Gestion distribuée d'identités résistante à la Sybil attack pour un réseau pair-à-pair

François Lesueur (francois.lesueur@supelec.fr)
Ludovic Mé (ludovic.me@supelec.fr)
Valérie Viet Triem Tong (valerie.viettriemtong@supelec.fr)

Supélec, Équipe SSIR (EA 4039), Avenue de la Boulaie - CS 47601, F-35576 Cesson-Sévigné cedex
Tel : 33 (0) 2 99 84 45 92 - Fax : 33 (0) 2 99 84 45 99

Les réseaux pair-à-pair structurés, permettant la mise en place de très grands réseaux efficaces et entièrement distribués, sont vulnérables à la *Sybil attack*. Dans cette attaque, une personne malveillante génère un grand nombre d'identités virtuelles afin d'apparaître comme un grand nombre de personnes physiques et ainsi pouvoir nuire à la disponibilité et à l'intégrité du réseau pair-à-pair. Dans ce papier, nous proposons un système d'identification distribuée pour prévenir cette attaque.

Mots-clés: P2P, Identification, Sybil Attack

1 Introduction

Les grands groupes virtuels, notamment les réseaux pair-à-pair, posent de nouveaux problèmes d'identification. Ainsi, il est difficile d'attribuer une identité virtuelle unique à une personne physique.

Dans le cadre des réseaux pair-à-pair, les propriétés de disponibilité et d'intégrité peuvent être gérées par réplication des données. Or la réplication repose sur le fait que les interlocuteurs utilisés sont physiquement différents. Sans système d'identification, un adversaire peut créer suffisamment d'identités virtuelles pour représenter tous les interlocuteurs d'une requête, et ainsi briser la disponibilité ou l'intégrité des données. Ce problème, connu sous le nom de *Sybil Attack*, est décrit par John R. Douceur dans [Dou02]. John R. Douceur démontre qu'il est impossible, sous des hypothèse raisonnables, de résoudre ce problème d'identification de manière distribuée dans un grand réseau, pendant le fonctionnement du réseau. Dans [Wal02], Dan S. Wallach soulève également le problème mais ne propose qu'une approche centralisée, reposant sur un serveur qui ne certifie qu'un nombre limité d'identités par minute. Dans un tel contexte, un attaquant peut toujours générer autant d'identités qu'il le souhaite.

Dans ce papier, nous étudions le principe de l'identification et proposons un système d'identification *pseudo-unique externe* au système pour un réseau pair-à-pair. Notre système permet de limiter fortement le nombre d'identités virtuelles qu'une personne physique peut présenter et ainsi de protéger le réseau d'une *Sybil attack*. Nous montrons que l'unicité de l'identifiant n'est pas nécessaire, la limitation du nombre d'identifiants suffit.

Dans une première partie, nous présentons les réseaux pair-à-pair structurés ainsi que la *Sybil attack*. Dans une deuxième section, nous étudions les différents mécanismes d'identification. Ensuite, dans une troisième partie, nous analysons l'état de l'art de l'identification distribuée à coût social. Enfin, dans une quatrième partie, nous présentons notre système d'identification distribuée résistante à la *Sybil attack*.

2 Les réseaux pair-à-pair structurés

Les réseaux Pair-à-Pair reposent sur une architecture entièrement décentralisée, où chaque participant a un rôle équivalent. L'objectif est d'éviter les points critiques et d'augmenter la disponibilité du système en agrégeant les ressources de tous les participants, tout en utilisant des algorithmes qui passent à l'échelle.

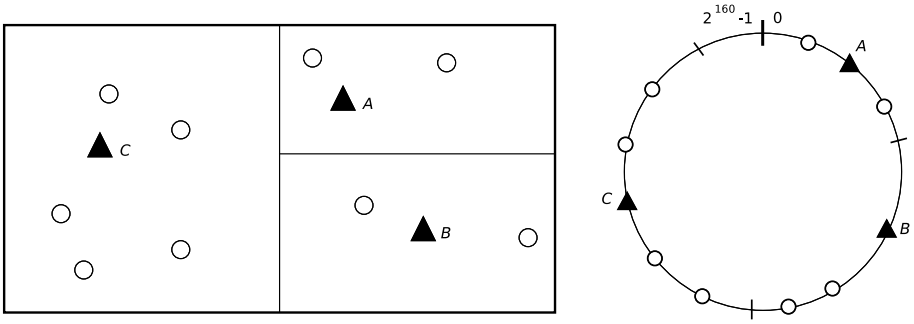


FIG. 1: Espace des identifiants d'un *overlay* structuré. À gauche, une représentation rectangulaire (CAN). À droite, une représentation sous forme d'anneau (Pastry). ▲ représente un nœud (PC) et ○ une ressource (fichier)

Les nœuds sont supposés volatils et peu fiables : un nœud peut rejoindre ou quitter le réseau sans prévenir. Il existe deux grandes familles de réseaux pair-à-pair : les réseaux non-structurés (par exemple Gnutella [Cli00]) et les réseaux structurés (par exemple Pastry [RD01], CAN [RFH+01], Chord [SMK+01], Tapestry [ZKJ01]). Les réseaux non-structurés sont plutôt adaptés à des recherches complexes (portant sur le contenu du fichier) ou à des applications de diffusions, alors que les réseaux structurés sont limités à des recherches simples (portant sur le nom du fichier). En contrepartie, les réseaux structurés sont beaucoup plus performants en terme de coût de communication [CCR04]. Nous présentons ici les réseaux pair-à-pair structurés qui sont les plus vulnérables à la *Sybil attack*. En effet, ces réseaux dépendent d'une organisation très précise qui est très facilement modifiable par cette attaque.

2.1 DHT et routage

Les réseaux structurés reposent sur des algorithmes de routage *proactifs*, c'est-à-dire que pour toute requête il existe une route déjà définie dans les tables de routage.

Ces réseaux implémentent une *DHT (Distributed Hash Table)*. La DHT représente un espace virtuel, l'espace des clés $\mathcal{K}eyIds$. Chaque nœud (PC) est repéré par un identifiant unique $nodeId \in \mathcal{K}eyIds$; de la même façon, chaque *ressource* (fichier, abonnement, ...) est identifiée de manière unique par un identifiant de clé *key*, $key \in \mathcal{K}eyIds$ (dans le cas d'un fichier, la clé est souvent l'empreinte SHA1 de ce fichier). Les nœuds et les ressources partagent donc le même espace d'identifiants, espace fini mais supposé suffisamment grand (couramment de 2^{160} éléments en raison de la taille des empreintes SHA1). Pour accéder à une ressource, un nœud doit connaître la clé de cette ressource. Ce nœud utilise alors la DHT, qui implémente de façon distribuée une fonction liant la clé à la ressource.

L'espace $\mathcal{K}eyIds$, contenant à la fois les identifiants de nœuds (*nodeId*) et les identifiants de ressource (*key*), forme un réseau logique appelé *overlay*. Il est muni d'une mesure sur les identifiants permettant de calculer des distances dans l'*overlay*. Chaque nœud (A) est responsable de la zone de cet espace autour de lui (Z_A). Les nœuds souhaitant insérer des données dans la zone Z_A doivent envoyer ces données à A et les requêtes sur Z_A sont servies par A. Sachant que les nœuds sont volatils et peu fiables, chaque ressource est dupliquée sur n autres nœuds *répliques* (fonction de réplication gérée par l'*overlay*), qui seront alors capable de remplacer A si il quitte l'*overlay*, en prenant le contrôle de Z_A.

La figure 1 donne une représentation rectangulaire et une représentation circulaire d'un *overlay*. Sur ces deux représentations, les nœuds sont responsables des ressources dont ils sont les plus proches. Chaque nœud possède une zone dont il est responsable et connaît les limites de cette zone.

Le routage s'effectue de proche en proche. Chaque nœud dispose d'une table de routage, contenant ses voisins ainsi que d'autres nœuds répartis dans l'espace virtuel. Cette table de routage est mise à jour en arrière-plan, en fonction des modifications du réseau. Pour accéder à une ressource R, le nœud demandeur A envoie sa requête au nœud B contenu dans sa table ayant l'identifiant le plus *proche* de R. Si B ne possède pas R, il redirige la requête de la même façon. Par convergence de l'algorithme de routage, le message arrive au nœud X qui est responsable de la zone contenant R, en un nombre de sauts de l'ordre de $log(\mathcal{N})$, $\mathcal{N}$ étant la taille du réseau.

2.2 Problème de la Sybil attack

Afin de garantir la disponibilité des ressources et du routage, l'attribution des identifiants de nœud est supposée aléatoire. Cette hypothèse est réaliste dans un système de test mais rien n'empêche un attaquant de la compromettre en choisissant son identifiant ou en en créant plusieurs.

