Annuaire distribué sécurisé pour un réseau de VoIP Peer-to-Peer

François Lesueur, Ludovic Mé, Hervé Debar

Supélec, équipe SSIR (EA4039)
FT R&D, MAPS/NSS

30 novembre 2006

Essor de la Voix sur IP

Intérêts de la VoIP par rapport au RTC

- Faible coût
- Déploiement simple

Problème : Garantir le même niveau de service que le RTC

- Qualité audio
- Sécurité

Sécurité et VoIP

Confidentialité

Pas d'écoute de la communication

Intégrité

Pas de modification des messages

Service d'authentification associé

Bon interlocuteur

Disponibilité

Accessibilité du service

Objectifs

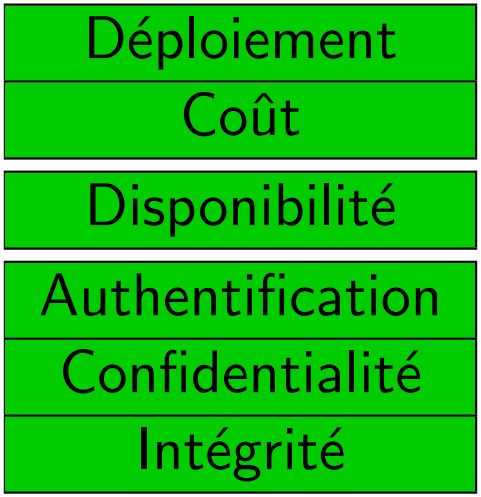
Notre objectif

1. Assurer les 3 propriétés de sécurité : confidentialité, intégrité et authentification
2. Conserver les intérêts de la VoIP : faible coût et déploiement simple
3. Garantir la disponibilité du service

Objectifs

Notre objectif

1. Assurer les 3 propriétés de sécurité : confidentialité, intégrité et authentification
2. Conserver les intérêts de la VoIP : faible coût et déploiement simple
3. Garantir la disponibilité du service



Plan

Partie 1 : Contexte

Les différentes solutions de VoIP actuelles et leurs limites

Partie 2 : Service de nommage certifié

Annuaire distribué avec liaisons certifiées : identifiant $\mapsto$ personne

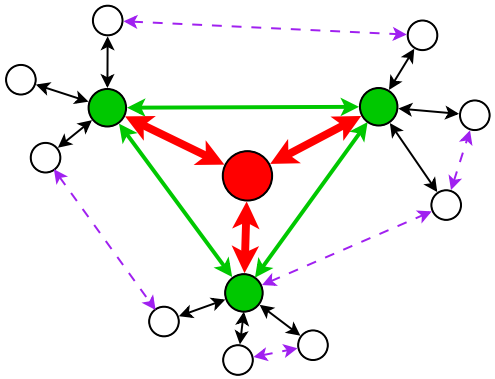
Partie 3 : Identifiants de nœuds uniques

Même pouvoir à chaque participant : personne $\leftrightsquigarrow$ identifiant

Skype

Skype :

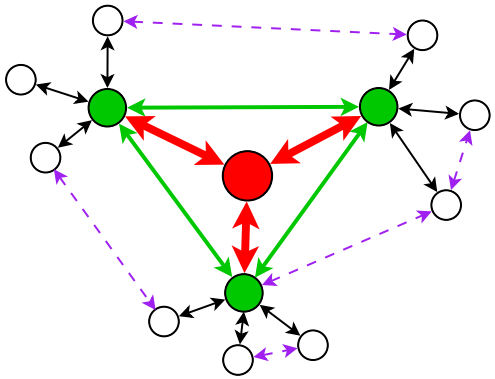
- Peer-to-Peer
- Propriétaire
- Sécurité par offuscation



Skype

Skype :

- Peer-to-Peer
- Propriétaire
- Sécurité par offuscation

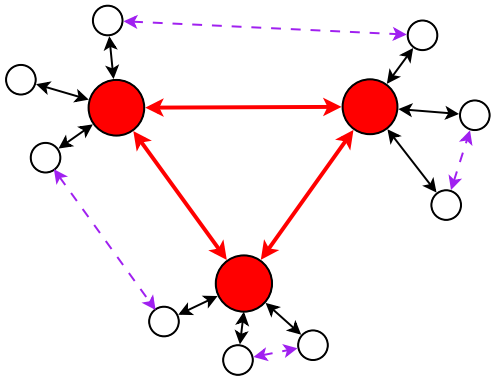


Déploiement
Coût
Disponibilité
Authentification
Confidentialité
Intégrité

SIP (+ Zfone)

SIP :

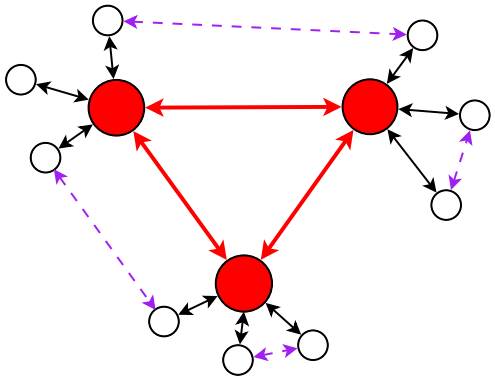
- Client/Serveur *puis* Peer-to-Peer
- Protocole ouvert



SIP (+ Zfone)

SIP :

- Client/Serveur *puis* Peer-to-Peer
- Protocole ouvert

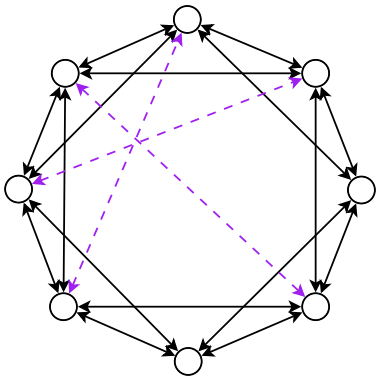


Déploiement
Coût
Disponibilité
Authentification
Confidentialité
Intégrité

P2PSIP (+ Zfone)

P2PSIP :

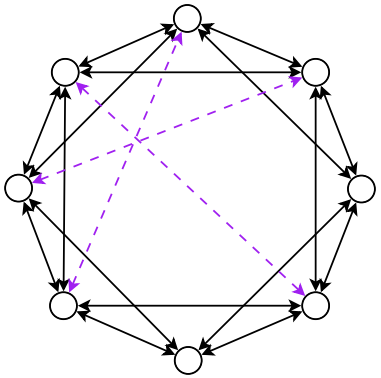
- Peer-to-Peer
- Protocole ouvert



P2PSIP (+ Zfone)

P2PSIP :

- Peer-to-Peer
- Protocole ouvert

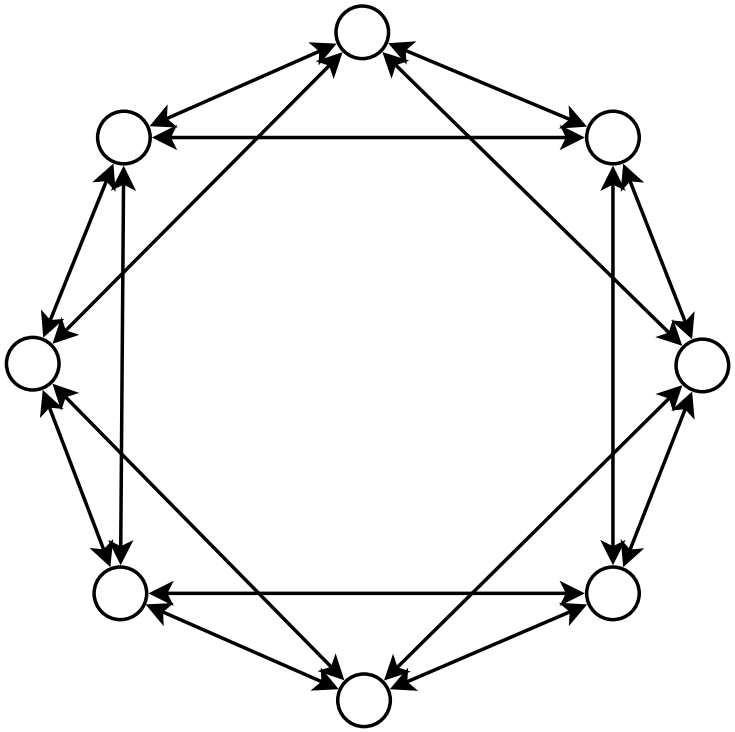


Déploiement
Coût
Disponibilité
Authentification
Confidentialité
Intégrité

Intérêts des réseaux P2P

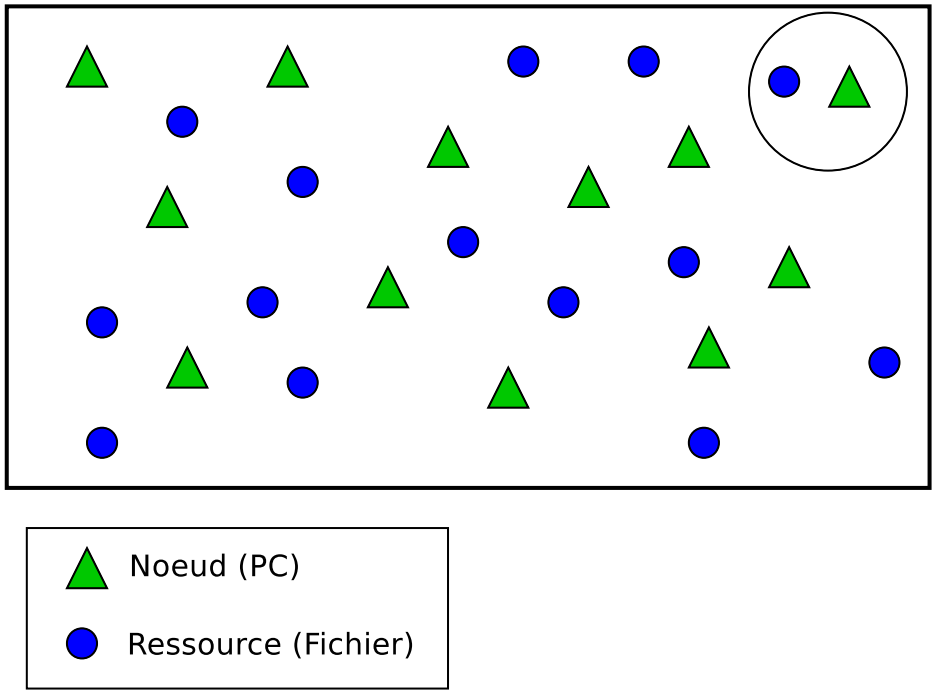
Intérêts des réseaux P2P :

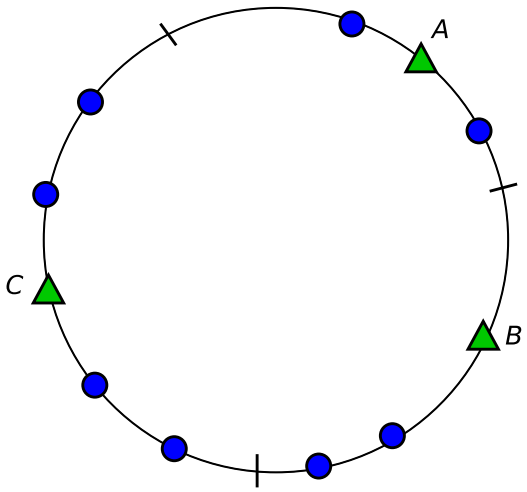
- Forte disponibilité
- Déploiement économique
- Passage à l'échelle
- Équité des pairs
- Absence d'autorité centralisée



Principe des réseaux P2P structurés

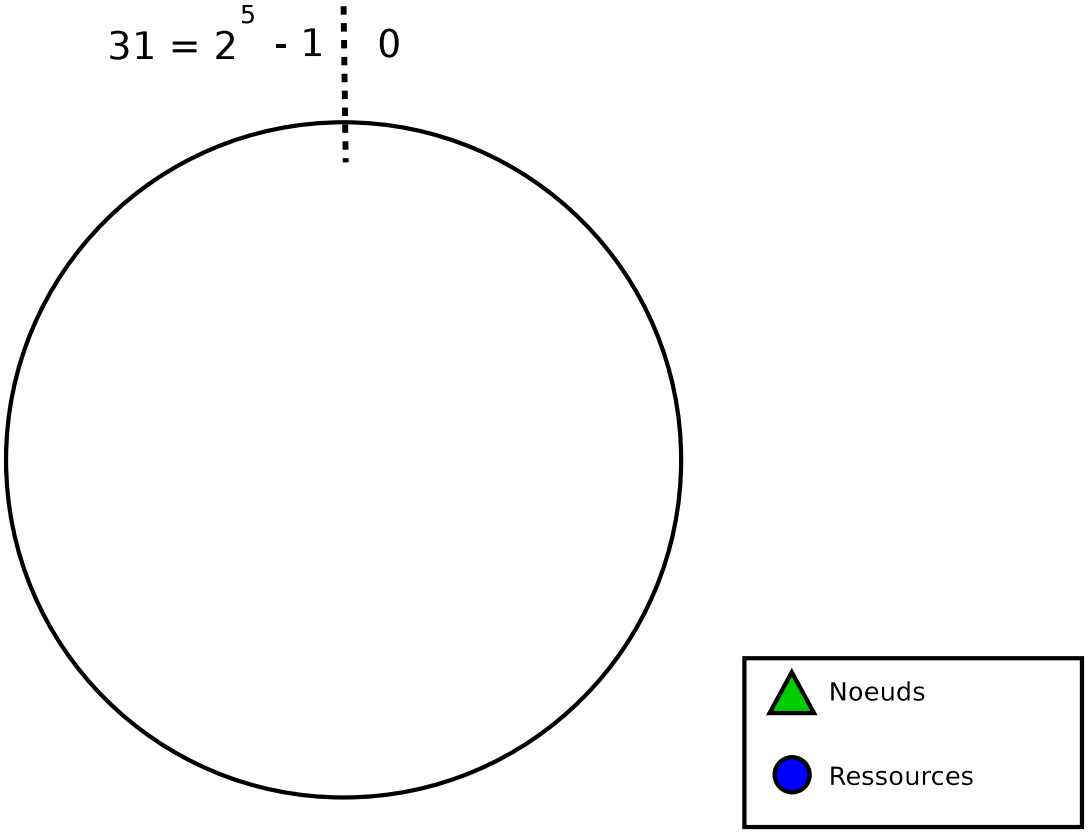
Implémentent une *Distributed Hash Table* (DHT) dans un *overlay*.





