Université de Montréal

Vers une plateforme holistique de protection de la vie privée dans les services géodépendants

par
Zakaria Sahnoune

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en Informatique

Avril, 2018

RÉSUMÉ

Les services géodépendants (*LBS : Location-Based Services*) sont présents dans la plupart de nos activités quotidiennes, ils représentent l'ensemble des services en ligne qui fournissent des informations basées sur la géolocalisation des individus. On peut les retrouver dans plusieurs domaines tels que les soins médicaux, le divertissement, le transport, et de nombreuses autres activités. En outre, avec leur présence dans presque toutes les tâches quotidiennes, leur utilité ne peut être négligée, ainsi que leur impact sur la façon dont les données des individus sont traitées n'est plus la même.

De plus, avec la croissance du nombre de propriétaires de dispositifs mobiles, il est devenu plus facile de localiser un individu, rendant ainsi presque inévitables les problèmes liés à la confidentialité. Par conséquent, des solutions plus sévères sont fortement nécessaires afin de gérer les problèmes de confidentialité et de conserver l'utilité de LBS.

Nous présentons dans cette thèse une recherche qui porte sur la protection de la vie privée des utilisateurs dans un LBS tout en gardant un maximum d'utilité des services. Autre que l'état de l'art et les fondements théoriques, ce travail est divisé en trois parties principales, chacune d'elles décrit un ensemble des composants connexes de la plateforme proposée, nommée *Deloc*.

- **Concept de délégation.** Nous proposons un mécanisme collaboratif où chaque utilisateur participe à la protection des autres utilisateurs sans avoir besoin de transformer ses coordonnées géographiques ni de perdre l'utilité du service. Nous évaluons également le concept sur des données réelles dans un environnement simulé avec des paramètres ajustables, et nous démontrons ses garanties de confidentialité, d'efficacité, et d'utilité dans la plupart des applications LBS actuelles.

- **Quantification des risques et confiance.** La dépendance du concept de délégation basée sur les utilisateurs LBS peut impliquer des risques de confidentialité, et éventuellement des violations de protection des données. Nous proposons dans

un premier lieu un modèle basé sur la théorie de l'information mutuelle qui sert à évaluer les risques associés à l'utilisation du concept de délégation. Par la suite nous présentons un modèle basé sur les chaines de Markov qui aide à estimer le niveau de confiance pour chacun des collaborateurs. Nous discutons également l'applicabilité des deux modèles et leur validation théorique et empirique.

- **Métrique de confidentialité pour les systèmes collaboratifs.** L'absence d'une métrique conçue pour les systèmes collaboratifs est notre principale motivation. Nous proposons une métrique de confidentialité en nous basant sur la théorie des ensembles flous. Les modèles de l'attaquant, de la confidentialité, et de l'utilité sont les trois modèles utilisés pour définir la métrique de confidentialité nommée δ-*fuzziness*. Nous discutons également la validation de la métrique, et son efficacité de mesure dans le cas des mécanismes collaboratifs, notamment dans la plateforme *Deloc*.

En conclusion, nous proposons une plateforme collaborative de protection de la confidentialité des utilisateurs LBS, qui est à la fois efficace, performante, et qui surtout n'affecte pas l'utilité estimée de l'utilisation d'un LBS donné.

Mots clés : Services géodépendants, Mécanisme de préservation de la confidentialité de localisation, Quantification des risques, Mesure de la confiance, Position jumelle, position indicatrice, Réseaux pair-à-pair, Transfert inconscient, Information mutuelle, Logique floue.

ABSTRACT

Location-Based Services (LBS) are present in most of our daily activities, and they repre sent all online services used to provide information based on the location of individuals. These services can be found in several fields such as healthcare, entertainment, transportation, and many other daily activities. Besides, along with their presence in almost all daily tasks, their utility cannot be ignored, and as a result, their impact on how individuals' data are processed is no longer the same.

With smartphone ownership growth, it has become easier to locate an individual, and privacy issues have become almost inescapable. Hence, more severe solutions are strongly required to handle privacy issues while keeping the utility of LBS.

We present in this thesis a research work about protecting the privacy of users in an LBS while keeping maximum utility of the service. In addition to the state of the art and the theoretical background, this work is divided into three main parts when each one describes a set of related components of the proposed framework, called *Deloc*.

We present in this thesis a research into about privacy protection in LBSs while maintaining the maximum utility of these services. In addition to the state of the art and the theoretical background, this work is divided into three main parts, where each one describes a set of related components of the proposed framework named *Deloc*.

- **Delegation concept.** We propose in this part the main concept behind this re search. The goal is to propose a collaborative mechanism where each user participates in the protection of other users without the need for transforming his geographical coordinates, nor losing the utility of the service. We also evaluate the concept on data issued from real-world users in a finely simulated environment with tuneable parameters, and we demonstrate its high guarantees of privacy, efficiency, and utility facing most of the current LBS applications.

- **Quantification of risks and trust.** This part is based on the fact that the depen-

dence of delegation concept on the LBS users may imply privacy issues. This part of the paper discusses two models of quantification and measurement of risks and trust in the context of LBS. First, we propose a model based on the theory of mutual information that is used to assess the risks associated with the use of the delegation concept. Then we discuss a model based on Markov chains that helps to estimate the level of trust for each of the collaborators. We also discuss the applicability of both models and their theoretical and empirical validation.

- **Privacy Metric for Collaborative Systems.** The lack of a metric designed for collaborative systems is the main motivation behind this part. We discuss our proposal for a privacy metric based on the theory of fuzzy sets. We propose three models that each one of them deals with a subset of *Deloc*privacy requirements. The attacker, the privacy, and the utility model are the three models used to define the privacy metric named δ-*fuzziness*. We also discuss the validation of the metric, and its measurement efficiency in the case of collaborative mechanisms, especially *Deloc*.

In conclusion, we propose a collaborative location privacy-preserving framework, which is at the same time efficient, powerful, and which does not affect the estimated utility of using LBSs.

Keywords: Location-based services, Location privacy, location privacy-preserving mechanism, risk quantification, trust measurement, twin positions, telltale positions, P2P networks, oblivious transfer, mutual information, fuzzy logic.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES

LBS	Location-Based Service
	Service géodépendant
LPPM	Location Privacy-Preserving Mechanism
	Mécanisme de préservation de la confidentialité de localisation
GSN	GeoSocial Network
	Réseau Geosocial
POI	Point Of Interest
	Point d'intérêt
P2P	Peer to Peer
	Pair-à-pair
SSID	Service Set Identifier
JSON	JavaScript Object Notation

À mes chers parents qui m'ont montré le chemin.

REMERCIEMENTS

En premier lieu , je louange Dieu le miséricordieux de m'avoir permis de mener ce travail de recherche à terme.

Je remercie infiniment mes parents pour leur soutien interminable. Malgré la distance, leur support continu a fait que ma motivation se renforce et me mène à achever mon doctorat dans les meilleures conditions.

Mes sincères remerciements à ma directrice de recherche Professeure Esma Aïmeur, pour m'avoir donné les bonnes directions à suivre avec confiance et patience. Son support inépuisable depuis mon premier jour au doctorat, ses conseils, remarques et questions ont éclairé le chemin de ce long parcours.

Merci à tous mes collègues du laboratoire HERON pour les discussions intéressantes et la compagnie agréable. Dans cette catégorie, Mouna, Fodè, et Mili m'ont été d'une aide précieuse en écoutant mes péripéties théoriques et en faisant de très inspirantes suggestions.

Merci à mon frère et à ma sœur. On passe trop souvent sous silence le rôle que les rires et les discussions peuvent jouer, spécialement après plusieurs heures devant un écran d'ordinateur.

Pour finir, je réserve mon ultime remerciement à ma femme. Sa patience lors des longues nuits de travail, son attention de tous les jours, son écoute et son humour m'ont été indispensables. Merci d'avoir toujours été là pour moi.

CHAPITRE 1

INTRODUCTION

Cette thèse fait partie d'un projet de recherche plus large mené au sein de notre laboratoire. Le projet, intitulé *"Privacy in the age of exposure"*, s'intéresse essentiellement à la protection de la confidentialité des données dans le contexte de services numériques. Les individus divulguent des informations personnelles et exposent une part toujours croissante et sensible de leurs données dans le but d'utiliser et d'interagir avec les différents services en ligne.

L'adoption continue de ces pratiques pose plusieurs risques en matière de vie privée. En fait, l'apparition des technologies et services en ligne remet en question la vie privée qui est un droit individuel et fondamental [117]. Ainsi, tout en étant conscient des risques de bris de confidentialité, l'individu est aujourd'hui poussé à divulguer et révéler de plus en plus de données pour accéder à des services en ligne [6].

Une réflexion sur la transmission de ces pratiques de divulgation et les risques inhérents dans le contexte des services de géolocalisation, connus souvent sous le nom de services géodépendants, nous a conduits à vouloir analyser la divulgation des coordonnées géographiques d'un individu (*par ex.* les services de navigation tels que *Google Maps*, *Here*, etc.).

Prenons le cas des dispositifs *Android* qui collectent depuis 2017 les adresses des stations de téléphonie cellulaire situées près de leurs utilisateurs, même lorsque les services de localisation sont désactivés. Cette information est ensuite transmise à *Google* qui peut l'exploiter pour déterminer les mouvements des utilisateurs[1]. Ce type de pratique est devenu hélas courant ces dernières années. Ainsi, les fournisseurs de services exploitent des efforts pour une collecte de données plus obscure au lieu de privilégier la vie privée des utilisateurs.

[1] https://qz.com/1131515

1.1 Contexte

Les services géodépendants (*Location-Based Services* LBS) accueillent chaque jour des millions d'utilisateurs. Les usagers de ces services, qu'ils soient des particuliers ou des entreprises, sont directement affectés par leur fulgurante expansion. Par ailleurs, leur utilité force en quelque sorte les usagers à divulguer leurs coordonnées géographiques. Ces derniers représentent l'information de base dans un service géodépendant, et forment en même temps l'entité qui pose le plus de risques de violation de vie privée.

Dans la plupart des pays, les données de géolocalisation ne peuvent être collectées sans fournir aux utilisateurs des garanties de confidentialité conformément à la législation de protection des données. Cependant, les règles de confidentialité ne peuvent pas protéger les informations personnelles contre les parties malveillantes essayant d'accéder à ces données sans le consentement de l'utilisateur.

Cette thèse porte essentiellement sur deux aspects majeurs liés à l'utilisation des services géodépendants : la protection de la vie privée des usagers, et la garantie de l'utilité estimée. Le premier aspect considère les LBS comme un adversaire à qui les utilisateurs ne doivent en aucun cas fournir leurs coordonnées géographiques. Un exemple peut s'illustrer par les applications mobiles qui collectent et traitent les données de leurs utilisateurs de façon obscure, et sans donner des détails concernant leur exploitation. Quant au deuxième aspect, il consiste à garder tous les avantages d'un LBS en respectant la vie privée des utilisateurs. Cela revient à trouver la meilleure façon de divulguer juste ce qu'il faut pour obtenir les résultats souhaités.

Bien que plusieurs travaux aient été réalisés afin de protéger la vie privée des utilisateurs, une protection totale reste encore difficile à atteindre. Les données collectées présentent une source précieuse et utile pour les compagnies et les entreprises. Le marketing, la publicité visée, et le profilage sont devenus des outils largement utilisés dans l'économie actuelle [120]. En outre, les utilisateurs sont souvent insoucieux en ce qui concerne la vie privée, et la décision de divulguer ou non une information est souvent

prise de façon instantanée [29]. En fait, c'est cette décision que les travaux en vie privée doivent prendre en considération.

Par ailleurs, l'évolution remarquable des technologies mobiles a réussi à contribuer à l'adoption répandue des LBS. En plus d'être des moyens de communication omniprésents, les dispositifs mobiles présentent aujourd'hui des outils équipés de toutes sortes de capteurs et technologies de géolocalisation. Cela signifie que maintenant et plus que jamais, les deux mondes virtuel et réel sont interconnectés. Nous vivons dans un village mondial de l'information, avec des liens forts de géolocalisation. En d'autres termes, la géolocalisation rajoute la dimension permettant au service de relier toute action, interaction, ou tout événement commis en ligne à des coordonnées géographiques. Par exemple, le cas du commerce géodépendant, ou un client reçoit de la publicité et des offres commerciales en fonction de sa position actuelle, représente la forte relation engendrée par l'évolution des LBS.

1.2 Motivations

Il est primordial de noter que, bien que les technologies actuelles soient utiles pour de nombreux aspects de notre vie quotidienne, l'information recueillie peut être mal-exploitée et par conséquent causer des dommages à notre vie privée. Une personne malveillante peut examiner les données d'un utilisateur, les analyser et créer de l'information "intelligente" qui pourrait être utilisée pour générer des modèles comportementaux basés sur la géolocalisation de l'utilisateur. Par exemple, les entreprises de marketing, telles que *Urban Airship*[2] ou autres, proposent maintenant des outils de profilage d'audience qui permettent d'intégrer des fonctionnalités de ciblage de clients en se basant sur leurs données de géolocalisation. Par conséquent, une exploration des modes d'utilisation et des canaux de divulgation dans le contexte des LBS est nécessaire.

Cette recherche traite le problème selon deux thématiques principales, à savoir : (1) les interactions entre les utilisateurs et les LBS, (2) la divulgation de données person-

[2]https://www.urbanairship.com/

nelles au cours de ces interactions et les risques encourus en matière de vie privée. Pour expliquer la relation entre ces deux thématiques et l'intérêt de les étudier nous commençons par souligner le rôle des LBS dans la vie actuelle, et l'utilité qu'ils peuvent fournir. Nous analysons ensuite les risques de bris de vie privée imposés par leur utilisation.

L'environnement fourni par les LBS assure aux utilisateurs la possibilité de *partager*, et *d'obtenir* des données géodépendantes (*par ex.* partager leurs coordonnées géographiques pour obtenir les positions des restaurants à proximité). Deux opérations souvent connexes qui abstraient l'ensemble des interactions entre utilisateurs et services géodépendants. En effet, des travaux ont constaté que les coordonnées géographiques d'un utilisateur *Android* régulier peuvent être partagées jusqu'à 5398 fois en seulement 2 semaines avec 10 applications populaires installées, et cela avec ou sans le consentement explicite de l'utilisateur [2]. Il est à noter que 70% des utilisateurs de dispositifs mobiles ont au moins 11 applications téléchargées.

À l'exception de quelques LBS populaires qui fournissent une politique de confidentialité, de nombreux services disponibles n'en respectent aucune, et n'informent pas leurs utilisateurs de comment, où, et quand leurs données seront utilisées.

Le problème de vie privée ne concerne pas seulement les coordonnées géographiques, mais aussi la valeur qu'elles rapportent aux *attaquants* (également connu sous le nom *adversaires*). Par exemple, la plupart des entreprises considèrent que l'ajout de données de géolocalisation aux données des clients peut être utile pour adapter les opérations commerciales en fonction des préférences et des positions des clients. En plus, la géolocalisation contribue largement dans un processus de ré-identification et de profilage des individus [4].

Cependant, tant que les données de géolocalisation ne sont pas liées à l'identité d'un individu, elles peuvent être utilisées positivement. C'est le cas de ce qui s'est passé au Canada. Les analystes de la police ont pu se faire une idée de ce qui se passait au centre-ville d'Ottawa lors de la fusillade du 22 octobre 2014 en utilisant des mots-clics *Twitter* et des outils de géolocalisation [112], ceci illustre une utilisation positive de données

de géolocalisation. Ce qui nous mène à la question de l'équilibre qui consiste à garder l'utilité d'un LBS tout en protégeant la vie privée de l'utilisateur. Une question que nous discutons dans les objectifs de cette thèse.

1.3 Objectifs de recherche

Cette thèse a pour but de trouver et définir l'équilibre entre l'utilité des services et la vie privée des usagers dans les LBS en proposant une plateforme divisée selon les quatre objectifs suivants :

1. **Un mécanisme de protection collaboratif.** Cet objectif vise à fournir une protection similaire ou meilleure par rapport aux mécanismes classiques sans avoir besoin d'altérer les coordonnées géographiques de l'utilisateur. Notons ici que nous visons à garantir des informations géodépendantes précises en ne divulguant pas d'information pouvant identifier l'utilisateur (*par ex.* coordonnées géographiques).

2. **Un module de quantification de risques.** Nous visons dans cet objectif la réalisation d'un mécanisme de quantification qui sert à évaluer et estimer d'un côté les risques associés à l'utilisation du service géodépendant, et celle d'un mécanisme collaboratif d'un autre côté. Il s'agit en fait de choisir les meilleurs "collaborateurs" en termes de risques potentiellement causés à l'utilisateur. Le terme collaborateur réfère aux autres utilisateurs participant à un mécanisme collaboratif, dont certains peuvent s'avérer être curieux ou malicieux.

3. **Un module d'estimation de confiance.** Cet objectif a pour but de fournir un mécanisme d'évaluation de confiance qui assure un environnement sûr et fiable pour l'utilisateur. Notons que le terme environnement réfère à un écosystème comprenant l'utilisateur, les autres collaborateurs, et le LBS lui-même. Cet objectif est fortement corrélé avec le précédent.

4. **Une métrique de confidentialité.** Cet objectif vise à proposer une métrique adaptée aux mécanismes collaboratifs. La motivation se résume par l'absence d'une métrique dans la littérature qui pourrait estimer efficacement l'efficacité d'un mécanisme de protection collaboratif, et par la nécessité d'une évaluation holistique qui prendrait en considération la majorité des modèles existants.

Afin de répondre à ces objectifs, nous proposons une plateforme de protection de confidentialité, nommée *Deloc*, composée de deux modules principaux qui sont : *le module de processus de délégation* qui assure la réalisation de l'objectif 1, et *le module de l'estimation de risques et confiance* qui couvre les objectifs 2 et 3. Quant à la métrique de confidentialité, représentée par l'objectif 4, elle ne fait pas partie des modules de *Deloc*, mais elle sert à évaluer son efficacité vis-à-vis la vie privée et l'utilité. L'architecture détaillée de chaque module est discutée dans le chapitre (3).

1.4 Organisation du document

Cette thèse est organisée en sept chapitres consignés dans le tableau suivant.

Chapitre	Contenu
Introduction	Introduire les motivations et les objectifs de la thèse.
Services géodépendants : une vue d'ensemble	Discuter les principales notions reliées aux LBS, et les différents mécanismes de protection existants.
Problématiques de recherche	Présenter les problématiques sur lesquelles porte cette thèse.
Délégation des tâches géodépendantes	Mettre les bases et les fondements théoriques et applicatifs du mécanisme de protection proposé.
Estimation de risques et confiance	Présenter les mécanismes et modèles responsables de la quantification des risques et l'évaluation de la confiance.
Métrique de confidentialité	Définir les fondements de la métrique de confidentialité proposée ainsi que son applicabilité au cas de *Deloc*.
Mise en œuvre et application	Proposer une implémentation efficace de l'ensemble des modules qui composent la plateforme.
Conclusion	Discuter nos contributions ainsi que les travaux futurs envisagés.

Première partie

Contexte Théorique et Problématiques

CHAPITRE 2

LBS : UNE VUE D'ENSEMBLE

Les services géodépendants représentent l'ensemble des services reposant sur les fonctionnalités de géolocalisation des dispositifs mobiles. La popularité croissante de ces derniers, ainsi que la disponibilité de connectivité Internet mobile (*par ex.* 4G et Wi-Fi) et l'évolution des capacités de positionnement global (*par ex.* GPS) ont pressé au développement de nombreuses applications géodépendantes. Par exemple, les utilisateurs peuvent explorer des points d'intérêt dans leur proximité (*par ex.* Google Maps), partager leurs positions avec leurs contacts dans le cas des réseaux géosociaux (GSNs), ou même participer à un réseau de détection (*crowd-sensing network*) pour divulguer des données géolocalisées (niveau de pollution).