Dans le cas où l'attribution des identifiants n'est plus aléatoire, [Wal02] décrit plusieurs problèmes qui peuvent se poser. Dans un premier temps, les identifiants de nœud peuvent ne plus être répartis uniformément. Si la densité de nœuds n'est pas constante dans tout l'espace, la répartition de la charge n'est plus équitable et les algorithmes de routage sont moins efficaces. Le réseau perd donc en qualité générale.

Dans un deuxième temps, un attaquant ou une coalition d'attaquants peut choisir un ensemble spécifique d'identifiants afin de contrôler une ressource ainsi que toutes ses répliques. Dans ce cas, l'attaquant peut censurer ou modifier la ressource attaquée, sans être détecté.

Enfin, dans un troisième temps, en choisissant également un ensemble spécifique d'identifiants, un attaquant peut contrôler tous les nœuds connus par le nœud attaqué (toutes les entrées de sa table de routage). L'attaquant se retrouve ainsi dans une position de mandataire et peut filtrer ou observer les requêtes du nœud attaqué.

Dans le cadre de la *Sybil attack* formalisée dans [Dou02], un attaquant génère un grand nombre d'identifiants. À partir de ce grand nombre d'identifiants, l'attaquant peut extraire un ensemble spécifique d'identifiants, lui permettant de réaliser une des attaques présentées. Le problème soulevé est donc à la fois de générer des identifiants aléatoires mais également d'empêcher un attaquant de générer un grand nombre d'identifiants lui permettant d'en choisir quelques uns spécifiquement.

Notre contribution permet de limiter le nombre d'identifiants qu'une personne peut obtenir tout en générant ces identifiants de manière réellement aléatoire, afin de prévenir la *Sybil attack*. Dans la suite, nous étudions le principe de l'identification.

3 Mécanismes d'identification

Dans cette section, nous analysons et classifions les différents mécanismes d'identification, afin d'en extraire de bonnes propriétés pour un système d'identification dans un réseau pair-à-pair.

3.1 Deux approches possibles de l'identification

Pour gérer les identités des utilisateurs, deux approches peuvent être utilisées. Dans la première, les identités demandent du temps pour être créées. Dans la seconde, l'identité est valable immédiatement mais peut être annulée plus tard. Dans cette section, nous présentons deux exemples d'utilisation avec les problèmes soulevés.

3.1.1 Réputation : présumé coupable

Une première approche possible repose sur la notion de *réputation*. Toute personne est d'abord mise à l'épreuve avant de faire partie du système. Dans le cadre d'un réseau pair-à-pair de partage de fichiers, cette approche peut être réalisée de la manière suivante : lors de sa première connexion, un nouveau membre ne peut pas profiter de toute la puissance du réseau. Dans un premier temps, seuls les membres entièrement inoccupés acceptent de lui envoyer les fichiers demandés. Ensuite, le nouveau membre doit lui même envoyer ces fichiers à d'autres membres les demandant. Sa réputation est calculée directement par la quantité de fichiers qu'il transmet à d'autres membres. Ainsi, une fois qu'il a envoyé lui même beaucoup de fichiers, il peut recevoir de nouveaux fichiers plus rapidement.

À l'inverse, si ce nouveau membre se comporte de manière égoïste et ne renvoie pas à d'autres personnes les fichiers qu'il télécharge, sa réputation reste faible. Le fait de ne pas envoyer de fichiers l'empêche donc de pouvoir télécharger rapidement d'autres fichiers.

Un membre a donc tout intérêt à partager les fichiers qu'il possède, afin d'augmenter sa réputation. De plus, ce membre a intérêt à conserver la même identité pour maximiser sa réputation et donc son pouvoir de téléchargement. Une personne peut donc créer beaucoup d'identités, mais une identité permettant l'utilisation complète du système met du temps à se créer.

Cependant, un tel système impose de fortes contraintes sur un nouvel utilisateur, qui est obligé de passer par une phase de test : il est présumé coupable. Le démarrage est donc lent et le calibrage du temps nécessaire avant l'insertion totale est problématique. En effet, les utilisateurs ont des capacités différentes (CPU, bande passante) et il faut donc trouver un critère le plus équitable possible.

3.1.2 Éviction : présumé innocent

Une seconde approche repose sur l'éviction des personnes malveillantes. Toute personne est initialement considérée comme étant membre à part entière du groupe. Dans le cadre d'un réseau pair-à-pair, lorsqu'un nouveau membre se connecte, il obtient instantanément les mêmes droits (accès aux services, vitesse de téléchargement) au sein du réseau qu'un membre plus ancien. En revanche, dès lors qu'un membre montre un mauvais comportement, il est exclu du réseau.

Ce fonctionnement, plus simple, suppose *a priori* les nouveaux membres comme bienveillants : ils sont présumés innocents. Cette présomption permet d'économiser des ressources et d'accélérer le démarrage de chaque personne. En revanche, si un membre exclu du réseau peut se reconnecter immédiatement sous une autre identité, alors ce mécanisme est inopérant.

Pour permettre un tel fonctionnement, le système doit être robuste à un faible nombre de membres malveillants, puisque leur exclusion ne peut se faire qu'après avoir constaté leur mauvais comportement. Dans le cas des réseaux pair-à-pair, cela ne pose pas de problème puisque ces réseaux sont par définition très redondants et tolérants aux fautes.

3.2 Taxonomie des systèmes d'identification

Les deux exemples présentés ne sont pas fondamentalement différents. Dans le système à réputation, l'identité est créée au sein du système. À l'inverse, dans le système à éviction, l'identité est créée à l'extérieur du système. Nous parlons ici d'identification interne ou externe. Dans les deux approches, pour limiter le nombre d'identités d'une même personne, la création d'une identité doit avoir un *coût*, matériel ou non. L'objectif d'un système d'identification est de lier une identité à une personne unique, et réciproquement une personne à une identité unique.

Dans une première partie, nous explicitons les deux types d'identification. Ensuite, nous proposons trois types de *coût*.

3.2.1 Type d'identification

Nous distinguons ici l'identification interne de l'identification externe.

Identification interne Dans ce type d'identification, l'identité est créée à l'intérieur du système. La personne doit donc faire partie du système, même de manière minimale, avant que son identité ne soit avérée et reconnue. Le système doit donc être robuste à un grand nombre de fausses identités non encore vérifiées. En revanche, la personne ne peut devenir membre à part entière qu'après avoir franchi certains tests.

Le coût doit être suffisamment faible pour que tout le monde puisse payer son insertion, mais en même temps suffisamment élevé pour dissuader de payer plusieurs fois. Cet équilibre est difficile à atteindre, cette contrainte limite les types de coût possibles. Le coût doit également se situer dans la boucle de fonctionnement du système, puisque c'est durant le fonctionnement du système que le coût doit être payé.

Identification externe Dans un système où l'identification est externe, l'identité est créée à l'extérieur de l'application. La personne doit donc s'identifier par un moyen tiers, éventuellement sans rapport avec l'application. Dès son insertion, une personne nouvellement identifiée est considérée comme un membre à part entière, il n'y a pas de tests d'entrée au niveau de l'application. Une telle personne peut donc éventuellement avoir un comportement malveillant envers le système, puisque le système d'identification ne garantit que l'identification, pas le comportement.

Tout comme pour l'identification interne, le coût doit être choisi selon plusieurs contraintes. Tout d'abord, il doit être suffisamment faible pour que tout le monde puisse payer son insertion, mais en même temps suffisamment élevé pour dissuader de payer plusieurs fois. En revanche, le type de coût est entièrement libre, puisqu'il n'est pas lié au système.

3.2.2 Coût

Afin de limiter le nombre d'identités, un *coût* doit être associé à la création d'une identité. Ce coût doit être le plus équitable possible, afin que tout le monde puisse le payer une et une seule fois. Il faut donc pouvoir calibrer ce coût, c'est-à-dire pouvoir fixer l'investissement nécessaire pour obtenir une identité. Dans cette section, nous présentons trois types de coût pouvant être utilisés.

Coût monétaire La création d'une identité peut coûter une certaine somme monétaire. Cependant, il existe de grandes disparités entre les quantités d'argent possédées par différentes personnes. Le calibrage de ce coût monétaire semble donc très complexe voire impossible. De plus, l'argent utilisé pour cette identification aurait pu servir à autre chose. L'utilisateur a donc un choix à faire entre le système d'identification et d'autres choses, la valeur ressentie du système doit donc être supérieure à la valeur ressentie d'autres biens du même prix.

Ce choix vient du fait que l'accès au système engendre une perte matérielle (argent).

