le groupe connaissant le fragment e_i
- $\forall p \in P$, $p \triangleleft x$: le pair p connaît la valeur x
- $\forall p \in P$, $p \not\triangleleft x$: le pair p ne connaît pas la valeur x

Bilan de l'autorité de certification distribuée

Fragmentation de la clé

- Décomposition de la clé en fragments additifs
- Distribution des fragments dans les groupes de fragmentation
- Signature par coopération d'un pair de chaque groupe

Maintenance

- Maintient la taille des groupes entre g_{min} et g_{max} et donc le ratio t entre $\frac{1}{g_{max}}$ et $\frac{1}{g_{min}}$
- Opérations réalisées sans consensus ni synchronisation

Aspects analysés

Robustesse de la fragmentation

- Partage de la clé entre les groupes de fragmentation
- Résistance aux attaquants internes

Évaluation de la maintenance

- Algorithmes de maintenance
- Sécurité des fragments générés et efficacité des algorithmes

Robustesse de la fragmentation de la clé

Obtention de la clé secrète

- L'obtention de la clé secrète permet de signer de faux certificats
- Cette clé est révélée s'il y a un attaquant dans chaque groupe

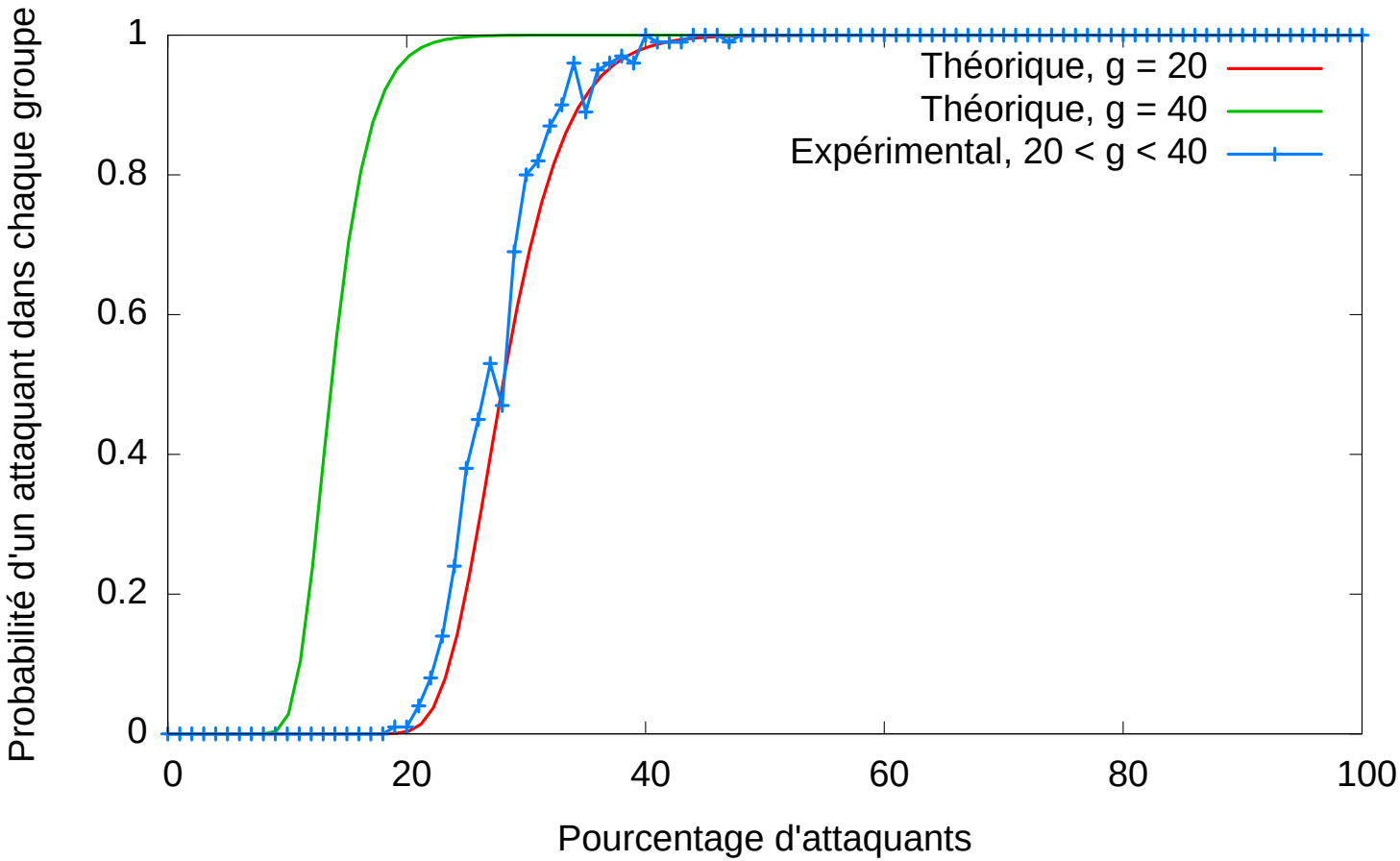
Destruction du partage

- Chaque fragment de la clé est utilisé pour signer un certificat
- Si un groupe ne contient que des attaquants, la signature n'est plus possible