Ces utilisations ne peuvent représenter tous les domaines d'application des LBS. Les développeurs innovent chaque jour, et ces cas ne représentent que quelques utilisations possibles. Par ailleurs, l'utilisation des fonctionnalités de géolocalisation d'un dispositif mobile n'est pas exclusive aux LBS, et ces derniers ne collectent pas les coordonnées géographiques seulement.

Autrement dit, la quantité des données collectées par des applications mobiles se duplique d'un jour à l'autre, et contient le plus souvent un vaste ensemble de données sensibles qui incluent les coordonnées géographiques. Ainsi, nous considérons une application mobile comme étant une application LBS si ses services majeurs se basent sur la position de ses utilisateurs.

Si nous supposons que toute application qui accède à la position d'un utilisateur comme une application LBS, ceci va inclure la majorité des applications mobiles, même celles qui n'utilisent la géolocalisation que pour des fins statistiques. En revanche, si nous nous limitons aux applications qui accèdent à la géolocalisation seulement, nous finirons par n'identifier aucune application LBS. De façon plus concrète, si nous prenons comme

exemple quelques applications LBS existantes sur *Android* et essayons d'identifier les données auxquelles elles peuvent accéder, nous retrouverons que la géolocalisation n'est en fait qu'une partie de l'ensemble des données collectées sur un dispositif. Le tableau 2.I liste quelques applications LBS et les permissions requises pour leur fonctionnement.

Tableau 2.I – Exemples des permissions requises dans les applications LBS

Application LBS	**Version**	**Lancement**	**Permissions en commun**	**Autres permissions**
Yelp	9.12.0	2004	Localisation Stockage Comptes	Camera Contacts Microphone
Google Maps	9.54.1	2005		Camera Contacts
Foursquare	2017.05.15	2009		Contacts Données WiFi et Bluetooth
Tinder	7.2.0	2012		Données d'appels Historique Identifiant
Pokémon GO	0.63.4	2016		Camera Contacts

Comme montré dans le tableau précédent, une application LBS typique peut accéder à plusieurs types de données à l'aide d'un dispositif mobile. Pendant que certaines paraissent nécessaires pour le fonctionnement de l'application (*par ex.* Camera sur *Pokémon GO* pour la réalité augmentée), d'autres le sont moins concernant la nécessité des permissions requises (*par ex.* données d'appels sur *Tinder*). Par conséquent, et afin d'éviter toute sorte de divulgation liée aux attaques par inférences ou par corrélation de

données, un mécanisme de préservation de la confidentialité de localisation (LPPM) doit considérer le fait qu'une application peut accéder à toutes les données sur un dispositif mobile.

En partant des faits mentionnés précédemment, nous discutons dans la suite de ce chapitre de l'état actuel des LBS, leurs implications, et des modèles et architecture les plus utilisés dans leur fonctionnement. Nous discutons également des problèmes liés à la vie privée, ainsi que des différents modèles, paradigmes et métriques utilisés dans les mécanismes de préservation de la confidentialité de localisation (LPPM).

2.1 Revue des services géodépendants actuels

Afin de pouvoir regrouper le maximum des services géodépendants existants, une modélisation formelle est requise. En effet, il est plus logique de les modéliser en termes de flux d'information avant de pouvoir proposer n'importe quelle classification. Notre choix de flux d'information repose sur le fait que la vie privée est liée spécifiquement aux données, et toute opération de protection ou de divulgation nécessite l'implication de ce flux d'information, qui représente le lien entre un service et ses utilisateurs. Nous tenons à mentionner qu'un LBS est le service géodépendant lui-même, et l'application LBS est l'application qu'un client utilise pour accéder à ce service. Ce qu'un LBS représente par rapport à son application est identique à ce que représente un service courriel par rapport à l'application cliente utilisée pour lire ses courriels (*par ex.* Mozilla Thunderbird). Nous définissons dans ce qui suit le modèle LBS utilisé dans ce travail, et nous passons par la suite à l'application de ce modèle aux différentes classes des services existants.

2.1.1 Modèles et architectures

La multitude des LBS existants et la diversité de leurs domaines d'application rendent leur classification délicate. En fait, le choix des cas d'utilisation commune entre les différents LBS est souvent insuffisant pour faire une distinction précise. De plus, la mise

en œuvre d'un mécanisme de préservation de confidentialité, qui est l'objectif de ce travail, nécessite une formalisation précise pour proposer une meilleure protection. Trois classifications majeures ont été proposées dans la littérature [27, 49, 68, 77, 102, 130], où leurs auteurs catégorisent les LBS selon le domaine d'application [27, 68, 77], selon le type d'information traitée [49, 130], ou en fonction de l'exactitude de l'information révélée [102].

La toute première classification des LBS en fonction de leur domaine d'application a été proposée dans un travail de Lopez *et al.* [77]. Elle les traite selon 6 catégories principales, où chacune reflète un type d'utilisation distinct et une catégorie spécifique d'utilisateurs :

- **Services de sûreté,** sont des services conçus pour fournir une assistance aux utilisateurs finaux en cas d'urgence. Par exemple, l'application *FlagMii*[1].

- **Services d'information,** s'intéressent principalement à la distribution numérique d'informations en fonction de la position, de la spécificité temporelle et du comportement de l'utilisateur. Il comprend les services d'informations routières, les services d'aide à la navigation, les pages jaunes, les services de voyage et tourisme, etc. L'application *TripAdvisor*[2] en est un exemple.

- **Services d'entreprise,** sont des services qui comprennent le suivi des véhicules, les systèmes logistiques, la gestion des flottes, la gestion de la main-d'œuvre et la recherche de personnes. Utilisée souvent en masse, et l'information qu'ils collectent représentent souvent une propriété de l'entreprise qui les exploite. La plateforme *Omnitracs*[3] est un bon exemple pour cette catégorie.

- **Services du portail du consommateur,** sont des services qui fournissent des informations locales, météorologiques ou de trafic et sont déterminés selon la posi-

[1] https://www.flagmii.com
[2] https://www.tripadvisor.com
[3] https://www.omnitracs.com

tion actuelle de l'utilisateur comme la plateforme *Google News*[4] par exemple.

- **Service Télématiques,** représentent la catégorie la plus utilisée pour décrire les systèmes de navigation de véhicules dans lesquels les conducteurs et les passagers utilisent la technologie de positionnement pour obtenir des instructions, suivre leurs trajectoires ou obtenir de l'aide lorsqu'un véhicule est impliqué dans un accident. Un pionnier dans cette catégorie des services est *Garmin*[5] avec tous les services de positionnement qu'il offre.

- **Services de localisation déclenchable,** sont des services qui se déclenchent lorsqu'un utilisateur entre dans une zone prédéterminée. Exemples : publicité sensible à la localisation, facturation sensible à la localisation et logistique sensible à la localisation. *Salesforce Marketing*[6] en est un exemple.

La classification telle que proposée par Lopez *et al.* [77] peut regrouper un bon nombre des services existants. Cependant, il n'est pas évident de mettre en œuvre un mécanisme de protection en partant d'une classification basée sur le domaine d'application. Dans la plupart des cas, chaque catégorie nécessite un mécanisme conçu selon ses exigences et doit agir en fonction des cas d'utilisation respectifs à la catégorie en question. Autrement dit, si nous partons de cette classification afin de concevoir un LPPM, ce dernier doit nécessairement fonctionner selon chacune des catégories mentionnées en ignorant la notion de flux d'information. Cette notion de flux d'information, non considérée dans cette classification, représente le point en commun entre les LBS qui permet l'élaboration d'un mécanisme holistique. L'information est l'entité qu'un mécanisme doit protéger, et formaliser son flux est une étape nécessaire dans la création d'un LPPM efficace.

Les 2 autres classifications proposées reposent soit sur l'analyse de type d'informations échangées, soit sur la précision des données géographiques. Zheng *et al.* par

[4]https://news.google.com
[5]https://www.garmin.com
[6]https://www.salesforce.com

exemple proposent trois catégories principales [130] :

- **Services de géomarquage,** sont des services qui permettent aux utilisateurs d'ajouter des étiquettes et des commentaires géodépendants au contenu multimédia tel que texte, photos, et vidéos.

- **Services conduits par la position exacte,** sont des services qui encouragent les gens à partager leurs positions actuelles en s'interagissant dans les endroits qu'ils visitent et en reliant leurs activités quotidiennes à un lieu donné.

- **Services conduits par la trajectoire,** sont des services qui fournissent non seulement les coordonnées exactes d'une position, mais aussi le chemin détaillé parcouru par un utilisateur en connectant les coordonnées de plusieurs positions.

Nous avons proposé une classification plus orientée vers le contexte de la vie privée dans laquelle nous avons catégorisé les LBS selon la précision des coordonnées géographiques divulguées [102]. Initialement proposée pour les réseaux géosociaux, notre classification traite les LBS en fonction de leur précision dans la divulgation de la position de l'utilisateur, vu l'influence directe de l'accès à la position précise sur la vie privée de l'utilisateur. En d'autres termes, la classification que nous avons proposée considère la vie privée comme une caractéristique de comparaison [102]. De ce fait, nous les séparons en deux catégories principales :

- **Les LBS qui nécessitent une position exacte,** sont des services qui récupèrent et, potentiellement, partagent la position exacte de l'utilisateur. Les LBS de ce type nécessitent le plus souvent la position la plus précise possible pour pouvoir fournir des résultats géodépendants précis (*par ex.* services de navigation).

- **Les LBS qui nécessitent une position relative,** se représentent par les services fournissant des informations en fonction d'un rayon géographique de leurs utilisateurs et non de leurs coordonnées géographiques exactes. En d'autres termes,

ils n'utilisent que de la distance relative entre utilisateurs, tout en cachant leurs positions exactes et en limitant la précision de localisation à une zone particulière. Ce type de services est souvent utilisé dans des applications qui exploitent la géolocalisation de l'appareil afin de permettre aux utilisateurs de localiser d'autres personnes à proximité (*par ex.* réseaux géosociaux). Un exemple de cette catégorie s'illustre par les applications de rencontres qui ne divulguent que la distance relative (Alice est à 3km de Bob).

En plus de la classification mentionnée, il est également important de faire la distinction entre deux différentes notions dans les LBS : la *localisation d'un service*, et la *localisation d'un individu*. Cette distinction aide à comprendre l'effet de la localisation d'un service (*par ex.* POI) sur la vie privée des personnes.

La localisation d'un service peut être définie par la détermination des coordonnées géographiques d'un lieu d'intérêt (POI). La position d'un individu par conséquent peut être révélée si ce dernier se trouve dans ce lieu. En d'autres termes, même si l'individu ne divulgue pas sa position explicitement, sa présence dans un lieu prédéfini en interagissant avec un LBS aide à inférer sa position exacte. Cette relation représente une connaissance cruciale qu'un LBS ou d'autres adversaires peuvent acquérir à propos d'un individu.

Vu la complexité et la multitude des catégorisations de LBS, et afin de pouvoir mettre en œuvre un LPPM holistique et efficace, une modélisation des services actuels est indispensable pour définir les scénarios d'intervention d'un LPPM, et regrouper le maximum de classes. En nous inspirant du modèle proposé dans un travail de Damiani [27], nous définissons les LBS en fonction du flux d'information. Une information échangée est représentée par une requête. Cette dernière se compose de trois propriétés principales : *la direction*, *le contenu*, et *la fréquence*.

- **La direction** décrit la manière de collection des données géodépendantes. Une information échangée avec un LBS est soit de type *interrogation* où l'utilisateur initie une requête en attendant une réponse, ou bien de type *transaction* ou l'utili-

sateur transmet sa requête sans avoir besoin d'une réponse. Deux exemples illustrant les deux types sont les applications de découverte des points d'intérêt (POI) pour le premier cas, et la télédétection en masse (*crowd-sensing*) pour le deuxième.

- **Le contenu** représente le noyau de la requête. Il se compose souvent de l'identifiant et coordonnée géographique de l'utilisateur, du temps d'émission de la requête, et d'autres informations supplémentaires (*par ex.* le type de POI à chercher). Parfois, il se peut qu'une application accède à d'autres informations et les envoie dans des requêtes qui ne sont pas de nature géodépendante (*par ex.* envoyer la liste de contacts dans une requête indépendante). Par conséquent, un LPPM efficace doit analyser chaque requête envoyée d'un utilisateur afin de pouvoir l'inspecter et imposer les protections nécessaires.

- **La fréquence** décrit le nombre de requêtes envoyées dans une période de temps précise. Les requêtes peuvent être échangées de façon *continue* ou *sporadique*. Le premier type peut s'illustrer par les applications de navigation, le deuxième par la majorité des services géodépendants (*par ex.* recherche des POI, télédétection en masse, services géosociaux).

La définition d'un tel modèle nous permet de mettre les bases de notre proposition de LPPM, et pouvoir éventuellement agir d'une façon holistique contre les différentes menaces de la vie privée. Autrement dit, le flux d'informations est le terrain d'activité, et la définition de chacune de ses propriétés est indispensable pour fournir un LPPM efficace, utile, et adaptable.

2.1.2 Implications et domaines d'application

Les implications des services géodépendants décrivent les conséquences attendues par leur utilisation. En outre, ils représentent une vaste catégorie des services en ligne, et leur développement rapide dans les dernières années est une des raisons majeures derrière leur implication dans la vie des individus. Afin de pouvoir identifier la majorité de

leurs implications, nous les regroupons selon trois aspects principaux qui sont les implications *sociales*, *comportementales*, et celles qui touchent directement à *la vie privée* des utilisateurs. Nous détaillons dans la suite chaque catégorie d'implication en mentionnant le domaine d'application le plus influençant dans chaque catégorie.

2.1.2.1 Implications sociales

L'utilisation des LBS a ses avantages et ses inconvénients. Ils offrent des services intéressants, avantageux et utiles. Aujourd'hui, un utilisateur peut localiser n'importe quel POI en tapant seulement son nom, ou encore il peut le visiter sans même connaître le chemin au préalable. Cette utilité représente un de leurs principaux avantages sur le plan social. La carte et l'annuaire téléphoniques peuvent maintenant se retrouver combinés dans une application mobile simple et intuitive.

Cependant, leur impact sur la vie sociale reste encore à discuter. Le couple qui a été tué au Brésil en suivant l'application *Waze* qui les a menés à un des quartiers les plus violents de la ville de Rio de Janeiro[7] en est un exemple. Dans le même contexte de l'application *Waze*, l'application peut diriger des centaines de véhicules dans un quartier résidentiel afin d'éviter un accident ou un embouteillage à proximité[8]. Une pratique qui semble déranger les résidents de ces quartiers et qui les pousse à chercher des moyens pour faire tromper l'application.

Les LBS peuvent aussi être utilisés dans plusieurs domaines tels que la santé et la protection civile. Des exemples concrets illustrés par les applications qui aident à localiser les participants dans les tournois sportifs et permettent, par exemple, à localiser un alpiniste blessé au milieu d'un endroit géographique difficilement accessible [102]. D'autres utilisations peuvent se résumer dans les applications de localisation de flottes d'entreprises et de voitures et objets volés. Un exemple concret des implications sociales est l'initiative proposée par l'université de Harvard qui utilise l'application *Foursquare*

[7] https://www.cnn.com/2015/10/05/americas/brazil-wrong-directions-death/index.html
[8] https://gizmodo.com/is-it-really-possible-to-trick-waze-to-keep-traffic-off-1660273215

pour aider les nouveaux étudiants à se localiser à l'intérieur du campus, et à pouvoir éventuellement le découvrir plus rapidement [85].

Une nouvelle tendance impliquant l'utilisation des LBS est celle des *réseaux antisociaux*. Par exemple, une application pour iPhone appelé *Cloak* a été lancée afin d'aider les individus à éviter les personnes indésirables à proximité. En se servant des données collectées à partir des autres services, *Cloak* permet à ses utilisateurs de suivre leurs contacts et connaissances afin d'éviter ceux qu'ils ne désirent pas rencontrer.

2.1.2.2 Implications comportementales

L'effet des LBS sur le comportement de leurs utilisateurs ne peut pas être négligé. Plusieurs travaux ont abordé le sujet des implications comportementales [44, 48, 84, 94], et ont essayé d'identifier l'impact d'adoption sur la vie quotidienne d'un individu. Les travaux mentionnés traitent les implications comportementales de point de vue éthique et juridique en particulier en ce qui concerne la vie privée, la propriété de l'information, l'accessibilité, les risques associés à l'utilisation, et les préoccupations juridiques et réglementaires.

Les LBS incitent le partage et la récupération des informations géodépendantes. Ainsi, un utilisateur peut se retrouver face à des situations qui impliquent plusieurs conséquences autres que le partage de ses propres informations. Par exemple, un utilisateur qui partage du contenu collectif (une photo en groupe) en précisant l'endroit, les personnes dans la photo, et le temps exact de la photo nuirait aux autres utilisateurs. Ce type d'action fait partie des implications sur la façon de partager l'information, de la récupérer, de la gérer, et même de la supprimer. En outre, plusieurs utilisateurs ignorent l'aspect juridique de l'utilisation des LBS et les obligations face à la loi, notamment face aux aspects des droits d'auteur et de la propriété numérique.

Une dimension des implications comportementales que nous considérons comme une entité indépendante est celle des implications sur la vie privée. Nous justifions notre décision par l'influence considérable des LBS sur la vie privée des utilisateurs.

2.1.2.3 Implications sur la vie privée

L'utilisation des LBS implique la collecte d'informations, et l'abus de cette collecte mène à des conséquences désastreuses concernant la vie privée des utilisateurs. Bien que le partage d'informations géodépendantes puisse avoir des bienfaits, les conséquences négatives qui facilitent les actes criminels sont à considérer. Ainsi, des informations géodépendantes combinées à d'autres informations personnelles peuvent être utilisées par des criminels pour identifier la position actuelle ou prévue d'un individu, facilitant ainsi la possibilité de causer des dommages à l'utilisateur et même à ses compagnons. Cela va du cambriolage et vol, au harcèlement criminel, kidnapping, et violence domestique. Les cybercriminels, principaux acteurs de crimes informatiques, menacent l'intégrité, la moralité, l'estime de soi et la sécurité des personnes. De plus, le manque de prudence lors de la publication des photos et commentaires, ainsi que leurs positions physiques, peut constituer le meilleur outil pour l'enlèvement, le vol, le viol, le renvoi du travail, le divorce, etc. Le site web *Please Rob me*[9] qui peut collecter tous les endroits liés à un compte *Twitter* montre le danger des données de géolocalisation partagées publiquement.

Le jeu *Pokemon Go !*[10] par exemple a ouvert une porte aux prédateurs sexuels selon une enquête menée par le sénat de l'état de New York en 2016[11]. Les enquêteurs ont visité les domiciles de 100 prédateurs sexuels qui ont commis des crimes haineux contre des enfants, et ils ont découvert des personnages de Pokémon devant les maisons de 57 prédateurs. Même en étant par inadvertance, cela résume bien les risques associés au partage non contrôlé de sa position géographique.

Par ailleurs, les LBS peuvent utiliser les informations de leurs utilisateurs à des fins malicieuses, ou autres que celles déclarées dans leurs politiques. Le marketing visé, et la vente d'informations privées sont devenus un commerce rentable. En fait, beaucoup

[9] http://pleaserobme.com
[10] https://www.pokemongo.com
[11] https://nypost.com/2016/07/29/pokemon-go-lures-children-near-homes-of-sex-offenders

d'entreprises tirent leurs revenus des "entrepôts de données" et des outils analytiques [5]. L'autre point remarquable à propos de l'utilisation des LBS réside dans le fait que les informations de localisation sont considérées comme sensibles. La collecte et le traitement des données de localisation sur une base régulière peuvent conduire à déduire des informations privées telles que le domicile ou le lieu de travail, préférences politiques, ou inclinaisons religieuses [7].