Les systèmes suivants utilisent ce coût :

- Le système **DNS** [Moc87] propose une identification externe, *via* un coût monétaire. Le problème du *cybersquatting* est bien connu, quand une personne achète des milliers de domaines avant les détenteurs des marques associées. Ce problème illustre qu'une identification contre simple valeur monétaire n'est pas suffisante pour proposer un système d'identification (une personne vers un seul domaine).
- Les **adresses IP** (identification externe avec coût monétaire) sont utilisées comme identifiant dans Chord [SMK+01] par exemple. En effet, une adresse IP devrait pointer vers un seul ordinateur et donc vers une seule personne. Dans l'Internet actuel, cette hypothèse n'est plus réaliste. Avec les NAT, plusieurs ordinateurs partagent la même adresse IP et avec IPv6, un ordinateur possède une grande plage d'adresses IP.
- **Les autorités de certification** [Int93] pratiquent également une identification externe avec coût monétaire. Ces autorités sont très hiérarchisés et plus strictes que les DNS, les identités physiques annoncées devant être prouvées. Cela permet d'obtenir un système d'identification robuste. Cependant, les autorités de certification manquent de souplesse, l'inscription étant complexe et relativement onéreuse.
- **Trusted Computing** [TCG05] pratique une identification externe avec coût monétaire. Trusted Computing repose sur la présence d'une puce de chiffrement intégrée directement sur la carte mère (le TPM) et l'utilisation de code signé. Le coût est celui du matériel nécessaire pour se connecter (un ordinateur), donc n'est pas une perte pour l'utilisateur. De l'autre côté, ce coût est suffisamment élevé pour empêcher un adversaire de générer un grand nombre d'identités. Trusted Computing pose cependant au moins deux problèmes. D'une part, ce système n'est pas encore déployé aujourd'hui, nécessitant une mise à jour du matériel et du logiciel. D'autre part, l'utilisation de Trusted Computing impose de fortes contraintes sur l'utilisation d'un ordinateur et les logiciels autorisés : son utilisation est limitée à un environnement logiciel propriétaire. Ces fortes contraintes rendent ce système moins idéal qu'il ne paraît.

Coût informatique La création d'une identité peut demander du calcul ou du stockage. Par exemple, dans [CDG+02], Castro *et al* propose l'utilisation d'un puzzle calculatoire. Ils proposent que la position du nœud du réseau soit l'empreinte de la clé publique du membre et que, pour s'insérer, les membres du réseau ne choisissent que des clés dont l'empreinte termine par au moins p bits égaux à 0. L'objectif est de limiter le nombre d'identités calculables. En effet, le coût de calcul nécessaire pour trouver une telle paire de clés est en $O(2^p)$. Le problème de l'approche de Castro *et al* est qu'un attaquant peut pré-calculer un grand nombre de clés et donc arriver à contrôler beaucoup de nœuds. Pour y remédier, Borisov propose dans [Bor06] un autre schéma de puzzle calculatoire dans lequel les challenges sont rafraîchis régulièrement. Un attaquant ne peut donc pas pré-calculer les résultats. Malheureusement, de façon générale, des attaquants possédant suffisamment de puissance de calcul peuvent toujours contourner la protection d'un puzzle calculatoire. Nous retrouvons finalement les mêmes problèmes que ceux du coût monétaire.

eMule [HBMS04], un réseau pair-à-pair de partage de fichiers, propose une identification interne et reposant sur un coût machine : de la bande passante. Le principe fonctionne puisque la bande passante est justement ce qui sert au partage de fichiers : le coût d'insertion est le comportement attendu de la personne,

qui est ici visible extérieurement. Le problème du stockage et de la communication de cette réputation dans un environnement pair-à-pair est hors de notre propos. Cependant, même si ce système est suffisant pour eMule, il ne s'agit pas réellement d'un système d'identification unique, puisqu'un utilisateur peut créer autant d'identités qu'il le souhaite. Il faut noter qu'eMule n'est pas un réseau pair-à-pair structuré et repose sur la présence de serveurs pour localiser les données.

Coût social La création d'une identité peut demander une reconnaissance sociale antérieure. Cette approche pose de nouveaux problèmes. Il faut d'abord être capable d'informatiser la reconnaissance sociale. Ensuite, ce coût ne doit pas être ressenti par l'utilisateur : le temps passé à créer cette reconnaissance doit apporter autre chose, l'identité ainsi créée étant un plus. La création d'une telle identité n'implique aucun coût matériel, et est égalitaire entre les différentes personnes. La reconnaissance sociale peut avoir lieu par un milieu virtuel. Il faut dans ce cas faire bien attention à différencier le temps humain du temps machine.

Cette identification peut se réaliser au travers de réseaux de connaissances ou de forums de discussion.

L'accès au système n'engendre aucune perte, le coût est en fait un temps humain antérieur.

Les systèmes suivants rentrent dans cette catégorie :

- **eBay** [eBa], site de vente en ligne entre particuliers, utilise le système de réputation. L'identification est interne et demande un coût social. Pour être reconnue, une identité doit avoir prouvé sa bonne foi et son existence par des actes réels. La réputation de l'identité virtuelle est donc générée par des interactions humaines réelles, ce qui lui donne sa valeur.
- **PGP** [Zim95] utilise un système d'identification externe avec un coût social. PGP informatise les réseaux de confiance. Chaque personne accorde sa confiance dans l'identité d'autres personnes qu'il connaît directement. Ensuite, ces personnes elles-mêmes accordent leur confiance à d'autres. PGP repose ensuite sur la transitivité de la confiance pour trouver un *chemin de confiance* entre deux personnes : un *chemin de confiance* est une suite de lien de confiance formant un chemin dans le graphe. La propriété sous-jacente est que la population mondiale forme un graphe *petit-monde* [Mil67], dans lequel chaque couple de personnes est séparé par au plus six personnes (un lien étant un lien de confiance).
- **GMail** [Goo] est le service mail de Google. GMail utilise une identification externe à coût social. Une de ses caractéristiques principales est la capacité de stockage permise à l'utilisateur : 1 Giga-octets lors du lancement, plus aujourd'hui. Afin d'empêcher des personnes de créer un grand nombre de boîtes pour les utiliser comme des espaces de partage et de pouvoir maîtriser la croissance du service, Google a mis en place un système d'invitations. Pour ouvrir un compte GMail, il faut y avoir été invité par un autre utilisateur. Google offre à chaque utilisateur un certain nombre d'invitations, afin de permettre à tout le monde d'inviter des personnes extérieures. Cette modération n'a pas entravé le développement de GMail, puisqu'il a été très vite largement utilisé et que chaque personne souhaitant ouvrir un compte légitime a la possibilité d'obtenir une invitation.

3.2.3 Choix du coût social

Le coût social semble le plus intéressant à exploiter. Il s'agit d'une ressource équitable entre les différentes personnes, ce qui permet à chacun de s'inscrire mais ne permet à personne d'abuser du système. De plus, ce coût ne provoque pas de perte matérielle et peut donc être plus facilement accepté. Étant donné que nous retenons la valeur sociale comme coût d'une identité et que les personnes physiques ne sont pas impliquées dans la boucle de fonctionnement des réseaux pair-à-pair structurés, nous proposons un système à identification externe. De plus, les réseaux pair-à-pair ayant comme spécificité de ne pas dépendre d'un ensemble de serveurs particuliers, le système d'identification doit être lui aussi entièrement distribué.

4 État de l'art de l'identification distribuée externe à coût social

Dans cette section, nous présentons un travail antérieur proposant une identification distribuée externe à coût social et nous montrons qu'il est en fait vulnérable à la *Sybil attack*.

4.1 Approche distribuée des réseaux de confiance

Le principe des réseaux de confiance (dont PGP fait partie) représente une identification externe à coût social et nous proposons donc d'utiliser ce principe pour l'identification. Cependant, le calcul d'un chemin de confiance est réalisé par des serveurs spécialisés. Ces calculs reposent sur de l'algorithmie des graphes au sein de très grands graphes. Dans un contexte de réseau pair-à-pair, il n'est pas souhaitable de dépendre de la présence de serveurs spécialisés, qui nuisent à la disponibilité du système. En même temps, il n'est pas possible pour des raisons de stockage et de coût de communication que chaque personne possède localement une copie du graphe complet. [CBH03] propose une approche distribuée des réseaux de confiance, que nous rappelons ici.

Chaque personne A possède un couple de clés publique/secrète (P_A, S_A). L'ensemble des personnes est vu comme un graphe $G(V,E)$, où V et E sont respectivement l'ensemble des sommets et l'ensemble des arcs. Les sommets représentent les clés publiques et et les arcs les liens de certification. Plus précisément, il existe un arc de P_u à P_w s'il existe un certificat signé par S_u liant P_w à une identité. Ainsi, une chaîne de certification de P_u à P_w (un chemin de confiance) est représentée par un chemin dans le graphe G de P_u à P_w. Afin que chaque nœud puisse calculer les chemins le concernant, chaque nœud conserve un cache local d'un sous-ensemble du graphe de certification.