Conditions expérimentales

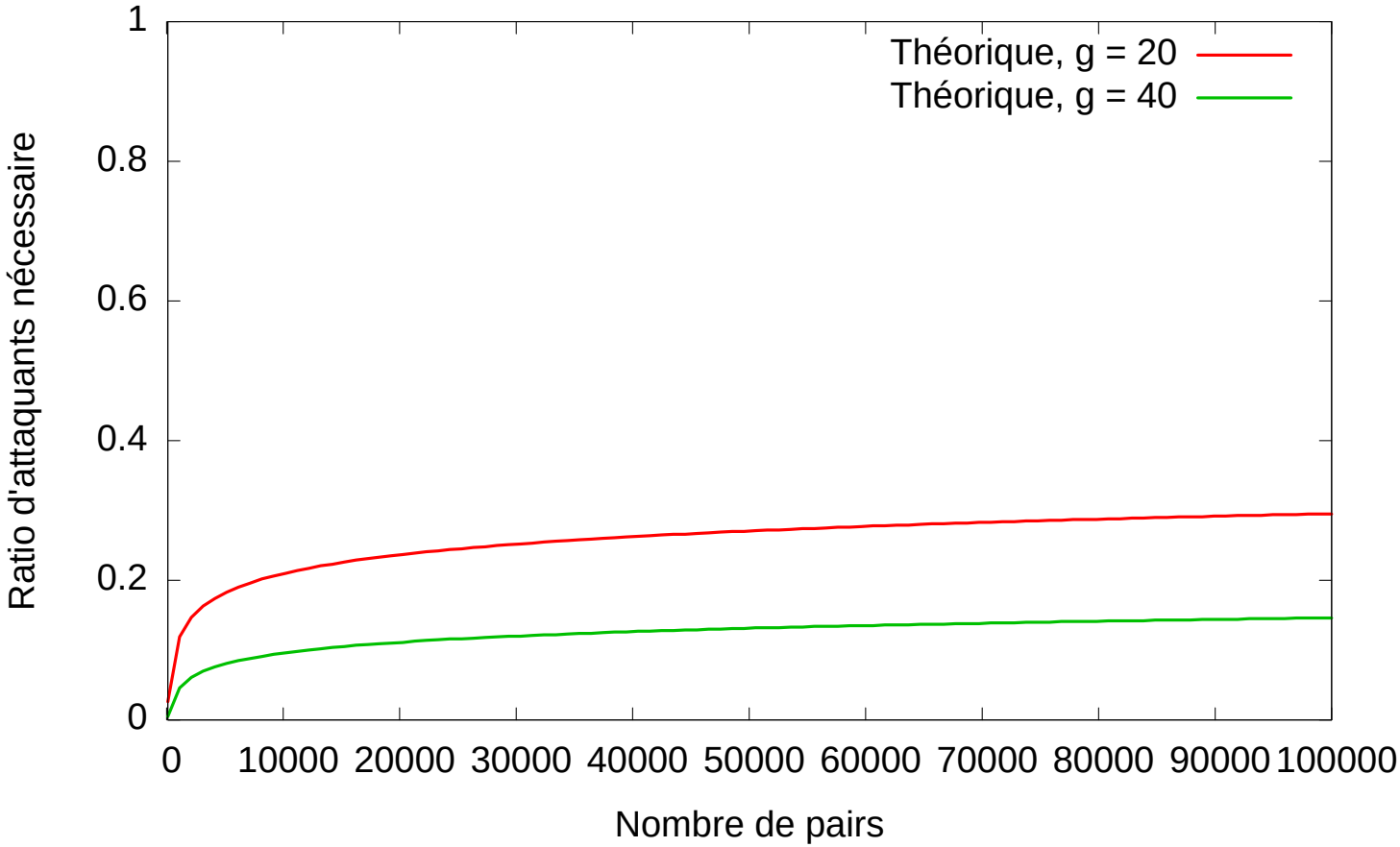
- 10.000 pairs
- $g_{min} = 20$, $g_{max} = 40$
- $2.5\% < t < 5\%$