Cependant, les avantages des LBS ne peuvent être ignorés, ce n'est pas à cause de risques qu'ils peuvent porter à notre vie privée, que nous devons les effacer tous de nos dispositifs mobiles. C'est la responsabilité des chercheurs et des fournisseurs de services d'assurer la vie privée, soit en construisant des applications qui respectent la vie privée ou en fournissant des mécanismes de protection qui répondent aux attentes des utilisateurs.

2.2 Vie privée et services géodépendants

Avant d'aller plus loin dans la description des différents aspects, risques, et menaces que les LBS peuvent causer à la vie privée des utilisateurs, il est important de détailler chaque notion reliée, notamment celles que nous croisons dans ce contexte. En général, les applications LBS exigent que les utilisateurs divulguent leurs positions sous une forme ou une autre, ce qui soulève des préoccupations en matière de protection de vie privée. En outre, avec une connaissance des positions des utilisateurs, un adversaire met en évidence un large éventail d'attaques contre des individus, allant de la surveillance physique et harcèlement, jusqu'au vol d'identité, tout en inférant des informations sensibles telles que l'état de santé de l'individu, les modes de vie alternatifs, et les affiliations politiques et religieuses.

2.2.1 Définitions et propriétés

Le droit à la vie privée ou à la confidentialité est un principe ancien, il désigne le droit d'un l'individu à avoir une protection complète de soi et de ses biens [122]. Cependant, il est parfois nécessaire de le redéfinir afin qu'il soit conforme aux nouvelles menaces et exigences.

Le droit à la vie privée assure à un individu la capacité de se cacher et se protéger, ou protéger ses informations personnelles. Néanmoins, ce qui est considéré comme privé dépend d'un individu à un autre, bien qu'il existe toujours de certaines exigences en commun.

La notion de la vie privée se réfère toujours au terme anglais Privacy, qui désigne selon la définition de Westin "Le droit à un individu de déterminer lui-même quand, comment et dans quelle mesure ses informations seront divulguées aux autres." [124]. Autrement dit, la vie privée est la capacité que possède un individu à propos de contrôler l'accès à ses informations, et de conserver son anonymat. Cela s'explique par le fait que l'individu est le seul propriétaire de ses informations et renseignements, et le seul à avoir le droit de décider de les divulguer ou non.

En fait, la plupart des gens considèrent que la vie privée est précieuse et difficile à récupérer en cas de perte causée par des manœuvres intentionnelles ou par inadvertance. Par conséquent, sont considérées comme des risques liés à la vie privée toutes les actions faisant intrusion dans l'intimité de la personne, notamment la surveillance de ses activités, l'enregistrement ou le traitement de ses informations personnelles.

Comme nous l'avons cité au début du chapitre, l'acception du concept de vie privée dépend d'un individu à un autre. En effet, il est nécessaire de mettre un ensemble de préoccupations communes entre les utilisateurs d'un côté, et de standard de développement pour les créateurs des LBS de l'autre côté. Nous discutons dans la prochaine partie les critères les plus utilisés et leurs applications.

Les principes fondamentaux de la vie privée selon *"Common Criteria for Information*

Technology Security Evaluation (CCITSE)" [22] sont au nombre de quatre. CCITSE est une norme internationale (ISO / IEC 15408) utilisée pour la certification de la sécurité informatique , elle définit dans sa dernière publication (version 3.1 révision 4) les critères suivants :

- **Anonymat.** Un utilisateur peut utiliser une ressource ou un service sans révéler son identité.
- **Pseudonymie.** Les autres utilisateurs sont incapables de lier l'identité d'un utilisateur à une action donnée, dans ce cas l'utilisateur est responsable de ses actions.
- **Non-associabilité.** L'utilisateur peut faire de multiples actions sans que les autres soient en mesure de les relier ensemble.
- **Non-observabilité.** L'utilisateur peut utiliser une ressource ou un service sans que les autres soient en état de constater que la ressource ou le service est utilisé.

Notons que la CCITSE s'intéresse à la définition des composants fonctionnels de la sécurité informatique, ainsi qu'aux bases sur lesquelles les exigences fonctionnelles de la sécurité informatique se bâtissent. Ces dernières sont exprimées en utilisant un profil de protection (Protection Profile) ou une cible de sécurité (Security Target).

Autrement dit, les composants fonctionnels de la sécurité assurent que le comportement défini dans les exigences fonctionnelles de sécurité dans un système informatique peut être abouti sans difficulté. Par exemple, si un créateur d'un système souhaite garantir l'anonymat pour ses utilisateurs, la définition des composants fonctionnels de la CCITSE l'aide à atteindre cet objectif en fournissant les meilleures pratiques et méthodes.

En outre, ces critères ne définissent que ce que les autres peuvent collecter comme informations au sujet d'une communication, mais ne concernent pas réellement ce que les partenaires dans une communication peuvent partager comme information. Par conséquent, elles sont insuffisantes pour couvrir tous les aspects de la vie privée, notamment la protection des données personnelles [118].

Ce standard fait également l'objet d'un travail exhaustif, qui essaye de mettre une terminologie intégrale en ce qui concerne la vie privée [95]. Les auteurs rajoutent les critères suivants sur ce qui a été déjà défini par la CCITSE :

- **Non-détectabilité.** Un attaquant ne peut détecter si un point d'intérêt commun à plusieurs utilisateurs existe.
- **Gestion des identités.** Un attaquant ne peut en aucun cas lier le pseudonyme d'un utilisateur à son identité réelle.

2.2.2 Menaces liées à la vie privée

Avant que les données ne soient présentes et utilisées dans un système, elles doivent d'abord être collectées, ensuite supprimées [71]. En projetant les standards vus précédemment sur les différentes étapes de cycle de vie des données, nous déduisons que chaque critère traite des risques qui sont liés à une des étapes suivantes.

2.2.2.1 Collecte des données

Pendant la collecte des données, un service doit s'assurer que l'utilisateur est conscient que ses informations seront collectées, tout en lui expliquant les raisons et les fins de l'utilisation. Les données ne doivent pas être collectées sans son accord, il doit être complètement conscient des données fournies [111]. Par exemple, un LBS ne pourra pas obtenir la liste de contacts de l'utilisateur pendant son inscription sans avoir un accord explicite de sa part. En plus, les données collectées doivent être minimisées, le réseau ne doit assembler que les données nécessaires à l'utilisation [118].

L'autre point réside dans les risques d'inférence de données et de techniques de désanonymisation [88]. Plusieurs études ont montré comment des données qui sont considérées comme anonymes peuvent être analysées et regroupées avec d'autres données afin d'identifier les personnes concernées [88, 114] ce qui contredit le critère de la non-associabilité.

2.2.2.2 Utilisation des données

Une fois que les données d'un utilisateur sont collectées et stockées, le système doit assurer le respect du contrat d'utilisation, les données ne doivent pas être utilisées à des fins autres que celles présentées à l'utilisateur. Le système doit assurer la protection de l'identité de l'utilisateur et de ses données, tout en lui permettant un contrôle total sur comment ses données sont divulguées.

Un exemple de données utilisées à des fins autres que l'utilisation du réseau social est l'emploi des données à des raisons de marketing ou de publicité [111]. Les données dans ce cas sont souvent collectées et utilisées par des parties tierces (les fournisseurs de publicité et les entreprises de marketing), qui ne donnent à l'utilisateur aucun moyen de contrôler ou d'accéder à ses propres données.

Parfois, les mesures minimales de sécurité ne sont pas mises en œuvre, ou ne sont pas suffisantes pour protéger les données stockées. Les dommages peuvent être sérieux, comme dans le cas de vol d'identité où une personne (voleur d'identité) passe pour une autre personne (victime) pour accéder à des données pour lesquelles seule la victime a le droit (données bancaires par exemple), ou pour réaliser une action malveillante dont la victime sera tenue responsable.

2.2.2.3 Effacement des données

Après la suppression d'un utilisateur du système, ses données doivent être supprimées, ce qui n'est pas une tâche facile, à cause de la quantité souvent énorme de données stockées. Le système doit assurer la suppression des données dans un délai limité et doit assurer le droit à l'oubli à ses utilisateurs.

Certains services en ligne conservent les données des utilisateurs le plus longtemps possible afin d'en tirer un profit maximum. C'est le cas pour certains sites de commerce électronique qui conservent les données et l'historique de recherche de leurs clients le plus longtemps possible afin de récolter les habitudes d'achat et de les exploiter grâce au

forage des données [119]. Notons que le fait de garder les données contredit le principe de droit à l'oubli d'une part, et augmente les risques de mauvaise utilisation d'autre part.

2.2.3 Problématiques de la vie privée

2.2.3.1 Sensibilité des données de géolocalisation

La géolocalisation a toujours été considérée comme une information personnelle, ou au moins connue que par les proches. Les conséquences négatives de sa divulgation ne peuvent être négligées. Ainsi, la combinaison des informations de géolocalisation avec d'autres renseignements personnels peut être un moyen précis d'identification des individus par des personnes potentiellement malveillantes. De même, la position de l'utilisateur peut révéler ses centres d'intérêt, qui à leur tour facilite son identification [13, 70].

De plus, l'un des principaux problèmes est celui de la perception des risques par les utilisateurs [25]. D'autres recherches sont nécessaires pour comprendre la relation entre le comportement des gens concernant la divulgation des renseignements personnels et leur perception des risques associés.

2.2.3.2 Perte de contrôle sur les données personnelles

L'utilisation des nouvelles technologies occupe déjà une partie majeure de notre vie quotidienne et, par conséquent, de nouveaux produits et services sont offerts tous les jours. Dans la littérature, des chercheurs dans de nombreux domaines ont essayé d'étudier l'impact des LBS sur la vie privée des utilisateurs. Les utilisateurs ne pensent pas ou ne sont pas conscients des risques liés au partage de leurs informations en ligne. Pour eux, la décision de partager quelque chose en ligne est "prise à l'action" [29]. De même, le comportement des individus vis-à-vis la vie privée est malléable selon le contexte de l'interaction, ils sont plus disposés à divulguer des informations personnelles sur un réseau social que sur d'autres sites. Ils divulguent leurs informations quand ils voient les autres le faire [53].

De l'autre côté, les utilisateurs ne sont pas vraiment conscients de la valeur de leurs données et les différents risques liés à la divulgation en ligne [3]. Effectivement, 68% des jeunes Canadiens croient à tort que si un site a une politique de confidentialité, il ne sera pas en mesure de partager leurs renseignements personnels avec les autres. De même, 39% d'entre eux pensent que les entreprises ne sont pas intéressées par ce qu'ils font en ligne[12]. Des études ont été en mesure de prouver que certains utilisateurs ont désinstallé des applications mobiles après qu'ils aient pris conscience qu'elles tirent profit de leur géolocalisation. La plupart préfèrent désactiver l'accès à la géolocalisation au lieu de désinstaller complètement l'application [43].

D'autres études ont également montré que les utilisateurs accordent l'accès aux applications pour lesquelles les informations de géolocalisation sont essentielles pour leur fonctionnement, et refusent l'accès quand les fins d'utilisation sont moins claires [41]. D'autres facteurs influencent aussi l'intention de partager la géolocalisation telle que la relation avec le demandeur (famille et amis), l'endroit où l'utilisateur se trouve, l'activité qu'il fait, ou son état émotionnel [23].

Malgré le fait qu'il est difficile de cerner tous les problèmes de vie privée dans les LBS, nous avons essayé de surligner ceux liés à la divulgation de la géolocalisation. En outre, plusieurs recherches et travaux ont été réalisés pour aborder certains des problèmes de vie privée. Nous discutons dans la partie suivante des travaux les plus connus en termes de protection de la vie privée.

2.3 Revue des mécanismes de préservation de vie privée

En partant des principes de vie privée, et des problématiques majeures liées à l'utilisation des LBS, plusieurs travaux ont été proposés afin de permettre aux utilisateurs de préserver leur vie privée tout en profitant des services géodépendants. En fait, les mécanismes de protection peuvent être vus en fonction de deux composants principaux : le modèle utilisé dans un LPPM pour atteindre les objectifs désirés de confidentialité et la

[12] http://mediasmarts.ca/ycww/life-online

métrique utilisée pour évaluer son efficacité face à un ensemble prédéfini des exigences de confidentialité.

2.3.1 Modèles utilisés dans les mécanismes de protection

Nous discutons dans cette section les LPPM selon les modèles qu'ils adoptent. Nous distinguons ainsi ceux qui sont basés sur la *transformation* géographique, et ceux qui utilisent les modèles de *collaboration*. Le premier type est le plus utilisé dans les mécanismes de protection de vie privée où ils utilisent des opérations de transformations géographiques computationnelles afin d'atteindre les objectifs de confidentialité. Le deuxième type par contre peut ne pas se baser sur une transformation des coordonnées géographiques, en utilisant la collaboration entre utilisateurs du même LBS.

2.3.1.1 Modèles de transformation

Les modèles de transformation regroupent les techniques qui effectuent des opérations géographiques sur les coordonnés d'un utilisateur afin de protéger sa position réelle. La figure 2.1 illustre les 4 principales transformations qui peuvent regrouper une partie majeure des LPPM existants. Nous voyons que la position initiale dans (2.1a) peut être divulguée de façon confidentielle en la remplaçant par une zone homogène plus large (2.1b), en la substituant par une autre position à proximité (2.1c), en la cachant parmi d'autres positions (2.1d), ou en ne la divulguant pas dans une zone précise (2.1e). Nous détaillons chacun des modèles dans la suite de cette partie.

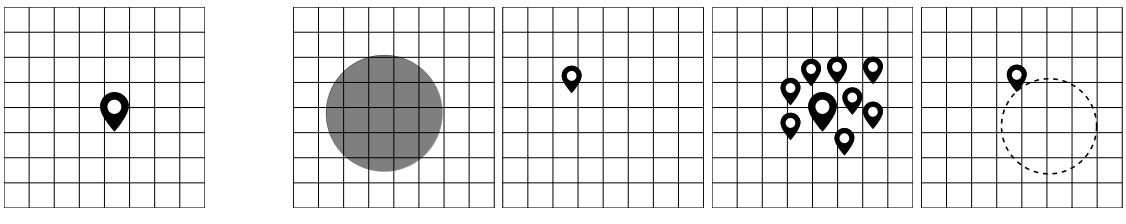


Figure 2.1 – Illustration des modèles les plus utilisés dans les LPPM

Obscurcissement

Les LPPM utilisant cette transformation visent à camoufler la position réelle de l'utilisateur dans une zone géographique plus large. Le but principal de l'obscurcissement de localisation et de dégrader délibérément la qualité de l'information sur la position d'un individu afin de protéger son intimité de localisation [35]. En d'autres termes, l'obscurcissement de localisation a pour but de représenter la position réelle de l'utilisateur loc, vers une zone plus grande r qui doit forcément contenir loc. Soit $\mathscr{L}$ le mécanisme utilisant l'obscurcissement comme modèle de transformation, et $\mathbb{E}^2$ l'espace sur lequel les opérations de localisation sont exécutées. La région d'obscurcissement r est définie comme suit :

$$\mathscr{L}(loc) = r \in \mathbb{E}^2 \quad \text{avec} \quad loc \in r$$

L'obscurcissement est largement utilisé dans différents LPPM. Bien que certaines applications l'adoptent pour protéger les coordonnées exactes de l'utilisateur [50, 129], d'autres utilisent des mécanismes basés sur l'obscurcissement pour éviter de divulguer la trajectoire et le comportement de l'utilisateur [28, 91], ou pour se protéger contre la ré-identification [54, 62].

L'obscurcissement d'une position seule ne peut pas fournir de garantie de confidentialité et peut être déjoué par des attaques contextuelles [123]. En outre, les coordonnées obscurcies peuvent compromettre l'utilité estimée dans les applications nécessitant une grande précision, comme les services d'urgence.

En conséquence, certaines variantes fondées sur l'obscurcissement ont été introduites telles que l'obscurcissement *sémantique* [28] qui prend en considération le réseau routier, et *sélectif* [91] qui ne transforme la position de l'utilisateur que dans des endroits prédéfinis. Les deux ont été utilisés dans des solutions de protection de trajectoires et ont pu garantir des exigences de confidentialité plus élevées par rapport aux mécanismes conventionnels d'obscurcissement [28, 91]. Cependant, l'utilité et la commodité des mé-

canismes basés sur l'obscurcissement restent leurs principales lacunes.

Substitution

Dans cette classe de transformations, le mécanisme remplace la position réelle de l'utilisateur par une autre position voisine. Ainsi, la position substitue loc' est transmise au lieu de la vraie position de l'utilisateur. Soit $\mathscr{L}$ le mécanisme utilisant la substitution comme transformation, et $\mathbb{E}^2$ l'espace sur lequel les opérations de localisation sont exécutées. La position substitue est définie comme suit :

$$\mathscr{L}(loc) = loc' \in \mathbb{E}^2$$

L'un des premiers LPPM qui utilisent la substitution, également connue sous le nom de perturbation de localisation permute les chemins des utilisateurs dans les zones où se trouvent au moins deux utilisateurs [58].

Une technique étroitement reliée et qui relève de la classe de substitution est l'arrondissement de la position [70, 103]. L'idée principale derrière les coordonnées de l'arrondissement est de faire correspondre la position réelle de l'utilisateur à un point de repère public (*par ex.* l'intersection de la route, le POI). La principale lacune lors de l'utilisation de cette classe de modèle réside dans la perte de la qualité des données. La substitution de localisation en soi ne peut être adaptée aux applications exigeant des données de localisation de haute précision [101] (*par ex.* la navigation en temps réel).

Confusion

Nous disons que la position réelle de l'utilisateur est confuse quand elle est contenue dans un ensemble d'autres positions fictives dans le but de la cacher [63]. En d'autres termes, un mécanisme utilisant un modèle de confusion transmet la position réelle de l'utilisateur dans un ensemble de n positions dont l'une est la véritable position. Soit $\mathscr{L}$ le mécanisme utilisant la confusion comme modèle de transformation. L'ensemble

associé à la confusion d'une position est défini comme suit :

$$loc \in \mathscr{L}(loc) = \{loc_i\}_{i \in [1,n]}$$

Dans cette technique, le LBS renvoie des résultats géodépendants pour chaque position dans l'ensemble reçu. Les résultats correspondant à la position réelle de l'utilisateur sont ensuite extraits de l'ensemble des résultats. Une approche légèrement modifiée a été proposée où l'extraction des réponses s'arrête lorsque les résultats souhaitables sont extraits [128]. Même lorsque cette classe de transformations peut assurer la confidentialité et l'utilité, la grande quantité de bruit qui peut s'ajouter aux données de localisation peut compromettre d'autres aspects de l'utilité comme dans les réseaux de détection par exemple.

Suppression

La suppression aussi connue sous le nom *dissimulation invisible* [125], interdit simplement les requêtes dans des zones géographiques précises. Autrement dit, le LPPM ne reporte pas les coordonnées géographiques de l'utilisateur en présence de certaines conditions prédéfinies (*par ex.* quand l'utilisateur est dans une zone de 500m autour de son domicile). Soit $\mathscr{L}$ le mécanisme utilisant la suppression comme modèle de transformation, la transformation basée sur la suppression est exprimée par :

$$\mathscr{L}(loc) = null$$

Cette technique a été initialement utilisée pour construire le modèle *mix-zones* [11], où les auteurs assument que chaque utilisateur devrait avoir au moins une région géographique où aucun LBS ne peut récupérer sa position. De même, lorsque l'utilisateur quitte la zone de mixage, on lui attribue un nouveau pseudonyme et le LBS peut reprendre le suivi de la position. Les travaux les plus récents utilisent le modèle de suppression dans différentes applications et surtout en conjonction avec d'autres modèles [82, 89, 90].