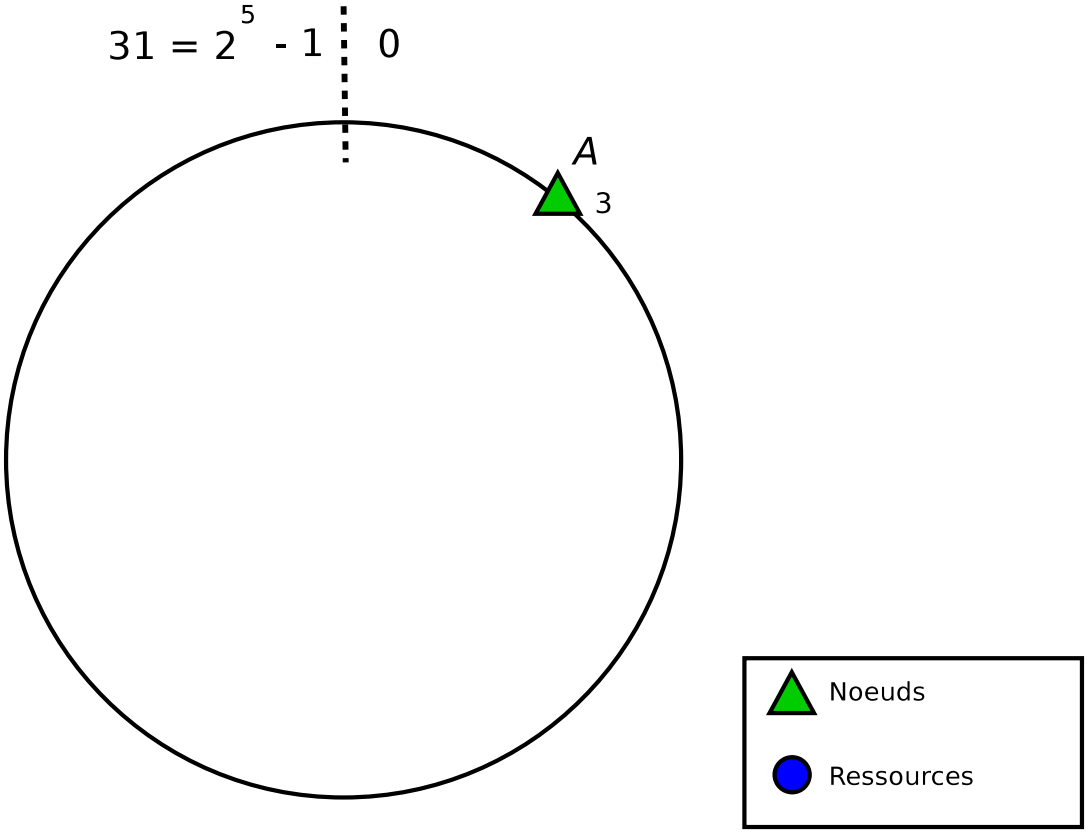
DHT : $key \mapsto value$

Exemple d'overlay



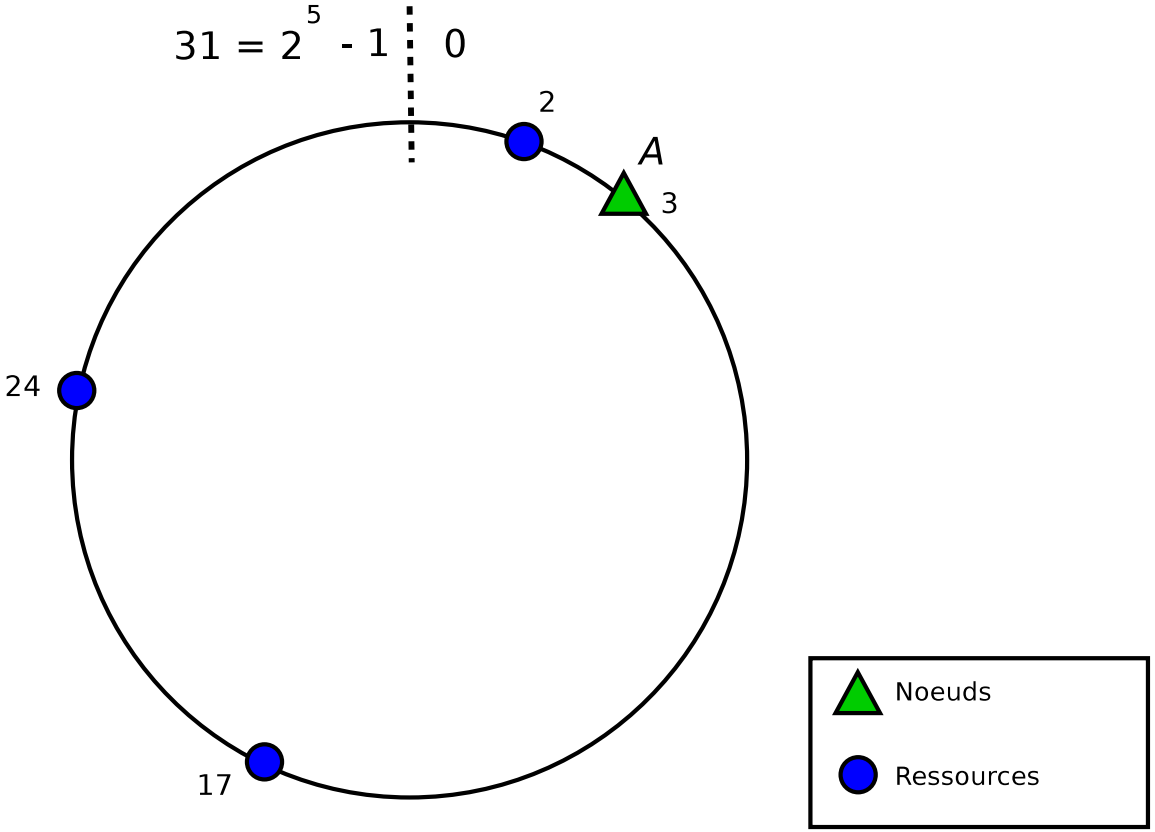
DHT : *key* $\mapsto$ *value*

Exemple d'overlay



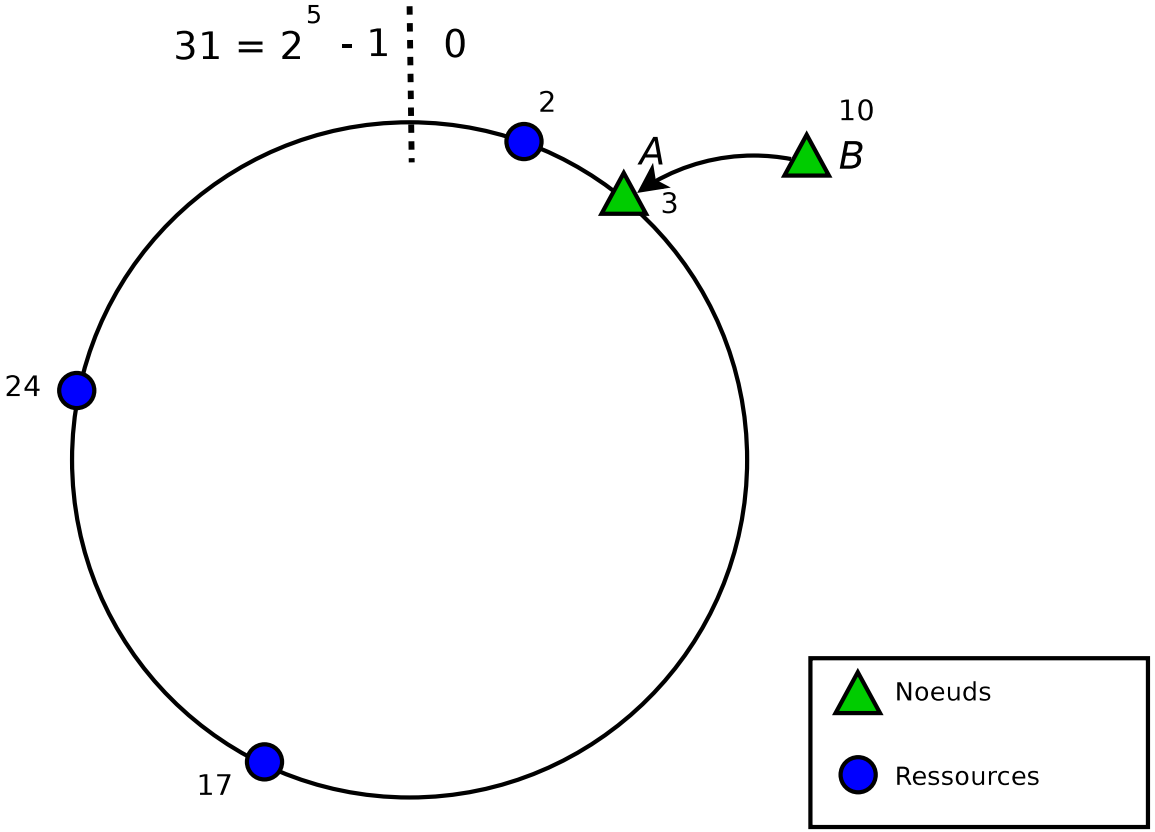
DHT : *key* $\mapsto$ *value*

Exemple d'overlay



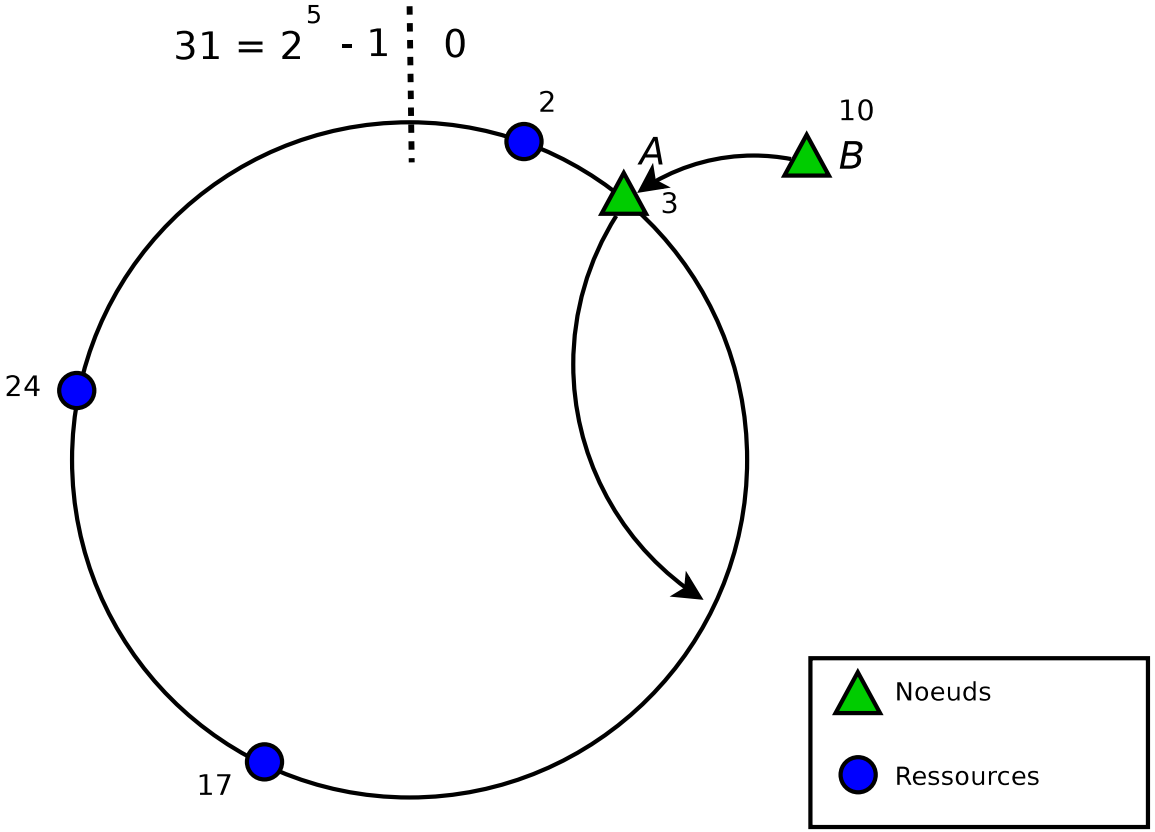
DHT : *key* $\mapsto$ *value*

Exemple d'overlay



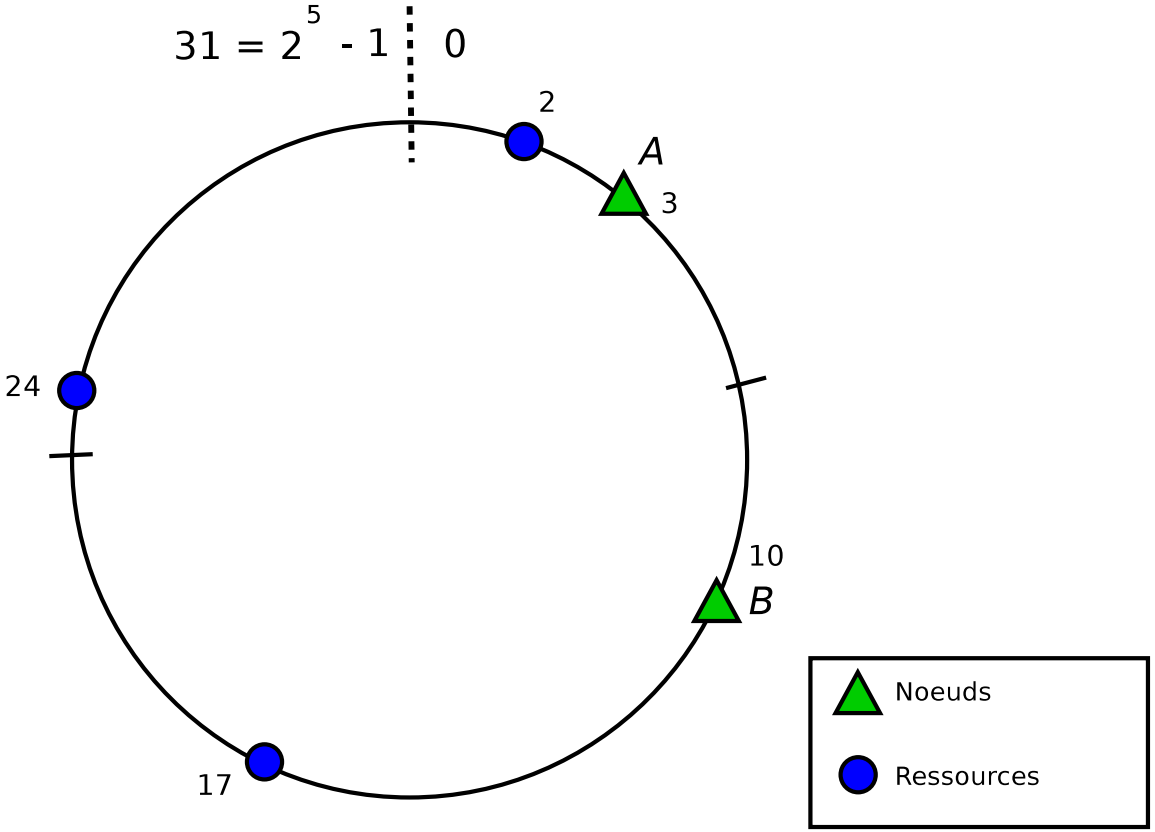
DHT : *key* $\mapsto$ *value*

Exemple d'overlay



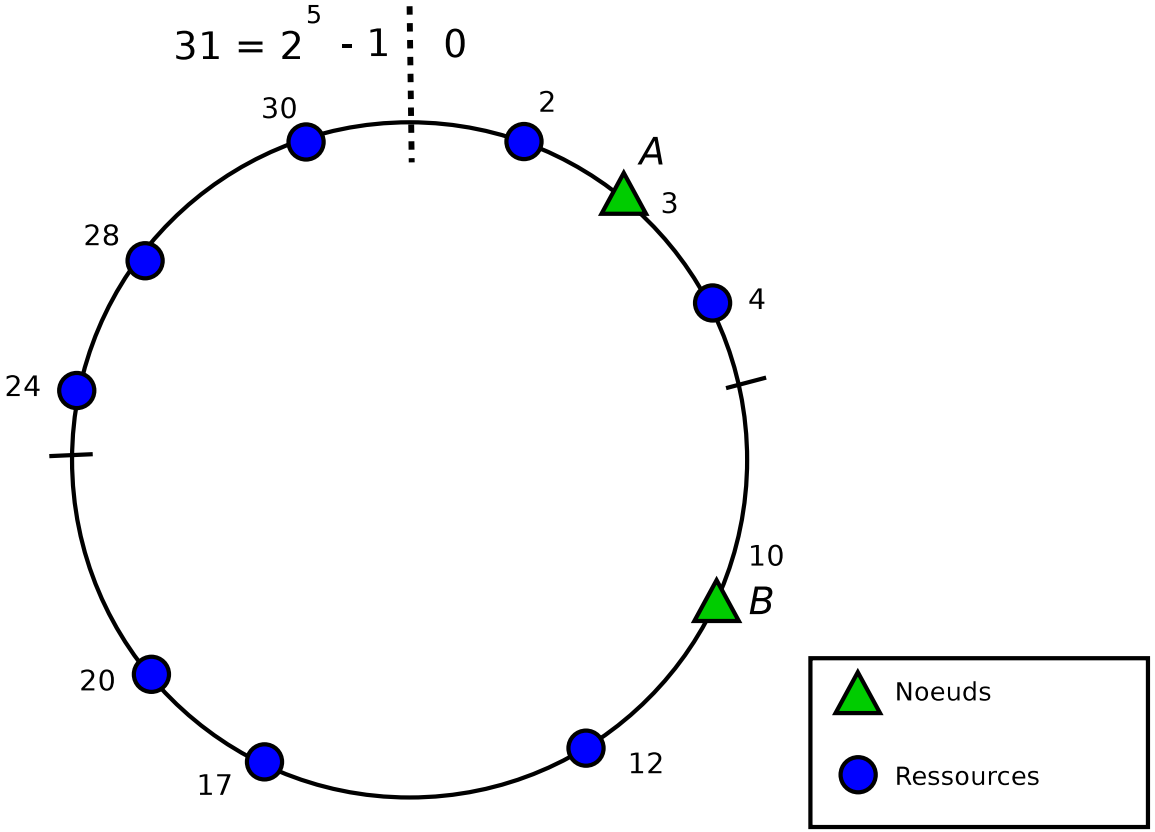
DHT : *key* $\mapsto$ *value*

Exemple d'overlay



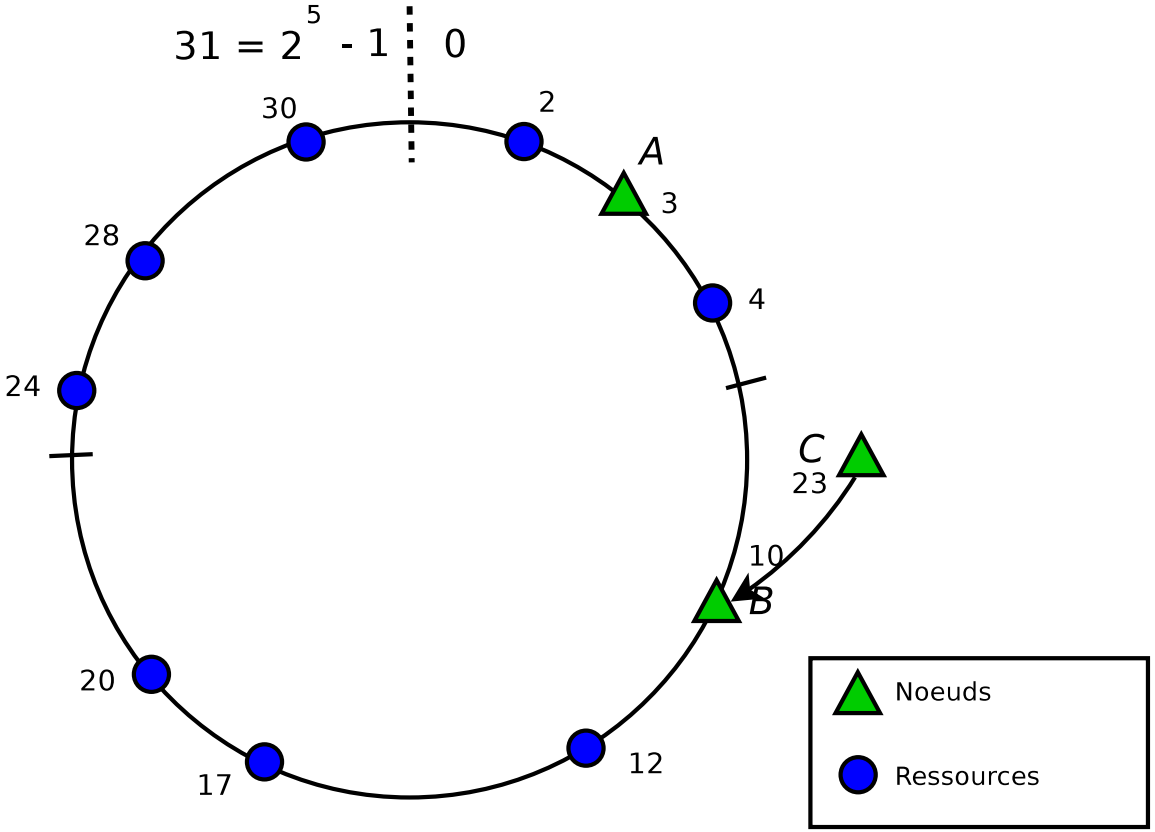
DHT : *key* $\mapsto$ *value*

Exemple d'overlay



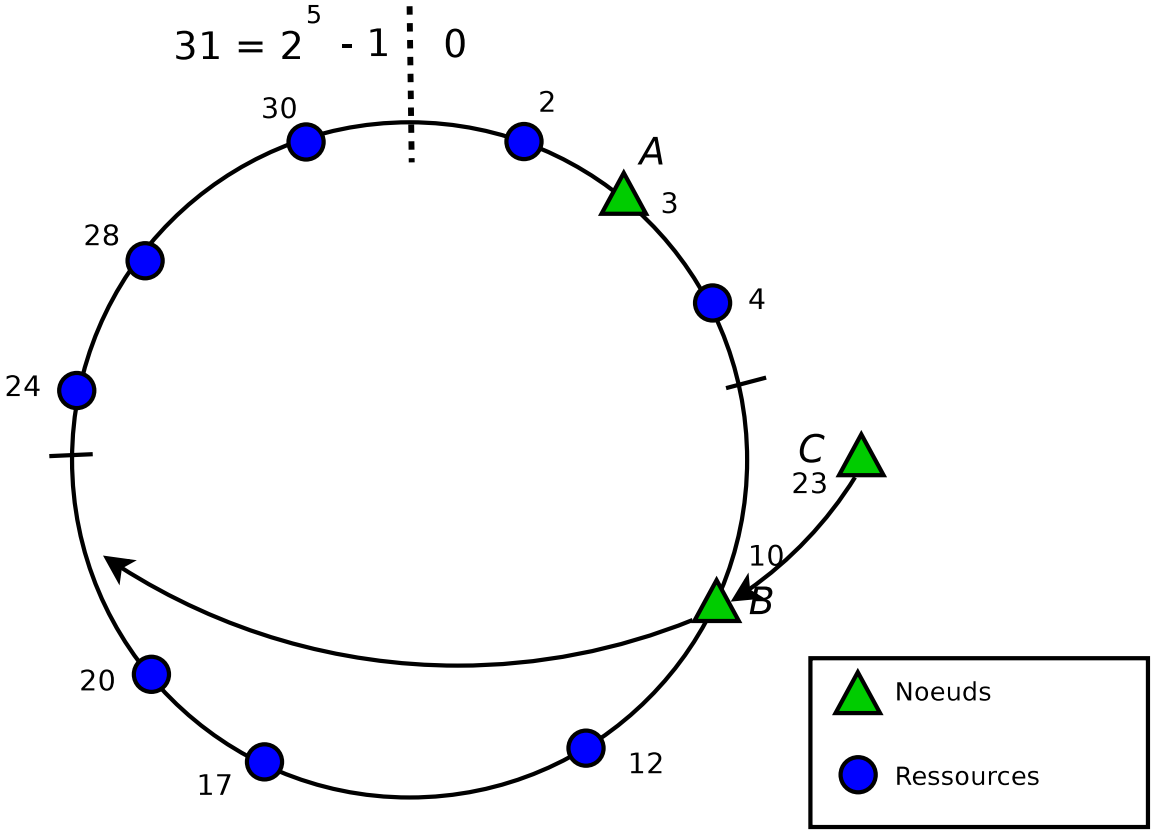
DHT : $key \mapsto value$

Exemple d'overlay



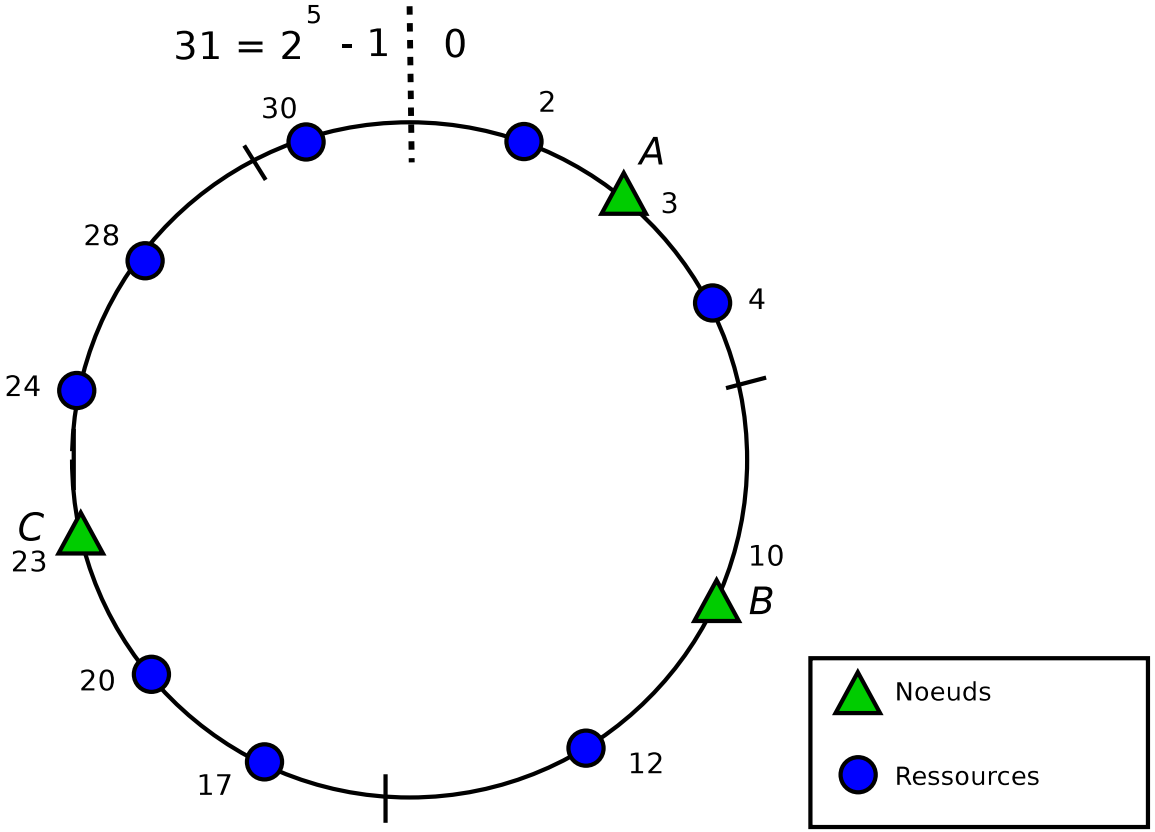
DHT : *key* $\mapsto$ *value*

Exemple d'overlay



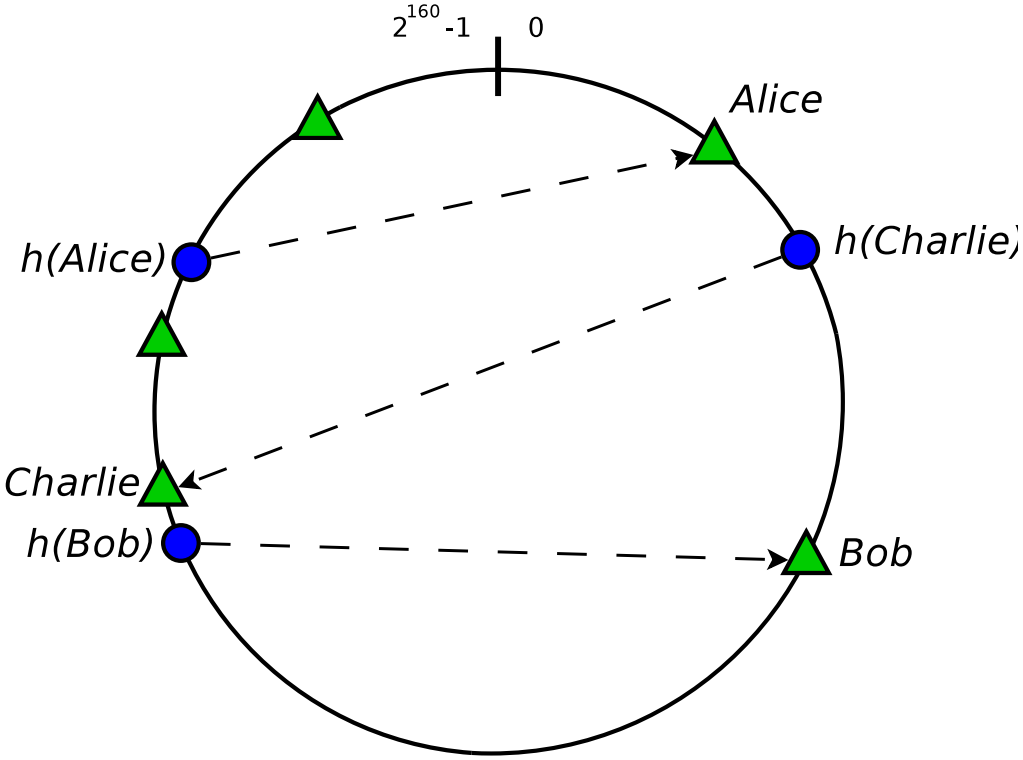
DHT : *key* $\mapsto$ *value*

Exemple d'overlay



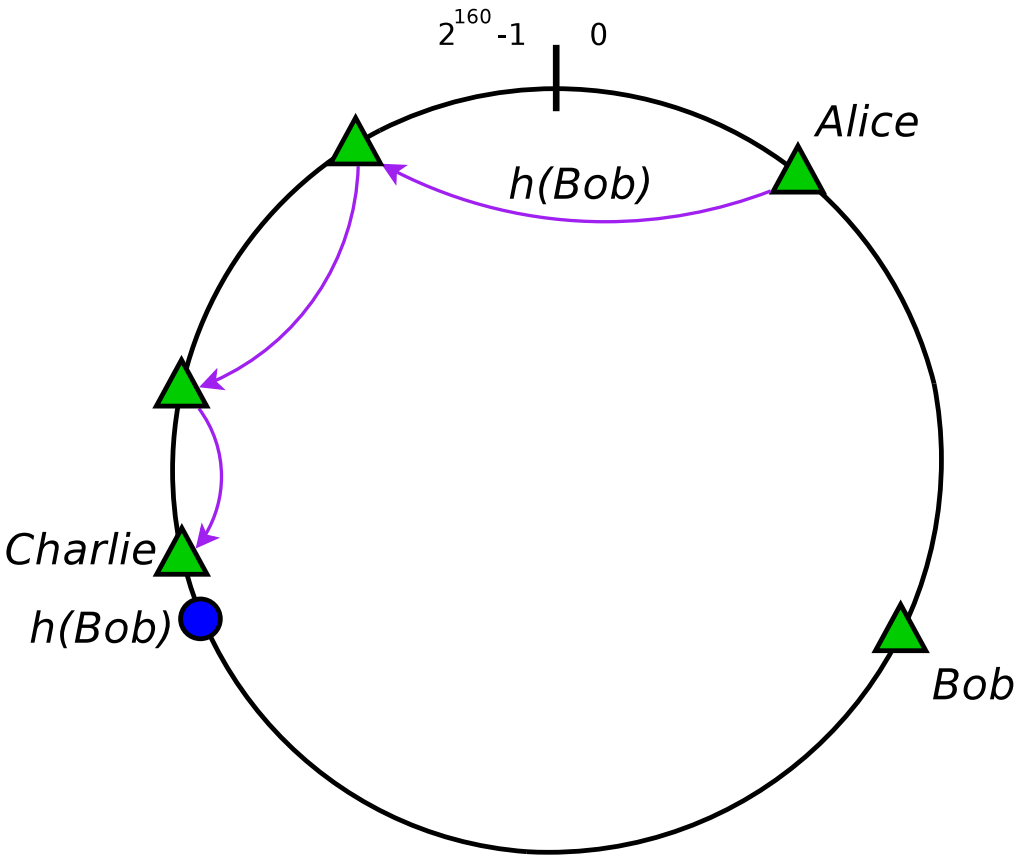
DHT : *key* $\mapsto$ *value*

Principe de P2PSIP



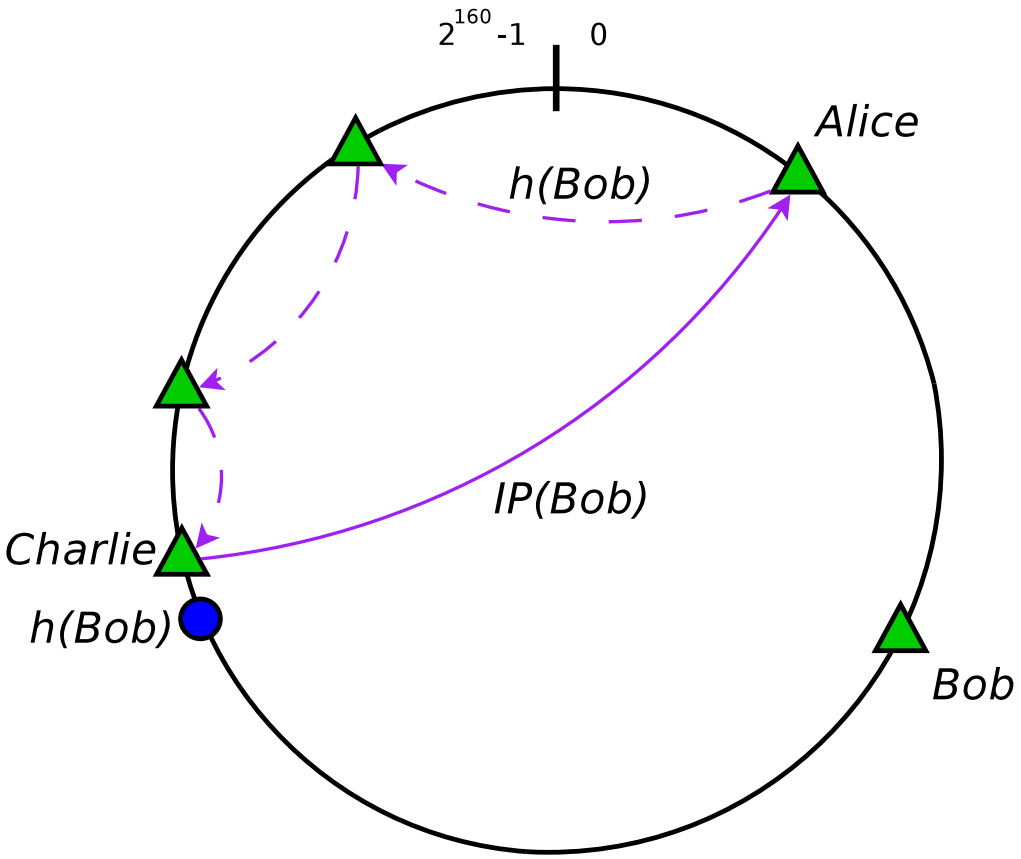
DHT : $h(UserID) \mapsto UserIP$

Recherche d'utilisateur



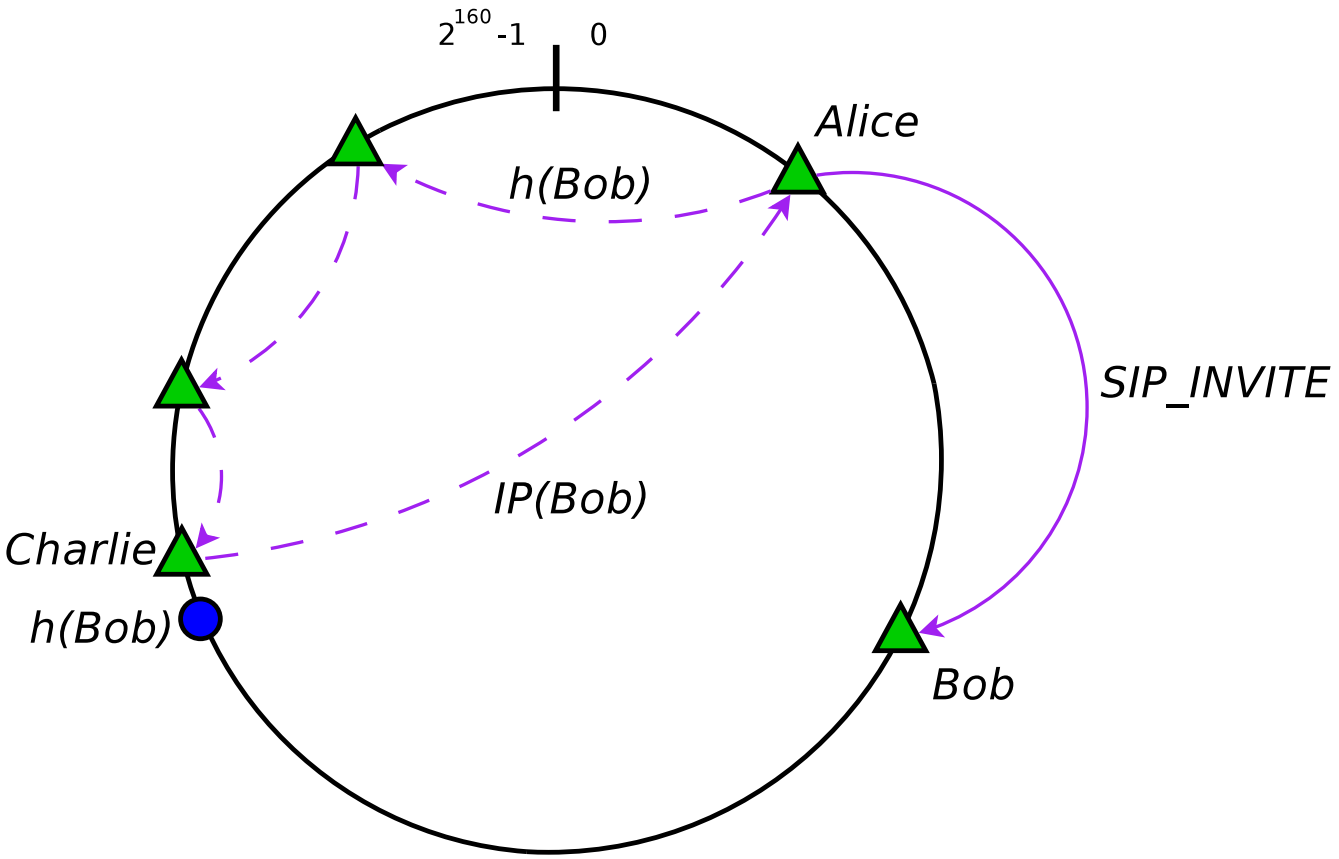
On suppose ici que chaque utilisateur est capable de se trouver un identifiant non utilisé

Recherche d'utilisateur



On suppose ici que chaque utilisateur est capable de se trouver un identifiant non utilisé

Recherche d'utilisateur



