République Algérienne démocratique et populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderahmane Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de Fin de Cycle

Présenté par :

KERKAR Amina

En vue de l'obtention du diplôme de Master en Informatique

Option : Réseaux et systèmes distribués

Thème :

La confiance dans le partage pair-à-pair dans les réseaux sociaux mobiles

Devant le jury composé de :

Présidente	YAICI Malika	Maitre assistant (UAMB , Béjaia)
Encadrant	CHALLAL Yacine	Maitre de conférence (UTC Compiègne, France)
Encadrant	OMAR Mawloud	Maitre de conférence classe B (UAMB, Bejaia)
Examinateur	FARAH Zoubeyr	Maitre assistant (UAMB, Béjaia)
Examinateur	SAADI Mustapha	Maitre assistant (UAMB, Béjaia)

Promotion 2012

Table des matières

Introduction générale **2**

1 Introduction aux réseaux Pair-à-Pair **5**

Introduction 5

1.1 Historique 6

1.2 L'architecture des systèmes P2P 7

1.2.1 Architecture purement décentralisée 7

1.2.2 Architecture partiellement décentralisée 8

1.2.3 Architecture hybride 8

1.3 La structure du réseau 9

1.3.1 Les réseaux structurés 9

1.3.2 Les réseaux non structurés 10

1.4 Les caractéristiques du partage P2P 10

1.4.1 L'integrité des données 10

1.4.2 La confidentialité 11

1.4.3 La disponibilité 11

1.4.4 L'autonomie des noeuds 12

1.4.5 La scalabilité 12
1.4.6 Les performances 12
1.4.7 La décentralisation 13
1.4.8 L'équité 13
1.5 Les services des systèmes P2P 14
1.5.1 La responsabilité et réputation 15
1.5.2 Le transport de données 16
1.5.3 La gestion de données 17
1.5.4 La gestion des ressources 18
1.5.5 L'identification et l'authentification 20
1.6 Cas d'étude 20
1.6.1 Chord 20
1.6.2 BitTorrent 24
Conclusion 25

2 Notions élémentaires sur la confiance et la cryptogrpahie 27

Introduction 27
2.1 La confiance 28
2.2 Les thechniques de cryptographie 29
2.2.1 La cryptographie á sens unique 30
2.2.2 La cryptographie symétrique 31
2.2.3 La cryptographie asymétrique 32
2.3 La cryptographie d'identité 33
2.4 La cryptographie a seuil 33
2.4.1 Le protocole de partage du secret 34

2.4.2 Le protocole de reconstruction du secret 34
2.5 PKI (Public Key Infrastructure) 34
2.5.1 Les cértificats 35
2.5.2 Les annuaires de certificats 35
2.5.3 L'organisation des autorités de certifications 35
Conclusion 37

3 Les differentes attaques dans les réseaux Pair-à-Pair 39

Introduction 39
3.1 Les attaques d'infrastructure 41
3.1.1 L'attaques de déni de service (DoS) 41
3.1.2 Les Botnets 44
3.1.3 La propagation des vers 46
3.1.4 Les attaques sybil 49
3.1.5 Les attaques éclipse 51
3.2 Les attaques de données 52
3.2.1 L'empoisonnement des fichiers 53
3.2.2 Le Free-Riding 54
Conclusion 56

4 Contribution 58

Introduction 58
4.1 Motivations 59
4.2 L'architecture du modèle 60
4.2.1 La dscription du modèle 60
4.2.2 La création du groupe 62

4.2.3 L'adhésion d'un utilisateur 63

4.2.4 L'accusation d'un utilisateur 66

Conclusion 70

5 Le comportement de notre modèle vis à vis des attaques 71

Introduction 71

5.1 Empoisonnement des fichier 72

5.2 Le Free Riding 73

5.3 L'attaques déni de service 74

5.4 L'attaque sybil 75

5.5 La propagation des vers 75

5.6 L'attaque eclipse 75

Conclusion 76

Conclusion générale 77

Bibliographie 80

Table des figures

1.1 Principe de fonctionnement d'un P2P purement décentralisé [39] 7

1.2 Principe de fonctionnement d'un P2P partiellement décentralisé [39] . . . 8

1.3 Principe de fonctionnement d'un P2P hybride [39] 9

1.4 Proprietés et services des systèmes P2P 14

1.5 Répartition des clés 22

1.6 Exemple de recherche dans Chord 24

1.7 Structure générale de BitTorrent 24

2.1 Chiffrement symétrique 31

2.2 Chiffrement asymétrique 32

2.3 Modèle hierarchique "$x \rightarrow y$" :x signe le cértificat de y "AC" : Autorité de Certification 36

2.4 Modèle croisé $''x \rightarrow y''$:x signe le cértificat de y "AC" : Autorité de Certification 37

2.5 Modèle anarchique "$x \rightarrow y$" :x signe le cértificat de y 37

4.1 Relations de confiance 59

4.2 Aperçu du modèle 61

4.3 Graphe de confiance partielle 62

4.4 Phase initiale 63

4.5 Cas d'utilisation du processus d'adhésion 64

4.6 Adhésion d'un utilisateur 65

4.7 Cas d'utilisation du processus d'exclusion 67

4.8 Exclusion d'un utilisateur 68

5.1 Echange de fichier P2P 72

5.2 Exemple de stabilité d'un partage 73

Liste des tableaux

3.1 Différentes méthodes d'attaque DDoS dans BitTorrent 42

3.2 Impacte des attaques P2P 56

5.1 Comportement du modèle vis-à-vis des attaques 76

Introduction générale

Aujoud'hui, les environnements de calcul et de communication sur Internet sont sensiblement plus complexes et divers que les systèmes répartis classiques, manquant de toute organisation centralisée ou commande hiérarchique. Ceci ne constitue qu'une simple face des protocoles et des utilisations du pair-à-pair, qui proposent une manière forte et équitable de collaborer en vue d'accroître le potentiel du réseau. Pour ce fait, on s'aperçoit que les architectures centralisées commencent à atteindre leurs limites pour le partage de fichiers de tailles importantes et le pair-à-pair a vu le jour.

Un réseau pair-à-pair ("P2P" ou "Pair-à-Pair") est une architecture où les machines connectées communiquent directement entre elles et partagent leurs ressources sans faire intervenir une entité centrale. Les réseaux P2P représentent aujourd'hui une partie considérable des échanges sur Internet, principalement parce qu'ils offrent aux utilisateurs du monde entier un moyen rapide et efficace pour partager des ressources (contenu, puissance de calcul, etc.). Les échanges de données sur Internet sont aujourd'hui tellement gigantesques que les modèles client/serveur classiques atteignent leurs limites malgré l'accroissement considérable de leurs puissances. Les serveurs ont de plus en plus de difficultés

à répondre efficacement à toutes les exigences de leurs clients. Dans le modèle P2P, chacune des machines connectées au réseau possède des responsabilités égales envers les autres machines du même réseau.

Les applications P2P peuvent être confrontées à différents types d'attaques qui cherchent à nuire à leur fonctionnement. Les systèmes P2P sont vulnérables aux traditionnelles attaques comme le déni de service distribué, la propagation des vers informatiques, les botnets, etc. En outre, les systèmes P2P sont particulièrement vulnérables à d'autres attaques spécifiques du fait de leur fonctionnement qui s'appuie sur la collaboration et la coopération des pairs. L'abus de confiance entre pairs peut poser de sérieuses menaces de sécurité sur un système P2P.

Le travail effectué dans ce mémoire traite l'un des axes majeurs qui est la sécurité des réseaux P2P, en particulier la confiance. En effet, tout développement de services de sécurité doit s'appuyer sur un modèle de confiance à la fois robuste et hautement disponible dans un milieu P2P ce qui représente notre objectif. Cependant, assurer la confiance dans une telle architecture est un véritable défi en raison de sa décentralisation et de l'absence d'une topologie fixe. Pour cela nous avons opté pour l'utilisation de la certification à seuil afin de réaliser ce travail.

Ce présent mémoire est organisé en cinq chapitres :

Le premier chapitre présente les différentes notions du système P2P d'une manière générale suivi de quelques exemples de protocoles P2P qui sont Chord et BitTorrent.

Le second chapitre introduit la confiance et les différents concepts de cryptographie moderne et les techniques existantes et l'infrastructure à clés publiques.

Le troisième chapitre est un état de l'art des différentes attaques qui existent dans le système P2P et les contre mesures.

Le quatrième chapitre est dédié à notre proposition où est présentée l'architecture de notre modèle de confiance.

Le cinquième chapitre est l'analyse de notre modèle vis-à-vis des attaques citées dans le troisième chapitre.

Nous terminerons ce mémoire par une conclusion générale et des perspectives.

Chapitre

1 Introduction aux réseaux Pair-à-Pair

Sommaire

Introduction 5

1.1 Historique 6

1.2 L'architecture des systèmes P2P 7

1.3 La structure du réseau 9

1.4 Les caractéristiques du partage P2P 10

1.5 Les services des systèmes P2P 14

1.6 Cas d'étude 20

Conclusion 25

Les réseaux Pair-à-Pair , comme eMule , BitTorrent , Skype et plusieurs autres systèmes ont connu un grand succès durant ces dernières années auprès des internautes. Ceci parce qu'ils offrent aux utilisateurs le partage gratuit et illimité des ressources. Toutefois, ces systèmes représentent un changement de paradigme du modèle client/serveur où le rôle de fournisseur de service et les consommateurs sont clairement distinguées et séparément attribués aux acteurs du système. Les réseaux P2P sont des systèmes distribués sans

aucune organisation hiérarchique ou de contrôle centralisé. Chaque pair s'auto-organise dans les réseaux qui opèrent sur internet et offre diverses fonctionnalités telles que les architectures pour un routage robuste de large zone, la sélection des pairs à proximité, la confiance et l'authentification, l'anonymat... etc. Cependant ces systèmes offrent de nombreux avantages en comparaison avec le système client/serveur telles que la scalabilité, la tolérance au pannes et les performances.

1.1 Historique

Tout a commencé en fin 1998, **Shawn Fanning**, un étudiant américain passionné d'informatique alors âgé de 19 ans vient bouleverser le monde bien établi du client/serveur. Il décide de quitter l'université et se lance dans l'écriture d'un logiciel pour permettre l'échange de fichiers musicaux. La raison d'être de ce logiciel repose sur le constat suivant : rechercher des MP3 sur les moteurs de recherche habituelle conduit à une perte de temps énorme et les réponses sont souvent inappropriées. Après quelques mois de travail acharné, une première version du logiciel est disponible. Fanning décide de tester une première version le 1er juin 1999 et appelle son logiciel **Napster**. Le logiciel qui ne devait être testé que par quelques-uns de ses amis remporte un succès des plus rapides. Il conquiert notamment les universités. Shawn Fanning se retrouve propulsé à la tête d'un star-up plein d'avenir. Lui qui déclare à propos de Napster qu'il n'avait "aucune envie d'en faire un business" voit les utilisateurs arrivés en masse. En moins d'un an, le nombre d'utilisateurs atteignit la barre des 50 millions. Cette fréquentation fut perçue comme un risque pour l'économie de la musique et la RIAA (Recording Industry Association of America) attaqua le site en justice. Il dut fermer en 2001.[59]

1.2 L'architecture des systèmes P2P

Le fonctionnement de tout système de partage de fichier P2P repose sur un ensemble de nœud et les connexions entre eux. Ce réseau est formé au dessus du réseau informatique physique et donc désigné comme un "recouvrement" du réseau. Le fonctionnement du réseau de recouvrement et sa performance sont réalisés par sa topologie, la structure et le degré de centralisation. Les réseaux de recouvrement peuvent être distingués et classés en fonction de leur centralisation et de leur structure.

1.2.1 Architecture purement décentralisée

Dans une architecture purement décentralisée, chaque nœud du réseau est un serveur et un client en même temps, il n'y a pas de coordination centrale de leurs activités. Gnutella[37] et Freenet[24] sont des instances de ce système. Les systèmes P2P pures sont évolutifs, tolérants aux pannes et ont un plus grand degré de contrôle d'autonomie sur leurs données et leurs ressources. D'autre part, ces systèmes présentent une découverte lente de l'information et il n'y a aucune garantie sur la qualité des services, en raison du manque de vision globale au niveau du système.

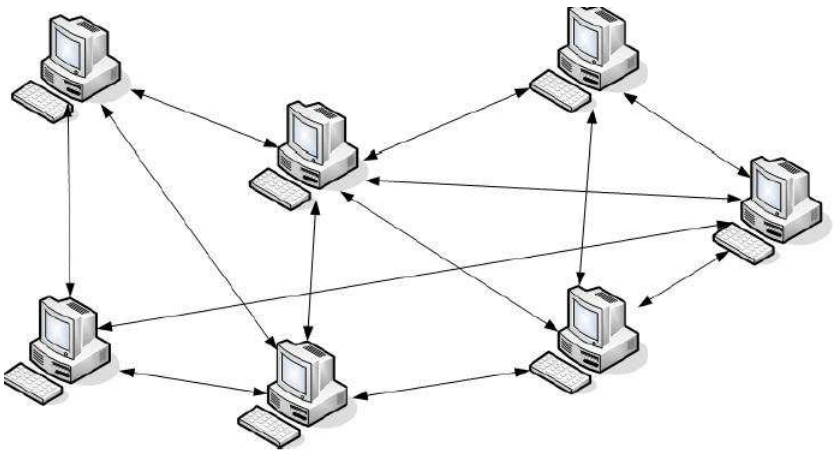

FIGURE 1.1 – Principe de fonctionnement d'un P2P purement décentralisé [39]

1.2.2 Architecture partiellement décentralisée

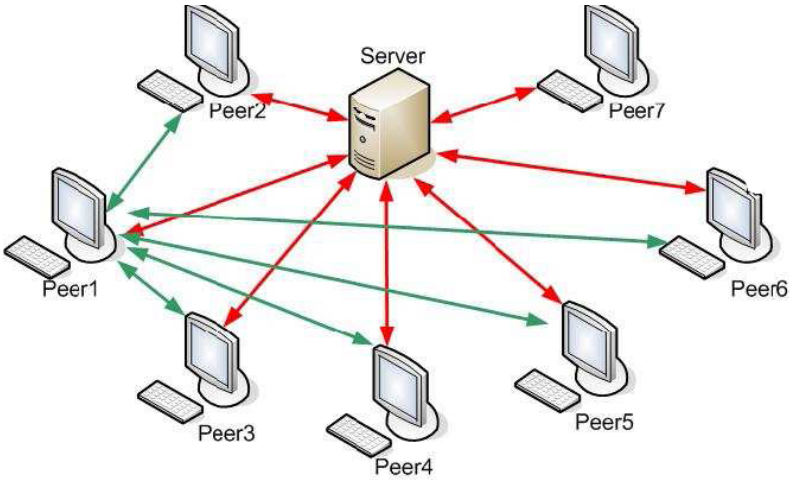

FIGURE 1.2 – Principe de fonctionnement d'un P2P partiellement décentralisé [39]

Le principe est similaire aux systèmes purement décentralisées. Toutefois, certains nœuds appelés super-nœuds, ont un rôle plus important. Ils agissent comme des indices locaux centraux pour les fichiers partagés par les pairs voisins. La façon dont sont élus super-nœuds varie d'un système à un autre. Il est important de noter que les super-nœuds sont des points de défaillances uniques pour un réseau P2P et sont affectés dynamiquement, s'ils échouent, le réseau prendra automatiquement des mesures pour les remplacer par d'autres. FastTrack [24] et Gnutella2 [51] sont des exemples de ces systèmes.

1.2.3 Architecture hybride

Dans ces systèmes, il ya un serveur central qui facilite l'interaction entre les pairs grâce à des répertoires de métadonnées, décrivant les fichiers partagés stockés par les nœuds. Bien que les échanges de fichiers peuvent directement avoir lieu entre deux noeuds,les serveurs centraux permettent de faciliter cette interaction en effectuant les recherches et l'identification des noeuds de stockage des fichiers. Cela les rend généralement plus vulnérables aux attaques malveillantes. eDonkey [1], et bitTorrent [10] sont des exemples

de ces systèmes.

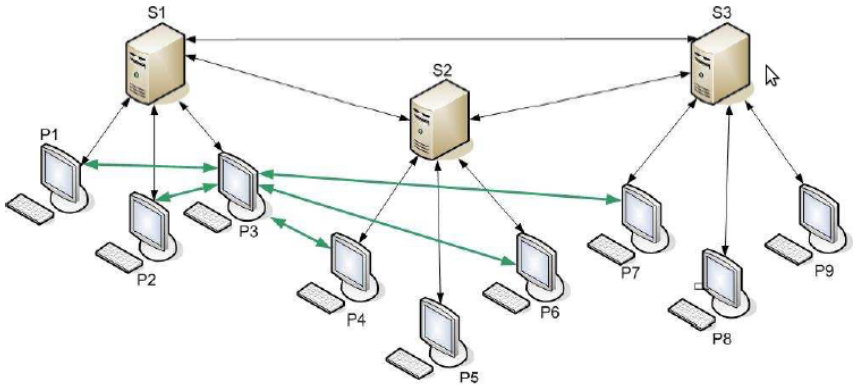

FIGURE 1.3 – Principe de fonctionnement d'un P2P hybride [39]

1.3 La structure du réseau

Le partage de fichiers dans un réseau de recouvrement peut être lié à un emplacement spécifique en utilisant une certaine règle déterministe ou non. Dans le premier cas, le réseau de recouvrement est dit structuré, autrement il est non structuré.

1.3.1 Les réseaux structurés

Les P2P structurés sont basés sur l'établissement d'une DHT (Distributed Hash Table) permettant de "placer" les nouveaux noeuds aux seins du réseau. Chaque noeud reçoit une liste de voisins avec lesquels il pourra communiquer. Il s'agit ici de voisins "logiques", ceux-ci pouvant se trouver à l'autre bout du monde. De plus, chaque pair est responsable d'une partie spécifique du contenu du réseau. Ils sont en général identifiés par un couple clé/valeur qui permet de réaliser des recherches avec un nombre de messages croissant de façon logarithmique, contrairement à certains algorithmes proposant des techniques de "flood [1]".Parmi les algorithmes structurés, on peut citer : CAN, Chord, Tapestry, Pastry,

1. (inondation en français) Cette technique permet de propager la requête de proche en proche, chaque pair la transmettant à ces voisins. Il y a donc des redondances, les pairs pouvant avoir des voisins similaires.

Kademlia (utilisé par Overnet), Viceroy et Freenet.[39]

1.3.2 Les réseaux non structurés

Un P2P est non structuré quand les liens entre les noeuds sont établis de façon arbitraire. La communication entre les noeuds est donc plus difficile à gérer. Parmi ces P2P, on peut citer : Gnutella ,Fastrack dont le client le plus connu est KaZaA, BitTorrent, eDonkey (utilisé par le logiciel eMule).[39]

1.4 Les caractéristiques du partage P2P

Les systèmes P2P possèdent des caractéristiques avantageuses par rapport aux autres systèmes basés sur le paradigme Client/Serveur. Pour cette raison, ils sont très populaires, chaque système P2P essaye de posséder un grand nombre de ces caractéristiques [23].

1.4.1 L'integrité des données

Dans un système de partage de fichiers P2P, les pairs qui téléchargent doivent être en mesure de verifier que le fichier téléchargé n'a pas été modifié par une tierce partie. En d'autres termes, chaque système P2P doit mettre en œuvre un mécanisme qui assure l'exactitude et l'exhaustivité des données et des méthodes de traitement. Des entités non autorisées ne peuvent pas modifier les données. Pour ce faire, certains systèmes P2P utilisent des fonctions de hachage qui permettent d'avoir un condensé du fichier. Sachant que chaque condensé est unique, les utilisateurs peuvent vérifier l'intégrité du fichier à télécharger. Cependant BitTorrent utilise l'algorithme de hachage SHA-1 tandis que eMule utilise l'algorithme MDA-5.