Même lorsque la suppression peut être efficace pour protéger la confidentialité de la position à l'intérieur des immeubles (*par ex.* un centre commercial), son inefficacité en requêtes continues (*par ex.* cas des LBS de navigation) représente sa principale lacune. En outre, les mécanismes utilisant la suppression sont plus efficaces dans les cas où la portée de déplacement des utilisateurs est large (*par ex.* en véhicule). Au centre-ville, il est difficile pour un piéton de se protéger en limitant les mises à jour de localisation dans certaines régions.

2.3.1.2 Mécanismes collaboratifs

Les mécanismes collaboratifs garantissent la co-utilité entre les utilisateurs, et il a été prouvé que, dans un contexte sécurisé pour la vie privée, ils peuvent non seulement fournir de solides garanties de confidentialité, mais ils sont aussi plus susceptibles d'être adoptés par les utilisateurs [34].

Bien que plusieurs mécanismes collaboratifs aient été proposés [93, 107, 109], la majorité se base essentiellement sur les mécanismes de transformation mentionnés auparavant. Par exemple, Shokri *et al.* proposent deux approches où les utilisateurs répondent "collaborativement" aux requêtes géodépendantes avant de faire appel à un service géodépendant [107, 109]. Un utilisateur tente d'abord d'obtenir des résultats géodépendants d'autres utilisateurs à proximité avant d'essayer de soumettre la requête. Peng *et al.* rajoutent à leur tour l'envoi de plusieurs requêtes dont chacune d'elles appartient à un utilisateur se trouvant à proximité du demandeur initial [93].

Un autre mécanisme collaboratif a été proposé par Rebello *et al.* dans lequel ils proposent une approche basée sur la transmission des requêtes géodépendantes d'un utilisateur à l'autre de telle sorte que la requête finale Q envoyée à un LBS à partir d'un réseau collaboratif soit composée du contenu issu de n utilisateurs [100]. De façon plus formelle, la composition de la requête Q est définie comme suit :

$$Q = \{req_i\}_{i \in [1,n]} \quad \text{tel que} \quad \exists req_i = req_u$$

Même lorsque cette approche peut assurer une bonne précision et garantir l'utilité des données, le fait que la requête de demandeur initial soit toujours envoyée au serveur constitue une menace sérieuse pour la vie privée [100]. D'un point de vue pratique, nous avons mentionné plus tôt que les LBS du monde réel rassemblent plusieurs types de données à partir des dispositifs de leurs utilisateurs. Limiter un mécanisme à la protection des données de localisation seulement ne pourra pas préserver la confidentialité de la position de l'utilisateur. De façon plus formelle, si un LBS reçoit deux requêtes Q_1 et Q_2 ne contenant qu'une seule entrée en commun req_j, il pourra déduire que les deux ensembles ont été générés à partir de la requête de l'utilisateur j.

D'autres systèmes collaboratifs qui servent à anonymiser les communications par Internet tels que *Tor* ont été proposés [33]. Cependant, ce genre de système n'est pas applicable dans notre contexte et ne peut fournir la confidentialité de localisation dont nous avons besoin. En fait, leur rôle sert essentiellement à crypter les canaux de communication. En d'autres termes, les données collectées à partir d'un périphérique utilisateur sont regroupées, liées et transmises via des canaux cryptés sans aucune autre mesure de protection, ce qui contredit les exigences d'un LPPM. Nous discuterons le cas de ces systèmes en détails et leur impraticabilité dans les LBS dans la section 4.1.1.

Les travaux de recherche sur les mécanismes de préservation de la vie privée de localisation ont atteint des résultats intéressants, et des garanties de protection acceptables. Cependant, le développement rapide des LBS implique la mise en œuvre de modèles et mécanismes adaptés aux scénarios et cas d'utilisation réels. Le tableau 2.II illustre un récapitulatif de modèles revus dans ce chapitre ainsi que leur applicabilité dans chacune des classes identifiées.

Tableau 2.II – Récapitulatif des modèles de préservation de la confidentialité de localisation

LPPM	Propriétés LBS						Efficacité	
Modèles	Direction		Fréquence		Contenu		Confidentialité	Utilité
	Interrogation	Transaction	Sporadique	Continue	Localisation	Autres		
Modèles computationnels								
Obscurcissement [28, 35, 50, 129, 54, 62, 91, 123]	•	•	•	•	•		•	
Substitution [58, 70, 101, 103]	•	•	•		•		•	
Confusion [63, 128]	•	•	•		•			•
Suppression [11, 82, 89, 90]	•		•		•		•	
Modèles collaboratifs								
Collaboration [93, 107, 109]	•	•	•		•	•	•	•

2.4 Conclusion

Le fait que les LBS envahissent rapidement nos vies ne peut être ignoré, car leur facilité d'utilisation attire chaque jour plus d'utilisateurs. Cela conduit à une forme d'intrusion lorsque nos informations de localisation deviennent facilement accessibles. En effet, construire des profils à partir du contenu d'un utilisateur est devenu une tâche facile, surtout lorsque l'historique des positions est également disponible.

En conclusion, nous avons discuté l'état actuel des LBS ainsi que les mécanismes et techniques proposés pour atteindre une protection efficace de vie privée. Nous abordons dans le chapitre suivant la formulation des problématiques de notre recherche, et notre proposition d'une plateforme holistique de protection de la vie privée dans les LBS.

CHAPITRE 3

PROBLÉMATIQUES DE RECHERCHE

En plus des problématiques générales de vie privée dans les services en ligne, et qui incluent les problèmes de sensibilité et de perte de contrôle sur les données en ligne, l'utilisation des LBS occasionne d'autres types de problèmes. Le fait que les LBS reposent principalement sur l'exploitation des données géographiques impose des préoccupations supplémentaires concernant la vie privée allant de l'identification au profilage du comportement. Nous discuterons dans la suite de ce chapitre les problématiques de la vie privée dans les LBS, ainsi que la solution que nous proposons.

3.1 Paradoxe de vie privée

Le paradoxe de vie privée suggère que bien que les utilisateurs des services en ligne aient presque tous l'impression d'être surveillés, et qu'ils soient soucieux des problèmes liés à leur vie privée, ils ne prennent pas de mesures sérieuses de protection, et continuent toujours à partager des informations sensibles [115]. Autrement dit, leur comportement ne reflète pas leur inquiétude à propos de leur vie privée. Par ailleurs, les différentes plateformes en ligne offrent souvent une multitude de services gratuits, et l'exploitation des données qu'ils hébergent représente de revenus importants (*par ex.* publicité ciblée) [5, 118]. Par conséquent, les problèmes de vie privée en ligne constituent un dilemme qui est à la fois important et sensible. Le cas des LBS est encore plus délicat, les utilisateurs adoptent le même comportement sans savoir qu'un LBS peut mettre leur vie privée en danger plus qu'un service en ligne traditionnel. Nous référons par un service traditionnel dans ce contexte à toute fonctionnalité mise à disposition par un composant logiciel pour assurer une tâche particulière sans que cette dernière nécessite les coordonnées géographiques d'un utilisateur.

Une simple exploration sur un réseau social géodépendant pourra révéler des in-

formations précieuses et uniquement identifiables comme dans le cas des publications taguées par des positions géographiques. Par exemple, un rapport publié par *Statistiques Canada*[1] en 2018 a révélé que l'utilisation accrue de la technologie se manifeste dans les types de harcèlement criminel. Le rapport mentionne que *"près de deux millions de Canadiens de 15 ans et plus, environ 8% des femmes et 5% des hommes, ont indiqué avoir été victimes de harcèlement criminel au cours des cinq années précédant l'enquête"*. Ces résultats suggèrent également que presque 15% de victimes ont subi des actes liés à l'espionnage ou la surveillance électronique ou en personne.

En fait, la localisation elle-même est considérée comme une information sensible. Il est à noter que quatre points spatiotemporaux distants suffisent pour identifier 95% des utilisateurs LBS [30]. Notons ici que les auteurs excluent dans leur recherche les coordonnées non distantes telles que celles partagées dans une même trajectoire durant l'utilisation d'une application de navigation. En outre, la collecte et le traitement des données de localisation sur une base régulière aident à l'inférence des données privées tels que les lieux de résidence ou de travail, les préférences sexuelles ou les inclinations religieuses.

3.2 Équilibre utilité et confidentialité

Afin d'utiliser les fonctionnalités LBS, un utilisateur doit hélas fournir ses coordonnées géographiques précises. Par exemple, lorsque 68% des utilisateurs mobiles sont préoccupés par la confidentialité et la sécurité de leurs appareils [21], 74% d'entre eux utilisent encore les LBS pour obtenir des itinéraires et des informations géodépendantes [131]. En 2015, l'Agence européenne des systèmes de navigation par satellite, également connue sous le nom d'Agence européenne des systèmes de positionnement par satellites, a publié un rapport sur les LBS et leur utilisation [40]. Parmi ses découvertes majeures, le rapport affirme que les applications mobiles reposant sur les informations de localisation ont atteint près de 3 milliards de téléchargements sur Android Play et

[1] https://www.statcan.gc.ca/daily-quotidien/180117/dq180117a-fra.htm

Apple App Store. De même, la plupart des appareils mobiles actuels prennent en charge de nombreux systèmes de positionnement tels que les services GPS[2] (Global Positioning System), le système de positionnement chinois Beidou[3], et le système russe GLONASS[4], ce qui améliore la précision de localisation au-delà de ce que les récepteurs GPS classiques peuvent offrir. Seulement 35% des utilisateurs mobiles pensent à désactiver les services de localisation sur leurs appareils [131], cela suggère que la quantité de contenu géolocalisé généré par les utilisateurs est considérable.

En revanche, les avantages des services géodépendants ne peuvent être ignorés, si les utilisateurs les installent c'est parce qu'ils ont besoin de leurs services. Par exemple, certains LBS tels que les dispositifs de traçage pour les enfants ou les personnes atteint d'Alzheimer[5], les applications conçues pour localiser les grimpeurs[6], ou même les applications de navigation[7] utilisées quotidiennement par des millions de conducteurs[8] sont quelques domaines dans lesquels les LBS ont pu prouver leur utilité. De ce fait, il est primordial d'identifier, définir et assurer un équilibre optimal entre la préservation de la vie privée des utilisateurs et l'assurance du maximum d'utilité.

3.3 Une plateforme holistique de protection de la vie privée

La motivation principale derrière ce travail est d'offrir cet équilibre optimal. La plupart des solutions existantes se concentrent sur la préservation de la vie privée en ignorant l'utilité. Néanmoins, la préservation de la vie privée seule représente une problématique compliquée vu les capacités de calcul et d'apprentissage que les LBS actuels possèdent. Ainsi, la considération de l'utilité rajoute à cette problématique une dimension supplémentaire qu'elle doit traiter minutieusement.

[2] https://www.gps.gov/
[3] http://mgex.igs.org/IGS_MGEX_Status_BDS.php
[4] https://www.glonass-iac.ru/en/
[5] https://alzheimer.ca/en/Home/Living-with-dementia/Day-to-day-living/Safety/Locating-devices
[6] https://reut.rs/2ndImfi
[7] https://www.waze.com/
[8] https://www.forbes.com/sites/petercohan/2013/06/11/four-reasons-for-google-to-buy-waze/

Une telle problématique inclut la nécessité d'un LPPM qui ne se base pas sur la transformation des coordonnées géographiques. En d'autres termes, nous visons à fournir un LPPM qui empêche toute identification potentielle d'utilisateurs sans manipulation des coordonnées géographiques. Notre objectif principal dans ce travail est de protéger la position et l'identité de l'utilisateur. Bien que la confidentialité de la localisation implique la protection des coordonnées exactes de l'utilisateur, celle de l'identité consiste à protéger les informations sensibles ou d'identification de l'utilisateur (*par ex.* nom et prénom, date de naissance, numéro de téléphone, etc.). Nous abordons maintenant les fondements théoriques de ce travail ainsi que de l'architecture générale de la plateforme proposée.

3.3.1 Cadres et fondements théoriques

Nous décrivons deux scénarios qui impliquent des interactions entre les utilisateurs et les LBS. Nous les utilisons pour discuter nos motivations et identifier les exigences clés d'un LPPM efficace. Nous supposons que les utilisateurs sont équipés d'appareils haut de gamme (c'est-à-dire des appareils mobiles compatibles GPS et WiFi) et peuvent accéder à divers services offerts par différents fournisseurs de services. La figure 3.1 illustre les deux types de services auxquels les utilisateurs de nos jours peuvent accéder via leurs appareils mobiles intelligents.

Nous nous intéressons dans cette recherche aux services d'application qui s'appuient sur les informations de géolocalisation transmises par les utilisateurs. Nous ne considérons pas les services de communication qui incluent la téléphonie et les services Internet, et qui doivent déterminer la disponibilité et la position des dispositifs mobiles dans le cadre de leur fonctionnement de base. Les services de communication doivent déterminer dans quelle cellule se trouve le dispositif mobile afin qu'il puisse être servi par la station émettrice-réceptrice la plus proche [97]. Décrivons maintenant les trois scénarios suivants qui décrivent de cas d'utilisation hypothétique des LBS, qui ne sont pas loin de la réalité et qui peuvent survenir à n'importe quel utilisateur LBS.

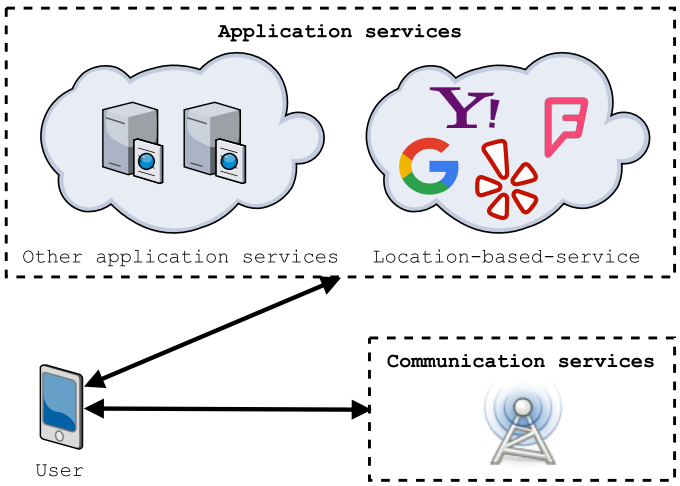


Figure 3.1 – Les services accessibles par un dispositif mobile

Scénario 1. Alice a un rendez-vous avec un médecin qu'elle n'a jamais consulté auparavant. Le médecin est un spécialiste de traitement du diabète et son bureau se trouve dans une région qu'Alice n'a jamais visitée auparavant. Le jour de son rendez-vous, Alice prend sa voiture et la conduit à proximité du bureau. Une fois là, elle ne peut localiser le bureau du médecin et pense à utiliser son téléphone intelligent pour utiliser son LBS préféré afin de le trouver. Bien qu'elle sache que l'utilisation de LBS l'amènerait chez le médecin rapidement, Alice est inquiète pour sa vie privée et sait qu'elle doit révéler son identité et son lieu de résidence au LBS lors de sa requête.

Scénario 2. Alice décide d'utiliser un LBS pour localiser le cabinet du médecin, car elle va être en retard. Une fois arrivée, elle rencontre un autre patient Bob, un ingénieur en informatique qui travaille sur le développement de logiciels. Alors qu'ils discutent dans la salle d'attente, Alice mentionne le fait qu'elle ne soit pas à l'aise à partager ses informations avec les services d'application, en particulier les coordonnées géographiques. Bob est d'accord avec le point de vue d'Alice et affirme que, tout en ayant une expérience dans la collecte de données auprès des utilisateurs, il peut lui assurer que la divulgation de l'information va bien au-delà de sa perception. Bob lui donne l'exemple de l'utilisation du calendrier de son appareil pour enregistrer les détails de ses rendez-vous en parallèle avec les services de localisation. Alice se rend compte que même si elle

conserve ses coordonnées géographiques, une application LBS installée peut toujours corréler d'autres données sensibles, telles que des photos, des événements d'agenda et des contacts.

Scénario 3. Une semaine plus tard, Alice rencontre ses amies au café. Son amie Carol prend une photo de groupe dans l'intention de la télécharger vers son profil dans son réseau géosocial préféré. Carol partage souvent en mentionnant les personnes prises en photo et en joignant les données de géolocalisation. Cette fois-ci, Alice refuse que la photo soit partagée en ligne, mais apparemment elle est la seule qui ait un souci. Le reste des amies d'Alice n'a aucun problème avec le partage de la photo ainsi que des données contextuelles telles que le marquage ou la position. Alice refuse de partager la photo, et vu qu'elle est dessus elle possède une partie de la propriété intellectuelle.

En analysant ces scénarios, nous confirmons que les LBS envahissent plusieurs aspects de notre vie quotidienne incluant notre façon de communiquer, de partager ou d'obtenir de l'information. Ainsi, un LPPM efficace doit prendre en considération tous ces aspects, être facile à utiliser, et surtout transparent afin de permettre aux utilisateurs de tirer le maximum du profit.

3.3.2 Solution proposée

Bien que les scénarios définis précédemment soient hypothétiques, ils sont proches de situations réelles et peuvent arriver à n'importe quel utilisateur. Ainsi, l'identification des problèmes de confidentialité dans les scénarios précédents mène à définir les 7 exigences suivantes qu'un LPPM efficace doit garantir.

- ***Confidentialité de la position***. Protéger efficacement la confidentialité de la position de l'utilisateur, c'est-à-dire ne pas être en mesure d'identifier ou de déduire une véritable position.
- ***Anonymat des données de l'utilisateur***. Ne pas être en mesure d'identifier la véritable identité de l'utilisateur. Plus précisément, assurer l'anonymat de l'émetteur

de la requête.

- ***Utilité maximale des données***. Garantir une haute précision et une utilité maximale.

- ***Déni de la position et de l'utilisateur***. Ne pas pouvoir lier une ou plusieurs requêtes géodépendantes à un autre utilisateur, ou à ses coordonnées dans le monde réel. En d'autres termes, l'insubmersibilité garantit que chaque requête géodépendante n'est pas liée à son émetteur, aux requêtes d'autres émetteurs ou à ses coordonnées réelles.

- ***Indétectabilité des données de localisation***. Ne pas être en mesure d'apprendre la relation entre les données stockées sur leurs serveurs et les requêtes géodépendantes de l'utilisateur. Par exemple, la détermination de l'existence d'une liste de contacts associée à l'émetteur de la requête actuelle et celle du contenu de la base de données à disposition de l'adversaire doit être non réalisable.

- ***Flexibilité et extensibilité***. Pouvoir prendre en charge un large éventail d'applications et être capable de gérer la croissance potentielle des utilisateurs, des demandes et des interactions.

- ***Efficience***. Être capable de se déployer et de fonctionner efficacement sur les dispositifs mobiles, tout en conservant un temps de calcul minimum, de bons facteurs d'efficacité, et une consommation modérée de la bande passante. Lorsque possible, l'exécution ne devrait pas affecter la latence et le temps de réponse.

Compte tenu des applications actuelles, les exigences ci-dessus sont fondamentales. En fait, les LBS d'aujourd'hui utilisent la localisation ainsi que d'autres données sensibles, et leur capacité à apprendre la position des utilisateurs évolue rapidement. Considérons une utilisatrice Alice préoccupée par sa vie privée et utilisait un LPPM donné pour y parvenir. Ce dernier essaie de rendre les requêtes géodépendantes d'Alice indiscernables parmi un ensemble de positions (*paradigme de confusion*). Cependant, le LBS

pourrait accéder à plusieurs types de données sur le dispositif d'Alice et pourrait éventuellement identifier si la requête provient d'Alice en corrélant les données actuelles et antérieures.

La plateforme que nous proposons repose sur deux composants principaux, un composant qui se charge de la transmission des requêtes et résultats géodépendants que nous appelons le *processus de délégation*, et un autre qui estime les risques et le niveau de confiance associés à une opération de partage, appelé processus de *quantification de risques et de confiance*.