Le fonctionnement de ce système peut être découpé en quatre phases : la création d'un couple de clés publique/secrète, la création de certificats, l'échange de certificats et la création des caches locaux de certificats. L'étape la plus intéressante est la création des caches locaux. En effet, le système est trop grand pour que chaque personne conserve l'intégralité des certificats : le système doit donc être capable de sélectionner à l'avance les certificats qui lui serviront pour établir des chemins de confiance avec ses futurs interlocuteurs. L'algorithme proposé crée un sous-graphe en forme d'étoile autour de la personne concernée. Cet algorithme sélectionne n certificats à conserver localement selon c chemins, n et c étant des valeurs configurées par l'utilisateur (n de l'ordre de 10 à 100 et c de 3 dans les simulations réalisées). Chaque utilisateur u possède $\frac{n}{2}$ certificats représentant un sous-graphe G_{out} de c chemins ayant pour origine u et $\frac{n}{2}$ certificats représentant un sous-graphe G_{in} de c chemins ayant pour arrivée u. Ces chemins sont construits de manière incrémentale à partir de u, en choisissant à chaque fois le candidat ayant le plus haut degré.

Lorsque u et v veulent vérifier leurs identités respectives, ils s'envoient mutuellement leurs G_{in}. Ensuite, chacun fusionne le G_{in} reçu avec le G_{out} possédé, l'objectif étant de trouver un chemin entre u et v. Les simulations sont très concluantes. Par exemple, dans le cas d'un graphe PGP contenant 5000 nœuds et 35000 certifications (graphe PGP de 1998), avec $n = 30$ et $c = 3$, le taux de réussite pour trouver un chemin entre deux personnes est de 90%.

Le problème de la tolérance à des personnes malveillantes est abordé. Trois cas différents sont envisagés, où une personne malveillante essaie de tromper le système en émettant de faux certificats : un certificat qui lie une clé publique P_u à u' au lieu de u, un certificat qui lie v à une clé publique P'_v au lieu de P_v ou un grand nombre de certificats entièrement factices. Les deux premiers cas sont traités par un mécanisme de résolution de conflits. En effet, dans ces deux cas, en plus du faux certificat, il existe dans le système le certificat original. Le troisième cas est plus problématique et correspond à la *Sybil attack* [Dou02].

Pour illustrer ce problème, prenons la situation suivante. Alice, Bob et Charlie sont trois personnes se faisant confiance deux à deux : leurs clés publiques sont donc certifiées par chacun des deux autres participants. Imaginons maintenant que Charlie, qui a gagné la confiance d'Alice et de Bob, soit en fait malveillant : soit Charlie a joué un rôle pour gagner la confiance de ses pairs, soit Charlie est par exemple victime d'un logiciel malveillant (vers). Charlie peut générer n identités, n arbitrairement grand, et certifier chacune de ces nouvelles identités avec sa clé publique. Alice et Bob faisant confiance à Charlie, ils vont aussi faire confiance aux n identités factices créées par Charlie.

4.2 Traitement de la Sybil attack dans le cas des réseaux de confiance

Nous présentons d'abord le comportement d'une *Sybil attack* dans un réseau de confiance puis nous étudions les métriques d'authentification, permettant de s'en prémunir.

4.2.1 Comportement d'une Sybil attack

Dans le cadre des réseaux de confiance, il existe deux types de *Sybil attack* diamétralement opposées : la première par la certification de *n* personnes par une unique personne Eve présente dans le réseau et la seconde par la certification chaînée de *n* personnes à partir d'Eve. Nous présentons ici les problèmes pour détecter chacune de ces attaques dans le cadre où les utilisateurs ne connaissent pas le graphe global, mais juste un sous-graphe. Dans tous les cas, l'attaque aboutit à un grand nombre de fausses identités présentes dans le réseau. Pour retirer les *n* personnes certifiées malicieusement à partir d'Eve, il suffit de retirer Eve du réseau de confiance, ce qui supprime naturellement les *n* autres fausses identités. Le problème est donc de détecter que c'est Eve qu'il faut retirer du réseau de confiance.

La première attaque (figure 2) consiste en la certification de *n* personnes directement par une unique personne, Eve. Dans le cadre d'une *Sybil attack*, l'attaquant génère un grand nombre d'identités dans le but de réaliser des actions néfastes pour le système. Si les utilisateurs légitimes du réseau ont un moyen de détecter ces actions néfastes, alors ils peuvent détecter que l'une des *n* identités certifiées par Eve est malveillante. Dans ce cas, les utilisateurs légitimes peuvent déclarer que la personne qui l'a certifiée, Eve, a soit fait preuve de négligence dans ses vérifications, soit fait preuve de malveillance. Eve est donc directement incriminée. Néanmoins, si les actions malveillantes des *n* fausses identités ne sont pas détectées, alors Eve n'est pas incriminé.

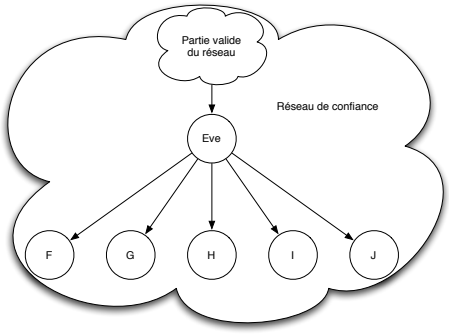


FIG. 2: *Sybil attack* en largeur.

La seconde attaque (figure 3) consiste en la certification chaînée de *n* personnes à partir d'une unique personne, Eve. Dans ce cas, la détection d'une identité frauduleuse I ne permet pas d'incriminer Eve aussi facilement que précédemment. En effet, Eve n'a pas certifié I directement. Pour incriminer Eve, la seule solution est d'incriminer tout le chemin de confiance menant à I, or cela incrimine également tous les nœuds légitimes situés avant Eve. Ce type d'attaque semble encore plus difficile à détecter que le type précédent.

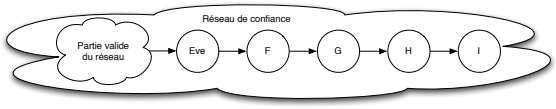


FIG. 3: *Sybil attack* en longueur.

Enfin, ces deux types d'attaque peuvent bien sûr être associées (figure 4).

Un système d'identification résistant à la *Sybil attack* doit donc être muni de protections contre ces deux types de fraude : certifications directes et certifications chaînées d'identités factices.

4.2.2 Métriques d'authentification

Pour détecter les identités factices et ainsi contrer la *Sybil attack*, [CBH03] renvoie à des métriques d'authentification, présentées notamment dans [RS99]. Cependant, nous pensons que le système proposé

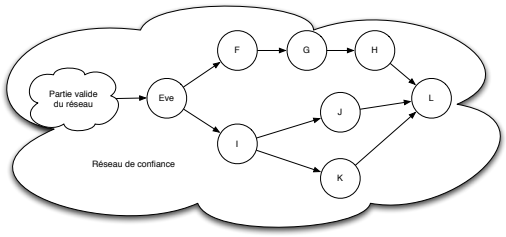


FIG. 4: *Sybil attack* mixte.

dans [CBH03] n'est pas compatible avec les métriques référencées.

Les métriques d'authentification servent à évaluer numériquement la confiance en un chemin de confiance et ainsi détecter les chemins pouvant être associés à des *Sybil attack*. [RS99] propose une comparaison de cinq métriques desquelles nous extrayons des principes communs. Les métriques de *Beth-Borcherding-Klein* [BBK94] et de *Maurer* [Mau96] différencient deux types de certifications : les certifications directes et les certifications indirectes. Chaque chemin de confiance est composé d'une suite de certifications indirectes et se termine par une certification directe. De plus, chaque certification est valuée dans $[0, 1]$, permettant de calculer une valeur pour chaque chemin dans $[0, 1]$ (1 étant la confiance la plus forte).

La métrique de *Reiter-Stubblebine* [RS98] n'utilise qu'un type de certification entre les nœuds du graphe, et les arcs ne sont pas valués. *Reiter* et *Stubblebine* proposent deux métriques différentes à partir d'un tel graphe. Une première qui retourne l'ensemble de taille maximale des chemins de taille inférieure à b entièrement disjoints entre deux nœuds ; une seconde qui retourne un ensemble de chemins de taille inférieure à b et le nombre minimum de nœuds à retirer du graphe pour briser tous ces chemins.