Obtention de la clé secrète



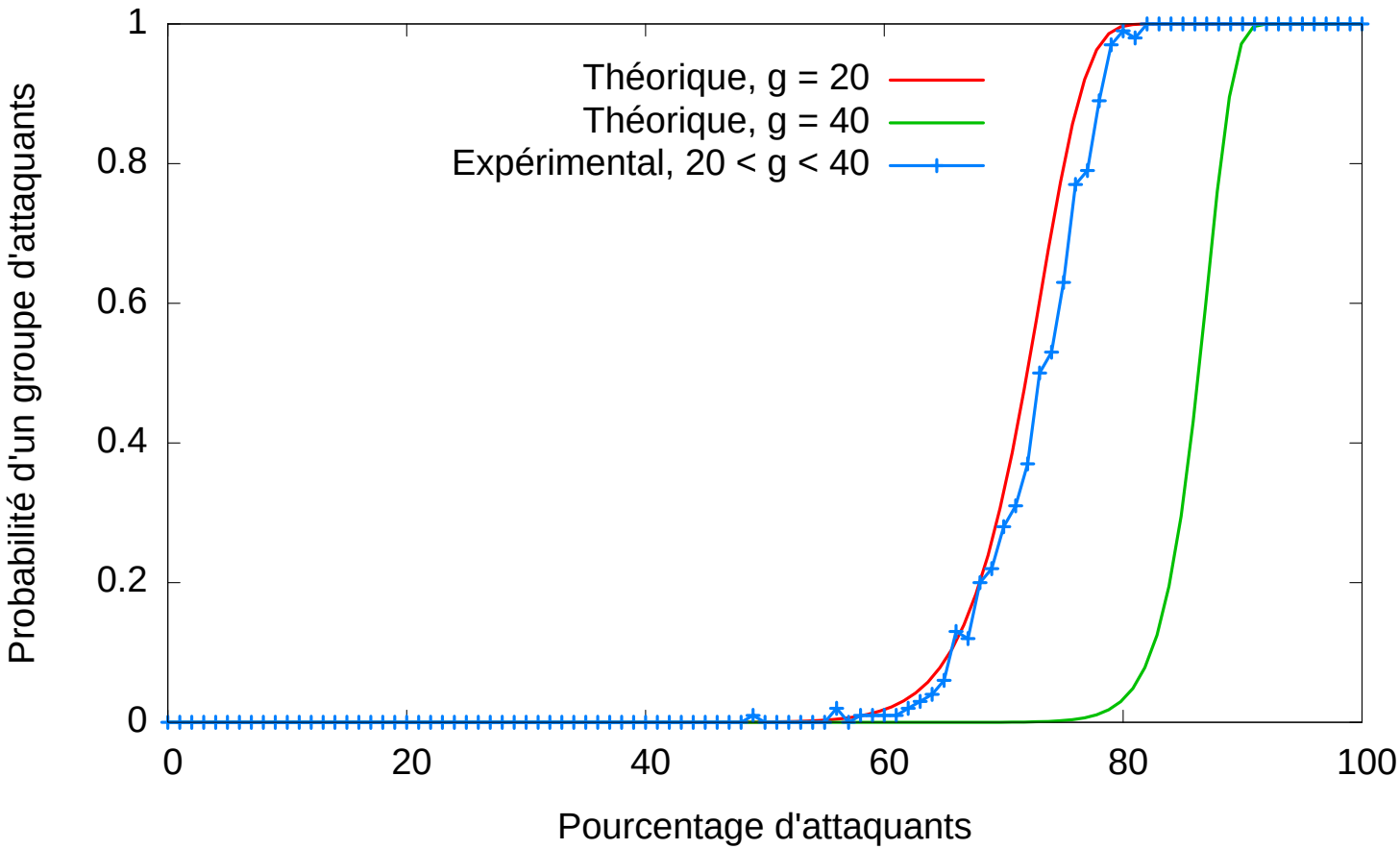
$$P_{revealed} = \prod_{i=1}^{s} 1 - (1-k)^{g_i}$$

Obtention de la clé secrète



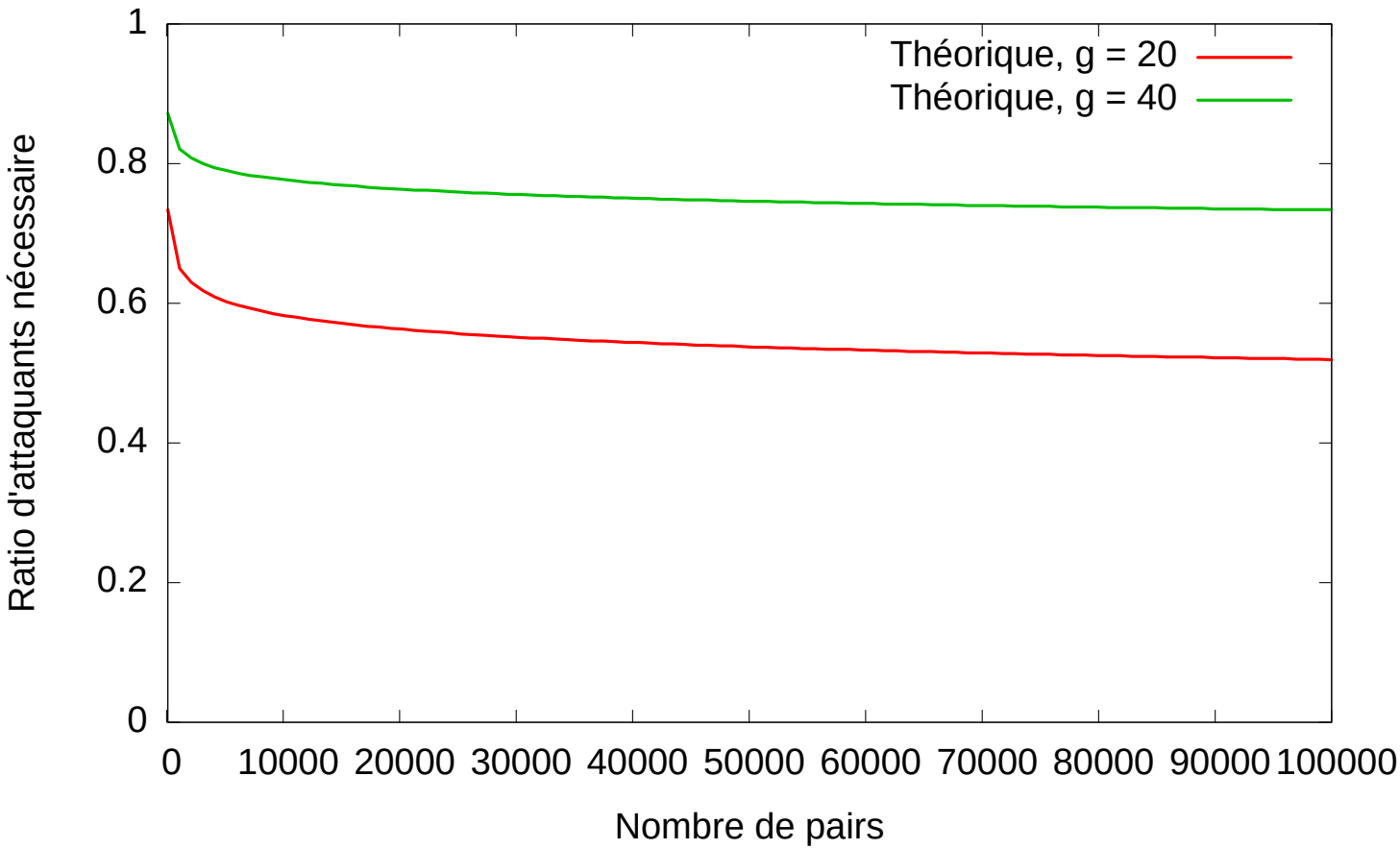
$$P_{revealed} = \prod_{i=1}^{s} 1 - (1-k)^{g_i}$$