1.4.2 La confidentialité

En générale, la confidentialité garantit que les données ne sont accessibles que pour ceux qui sont autorisés à avoir accès à elles. Cependant elle est étroitement associée à l'anonymat de l'utilisateur qui est une caractéristique principale du partage P2P des fichiers, son objectif est d'éviter la violation de la vie privée des utilisateurs. Il y a plusieurs systèmes P2P qui offrent une sorte d'anonymat. Par exemple, Freenet permet l'anonymat des utilisateurs en rendant difficile pour un opérateur de :

1. Découvrir l'origine ou la destination d'un fichier en passant par son réseau,
2. Donner la responsabilité aux clients pour les contenus physiques de leurs fichiers.

Quand un utilisateur de Freenet effectue une recherche d'un fichier, il procède par une inondation de son voisinage à l'aide d'une requête de recherche. Quand un pair reçoit la requête, il vérifie s'il possède le fichier ou pas. si le pair possède le fichier, il répond à la requête, sinon il la propage a son tour à ses voisins. Après que la recherche de fichier soit réussie, le fichier est transmis par le pair qui détient la requête du demandeur , celui-ci récupère à travers chaque nœud qui a transmis la requête de recherche. Pour atteindre l'anonymat n'importe quel nœud dans ce chemin peut décider de sa propre revendication ou d'un autre pair choisi comme la source de données. De cette façon, il n'existe aucun moyen de déterminer la source réelle du fichier téléchargé.

1.4.3 La disponibilité

La disponibilité d'un système est la capacité aux utilisateurs de trouver des ressources à tout moment. En effet, avec l'absence d'une autorité centrale, les ressources publiées devraient être disponibles aux autres utilisateurs, même si une partie du réseau est absente.

Cependant la réplication est une technique bien connue pour améliorer la disponibilité du partage P2P des fichiers. Si plusieurs copies des données sont maintenues sur des nœuds indépendants, les chances d'accéder au fichier sont augmentées et la charge totale du réseau aura tendance à diminuer si les répliques sont réparties de manière équitable. Mais déterminer quand et où répliquer les données de façon optimale est un problème difficile.

1.4.4 L'autonomie des noeuds

Chaque noeud gère ses ressources d'une façon autonome. Il décide quelle partie de ses données devrait-il partager. Il peut se connecter ou/et se déconnecter à n'importe quel moment. Il possède également l'autonomie de gérer sa puissance de calcul et sa capacité de stockage.

1.4.5 La scalabilité

Il s'agit de faire coopérer un grand nombre de noeuds (jusqu'à des milliers ou des millions) pour partager leurs ressources tout en maintenant une bonne performance des systèmes. Cela signifie qu'un système P2P doit offrir des méthodes bien adaptées avec un environnement dans lequel il y a un grand volume de données à partager, un nombre important de messages à échanger entre un grand nombre de noeuds partageant leurs ressources via un réseau largement distribué.

1.4.6 Les performances

La performance représente le temps nécessaire pour exécuter des opérations P2P, tel que le routage. Intuitivement, la plus courte durée d'exécution est la meilleure performance du système. Les systèmes P2P structurés assurent une haute performance en utilisant les

DHT [2] pour distribuer et localiser les ressources. En effet, par application d'une fonction de hachage sur un fichier sélectionné, l'utilisateur obtient un identificateur unique qui identifie le fichier. En outre, chaque pair a un identificateur unique à partir du même espace d'identification. Selon l'architecture du système P2P, le pair qui détient actuellement le fichier est celui dont son identifiant est proche de celui du fichier. En conséquence le demandeur de ressources peut facilement localiser la source de données en interrogeant les pairs au sein d'un intervalle étroit autour de l'identificateur du fichier. Cette méthode de recherche permet au demandeur de réduire considérablement son temps de recherche et de la zone de recherche, à la différance des systèmes P2P non structurés où les pairs doivent effectuer une inondation du réseau pour trouver un fichier.

1.4.7 La décentralisation

Le fait que chaque noeud gère ses propres ressources permet d'éviter la centralisation du contrôle. Un système P2P peut fonctionner sans avoir aucun besoin d'une administration centralisée ce qui permet d'éviter les goulets d'étranglements et d'augmenter la résistance du système face aux pannes et aux défaillances.

1.4.8 L'équité

L'équité veille à ce que les utilisateurs offrent et consomment des ressources d'une manière juste et équilibrée. Cette propriété se fonde principalement sur la responsabilisation, la réputation, et les mécanismes d'échange de ressources. Par exemple, pour assurer l'équité dans BitTorrent le principe de réciprocité est appliqué dans les échanges de fichiers. Seuls les pairs participants sont privilégiés pour l'échange des fichiers. Chaque

2. Table de hachage distribué

pair maintient un historique de pairs à partir duquel il télécharge, et quand celui-ci le contacte pour télécharger, il est servit en premier.

1.5 Les services des systèmes P2P

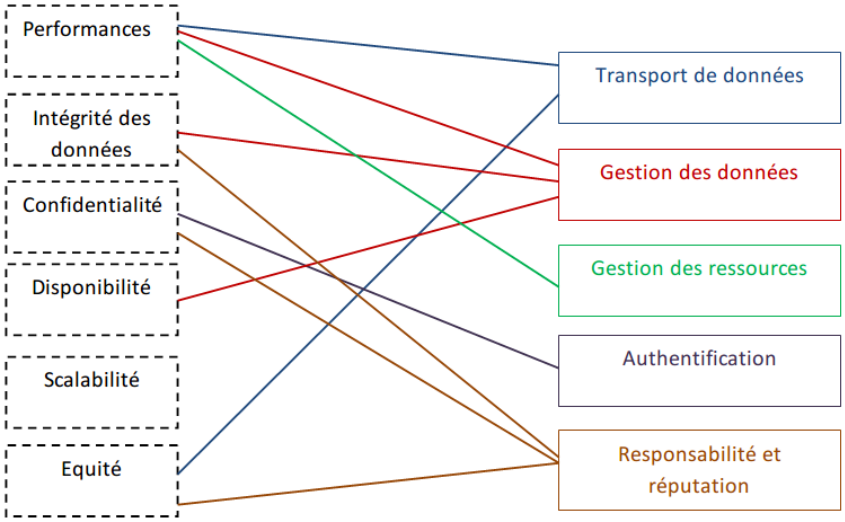

FIGURE 1.4 – Proprietés et services des systèmes P2P

L'architecture des systèmes P2P présente un défi pour fournir un bon niveau de disponibilités, de confidentialité, d'intégrité et d'authenticité. Toutefois, en raison de leur nature ouverte et autonome, les nœuds du réseau doivent être considérées comme des parties non fiables et aucune hypothèse ne peut être faite par rapport a leurs comportements. Les développeurs du système P2P ont apporté des solutions différentes pour répondre et mettre en œuvre toutes les fonctionnalités P2P. Les systèmes P2P sont généralement composés de cinq services : La responsabilité et la réputation, le transport de données, la gestion des données, la gestion des ressources, et l'authentification.

1.5.1 La responsabilité et réputation

Le fonctionnement, la performance et la disponibilité d'un système P2P décentralisé incontrôlé dépendent de la participation volontaire de ses utilisateurs. Il est, par conséquent, nécessaire d'imposer la coopération entre les utilisateurs du système. Dans le but d'imposer une telle coopération, des mécanismes d'incitation doivent être mis en œuvre. Il en existe deux catégories générales [8] :

1. **Mécanismes d'incitation basés sur la confiance**

   La confiance joue un rôle crucial dans la communication quotidienne. En d'autre part , le réseau de communication est totalement différent des interactions naturelles car il est souvent difficile d'évaluer la fiabilité des entités distantes. Par conséquent, les chercheur ont proposé d'évaluer la fiabilité des participants au système en fonction de leur réputation. Toutefois, dans un réseau P2P il n'existe aucune autorité centrale pour maintenir la note de réputation pour les utilisateurs du système. Dans la plupart des systèmes P2P, la réputation d'un pair est un agrégat de sa notation à l'issue d'un transfert de fichier ou de toute autre transaction, les précédentes parties engagées doivent se notter les unes les autres. Une opération globale est obtenue ou un score de confiance d'un pair par l'agrégation de toutes ses notes. Ce score peut aider les autres pairs en décidant si l'on peut interagir avec les pairs concernés ou pas. Ce système de réputation constitue une incitation pour une bonne conduite ; et génère par conséquent un effet positif sur la coopération des pairs. Malheureusment un système de réputation pourrait être attaqué par un ensemble de pairs malveillants. Ils pourraient augmenter leur score de réputation, et donc acquérir une influence sur le réseau, ou bien ils pourraient l'attaquer pour diminuer le score

de réputaiton d'une victime et de la priver d'obtenir tout services.

2. **Mécanismes incitatifs fondés sur la négociation**

   Pour mettre en œuvre un mécanisme d'incitation basé sur la négociation, comme dans le monde réel, un paiement est requis pour compléter une transaction. Le paiement peut être réclamé a l'avance comme il peut être effectué après la transaction avec le fournisseur de la ressource. Il ya deux étiquettes différentes pour l'incitation à la négociation :

   (a) Jeton ou paiements monétaire : comme dans le système de micropaiement (un pair paie pour consommer des ressources et il est payé à fournir des ressources).

   (b) Service differentiel : plus le pair contribue a l'enrichissement du réseau plus il obtiendra une meilleure qualité de service.

1.5.2 Le transport de données

La topologie, la structure et les mécanismes de recherche de routage d'un système P2P sont crucials pour son fonctionnement. Ils affectent sa tolérance aux pannes, son auto-maintenabilité, sa performance, et sa scalabilité. Tout système de partage de fichiers P2P s'appuie sur un réseau de pairs au sein duquel les requêtes et les messages doivent être acheminés avec efficacité, et à travers laquelle les pairs et les ressources peuvent être efficacement situées. Pour localiser les pairs et les fichiers dans les réseaux P2P non structurés, les pairs doivent s'appuyer sur une tierce entité d'indexation comme dans BitTorrent et eMule ou bien sur les inondations comme dans Gnutella et freenet. Dans BitTorrent et eMule, un pair interroge un serveur d'indexation dédié à la recherche d'un certain fichier, le serveur répond alors avec une liste de pairs qui détiennent le contenu.

Après la réception de la liste, le pair va les contacter en utilisant leurs adresses IP déclarées et les numéros de port. D'autre part, dans FreeNet et Gnutella, Le pair inonde ses voisins avec une requête de recherche de fichiers et attend les réponses. Quand un pair reçoit la requête, il répond au demandeur si il détient le fichier, ou il transfert le message au cas où il ne dispose pas du fichier demandé. Chaque message a un TTL [3] limité, une fois la limite atteinte les pairs arrêteront sa transmission. Dans les réseaux structurés, la topologie du réseau est contrôlée et les fichiers sont mis dans des endroits spécifiés. Ces systèmes fournissent essentiellement un mappage entre le contenu et la source sous forme d'une table de routage distribué (DHT), de sorte que les requêtes peuvent être efficacement acheminées vers le nœud avec le contenu désiré. Cette méthode de recherche de fichiers permet au demandeur de réduire considérablement son temps de recherche et de zone de recherche.

1.5.3 La gestion de données

Les systèmes P2P s'appuient sur la réplication des fichiers sur plus d'un noeud pour l'amélioration de la la disponibilité du contenu, l'amélioration de la performance, et la résistance aux tentatives de censure. Ces systèmes diffèrent sur la façon dont ils gèrent la réplication. La réplication de fichiers peuvent être classés en trois catégories principales [8] :

1. **Réplication passive** : la réplication d'une ressource se produit naturellement dans le système P2P ou chaque nœud copie les ressources de l'autre. Par exemple, dans **BitTorrent**et **eMule**, le fournisseur du fichier le divise en morceau (1 Mo à 4Mo), ensuite il procède en partageant les pièces du fichier aux autres pairs. L'avantage

3. Time To Live : « temps de vie en français » indique le temps pendant lequel une information doit être conservée

de cette méthode est qu'elle permet une propagation plus rapide du fichier dans le système, et permet également au pair qui télécharge de participer au partage.

2. **Réplication basée sur la mise en cache** :C'est le processus de réplication des objets de données lors de son passage à travers les nœuds pendant le transfert de fichiers. Par exemple dans Freenet [24] un pair qui recherche un fichier inonde le système avec une requête de recherche de fichiers. Si sa demande de recherche réussit à localiser le fichier, il sera ensuite transféré à travers le réseau nœud par nœud jusqu'a ce qu'il atteint le nœud demandeur. Pendant le transfert du fichier, des copies sont mises en cache sur tous les nœuds intermédiaires ce qui augmente sa disponibilité.
3. **Réplication active** : c'est le processus de distribution des répliques de fichiers sur des zones stratégiques du réseau afin d'améliorer la localité et la disponibilité du contenu. MojoNation [15] et OceanStore [21] sont des exemples de systèmes qui mettent en œuvre la réplication active.

En plus de la réplication du contenu, la cohérence des données et l'intégrité doivent être gérés correctement. La plupart des systèmes P2P utilisent les empreintes digitales pour assurer l'intégrité des fichiers partagés. BitTorrent utilise l'algorithme SHA-1 pour prendre les empreintes de son contenu, tandis qu'eMule utilise l'algorithme MDA-5.

1.5.4 La gestion des ressources

Le domaine de recherche qui a attiré une attention considérable récemment est l'organisation des ressources dans les réseaux P2P. Les fichiers, le stockage et la bande passante sont traité dans la gestion des ressources des systèmes P2P. Les opérations de base de tout système P2P nécessaires sont les suivants : l'insertion la localisation et la récupération des

fichiers. Cependant, plusieurs système de partage de fichier P2P tels que Chord [25] et CAN [41] permettent des gestions de ressources supplémentaires telles que la suppression ou la mise à jour d'un contenu, la gestion de l'espace de stockage et la mise en limite de la bande passante.

1. **Suppression et mise à jour du contenu**

   Suppression et mise à jour du contenu du fichier dans un système P2P n'est pas une opération simple. Une fois qu'un utilisateur publie une partie du contenu sur le réseau, d'autres pairs commencent à télécharger et donc le copier sur leurs machines. Donc, même si certains systèmes ne permettent pas la réplication des données, si le contenu éditeur décide de supprimer son fichier à partir du réseau, certains pairs peuvent publier le même fichier comme si c'était le sien. Par exemple, PAST [17] offre une fonctionnalité de suppression pour récupérer l'espace du disque occupé par un fichier, mais il ne garantit pas que le fichier ne sera plus disponible partout dans le réseau. En effet, il est très difficile de permettre la suppression du contenu et mettre à jour dans les systèmes P2P décentralisés, il est, cependant, possible dans les systèmes P2P centralisés. Publius [38] permet à la fois la suppression et la mise à jour du contenu. Il est basé sur un ensemble statique de super-nœuds qui stockent le contenu. Les fichiers sont publiés avec un mot de passe qui garantit que seul l'éditeur sera en mesure de supprimer ou de modifier le contenu.

2. **Stockage et gestion de bande passante**

   La gestion de l'espace de stockage disponible pour les pairs varie également entre les différents systèmes de partage P2P. La majorité des systèmes permettent à leurs utilisateurs de manière indépendante à choisir la quantité d'espace disque à y con-

tribuer. Par exemple, les utilisateurs de MojoNation [15] contribuent au stockage pour une compensation économique.

1.5.5 L'identification et l'authentification

Les questions de contrôle d'accès, l'authentification et la gestion des identités sont rarement prises en compte dans les systèmes P2P. L'absence d'identification et d'authentification constitue une grande menace pour la sécurité sur un système. Par exemple, dans des systèmes P2P largements peuplées, il est possible pour la même entitée physique de comparaître en vertu différentes identitées logiques. Un nœud malveillant peut tirer parti de cette vulnérabilité, et commence à attaquer le système. Dans plusieurs systèmes, l'absence d'authentification est vaincue par la distribution de clés à un sous-ensemble d'utilisateurs privilégiés pour accéder au contenu. Par exemple, OceanStore [21] permet l'utilisation de certificats signés par les listes de contrôle de l'attribution des objets. Toutes les modifications du contenu d'un objet sont vérifiées à l'aide de la liste de contrôle d'accès qui lui est assigné et toutes les modifications non autorisées sont ignorées.

1.6 Cas d'étude

Dans cette section nous présenterons un exemple du système P2P structuré et un autre du système P2P non structuré.

1.6.1 Chord

Chord est un protocole pour les applications Internet : pour une clé de données, Chord y associe un nœud. Chord repose sur une topologie en anneau ; un nœud Chord a la connaissance de son prédécesseur et son successeur. Une fonction de hachage régulière

génère une clé pour chaque noeud à partir de son adresse IP. Ensuite, chaque nœud est placé dans l'anneau de manière à ordonner les clés par ordre croissant. Ainsi, chaque nœud Chord est responsable de l'intervalle de clés [clé (nœud actuel), clé (suivant)][25]

1.6.1.1 Le principe de fonctionnement de Chord

Le principe clé de Chord est d'utiliser une fonction de hachage qui soit à la fois rapide et répartie uniformément afin que la charge soit égale sur chaque pair du réseau. Un tel hachage est dit consistant. De plus, lorsqu'un pair quitte ou entre sur le réseau, il est hautement probable que seulement $\frac{1}{N}$ (N nombre total de pairs) clés soient déplacées. La scalabilité de Chord est maintenue grâce à la gestion d'une table de routage ne contenant que m entrées (m nombre de bits d'une clé). Grâce à cela, on s'assure qu'une requête sera effectuée via seulement $log_2(N)$ pairs

1.6.1.2 L'identificateur du nœud

Dans Chord, les identificateurs se construisent de la même manière pour les données et les nœuds, ils sont de la forme $h(x)$ où h est la fonction de hachage comme par exemple SHA-1(Secure Haching Algorithm) et x est l'adresse IP du nœud concaténée avec un index d'un nœud virtuel entre zéro et une valeur maximale

1.6.1.3 La fonction de hachage

La fonction de hachage assigne à chaque nœud et à chaque clé un identifiant de m bits. La valeur m doit être bien entendu assez grande pour que la probabilité de collision entre les identifiants soit faible. L'identifiant d'un pair peut être créé par exemple en hachant son adresse IP ou bien son nom d'hôte. La politique d'assignation d'une clé hachée à un pair est simpliste. La clé est associée au premier nœud dont la valeur de l'identifiant

est supérieure ou égale à la valeur de la clé. Ce nœud particulier est appelé successeur . Comme on peut le voir sur la figure suivante, on représente le réseau Chord sous la forme d'un anneau. Chaque nœud est marqué par la lettre N (pour node) suivie de son identifiant.