On suppose ici que chaque utilisateur est capable de se trouver un identifiant non utilisé

P2PSIP et Sécurité

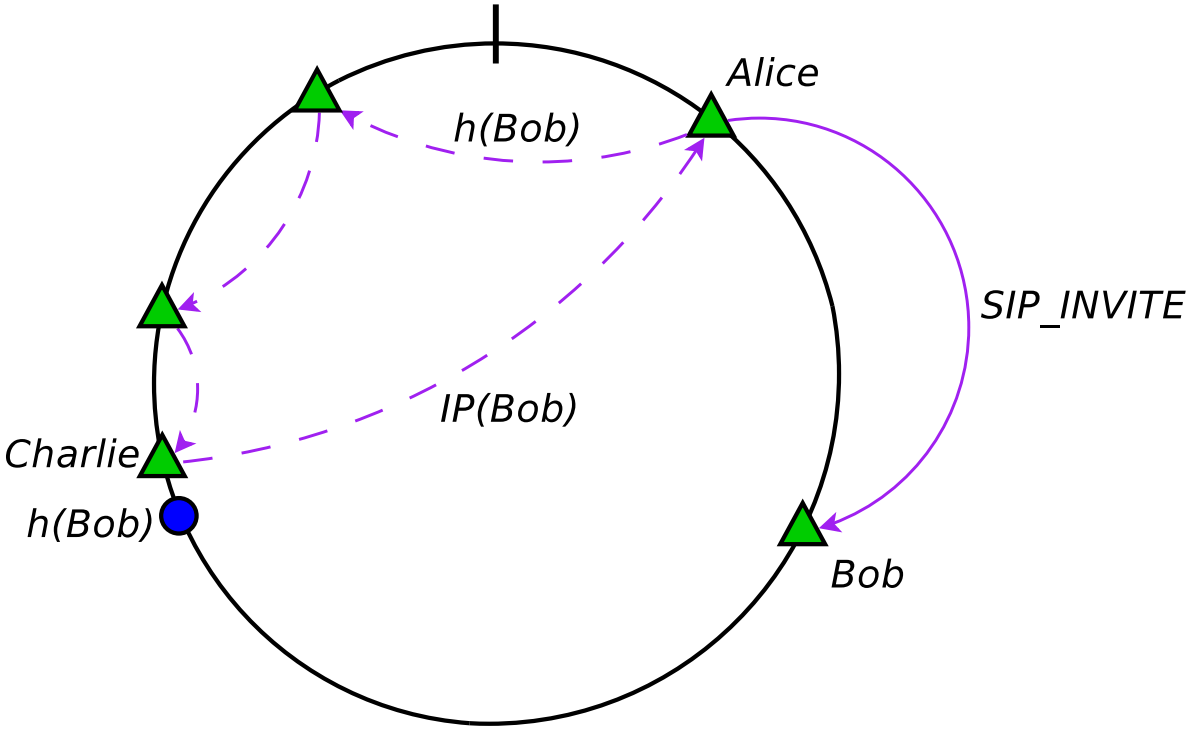
P2PSIP apporte des solutions aux problèmes de SIP (déploiement, disponibilité),
Mais peu de sécurité présente dans P2PSIP

Extrait du Draft Charter IETF (MÀJ 09/2006)

The following topics are *excluded* from the Working Group's scope : . . . fully distributed schemes for assuring unique user identities

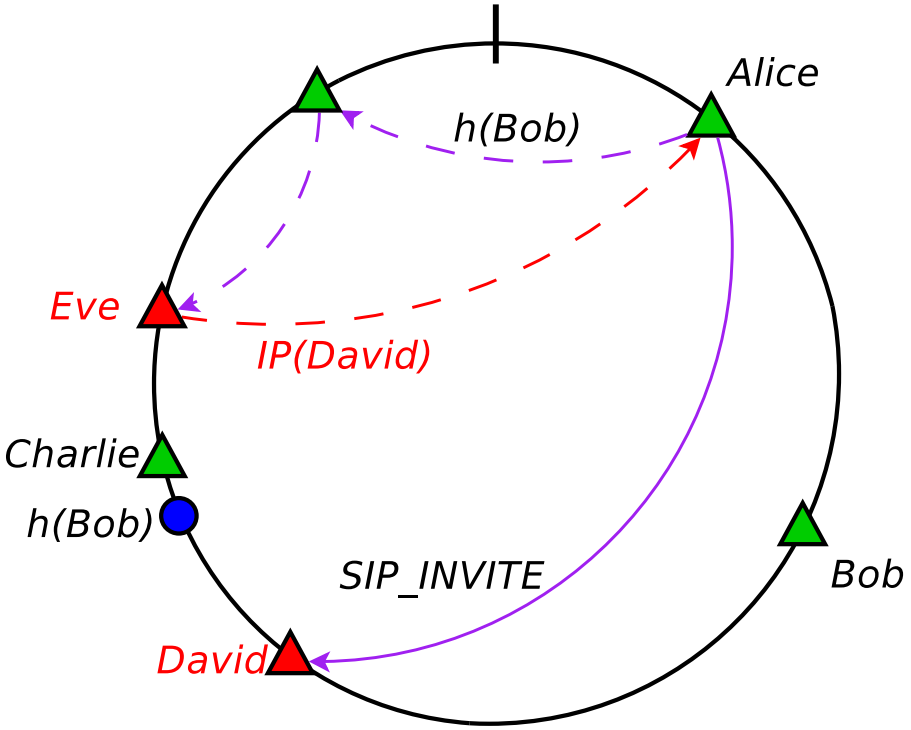
Problème

- Instant t : *Alice* appelle *Bob*, *Bob* répond



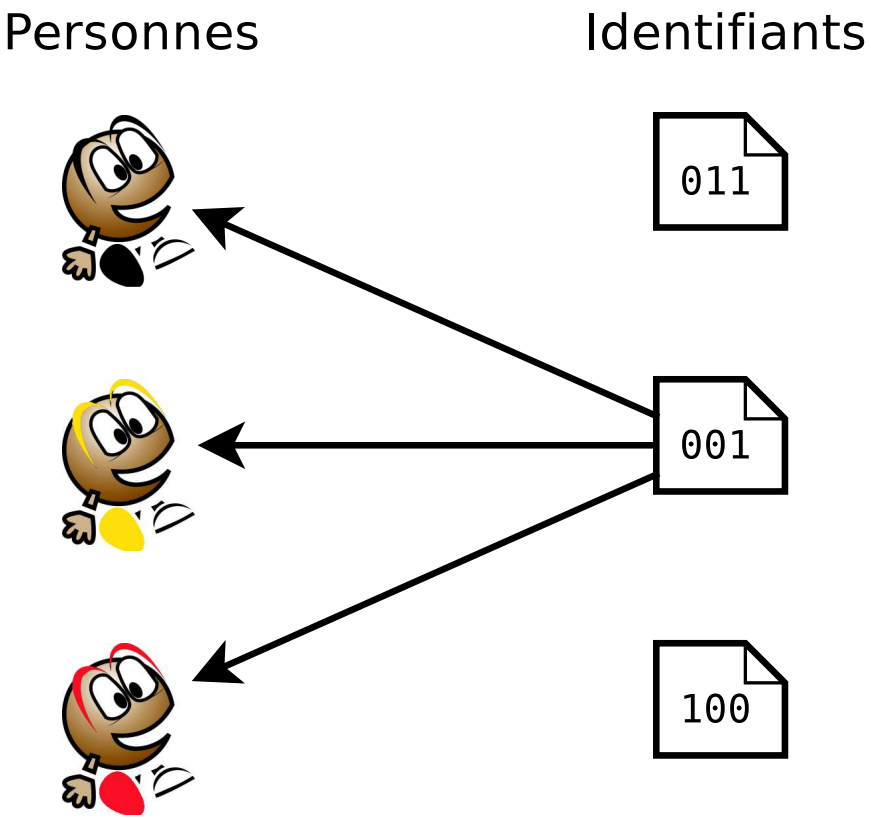
Problème

- Instant t : *Alice* appelle *Bob*, *Bob* répond
- Instant t' : *Alice* appelle *Bob*, *David* malveillant répond



Alice peut stocker une empreinte de *Bob* pour les futures connexions, mais que faire lors de la 1ère connexion ?

Problème



Besoin

Notre objectif : un service de nommage certifié

1. Assurer des identités pérennes et humainement intelligibles
2. De manière entièrement distribuée
3. Selon le principe du *premier arrivé, premier servi*

Besoin

Notre objectif : un service de nommage certifié

1. Assurer des identités pérennes et humainement intelligibles
2. De manière entièrement distribuée
3. Selon le principe du *premier arrivé, premier servi*

Déploiement
Coût
Disponibilité
Authentification
Confidentialité
Intégrité

Pourquoi certifier les *ScreenNames* ?