Notre plateforme possède également un composant d'évaluation de confidentialité qui se base sur une nouvelle notion nommée "δ-*fuzziness*" que nous détaillerons dans le chapitre 6. La figure 3.2 illustre l'architecture globale de notre plateforme en spécifiant le flux de données initial et les principales étapes qui en font partie. Nous utilisons le terme "nœud" pour décrire un dispositif mobile utilisé dans les processus de la plateforme.

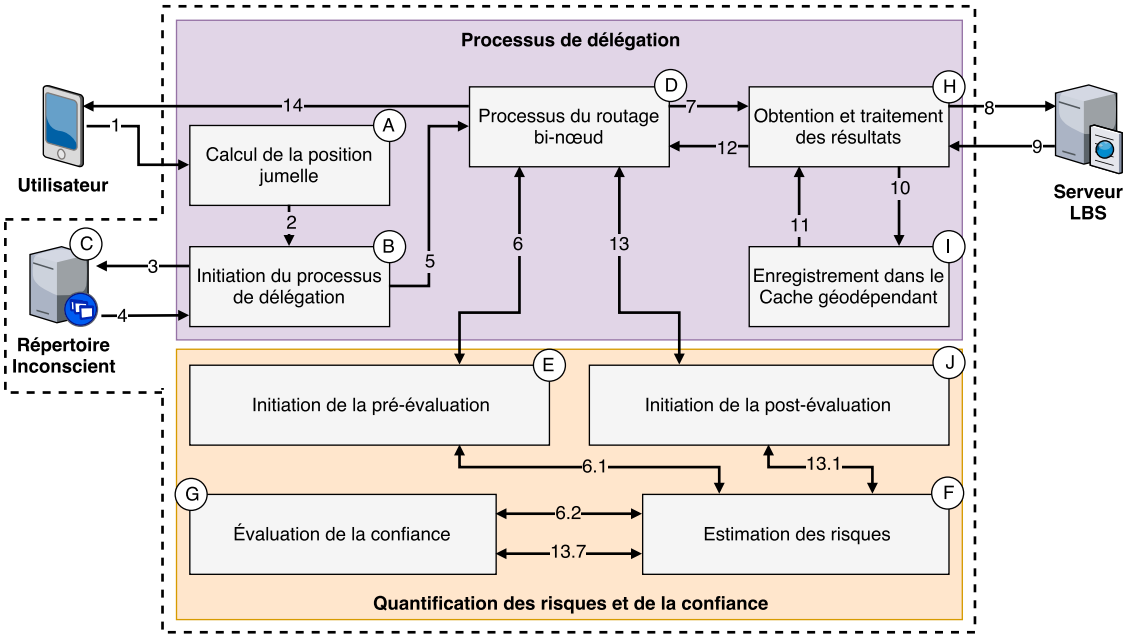


Figure 3.2 – Architecture globale de Deloc

Tel qu'illustré dans la figure 3.2, le flux de données est initié par un utilisateur qui veut récupérer des informations géodépendantes dans (A), la plateforme calcule ensuite

une position de substitution que nous appelons *position jumelle* (section 4.1.2), (B) puis elle initie le processus de délégation qui utilise un (C) répertoire qui fournit les utilisateurs susceptibles de participer aux processus, appelé *répertoire inconscient*, basé sur la primitive cryptographique connue sous le nom de *transfert inconscient*. Une fois le nœud actif récupéré, (D) la plateforme débute le déroulement du processus du routage (section 4.1.5), décrit en détail dans le chapitre suivant. Quand la requête est en chemin vers le LBS, (J) le processus de routage implique la vérification des scores de confiance et des risques associés à chaque nœud. À la fin du processus (H) le dernier nœud délégué communique avec le LBS pour récupérer les résultats géodépendants. Ce même nœud (I) enregistre les résultats dans son cache géodépendant pour des utilisations futures (section 4.1.6). Les résultats ensuite prennent le même chemin pour revenir au demandeur initial. La plateforme ré-estime les risques possibles ainsi que le niveau de confiance occasionnés par les nœuds participants pour des fins de validation et de vérification dans (E). Les résultats continueront leurs chemins vers le demandeur initial qui pourra utiliser le LBS sans avoir besoin d'être identifié.

Mentionnons ici que les numéros associés aux flèches n'ont pas de valeurs dans cette figure, et seront discutés en détail dans les chapitres qui suivent. Leur mention ne sert ici qu'à donner une idée sur l'ordre de déroulement des processus.

3.4 Conclusion

Les premières implications sur la confidentialité dans les services géodépendants ont été démontrées en 2010 par le site "Please Rob Me" [61], qui reliait les positions physiques des individus à travers des services de géolocalisation avec des données sur leur lieu de résidence et d'autres données publiques. Même si l'intention de ce site n'est pas, et n'a jamais été, d'avoir des gens cambriolés, il a réussi à démontrer que lorsque les utilisateurs partagent leurs positions, des individus malintentionnés pourraient les découvrir et profiter de ces informations.

En outre, beaucoup d'utilisateurs ne connaissent peut-être pas la quantité d'information qu'ils diffusent publiquement, et les services qui collectent et exploitent ces données.

Pour avoir une idée précise de la sensibilité des données de localisation, des travaux ont démontré qu'en utilisant des attaques d'inférence sur les traces d'un utilisateur, il est possible de découvrir ses points d'intérêt (POI), où il vit, où il travaille, ses itinéraires habituels, ses habitudes, intérêts, activités, relations, et autres [45].

Notons que cette technologie soit utile pour de nombreux aspects de la vie quotidienne, les informations recueillies peuvent également exploitées par des parties malveillantes. Une personne malveillante peut examiner les données d'un utilisateur et analyser et inférer des informations qui pourraient être utilisées pour générer des modèles comportementaux concernant ses déplacements et activités. Par exemple, que se passait-il si un tueur en série voulait localiser des victimes potentielles ? Paul Bernardo, le tueur en série bien connu au Canada, aurait plus de facilité à trouver ses victimes que dans les années 90 [86]. Si le fait d'exposer nos informations présente déjà un sérieux problème de confidentialité, il est encore plus grave lorsque des données de localisation sont ajoutées. Étant donné que l'adoption de telles technologies exige plus de sensibilisation et de vigilance, il n'est pas difficile d'imaginer des dangers liés à la vie privée. En conclusion, la facilité de localiser des individus est la même que d'être localisée par les autres, incluant les adversaires.

Deuxième partie

Deloc : Un LPPM Collaboratif

CHAPITRE 4

DÉLÉGATION DES TÂCHES GÉODÉPENDANTES

Une des principales étapes de la réalisation de la plateforme holistique est bien celle de la délégation des requêtes géodépendantes. L'idée générale derrière cette opération consiste à protéger la vie privée d'un utilisateur des services géodépendants, dit utilisateur LBS, en transférant le contenu de sa requête à un autre utilisateur qui à son tour effectuera la communication avec le service géodépendant. La transmission du contenu est appelée le processus de délégation, et elle est accomplie selon des règles, des mécanismes et des protocoles bien définis.

En revenant à l'architecture proposée dans la figure 3.2, nous représentons les composants du processus de délégation dans la figure 4.1. Notons que le processus de délégation peut s'exécuter indépendamment du module de quantification de risques et de confiance, ce dernier pouvant interagir avec le processus de délégation que pendant l'opération du routage bi-nœud.

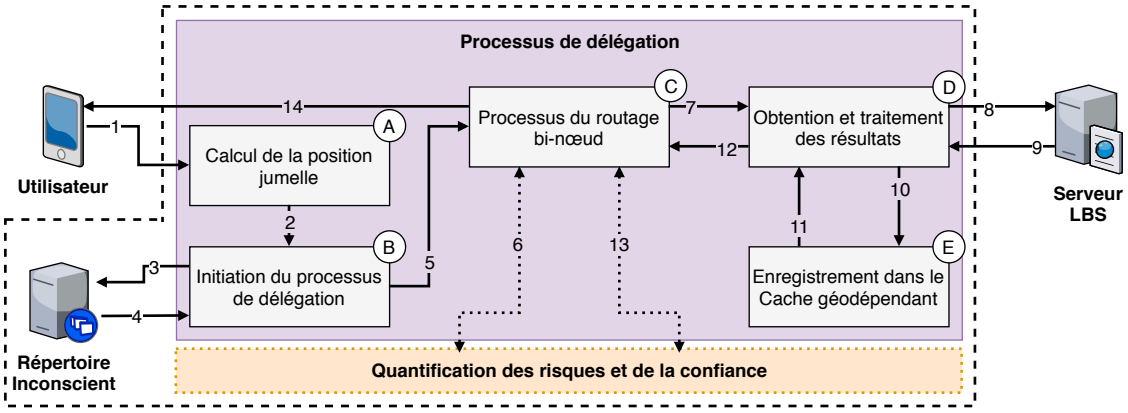


Figure 4.1 – Architecture des composants du processus de délégation

En résumé, au lieu de transformer les coordonnées géographiques, le mécanisme de délégation que nous proposons transmet la requête d'un utilisateur à un autre de façon itérative jusqu'au dernier utilisateur qui est responsable de l'exécution de la requête. Les

utilisateurs font partie d'un réseau géré par *Deloc* que nous appelons le *réseau de la foule de délégation*. Le processus commence par le calcul de la *position jumelle* dans (A) qui représente une alternative aux mécanismes de substitution. Ensuite, les requêtes sont transmises entre utilisateurs en suivant un chemin défini dynamiquement dans (C) à chaque exécution selon un mécanisme que nous nommons le *routage bi-nœud de délégation*. La construction du chemin dynamique dépend essentiellement de la liste des membres disponibles dans le réseau fourni dans (B) par l'entité appelée *répertoire inconscient*. À la fin de l'exécution de chaque processus de délégation, les résultats associés à la requête géodépendante sont enregistrés localement dans le dispositif du dernier utilisateur délégué dans (E), et cela dans une partition gérée par *Deloc* que nous appelons le *cache géodépendant*. Nous détaillerons chacun de ces composants dans le reste du chapitre, ainsi que l'architecture globale et les principaux modèles inhérents au processus de délégation. Nous discuterons également de notre implémentation du prototype, ainsi que de différents tests de validation effectués et leurs résultats.

Nous supposons que toutes les coordonnées géographiques du monde réel sont initialement projetées vers des points dans un espace euclidien fini et bidimensionnel représentant les coordonnées géographiques (longitude et latitude). Désigné par $\mathbb{E}^2$, toutes les opérations géographiques s'effectuent dans ce même espace euclidien. De même, nous supposons que les LBS ne collectent pas de données satellitaires autres que les coordonnées géographiques telles que l'azimuth (angle dans le plan horizontal entre la direction des coordonnées et une direction de référence), la dilution de précision (effet multiplicatif de la géométrie satellite sur la précision d'un système de positionnement), ou le nombre de satellites utilisés dans la détermination de la position, des données qui sont souvent extraites pour corriger et augmenter la précision de la position [104].

Même en ignorant cette dernière supposition, et en supposant qu'un LBS collecte des données satellitaires, le dispositif impliqué dans notre contexte est toujours le dispositif final du processus de délégation. De ce fait, déléguer une requête du demandeur initial, et collecter des données satellitaires depuis le dispositif final ne rajoute aucune précision

aux coordonnées géographiques. Dans le reste de ce chapitre, nous utiliserons les termes "nœud", "dispositif" et "utilisateur" pour mentionner un utilisateur de *Deloc*.

4.1 Modélisation et architecture

La délégation des requêtes géodépendantes peut être considérée comme un mécanisme collaboratif, où les utilisateurs participent eux-mêmes à la protection de leurs pairs. Le principe d'un mécanisme collaboratif est que soit les autres utilisateurs fournissent des résultats géodépendants, soit ils participent aux étapes du processus de protection (*par ex.* transformation des coordonnées). L'efficacité de ce genre de mécanisme réside dans le fait que les utilisateurs ont souvent tendance à collaborer, afin d'accomplir une co-utilité qui sera bénéfique pour tous les membres participants [34].

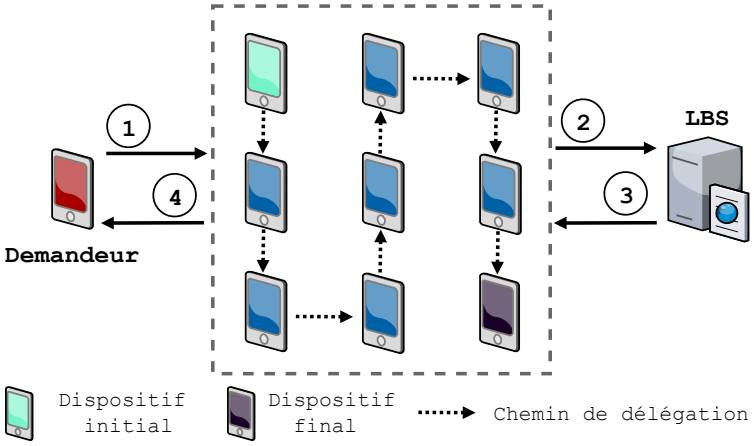


Figure 4.2 – Architecture globale du processus de délégation

Le modèle de base derrière *Deloc* est conçu de façon à minimiser les opérations de transformation des coordonnées géographiques. Les utilisateurs collaborent à l'exécution des processus de délégation afin de garantir la confidentialité des données de chacun. La figure 4.2 illustre l'architecture globale de notre proposition d'un LPPM basé sur la délégation. Supposons que Bob, un utilisateur, veuille explorer des restaurants autour de sa position actuelle. D'abord, le dispositif de Bob calcule la position jumelle, puis il

envoie la requête au réseau de la foule de délégation géré par *Deloc* dans (1). Ensuite, *Deloc* effectue les opérations de délégation avant de transmettre la demande au service géodépendant dans (2). Le LBS renvoie la liste des restaurants au dernier dispositif dans le chemin de délégation (*route end device*) (3), qui le stocke dans son propre cache avant de la renvoyer vers le dispositif adjacent. Enfin, le dispositif adjacent à Bob (*route start device*) renvoie les résultats obtenus à Bob dans (4).

4.1.1 Les systèmes de communication anonyme

Ces systèmes, qui incluent Tor, assurent l'anonymat et la vie privée des utilisateurs lors des communications réseau, souvent effectuées via Internet [33]. Cependant, ces systèmes se comportent le plus souvent de la même façon qu'un Réseau Privé Virtuel (RPV). Par conséquent, les requêtes envoyées en utilisant Tor ou d'autres alternatives contiennent la totalité du contenu initial. Autrement dit, le service en ligne pourra accéder à toutes les informations contenues dans une requête, mais sans connaître son origine. Une telle approche est avantageuse quand le contenu de la requête ne contient pas de données identificatrices, ce qui n'est pas le cas dans les LBS.

Un service géodépendant reste en mesure de collecter toutes les données de ses utilisateurs avec une seule particularité, qui est le chemin que la requête prendra. En d'autres termes, un système de communication anonyme prend la requête en entier, et essaye de brouiller son chemin afin de minimiser la possibilité de la retracer. Une pratique insuffisante dans le cas des LBS, où la requête contient déjà des données concernant le profil du demandeur initial qui peuvent être utilisées pour l'identifier.

L'inefficacité des systèmes de communication anonyme dans le contexte des services géodépendants réside essentiellement dans la non-séparation du contenu géographique inclus dans la requête, et le reste de la requête. En utilisant Tor, par exemple, un LBS pourra toujours accéder aux coordonnées géographiques qui feront partie du contenu de la requête. La figure 4.3 démontre la possibilité de déterminer des coordonnées géographiques, et cela même en utilisant le système de communication anonyme le plus utilisé

Tor. En définitive, l'architecture et les concepts derrière un système de communication anonyme ne peuvent être applicables aux services géodépendants, où les coordonnées géographiques sont des données sensibles.

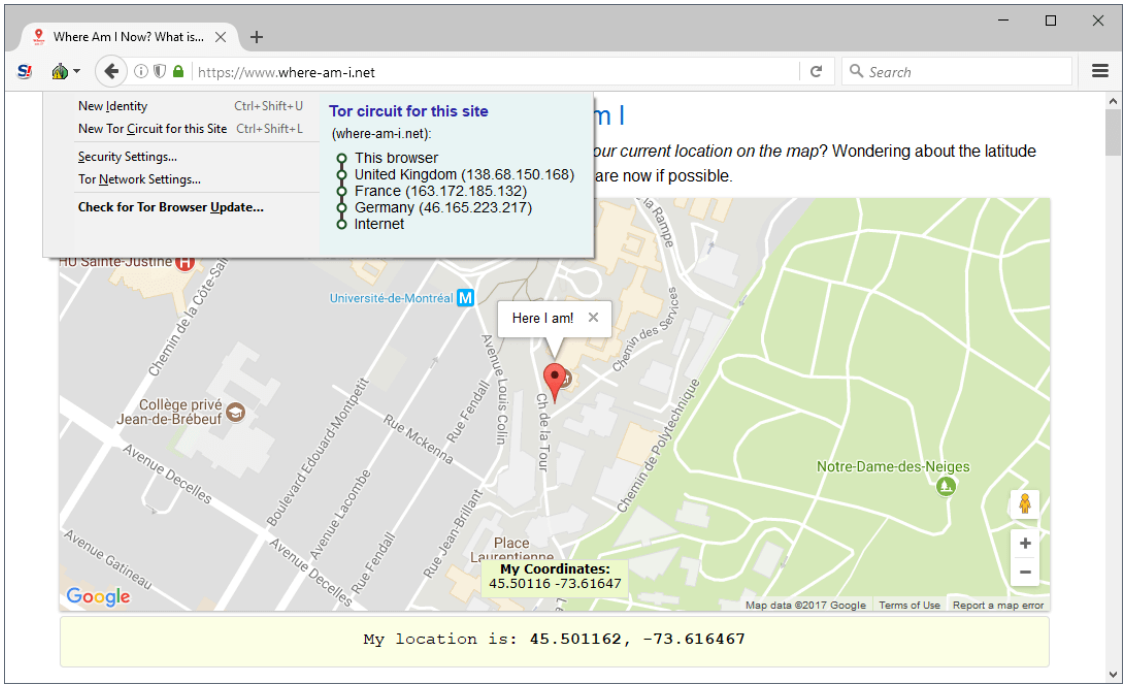


Figure 4.3 – Détermination de localisation en utilisant Tor pour l'anonymisation

Tel qu'illustré dans la figure 4.3, même si Tor anonymise la requête en la passant par des clients situés au Royaume-Uni, en France et en Allemagne respectivement, le fait qu'il ne considère pas le contenu géographique comme entité indépendante le rend inefficace face au LBS.

De plus, plusieurs travaux ont montré que la corrélation entre des requêtes anonymisées et des coordonnées géographiques associées à chaque requête peut révéler l'identité des utilisateurs [46, 52, 59, 70]. L'altération du chemin de la requête n'est pas suffisante pour garantir la vie privée dans certains services en ligne, notamment les LBS.

Dans notre cas, la différence réside essentiellement dans la façon de composer des requêtes et de les transmettre. Au lieu d'une simple altération du chemin, le processus de délégation assure que le dispositif délégué agit au nom du demandeur de façon indépen-

dante, tout en gardant la vie privée du délégant et des délégués utilisateurs, et en délivrant les résultats les plus exacts possible. De plus, *Deloc* vise la protection des données géographiques dans un contexte mobile, où les utilisateurs exploitent les fonctionnalités de localisation de leurs dispositifs mobiles.

En conclusion, le processus de délégation est différent d'un système de communication anonyme dans la façon de traiter le contenu d'une requête. De prime abord, les deux concepts semblent similaires du fait que les deux assurent que la requête est transmise par un autre utilisateur. Cependant, la comparaison au niveau de l'architecture et des objectifs de protection démontre une dissemblance entre les deux types de systèmes.