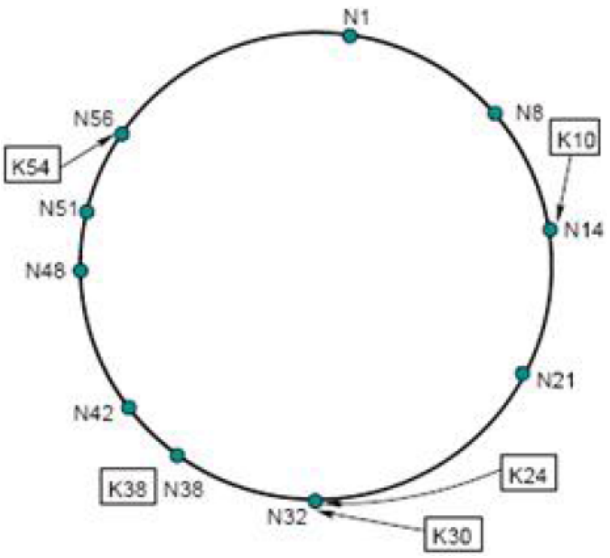

FIGURE 1.5 – Répartition des clés

Les clés stockées sur les différents nœuds sont représentées par des cadres à l'intérieur desquels, on retrouve la lettre K(pour Key) suivie de l'identifiant de la clé. Une flèche indique qu'une clé donnée est sur un nœud donné. Ici, on peut voir que la clé $K10$ est stockée sur le nœud $N14$ car $N14$ est le nœud le plus proche supérieur à $K10$. L'un des principaux intérêts de cette fonction de hachage est de minimiser les déplacements des clés entre pairs lorsque de nouveaux pairs arrivent ou sortent du réseau. Afin que les clés soient équitablement réparties, lorsqu'un nouveau nœud apparaît sur le réseau, son successeur lui transmet automatiquement les clés dont il prendra la charge. Il est prouvé que le nombre de clé déplacée est égale à environs $\frac{1}{N}(N$ nombre de pairs). Dans la figure précédente, si un nouveau nœud d'identifiant 26 venait à se connecter, il capturerait toutes les clés d'identifiant compris entre 21 et 26.

On utilise fréquemment comme fonction de hachage le SHA-1 qui possède de très bonnes propriétés de distribution. Il est malgré tout possible de générer des clés qui sont maladroitements réparties par exemple en fournissant à la fonction de hachage des clés particulières qui cibleront le même pair. On considére ce cas comme hautement improbable.

1.6.1.4 La recherche des données

Chord présente un algorithme de recherche de données simple, il permet la recherche d'une manière exhaustive, c'est-à-dire qu'il garantit la localisation de ressources si elles sont présentes dans le système.

Exemple de recherche Dans la figure suivante, la clé $K16$ est localisée dans le nœud $N21$ et la clé $K54$ dans le nœud $N56$. La simple connaissance du prédécesseur et du successeur permet certes de construire une topologie en anneau, mais elle présente des performances médiocres en terme de nombre de nœuds à parcourir pour acheminer une requête, cette valeur pouvant atteindre $(n-1)$ pour un nœud N envoyant une requête de clé (précèdent (n) - 1). Afin de palier à ce problème, une table de routage distribuée est adoptée dans le protocole Chord, ainsi chaque nœud ne nécessite pas trop d'informations de routage concernant les autres nœuds, c'est uniquement $O(ln(n))$. Pour un espace de clés compris dans l'intervalle $[0, 2^m]$, chaque nœud N se voit doter d'une entrée vers les nœuds (appelés fingers) de clé suivant $(N + 2^{i-1})$ avec $1 < i < m$. Ainsi, chaque nœud maintient une table des successeurs de m entrées, le successeur K qui correspond à la K^{iemme} entrée est le premier nœud sur l'anneau qui vérifie $(i + 2^{k-1}) mod 2n$, $1 \leq k \leq m$. Le nombre maximum de nœuds parcourus pour acheminer une requête est alors exprimé en $O(ln(n))$.

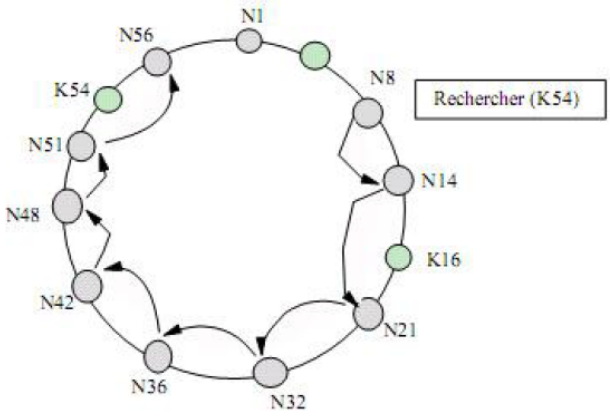

FIGURE 1.6 – Exemple de recherche dans Chord

1.6.2 BitTorrent

BitTorrent est un système P2P structuré qui utilise un endroit central (tracker) pour contrôler les téléchargements des utilisateurs. Le terme BitTorrent désigne à la fois le protocole utilisé pour le partage et les logiciels clients qui l'implémentent.[10]

1.6.2.1 Le principe de fonctionnement

Pour partager un fichier, les clients commencent par créer un fichier "torrent" (extention *.torrent*) qui contient des « métadata » sur le fichier, notamment le HASH File. Il contient aussi l'adresse URL du tracker qu'il faut interroger pour pouvoir télécharger le fichier. Pour télécharger un fichier, le client BitTorrent (programme client) recherche le .torrent

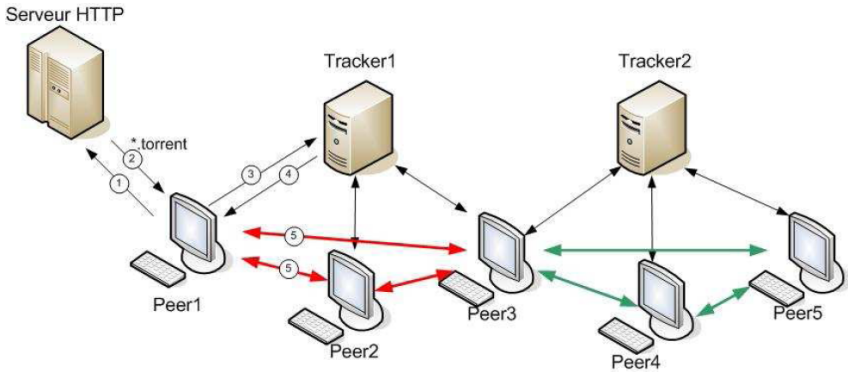

FIGURE 1.7 – Structure générale de BitTorrent

correspondant (sur un serveur web par exemple – flèche 1 sur la figure) et le télécharge (flèche 2). Il utilise ensuite l'URL contenu dans le ".torrent" pour contacter le tracker responsable du fichier pour lui demander la liste des pairs partageant actuellement le fichier (flèche 3). Le tracker renvoie alors une liste d'adresse IP à contacter (flèche 4). Dès lors, le client BitTorrent contacte directement ces adresses et commence le téléchargement (requêtes 5). La liste d'IP est constituée simplement à partir des adresses des personnes qui téléchargent le fichier désiré en même temps que nous : notre client télécharge le fichier sur les autres Pairs, et ceux-ci téléchargent chez nous.

Cette technique permet de faire du **multisourcing** : il est possible de télécharger plusieurs morceaux simultanément à partir de plusieurs pairs différents. De même, après avoir obtenu un morceau, le client servira de source pour l'envoyer vers d'autres utilisateurs. Il est possible de stopper son téléchargement et de le reprendre là ou on l'avait laissé. Enfin le client BitTorrent informe le tracker, à des intervalles de temps réguliers de son état dans le téléchargement (ce qu'il a téléchargé, reçu, ... etc.) et reçoit du tracker une liste actualisée des pairs.

Conclusion

Le système P2P a connu et connait toujours un succès auprès du grand public grâce aux logiciels de partage et de communication, le P2P ne doit pas être systématiquement associé aux applications illégales de partage de fichiers car il est actuellement utilisé dans de nombreux autres domaines d'application tels que : le travail collaboratif ou le calcul distribué ... etc. Si on considère l'évolution actuelle de l'informatique avec les réseaux *adhoc*, les réseaux sans fils, la mobilité et l'accroissement de la puissance des machines

et de la bande passante, il nous semble que dans ce contexte, les réseaux P2P prennent leurs places et occupent une partie importante. Dans ce chapitre, nous avons fait un tour d'horizon sur les systèmes P2P les plus connus, qui sont classé en deux catégories : les systèmes structurés et non structurés puis nous avons présenté les différents services des systèmes P2P. Enfin nous avons présenté un exemple de système P2P structuré et non structuré.

Chapitre

2 Notions élémentaires sur la confiance et la cryptogrpahie

Sommaire

Introduction **27**

2.1 La confiance **28**

2.2 Les thechniques de cryptographie **29**

2.3 La cryptographie d'identité **33**

2.4 La cryptographie a seuil **33**

2.5 PKI (Public Key Infrastructure) **34**

Conclusion **37**

La sécurité informatique sur un réseau pair à pair consiste à garantir aux données et aux services les trois propriétés suivantes : la confidentialité, l'intégrité et la disponibilité. Sur chaque système, un modèle de confiance est défini pour assurer les propriétés précédentes. Dans ce chapitre, en plus de la confiance, nous rappellerons des définitions de la cryptographie en générale. Puis, en particulier, nous présenterons les notions de cryptographie d'identité, à seuil et l'infrastructure à clés publique. Le rappel de ces notions facilitera la présentation de notre solution où l'autorité de certification est partagée et distribuée.

2.1 La confiance

Un cadre de gestion de la confiance doit permettre à une entité de prendre une décision en fonction de son expérience et d'une analyse des risques encourus. L'idée principale est d'évaluer le trait prévisible d'une autre entité et d'établir le niveau de confiance qu'il lui est porté, c'est-à-dire paraît-il digne de confiance ? Est-il honnête dans les réponses aux requêtes ? Dans [60] les auteurs montrent qu'un tel cadre de gestion de la confiance peut revêtir trois formes :

1. Les systèmes à base de certificats.
2. Les systèmes de réputation et de recommandation.
3. Les systèmes développés à partir du réseau social de l'utilisateur.

Les deux premiers systèmes reposent en général sur une infrastructure à clés publiques et sont aujourd'hui les plus répandus. Ils garantissent l'identité de chaque entité par l'émission d'un certificat. L'autorité peut se répartir suivant deux modèles : centralisé ou distribué. Le modèle distribué offre une meilleure disponibilité du service du fait de la décentralisation des informations de confiance mais se heurte cependant à la difficulté de répartir la clé privée avec cohérence entre chaque membre. Dans le modèle distribué, l'autorité est distribuée en plusieurs entités de certification, La cryptographie à seuil est en charge de la problématique de la distribution des clés privées.

Les modèles partagés ne gèrent pas l'identité des entités, et sont statiques. Le secret, distribué au préalable, identifie le groupe et se partage entre l'ensemble des membres. L'authentification s'effectue sur la connaissance du secret partagé. La compromission d'un seul membre met en danger l'ensemble du groupe. Les modèles coopératifs ne nécessitent

pas la présence du tiers de confiance. Chaque entité contribue au calcul du secret du groupe.

Dans le cadre de ce projet, notre étude est portée sur les modèles de confiance à base de certification. Un nœud peut faire confiance à un autre nœud si et seulement si ce dernier est certifié par un tiers nœud envers lequel le premier fait confiance. Ainsi, les nœuds utilisent la vérification des certificats pour établir des liens de confiance avec les autres nœuds. En effet un certificat est une structure de données dans laquelle une clé est liée à une identité délivrée par une tierce partie de confiance. Si cette dernière estime qu'un nœud donné est digne de confiance, elle lui délivre un certificat qui va lui permettre de prouver sa légitimité envers les autres nœuds du réseau. Dans cette catégorie de modèles de confiance, la relation de confiance entre les nœuds est gérée d'une manière transitive, telle que si A fait confiance à B et B fait confiance à C alors A peut faire confiance à C. Dans cette relation, l'intermédiaire B est la tierce partie de confiance. Cette dernière pourrait être une autorité centrale ou un simple nœud intermédiaire. Dans la suite de ce chapitre nous donnerons une description des différentes techniques de cryptographie nécessaires pour la mise en œuvre des modèles de confiance à base de certification.

2.2 Les thechniques de cryptographie

Le mot "Cryptographie" est un mot d'origine grecque composé de deux parties : "Crypto" (kruptos) qui signifie caché et "graphie" (graphein)qui signifie écrire. D'une manière générale, la cryptographie est la science de la dissimulation de messages de sorte que seuls certains initiés disposant d'une donnée secrète soient en mesure de l'opération. La première opération E est le chiffrement (appelée aussi cryptage). Elle permet de conver-

tir un message intial M, (dit message en claire) en un autre message C (dit message chiffré) incompréhensible par une tierce partie. La forme de C dépend d'un paramètre K appelé clé de chiffrement. La deuxième opération D est le déchiffrement (appelée aussi décryptage). le déchiffrement permet de reconstruire le message clair à partir du message chiffré. Cette reconstruction requiert une deuxième clé K^{-1} dépendante de la clé de chiffrement, dite clé de déchiffrement. la définition de la paire (E,D) constitue un système cryptographique. Les systèmes cryptographiques les plus utilisés peuvent être classés en trois types :

(1)Systèmes cryptographiques à sens unique,(2)Systèmes cryptographiques symétriques,(3)Systèmes cryptographiques asymétriques.

2.2.1 La cryptographie á sens unique

Dans ce type de cryptographie [2], le message est chiffré de telle sorte qu'il est impossible de reconstruire le message original à partir du message chiffré. Cette technique de cryptographie a donné naissance à une catégorie de fonctions, appelées fonctions de hachage, qui sont largement utilisées pour le contrôle d'intégrité des données. Etant donné une fonction de hachage H et un message M. On appelle le résultat de hachage $h = H(M)$ le condensé de M. Les fonctions de hachage sont caractérisées par :

1. Il est impossible de retrouver le message M à partir de h.
2. Il est difficile de trouver un message $´M$ tel que : $H(M)= H(´M)$.

Généralement, la taille de h est toujours constante et elle est très petite par rapport à M. En effet, cette catégorie de fonctions est trés utilisée dans les opérations cryptographiques, principalement dans le but de réduire la taille des données à traiter par la fonction de chiffrement. Une fonction de hachage reçoit une entrée de longueur variable et produit

une sortie de longueur fixe. Ce type de fonctions assure que si l'information est modifiée, même d'un seul bit, une sortie totalement différente serait produite. Il existe deux types de fonctions de hachage : les fonctions de hachage avec et sans clé. Les fonctions de hachage sans clé peuvent être calculées par n'importe quelle entité participante à la communication. La valeur calculée dans ce cas ne dépend que du message initial, alors que les fonctions de hachage avec clé sont en fonction du message initial et d'une clé de hachage. Seuls ceux qui possédent cette clé peuvent calculer la valeur de hachage correspondante au message initial. Les fonctions de hachage les plus répandues sont $MD4$ et $MD5$ (Message Digest) [63] et SHA (Secure Hash Algorithm) [7].

2.2.2 La cryptographie symétrique

Dans ce type de systèmes [2], la même clé K est utilisée à la fois pour le chiffrement et le déchiffrement. Si nous supposons que $´M = E(M, K)$ est le chiffrement du message M en utilisant la clé K, et $D(´M,K)$ pour le déchiffrement du message $´M$ en utilisant la même clé K, alors la caractéristique fondamentale de cette technique de cryptographie est : $D(E(M, K), K) = M$.

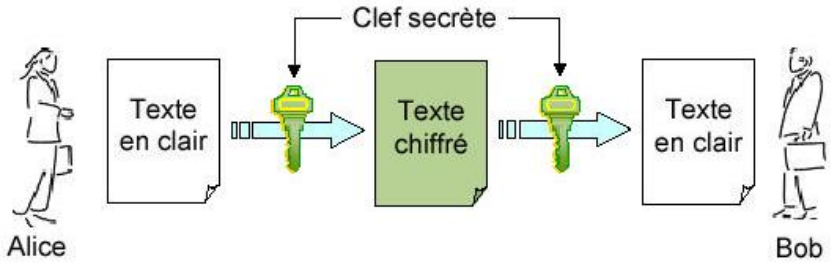

FIGURE 2.1 – Chiffrement symétrique

Chaque communiquant dans le systmèe désirant transmettre des données confidentielles doit partager avec son partenaire une clé secrète, et cette dernière sera utilisée par l'-expéditeur pour chiffrer les données avant de les envoyer et par le destinataire pour les

déchiffrer une fois reçues. Les systémes cryptographiques symétriques les plus répandus sont : DES (Data Encryption Standard) [45], AES (Advanced Encryption Standard) [46], et IDEA (International Data Encryption Algorithm) [64].

2.2.3 La cryptographie asymétrique

Le concept de la cryptographie asymétrique a été inventé par Diffie et Hellman [20]. Cette technique utilise une paire de clés complémentaires : une clé publique qui chiffre les données, et une clé privée pour les déchiffrer. La clé publique doit être diffusée à tous les correspondants dans le système, par contre la clé privée doit rester secrète au niveau de son propriétaire. Toute entité en possession d'une copie de la clé publique peut chiffrer des informations que seul le propriétaire de la clé privée pourra les déchiffrer [2]. Les systèmes cryptographiques à clés asymétriques les plus répandus sont Elgamal [20], et RSA (Rivest Shamir Adelman) [49]. La signature numérique est l'un des services réalisés grâce à la cryptographie

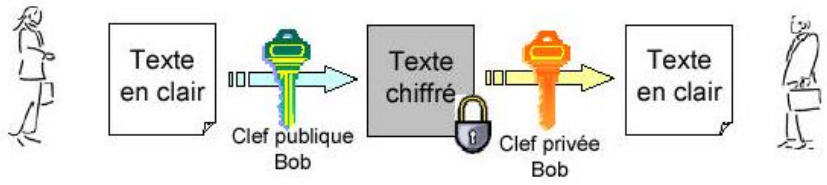

FIGURE 2.2 – Chiffrement asymétrique

asymétrique. Elle fournit les services d'authentification, d'intégrité des données, et la non-répudiation [1]. Sur le plan conceptuel, la façon la plus simple de signer un message consiste à chiffrer celui-ci avec la clé privée de l'expéditeur. Seul le possesseur de cette clé est capable de générer la signature. Cependant dans la pratique, cette méthode est peu utilisée du fait de sa lenteur. La méthode réellement utilisée consiste à calculer une

1. Garantir qu'aucun des correspondants de l'information ne pourra nier la transaction

empreinte du message à signer et de ne chiffrer que celle-ci. Pour calculer l'empreinte, on utilise les fonctions de hachage, là où le condensé généré sera considéré comme résumé du message, et ainsi on peut le considérer en tant qu'empreinte du message.

2.3 La cryptographie d'identité

Le cryptage à base d'identité (IBE - Identity-Based Encryption) [56] est une technique de cryptographie asymétrique, dans laquelle la clé publique de chaque utilisateur est liée à son identité. Cette identité sera traitée par une tierce partie centralisée, appelée PKG (Private Key Generator), qui va la combiner avec sa propre clé privée en générant celle qui est propre à l'utilisateur. Ainsi, les utilisateurs peuvent envoyer des messages chiffrés (ou signés) sans avoir besoin de solliciter le PKG en utilisant directement les identités des correspondants comme des clés publiques.

2.4 La cryptographie a seuil

Le partage de secret repose sur le concept de détention d'une portion d'une information secrète par plusieurs personnes, comme un coffre-fort bancaire dont l'ouverture est commandée par l'introduction simultanée de plusieurs clés. Le partage de secret est traditionnellement utilisé en informatique pour scinder des clés de déchiffrement en plusieurs parties de sorte que chacune d'elles possèdent une portion de la clé. Le concept de la cryptographie à seuil a été inventé par *Shamir* [55]. Il a proposé un mécanisme basé sur l'interpolation polynomiale. Il permet le calcul et le partage d'une valeur secrète S à un ensemble de n serveurs, sans que chacun d'eux connaisse sa valeur. A partir d'au moins k serveurs on peut reconstruire le secret. Si le nombre de serveurs est inférieur à k, aucune

information n'est obtenue sur le secret S. Cette technique de cryptographie a été combinée avec le système cryptographique asymétrique RSA pour avoir un système qui permet de partager le pouvoir de signature à un ensemble de serveurs [31].