La métrique de *Zimmermann* [Zim95] est celle utilisée dans PGP. Les certifications sont assorties d'une *valeur de confiance* : *unknown*, *untrusted*, *marginally trusted* ou *fully trusted*. L'identité d'une personne u est considérée comme valide par w s'il existe un chemin *fully trusted* entre u et w ou deux chemins *marginally trusted*. Le nombre de chemins nécessaires peut-être adapté.

Chacune de ces métriques permet donc d'évaluer la confiance en un chemin de confiance, en défavorisant les structures de confiance liées aux attaques. Dans le cas de la *Sybil attack*, il est attendu que les chemins menant vers les nœuds fictifs aient une valeur de confiance faible, permettant de détecter qu'il ne s'agit pas de personnes réelles.

Dans tous les cas, ces métriques supposent une connaissance globale du graphe pour calculer la confiance en un chemin. De plus, à l'exception de *Reiter-Stubblebine*, les arcs du graphe sont valués. Dans le système proposé dans [CBH03], chaque nœud ne connaît qu'un sous-graphe du système local et les arcs ne sont pas valués. [CBH03] crée un simple chemin entre deux nœuds du graphe, le succès de l'opération consistant à créer un chemin éventuellement unique là où l'étude du graphe complet pourrait en faire apparaître un grand nombre. Si chaque nœud connaît le graphe entier, le coût en stockage et en communications devient problématique : cette solution n'est pas envisageable. Si les arcs sont valués, la condition de succès n'est plus de trouver un chemin quelconque entre deux nœuds mais de trouver un chemin avec une certaine valeur minimale : la valuation des arcs entraîne nécessairement une chute du taux de réussite. Il semble donc difficile de protéger [CBH03] de la *Sybil attack*.

Dans la suite de ce papier, nous proposons un nouveau système d'identification distribuée externe à coût social, résistant à la *Sybil attack*.

5 Identification distribuée résistante à la Sybil attack

Dans cette partie, nous proposons notre système d'identification distribuée résistante à la *Sybil attack* . Nous présentons d'abord le principe, puis nous introduisons le modèle ainsi qu'une instanciation de ce modèle. Enfin, nous montrons la résistance de ce système et son adaptation aux réseaux pair-à-pair structurés.

5.1 Principe de l'identification proposée

Nous proposons un mécanisme d'identification pour les réseaux pair-à-pair. Notre méthode est résistante à la *Sybil attack* puisqu'elle empêche un membre de contrôler une partie disproportionnée des identifiants. Cette méthode repose sur une identification externe contre un coût social et s'appuie sur un mécanisme d'invitations permettant à un membre d'*inviter* d'autres membres dans le réseau. Lorsqu'un membre rejoint le réseau, son parrain lui attribue des invitations lui permettant à son tour de parrainer de nouveaux membres. Le nombre d'invitations que le nouveau membre reçoit dépend de la taille courante du réseau et du nombre d'invitations dont dispose son parrain. Enfin, de nouvelles invitations sont créées au fur et à mesure de l'augmentation de la taille du réseau. La distribution des nouvelles invitations est un point crucial du système et doit être réalisée avec précaution par chaque parrain.

Les membres fondateurs du réseau forment un groupe de membres privilégiés. Ces membres doivent s'accorder régulièrement sur le nombre de nouvelles invitations à créer, nombre qui doit être fonction du nombre de membres déjà invités par chacun des membres fondateurs.

Ensuite, les invitations en possession d'un membre doivent être réparties de manière équilibrée entre ses différents filleuls. De cette manière, un filleul qui s'avèrerait être malveillant ne peut pas contrôler une proportion trop importante des membres du système.

5.2 Représentation arborescente des membres du réseau

Nous proposons de représenter les membres du réseau sous forme d'un arbre n-aire étiqueté appelé *arbre des membres* dont les nœuds sont les membres et les étiquettes le nombre d'invitation non utilisées en possession de chacun de ces membres. Dans la suite, nous proposons également une instanciation efficace de ce modèle.

Définition 1 (Arbre des membres) *L'arbre des membres est un arbre n-aire étiqueté.*
- *Chaque nœud terminal représente un membre n'ayant parrainé personne.*
- *Chaque nœud interne représente un membre ayant déjà parrainé d'autres membres. Les filleuls de ce membre sont les fils de ce nœud.*

L'étiquette d'un nœud est le nombre d'invitation non encore utilisées possédées par le membre représenté par le nœud.

Un réseau pair-à-pair est représenté par son arbre des membres. La racine de cet arbre est le fondateur du réseau pair-à-pair. Si le réseau a été fondé par un ensemble de membres, alors cette racine n'est pas étiquetée et les nœuds de profondeur 1 représentent les membres fondateurs. Notre approche vise à restreindre le nombre de membres que peut parrainer un même membre. Pour cela, il est nécessaire de prendre en compte d'une part le nombre de nœuds effectivement parrainés par un membre et d'autre part le nombre d'invitations restant à ce membre et à ses filleuls. Sur la représentation sous forme d'*arbre des membres*, le nombre de membres effectivement parrainés par un même membre se traduit sous forme de *poids*.

Définition 2 (Poids d'un nœud) *Dans un arbre des membres, le* poids *d'un nœud est noté w et vaut :*
- 1 *si le nœud est terminal ;*
- $w_1 + \ldots + w_n$ *si le nœud est interne et si* $w_1, \ldots, w_n$ *sont les poids des nœuds filleuls.*

De la même manière, nous définissons le *potentiel* d'un nœud comme la somme de son poids et du nombre d'invitations restant à ce membre et à chacun de ses filleuls.

Définition 3 (Potentiel d'un nœud) *Dans un arbre des membres, le* potentiel *p d'un nœud étiqueté par i vaut :*
- $i+1$ *si le nœud est terminal (somme des invitations et du nœud en question) ;*
- $i+1+p_1+\ldots+p_n$ *si le nœud est interne et si* $p_1, \ldots, p_n$ *sont les potentiels des nœuds filleuls.*

À tout instant, la valeur du poids $w_{\mathcal{P}}$ d'un membre $\mathcal{P}$ de l'arbre est inférieure à son potentiel $p_{\mathcal{P}}$.

Dans notre approche, la protection à la *Sybil attack* repose sur un système d'invitations et sur la modération de la distribution de ces invitations. Un membre a le droit d'inviter de nouveaux membres seulement si son poids est strictement inférieur à son potentiel (il reste au moins une invitation à ce membre) et un

membre ne peut recevoir de nouvelles invitations que si son potentiel est conforme à la politique de distribution des invitations. La valeur du potentiel d'un membre est défini par la politique de sécurité de sorte que ce membre ne puisse pas d'inviter un nombre disproportionné de nouveaux membres.

Initialisation du réseau, création des premières invitations :
Supposons que le réseau pair-à-pair soit fondé par un groupe de m membres appelés *membres fondateurs*. L'arbre des membres représentant cette situation est un arbre formé d'un nœud duquel partent m sous arbres de hauteur 1 (figure 5(a)). Nous proposons que les membres fondateurs s'accordent sur le potentiel de chaque membre. De cette façon, les membres fondateurs définissent le nombre d'invitations maximal que chaque membre fondateur est en droit de distribuer dans un premier temps. Supposons que les m membres fondateurs décident que chaque membre fondateur possède autant d'invitations, soit une proportion de $\frac{100}{m}\%$ des membres du réseau, et distribuent i invitations en tout. Cela signifie que le potentiel de chaque membre fondateur est $\frac{i}{m}$ (figure 5(b)). Les poids initiaux sont de 1, puisque personne n'a encore été invité.

Invitation d'un nouveau membre :
Lorsqu'un membre $\mathcal{P}$ parraine un nouveau membre $\mathcal{F}$, il utilise une invitation. $\mathcal{F}$ ne possède initialement aucune invitation : il s'agit d'un nœud terminal étiqueté 0 (figure 5(c)). L'étiquette de $\mathcal{P}$ est décrémentée et son potentiel ne change donc pas. L'invitation d'un nouveau membre n'agit pas sur le potentiel du parrain.

Distribution d'invitations :
Lorsqu'un membre $\mathcal{F}$ n'a pas ou plus d'invitations à accorder, il peut demander à son parrain $\mathcal{P}$ de lui en donner de nouvelles. $\mathcal{P}$ ne donne de nouvelles invitations que si l'ensemble de ses filleuls remplit certaines conditions définies dans sa politique de distribution des invitations, politique qui vise à empêcher la *Sybil attack*. De plus, $\mathcal{P}$ ne peut donner de nouvelles invitations que si il en possède encore lui-même. Dans ce cas, l'étiquette de $\mathcal{F}$ augmente du nombre d'invitations que $\mathcal{P}$ lui délègue, alors que le nombre d'invitations de $\mathcal{P}$ diminue d'autant (figure 5(d)). Le potentiel de $\mathcal{F}$ augmente de la valeur de la transaction tandis que celui de $\mathcal{P}$ reste inchangé. La distribution d'invitations agit sur le potentiel du filleul bénéficiaire mais pas sur celui du parrain.