La certification des *ScreenNames* permet de lier un *ScreenName* à un couple *(clé privée, clé publique)*.

Définition d'un conflit

Un conflit correspond à un même *ScreenName* lié à plusieurs couples *(clé privée, clé publique)*
Lors de la certification :
⇒ Si pas de conflit, certification du *ScreenName*
⇒ Si conflit, pas de certification

Notre proposition

Réaliser cette certification par l'accord de $k\%$ des nœuds du réseau

Fonctionnement

Fonctionnement général de la certification :

- Un nœud A demande le *ScreenName Alice*
- **Pour être certifié, il a besoin de l'accord de $k\%$ des nœuds présents**
- Chacun de ces nœuds vérifie la présence éventuelle d'un conflit
- S'il n'y a pas de conflit, alors A obtient son certificat *Alice* et l'insère dans la DHT
- Plus personne ne pourra faire certifier le *ScreenName Alice*

Cryptographie à seuil *adaptative*

Principe

- Le réseau possède un couple *(clé privée, clé publique)*
- La clé publique est connue de tous
- La clé privée est fragmentée sur l'ensemble des nœuds
- Le chiffrement d'un message avec la clé privée demande la coopération de $k\%$ des nœuds
- **Aucun nœud ne connaît toute la clé privée à un instant**

Fonction RSA homomorphique

Fragmentation de rang 0

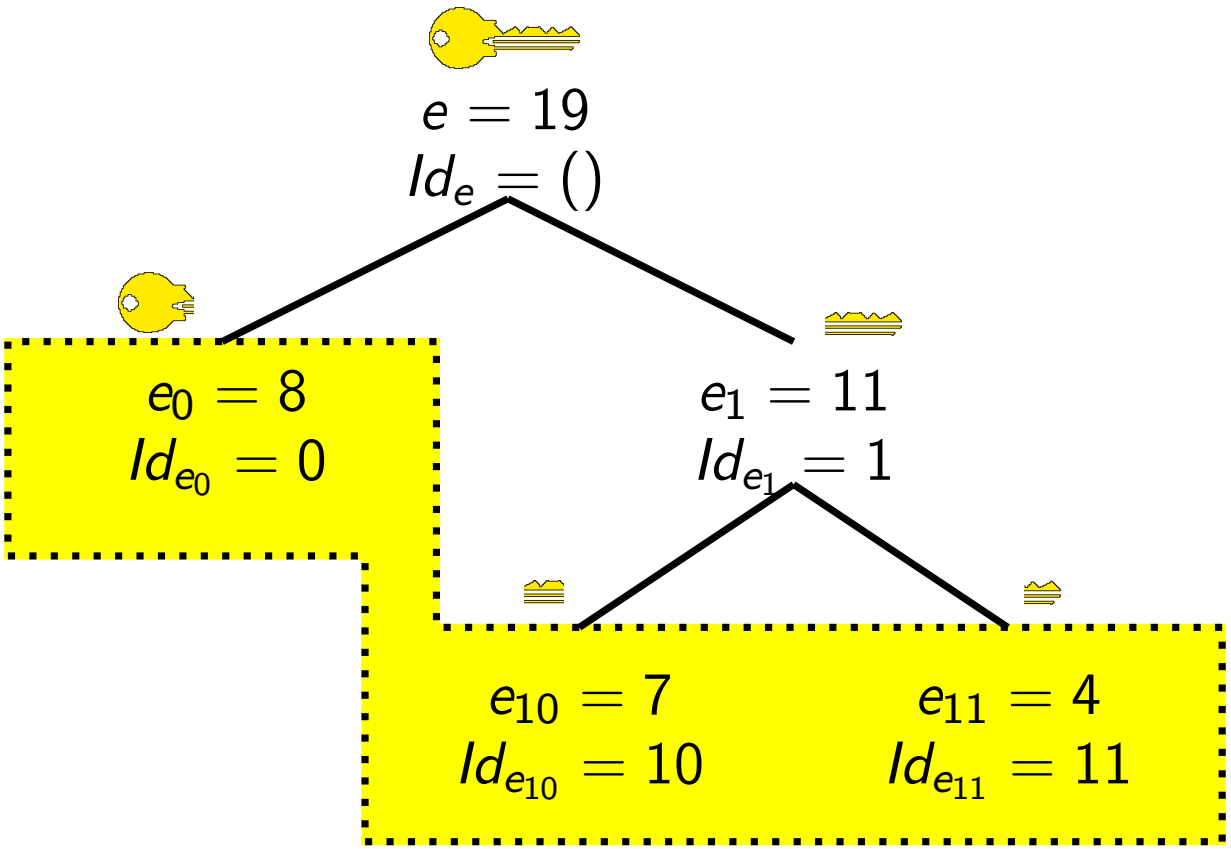
Soit (e, n) la clé secrète du réseau
Soient e_0, e_1 tels que $e = e_0 + e_1$ (+ arithmétique)
Alors $m^e[n] = m^{e_0+e_1}[n] = (m^{e_0}[n] \times m^{e_1}[n])[n]$