2.4.1 Le protocole de partage du secret

Ce protocole permet de mettre en commun un secret S entre plusieurs serveurs (s_1, s_2, ..., s_n) de telle sorte qu'à partir seulement de k parts on peut reconstruire le secret S. On crée un polynôme $F(x)$ de degré $k-1$ avec des coefficients aléatoires en mettant $a_0 = S$. On choisit ensuite publiquement n points distincts X_i, tel que $X_i \neq 0$, et on distribue secrètement à chaque serveur S_i une part privée (X_i,$F(X_i)$). Le point X_i pourrait être n'importe quelle valeur publique qui identifie le serveurs S_i d'une manière unique. Pour simplifier la notation, nous mettons $X_i = i$, par conséquent les parts privées sont dénotées par $F(1)$,$F(2)$, ...,$F(n)$.

2.4.2 Le protocole de reconstruction du secret

Ce protocole permet de reconstruire le secret S à partir d'un sous-ensemble de k parts : $F(1), F(2), ..., F(k)$. Etant donné k paires de points distincts $(i, F(i))$, il existe un polynôme unique $F(x)$ de degré $k-1$ passant par tous les points. Ce polynôme peut être calculé à partir des points $(i, F(i))$ en utilisant l'interpolation de Lagrange [55].

2.5 PKI (Public Key Infrastructure)

Une infrastructure à clés publiques est un ensemble d'outils et de fonctions dédiés à la gestion de clés publiques. Elle est composée d'un ensemble d'autorités de certifications qui assurent la gestion des certificats. Une autorité de certification est un serveur particulier

qui a le rôle de : générer, émettre, révoquer, et renouveler les clés publiques des utilisateurs.

2.5.1 Les cértificats

C'est un document électronique attestant qu'une clé publique est bien liée à une organisation, une personne physique, une machine, ou une application. Il contient une clé publique, une identité du possesseur, et un certain nombre de propriétés, le tout signé par une autorité de certification. C'est la liaison de l'identité du possesseur et des propriétés à la clé publique, crée par la signature numérique qui constitue un certificat.

2.5.2 Les annuaires de certificats

Les certificats générés par l'infrastructure de gestion de clés publiques doivent être rendus publiques afin que toutes les autorités de certification puissent y accéder. Pour cela, les certificats sont publiés dans un annuaire d'accès libre [52]. L'annuaire constitue en quelque sorte une base de données centrale de certificats. Elle est accessible en mode d'ajout (nouveau certificat) et en mode de suppression (révoquer un certificat existant) par le biais des autorités de certification.

2.5.3 L'organisation des autorités de certifications

Les certificats générés pour tous les utilisateurs ne peuvent pas être issus d'une même autorité de certification. Ainsi, il est nécessaire que ce rôle soit réparti à travers plusieurs autorités organisées d'une manière particulière.

2.5.3.1 Le modèle hiérarchique

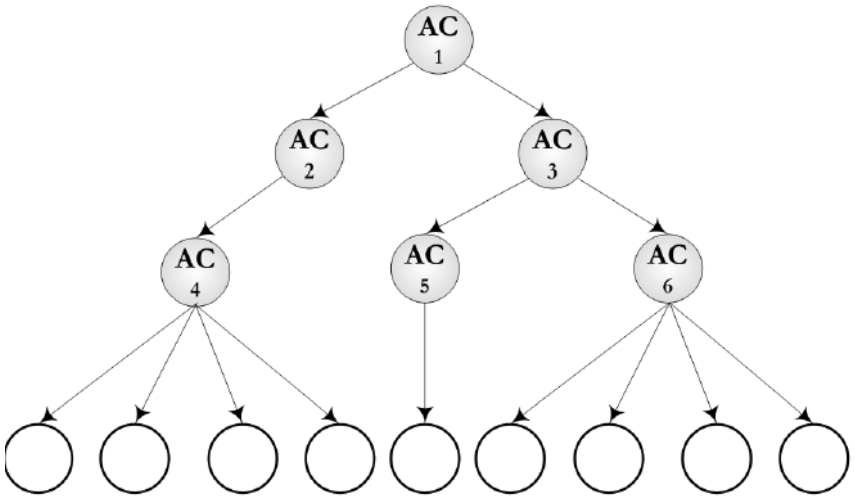

FIGURE 2.3 – Modèle hierarchique "$x \rightarrow y$" :x signe le cértificat de y "AC" : Autorité de Certification

Le modèle hiérarchique est organisé comme suit. L'autorité de certification mère délivre un certificat spécifique, qui confère le rôle de certification à une ou plusieurs autres autorités, qui elles-mêmes à leurs tours délivrent des certificats à d'autres, et ainsi de suite. Au sommet, on trouve l'autorité racine dont le certificat est signé avec sa propre clé privée.

2.5.3.2 Le modèle croisé

Les relations croisées servent à relier deux hiérarchies d'autorités de certification de deux organismes différents. La racine de chaque hiérarchie signe le certificat à clé publique de l'autre pour former une passerelle. Ainsi, n'importe quel utilisateur de la première hiérarchie pourra vérifier la clé publique de n'importe quel utilisateur de l'autre hiérarchie.

2.5.3.3 Le modèle anarchique

Ce type de modèle considère chaque utilisateur comme une autorité de certification où il peut générer et signer des certificats pour les autres utilisateurs. Il utilise un graphe de confiance particulier, appelé web-of-trust, il consiste en l'établissement d'un réseau de

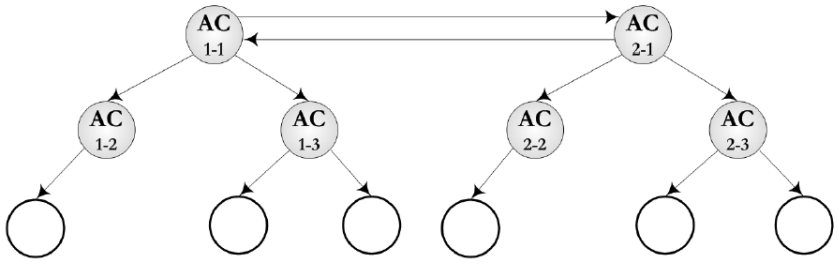

FIGURE 2.4 – Modèle croisé $''x \rightarrow y''$:x signe le cértificat de y "AC" : Autorité de Certification

gestion complètement distribué de clés publiques. Nous illustrons dans la figure suivante un exemple de graphe de confiance qui comporte un certain nombre d'utilisateurs. Chaque utilisateur délivre des certificats à l'ensemble des utilisateurs à qui il estime être digne de confiance.

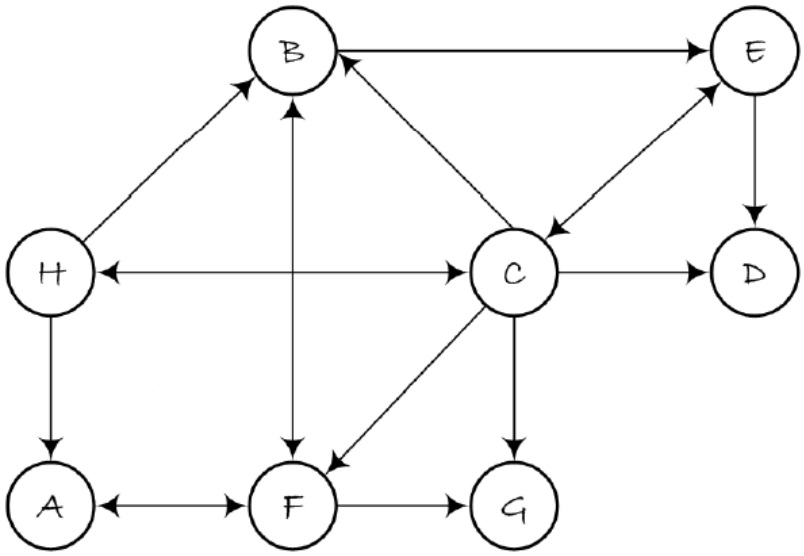

FIGURE 2.5 – Modèle anarchique "$x \rightarrow y$" :x signe le cértificat de y

Conclusion

Dans ce chapitre, nous avons présenté les différents concepts de la cryptographie moderne. Nous avons, en particulier présenté les différentes techniques existantes et l'infras-

tructure à clé publique en donnant un aperçu sur ses différents types d'organisation. Après avoir rappelé ces concepts, nous allons présenter quelques attaques sur ces réseaux qui visent les propriétés de sécurité et les contre-mesures utilisées pour se défendre contre elles.

Chapitre

3 Les differentes attaques dans les réseaux Pair-à-Pair

Sommaire

Introduction **39**

3.1 Les attaques d'infrastructure **41**

3.2 Les attaques de données **52**

Conclusion **56**

Les systèmes P2P ont été soumis à une analyse approfondie et une conception soignée pour garantir la saclabilité et l'efficacité. Ces systèmes fournissent une plate-forme puissante pour la construction d'une variété de services décentralisés, y compris le stockage en réseau, la mise en cache, la recherche et l'indexation, et l'application au niveau de multidiffusion . Cependant, ce qui rend ces systèmes "sécurisé" est un défi de taille [33] [58] [62]. Un nœud malicieux peut y répondre avec des messages erronés à une demande, que ça soit au niveau de l'application (envoi de données fausses à une requête) ou au niveau du réseau (envoi de fausses routes).Les attaquants ont d'autres objectifs comme l'analyse du trafic contre les systèmes assurant la communication anonyme, ou la mise en œuvre de censure contre les systèmes fournissant des fichiers à haute disponibilité. Certaines at-

taques sont conçues pour induire un impact sur les infrastructures réseaux. Ces attaques peuvent causer des dommages et de grandes conséquences sur la qualité du service et la fiabilité des fournisseurs de services Internet, tel que le déni de service distribué (DDoS). DDoS est une attaque à laquelle l'attaquant tente de faire un certain service indisponible pour ses utilisateurs prévus. En outre, il existe plusieurs autres attaques qui peuvent être menées contre les réseaux P2P avec plus de succès en raison de la nature des réseaux P2P. De telles attaques sont comme les attaques Sybil là où les attaquants attaquent le système de réputation d'un réseau P2P d'acquérir une grande influence de manière disproportionnée dans le réseau [16] [22]. Un autre exemple d'attaques spécifiques aux réseaux P2P est à l'attaque Eclipse. Dans celle-ci, un ensemble de pairs malveillants trompent d'autres paires en se connectant uniquement avec eux. Si l'éclipse est menée avec succès, l'attaquant peut gérer toutes les communications vers et à partir de la victime, et lorsque l'attaque est appliquée sur une plus grande échelle ; il peut diviser le réseau P2P [5] [4]. Des vers actifs peuvent également créer des dommages sur les réseaux P2P, ce sont des programmes qui s'auto-propagent à travers internet en exploitant des failles de sécurité dans les services couramment utilisés[53]. En plus de ces attaques « difficiles », certains utilisateurs malveillants peuvent tout simplement avoir envie de profiter des services des réseaux sans participer en retour. Comme le free-riding et attaques badigeonnage, où les attaquants profitent des autres pairs sans aucune contribution dans le réseau. En outre, les réseaux P2P sont infectés par des spywares. Un spyware est un programme malveillant qui est habituellement distribué avec le logiciel du client P2P, et qui tente d'envoyer des informations personnelles à des utilisateurs. Cydoor[12] est un exemple de logiciel espion qui est distribué avec quelques-uns des clients P2P les plus populaires. Enfin, les clients P2P rendent les réseaux P2P encore plus vulnérables depuis qu'ils exposent aux utilisa-

teurs tous leurs défauts de sécurité. La sécurité a été rarement prise en considération lors de l'élaboration des systèmes P2P. Les systèmes P2P en matière de sécurité sont sacrifiés dans le souci de simplicité et de performance.En raison de leur design et leur architecture ouverte, les systèmes P2P ont un certain nombre de vulnérabilités qui pourraient être exploitées par des attaquants. Dans ce qui suit, nous identifions les failles de sécurité et les attaques actuelles contre les réseaux P2P. Nous classons les attaques en deux classes différentes : l'attaque de donnés et l'attaque d'infrastructure.

3.1 Les attaques d'infrastructure

Le fonctionnement de tout systèmes de distribution de contenu P2P est formé sur le dessus d'un sous-jacent réseau physique (généralement Internet) est considéré comme un "overlay" du réseau. Ainsi, les attaques sur les systèmes P2P pourraient avoir un impact direct sur l'infrastructure d'Internet. Les attaques d'infrastructure ciblent principalement la scalabilité et les performances du système P2P en attaquant le gestionnaire des ressources et des services de transport de données.

3.1.1 L'attaques de déni de service (DoS)

3.1.1.1 Description de l'attaque

Un déni de service (DoS) est une attaque sur un ordinateur ou un réseau qui rend un service non disponible [47]. Il existe de nombreuses formes ou méthodes pour lancer une attaque DoS. Dans le cas des réseaux P2P, la forme la plus commune d'une attaque DoS est une tentative d'inonder le réseau avec des de faux paquets, empêchant ainsi le trafic du réseau légitime. Une attaque DoS peut être effectuée en utilisant des hôtes multiples dans

l'attaque, créant ainsi un déni de service distribué (DDoS) [18] [13]. Dans une attaque DDoS, les attaquants sont souvent des ordinateurs personnels compromis par un ver ou un cheval de Troie. L'attaquant peut alors contrôler à distance ces machines (qualifié en tant que zombies ou des esclaves) et de diriger une attaque distribuée contre n'importe quel hôte ou réseau.

Type	Methode d'attaque dans BitTorrent	Mode	Exigences
1	La victime est un pair participant	Mode tracker centralisé	Envoyer un message usurpé au tracker annonçant la victime comme un pair participant dans l'essaim. Ou si l'un des trackers est compromis,inclure l'adresse de la victime dans la liste des pairs.
2	La victime est un tracker centralisé	mode tracker	Publier le fichier torrent avec trackers multiples. au moins une entrée contient l'adresse de la victime. une autre entrée contient l'adresse d'un suiveur de modification, qui répond avec un faux numéro de seeders ou leechers

TABLE 3.1 – Différentes méthodes d'attaque DDoS dans BitTorrent

Des recherche ont porté sur la faisabilité d'utiliser les réseaux P2P comme une plate-forme pour lancer des attaques DDoS [19] et [44]. Récemment, il a été démontré qu'une attaque DDoS lancée par les clients BitTorrent peut provoquer de graves dommages. plusieurs des caractéristiques qui font que BitTorrent soit populaire et puissant peuvent être malicieusement [29] exploité pour transformer son réseau en une plate-forme DDoS. L'ouverture de trackers et des moteurs de recherche de torrent permet à quiconque de publier un torrent facilement et sans authentification. Les attaquants peuvent facilement exploiter cette ouverture et l'absence d'authentification pour établir leurs attaques DDoS. Il ya deux méthodes d'attaques qui pourraient être utilisés pour lancer des attaques DDoS

en utilisant BitTorrent comme indiqué dans le tableau 3.1 [57].

1. **Attaque type 1** : L'attaquant envoie un message usurpé sur le tracker annonçant un nœud spécifié est un participant dans l'essaim. Après la réception du message, le tracker va ajouter la victime à sa liste de pairs et envoi son adresse aux pairs qui téléchargent. Si la victime n'est pas un client BitTorrent, il ne peut pas correctement traiter les demandes des fichiers reçus. Par conséquent, à chaque fois qu'un pair téléchargeant essaye de se connecter avec la victime, il va inonder sa connexion avec une requête de fichier non gérée.
2. **Attaque type 2** : L'attaquant exploite le fait que BitTorrent repose sur les trackers centrales pour trouver les pairs participants et les fichiers nécessaires. En faisant croire aux clients que la victime est un tracker, Ils inonderont la victime de leurs messages qui créera une attaque DDoS

DDos attaque les services de transport de données et de gestion des ressources et diminue les performances et la scalabilité du système attaqué.

3.1.1.2 Les défenses contre les attaques de déni de service

Il est difficile de détecter une attaque de type DoS, car elle peut être confondue avec une utilisation intensive du réseau. Les attaques DDoS sont difficiles à bloquer en raison du nombre énorme de machines. Un utilisateur malveillant peut être impliqué dans l'attaque (pratiquement n'importe quel appareil peut être transformé en un zombie). Par ailleurs, l'attaquant est rarement directement impliqué car il attaque par le biais des zombies. Par conséquent, il est souvent difficile d'identifier sa source. Une technique largement utilisée pour empêcher les attaques DOS est la tarification [47]. Le nœud contacté présente des énigmes à l'expéditeur avant de continuer le calcul demandé, assurant ainsi que les

clients passent par un calcul tout aussi coûteux. Si à chaque fois un attaquant tente d'inonder une connexion victime, il reçoit une énigme à résoudre à l'avance, il devient plus difficile de lancer une attaque DoS avec succès. Cette méthode est appelée tarification. La tarification peut être modifiée pour envoyer des énigmes plus dures de façon exponentielle si la connexion d'hôte est largement utilisée, et réduit donc l'effet d'une attaque potentielle. Bien que cette méthode soit efficace contre un petit nombre d'attaquants simultanés, il échoue contre les attaques largement distribués. L'envoi des énigmes pour un grand nombre de «demandeurs» est coûteux. Un autre inconvénient de la tarification, c'est que certains clients légitimes, tels que les appareils mobiles, pourraient percevoir des énigmes trop difficile à résoudre, et/ou pourraient gaspiller de l'énergie de la batterie limitée.

3.1.2 Les Botnets

3.1.2.1 Description de l'attaque

Un botnet est constitué d'un réseau d'ordinateurs zombies contrôlés par un attaquant (ou botmaster). Le terme "botnet" est dérivé de robots logiciels [1] [50] . Ces robots sont contrôlés à distance pour effectuer des dénis de service distribués(DDoS) à grande échelle, envoyer du spam, chevaux de Troie livrer, envoyer des courriels d'hameçonnage, ou à distribuer aux médias des droits d'auteurs ou procéder à d'autres activités illégales [3] [40]. Un botnet est un réseau de communication contrôlé, son architecture la plus courante est l'architecture centralisée. Dans une telle architecture, les robots sont reliés à un serveur de commande et de contrôle (C et C) (botmaster) qui les contrôle et les dirige. Toutefois, de telles architectures ont une faiblesse majeure : le botnet peut être arrêté si le défenseur capture le serveur C etC [14] [50]. Les botmasters commencent à se développer sur des

1. de l'anglais "Software robots" ou "bots"

architectures différentes afin d'éviter cette faiblesse. Dans l'architecture P2P, un nœud peut agir comme un serveur et comme un client, il n'ya pas de point central d'échec, la même architecture pourrait être appliqué pour construire les réseaux de zombies. Un botnet P2P nécessite peu ou pas de coordination formelle et même si un nœud est mis hors ligne par un défenseur, le réseau reste sous le contrôle de l'attaquant. Ainsi les robots collecteurs de P2P gagnent beaucoup d'intérêts dans la communauté des attaquants ainsi que dans la recherche [40].les botnets P2P sont un défi de la sécurité de l'Internet et les réseaux P2P. Comme les attaques DDoS, les botnets attaquent les services de transport de données et de gestion des ressources et diminue la performance et la scalabilité du système attaqué.