Si $\mathcal{P}$ ne possède plus d'invitations, il retransmet la demande à son propre parrain qui applique la même démarche. Lorsque cette demande parvient au membre fondateur et que ce membre fondateur n'a lui-même plus d'invitations, ce membre fondateur consulte l'ensemble des membres fondateurs afin de recréer des invitations.

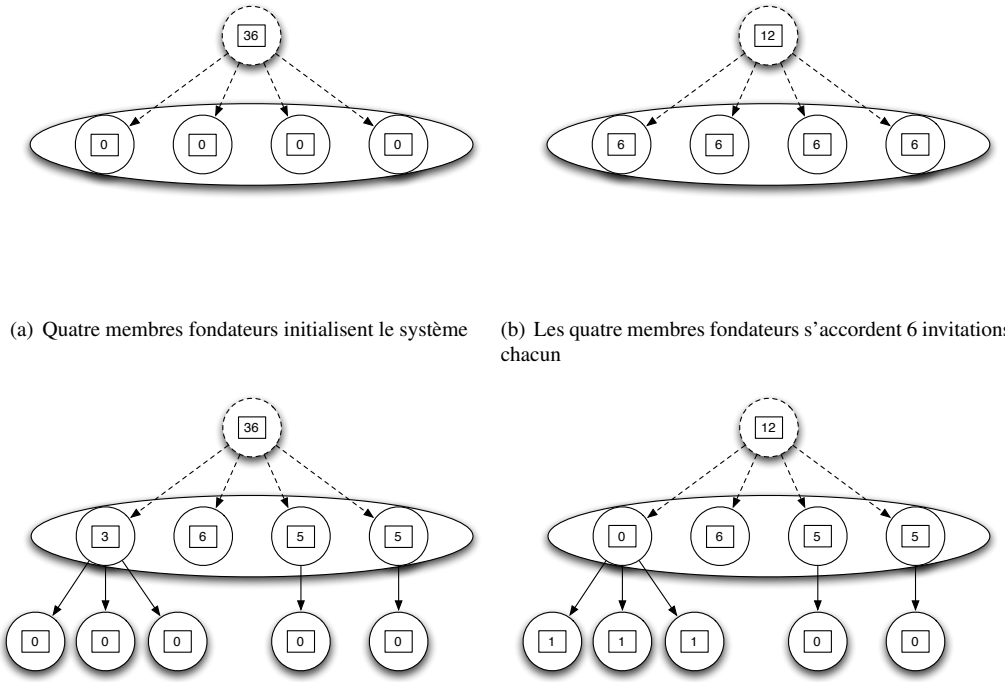
Créations de nouvelles invitations :
Lorsque les membres fondateurs ne disposent plus d'invitations, ils peuvent d'un commun accord créer de nouvelles invitations et les répartir entre eux. Cet accord est lui aussi dicté par la politique de distribution des invitations définie par ces membres, politique qui doit prévenir la *Sybil attack* et qui est décrite précisément dans la section 5.3. À l'issue de cet accord, chacun des membres fondateurs obtient de nouvelles invitations et leur potentiel est augmenté d'autant que le nombre d'invitations reçues. La création de nouvelles invitations agit sur le potentiel des membres fondateurs.

L'ensemble de ces opérations assure que les potentiels sont décroissants en descendant dans l'arbre et donc que plus un membre est bas dans l'arbre, plus son potentiel est faible par rapport aux potentiel des membres hauts.

5.3 Politique de distribution des invitations

La politique de distribution des invitations est définie de manière à limiter l'action des membres qui cherchent à inviter un grand nombre de personnes. De cette manière, nous proposons une défense contre la *Sybil attack*, en se protégeant des personnes *potentiellement* dangereuses. Toutefois, il faut noter qu'un membre exhibant ce comportement n'est pas nécessairement malveillant et, dans ce cas, une personne normale peut être bridée à tort. Dans l'approche que nous proposons, le réseau pair-à-pair grandit sous le contrôle de la politique de distribution des invitations. Le bénéfice est que le réseau pair-à-pair est protégé contre la *Sybil attack*, mais en contrepartie l'accès au réseau est plus difficile.

Afin de prévenir la *Sybil attack*, les politiques de distribution des invitations doivent respecter deux contraintes. Tout d'abord, il faut empêcher un nœud malveillant d'avoir un potentiel supérieur aux autres nœuds : le parrain doit donc éviter de donner des invitations qui provoqueraient des déséquilibres trop importants entre ses filleuls. Ensuite, la distribution des invitations doit être progressive. En effet, une personne



(a) Quatre membres fondateurs initialisent le système

(b) Les quatre membres fondateurs s'accordent 6 invitations chacun

(c) De nouvelles personnes sont insérées

(d) Un membre fondateur distribue une invitation à chacun de ses filleuls

FIG. 5: Fonctionnement du modèle arborescent. Chaque nœud est étiqueté par le nombre d'invitations possédées

malveillante utilisera toutes les invitations disponibles le plus tôt possible, afin de faire croître le pourcentage des identifiants qu'elle contrôle ; à l'inverse, une personne normale utilisera ses invitations au fur et à mesure.

De façon générale, la politique de distribution doit permettre de décider quand les poids des filleuls d'un même membre sont suffisamment équilibrés pour redistribuer des invitations. Par exemple, sur la figure 6, l'arbre des membres est déséquilibré, P_{00} a invité beaucoup plus de membres que P_1, P_2 ou P_3. Par contre, sur la figure 7, l'arbre des membres est parfaitement équilibré.

Formellement, un arbre des membres est parfaitement équilibré si pour chaque membre dans l'arbre, le poids est égal au potentiel. Intuitivement, cette situation se produit lorsque toutes les invitations ont été utilisées et qu'à chaque niveau de profondeur de l'arbre, chaque membre a invité autant de filleuls que ses frères.

L'équilibre parfait est une situation difficile à atteindre. Afin de faciliter la distribution de nouvelles invitations et donc l'insertion de nouveaux membres, nous proposons d'autoriser certains déséquilibres spécifiques. Ainsi, chaque parrain attribue un facteur f à chacun de ses filleuls, représentant la proportion d'invitations qu'il doit recevoir par rapport aux autres filleuls (la somme des facteurs des filleuls d'un même membre vaut 1) et nous définissons la règle de distribution des invitations suivante.

Règle de distribution des invitations :
Un parrain $\mathcal{P}$ possédant j filleuls $\mathcal{F}_1, \ldots, \mathcal{F}_j$ peut donner k nouvelles invitations à l'un de ses filleuls, k faible et défini par la politique, lorsque les deux conditions suivantes sont remplies :

1. pour ce filleul, le poids est égal au potentiel ;
2. la variance des rapports $\frac{w_{\mathcal{F}_j}}{f_{\mathcal{F}_j} \cdot (w_{\mathcal{P}} - 1)}$ est inférieure ou égale à une borne $\mathcal{V}$ fixée par la politique.

L'équilibre parfait est atteint lorsque la variance des rapports $\frac{w_{\mathcal{F}_j}}{f_{\mathcal{F}_j} \cdot (w_{\mathcal{P}} - 1)}$ vaut 0.

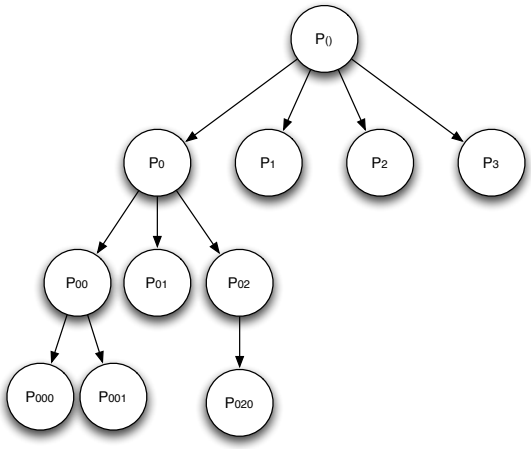


FIG. 6: Arbre d'identification déséquilibré.

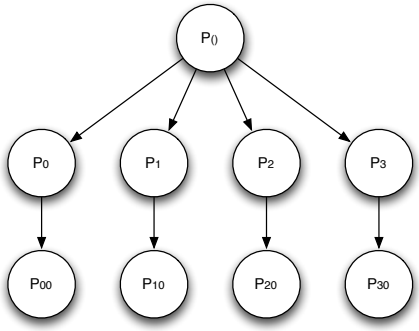


FIG. 7: Arbre d'identification équilibré.