Exemple

$(e, n) = (19, 187)$
$e_0 = 8, e_1 = 11$ tels que $19 = 8 + 11$
$m = 18$
Alors $18^{19}[187] = (18^8[187] \times 18^{11}[187])[187] = 52$

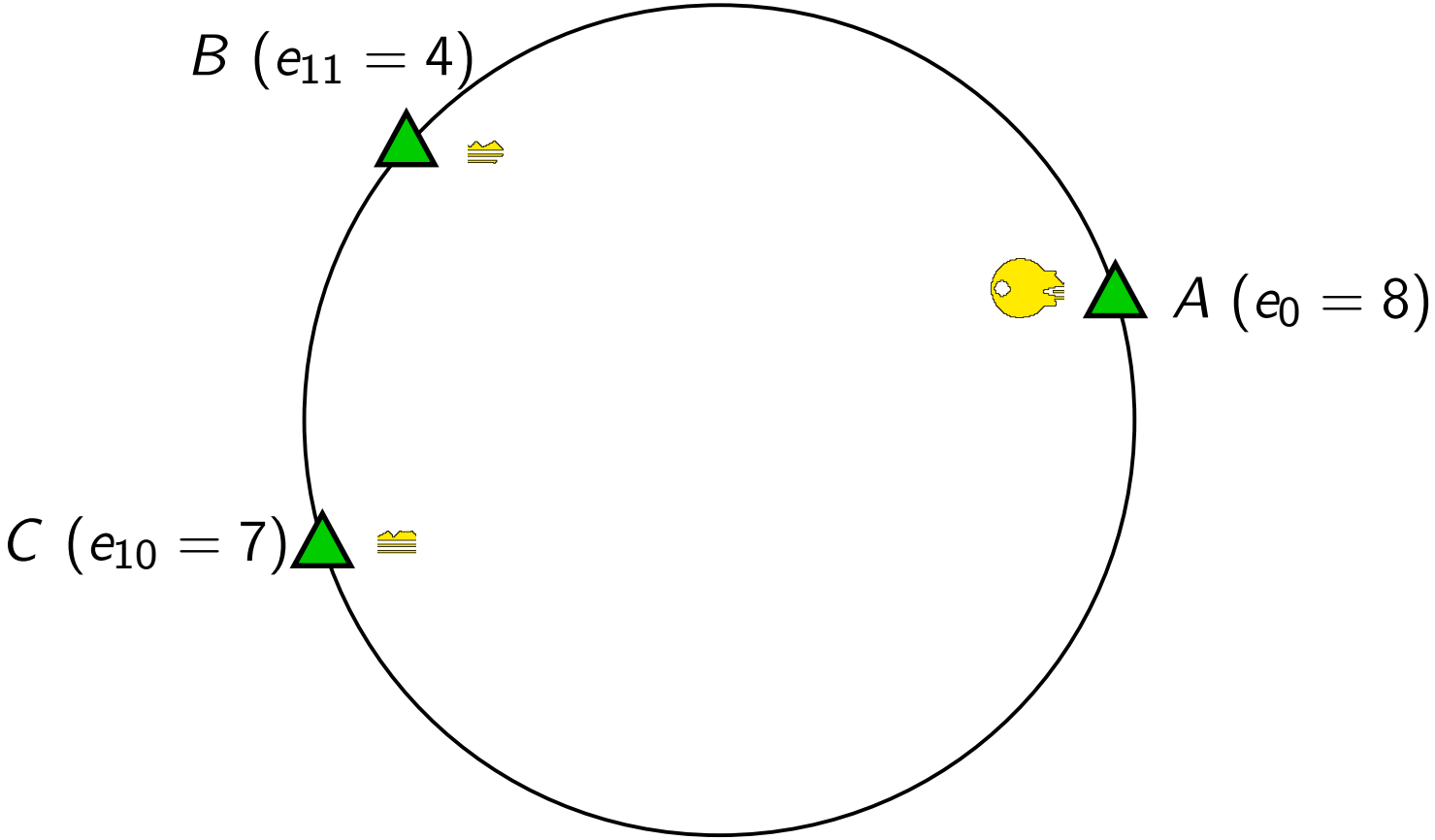
$\Rightarrow$ On répartit les fragments e_i et on itère récursivement au fur et à mesure de la croissance du réseau