Destruction du partage



$$P_{broken} = 1 - \prod_{i=1}^{s}(1 - k^{g_i})$$

Destruction du partage



$$P_{broken} = 1 - \prod_{i=1}^{s}(1 - k^{g_i})$$

Algorithmes de maintenance

Taille des fragments

- Chaque division crée deux fragments à partir d'un seul
- La taille des fragments doit rester élevée

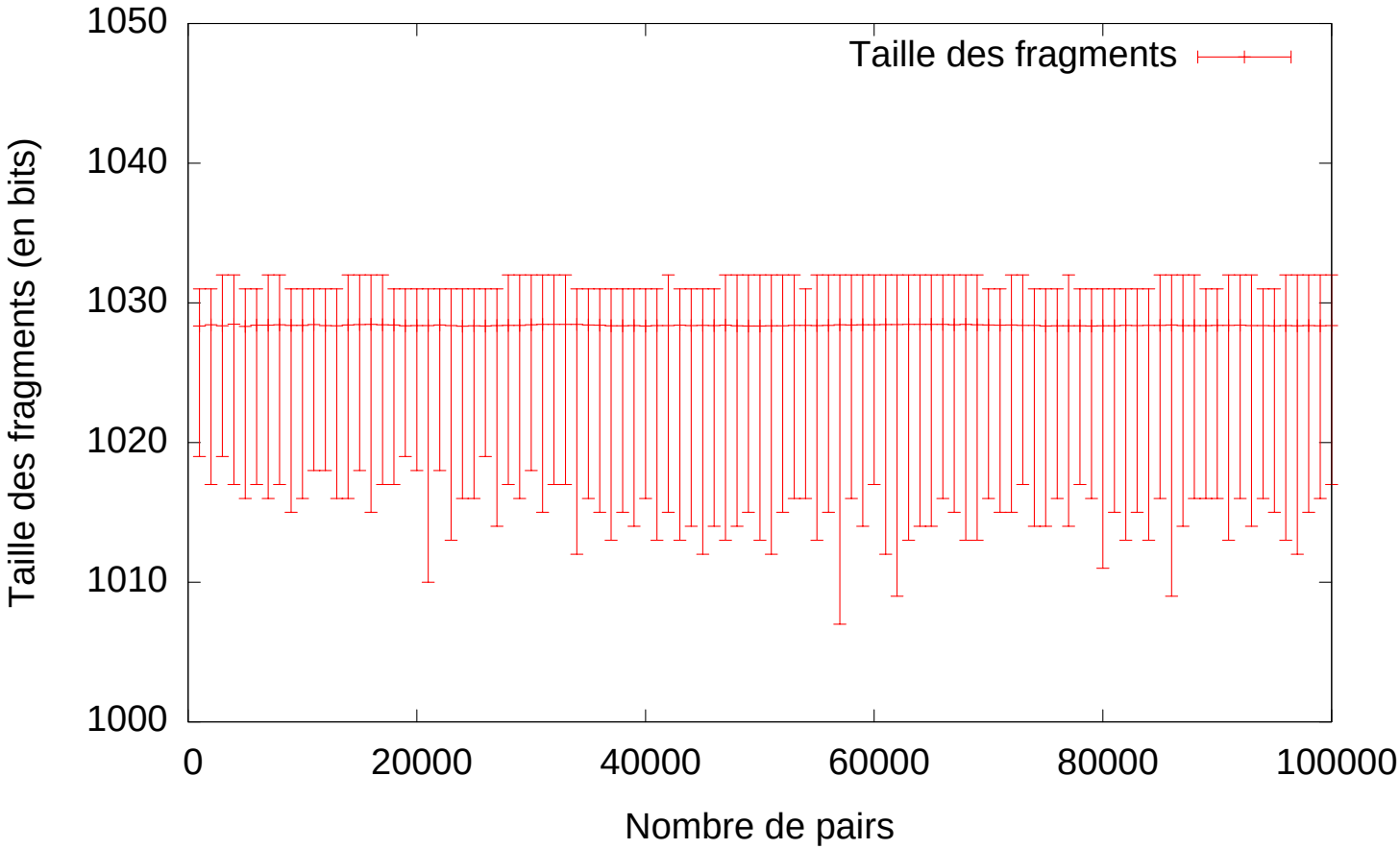
Taille des arbres de fragmentation

- Les arbres possèdent une partie stockée explicitement
- Ils doivent être stockables et transférables

Conditions expérimentales

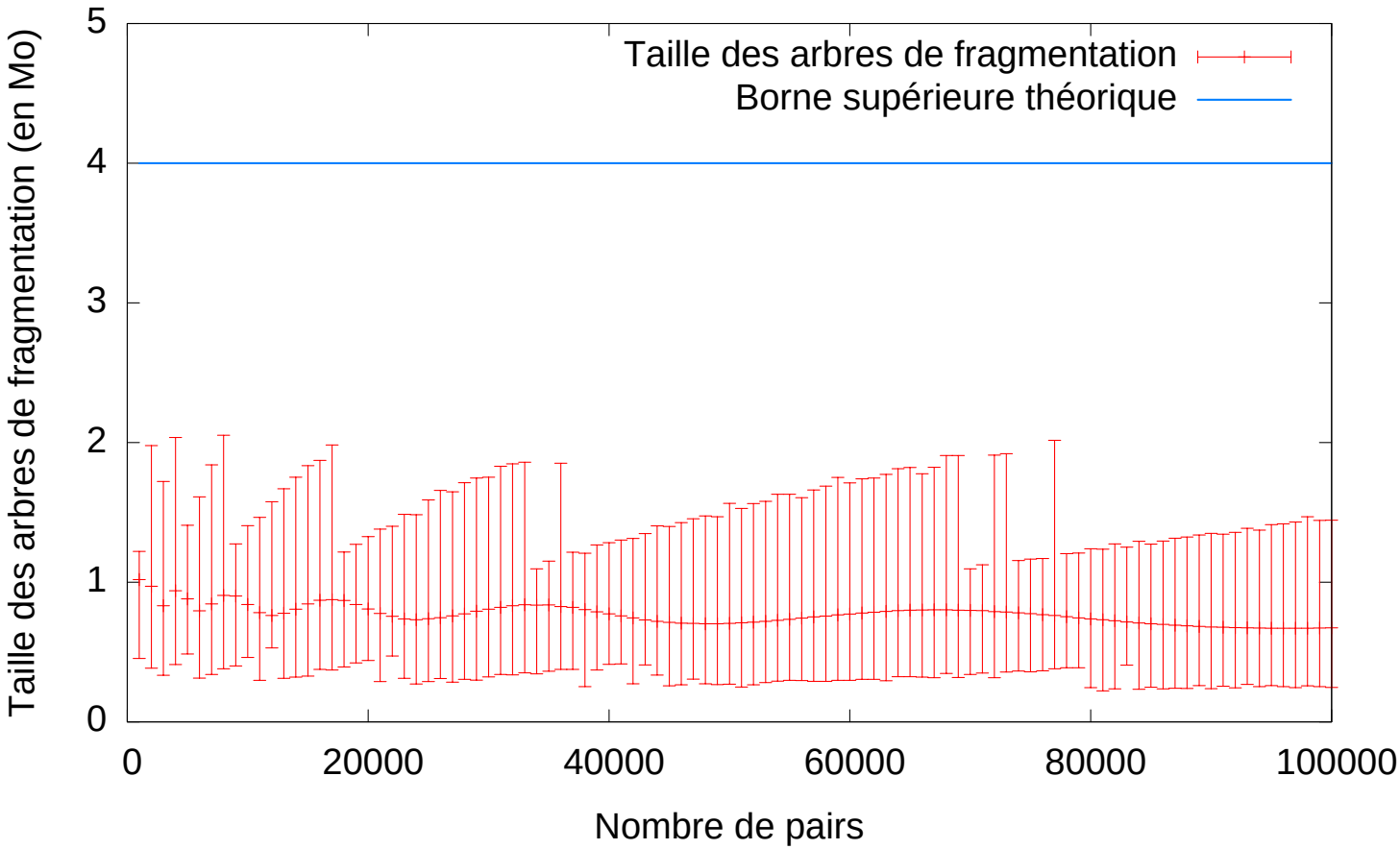
- $g_{min} = 20$, $g_{max} = 40$
- $2.5\% < t < 5\%$