3.1.2.2 Les défenses contre les P2P botnets

Les approches pour détecter les réseaux de zombies P2P s'appuient sur l'effort humain. Tout d'abord, un spécimen du robot P2P doit être capturé et étudié. Les modes d'échange de messages et des signatures doivent être extraites. Ensuite, un logiciel est développé et utilisé pour inspecter l'infiltration des nœuds par un botmaster. Enfin, les robots détectés seraient mis en quarantaine et isolés du réseau ainsi que le botnet. Un nombre incalculable de techniques créatives sont appliqués pour analyser les réseaux et de détecter les robots. On peut citer des visualisations, identifiant les caractéristiques anormales dans des recherches dans la liste noire, la compréhension des mécanismes de propagation, et ainsi de suite (par exemple, [40] [50]). Cependant, tandis que des approches créatives et à forte intensité de connaissances sont évidemment utiles, il est important d'être en mesure de détecter automatiquement et de façon générique les botnets. Idéalement, le réseau de surveillance et des outils filtrage sont installés au niveau des routeurs et autres composants du réseau doivent veiller à ce contrôle avec une intervention humaine. Un certain nombre

de chercheurs travaillent actuellement sur des solutions automatisées pour la détection botnet. Les auteurs de [28] ont étudié la faisabilité d'appliquer un système graphique basé sur la détection automatique pour les applications P2P, afin de détecter les réseaux de zombies P2P. Leurs résultats montrent que les botnets primaire P2P cachent leur trafic en appliquant quelques techniques P2P qui sont connus et facile à détecter. Ces résultats sont vraiment prometteurs pour la conception de l'avenir de détection botnets dans les systèmes P2P.

3.1.3 La propagation des vers

3.1.3.1 Description de l'attaque

Un ver informatique est un programme qui se propage en se reproduisant sur le réseau. En raison de leur nature récursive, les vers posent une grave menace sur l'infrastructure Internet en tant qu'un ensemble.les vers modernes peuvent contrôler une partie importante d'Internet au bout de quelques minutes. Pas de réponse humaine est capable d'arrêter une attaque qui est si rapide. Les vers engendrent des effets dévastateurs sur le fonctionnement d'Internet. En outre, le trafic généré par la propagation des vers est si énorme qu'elle peut être considérée comme une attaque DDoS sur Internet et qu'elle pourrait être utilisée pour faire baisser l'infrastructure d'Internet d'un pays entiers. Par conséquent, une foule de travaux de recherche ont été menées afin de développer la détection des vers et des systèmes de confinement. Il ya une nouvelle tendance de vers qui s'émergent et qui ont un potentiel de destruction énorme, ces vers sont appelés les vers P2P. Un ver P2P est un ver qui exploite les vulnérabilités d'un réseau P2P afin de se propager sur le réseau et d'accélérer sa propagation à travers Internet. Les vers P2P pourraient être beaucoup plus rapides que les vers classiques, en outre, ils devraient être l'un des meilleurs animateurs

de la propagation des vers sur Internet dues aux raisons suivantes : [53] [61] [43] :

1. Comme les systèmes P2P ont un grand nombre de pairs inscrits actifs, s'ils sont contaminés, il peut facilement accélérer la propagation des vers sur Internet.
2. Depuis que les pairs dans les systèmes P2P maintiennent un ensemble de voisins pour le routage, si un pair est infecté par un ver, celui-ci sera facilement utiliser cette topologie pour attaquer ses prochaines victimes.
3. Depuis que les réseaux P2P sont composés de tous les ordinateurs exécutant le même logiciel, un attaquant peut ainsi compromettre l'ensemble du réseau en trouvant un seul trou de sécurité exploitable.
4. Étant donné que les programmes exécutent souvent sur des ordinateurs personnels, les vers peuvent avoir accès aux fichiers sensibles tels que mots de passe, numéros de cartes de crédit, carnets d'adresses... etc.

Un ver typique fonctionne comme suit : il balaye d'abord un réseau pour trouver des victimes potentielles. Une fois qu'il localise une machine, le ver tente de le sonder en exploitant une vulnérabilité commune. En cas de succès, il transfère une copie de son code malveillant à la victime choisie. Ensuite, la victime infectée sera utilisée de manière récursive pour attaquer le reste du réseau. La clé du succès d'un ver est sa vitesse de propagation plutôt que la vulnérabilité qu'il exploite. Depuis la détection des courants déployés, des systèmes de confinement sont capables de bloquer la propagation des vers, un ver devrait se propager rapidement dans le but de parvenir à un taux élevé d'infection. Le choix d'une stratégie de numérisation efficace permet au ver de parvenir à une grande population en un temps record. La stratégie de propagation des vers P2P pourraient être classés en deux catégories :

1. Listes d'attaques (hitlist) basées sur des vers qui attaquent un réseau en utilisant une liste pré-construit du potentiel des machines vulnérables.
2. Topologique à base de vers qui attaquent un réseau basé sur les informations topologiques trouvé sur leurs victimes.

3.1.3.2 Les défenses contre la propagation des vers

Les possibilités de configuration actuels des systèmes P2P pour protéger leurs utilisateurs contre la propagation des vers sont limitées. Par conséquent, un système de défense proactif doit être mis en œuvre. S. Antonatos et al. [26] ont proposé une méthode pour étrangler la propagation des vers Hitlist. Ils ont observé que la propagation des vers Hitlist a tendance à échouer sous deux conditions :

1. Lorsque les pairs rejoignent et quittent régulièrement le réseau (donc obtenir un nouvel ID du réseau à chaque nouvelle connexion).
2. Lorsque les applications P2P des clients termine anormalement

Par conséquent, les auteurs proposent intentionnellement d'accélérer la décomposition des Hitlist des vers. Pour ce faire, ils utilisent une technique particulière appelée randomisation d'espace d'adressage du réseau (NASR) [2]. Dans sa forme la plus simple, NASR peut être mis en œuvre en adaptant des allocations dynamiques d'adresses des services pour forcer le changement fréquent d'adresses. Si cette approche est déployée sur une grande échelle, elle peut entraver de manière significative la propagation des hitlist. Cependant, ce mécanisme de protection est spécifique aux Hitlists, NASR n'est qu'une solution partielle au problème de confinement des vers. En outre, les études montrent que bien que la randomisation des hitlist limite la propagation des vers, elle ralentit considérablement

2. Network address space randomization

les performances du réseau.

Zhou et al. [65] ont proposé une infrastructure d'auto-défense à l'intérieur d'un réseau P2P, par la définition de certains "nœuds gardiens" parmi les pairs du système. Ces "nœuds gardiens" ont une propriété de détection automatique de vers. Les auteurs suggèrent d'utiliser une approche basée sur l'observation que la majorité des vers modifient le flux de contrôle d'un programme vulnérable à exécuter. Un code malveillant est injecté à partir du réseau ou à partir de la mémoire. Le rôle d'un nœud gardien est de déclencher une alarme quand il détecte un tel comportement. Le but de ces alarmes est d'avertir les autres nœuds du système. Dès la réception de l'alarme, le pair alerté prendra les mesures voulues pour devenir immunitaire. Cette approche est plus complète car il peut être mis en œuvre pour lutter contre la plupart des vers P2P susceptibles d'exister.

Plusieurs problèmes d'implémentation pourraient se produire dans ce système. Par exemple, à partir d'une action de déclenchement des alertes, un adversaire pourrait attaquer un réseau P2P en diffusant de fausses alertes, ce qui pourra engendrer une attaque déni de service. La solution à ce problème est de limiter l'utilisation des alertes.

3.1.4 Les attaques sybil

3.1.4.1 Description de l'attaque

De nombreux mécanismes de sécurité implémentés dans les systèmes P2P sont basés sur des hypothèses spécifiques sur l'identité des pairs et sont donc vulnérables aux attaques lorsque ces hypothèses sont violées. L'attaque Sybil est une attaque où un seul nœud malveillant crée un grand nombre d'entités pseudonymes, et les utilise pour gagner une grande influence de manière disproportionnée [16]. Une fois établi, l'attaquant peut

abuser du protocole de toutes les manières possibles. Par exemple, il pourrait gagner la responsabilité de certains dossiers et de choisir de les polluer. Si l'attaquant peut placer ses identités d'une manière stratégique, les dommages peuvent être encore plus considérables. Il peut choisir de continuer dans une attaque éclipse (section suivante), ou de ralentir le réseau en redirigeant toutes les requêtes dans une mauvaise direction. Par conséquent, elle réduit le niveau de l'équité dans le système.

3.1.4.2 Les défenses contre les attaques Sybil

L'analyse formelle de l'attaque Sybil a été faite. Cependant, il n'ya pas de solution générale à l'attaque. La plupart des solutions proposées font appel à des certifications de confiance [32]. Douceur [16] a prouvé que la certification de confiance est la seule approche qui a le potentiel d'éliminer complètement les attaques Sybil. Toutefois, la certification de confiance repose sur un système centralisé. L'autorité qui doit s'assurer que chaque entité est affecté exactement une identité, comme indiqué par possession d'un certificat, qui n'est pas le cas de la plupart des systèmes de P2P. Une autre solution proposée pour prévenir une attaque Sybil est le test des ressources. L'objectif de cette solution est de tenter de déterminer si un certain nombre d'identités possèdent moins de ressources que prévu que si elles étaient indépendantes [32]. Ces tests comprennent des contrôles pour la capacité de calcul, la capacité de stockage, et la bande passante du réseau, ainsi que les adresses IP limitées. Les auteurs de [11] [36] proposent des tests pour les adresses IP dans les différents domaines ou des systèmes autonomes. Exiger des adresses IP hétérogènes empêche certaines attaques, mais limite l'utilisation d'une application. Douceur [16] a prouvé l'inefficacité de tests de ressources, mais un certain nombre de chercheurs les suggèrent comme une défense minimale à l'attaque Sybil [32]. Dans ces cas, l'objectif déclaré est de décourager plutôt que de prévenir les attaques Sybil. En théorie, le nombre

d'identités d'un attaquant peut avoir est limitée. Pour beaucoup d'applications, ce n'est pas suffisant si un attaquant peut obtenir assez d'identités pour que l'attaque soit un succès, même si elle est coûteuse. Dans le système de communication Tor, par exemple, seulement deux identités sont nécessaires pour une attaque sur l'anonymat [48].

3.1.5 Les attaques éclipse

3.1.5.1 Description de l'attaque

Dans l'attasue éclipse, un ensemble de pairs malveillants piègent les autres pairs en se connectant uniquement avec eux. Dans cette attaque, l'attaquant tente de placer ses machines compromises sur des points stratégiques dans la topologie du système sous l'attaque. En plaçant son ou ses machines en tant que telle, l'attaquant peut servir de médiateur de toutes les communications de la victime, et quand l'attaque est appliquée sur une plus grande échelle, il peut diviser le réseau P2P en sous-réseaux séparés par des nœuds attaquant [6]. Ainsi, si un nœud veut communiquer avec un autre nœud à partir de l'autre sous-réseau, son message doit à un certain point être acheminé à travers l'un des nœuds de l'attaquant. L'attaquant alors va "éclipser" chaque sous-réseau de l'autre. Un attaquant peuvent exploiter des machines physiques avec des identités distinctes, ou d'exploiter des identités pseudo applicative pour procéder à son attaque. Une attaque éclipse peut être la continuation d'une attaque Sybil. En effet, l'attaquant essaye de placer ses nœuds sur les axes stratégiques de routage. Si un attaquant parvient à gérer une attaque éclipse, il peut :

1. Attaquer l'infrastructure du réseau par un reroutage inefficace de chaque message.
2. Décider d'abandonner tous les messages qu'il reçoit, donc totalement séparer les deux sous-réseaux.

3. Injecter au réseau des fichiers pollués.

L'attaque éclipse vise les services de transport de données dans les systèmes P2P et réduit ses performances et sa scalabilité.

3.1.5.2 Les défenses contre l'attaque eclipse

Castro et al. [34] proposent d'utiliser de fortes contraintes structurelles sur le réseau de recouvrement pour se défendre contre l'attaque Eclipse. Chaque nœud se voit attribuer un identifiant et ses voisins sont les nœuds de recouvrement ayant les plus proches identificateurs. Cette défense est efficace, il ne permet pas au nœud malveillant de prendre le contrôle sur le réseau. Cependant, elle laisse aucune flexibilité dans le choix des voisins et empêche donc les optimisations comme la proximité de la sélection du voisin(SNP) [3], une technique importante et largement utilisée pour améliorer l'efficacité du recouvrement. Les auteurs de [6] présente une défense contre les attaques basées sur Eclipse anonyme l'audit des nœuds voisins. Si un nœud a des liens beaucoup plus que la moyenne, il pourrait être une attaque de montage Eclipse. Lorsque tous les nœuds du réseau effectuent cet audit régulièrement, les attaquants sont découverts et peuvent être retiré du voisinage de l'ensembles des nœuds corrects. La défense est applicable à des recouvrements homogènes structurés.

3.2 Les attaques de données

Les attaques de données ciblent principalement l'intégrité et la disponibilité des fichiers partagés en attaquant le service de gestion de données dans le système P2P. Cependant ; elles ne sont pas complètement indépendantes contre les attaques d'infrastructures. Par

3. Proximity Neighbor Selection

exemple, en s'attaquant aux données partagées et en corrompant de nombreux fichiers, les utilisateurs ont tendance à télécharger d'autres instances d'un fichier ce qui ralentit le trafic qui est généralement le but d'une attaque d'infrastructure.

3.2.1 L'empoisonnement des fichiers

3.2.1.1 Description de l'attaque

Les attaques d'empoisonnement de fichiers sont devenues extrêmement courante dans les réseaux P2P. L'objectif de cette attaque est de remplacer un fichier dans le réseau par un faux. Les auteurs de [27] [42] ont rapporté que l'industrie de la musique a libérées massivement des faux contenus sur les réseaux P2P. En outre, des sociétés spécialisées offrent leurs services basés sur la pollution à l'industrie du divertissement pour défendre les matériaux protégés par copyright. Afin d'établir une attaque d'empoisonnement de fichier, les nœuds malveillants prétendent faussement posséder un fichier, et répondent avec un fichier corrompu sur demande. En outre, tous les messages de contrôle de routage en passant par un nœud malveillant qui peut être empoisonnés. Par conséquent, le fichier sera empoisonné très répandue à travers le Système de P2P, le rendant plus attrayant que le vrai fichier. L'empoisonnement Fichier attaque le service de gestion des données et des services de transport de données dans les systèmes P2P. En conséquence, il corrompt l'intégrité du contenu et diminue la disponibilité du contenu ainsi. En outre, elle dégénère la scalabilité et les performances des protocoles de routage de recouvrement.

3.2.1.2 Les préventions de l'attaque

Identifier les pairs malveillants qui injectent des fichiers pollués est plus efficace que l'identification des fichiers pollués eux-mêmes. En effet, les pairs malveillants peuvent facile-

ment générer de grands nombres de fichiers polluées s'ils ne sont pas interdits de participer dans le réseau. Par conséquent, les chercheurs sont venus avec l'idée de développer des systèmes basés sur la réputation. Les nœuds dans de tels systèmes ont une «réputation» déterminée par tous les autres nœuds [30] [54]. Typiquement, chaque nœud publie une liste de nœuds de confiance, ce qui rend impossible pour un nœud de changer sa réputation en lui-même. Avant d'entamer un échange de fichiers ou de téléchargement, un nœud vérifie d'abord la réputation de celui avec qui il communique, puis il décide de poursuivre ou pas. Dans un sens, plus la réputation d'un nœud est bonne, plus il a de l'importance dans le réseau. La réputation dans ce cas est influencée par l'histoire du comportement par les pairs. Bien que cette approche semble être une bonne direction, les auteurs de [47] affirment que même si les systèmes de réputation introduisent une notion de hiérarchie pour les réseaux P2P, ils constituent une faiblesse. Le problème est que les nœuds avec une meilleure réputation ont plus d'influence sur le réseau que les autres nœuds.En d'autres termes, ils sont capables d'influencer la réputation d'autres nœuds des plus efficaces. Un attaquant attend simplement que l'un de ses nœuds gagne une confiance suffisante pour lancer son attaque. Si l'attaquant déploie de nombreux nœuds malveillants, ils peuvent donner mutuellement une grande réputation qui les rend tous dignes de confiance.

3.2.2 Le Free-Riding

3.2.2.1 Description de l'attaque

Le Free-Riding (opportunisme ou passagers clandestin en Français) a un comportement très similaire à un fichier empoisonné. Le Free-Riding comme son nom l'indique est une attaque où un utilisateur du P2P tire parti des autres pairs sans faire aucune contribution au réseau. Les système P2P compte sur la contribution volontaire des ressources

des participants individuels [35]. Cependant, les résultats de rationalité individuelle dans un comportement égoïste entre pairs, au détriment du bien-être collectif. Des études ont montré que le Freeride est répandu dans les systèmes P2P [35]. Divers mécanismes d'incitation ont été proposés pour encourager la coopération dans les systèmes P2P [9] ,il diminue la disponibilité et l'équité du système d'échange de fichiers.

3.2.2.2 Les préventions de l'attaque

Il est nécessaire d'imposer la coopération entre les utilisateurs du système, ainsi dans le cas d'un pair d'identifier un Free-riding malveillant afin que d'autres pairs puissent être alertés. Pour encourager les utilisateurs à coopérer, un mécanisme d'incitation doit être mis en œuvre. Il existe deux catégories générales de mécanismes d'incitation [8] :

1. **Mécanismes d'incitation basés sur la confiance :** La confiance est une incitation directe à la coopération, dans laquelle on s'engage dans une transaction basée sur si le client fait confiance à un autre.
2. **Mécanismes incitatifs fondés sur le commerce :** Dans les mécanismes d'incitation fondés sur commerce, une partie propose un paiement ou un service en retour quit est explicitement rémunéré quand on fournit un service à d'autres parties. Il ya deux étiquettes différentes pour le mécanisme d'incitation basé sur le commerce :
   (a) L'échange monétaire ou d'un jeton, comme le système de micropaiement. Un pair paie pour consommer ressources et est payé pour apporter des ressources.
   (b) Service d'échange, comme l'algorithme tit-for-tat dans BitTorrent (si un pair contribue plus il obtiendra une meilleure qualité de service).

Conclusion

La facilité d'utilisation et le bas prix contribuent à l'augmentation du nombre d'utilisateurs P2P. Toutefois, ce fait a également inspiré des attaquants afin de s'en prendre aux réseaux P2P. La fabrication de ces systèmes "sécurisé" est un défi de taille. Les réseaux P2P sont constamment sous attaques, de telles attaques sont comme la propagation de vers, ou déni de service distribué (DDoS). Dans ce chapitre, nous avons identifié les faiblesses de sécurité des réseaux P2P, et présenté différentes attaques contre les réseaux P2P et leurs contre-mesures Nous avons classé l'attaque en deux classes différentes : les attaques de données et les attaques d'infrastructures. Les attaques de données visent les données partagées dans l'application P2P, par exemple empoisonner des fichiers ou les rendre indisponibles. En d'autre part, attaquer l'infrastructure signifie s'attaquer directement à l'architecture en essayant de ralentir et le rendre aussi inefficace que possible. Il ya plusieurs attaques qui peuvent être menée contre les réseaux P2P avec plus de succès. De telles attaques sont comme l'attaque Sybil là où un attaquant attaque le système de réputation d'un réseau P2P, ou l'attaque Eclipse où un ensemble de pairs malveillants essaie de diviser le réseau P2P ou tout simplement de le contrôler. Le tableau 3.2 résume les attaques contre les réseaux P2P.

classe d'attaque	Réputation et résponsabilité	Transport de données	Gestion de données	Gestion des ressources	Identification et authentification
D'infrastructure	Sybil	DoS, Botnet, Propagation des vers, Sybil	X	DoS , Botnet, propagation de vers	X
De donnée	Free-Riding	Empoisonnement de fichier	Empoisoinement de fichiers	X	X

TABLE 3.2 – Impacte des attaques P2P

Après avoir analysé les attaques qui pourraient être menées contre les réseaux P2P, et avoir étudié les causes et les conséquences de chaque attaque. Nous constatons qu'il est nécessaire d'établir un modèle de confiance. Dans le chapitre suivant nous allons présenter l'architecture de notre proposition.