5.4 Instanciation du modèle

Nous proposons une instanciation compacte et efficace de ce modèle par des moyens cryptographiques. Chaque personne p est munie d'un couple de clés publique/privée (P_p, S_p). Soit (P_r, S_r) le couple de clés du nœud racine, P_r étant connue de tous les membres. Si le groupe est fondé par une unique personne, alors (P_r, S_r) est le couple de clés de cette personne. Si le groupe est fondé par un ensemble de membres fondateurs, alors (P_r, S_r) est fragmentée sur l'ensemble des membres fondateurs en utilisant de la cryptographie à seuil. La cryptographie à seuil est basée sur le partage de secret de Shamir [Sha79], et nous proposons d'utiliser ici la solution de [Rab98]. Le principe est de répartir une clé secrète S_r (de type RSA dans ce cas) sur n participants, parmi lesquels n'importe quels k ($k \leq n$) suffisent à chiffrer un message avec la clé S_r. Les propriétés intéressantes sont qu'aucun des membres ne connaît à aucun instant la clé S_r en entier, k membres suffisent à chiffrer un message avec S_r et la connaissance de $k-1$ fragments n'apporte aucune information sur S_r. L'ensemble des membres fondateurs peut ainsi émuler le nœud racine par l'accord de k d'entre eux.

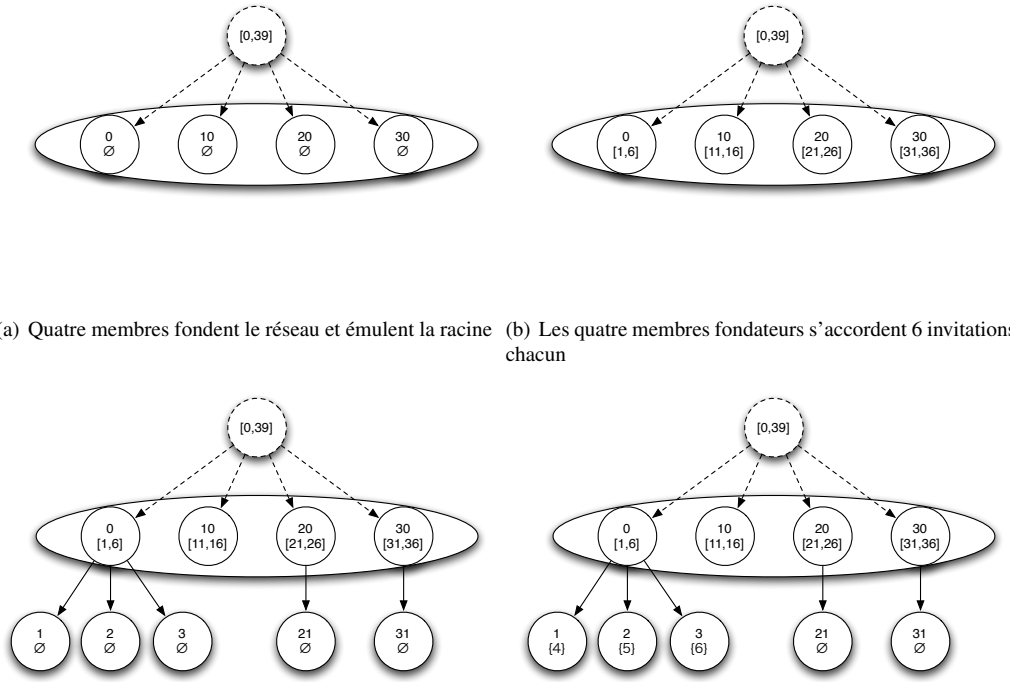
La preuve d'appartenance de p au système d'identification est une chaîne de certificats (chemin de confiance) allant de la racine de l'arbre à p. Chaque certificat, signé par le parrain, contient l'identifiant Id du filleul (un nombre entier), le haché de la clé publique de ce filleul ainsi éventuellement que des invitations, sous la forme d'un ensemble d'identifiants à distribuer. Les identifiants étant de simples entiers, n invitations correspondent à n nombres entiers. Afin de rendre la représentation des invitations compacte, elles sont distribuées de manière contiguë, permettant de représenter n invitations par l'intervalle $[id_1, id_n]$.

Pour vérifier la validité d'une identité, il faut remonter le chemin de confiance, en vérifiant que les signatures de certificat sont valides et que les identifiants et les ensembles d'invitations des filleuls appartiennent bien aux ensembles d'invitations des parrains. Chaque personne doit donc conserver la chaîne de confiance entre la racine de l'arbre et lui-même afin de prouver son appartenance au système d'identification.

Pour distribuer de nouvelles invitations, il suffit au parrain de mettre à jour les ensembles d'identifiants à distribuer de ses filleuls. Afin de conserver la représentation efficace des invitations, le parrain doit optimiser les identifiants qu'il distribue afin de fournir des ensembles contigus.

5.5 Résistance à des personnes malveillantes

Le système présenté est résistant à la *Sybil attack*. En effet, une personne malveillante ne peut pas certifier un grand nombre d'identités, puisqu'elle est limitée par le taux de certification des autres filleuls. Chaque membre est responsable de maintenir l'équilibre entre ses filleuls et chaque membre normal permet de réduire le potentiel d'un nœud malveillant inséré en dessous de lui. En effet, dans le cadre d'un arbre équilibré, un membre invitant équitablement n filleuls divise son potentiel en n parties. Si un de ses filleuls



(a) Quatre membres fondent le réseau et émulent la racine

(b) Les quatre membres fondateurs s'accordent 6 invitations chacun

(c) De nouvelles personnes sont insérées

(d) Le membre d'identifiant 0 délègue 1 invitation à chacun de ses filleuls

FIG. 8: Instanciation du modèle. Chaque nœud possède un identifiant et un ensemble d'invitations

est malveillant alors son potentiel est n fois plus petit que celui de son parrain. En quelques étages de l'arbre, l'insertion d'un membre malveillant devient négligeable. La seule solution pour créer un très grand nombre d'identités est de trouver un très grand nombre de parrains réels, ce qui ne paraît pas réaliste. En effet, un parrain étant une personne faisant confiance à l'invité, chaque invité ne peut disposer que d'un nombre limité de parrains.

En revanche, les personnes présentes en haut de l'arbre ont un rôle critique. Si l'une de ces personnes devient malveillante ou est victime d'une attaque, un attaquant peut obtenir un rang très haut dans l'arbre et éventuellement générer un très grand nombre d'identités. Par conséquent, plus il y a de membres fondateurs de l'arbre, plus le système est robuste aux attaques. Ce paramètre doit donc être pris en compte à la création du réseau.

Un attaquant peut également tenter de bloquer le système. En effet, si il n'invite personne, son parrain risque de juger l'ensemble de ses filleuls déséquilibrés et de ne pas distribuer de nouvelles invitations. La politique de distribution des invitations du parrain doit être robuste à ce type d'attaque. Une première possibilité consiste en la modification manuelle des potentiels des filleuls, suite au constat par le parrain du dysfonctionnement. Une seconde possibilité est l'utilisation de règles de distribution naturellement robustes, par des moyens statistiques, afin d'écarter les personnes bloquantes des calculs.

Enfin, il est intéressant de noter que pour qu'un parrain $\mathcal{P}$ obtienne le nombre d'invitations utilisées par chacun de ses filleuls $\mathcal{F}_x$ (correspondant au poids du filleul), qui est différent du nombre d'invitations qu'il leur a transmis (correspondant au potentiel), il lui suffit de le leur demander. En effet, même si un filleul peut tout à fait mentir en disant à tort qu'il n'a pas utilisé d'invitation (cela correspond à un cas de blocage), cela est géré par l'arbre n-aire. Un filleul peut également répondre à tort qu'il a utilisé toutes les invitations : cela n'est pas gênant non plus puisque s'il est malveillant, il peut utiliser ces invitations légalement. Dans tous les cas, le mensonge d'un filleul ne pose pas de problème.

5.6 Adaptation aux réseaux pair-à-pair

Le système proposé est adapté à être utilisé au sein d'un réseau pair-à-pair, et plus particulièrement encore dans un réseau pair-à-pair structuré. Dans un premier temps, la limitation du nombre d'identifiants d'une même personne permet de garantir que le taux de nœuds malveillants présents dans le système reste minoritaire, en supposant que la plupart des personnes se comportent bien. Cela assure la qualité globale du réseau.