Taille des fragments



Clé et valeurs aléatoires : 1024 bits

Taille des arbres de fragmentation



Clé et valeurs aléatoires : 1024 bits

Bilan des expérimentations

Robustesse de la fragmentation

- 20% d'attaquants pour obtenir la clé
- 60% d'attaquants pour détruire le partage

⇒ Bonne résistance aux attaquants internes

Évaluation de la maintenance

- Taille des fragments élevée
- Taille des arbres de fragmentation suffisamment basse

⇒ Fragments sûrs, maintenance efficace

Plan

Principe de notre proposition

Attaque sybile

Un unique attaquant physique crée :

- un grand nombre de pairs
- certains pairs spécifiques

Protection

- Limiter le nombre d'identifiants de chaque participant
- Contraindre l'aléa de ces identifiants

Notre proposition

- Réutilisation de SybilGuard pour limiter les identifiants
- Couplage avec l'autorité distribuée pour l'aléa

SybilGuard [Yu *et al.*, 06]

Caractéristiques

- Protection sociale
- Limite le nombre d'identifiants

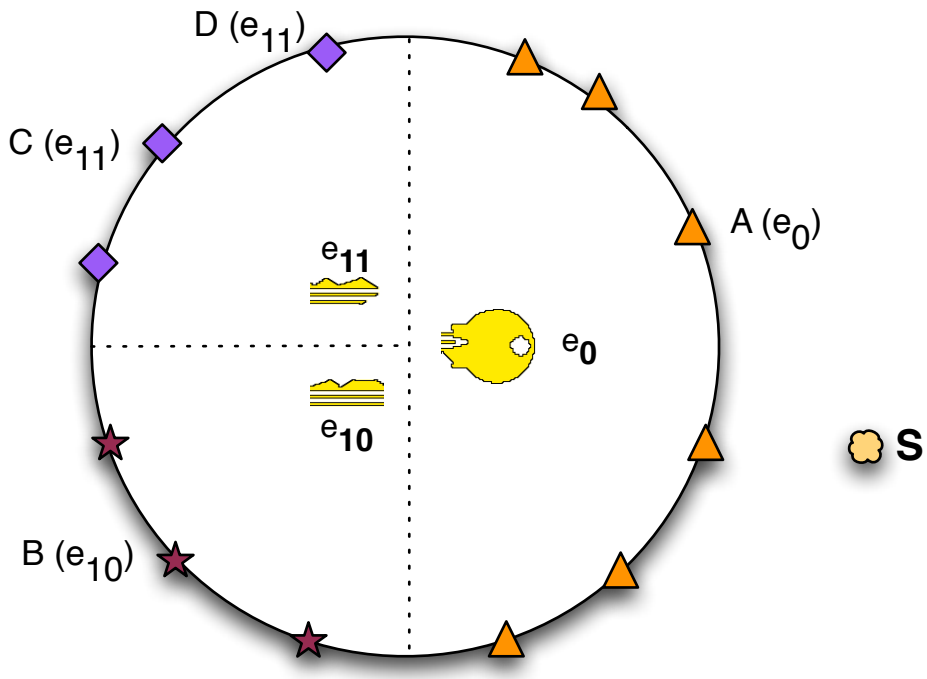
Fonctionnement

- Création d'arcs de confiance avec ses amis
- Évaluation des pairs suspects du point de vue local

Inconvénients

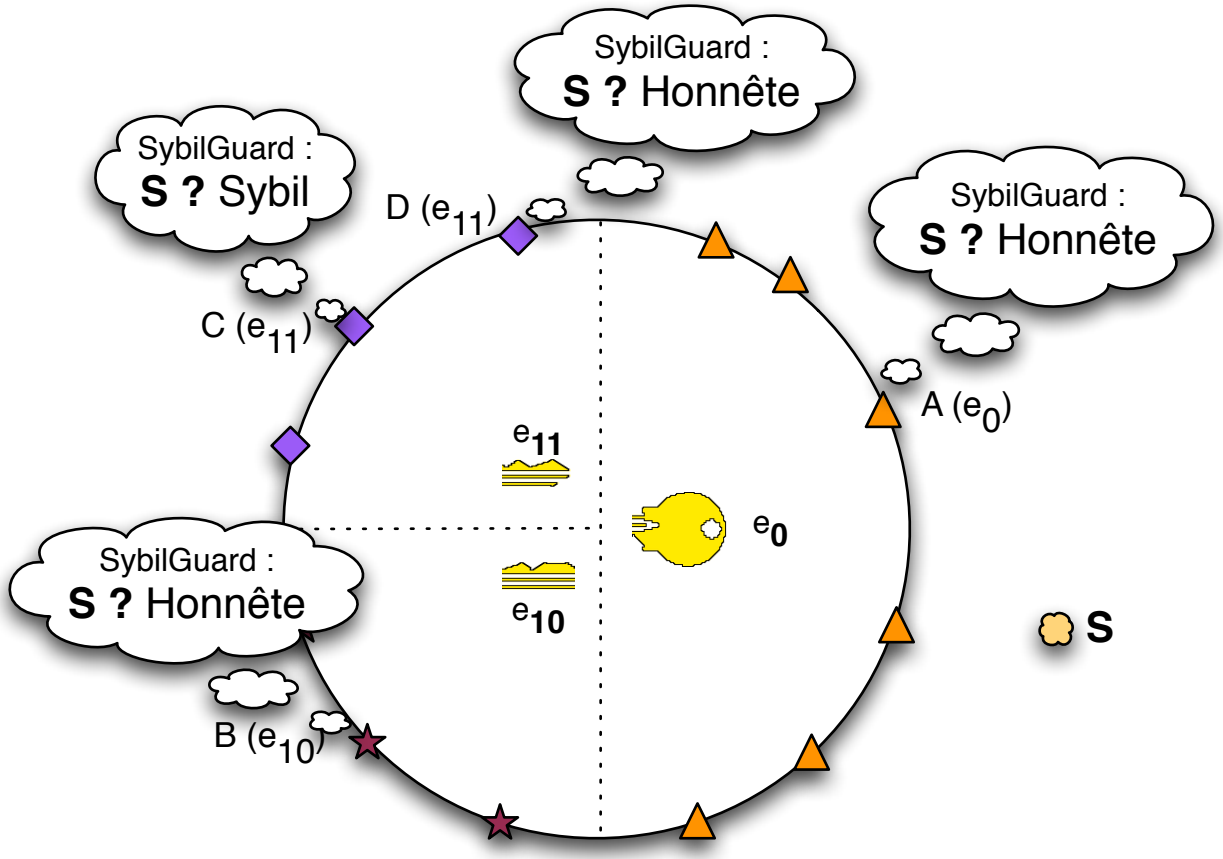
- Pas de contraintes sur les identifiants choisis
- Pas de cohérence globale des décisions