Chapitre

4 Contribution

Sommaire

Introduction **58**

4.1 Motivations **59**

4.2 L'architecture du modèle **60**

Conclusion **70**

Dans ce chapitre, nous proposons une architecture d'un modèle de confiance pour le cas du partage Pair-à-Pair dans les réseaux sociaux mobiles. La solution proposée est complètement distribuée où la gestion des certificats est établie à travers un graphe de confiance. Le système permet aux nœuds de générer, stocker et échanger leurs certificats de manière distribuée étant donné qu'une autorité centralisée d'une communauté Pair-à-Pair est inexistante.

Notre solution repose sur la notion de groupes dans les réseaux sociaux en incluant la mobilité des terminaux. La contribution est basée sur l'intégration des techniques de cryptographie à seuil au graphe de confiance afin de résister aux faux certificats qui peuvent être délivrés par des nœuds malicieux dans le réseau et gérer ses différents comportements.

Notre solution est développée pour les réseaux ouverts dans lesquels les nœuds peuvent rejoindre le système à tout moment. Cette opération est permise seulement si le nouveau nœud avait l'autorisation auprès de l'ensemble de nœuds membres du système.

4.1 Motivations

Contrairement aux autres types de réseaux, la caractéristique principale du réseau **Pair-à-Pair** est l'absence d'une administration centrale et par ce fait l'absence d'une autorité de certification. Cependant dans divers modèles de confiance, un concept pour la gestion des relations de confiance est élaboré en se basant sur la règle transitive " si A fait confiance à B et$B$ fait confiance à C alors A fait confiance à C ". Le problème avec cette règle est que si l'élément intermédiaire de la chaine est suspecté malicieux alors toute les chaines qui passent par ce dernier ne sont pas considérées de confiance.

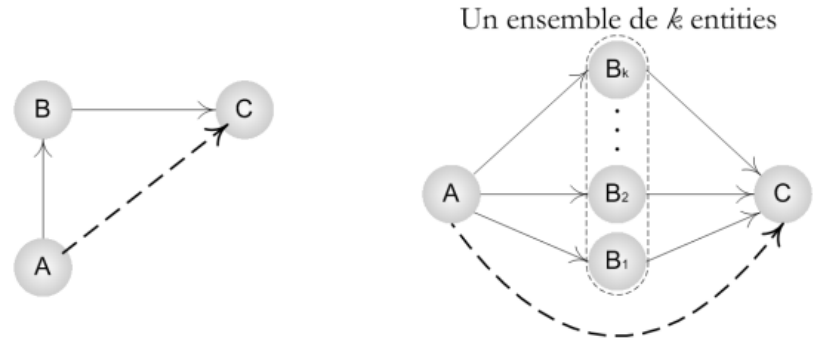

FIGURE 4.1 – Relations de confiance

En se basant sur ce raisonnement nous élaborons une nouvelle règle de confiance plus robuste :"si A fait confiance à B et B fait confiance à C alors A fait confiance à C seulement s'il existe d'autres $K-1$ entités qui font confiance à C". De cette manière l'utilisateur

A aura un meilleur jugement du degré de confiance à attribuer pour l'utilisateur C. Pour ce faire, un schéma de cryptographie à seuil (k, n) est utilisé où n représente le nombre d'utilisateur dans le système et k représente le quorum de confiance. Nous adapterons cette approche au contexte des réseaux sociaux où une communauté d'utilisateurs prend des décisions que ce soit lors d'une adhésion ou une exclusion. Nous détaillerons cette architecture dans la section suivante.

4.2 L'architecture du modèle

4.2.1 La dscription du modèle

Notre modèle permet à des utilisateurs d'un même groupe de générer, stocker et distribuer leurs certificats. Tous les nœuds ont un rôle identique et la notion d'autorité de certification est inexistante. Le système est défini comme suit :

1. La paire de clés K_i/ K_i^{-1} (Publique / privée) de chaque utilisateur i est créée localement par l'utilisateur lui-même.
2. L'authentification des clés publiques est établie à travers la vérification des chaines de certificats.
3. Les certificats sont stockés et distribués par les nœuds mobiles.
4. Lors d'une adhésion, un nouvel utilisateur n'est autorisé à joindre le groupe que s'il obtient un nombre de certificats partiels supérieur ou égale à un seuil précis.
5. Les échanges de fichiers se font en Pair-à-Pair, chaque membre doit être authentifié par l'ensemble des membres du groupe selon **(4)**.
6. Les parts privées seront utilisées pour générer des accusations partielles vis-à-vis des membres malveillants et authentification partielles vis à vis des nouveaux membres.

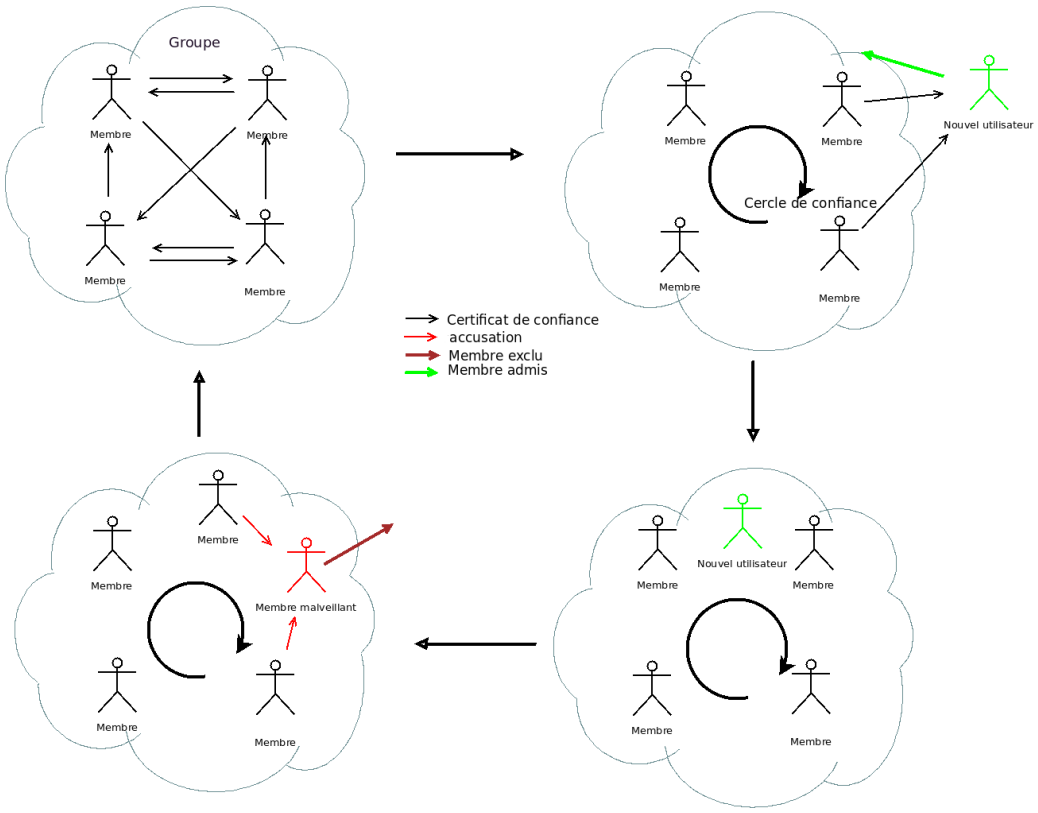

FIGURE 4.2 – Aperçu du modèle

7. Les accusations seront combinées afin de prendre une décision (soit une exclusion ou bien une punition).

Notre proposition permet d'assurer le partage des fichiers dans les réseaux sociaux en toute sécurité c'est-à-dire, éviter la prolifération des fichier infecté et la possibilité de remonter à la source, exclure les membres malveillants (les membres perturbateurs du réseau et les free riders [1]) et enfin assurer la disponibilité des ressources. Pour ce faire, un schéma de cryptographie à seuil est intégré au graphe de confiance. Après la création du groupe chaque nœud i reçoit une part privée S_i de la clé privée du groupe, notée k^{-1}. Qui est maintenu secrètement par le système. Grâce à ces parts de clés les membres peuvent donner leurs avis lors des différentes décisions concernant le groupe, qu'il s'agisse d'adhésion d'un nouveau membre ou bien d'exclusion. Pour ce faire, chaque

1. passagers clandestins

nœud utilise sa part privée pour délivrer des certificats de confiance partiels ou bien délivrer des accusations partielles.

Nous illustrons dans la figure [4.3] un modèle du graphe de confiance partielle. L'existence d'un arc qui relie i vers j signifie que l'utilisateur i fait confiance à l'utilisateur j et lui délivre un certificat de confiance. La génération de ces derniers permet donc la création d'un graphe particulier qu'on appellera graphe de confiance partielle. L'existence d'un chemin qui relie i vers l signifie qu'il y a une chaîne de confiance partielle entre l'utilisateur i et l'utilisateur l, qui est représenté par une chaîne de certificats partiels.

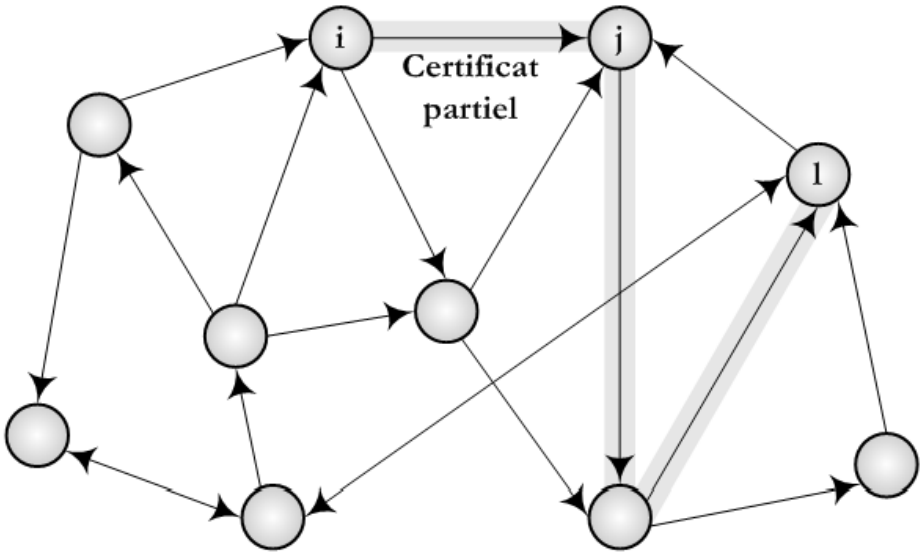

FIGURE 4.3 – Graphe de confiance partielle

4.2.2 La création du groupe

La création du groupe est la phase initiale du système. un groupe contient un ensemble de membre. Chaque membre du groupe va recevoir sa propre part privée qui sera générée par le système que ce dernier créera ensuite un polynôme de degré $k-1$ tel que $F(x) = a_0 + a_1x + .. + a_{k-1}x^{k-1}$ en mettant $a_0 = K^{-1}$. A l'aide de ce polynôme le système calcul

pour chaque nœud i sa part privée S_i tel que $S_i = F(i)$.

Grâce a leurs parts privées, les utilisateurs génèrent les certificats partiels à ceux qu'ils leurs font confiance. cette dernière action permettra de construire le graphe de confiance partiel comme montré dans la figure [4.4].

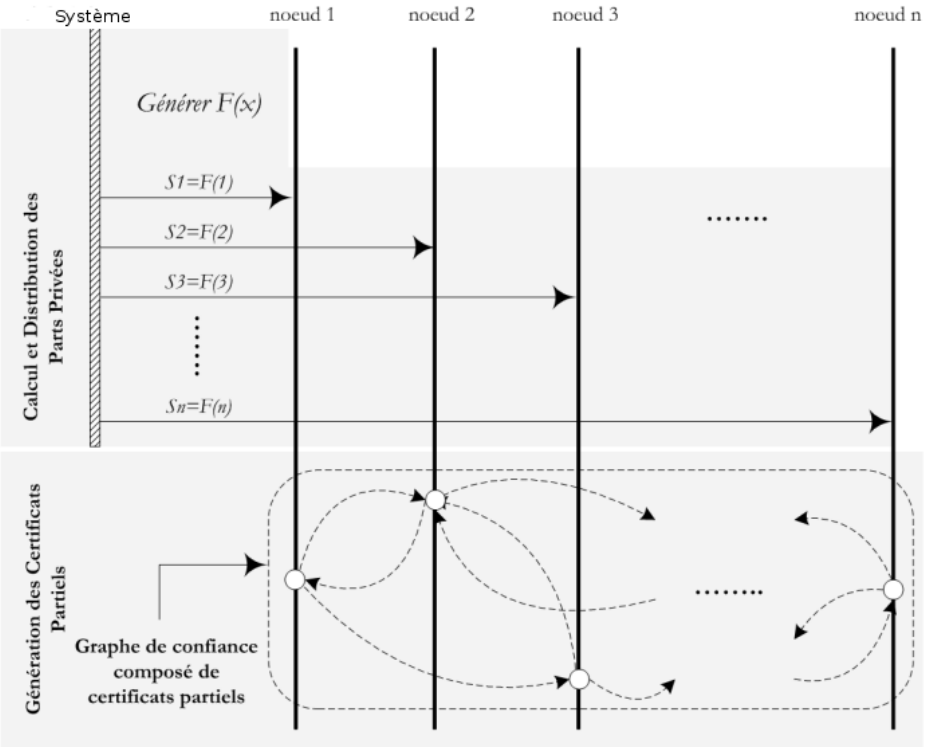

FIGURE 4.4 – Phase initiale

4.2.3 L'adhésion d'un utilisateur

Le processus d'adhésion du groupe se fait en deux parties : la délivrance de l'adhésion et la délivrance des parts privées.

4.2.3.1 La déliverance de l'adhésion

Lorsqu'un nouveau membre souhaite adhérer un groupe, il envoie une requête au système que celui-ci va diffuser à tous les membres du groupe afin de récolter un nombre suffisant de certificats partiels. Si un nœud membre estime que le nouveau nœud est digne

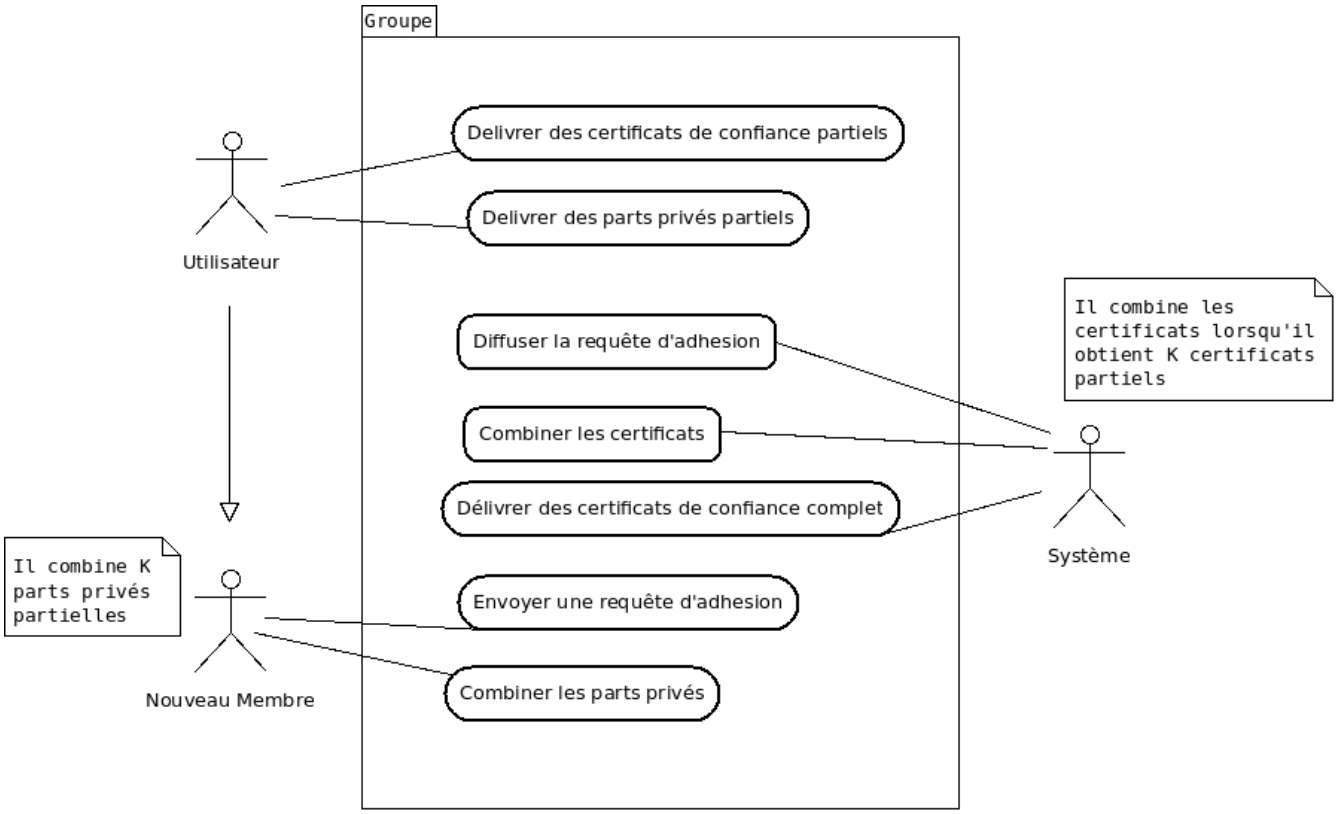

FIGURE 4.5 – Cas d'utilisation du processus d'adhésion

de confiance, il lui génère un certificat partiel de confiance. Cette opération est répétée jusqu'à ce que le système récolte k certificats partiels indépendants, ensuite il procède à les combiner. Si la combinaison échoue ou le nœud n'arrive pas à collecter k certificats partiels, il transmet un message d'échec au nouveau nœud. Autrement, il transmet le certificat complet (signé par la clé privée du système) au nouveau nœud. Ce certificat est la preuve d'adhésion qui sera utilisée pour obtenir une part privée.