Fragmentation de la clé

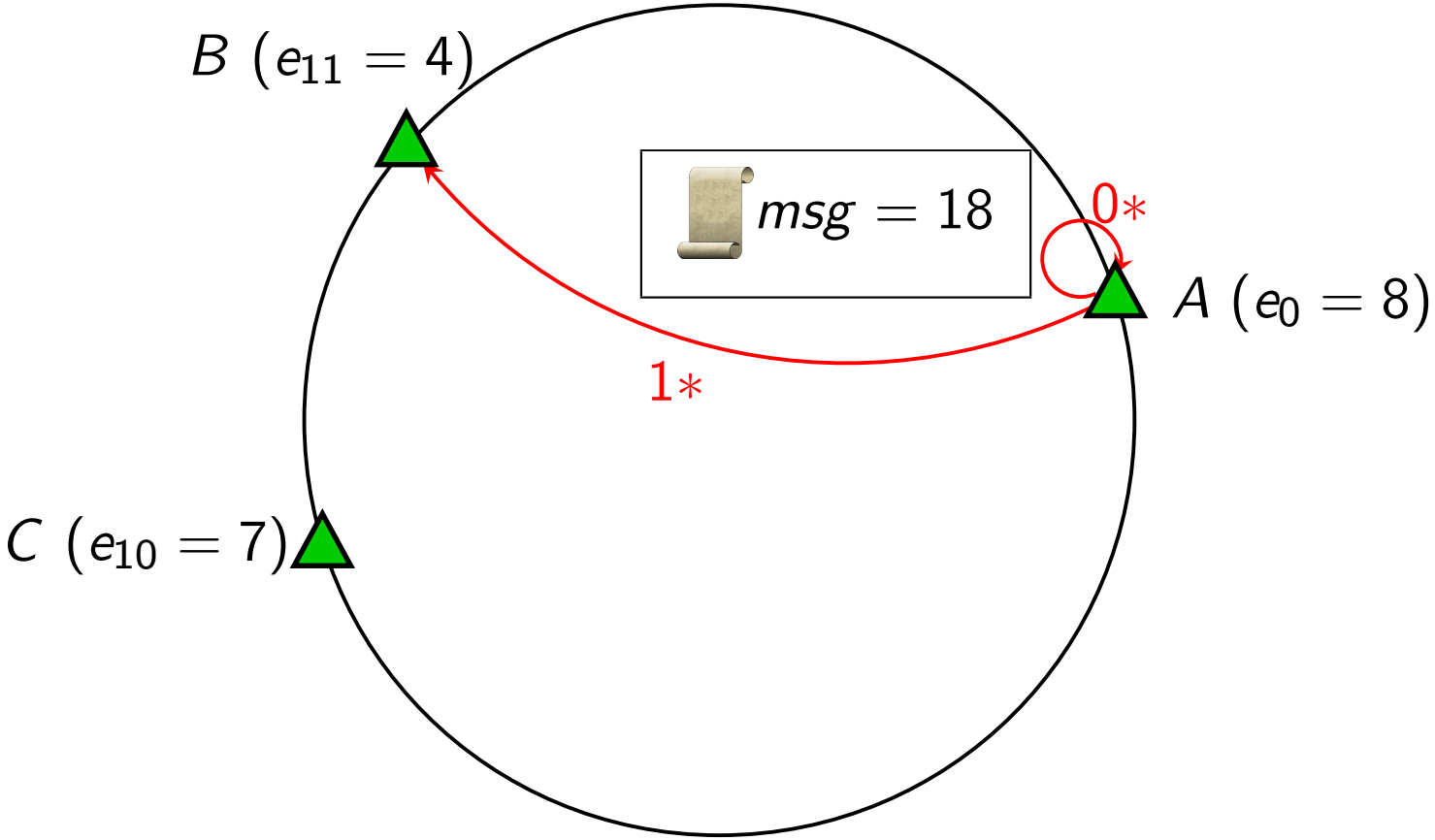


Fragments présents dans le réseau

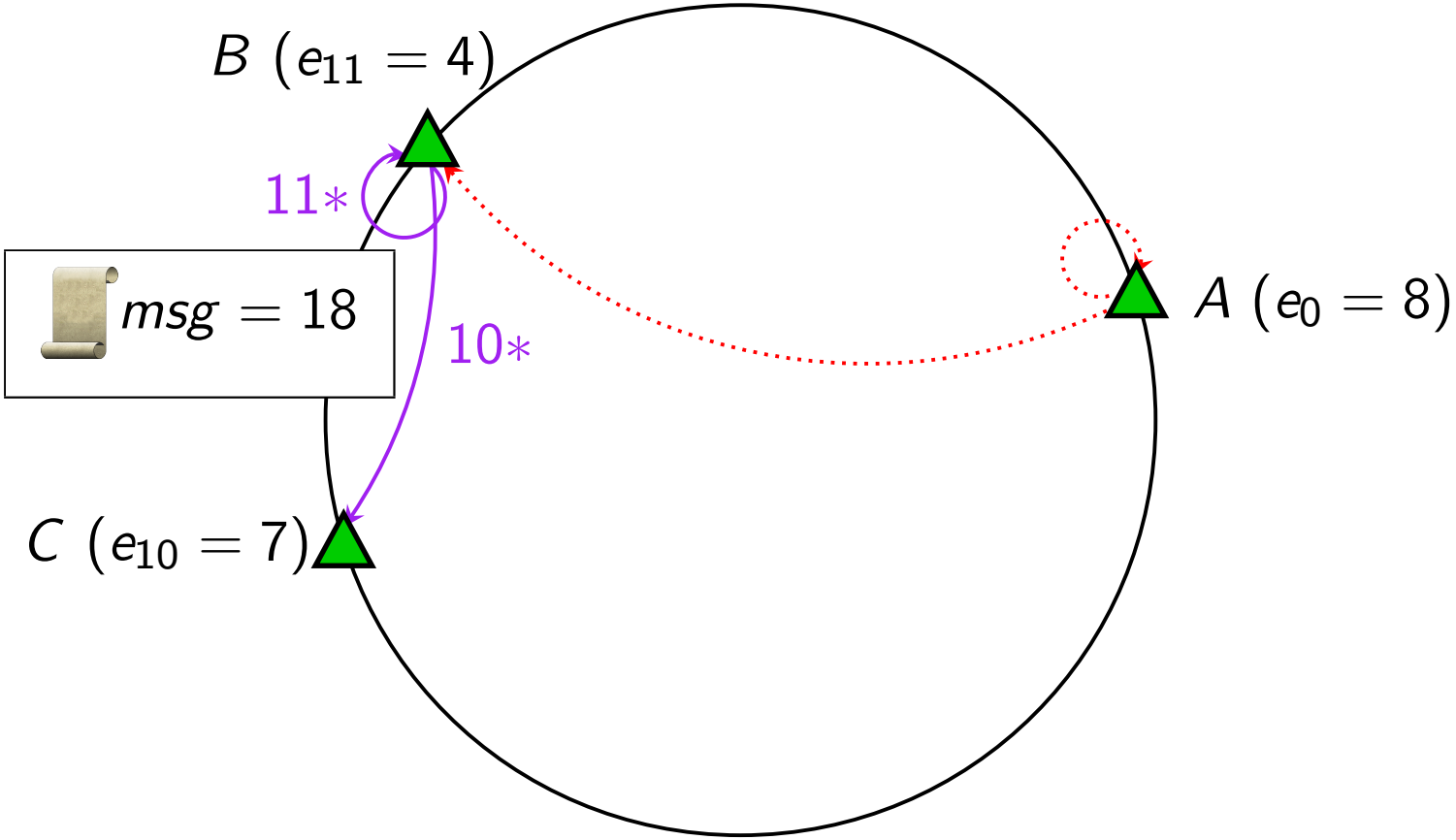
Chiffrement distribué



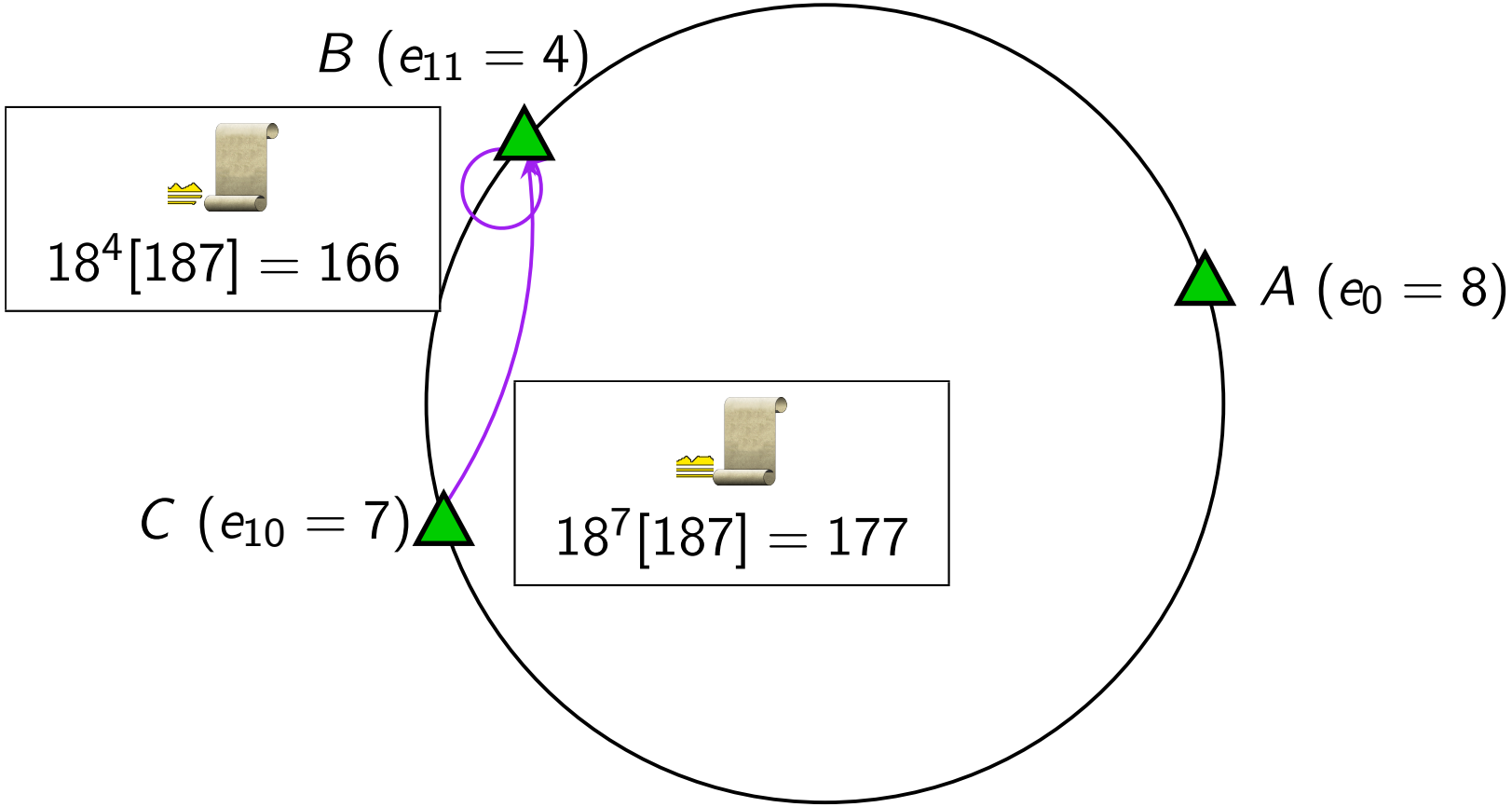
Chiffrement distribué



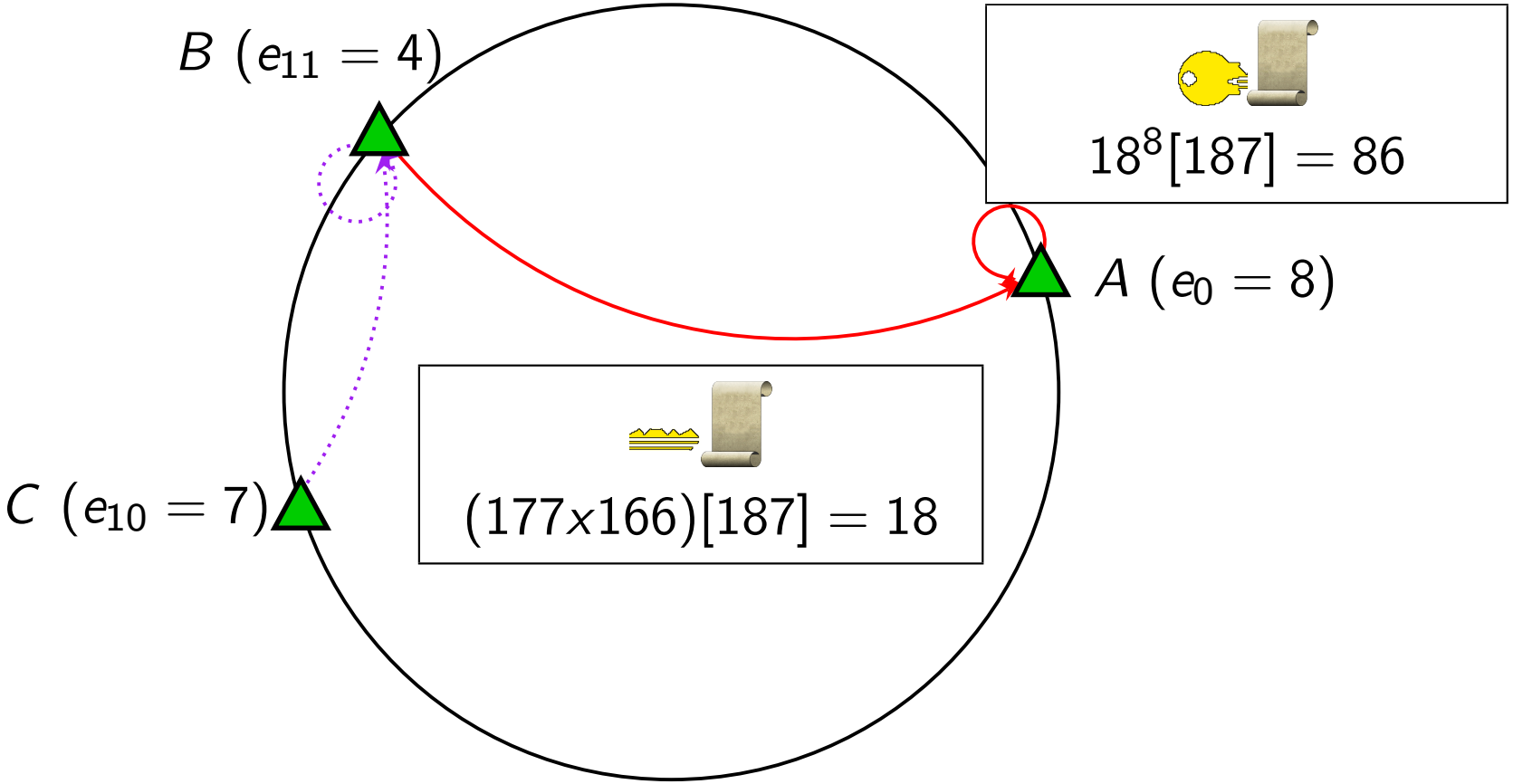
Chiffrement distribué



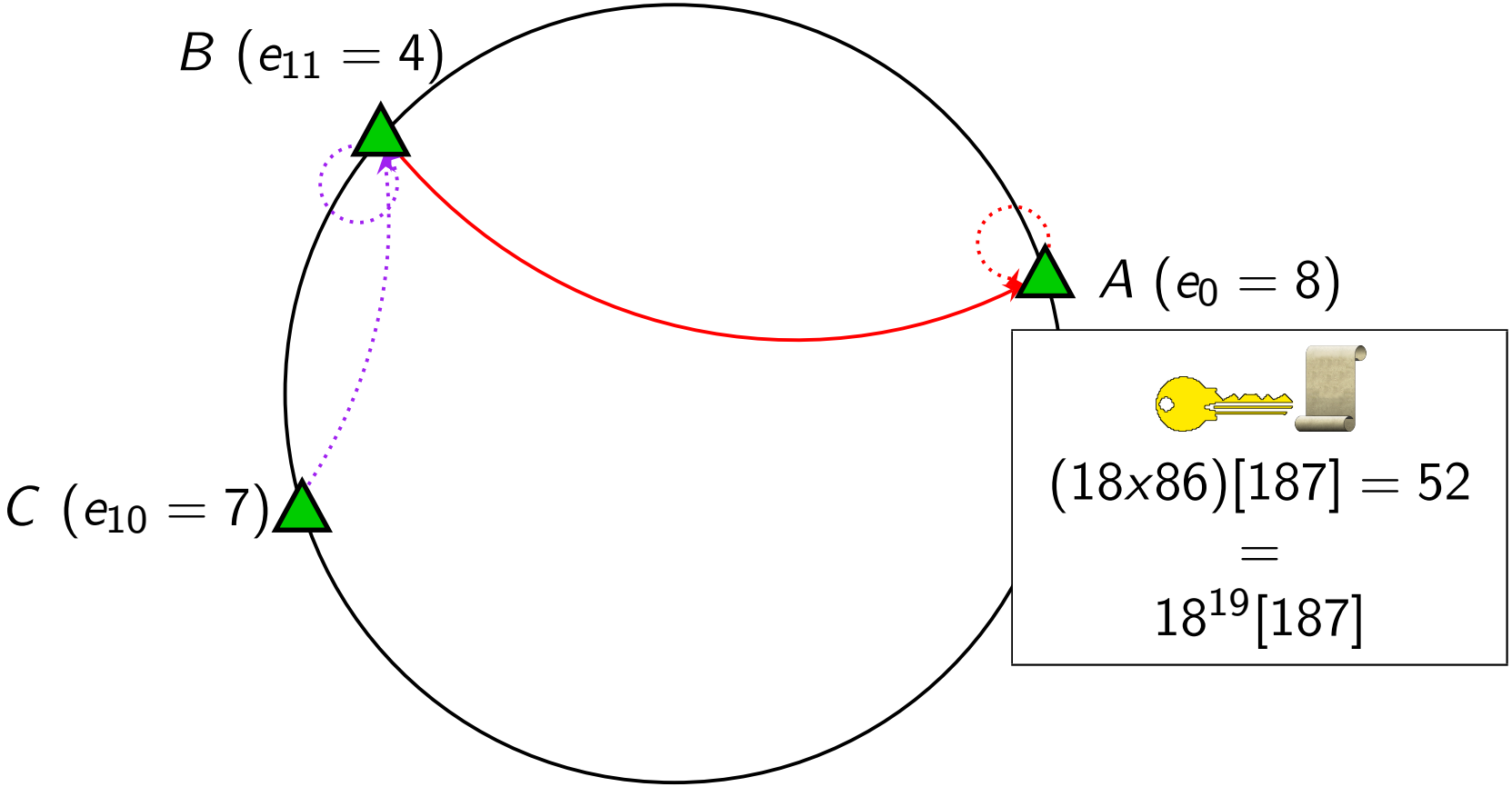
Chiffrement distribué



Chiffrement distribué



Chiffrement distribué



De la théorie à la pratique

Taille des fragments

$taille_{fragments} = taille_{clé} - log_2(NbFragments)$
Pour 1 000 000 de fragments, $taille_{fragments} = taille_{clé} - 20$ (bits)

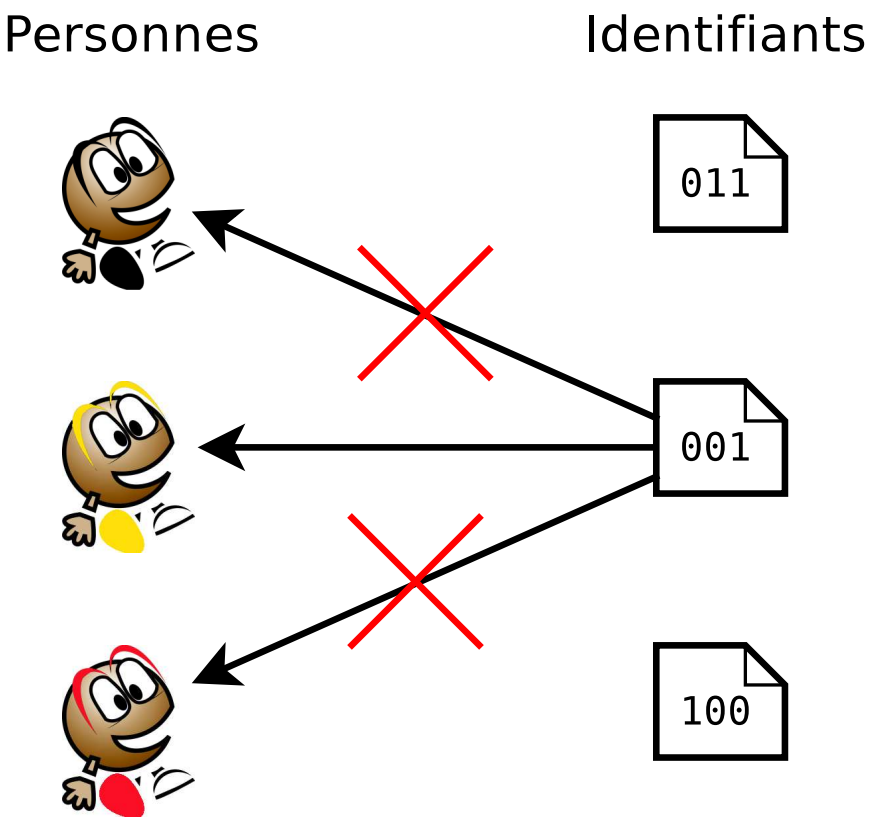
Coût en calcul

$Nb_{exp} = 1$
$NbMax_{mul} = \mathcal{O}(log_2(NbFragments))$
$NbMoy_{mul} = 0.5$
Pour 1 000 000 de fragments, $NbMax_{mul} = 20$

Coût en communication

$NbMax_{msg} = log_2(NbFragments)$
Pour 1 000 000 de nœuds, $NbMax_{msg} = 20$ et $NbMoy_{msg} = 2$

Apport de la certification des *ScreenNames*



Avec certification des *ScreenNames*

Problème

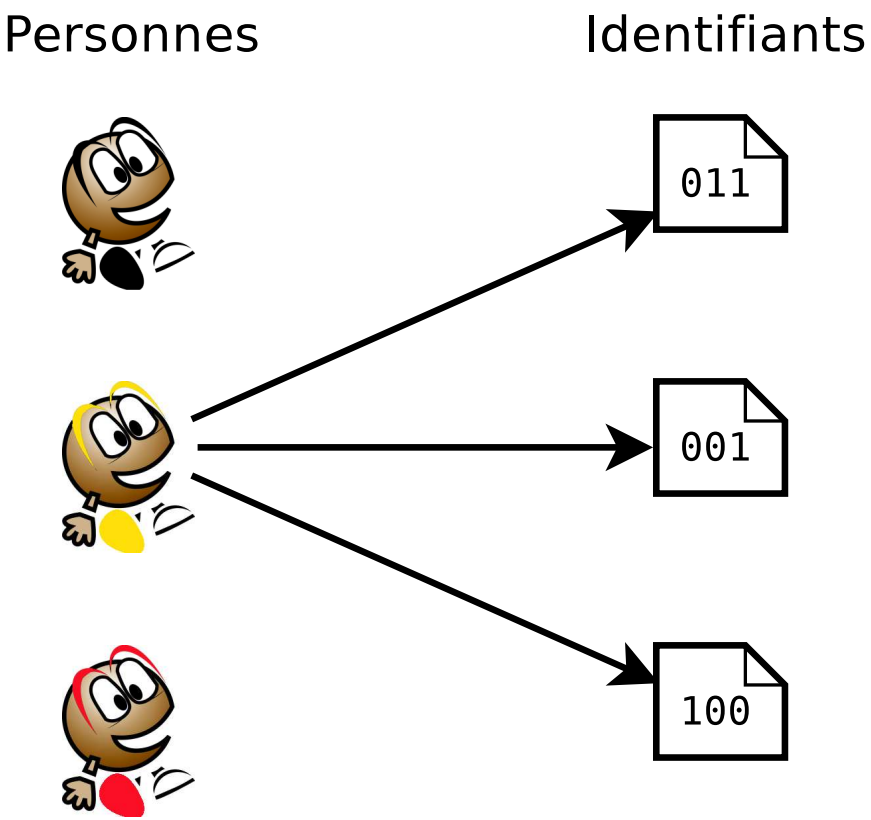
Cas problématique

1. Eve, malveillante, génère N identifiants de nœuds *NodeId*, N arbitrairement grand
2. Eve obtient N fragments de la clé privée du réseau
3. Si N assez grand, elle peut reconstituer toute la clé privée

Objectif

Limiter le nombre de *NodeId* d'une même personne

Problème



Sans limitation des *NodeId*

Identifier quoi ? Comment ?

Différentes approches de l'identification :

- Trusted Computing
- Autorités de certification
- Réseaux de confiance (PGP)

On veut identifier :

- Des machines
- Des nœuds
- Des personnes

Identifier quoi ? Comment ?

Différentes approches de l'identification :

- ~~Trusted Computing~~ ⇒ Problèmes de déploiement
- ~~Autorités de certification~~ ⇒ Pouvoir centralisé, très personnel
- Réseaux de confiance (PGP) ⇒ Chemin de confiance

On veut identifier :

- ~~Des machines~~
- ~~Des nœuds~~
- Des personnes

Identification pyramidale

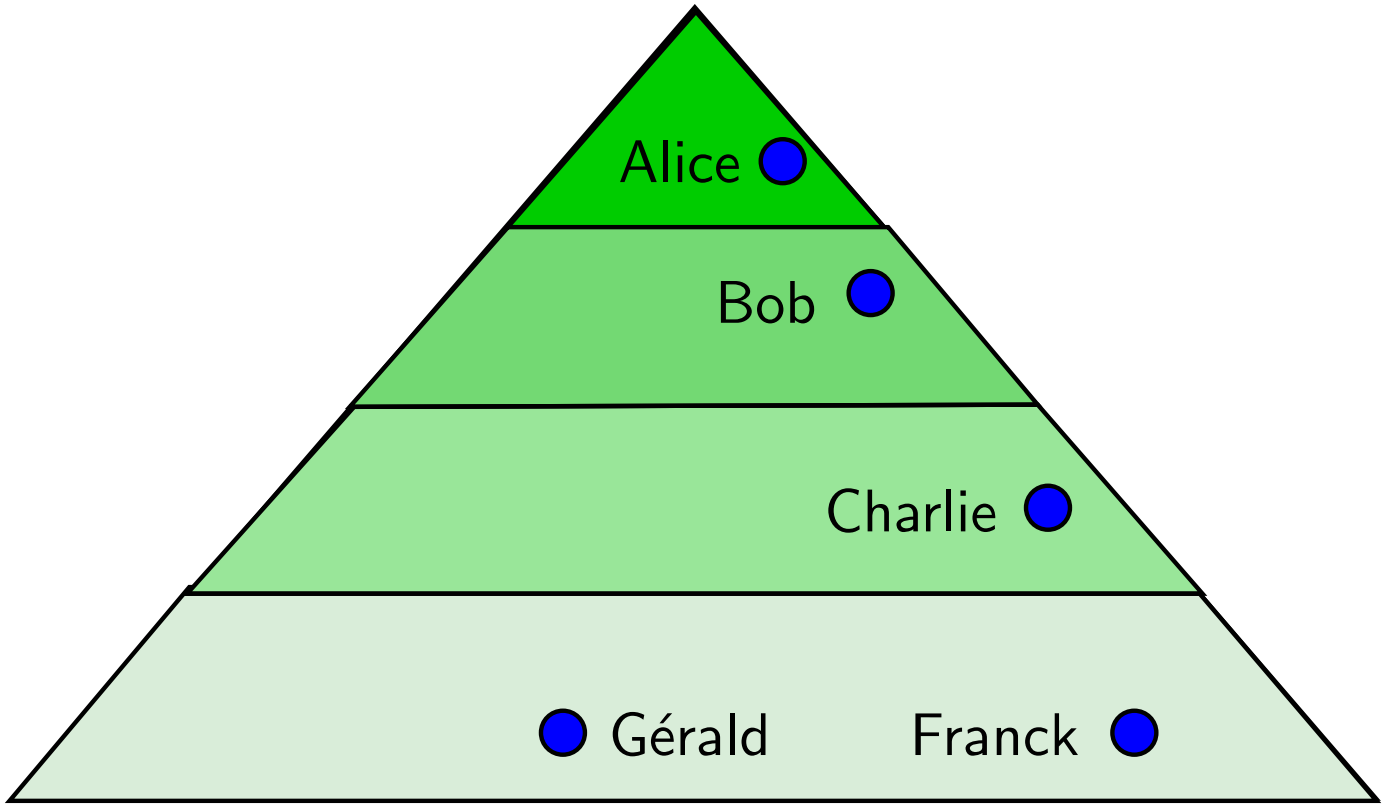
Problèmes des réseaux de confiance

- Complexité pour créer un chemin de confiance
- Actuellement, utilisation d'un dépôt centralisé