Dans un deuxième temps, les identifiants dans un réseau pair-à-pair structuré doivent être répartis de manière uniforme, afin de répartir la charge du réseau équitablement. De plus, les identifiants proches sont responsables des mêmes données du réseau, il faut donc également éloigner dans le réseau virtuel pair-à-pair les personnes proches dans l'arbre pour éviter les défaillances corrélées (malveillantes ou non). Ces deux propriétés sont assurées si l'on choisit comme identifiant $h(Id)$, h étant une fonction de hachage et Id l'identifiant du membre dans l'arbre.

Enfin, un groupe de personnes malveillantes peut également essayer d'obtenir des identifiants spécifiques afin de contrôler certaines ressources ou de filtrer toutes les requêtes d'un nœud. Avec l'identifiant choisi précédemment, ce choix est impossible puisqu'une personne n'a pas de contrôle sur son identifiant. Le système proposé permet donc de prévenir la *Sybil attack* dans les réseaux pair-à-pair structurés.

Nous pensons que ce système n'empêche personne de joindre le réseau. Un réseau pair-à-pair se développant progressivement par un phénomène de notoriété, une personne voulant joindre le réseau a toujours la possibilité de connaître un groupe de personnes appartenant déjà au réseau. Parmi ce groupe, il est très probable de trouver un parrain possédant une invitation. De plus, la notion d'invitation est perçue comme un privilège et peut donc même participer au développement plus rapide du réseau pair-à-pair (ce qui est perçu comme rare est cher).

Il faut noter que la création d'un arbre des membres introduit une certaine centralisation dans les réseaux pair-à-pair, puisque les membres fondateurs reçoivent implicitement la confiance de tous les utilisateurs. Cependant, Cheng *et al.* ont prouvé dans [CF05] qu'il existe nécessairement un sous-ensemble de nœuds en lesquels tout le monde doit accorder sa confiance pour qu'un système soit résistant à la *Sybil attack* : dans notre système, ce sous-ensemble est le groupe des membres fondateurs.

6 Conclusion et perspectives

Nous avons proposé un mécanisme distribué permettant l'identification pseudo-unique des personnes, notamment dans le cadre des réseaux pair-à-pair structurés. Ce mécanisme est basé sur une certification arborescente des identités. Un ensemble de personnes responsables du système d'identification est situé tout en haut de l'arbre, chacune de ces personnes pouvant inviter d'autres personnes. Afin de prévenir la *Sybil attack*, l'arbre est créé de façon équilibrée, empêchant une personne malveillante de générer un grand nombre d'identités ou d'en choisir une.

La création de l'arbre est régulée par la délivrance d'invitations. Nous avons donc également proposé une instanciation du modèle permettant de matérialiser de manière compacte un très grand nombre d'invitations.

Enfin, nous avons précisé comment cet arbre pouvait être utilisé au sein des réseaux pair-à-pair structurés.

Ce système est externe au réseau pair-à-pair utilisé. Il contient donc l'ensemble de toutes les personnes *pouvant* utiliser le réseau et non pas de toutes les personnes utilisant le réseau *à un instant précis*. Ceci implique que si il y a *n%* de personnes malveillantes dans l'arbre et que toutes ces personnes sont connectées au réseau pair-à-pair alors le il y a *plus* de *n%* des nœuds du réseau pair-à-pair qui sont malveillants. Il sera donc intéressant d'étudier un système similaire *online*, assurant le même taux de personnes malveillantes dans le système d'identification que dans le réseau pair-à-pair.

Références

[BBK94] T. Beth, M. Borcherding, and B. Klein. Valuation of trust in open networks. In Dieter Gollmann, editor, *Proceedings of the 3rd European Symposium on Research in Computer Security (ESORICS)*, volume 875 of *Lecture Notes in Computer Science*, pages 3–18. Springer-Verlag, 1994.

[Bor06] Nikita Borisov. Computational puzzles as sybil defenses. In *Proceedings of the 6th IEEE International Conference on Peer-to-Peer Computing (P2P)*, volume 0, pages 171–176. IEEE Computer Society, 2006.

[CBH03] Srdjan Capkun, Levente Buttyán, and Jean-Pierre Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(1) :52–64, 2003.

[CCR04] Miguel Castro, Manuel Costa, and Antony Rowstron. Peer-to-peer overlays : structured, unstructured, or both ? Technical Report MSR-TR-2004-73, Microsoft Research (MSR), 2004.

[CDG+02] Miguel Castro, Peter Druschel, Ayalvadi J. Ganesh, Antony I. T. Rowstron, and Dan S. Wallach. Secure routing for structured peer-to-peer overlay networks. In *Proceedings of the 5th ACM Symposium on Operating System Design and Implementation (OSDI)*, Operating Systems Review, pages 299–314. ACM Press, 2002.

[CF05] Alice Cheng and Eric Friedman. Sybilproof reputation mechanisms. In *Proceeding of the ACM SIGCOMM workshop on Economics of peer-to-peer systems (P2PECON)*, pages 128–132, New York, NY, USA, 2005. ACM Press.

[Cli00] Clip2. The gnutella protocol specification v0.4. `http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf`, 2000.

[Dou02] John R. Douceur. The sybil attack. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS)*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer-Verlag, 2002.

[eBa] eBay. eBay : What is feedback ? `http://pages.ebay.com/help/feedback/questions/feedback.html`.

[Goo] Google. Gmail. `http://gmail.google.com`.

[HBMS04] Oliver Heckmann, Axel Bock, Andreas Mauthe, and Ralf Steinmetz. The eDonkey File-Sharing Network. In Peter Dadam and Manfred Reichert, editors, *Proceedings of the Workshop on Algorithms and Protocols for Efficient Peer-to-Peer Applications (Informatik)*, volume 51 of *LNI*, pages 224–228. GI, 2004.

[Int93] International Telecommunication Union. The directory — authentication framework. ITU-T Recommendation X.509, 1993.

[Mau96] Ueli M. Maurer. Modelling a public-key infrastructure. In Elisa Bertino, Helmut Kurth, Giancarlo Martella, and Emilio Montolivo, editors, *Proceedings of the 4th European Symposium on Research in Computer Security (ESORICS)*, volume 1146 of *Lecture Notes in Computer Science*, pages 325–350. Springer-Verlag, 1996.

[Mil67] S. Milgram. The small world problem. *Psychology Today*, 1(1) :60–67, 1967.

[Moc87] P. Mockapetris. Domain names - concepts and facilities. Internet Request for Comment RFC 1034, Internet Engineering Task Force, 1987.

[Rab98] Tal Rabin. A simplified approach to threshold and proactive RSA. In Hugo Krawczyk, editor, *Proceedings of the 18th Annual International Cryptology Conference (Crypto)*, volume 1462 of *Lecture Notes in Computer Science*, pages 89–104. Springer-Verlag, 1998.

[RD01] Antony I. T. Rowstron and Peter Druschel. Pastry : scalable, decentralized object location and routing for large-scale peer-to-peer systems. In Rachid Guerraoui, editor, *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, volume 2218 of *Lecture Notes in Computer Science*, pages 329–350. Springer-Verlag, 2001.

[RFH+01] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard M. Karp, and Scott Shenker. A scalable content-addressable network. In Roch Guerin, editor, *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*, volume 31, 4 of *Computer Communication Review*, pages 161–172. ACM Press, 2001.

[RS98] Michael K. Reiter and Stuart G. Stubblebine. Resilient authentication using path independence. *IEEE Transactions on Computers*, 47(12) :1351–1362, 1998.

[RS99] Michael K. Reiter and Stuart G. Stubblebine. Authentication metric analysis and design. *ACM Transactions on Information and System Security*, 2(2) :138–158, 1999.

[Sha79] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11), 1979.

[SMK+01] Ion Stoica, Robert Morris, David R. Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord : A scalable peer-to-peer lookup service for internet applications. In Roch Guerin, editor, *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*, Computer Communication Review, pages 149–160. ACM Press, 2001.

[TCG05] Trusted Computing Group. TPM main specification. Main Specification Version 1.2 rev. 85, Trusted Computing Group, 2005.

[Wal02] Dan S. Wallach. A survey of peer-to-peer security issues. In Mitsuhiro Okada, Benjamin C. Pierce, Andre Scedrov, Hideyuki Tokuda, and Akinori Yonezawa, editors, *Proceedings of the International Symposium on Software Security (ISSS)*, volume 2609 of *Lecture Notes in Computer Science*, pages 42–57. Springer-Verlag, 2002.

[Zim95] Philip R. Zimmermann. *The Official PGP User's Guide*. MIT Press, pub-MIT :adr, 1995.

[ZKJ01] Ben Y. Zhao, John Kubiatowicz, and Anthony D. Joseph. Tapestry : an infrastructure for fault-resilient wide-area location and routing. Technical Report UCB//CSD-01-1141, University of California at Berkeley, apr 2001.