4.2.3.2 La délivrance de la part privée

Après avoir eu l'accord d'adhésion, le nouveau membre (qu'on appellera i) doit obtenir sa part privée pour pouvoir participer à la gestion des certificats. Celui-ci doit envoyer des messages qui vont contenir ses parts privées partielles qui seront transmises confidentiellement et diffuse sa requête signée par sa clé privée en joignant son certificat délivré

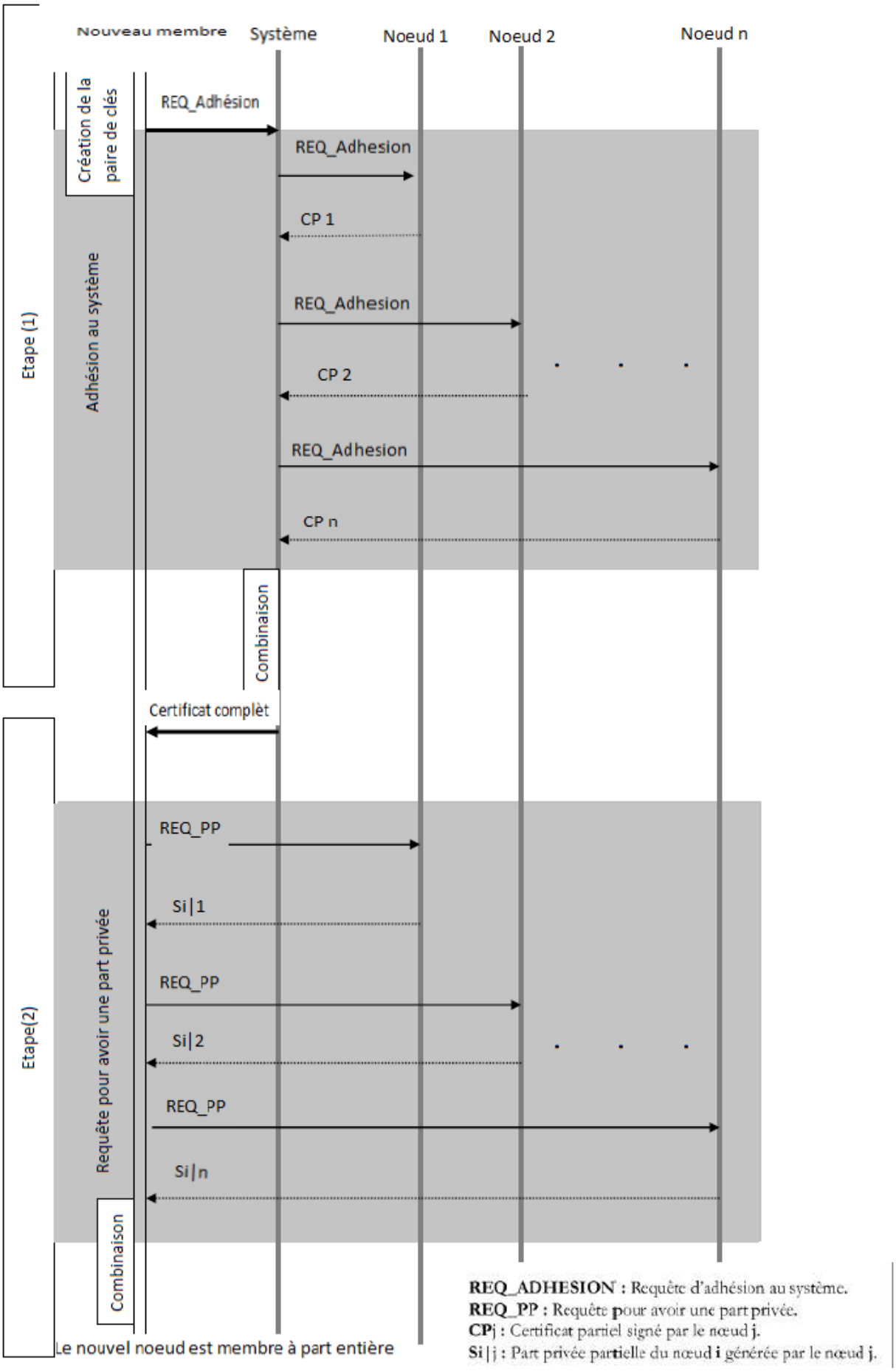

FIGURE 4.6 – Adhésion d'un utilisateur

lors de l'étape de la décision d'adhésion. Lorqu'un nœud membre j reçoit la requête, il vérifie la validité du certificat et la signature du message afin d'authenitifier l'expediteur de la requête, et ensuite il lui délivre une part privée partielle $S_{i/j}$. Après avoir reçu au moins k parts privées partielles, il les combine pour construire sa propre part privée S_i. A ce moment-là, le nœud i devient membre du système.

4.2.4 L'accusation d'un utilisateur

Nous avons recours à l'exclusion d'un utilisateur quand celui-ci devient malveillant. Dans notre cas, nous mesurons la malveillance selon certains critères qui sont :

- Le partage de fausses informations : Il est possible que certains utilisateurs mettent en ligne des informations qui ne sont pas celle qu'ils prétendent être (un nouveau film par exemple). ceci engendre une saturation dans le réseau qui deviendra par la suite vulnérable.
- L'égoïsme : Un nœuds est dit égoïste quand il ne contribue pas à l'enrichissement des ressources dans le réseau. si tous les nœuds se comportent de la sorte, cela provoquera l'alourdissement et la saturation du réseau.

Dans le premier comme dans le second cas, la procédure d'exclusion se fait en deux étapes : délivrance de la décision d'exclusion et le retrait des droits.

4.2.4.1 La délivrance de la décision d'exclusion

Lorsqu'un nœud i juge qu'un nœud j n'est plus digne de confiance, il diffuse dans le réseau une accusation partielle signée avec sa part privée et envoi par la suite une requête de suspicion au système pour que l'information soit diffusée à tous les membres du groupe. Si un nœud estime que j n'est pas digne de confiance, il lui génère une accusation partielle.

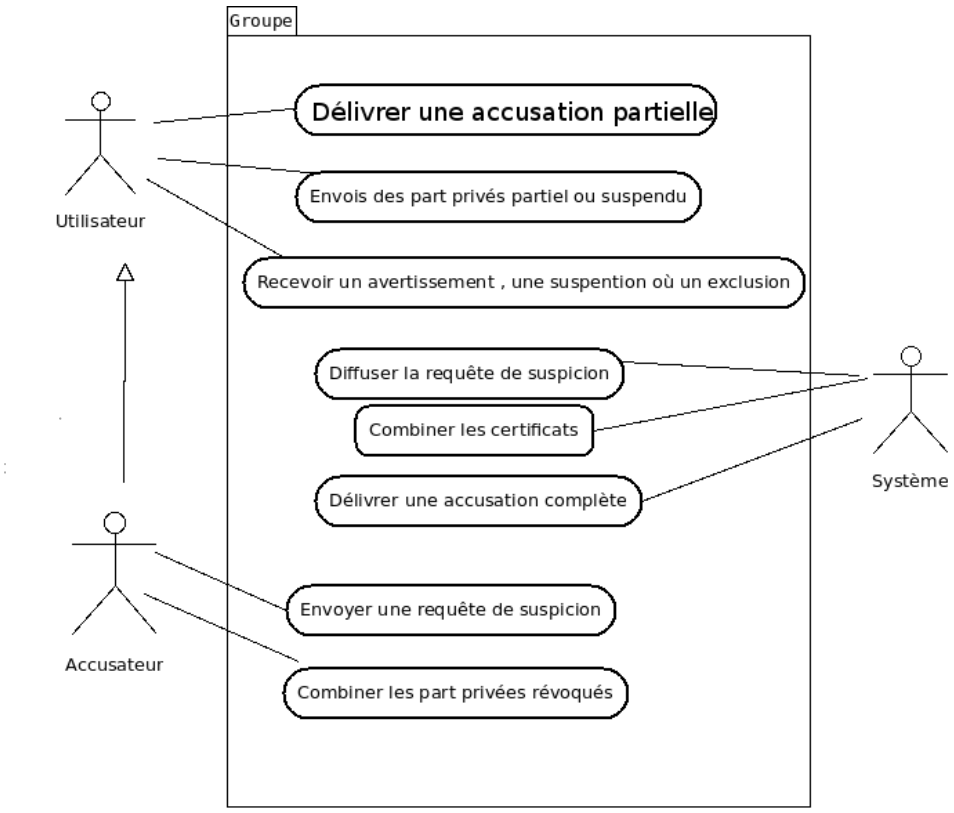

FIGURE 4.7 – Cas d'utilisation du processus d'exclusion

Cette opération est répétée jusqu'à ce que le système récolte k accusations partielles indépendantes, ensuite ils les envoient au système qui se chargera de les combiner. Si la combinaison échoue ou le nœud accusateur n'arrive pas à collecter k accusations partielles, le système alors transmettra un message d'échec. Dans le cas contraire il transmet une accusation complète (signée par la clé privée du système) afin de décider du sort de j.

4.2.4.2 Le retrait des droits

Une accusation partielle contient un champ qui permet de juger de degré de malveillance de l'utilisateur. Chaque membre qui délivre cette accusation partielle porte un jugement personnel à propos de l'utilisateur suspecté. Après avoir combiné ces accusations, le système applique des règles d'arbitrage et délivre l'accusation complète et le jugement décidé. Les cas sont les suivants :

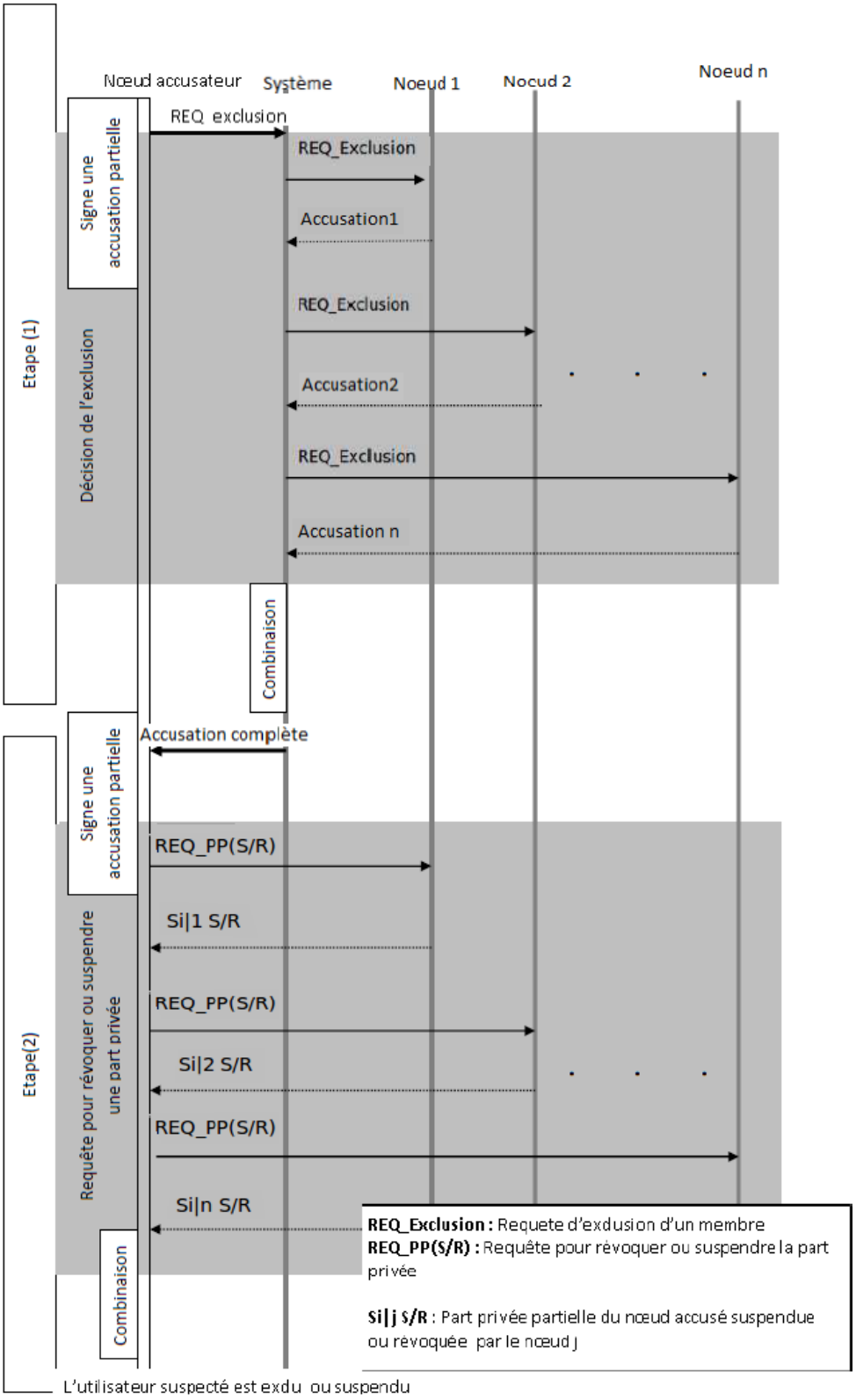

FIGURE 4.8 – Exclusion d'un utilisateur

- **Degré de malveillance faible :**

  Le système envoie un message d'avertissement et s'il dépasse 3 avertissements il sera exclu du groupe.

- **Degré de malveillance moyenne :**

  Le nœud suspecté sera suspendu. Pour ce faire, le nœud accusateur envoie un message aux autres nœuds du réseau afin de récolter les parts privées partielles suspendues qui seront transmises confidentiellement. Lorsqu'un nœud membre j reçoit sa requête, il vérifie la validité de l'accusation et la signature du message afin d'authentifier l'expéditeur de la requête, et ensuite il lui délivre la part privée partielle de l'accusé suspendu. Après avoir reçu au moins k part privées suspendu, l'accusateur les combine pour suspendre la part privée S_i. Et donc, l'utilisateur sera exclu temporairement.

- **Degré de malveillance fort :**

  Le nœud suspecté sera exclu de manière définitive, nous procéderons de la même manière que la suspension sauf que dans ce cas les parts de clés partielles seront révoquées. En combinant celle-ci, la clé de l'accusé est révoquée et il sera exclu définitivement du système.

Cependant, il arrive que certains utilisateurs lancent des accusations mensongères afin de perturber le réseau. Si un utilisateur envoie plusieurs fausses accusations, c'est-à-dire que celui-ci n'obtient pas K accusation partielles, alors il recevra un avertissement en premier lieu et ensuite il sera accusé. La figure suivante représente le processus d'exclusion dans le groupe.

Conclusion

Nous avons proposé dans ce chapitre un modèle de confiance où le service de certification est basé sur la collaboration du nœud au sein du réseau. La cryptographie à seuil et le graphe de confiance permettent la gestion des certificats indépendamment d'une autorité de certification centrale et résistent aux utilisateurs malveillants qui perturbent le système, mais aussi ils assurent la disponibilité du processus d'authentification.

Dans le prochain chapitre, nous allons démontrer comment notre modèle se comporte vis-à-vis des attaques citées dans le chapitre précédent.

Chapitre

5 Le comportement de notre modèle vis à vis des attaques

Sommaire

Introduction **71**

5.1 Empoisonnement des fichier **72**

5.2 Le Free Riding **73**

5.3 L'attaques déni de service **74**

5.4 L'attaque sybil **75**

5.5 La propagation des vers **75**

5.6 L'attaque eclipse **75**

Conclusion **76**

Nous avons vu dans le chapitre 3, les différentes attaques contre les réseaux P2P. En effet Il existe deux catégories d'attaques : les attaques de données et les attaques d'infrastructure. Après avoir défini l'architecture de notre modèle de confiance, nous allons procéder à la description du comportement de notre modèle contre ses attaques.

5.1 Empoisonnement des fichier

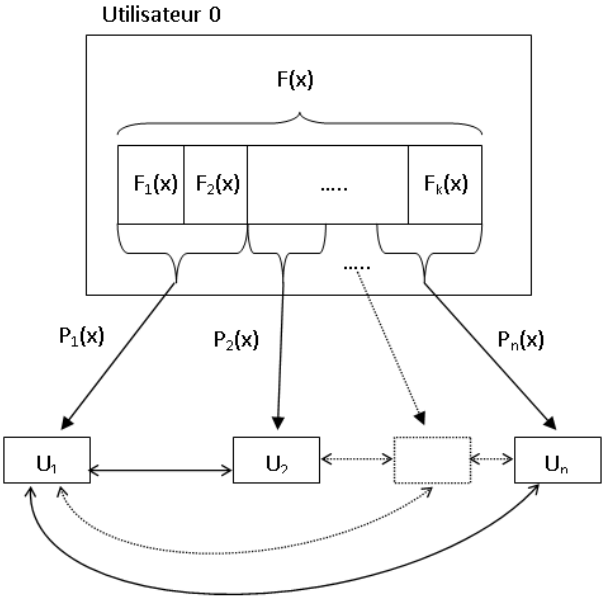

FIGURE 5.1 – Echange de fichier P2P

La contre mesure pour cette attaque est de vérifier la provenance des fichiers usurpés afin d'exclure celui qui est à l'origine du problème.

Lemme1 :

Pour tout fichier F nous pouvons remonter à la source x.

Preuve :

Lorsqu'un utilisateur x partage un fichier $F(x)$, ce dernier va être segmenté en plusieurs segments $F_1(x)$, $F_2(x)$ ... $F_k(x)$. Chaque utilisateur alors télécharge une partie P des segments et récupère le reste $(1 - p)$ chez les autres utilisateurs (voir figure 5.1). Dans notre modèle chaque fichier f est rattaché à l'identité de l'utilisateur x qui l'a mis en téléchargement dans le groupe. Si le fichier $F(x)$ circule dans le réseau est empoisonné, l'identité du propriétaire est localisée. En remontant a sa source.

5.2 Le Free Riding

Lemme 2 :

$$\forall x, R(D_x(nd), U_x(nu)) \geq R_{moy} \Rightarrow A(x) \tag{5.1}$$

x : Utilisateur.

nd :Nombre de fichiers téléchargés.

nu :Nombre de fichiers partagés.

R_{moy} : Rapport moyen des fichier partagés et téléchargés dans le réseau.

$R(D_x(nd), U_x(nu))$: Rapport des fichiers partagés et téléchargés par l'utilisateur x .

D_x : Fonction de de retribution pour l'utilisateur x.

U_x : Fonction de contribution pour l'utilisateur x.

$A(x)$: Accusation de x.

Preuve :

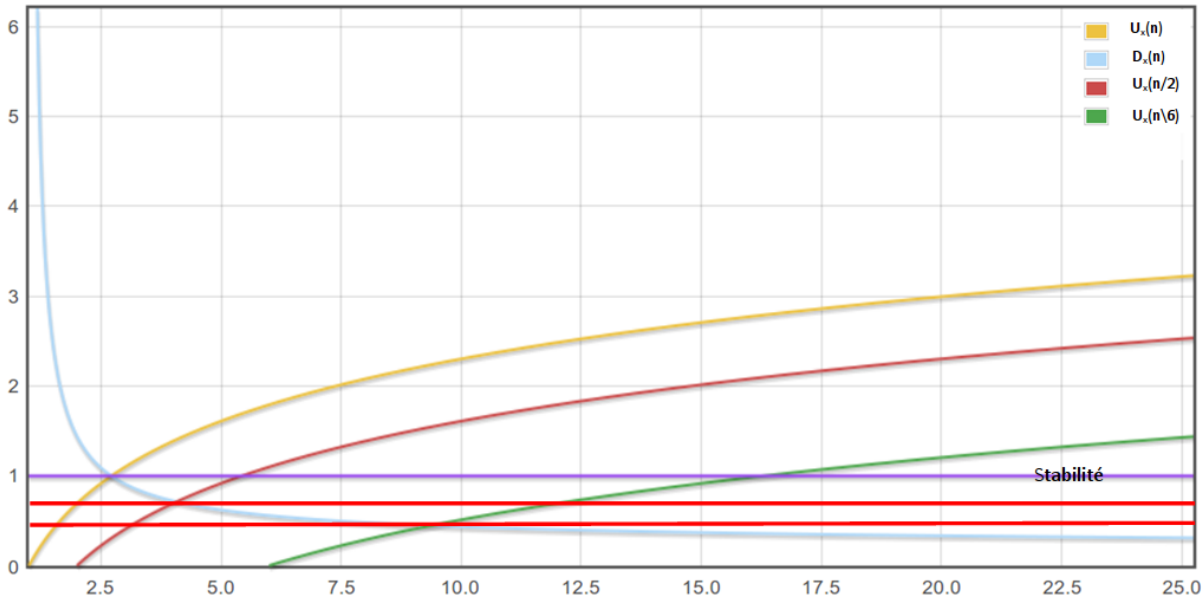

FIGURE 5.2 – Exemple de stabilité d'un partage

Supposons qu'à chaque fois qu'un utilisateur x partage un nombre de fichier nu son ratio

augmentera de $U_x = ln(nu)$ et dans le cas contraire son ratio diminuera de $D_x = \frac{1}{ln(nd)}$. Supposons aussi que pour avoir un partage stable il faut que $nu = nd$ ce qui nous donne $R_{moy} = D_x(nd)U_x(nu) = 1$ comme montré dans la figure si dessus.
Nous remarquons que plus $nu \leq nd$ plus $R(D_x(nd), U_x(nu)) \leq R_{moy}$. Les lignes rouges représentent le rapport des fichiers partagés et téléchargés par des utilisateurs dans le cas ou $nu = nd/2$ et $nu = nd/6$. Dans ce cas nous remarquons que le partage est instable. Afin de résoudre ce problème, nous excluons les membres susceptibles de nuire à la stabilité du partage des ressources.