4.1.2 Position jumelle

Afin d'éviter certains problèmes de vie privée, qui sont dus à la génération aléatoire des positions, nous proposons la notion de *position jumelle*. Elle a pour but d'assurer la protection contre les attaques géographiques telles que les attaques du mouvement maximal qui visent à définir la position réelle d'un individu en analysant la zone maximale de génération randomisée des positions.

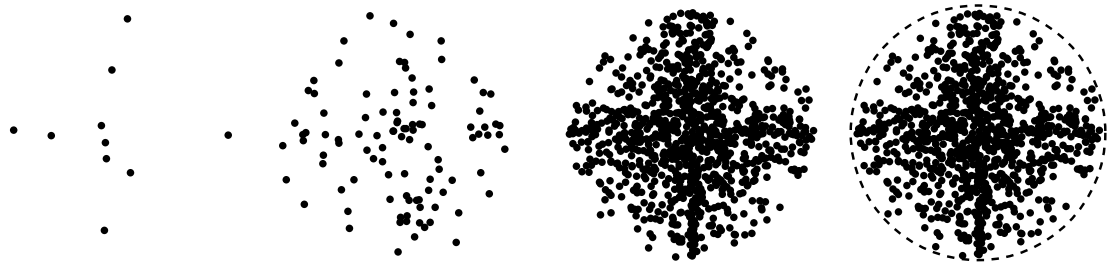
Une position jumelle l_t est une position *unique* associée à chaque position distincte l_u de l'utilisateur u. Elle est unique dans le sens où chaque position réelle n'est associée qu'à une position jumelle. Elle est définie par une paire de coordonnées géographiques $(t_x, t_y) \in T_{loc}$ calculée à partir de la position actuelle de l'utilisateur. Par conséquent, une position jumelle fait partie de l'ensemble T_{loc} composé des coordonnées géographiques situées dans un cercle défini par l'équation $(x - l_x)^2 + (y - l_y)^2 = r^2$, où (l_x, l_y) sont les coordonnées de la position actuelle de l'utilisateur, et chacun de x et y est délimité par le rayon d'une zone de calcul prédéfinie d_u que nous discutons dans la suite.

La caractéristique principale des positions jumelles est l'unicité des coordonnées générées. Plus formellement, la définition de la position jumelle se fait par le calcul de la longueur len_t et de l'angle de rotation θ_t du vecteur $\overrightarrow{l_u l_t}$ en utilisant des propriétés que nous appelons les *attributs de calcul*. Ces derniers, qui permettent le calcul des po-

sitions jumelles, représentent des propriétés uniques associées à chaque position dans $\mathbb{E}^2$. Des exemples d'attributs d'association peuvent être les coordonnées de la position réelle, les identifiants uniques de l'utilisateur et du LBS, le contexte actuel de la position, etc. Une bonne combinaison de différents attributs d'association permet de calculer une position jumelle unique pour n'importe quel utilisateur, en utilisant n'importe quel service géodépendant. Ainsi, une fonction de calcul de positions jumelles *twin* prend la position réelle de l'utilisateur l_u et un ensemble d'attributs de calcul $attr_u$ pour renvoyer les propriétés d'un vecteur de position délimité par le rayon de la zone de calcul d_u et ayant l_u comme origine.

$$twin(l_u, d_u, attr_u) = (len_t, \theta_t) \quad \text{tel que} \quad len_t \leq d_u \tag{4.1}$$

L'avantage principal d'utiliser une position unique pour remplacer la position réelle de l'utilisateur réside dans la prévention contre les attaques multi-requêtes et celles qui essayent de définir les limites maximales de mouvement (chapitre 2). En d'autres termes, un LPPM qui génère une position aléatoire à partir de la même position réelle à chaque requête finira par révéler la circonférence et le centre de la zone de calcul utilisée. Cette zone est souvent utilisée pour cacher la position réelle de l'utilisateur. La figure 4.4 illustre un exemple d'une attaque liée à l'utilisation de positions générées de manière aléatoire.



(a) 10 requêtes (b) 100 requêtes (c) 1000 requêtes (d) zone de calcul

Figure 4.4 – Exemple des attaques causées par la génération randomisée des positions

Tel qu'illustré dans la figure 4.4, la génération aléatoire des positions fictives autour

des mêmes coordonnées géographiques mène à révéler la zone de calcul, et éventuellement la position exacte de l'utilisateur. Ce type de problèmes peut être évité par l'utilisation d'une seule position calculée de façon unique sans l'utilisation de génération randomisée.

Pour bien comprendre le calcul de positions jumelles, prenons l'exemple d'un utilisateur u qui a l'identifiant ID_u, et qui est situé dans les coordonnées géographiques (45.500936, -73.616637). Supposons qu'il envoie une requête à *Google Maps* pour localiser les bibliothèques à proximité. La fonction *twin* dans ce cas prend comme entrée toutes ces informations comme suit : *twin((45.500936, -73.616637), 500, {ID_u, "Google Maps", "libraries" })* tel que 500 représente la taille de la zone de calcul. La fonction *twin* dans ce cas retourne des valeurs *len* et θ qui dépendent des attributs envoyés en entrée. Un exemple de déroulement détaillé est discuté dans le chapitre d'implémentation (chapitre 7).

4.1.3 Réseau de la foule de délégation

Les opérations de délégation s'effectuent dans un réseau superposé qui se compose des utilisateurs de *Deloc*, appelé *le réseau de la foule de délégation*. Les dispositifs des utilisateurs (appelés aussi nœuds) sont reliés via Internet, et communiquent par le protocole HTTP. La communication entre nœuds est cryptée par un chiffrement de bout en bout qui assure la confidentialité des données échangées au sein du réseau de la foule de délégation.

Définition 4.1. *Un réseau de la foule de délégation géré par un LPPM $\mathscr{P}$ est un réseau pair-à-pair superposé*[1] *défini par le graphe $G_d = (V_d, E_d)$, où les nœuds dans V_d désignent les dispositifs utilisant $\mathscr{P}$, et les segments de E_d représentent les liens de communication.*

Comme tout réseau superposé ou non, les caractéristiques clés peuvent être illustrées

[1]Un réseau superposé est un réseau construit au-dessus d'un autre réseau. Dans notre cas, c'est un réseau logique au-dessus du réseau Internet.

par la topologie, et les délais de transmission. Dans notre contexte, pendant que la topologie est de type pair-à-pair (P2P), les délais de transmission sont principalement évalués par la *latence* et la gestion de *congestion* dans le réseau.

4.1.3.1 Latence et congestion du réseau

Étudier les problématiques de latence et de congestion nécessite tout un domaine de recherche [38, 76]. Dans notre cas, nous ne considérons pas celles liées aux infrastructures et aux interfaces de connexion réseau (*par ex.* WiFi, réseau mobile), nous nous concentrons plutôt sur l'efficacité de la communication au sein du réseau de la foule de délégation.

Contrairement aux réseaux P2P conventionnels, la taille des requêtes transmises au sein du réseau de la foule de délégation est fixe et relativement petite. Ce qui fait que l'efficacité de la transmission de données entre deux nœuds dépend uniquement de *la latence* du réseau, un paramètre affecté essentiellement par les canaux de communication. En d'autres termes, les données échangées dans le réseau peuvent être soit des requêtes ou des résultats géodépendants. Ainsi, la taille maximale autorisée pour une requête à l'intérieur du réseau est définie pour permettre de gérer ce type de données.

Le délai de transmission d'une requête entre deux utilisateurs à l'intérieur du réseau est mesuré par le temps passé par un paquet lors de l'échange entre 2 usagers. Par conséquent, *la congestion* ne peut se produire que lorsque les paquets passent beaucoup de temps à la transmission, un cas souvent causé par une faible bande passante ou par une requête de grande taille. Un exemple du monde réel pourrait être comme suit : étant donné que 0.41 Mbit/s est la moyenne du débit d'Internet le plus bas au monde [116], une requête de 512 octets peut être échangée en $\approx$ 10 ms, une taille qui permet facilement l'échange d'une requête géodépendante.

La taille des résultats géodépendants à son tour, ne représente pas une contrainte, par exemple, l'ensemble des résultats géodépendants autour d'une position, extrait de *Google Maps API*, mesure à peine 1024 octets. En fait, une approche simple mais effi-

cace pour remédier aux délais de transmission élevés et la possibilité de congestion au sein du réseau de la foule de délégation, est la délimitation des tailles de requêtes.

4.1.3.2 Latence et vie privée

Un autre point à souligner est celui de l'impact de la latence sur la vie privée [60]. L'utilisation du même routage consécutivement pour effectuer les processus de délégation peut révéler l'identité du demandeur initial. Par exemple, si l'un des utilisateurs du réseau est un adversaire et fait partie des dispositifs du chemin de routage, il peut identifier le nombre de nœuds impliqués dans le processus en cours en mesurant le temps qu'une requête prendra pour l'atteindre. La solution que nous proposons est l'utilisation de routage dynamique aléatoire discuté en détail dans la section 4.1.5. En résumé, le chemin de routage doit être recalculé et changé à l'exécution de chaque opération, en se basant sur les mécanismes du répertoire des dispositifs actifs (section 4.1.4).

Nous avons mentionné dans l'exemple précédent qu'il est parfois possible qu'un des utilisateurs du réseau soit malveillant (c.-à-d. adversaire). Cependant, même si un adversaire infiltre le réseau de la foule de délégation, il ne sera pas en mesure d'identifier l'origine de la requête ni d'inférer des informations concernant le trafic dans le réseau. Les seules informations auxquelles un utilisateur du réseau peut accéder sont la requête et les résultats géodépendants, ainsi que le dispositif adjacent direct dans le chemin de routage. Autrement dit, les données échangées au sein du réseau ne comportent aucune information qui peut distinguer le demandeur initial des autres membres du chemin de routage.

4.1.4 Répertoire inconscient

Inspiré du protocole de transfert inconscient (*Oblivious Transfer*) [98], le répertoire inconscient renvoie la liste des dispositifs actifs aux utilisateurs autorisés sans connaître quel dispositif est utilisé dans le processus de délégation. Un utilisateur autorisé est un participant au réseau de la foule de délégation et qui détient un jeton d'authentification

valide lui permettant de demander au répertoire la liste des dispositifs disponibles.

Même s'il est indépendant et qu'il n'est hébergé sur aucun dispositif d'utilisateur, le répertoire inconscient est considéré comme faisant partie du réseau de la foule de délégation. Son objectif principal est de fournir à *Deloc* la liste des utilisateurs actuellement disponibles pour un processus de délégation. Le répertoire inconscient est hébergé dans un serveur indépendant en dehors du réseau de la foule de délégation, et il est géré par ses propres mécanismes internes. En fait, ses principales fonctionnalités peuvent se résumer à la génération de la liste des dispositifs actifs et à la gestion d'utilisation des différents nœuds du réseau.

Afin d'assurer les meilleures performances au sein du réseau, la liste renvoyée par le répertoire inconscient ne contient pas tous les dispositifs disponibles dans le réseau. Lorsqu'une demande est reçue, le répertoire renvoie une liste avec les nœuds les moins utilisés. Le principe est le suivant : quand un dispositif utilisant *Deloc* se connecte à Internet, il rejoint automatiquement le réseau de la foule de délégation, son identifiant et sa fréquence d'utilisation sont ensuite envoyés au répertoire inconscient. Ce dernier vérifie périodiquement la disponibilité des dispositifs au réseau, et supprime automatiquement les données des dispositifs inactifs. Autrement dit, les informations enregistrées dans le répertoire inconscient ne sont pas persistantes, et sont gardées seulement si leur propriétaire est un utilisateur potentiel de délégation.

Afin d'atteindre l'indiscernabilité des dispositifs sélectionnés, nous utilisons une variante légère du transfert inconscient pour assurer que la liste retournée est anonyme. Plus formellement, *la liste des dispositifs actifs* doit se conformer aux exigences suivantes afin d'assurer l'efficacité et l'anonymat :

- Elle ne doit pas contenir tous les dispositifs enregistrés dans le répertoire si la taille de ce dernier est relativement large.
- Elle doit contenir au moins 2 appareils, ce qui laisse la probabilité qu'un dispositif soit impliqué dans le processus de délégation égale à 0,5.

- Elle doit garder une taille relative à la taille totale du répertoire inconscient.

En considérant les exigences ci-dessus, la méthode la plus appropriée pour calculer la taille de la liste pourrait être la *factorisation entière* de la taille actuelle du répertoire. La plus petite taille de liste des dispositifs actifs est de 2 (cas des nombres premiers) ; la plus grande taille est toujours inférieure à la taille totale du répertoire (sauf dans le cas d'un répertoire contenant 1 ou 2 dispositifs) ; et la taille de la liste pour un répertoire contenant un nombre de tuples aussi grand que 10^{12} par exemple, est de seulement 170 dispositifs.

En conséquence, étant donné un répertoire inconscient A contenant n dispositifs actifs, la taille s_l de la liste renvoyée l est le nombre de facteurs entiers de n. La figure 4.5 illustre un exemple d'une liste envoyée en réponse au répertoire inconscient contenant 5 éléments. Le répertoire est d'abord trié en fonction de la fréquence d'utilisation dans (b), puis les dispositifs disponibles sont sélectionnés en fonction du nombre de facteurs entiers dans (c).

ID	Usage
device_1	25
device_2	13
device_3	7
device_4	43
device_5	5

(a)

ID	Usage
device_4	43
device_1	25
device_2	13
device_3	7
device_5	5

(b)

ID	Usage
device_4	43
device_1	25
device_2	13
device_3	7
device_5	5

(c)

Figure 4.5 – Exemple de sélection des dispositifs actifs dans le répertoire inconscient

Afin de bien comprendre le processus de génération de liste, la figure 4.6 illustre en détail les actions suivies par une demande de la liste des appareils actifs.

Tel qu'illustré dans la figure, la communication entre l'utilisateur et le répertoire est chiffrée à l'aide de la paire de clés (d,e), et l'accès à la liste des dispositifs actifs est réservé aux utilisateurs autorisés uniquement en utilisant le jeton d'identification

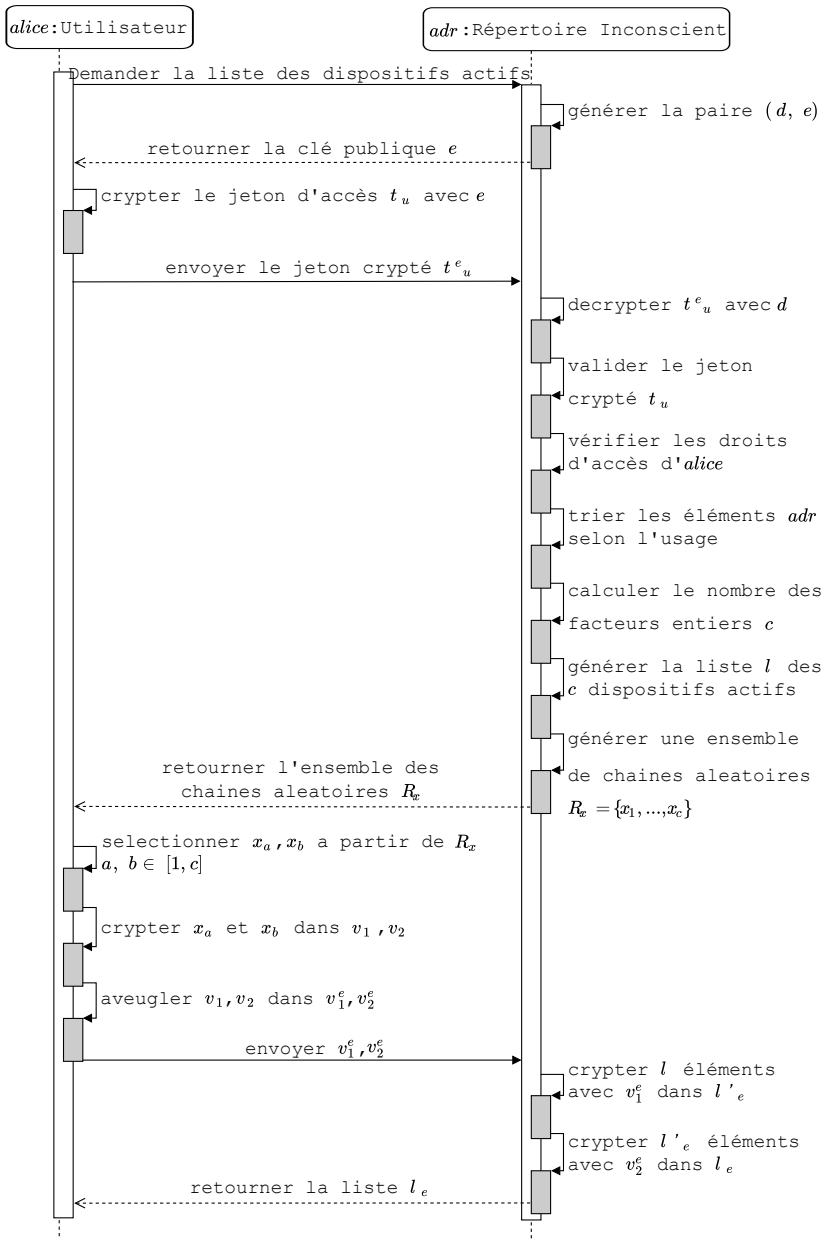


Figure 4.6 – Génération de la liste des dispositifs actifs

t_u. De même, le répertoire inconscient ne peut savoir quel dispositif est destiné à être sélectionné dans le processus de délégation. Lorsque le répertoire calcule les facteurs entiers et génère la liste des appareils actifs l, il génère une liste supplémentaire de chaînes aléatoires R_x de la même taille que l. L'utilisateur sélectionne alors deux chaînes

aléatoires x_a et x_b de R_x, et crée deux clés de cryptage v_1, v_2 selon la sélection.

La version aveugle de ces clés, v_1^e et v_2^e respectivement, est utilisée par le répertoire inconscient pour chiffrer la liste l dans l_e. Une version aveugle de ces clés réfère dans ce contexte aux mécanismes de *blinding* utilisés dans la cryptographie pour permettre une fonction de cryptage de chiffrer des données sans avoir besoin de connaître les entrées et les sorties [51]. Dans notre contexte, la version aveugle des clés (sortie) est inconnue pour l'utilisateur (propriétaire de la fonction de cryptage) afin de respecter les exigences du transfert inconscient, et d'assurer la confidentialité de la liste.

Enfin, l'utilisateur décrypte l_e dans deux opérations successives dans l'ordre inverse du cryptage, pour obtenir deux dispositifs actifs, un seul à chaque opération. De plus, pour empêcher l'identification du dispositif, le protocole dans la figure 4.6 empêche les adversaires potentiels du réseau de délégation d'identifier l'émetteur des requêtes. Ce dernier cas se produit lorsqu'un nœud interroge continuellement les dispositifs actifs et les compare avec les demandes de délégation reçues pour distinguer les dispositifs impliqués dans la délégation des émetteurs de la demande.

Un tel scénario n'est pas réalisable lors de l'utilisation du transfert inconscient décrit par la figure 4.6 en raison de deux caractéristiques principales : la liste des appareils actifs continue de changer en fonction de la fréquence d'utilisation et de la taille du répertoire inconscient, et le dispositif qui demande la liste ne peut accéder à la fois qu'à deux dispositifs et qui sont susceptibles de changer à l'initiation d'une nouvelle requête.

Prenons l'exemple de notre utilisateur Alice qui demande la liste de dispositifs actifs à partir du répertoire *adr*. Pour des raisons de simplification, nous représentons les clés par des valeurs considérablement petites par rapport aux clés utilisées dans un environnement réel.

1. Le répertoire génère une paire de clé publique et privée (d, e) et renvoi e à Alice. Supposons que la valeur de e est 16.