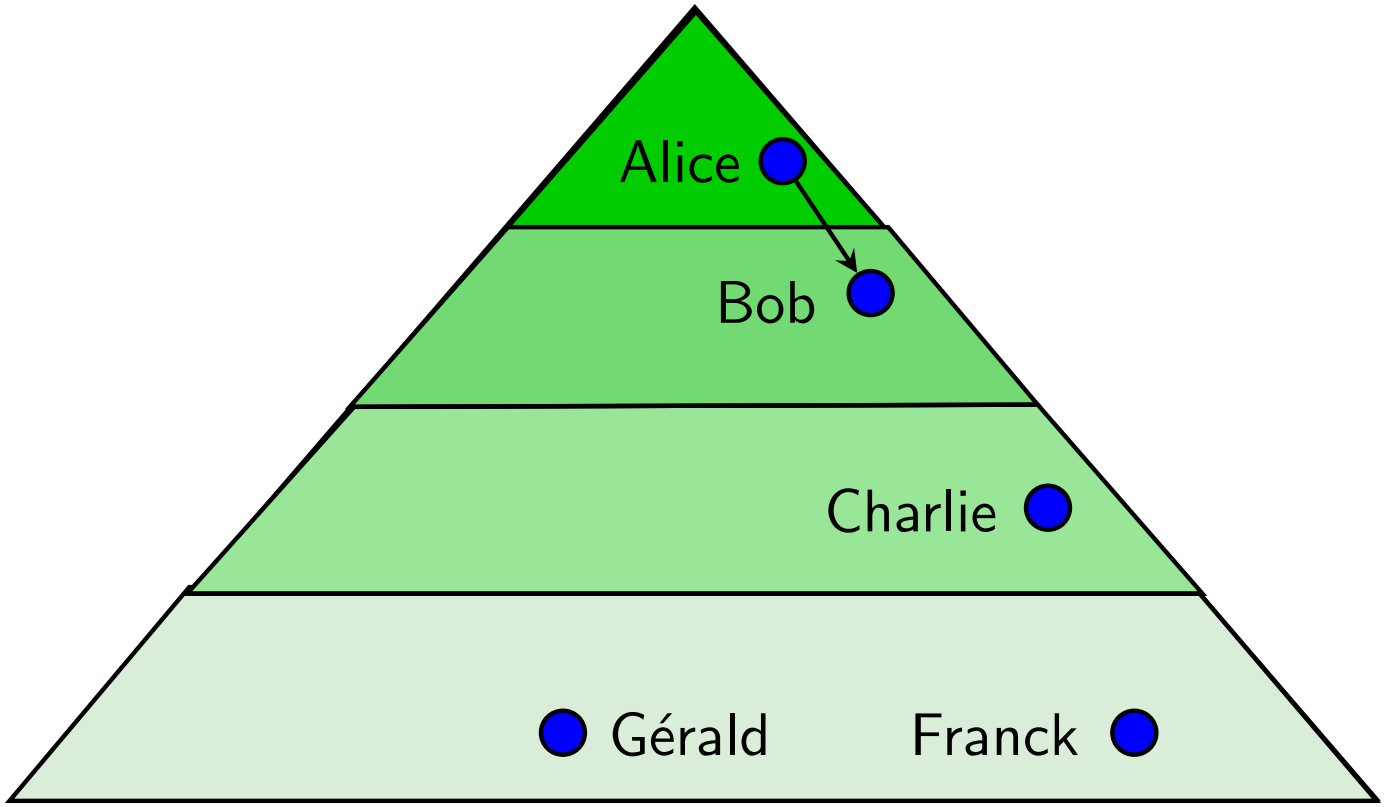
Notre proposition

- Identification pyramidale
- Pouvoir réparti en haut
- Chaînes de confiance simples
- Besoin d'être coopté par un parrain

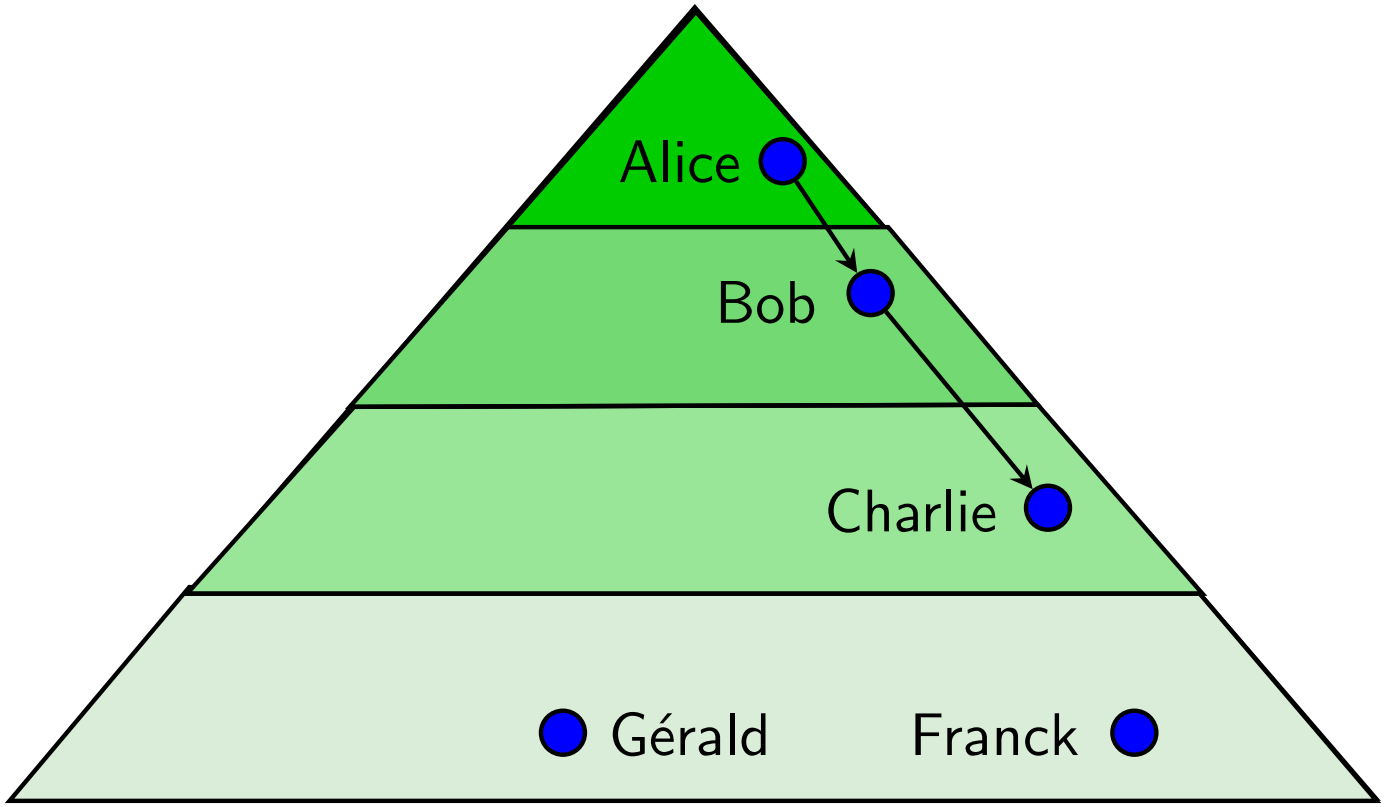
Exemple de chaîne de confiance



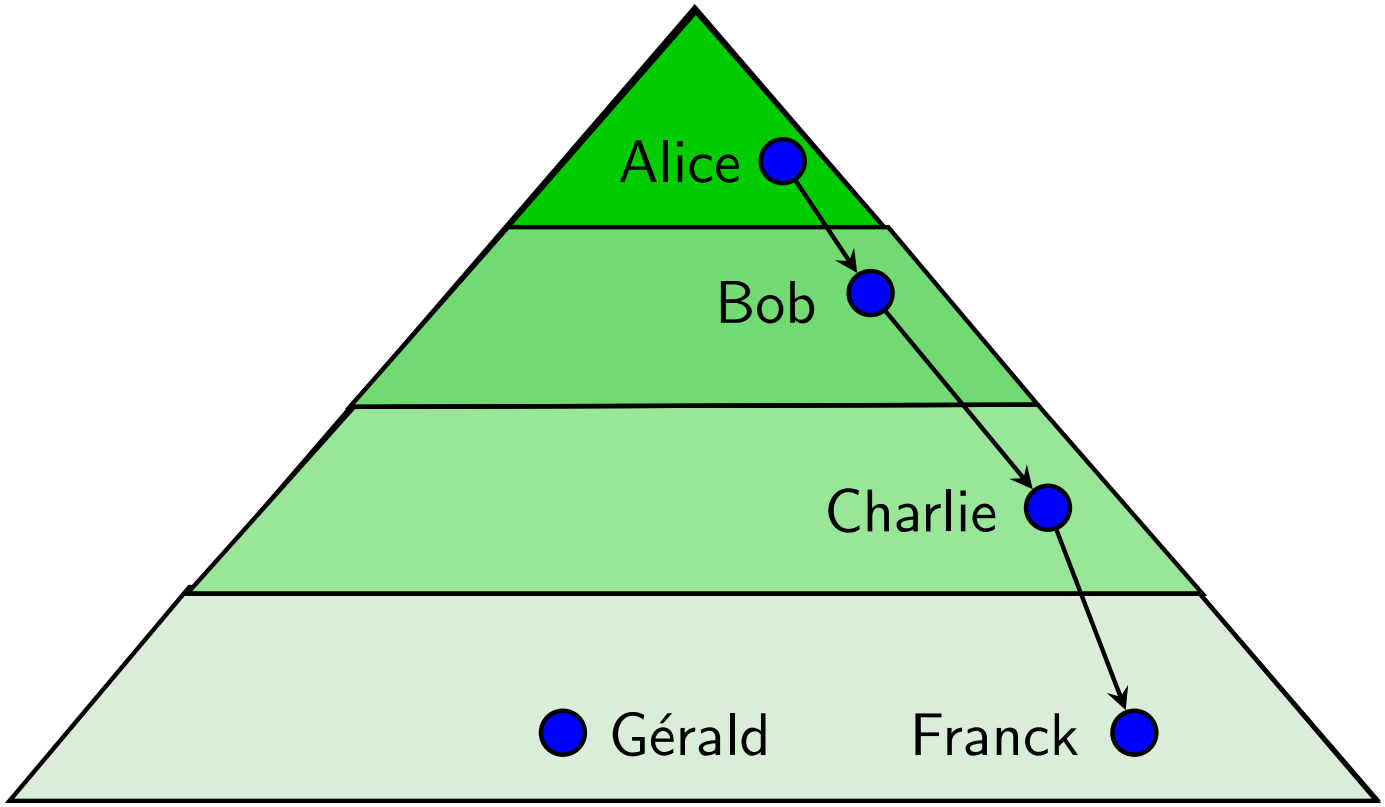
Exemple de chaîne de confiance



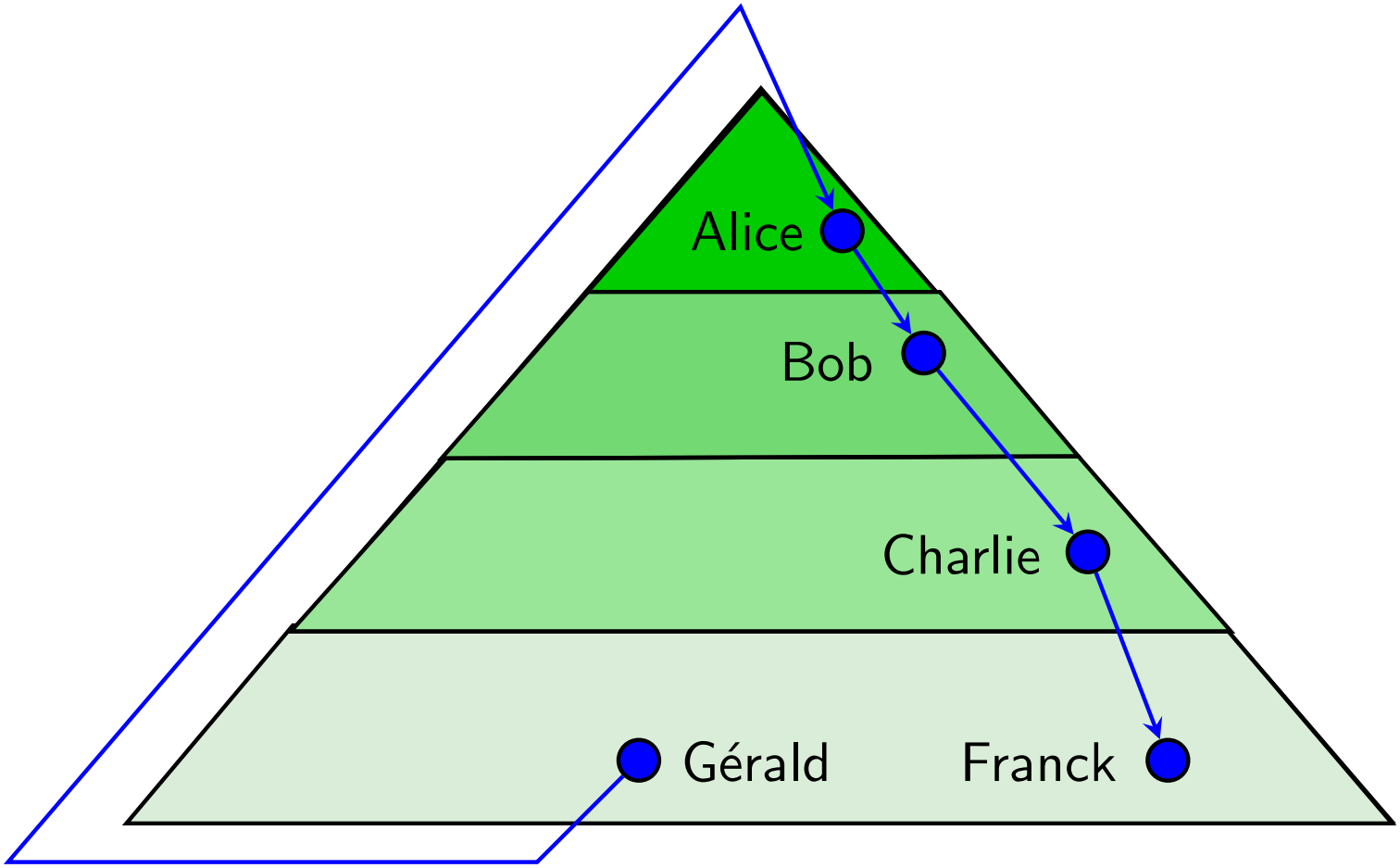
Exemple de chaîne de confiance



Exemple de chaîne de confiance



Exemple de chaîne de confiance

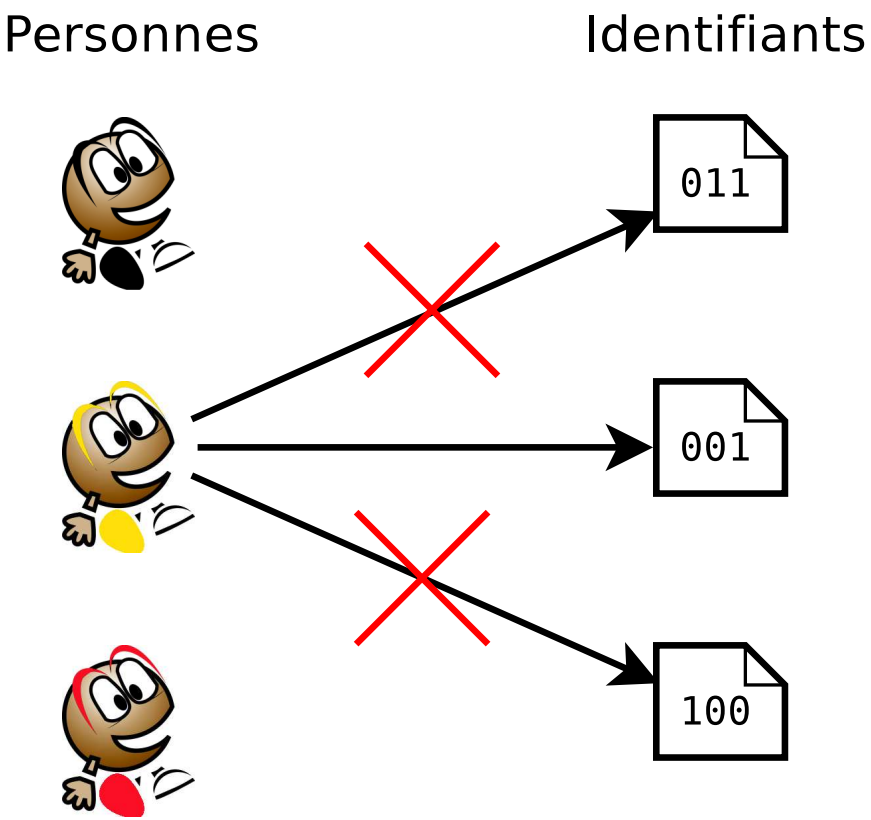


Chaîne de responsabilité

La chaîne de responsabilité :