5.3 L'attaques déni de service

Il est difficile de prévoir l'attaque DoS que ce soit dans les réseaux Internet traditionnels ou dans les réseaux P2P. Certains utilisateurs malveillants partagent de fausses informations où envoient plusieurs requêtes que ça soit d'adhésion ou d'exclusion au système. Ceci permet de provoquer un déni de service.

Suivant le **lemme1** nous pouvons récupérer l'identité de l'émetteur de fausses informations ceci engendrera son accusation puis son exclusion.

Quant aux requêtes, nous avons prévu de limiter leur nombre selon un délai. Par exemple, un utilisateur ne peut pas envoyer plus de 5 accusations par jour.

Certes, ces prévoyances n'empêchent pas que cette attaque ait lieu, mais ceux-ci réduiront considérablement la possibilité de l'arrivée de celle-ci.

5.4 L'attaque sybil

Lemme 3

Il est impossible pour un utilisateur d'avoir plusieurs identités dans un modèle à base de certifications.

Preuve

Douceur [16] a prouvé que la certification de confiance est la seule approche qui a le potentiel d'éliminer complètement les attaques Sybil. Chaque utilisateur possède un couple de clés (Privé /Publique) qu'il utilise pour générer des certificats partiels de confiance. L'auteur prouve qu'il est difficile pour l'attaquant d'avoir plusieurs clés publiques avec la même clé privée.

5.5 La propagation des vers

Pour diminuer l'efficacité du ver, nous pouvons éviter les réseaux hybrides (qui contiennent des supernœuds). Ces supernœuds fournissent une augmentation du taux de propagation du vers en raison de leur degré de connectivité élevé. Notre modèle ne possède pas de supernœuds. La propagation des vers sera éventuellement réduite.

5.6 L'attaque eclipse

Lemme4

Un utilisateur qui possède des liens beaucoup plus que la moyennent peut être accusé.

Preuve

Selon les auteurs de [6], lorsqu'un nœud possède des liens beaucoup plus que la moyennent

il pourrait être un montage éclipse. Alors si les membres du groupe effectuent un audit régulièrement les nœuds malicieux seront découverts et exclus.

Le tableau ci-dessous résume le comportement de notre modèle vis-à-vis des attaques citées précédemment.

Attaques	Comportement de notre modèle
Déni de service	Le nombre de requêtes d'accusation et de demande d'intégration au réseau est réduit, ceci réduit la possibilité d'avoir un déni de service.
Botnet	Ce modèle ne résout pas cette attaque
Propagation des vers	L'absence de supernœuds réduit considérablement la propagation des vers
Attaque Sybil	Les modèles à base de certifications éliminent les attaques sybil.
Attaque eclipse	Les membres du groupe effectuent un audit régulièrement les nœuds malicieux seront découverts et exclus
Empoisonnement des fichiers	Les membres du groupe vérifient la provenance des fichiers usurpés et sanctionnent le membre perturbateur.
Free riding	Lorsqu'un pair se comporte de manière égoïste, l'ensemble des membres du groupe génère une accusation pour l'exclure.

TABLE 5.1 – Comportement du modèle vis-à-vis des attaques

Conclusion

Dans ce présent chapitre, nous avons démontré comment notre modèle réagit contre quelques attaques définies dans le chapitre 3. Pour ce faire, nous avons utilisé des lemmes afin d'expliquer les contre-mesures et les défenses contre les attaques empoisonnement de fichier, free riding , déni de service , sybil , éclipse et la propagation des vers. Cette démonstration nous prouve que notre modèle de confiance est robuste et assure les principes de la sécurité.

Conclusion générale

Les systèmes Pair-à-Pair sont aujourd'hui de plus en plus présents dans le monde informatique, soit pour les simples utilisateurs à travers les réseaux d'échanges de fichiers, de messageries instantanées de voix sur IP, ou encore au sein des entreprises, pour la mise en place d'applications réparties comme le partage d'agendas, etc. D'une manière générale, les P2P actuels souffrent de nombreux problèmes tels que les fichiers truqués, virus et autres menaces. De plus leurs architectures ne sont pas totalement décentralisées, ceux-ci constituent une véritable faiblesse.

Dans ce mémoire, nous avons étudié les systèmes P2P et leurs différentes classes particulièrement les P2P structurés et non structurés. Nous avons présenté une synthèse des différentes attaques qui peuvent être menées sur ce type de réseau et les différentes contremesures proposées dans la littérature. Afin de contrer ces attaques, nous avons proposé une architecture d'un modèle de confiance pour le partage P2P au sein d'un groupe dans un réseau social.

Ce modèle de confiance est basé sur la certification autonome qui repose sur une approche collaborative de tous les nœuds pour une gestion complètement distribuée des certificats. Le modèle est basé à la fois sur les graphes de confiance et la cryptographie à seuil, ce

qui nous a permis de définir la notion de graphe de confiance partielle. Ceci vient du fait que les nœuds du réseau ont un pouvoir limité de certification, de telle sorte que si un nœud A estime que B est digne de confiance, il lui délivre seulement un certificat partiel. Ce certificat ne sera pris en considération que seulement si le nœud B est estimé digne de confiance d'au moins k nœuds. Ainsi, pour authentifier le nœud B, on combine l'ensemble des certificats partiels qu'on lui a délivrés pour avoir un certificat complet. Grace à ce dernier, le nœud B adhérera au groupe.

Pour se débarrasser des nœuds malveillants, nous procéderons à la phase de l'exclusion de ces derniers qui est basée sur le même raisonnement que l'adhésion. Si un nœud A estime que M n'est pas digne de confiance, il lui délivre une accusation partielle. Cette accusation ne sera prise en considération que seulement si le nœud M est accusé par au moins k nœuds.

Pour évaluer la pertinence de notre solution, nous avons démontré que notre modèle est robuste envers quelques attaques présentées précédemment par rapport aux défenses existantes.

En perspectives, nous envisagerons de réaliser un simulateur en utilisant une plate-forme d'un réseau P2P pour implémenter notre modèle de confiance. Nous envisageons aussi dans un travail futur, d'étendre nos études à d'autres types d'attaques afin d'élaborer une architecture plus robuste pour les réseaux P2P purs.
L'utilisation du P2P comporte donc de nombreux risques, mais parallèlement à ces attaques, les défenses et protections se développent. Ainsi, face à l'amélioration du filtrage et de l'identification des pairs, les P2P cryptés ou encore anonymes se développent et prennent de l'ampleur pour constituer les P2P de demain.

Bibliographie

[1] *EmuleProject*. edonkey 2000, DECEMBER 2010.

[2] RFC 2828. Internet security glossary. 2000.

[3] B. Rexroad A. Karasaridis and D. Hoeflin. Wide-scale botnet detection and characterization. *In USENIX Workshop on Hot Topics in Understanding Botnets (HotBots' 07)*, 2007.

[4] M. Castro P. Druschel A. Singh and A. Rowstron. Defending against eclipse attacks on overlay networks. *In Proceedings of the 11th workshop on ACM SIGOPS European workshop, page 21. ACM*, 2004.

[5] T.W. Ngan P. Druschel A. Singh and D.S. Wallach. Eclipse attacks on overlay networks : Threats and defenses. *In IEEE INFOCOM, pages 1–12*, 2006.

[6] T.W. Ngan P. Druschel A. Singh and D.S. Wallach. Eclipse attacks on overlay networks : Threats and defenses. *In IEEE INFOCOM*, 2006.

[7] National Security Agency. Secure hash algorithm. *Federal Information Processing Standard du National Institute of Standards and Technology*, 1993.

[8] S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys (CSUR)*, 2004.

[9] D. Agrawal C. Buragohain and S. Suri. A game theoretic framework for incentives in p2p systems. *In Proceedings of the 3rd International Conference on Peer-to-Peer Computing, volume 56*, 2003.

[10] B. Cohen. *Incentives build robustness in bittorrent.* In Workshop on Economics of Peer-to-Peer systems, 2003.

[11] F. Cornelli, S. di Vimercati S. Paraboschi E. Damiani, and P. Samarati. Implementing a reputation-aware gnutella servent. *Web Engineering and Peer-to-Peer Computing*, 2010.

[12] Cydoor. Cydoor spyware. *Technical report, Cydoor Technologies, Inc.*, 2009.

[13] E. Knightly A. Kuzmanovic I. Stoica D. Dumitriu and W. Zwaenepoel. Denial-of-service resilience in peer-to-peer file sharing systems. *ACM SIGMETRICS Performance Evaluation Review*, 2005.

[14] A.P. de Barros A. Fucs and V. Pereira. New botnets trends and threats. *Web document, NA.*

[15] M.N. Docs. Technology overview of mojo nation. *Internet Citation*, February 2000.

[16] J. Douceur. The sybil attack. *Peer-to-Peer Systems, pages 251–260*, 2002.

[17] P. Druschel and A. Rowstron. Past : A large-scale, persistent peer-to-peer storage utility. *In Proc. HotOS VIII*, 2003.

[18] T. Dubendorfer and A. Wagner. Past and future internet disasters : Ddos attacks. 2003.

[19] K. Anagnostakis E. Athanasopoulos and E. Markatos. Misusing unstructured p2p systems to perform dos attacks : The network that never forgets. *In Applied Cryptography and Network Security.*

[20] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *Springer Verlag*, 1998.

[21] D. Bindel Y. Chen S. Czerwinski P. Eaton D. Geels R. Gummadi S. Rhea H. Weatherspoon C. Wells et al. Oceanstore J. Kubiatowicz. An architecture for global-scale persistent storage. *ACM SIGARCH Computer Architecture News*, 2000.

[22] C. Lesniewski-Laas M.F. Kaashoek G. Danezis and R. Anderson. Sybilresistant x dht routing. *Computer Security–ESORICS 2005, pages 305–318*, 2005.

[23] Sinan HATAHET. *Security of Unstructured P2P Systems*. PhD thesis, University of Technology of Compiegne.

[24] O. Sandberg B. Wiley I. Clarke and T. Hong. *Freenet : A distributed anonymous information storage and retrieval system. In Designing Privacy Enhancing Technologies*. Springer, DECEMBER 2009.

[25] R. Morris D. Karger M.F. Kaashoek I. Stoica and H. Balakrishnan. Chord : A scalable peer-to-peer lookup service for internet applications. *In Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, 2001.

[26] C. Leckie J. Chan and T. Peng. Hitlist worm detection using source ipaddress history. *In Proceedings of Australian Telecommunication Networks and Applications Conference*, 2006.

[27] R. Kumar Y. Xi J. Liang and KW Ross. Pollution in p2p file sharing systems. *In Proceedings IEEE INFOCOM*, 2005.

[28] M. Jelasity and V. Bilicki. Towards automated detection of peer-to-peer botnets : On the limits of local approaches. 2009.

[29] M. Gjoka K. El Defrawy and A. Markopoulou. Bottorrent : Misusing bittorrent to launch ddos attacks. *In Proc. 3rd Workshop on Steps to Reducing Unwanted Traffic on the Internet, SRUTI*, 2007.

[30] E. Koutrouli and A. Tsalgatidou. Reputation-based trust systems for p2p applications : design issues and comparison framework. *trust and Privacy in Digital Business*, 2006.

[31] Z. Haas L. Zhou. Securing ad hoc networks. *IEEE Networks*, 1999.

[32] B.N. Levine, C. Shields, and N.B. Margolin. A survey of solutions to the sybil attack. *University of Massachusetts Amherst, Amherst*, 2006.

[33] P. Druschel A. Ganesh A. Rowstron M. Castro and D.S. Wallach. Secure routing for structured peer-to-peer overlay networks. *ACM SIGOPS Operating Systems Review,*, 2002.

[34] P. Druschel A. Ganesh A. Rowstron M. Castro and D.S. Wallach. Secure routing for structured peer-to-peer overlay networks. *ACM SIGOPS Operating Systems Review*, 2002.

[35] C. Papadimitriou J. Chuang M. Feldman and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 2006.

[36] E. Sit J. Cates M. Freedman and R. Morris. Introducing tarzan, a peer- to-peer anonymizing network layer. *Peer-to-Peer Systems*, 2002.

[37] I. Foster M. Ripeanu and A. Iamnitchi. *Mapping the gnutella network :Properties of large-scale peer-to-peer systems and implications for system design.* IEEE Internet Computing jjournal, DECEMBER 2002.

[38] A.D. Rubin M. Waldman and L.F. Cranor. Publius : A robust, tamperevident, censorship-resistant web publishing system. *In Proceedings of the9th conference on USENIX Security Symposium-Volume 9, page 5. USENIX Association*, 2007.

[39] Patrick MARLIER. Sécurité du peer-to-peer. www.labo-asso.com.

[40] J. Morparia. *Peer-to-Peer Botnets : Analysis and Detection.* PhD thesis, San Jose State University, 2008.

[41] A.S. Weigend N. Christin and J. Chuang. Content availability, pollution and poisoning in file sharing peer-to-peer networks. *In Proceedings of the 6th ACM conference on Electronic commerce*, 2005.

[42] A.S. Weigend N. Christin and J. Chuang. Content availability, pollution and poisoning in file sharing peer-to-peer networks. *In Proceedings of the 6th ACM conference on Electronic commerce*, 2005.

[43] Y. Charlinet N. Khiat and N. Agoulmine. The emerging threat of peer-to-peer worms. *In Proc. 1st EEE Workshop on Monitoring, Attack Detectionand Mitigation*, 2006.

[44] N. Naoumov and K. Ross. Exploiting p2p systems for ddos attacks. *In Proceedings of the 1st international conference on Scalable information systems*, 2006.

[45] National Bureau of Standards. Data encryption standard (des). *Federal Information Processing Standards Publication, National Technical Information Service , Springfield*, 1977.

[46] National Institute of Standards and Technology (AES). Specification for the advanced encryption standard. *FIPS 197*, 2001.

[47] B. Pretre. Attacks on peer-to-peer networks. 2005.

[48] N. Mathewson R. Dingledine and P. Syverson. Tor : The second-generation onion router. *In Proceedings of the 13th conference on USENIX Security Symposium-Volume*, 2004.

[49] A. Shamir L. Adleman R. Rivest. A method for obtaining digital signatures and public-key cryptosystems. *Communication of ACM*, 1978.

[50] A. Ramachandran, N. Feamster, and D. Dagon. Revealing botnet membership using dnsbl counter-intelligence. *In USENIX 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI'06)*, 2006.

[51] M. Ripeanu and I. Foster. *Peer-to-peer architecture case study : Gnutella network.* University of Chicago ,Chicago, 2001.

[52] C. Gross P. Leca S. Aumont. Certificats x509 et infrastructures de gestion de clés. *CNRS/UREC*, 2001.

[53] V. Paxson S. Staniford and N. Weaver. *How to 0wn the internet in your spare time.* 2002.

[54] M.T. Schlosser S.D. Kamvar and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. *In Proceedings of the 12th international conference on World Wide Web*, 2003.

[55] A. Shamir. How to share a secret. *Communication of the ACM*, 1979.

[56] A. Shamir. Identity-based cryptosystems and signature schemes. *Lecture Notes in Computer Science, Berlin, Springer-Verlag*, 1985.

[57] K.C. Sia. Ddos vulnerability analysis of bit-torrent protocol. *University of California, Los Angeles.*

[58] E. Sit and R. Morris. Security considerations for peer-to-peer distributedhash tables. *Peer-to-Peer Systems, pages 261–269*, 2002.

[59] Ralf Steinmetz. *Peer-to-peer systems and Applications*. Springer, 2005.

[60] G. Suryanarayana and R.N. Taylor. A survey of trust management and resource discovery technologies. *in peer-to-peer applications*.

[61] C. Boyer S. Chellappan W. Yu and D. Xuan. Peer-to-peer system-based active worm attacks : Modeling and analysis. 2005.

[62] D. Wallach. A survey of peer-to-peer security issues. *Software Security—Theories and Systems, pages 253–258*, 2003.

[63] X. Wang, D. Feng, X. Lai, and H. Yu. Cx. wang, d. feng, x. lai, h. yuollisions for hash functions md4, md5. *Institute of Software, Chinese Academy of Sciences*, 2004.

[64] J. Massey. X. Lai. Markov ciphers and differential cryptanalysis. *Cryptology - EUROCRYPT'91, Springer-Verlag*, 1991.

[65] L. Zhou, F. McSherry N. Immorlica M. Costa L. Zhang, and S. Chien. Afirst look at peer-to-peer worms : Threats and defenses. *Peer-to-Peer Systems IV*, 2005.

Résumé

Les réseaux P2P représentent aujourd'hui une partie considérable des échanges sur Internet, car qu'ils offrent principalement aux utilisateurs du monde entier un moyen rapide et efficace pour le partage des ressources. Les réseaux P2P offrent plusieurs avantages tels qu'un meilleur passage à l'échelle, un meilleur rendement et une meilleure qualité de service. De tels réseaux sont plus vulnérables aux attaques que les réseaux client /serveur classiques, notamment en raison de l'absence d'infrastructure fixe et de centralisation. Nous pouvons nous attendre à des usurpations et de propagations de virus par des utilisateurs malveillants. Par ailleurs, les entités doivent détecter les nœuds perturbateurs. Ceci nécessite la mise au point d'un modèle de confiance qui permet de définir qui fait confiance à qui et comment. Le travail, réalisé dans le cadre de ce mémoire, porte sur un modèle de confiance à base de certification pour le partage P2P dans les réseaux sociaux.

Mots clés : Réseaux P2P, modèle de confiance, certificat, groupe, accusation.

Abstract

P2P networks now account for a significant portion of the exchanges on the Internet, mainly because they provide to the users worldwide with quick and effective sharing resources. P2P networks offer several advantages such as better scalability, more efficient and better quality of service. These networks are more vulnerable to attacks then client / server ones , especially because of the lack of fixed infrastructure and centralized architecture. One can expect to theft and virus outbreaks by malicious users. In addition, entities must detect disruptive nodes. This requires the development of a trust model that defines who trusts who and how. The work done within the framework of this thesis deals with a trust model based certification for P2P sharing in social networks.

Keywords: P2P networks, trust model, certificate, group ,accusation.