2. Alice crypte son jeton d'accès t_u avec e et rend le résultat au répertoire inconscient.

3. Le répertoire décrypte le jeton d'Alice avec la clé privée qu'il possède, vérifie ses droits d'accès, trie les dispositifs enregistrés par ordre de fréquence d'usage, et calcule le nombre des facteurs entiers c. Supposons qu'il possède 18 dispositifs enregistrés, la valeur de c est donc égale à 6 (les facteurs de 18 sont 1, 2, 3, 6, 9, 18).

4. Le répertoire inconscient sélectionne les 6 premiers dispositifs parmi ceux enregistrés et ordonnés par fréquence d'usage en les mettant dans une liste l.

5. Ensuite, il génère un ensemble de six chaines de caractère aléatoire R_x et il le transmet à Alice. Supposons que $R_x = \{A, B, C, D, E, F, G\}$.

6. Alice sélectionne deux chaines de caractères au hasard à partir de R_x. Supposons qu'elle choisisse D et G.

7. Elle les crypte et les aveugle ensuite afin de donner deux autres chaines de caractères. Supposons que le résultat est M et Y. Notons ici que les clés et les mécanismes de cryptage utilisés par Alice sont inconnus par le répertoire inconscient.

8. Après la réception des chaines de caractères M et Y par le répertoire inconscient, il crypte la liste l à deux reprise. La première avec la chaine M et la deuxième avec Y. Il envoie ensuite le résultat à Alice.

9. À la réception de la liste cryptée, Alice utilise Y pour un premier décryptage qui donne en résultat le premier dispositif, et M pour le deuxième décryptage qui produit le deuxième dispositif.

En résultat, Alice peut utiliser les deux dispositifs comme paire de délégations sans que le répertoire inconscient prenne connaissance des identifiants des dispositifs choisis.

Les étapes de génération de la liste sont simples, transparentes à l'utilisateur, et suffisantes pour fournir des résultats efficaces du côté vie privée et communication. Un processus supplémentaire qui pourrait améliorer la communication au sein du réseau est

la sélection des dispositifs actifs proches en fonction de leurs adresses IP. Cette approche aide à réduire le temps d'une requête pour passer d'un nœud à un autre. Sa réalisation n'exige que l'utilisation de l'adresse IP d'un nœud, obtenue au moment de son enregistrement au répertoire.

4.1.5 Chemin bi-nœud de délégation dynamique

Comme son nom l'indique, le chemin de délégation est le chemin emprunté par la requête de l'utilisateur dans le réseau de la foule de délégation. Après la réception de liste des dispositifs actifs à partir du répertoire inconscient, *Deloc* sélectionne la première paire de dispositifs susceptibles de participer dans le processus de délégation en cours.

Une paire de dispositifs est composée de celui utilisé pour le processus de délégation, et d'un dispositif du support (*backup device*) qui sera utilisé pour assurer le routage dans le réseau si le premier dispositif devient indisponible. À son tour, l'utilisateur sélectionné choisit la prochaine paire à recevoir la requête, tout en gardant les identifiants de la source adjacente et le dispositif du support associé à cette source.

La requête est récursivement transmise d'un dispositif à l'autre jusqu'à la dernière paire de dispositifs actifs, qui à son tour la renvoie au LBS. Une fois que le dispositif final reçoit les résultats souhaités à partir du service géodépendant, il les transmet au dispositif source adjacent. Pour des raisons de confidentialité, chaque dispositif dans le processus de délégation ne conserve que les traces de la paire de dispositifs adjacente, et ne peut, en aucun cas, identifier d'autres dispositifs impliqués dans le processus.

Les résultats reviennent en empruntant le même chemin jusqu'au dispositif initial au chemin routage, qui à son tour les retourne au demandeur. La raison derrière le choix du même chemin réside dans le temps nécessaire afin de créer un autre chemin pour retourner les résultats.

En ce qui concerne la vie privée des participants au processus de délégation, le réseau de la foule de délégation garantit l'indiscernabilité des requêtes lors d'un processus de délégation. En d'autres termes, la requête ne contient aucune propriété qui pourra

identifier le demandeur initial.

Les principaux défis auxquels nous sommes confrontés lors de la construction du chemin dans le réseau de la foule de délégation peuvent se résumer dans le temps et l'efficacité de la communication. Le premier décrit le délai pris lors de l'initiation d'une requête géodépendante et l'obtention d'une réponse, alors que le dernier concerne la capacité de transmettre efficacement les données dans le réseau sans distorsion ou échec. Le terme distorsion dans ce contexte fait référence à l'altération et la modification du contenu d'une requête géodépendantes (*par ex.* modifier les résultats pour forcer le demandeur à renvoyer une autre requête).

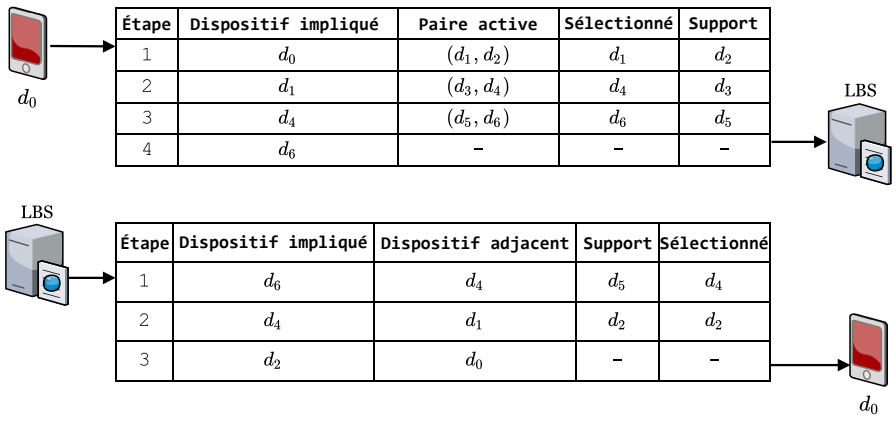
Dans le cas des requêtes en direction demandeur-à-LBS, une approche intuitive utilise les accusés de réception et les délais d'expiration. Lorsqu'un dispositif envoie la requête déléguée à un autre appareil au temps t_0, il attend un accusé de réception avec succès. S'il ne le reçoit pas à l'instant $t_0 + \delta$, où δ est le délai d'attente, il renvoie la requête au dispositif du support. Si ce dernier échoue aussi à la réception de la requête, le demandeur initial refait appel au répertoire inconscient pour récupérer une autre paire des dispositifs actifs.

L'idée consiste à éviter la congestion du réseau en veillant à ce que chaque demande soit livrée au prochain dispositif dans les plus brefs délais. La valeur de δ est définie dynamiquement lorsque le dispositif rejoint le réseau, et elle est égale au temps le plus long pris par un dispositif lors des échanges avec le répertoire inconscient. Autrement dit, lorsqu'un appareil enregistre ses informations au répertoire inconscient, il envoie le temps pris à la fin de l'opération. Le répertoire à son tour détient la valeur de temps la plus longue en tant que seuil de délai maximal δ. Une mesure préventive supplémentaire pourrait être de fixer le seuil de délai δ à deux fois la valeur de temps la plus longue.

Dans le cas des requêtes en direction de retour (LBS-à-demandeur), la congestion est plus susceptible de se produire lorsque l'un des dispositifs impliqués dans le chemin de routage se déconnecte du réseau avant l'achèvement du processus de délégation. Pour gérer ce problème, nous utilisons des paires de dispositifs de sorte qu'un seul dispositif

est utilisé pour le processus de délégation et l'autre agit comme un dispositif de support.

Par exemple, lorsque le dispositif demandeur d_r reçoit la paire de dispositifs (d_a, d_b) du répertoire inconscient, il sélectionne un dispositif pour gérer le processus de délégation et envoie l'identifiant du second (dispositif du support) avec la requête initiale. Ensuite, le dispositif sélectionné, d_a par exemple, choisit l'autre paire de dispositifs (d_c, d_d) et répète le même processus. Nous notons ici que les dispositifs de sauvegarde d_b et d_d gardent les traces du dispositif source et du dispositif de route correspondants (c'est-à-dire l'autre dispositif dans la paire). La figure 4.7 illustre les deux cas de la sélection des dispositifs demandeur-à-LBS et LBS-au-demandeur plus les informations contenues dans chaque dispositif.



Étape	Dispositif impliqué	Paire active	Sélectionné	Support
1	d_0	(d_1, d_2)	d_1	d_2
2	d_1	(d_3, d_4)	d_4	d_3
3	d_4	(d_5, d_6)	d_6	d_5
4	d_6	-	-	-

Étape	Dispositif impliqué	Dispositif adjacent	Support	Sélectionné
1	d_6	d_4	d_5	d_4
2	d_4	d_1	d_2	d_2
3	d_2	d_0	-	-

Figure 4.7 – Tableaux de délégation dans le routage bi-nœud

Le premier tableau de la figure 4.7 illustre un chemin de routage dans le cas des requêtes demandeur-à-LBS. Dans la première étape, le demandeur obtient la paire de dispositifs actifs(d_1, d_2), sélectionne d_1 comme le dispositif à recevoir la requête et définit d_2 comme dispositif du support. Chaque dispositif sélectionné répète l'opération jusqu'à l'étape finale (étape 4) lorsque le dispositif sélectionné envoie la requête.

Le deuxième tableau représente le cas de la communication LBS-au-demandeur. À l'étape 1, la requête est immédiatement transmise au dispositif source adjacent (dispositif source par rapport à la direction demandeur-à-LBS), elle est transmise au dispositif de support qui contient les mêmes informations que celui sélectionné, à l'étape 2. Lorsque le

dispositif de support est lui aussi indisponible, la requête est perdue, et *Deloc* redémarre le processus de délégation.

4.1.6 Cache géodépendant

Afin de minimiser les coûts de communication, *Deloc* utilise un mécanisme de cache pour exécuter les requêtes géodépendantes. Chaque dispositif possède une partition du cache, gérée par *Deloc* qui sert à stocker les résultats géodépendants. Les principaux avantages de l'utilisation du cache résident dans le fait de réduire les coûts de communication et diminuer la latence au sein du réseau en minimisant le nombre des requêtes envoyées au LBS et le nombre d'opérations de délégation. L'idée est de réutiliser les résultats des requêtes similaires (*par ex.* deux utilisateurs demandant des adresses de restaurants) récupérés à partir des autres dispositifs.

Le défi majeur auquel nous sommes confrontés lors de la définition du cache géodépendant est la nature de la localisation elle-même. La localisation est exprimée par des coordonnées géographiques et la probabilité que deux utilisateurs demandent les mêmes informations à un LBS à partir des mêmes coordonnées est considérablement faible. Ainsi, chaque fois qu'une requête de délégation est reçue, *Deloc* vérifie si les coordonnées reçues sont à proximité de l'une des coordonnées en cache. S'il trouve une correspondance, il renvoie les résultats, sinon il continue le processus de délégation. Les positions proches sont définies selon deux contraintes principales : la précision et le temps.

La précision est gérée par le demandeur initial, et la région dans laquelle les positions sont censées être à proximité est définie en fonction de celle-ci. Plus précisément, si le demandeur impose un seuil de précision de 50 mètres, toutes les coordonnées à moins de 50 mètres de l'une des positions mises en cache sont censées être géographiquement proches, et leurs résultats respectifs sont des résultats potentiels. Ces derniers ne sont pas retournés au demandeur avant de vérifier l'autre contrainte, qui est le temps associé à la requête.

Chaque résultat de cache a un temps de requête exact, et la comparaison du temps de cette dernière, avec celui de la requête déléguée courante améliore l'exactitude des résultats renvoyés. Par exemple, si l'utilisateur recherche des stations-service proches de sa localisation à minuit, alors que deux des plus proches sont fermées, le retour des résultats du cache qui ont été récupérés à 10h du matin l'induit en erreur. Par conséquent, les résultats cachés près d'un utilisateur sont filtrés en fonction de leur temps d'émission avant d'être retransmis au demandeur courant.

L'utilisation du cache dans notre contexte aide à réduire les coûts de communication associés à chaque processus de délégation. De plus, il renforce notre objectif de réalisation d'un mécanisme collaboratif. Cependant, l'objectif de *Deloc* n'est pas l'utilisation du cache comme un seul moyen de protection tel qu'il a été proposé dans les travaux de Yang *et al.* ou de Ma *et al.* [79, 127], mais plutôt l'utilisation du cache pour optimiser le fonctionnement global de *Deloc*.

L'ensemble des composants décrits dans cette partie constitue les bases du processus de délégation dans *Deloc*. Ils interagissent selon l'architecture définie précédemment dans la figure 4.1 afin d'assurer le bon déroulement du processus de délégation. Ce processus de délégation est nouveau et basé sur une nouvelle architecture. Il ne peut se compléter qu'avec l'interaction de ses composants.

4.2 Tests et simulation

Après les définitions théoriques et applicatives des composants liés au processus de délégation, nous discutons dans la suite les différents tests élaborés afin de valider le concept de la délégation et son efficacité concernant la protection de la vie privée et l'assurance de l'utilité. Les tests ont été effectués en deux étapes : **la première** concerne la mise en œuvre de l'environnement de simulation, et **la deuxième** se rapporte à l'analyse et l'évaluation des résultats issus de la simulation.

4.2.1 Environnement de simulation

Nous avons implémenté le mécanisme de délégation proposé en Java, et nous avons mesuré la performance du mécanisme sur des terminaux stationnaires et mobiles. En outre, nous avons implémenté un serveur LBS en PHP, basé sur les services de géolocalisation de Google *(Google Places API)*, qui collecte et stocke localement les requêtes des utilisateurs avant de renvoyer les résultats reçus de l'API. Les requêtes stockées sont utilisées pour évaluer le niveau de confidentialité garantie par *Deloc*.

De même, nous avons implémenté le serveur du répertoire inconscient, pour fournir les données de dispositifs actifs. Les deux serveurs ont été exécutés sur une machine équipée d'un processeur Intel Core i7-3960X 3.30Ghz et 32 Go de mémoire vive. Nous avons également utilisé un ensemble de données de BrightKite [20] qui contient 58228 nœuds avec 4491143 ($\approx 4,5M$) coordonnées géographiques. Chaque tuple est composé de l'identifiant de l'utilisateur, sa position et de l'horodatage respectif.

Comme notre plateforme fonctionne dans le cas de requêtes en temps réel, nous avons mis en place un programme de simulation qui imite le comportement des utilisateurs. Nous avons d'abord préparé le nouveau jeu de données en limitant l'ensemble de données original aux nœuds ayant plus de 70 tuples, de façon à ce que notre nouvel ensemble de données B soit composé de 9963 nœuds avec 3956113 ($\approx 4M$) coordonnées géographiques.

Étant donné que l'ensemble de données utilisé contient des données réelles des utilisateurs, les coordonnées partagées ne sont pas uniques et peuvent être dupliquées ; un utilisateur peut émettre une requête plusieurs fois à partir de la même position. Par ailleurs, et pour imiter le contenu d'un dispositif réel, nous avons généré un répertoire pour chaque nœud à partir d'un ensemble aléatoire de données.

Le répertoire contient l'ensemble des coordonnées géographiques respectives échantillonnées à partir de l'ensemble B, une liste de contacts générée contenant 50 à 200 entrées aléatoires, et les données du propriétaire du dispositif généré (*par ex.* nom, cour-

rier électronique, identifiant unique). Les données contenues dans le dossier sont utilisées pour calculer une empreinte unique qui reflète le cas des applications actuelles où l'empreinte digitale est utilisée pour identifier un dispositif de manière unique.

Dans notre cas, les informations utilisées pour les empreintes digitales comprennent le nom complet du propriétaire, le courrier électronique (c'est-à-dire compte du propriétaire), le numéro de téléphone, l'adresse MAC de la puce WiFi et la liste de contacts. À l'exception de la liste de contacts, toutes les informations ont été stockées dans un seul fichier JSON dans le répertoire généré. Le listing 4.1 illustre un exemple d'un fichier JSON contenant des données générées aléatoirement pour imiter leur équivalent dans un dispositif réel. Il est important de mentionner ici que les applications du monde réel peuvent recueillir des données plus précises sur un utilisateur (*par ex.* date d'anniversaire, contenu multimédia, image de profil), notre sélection est basée sur des informations minimales qui peuvent fournir, une fois combinées, une empreinte numérique unique pour chaque utilisateur.

Le programme de simulation exécute pour chaque répertoire généré une unité d'exécution (*Thread*) indépendante qui l'enregistre dans le serveur de répertoire inconscient en tant qu'un dispositif. Chaque unité d'exécution exécute un ensemble de requêtes géo-dépendantes (1 à 20 requêtes) séparées par des périodes de temps aléatoires (1 à 15 minutes). À chaque requête, l'unité d'exécution recherche des POI autour des coordonnées échantillonnées à partir des coordonnées de l'ensemble B.

Listing 4.1 – Exemple des données générées pour simuler un dispositif

```
{
 "uid": "5620fbfd1f3c",
 "sex": "M",
 "phone": "(758)163-7968",
 "mac": "99:ef:27:e3:55:06",
 "owner": "Joshua Hurst",
 "email": "ronald52@example.org"
}
```

L'unité d'exécution qui émet la requête la transmet à d'autres unités en récupérant la liste de celles disponibles depuis le répertoire inconscient, jusqu'à ce que l'unité d'exécution finale la reçoive et interroge le serveur. Ainsi, la requête reçue par ce dernier contient l'identifiant unique de l'unité d'exécution finale, son empreinte digitale et les coordonnées géographiques et la requête géodépendante de l'émetteur. En conséquence, la procédure de simulation a permis de fournir 51783 requêtes géodépendantes représentant la connaissance du LBS concernant le trafic du réseau de la foule de délégation.

Notons ici que la sélection des coordonnées géographiques à partir de l'ensemble B dépend essentiellement de l'horodatage extrait du fichier original. À chaque fois qu'une unité d'exécution sélectionne une nouvelle position, elle sélectionne celui avec l'horodatage le plus proche de la demande précédente. Cette sélection permet une simulation plus réaliste où les positions dépendent du temps. De même, après la sélection, la position jumelle unique est calculée pour être utilisée à la place des coordonnées originales.

Nous avons conduit une simulation supplémentaire qui vise à valider les facteurs d'efficience en utilisant l'émulateur des appareils Android *Nox App Player*[2] . Chaque dispositif émulé a été configuré pour fonctionner sous Android KitKat 4.4.2, avec un processeur central et 512 Mo de RAM. À l'aide des mêmes serveurs du répertoire inconscient et LBS, les 24 dispositifs émulés traitent des requêtes basées sur les délégations à l'aide de coordonnées échantillonnées à partir de l'ensemble de données B. Chaque dispositif émulé stocke le temps pris par une requête pour se compléter, ainsi que la taille des requêtes et des résultats géodépendants.

4.2.2 Résultats des tests

Nous discutons dans cette section les résultats de l'évaluation issus de la simulation. Comme nous l'avons mentionné auparavant, il est vrai que les tests de validation sont effectués en simulant le comportement des dispositifs, mais les données utilisées sont des données issues du monde réel. De ce fait, le comportement observé n'est pas loin du

[2]https://www.bignox.com/

comportent au sein d'un réseau des dispositifs réels.

Les évaluations faites à cette étape sont empiriques et servent à attester l'efficacité de *Deloc*. Une évaluation formelle est discutée dans le chapitre 6 où nous utilisons une métrique de confidentialité adaptée pour les mécanismes collaboratifs.

4.2.2.1 Évaluation de la confidentialité

Tel que mentionné auparavant, l'évaluation de la confidentialité à ce stade fait partie de l'évaluation empirique du processus de délégation. Elle sert essentiellement à discuter la conformité du mécanisme de délégation vis-à-vis les exigences de protection discutées dans la section 3.3.2. Ainsi, une évaluation formelle de la confidentialité que *Deloc* peut assurer est discutée en détail dans le chapitre 6.