Contrôle d'admission d'un nouveau pair



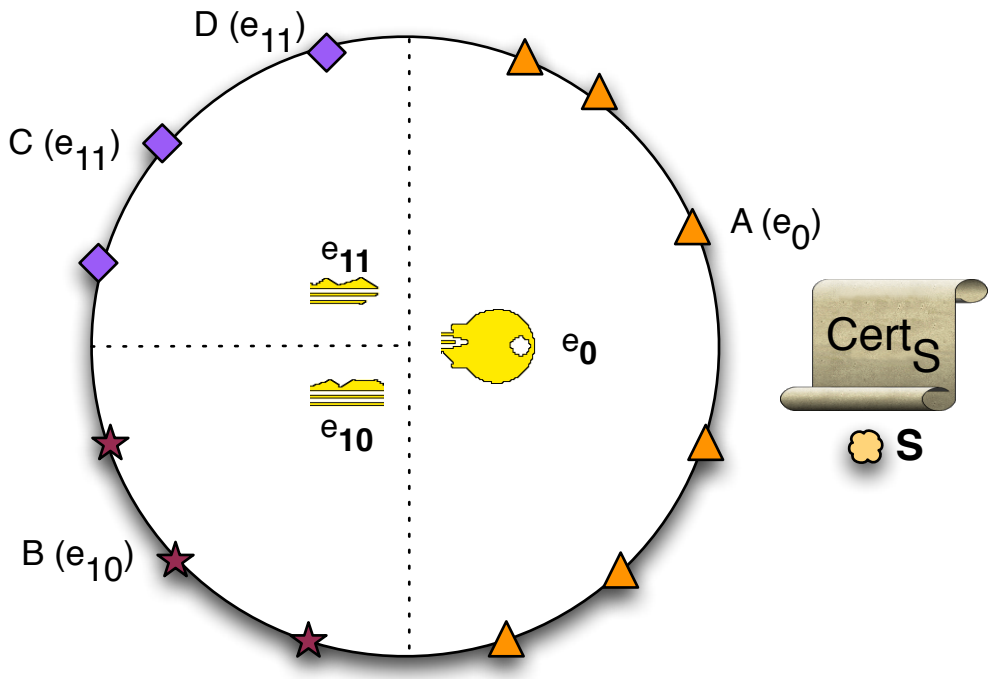
Un pair S souhaite intégrer le réseau

Contrôle d'admission d'un nouveau pair



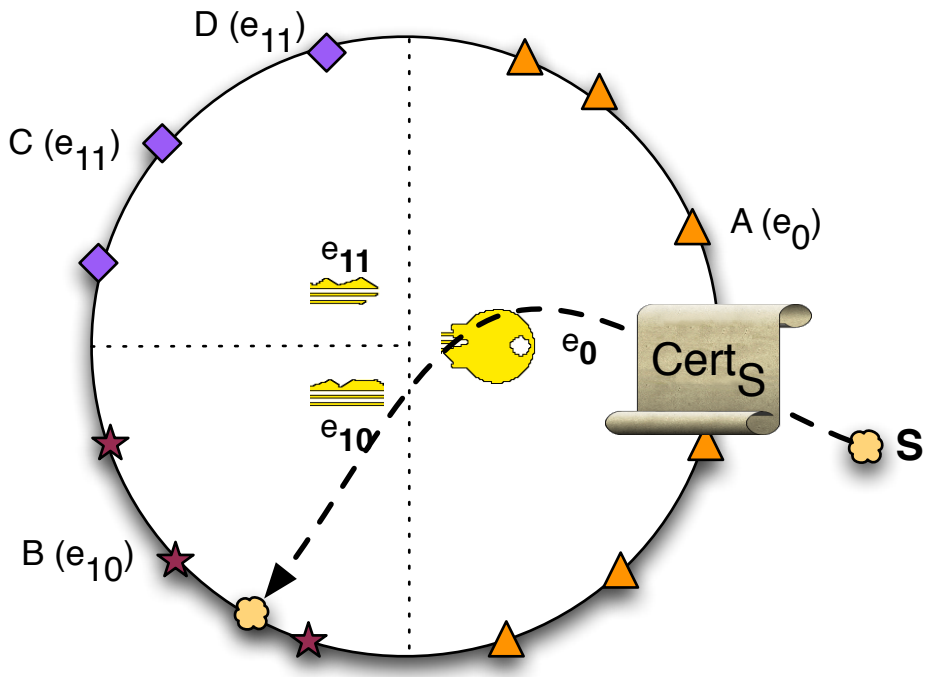
Le pair S demande un certificat

Contrôle d'admission d'un nouveau pair



Le pair S obtient son certificat s'il est honnête

Contrôle d'admission d'un nouveau pair



Le pair S s'insère à l'emplacement indiqué

Réponse à l'attaque sybile

Limitation des identifiants

- Chaque identité doit être préalablement enregistrée dans le réseau SybilGuard
- Un pair n'est accepté que si chaque groupe de fragmentation l'accepte

Aléa des identifiants

- Chaque pair admis obtient un certificat signé
- L'identifiant est dérivé de la signature, non prévisible

Résultats

Nombre de pairs sybils acceptés, 10.000 pairs dans le réseau

	Nb sybils/attaquant
SybilGuard seul	< 144
SybilGuard avec autorité distribuée	< 38

Implémentation

- Partie SybilGuard réalisée par Laurent Bonnet durant son stage (4A/INSA Rennes)
- Couplé au code de l'autorité de certification distribuée

Obtention des clés par identifiant intelligible

Problème de la gestion des clés

- Confidentialité/Intégrité assurées par cryptographie asymétrique
- Comment obtenir la clé publique de l'interlocuteur ?

Procédure d'enregistrement

- Unicité des noms enregistrés
- Premier arrivé, premier servi

Possession d'un nom

- Certificat signé par la clé de réseau
- Certificat stocké en $h(\textit{nom enregistré})$

Garantir l'unicité des noms enregistrés

Pas de signature pour un certificat déjà existant

- Les certificats sont stockés dans la DHT
- L'existence antérieure est vérifiée lors de la signature

Pas d'obtention préalable

- L'obtention d'un certificat laisse une trace dans la DHT
- Cette trace prouve une signature antérieure

Pas d'obtention conflictuelle

- L'obtention d'un certificat doit être planifiée
- Seule cette planification autorise la signature

Application à la voix sur IP pair-à-pair

SIP pair-à-pair avec nommage sécurisé

- Certificats nommés
- Obtention de la clé du correspondant
- Transparent pour le client SIP