- Marque la responsabilité des parrains
- Contient la liste des parrains responsables
- Permet de détecter les parrains certifiant des personnes malveillantes

Apport des *NodeId* uniques



Avec limitation des *NodeId*

Complémentarité des 2 mécanismes

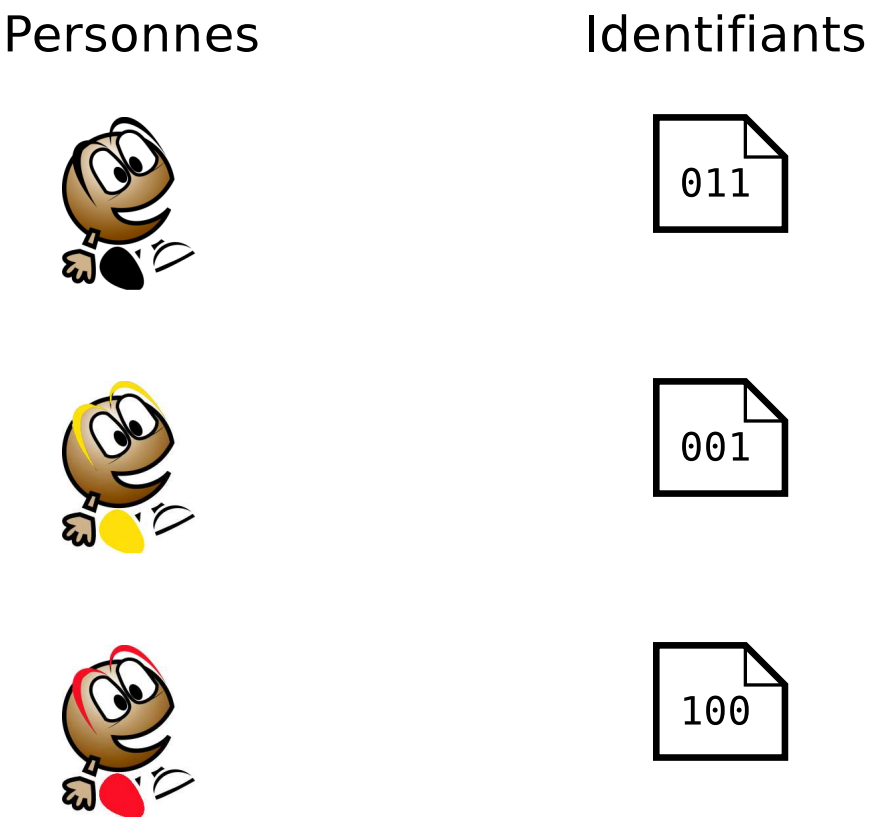
Certification du *ScreenName*

- Permet des identifiants uniques sur tout l'*overlay*
- Assure cette unicité dans le temps
- Nécessite une identification unique des nœuds

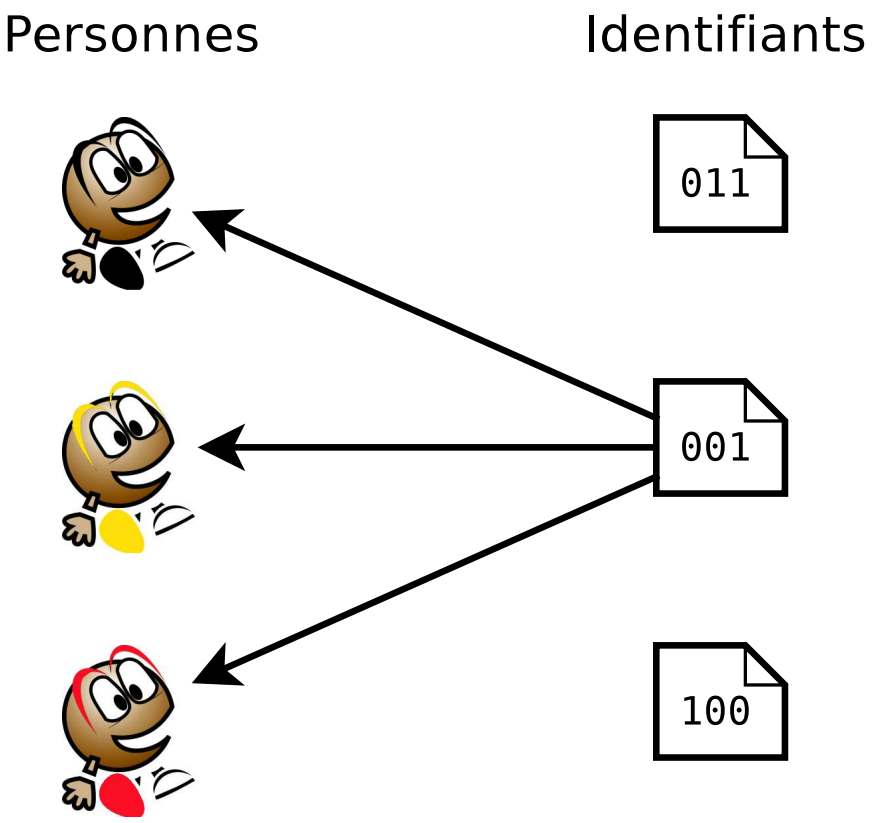
NodeId unique

- Empêche un utilisateur d'apparaître comme beaucoup
- Empêche un utilisateur d'obtenir un grand nombre de fragments de clé et donc de compromettre le réseau
- N'empêcherait pas un utilisateur d'usurper un *ScreenName*

Complémentarité des 2 mécanismes

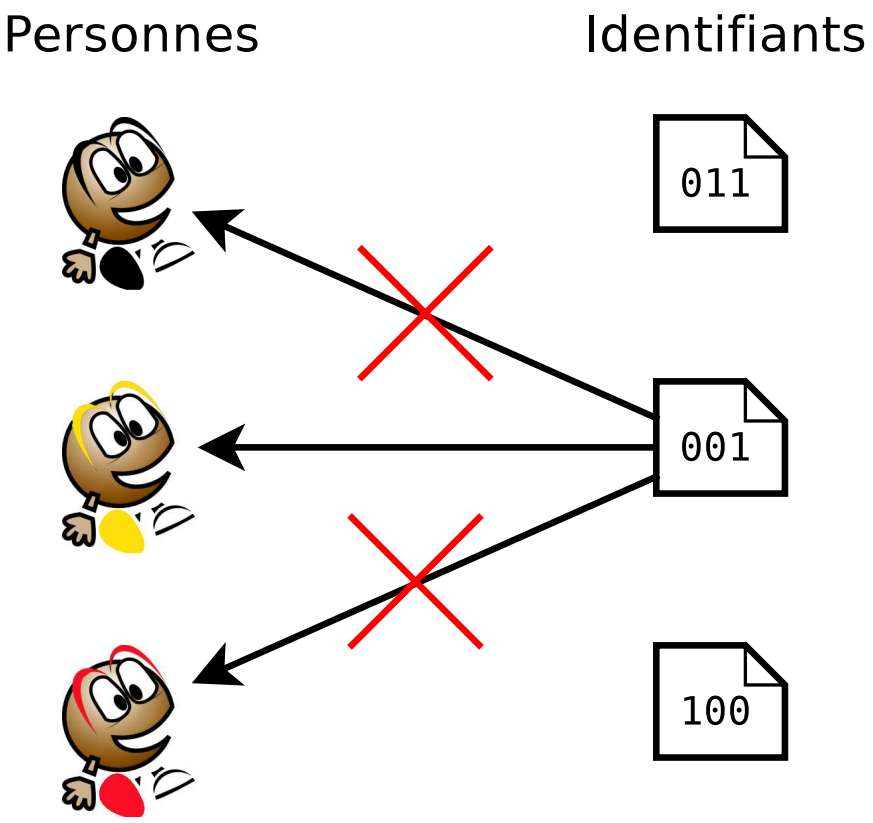


Complémentarité des 2 mécanismes



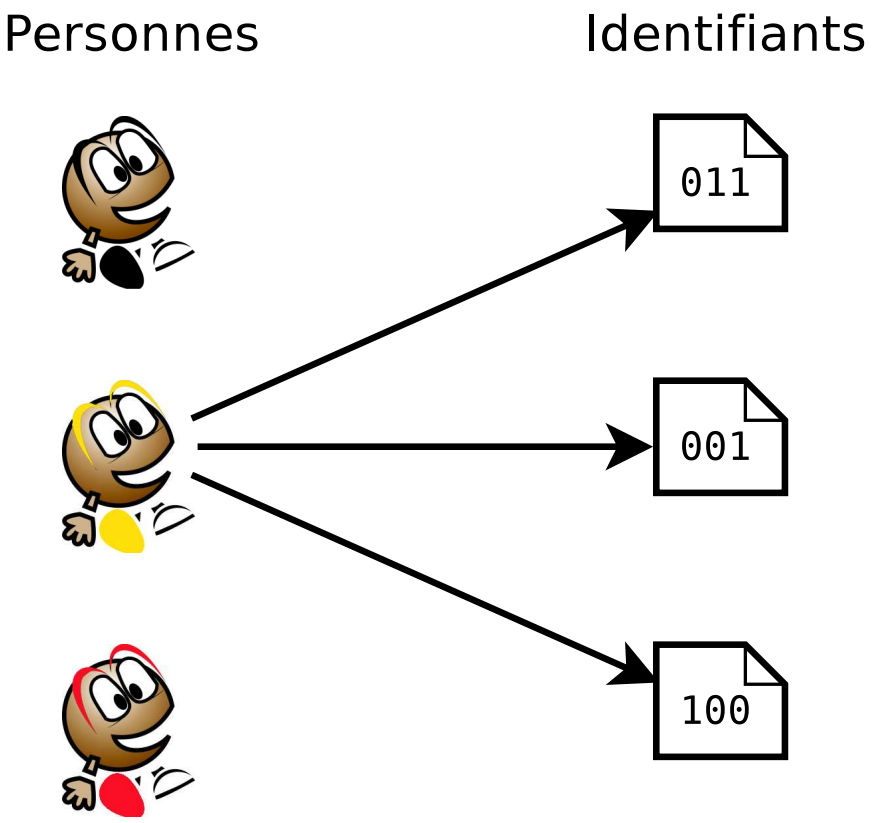
Sans certification des *ScreenNames*

Complémentarité des 2 mécanismes



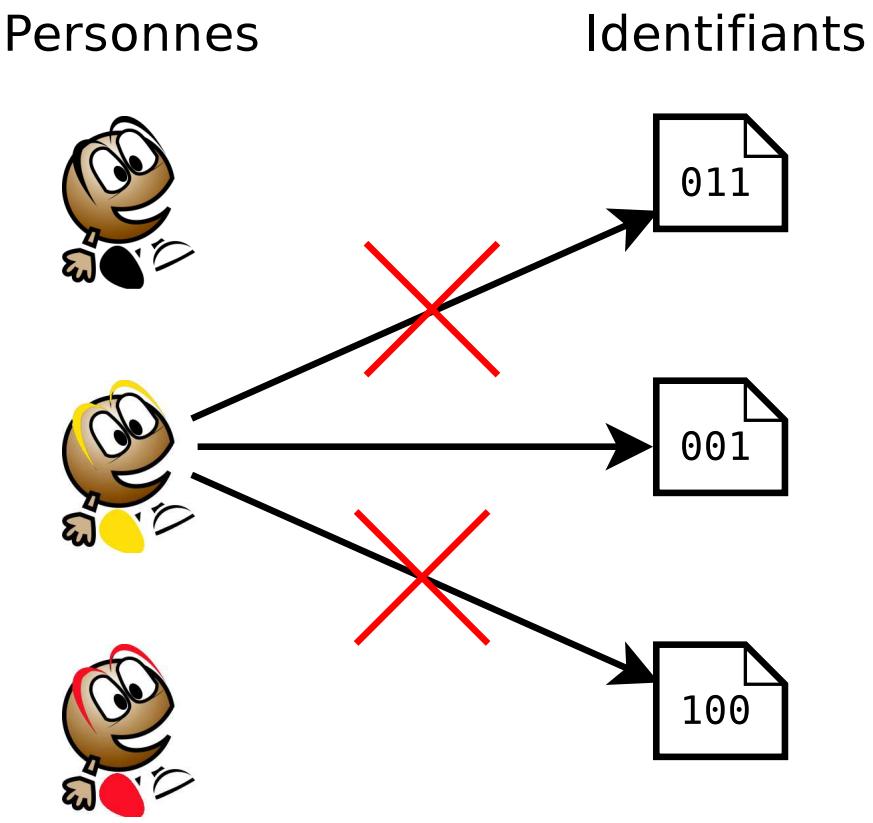
Avec certification des *ScreenNames*

Complémentarité des 2 mécanismes



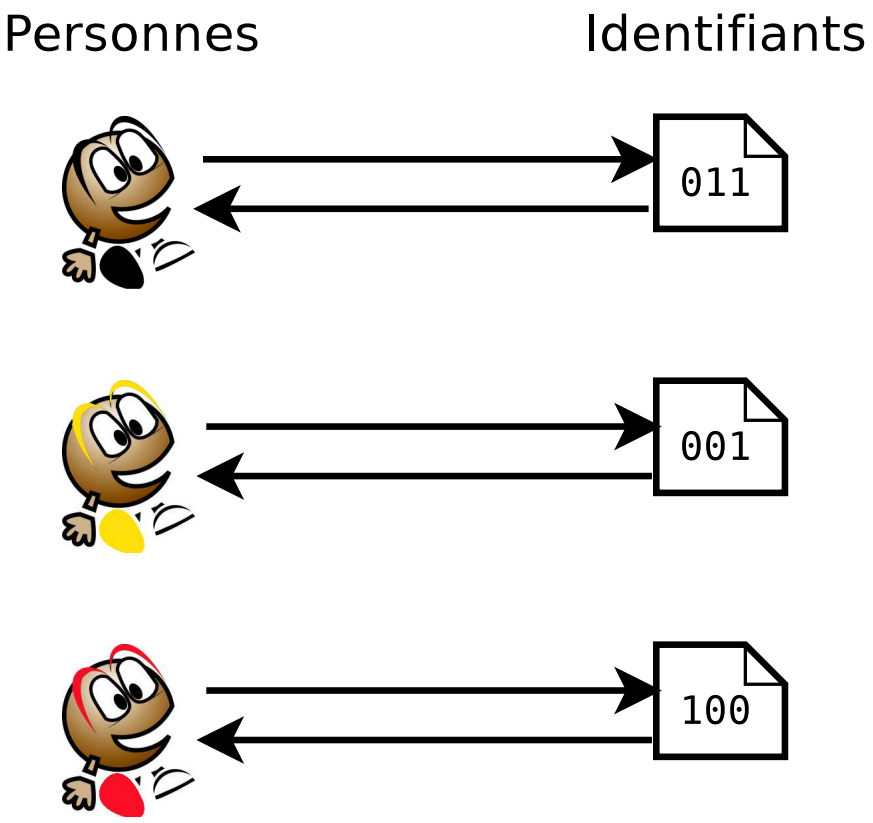
Sans *NodeId* uniques

Complémentarité des 2 mécanismes



Avec *NodeId* uniques

Complémentarité des 2 mécanismes



Avec certification des *ScreenNames* et *NodeId* uniques

Travail réalisé

Notre travail permet de cumuler les avantages de SIP et de P2PSIP :

- Authentification (Annuaire sécurisé)
- Haute disponibilité
- Pas d'infrastructure à maintenir

En revanche, certains problèmes restent à étudier, dont :

- Expressivité de la pyramide de confiance
- Présence de nœuds malveillants
- Libération de *ScreenNames* inutilisés

Perspectives

Perspectives :

- Remplacement de la pyramide de confiance
- Détection de nœuds malveillants
- Révocation de nœuds malveillants
- Évaluation de performances
- Implémentation

Annuaire distribué sécurisé pour un réseau de VoIP Peer-to-Peer

François Lesueur, Ludovic Mé, Hervé Debar

Supélec, équipe SSIR (EA4039)
FT R&D, MAPS/NSS

30 novembre 2006