- Implémenté et testé avec 3 clients non modifiés

Plan

Étude réalisée

Contexte

- Réseaux pair-à-pair : distribués, sans centre
- Mécanismes de sécurité décentralisés

Objectifs

- Déployable à grande échelle
- Analyses théoriques et expérimentales
- Implémentation et évaluations (PlanetLab)

Autorité de certification distribuée

Service fourni [AIMS 2008]

- Preuve cryptographique de l'accord d'un pourcentage fixé des pairs
- Résistant à un certain nombre d'attaquants internes

Passage à l'échelle [IEEE P2P 2009]

- Opérations de maintenance locales à chaque groupe
- Pas de consensus ni de synchronisation

Applications

Protection contre l'attaque sybile [COPS 2008]

- Couplage avec SybilGuard
- Limitation des identifiants et identifiants aléatoires

Service de nommage sécurisé

- Distribution des clés cryptographiques
- Nommage intelligible et unique

Autres travaux réalisés

Génération distribuée de la clé de réseau [STM 2009]

- Testé avec 37 pairs (37 fragments) sur PlanetLab
- Objet du stage de Thierry Congos (M1/ENS Cachan)

Protection contre l'attaque sybile [ISCC 2008]

- Première proposition
- Principe d'invitations par les membres déjà admis

Détection et exclusion de pairs malveillants [I2CS 2008]

- Détection des pairs malveillants pendant le processus de signature
- Exclusion par révocation du certificat d'accès

Perspectives

Taille des groupes variable

- Réduction du nombre de participants à une signature distribuée
- Impacts sur la sécurité et la maintenance ?

Renouvellement de la clé de réseau

- Durée de vie de la clé de réseau limitée
- Processus de redistribution d'une nouvelle clé

Représentation des pairs honnêtes déconnectés

- Les pairs honnêtes se déconnectent, les attaquants restent en ligne
- Les pairs déconnectés pourraient déléguer leur représentation

Autorité de certification distribuée pour des réseaux pair-à-pair structurés : modèle, mise en œuvre et exemples d'applications

François Lesueur

Soutenance de thèse
27 novembre 2009

Thèse préparée à Supélec dans l'équipe SSIR (EA 4039)

Sous la supervision de Ludovic Mé et Valérie Viet Triem Tong