Dans notre évaluation, nous supposons que l'information détenue par un LBS représente une violation de la confidentialité pour un utilisateur et nous prouvons l'impraticabilité de cette hypothèse. Nous considérons qu'un LBS pourrait être un adversaire, et ses serveurs pourraient être attaqués par des adversaires si ce n'est pas l'un d'entre eux. De même, nous supposons que la seule partie de confiance est l'utilisateur lui-même. En d'autres termes, chaque composant pourrait être compromis, y compris le réseau de foule de délégation et le répertoire inconscient.

Par conséquent, nous allons prendre en considération deux cas principaux sur lesquelles nous allons évaluer la confidentialité du mécanisme de délégation ; le premier représente la confidentialité au cours du processus de délégation et le second lorsque la requête atteint le LBS. Pour chaque cas, nous évaluons si l'une des requêtes géodépendantes peut être liée à son émetteur original et la similitude entre les demandes originales et les demandes déléguées.

Nous supposons pour le reste de cette section que u_i, un utilisateur parmi l'ensemble des utilisateurs U d'un LBS, exécute un ensemble L_i des requêtes géodépendantes l_i pour récupérer les résultats géodépendants. Chaque requête $l_i \in L_i$ est exécutée à un instant $t(l_i)$ tel que $t(l_i) \neq t(l_j)$ pour $l_i, l_j \in L$. Les requêtes résultant du processus de

délégation sont représentées par l'ensemble D_i contenant les requêtes déléguées d_i^x pour chaque $l_i \in L_i$, où x indique le nombre d'opérations de délégation accomplies au moment de l'observation. Notons ici que nous utilisons x pour des raisons de clarté et de preuve seulement, et que cela n'existe pas dans l'implémentation finale de *Deloc*. Les requêtes dans *Li* et *Di* sont composées des coordonnées de localisation loc_i, du profil de demandeur p_i (*par ex.* identifiant, ID de l'appareil, empreinte digitale) et le contenu de la requête c_i.

Comme décrit précédemment (section 4.1.2), avant de lancer le processus de délégation, le mécanisme de délégation commence par calculer la position jumelle de l'utilisateur. Les positions jumelles sont la première couche de confidentialité dans *Deloc*, elles permettent d'éviter différentes attaques, comme celle du mouvement maximal par exemple.

Intuitivement, pour construire suffisamment de connaissances sur une position afin d'effectuer de telles attaques, un adversaire a besoin d'un certain nombre de positions jumelles égales au nombre de positions délivrées dans les mécanismes de génération aléatoire.

De même, les positions jumelles dépendent d'attributs uniques, et le calcul d'une nouvelle position jumelle unique nécessite un nouvel ensemble unique, une opération pratiquement impossible à atteindre. Par exemple, si un adversaire a besoin de 100 positions pour exécuter des attaques du mouvement maximal, un nombre équivalent de combinaisons d'attributs de calcul uniques (*par ex.* identifiant LBS, identifiant d'utilisateur, coordonnées géographiques) est nécessaire pour construire suffisamment de connaissances.

Ainsi, nous supposons qu'une position jumelle a pour objet de fournir une protection de la vie privée contre ce genre d'attaques, et nous nous concentrons sur le processus de délégation. Le but de l'adversaire est d'identifier la véritable position de l'utilisateur u_i. Ainsi, compte tenu de la demande déléguée d_i, l'adversaire essaye de déduire la requête originale l_i et augmente la certitude du profil du titulaire contenu dans ce dernier en étant

le profil de l'émetteur. Pour atteindre cet objectif, l'adversaire calcule la probabilité suivante, en s'appuyant sur D_i l'ensemble contenant les requêtes déléguées de l'utilisateur u_i.

$$Pr\{p_i = u_i | d_i\}$$

Dans un scénario idéal, l'adversaire extrait la requête déléguée d_i et tente d'identifier tout lien potentiel avec les requêtes précédentes dans D_i. Cependant, l'adversaire ne peut pas connaître l'émetteur de la demande initiale et doit trouver les liens potentiels de chaque nouvelle demande avec toutes les demandes déléguées enregistrées. Nous explorons les deux cas où un adversaire peut vouloir compromettre *Deloc*.

4.2.2.1.1 Requêtes au sein du réseau

Pour évaluer une violation potentielle de la confidentialité pendant le processus de délégation, nous évaluons la quantité de données d'identification qu'un adversaire peut déduire soit en faisant partie du processus de délégation lui-même, soit en attaquant le répertoire inconscient. Nous définissons $S = \{s_i\}_{s \in [1,n]}$ comme l'ensemble des dispositifs participant au processus de délégation en cours. Nous supposons qu'une attaque permet l'observation de S, qui comprend l'adresse et le nombre d'utilisations de chaque $s_i \in S$. Ainsi, un attaquant peut faire partie de S lui-même ou accéder au répertoire inconscient.

Intuitivement, si l'attaquant est le premier élément de S, il ne peut évaluer si la source adjacente est l'émetteur de la demande puisque la demande ne contient aucune information spécifique sur l'émetteur. De même, si l'attaquant est le dispositif final ou un autre dispositif dans S, les chances d'identifier le demandeur sont encore plus petites. D'autre part, si l'attaquant compromet le répertoire inconscient, il ne peut pas former de connaissances sur le demandeur puisque le répertoire lui-même ne contient que des adresses de dispositifs.

4.2.2.1.2 Requêtes hors réseau

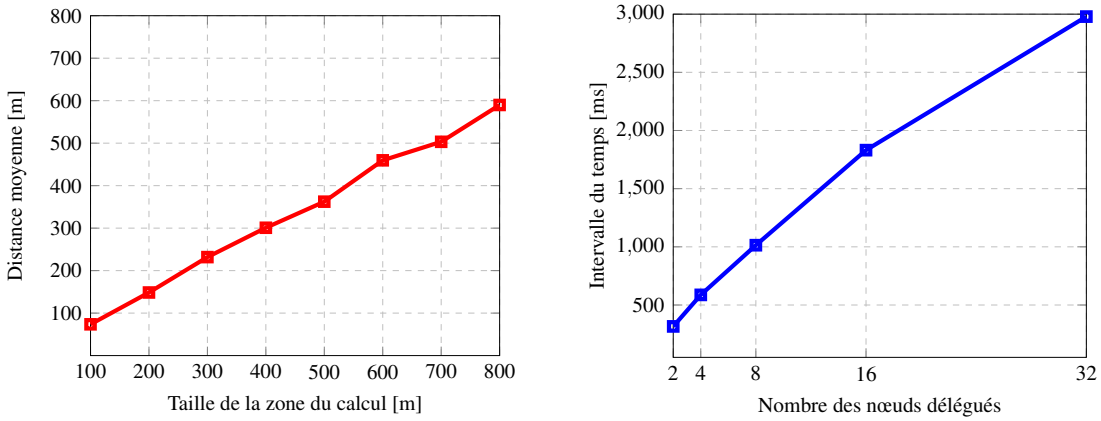
Le service géodépendant stocke chaque demande et tente d'inférer la véritable position d'un utilisateur en évaluant pour chaque utilisateur existant la probabilité de la requête actuelle contenant sa position réelle. En d'autres termes, lorsque le LBS reçoit une demande basée sur la position, il évalue pour chaque utilisateur stocké la probabilité que la demande actuelle soit liée ou représente la position réelle de l'utilisateur considéré.

Cependant, même si le même utilisateur émet des requêtes au même service géodépendant plusieurs fois, la demande reçue est émise à partir d'un dispositif différent à chaque fois. De plus, même lorsque le LBS évalue une proximité potentielle entre deux emplacements, le fait qu'il le reçoit de différents dispositifs finaux ne lui permet pas d'identifier un lien entre l'émetteur et les coordonnées de la position. Ainsi, un adversaire ne sera pas en mesure d'identifier le demandeur même en étant l'utilisateur commun entre plusieurs processus issus du même demandeur.

4.2.2.2 Évaluation empirique de l'utilité

Une approche intuitive pour mesurer le niveau d'utilité assurée lors de l'utilisation du mécanisme de délégation consiste à explorer les résultats des voisins les plus proches (k-NN) et la distance physique entre la position divulguée et la position réelle. Dans le cas de *Deloc*, la position divulguée est soit la position réelle, soit la position jumelle. Dans le **premier cas**, k-NN pour une position divulguée est identique à la position originale, dans **le second cas**, les positions jumelles sont des positions exclusives qui dépendent de plusieurs paramètres autres que les coordonnées géographiques. En d'autres termes, pour la même position, avec les mêmes paramètres de calcul, la position jumelle ne change pas et l'utilisation de k-NN ne peut pas aider à mesurer l'utilité. Par conséquent, nous nous concentrons sur la mesure des distances physiques et des intervalles de temps entre les positions réelles et divulguées des utilisateurs.

Nous évaluons l'utilité assurée par *Deloc* dans le cas de l'utilisation d'une position jumelle avant de lancer le processus de délégation. Pour ce faire, nous mesurons la distance moyenne selon la zone de confidentialité prédéfinie et la différence de temps moyenne selon le nombre de nœuds de délégation. La figure 4.8 illustre les deux mesures.



(a) (b)

Figure 4.8 – Distance moyenne et intervalle du temps moyen dans le mécanisme de délégation

Comme on peut le voir sur la figure 4.8, la distance physique (4.8a), délimitée par la zone de calcul, peut fournir une utilité similaire à celle des mécanismes utilisant la transformation de substitution. Un fois de plus l'avantage de *Deloc* est son efficacité, même sans utiliser de positions jumelles. Une telle distance est en fait tolérable, et même lors de son utilisation, les LBS peuvent fournir des résultats précis. Cependant, certains (*par ex.* les services d'urgence) nécessitent une grande précision, c'est dans ce cas que *Deloc* peut ignorer l'utilisation de positions jumelles pour permettre une précision optimale tout en conservant les mêmes garanties de confidentialité. L'intervalle de temps à son tour ne présente pas de problème de précision (4.8b), l'intervalle moyen maximum entre une requête originale et une déléguée est de 3 sec. lorsque la requête déléguée passe par 32 appareils. Pour conclure, l'exactitude de la position divulguée par *Deloc* dépend de l'utilisation d'une position jumelle. Cette dernière est une couche de confidentialité

supplémentaire qui peut être omise sans détériorer la confidentialité assurée. En fait, l'utilisation de positions jumelles empêche certaines attaques classiques (*par ex.* attaque de l'homme du milieu), et restreint la déduction de la position réelle de l'utilisateur en recueillant des informations contextuelles liées à sa position (*par ex.* ID de la cellule, SSID WiFi).

Une évaluation complémentaire concernant l'utilité considère les résultats obtenus en utilisant *Deloc*. Pour évaluer les résultats, nous reprenons les réponses retournées avec et sans l'utilisation de *Deloc*, et nous comparons leur similarité. Pour des ensembles de résultats A et B retournés dans le cas de la non-utilisation et de l'utilisation de *Deloc* respectivement, le nombre des résultats différent d est égale à $|(A \cup B) - (A \cap B)|$. Le tableau 4.I liste le nombre des résultats différents quand *Deloc* est utilisé, et cela en fonction de la zone du calcul de la position jumelle.

Tableau 4.I – Nombre des résultats différents en fonction de la zone de calcul

Taille de la zone de calcul (m)	100	200	300	400	500	600	700	800
Nombre des résultats différents	5	9	10	11	14	13	12	14

Tel que constaté à partir du tableau 4.I, l'utilisation de la position jumelle au sein de *Deloc* n'implique pas une différence remarquable dans les résultats obtenus par rapport à l'utilisation de la position réelle. Notons que la requête initiale est de trouver un restaurant dans le centre ville de Montréal. Nous remarquons aussi que la différence des résultats est influencée par la taille de la zone de calcul de la position jumelle.

L'autre point de l'évaluation empirique de *Deloc* est celui de l'efficience[3] en termes de *temps de traitement* et de *taille de communication*. Nous discutons dans la suite l'évaluation de l'efficience de *Deloc* dans le contexte des LBS actuels.

[3]Contrairement à l'efficacité qui décrit le rapport entre les résultats obtenus et les résultats attendus, l'efficience réfère à celui entre les résultats obtenus et les moyens mis en œuvre pour y parvenir. *[Source : Office québécois de la langue française - le grand dictionnaire terminologique]*

4.2.2.3 Évaluation empirique de l'efficience

Nous analysons l'efficience du mécanisme de délégation dans deux cas : le cas d'utilisation et de non-utilisation du transfert inconscient au moment de la récupération de la liste des dispositifs actifs. Nous voulons démontrer par une telle comparaison que les coûts généraux associés à l'utilisation du mécanisme de délégation sont acceptables même en utilisant le transfert inconscient. Nous mesurons quatre indicateurs d'efficacité : le *temps du traitement de la requête*, la *taille des données communiquées*, le *temps du traitement du répertoire inconscient*, et le *temps du traitement du serveur LBS*. La figure 4.9 illustre chaque résultat.

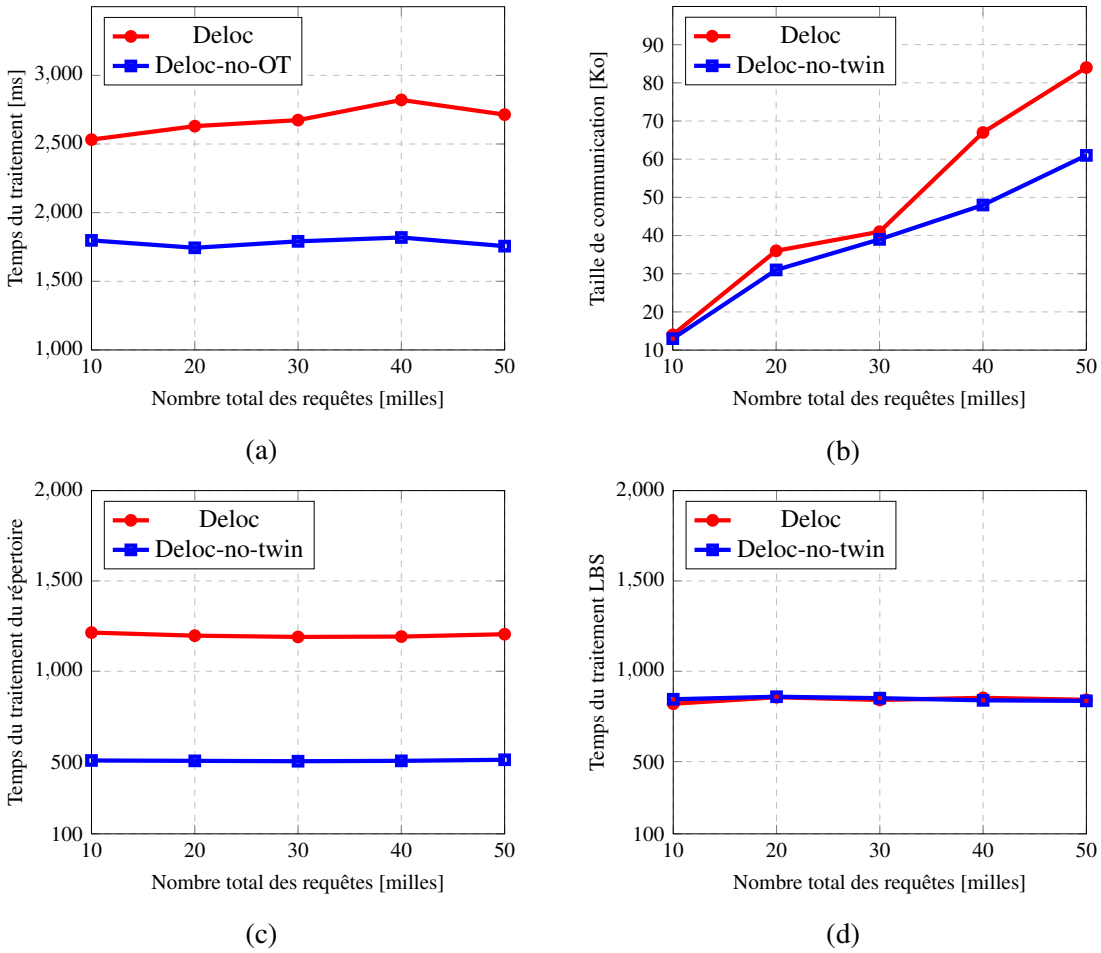


Figure 4.9 – Indicateurs d'efficience du mécanisme de délégation

Tel qu'illustré à la figure 4.9, les indicateurs d'efficience du mécanisme de délégation suggèrent une faible surcharge pour tous ses composants. Les mesures ne tiennent pas compte du nombre de nœuds dans le processus de délégation, un nombre attribué de manière aléatoire à une valeur entre 2 et 32 au début de chaque processus de délégation. Le temps moyen du traitement (4.9a) est affecté par l'utilisation d'un transfert inconscient dans le répertoire inconscient, qui rajoute un temps moyen de traitement d'environ une seconde.

La taille moyenne des données transmises ne représente pas une surcharge notable (4.9b). Nous mesurons pour chaque demande la taille des données échangées à partir du moment de l'émission d'une requête jusqu'à son retour à l'émetteur. La valeur en (4.9b) représente la taille moyenne de données échangées entre l'émetteur et le LBS du moment de l'envoi de la requête jusqu'à la réception des résultats, et cela en fonction du nombre total des requêtes. Par exemple, la taille totale de données échangées pour quarante mille requêtes est de 67 Ko. Ainsi, nous constatons que la taille de communication moyenne pour un utilisateur par requête est d'environ 1,4 Ko ce qui ne représente pas une surcharge.

Le temps de traitement dans le répertoire inconscient (4.9c) est affecté par le transfert inconscient uniquement, en moyenne, un transfert inconscient réussi prend 1200ms à compléter. Le temps de traitement (4.9d) à son tour n'est affecté par aucun des composants de *Deloc*.

Les valeurs de mesure de l'efficience prouvent que l'utilisation de *Deloc* impose des coûts faibles, et démontrent la praticabilité du mécanisme de délégation dans la plupart des LBS actuels. L'utilisation de *Deloc* est transparente pour l'utilisateur final et n'ajoute aucun temps ou coût de traitement significatif en comparant à une exécution sans *Deloc*.

4.3 Conclusion

Le mécanisme de délégation est un nouveau concept qui représente un composant important de *Deloc*, et qui sert à assurer la protection des utilisateurs de façon collaborative. Nous avons proposé dans ce chapitre l'architecture de ce mécanisme, ainsi que la mise en œuvre des différents composants qui le constituent.

Le processus de délégation dépend principalement de deux composants, le répertoire inconscient, et le réseau de la foule de délégation. Le premier assure que les utilisateurs de *Deloc* peuvent accéder aux informations sur les nœuds disponibles en tout temps, et le second est le réseau superposé où les processus de délégation sont exécutés. En plus de ces deux composants, *Deloc* est également composé d'autres composants et mécanismes, tels que les positions jumelles et les caches géodépendants.

Nous avons utilisé une évaluation empirique basée sur la simulation et l'échantillonnage à partir de données réelles, et nous avons constaté que le mécanisme de délégation n'ajoute pas de coûts significatifs au flux de données existant. En outre, notre prototype fonctionne efficacement sur des appareils mobiles émulés avec des caractéristiques minimales (processeur monocœur et 512 Mo de mémoire vive).

Dans notre supposition de base, le mécanisme de délégation peut être évalué en utilisant n'importe quelle métrique de confidentialité existante (*par ex. k*-anonymity, modèles probabilistes, etc.). Dans notre cas, nous proposons un modèle probabiliste adapté aux mécanismes collaboratifs, où la connaissance qu'un adversaire peut avoir sur la cible réside dans l'information échangée par les membres réalisant le mécanisme. Autrement dit, les métriques actuelles peuvent affirmer l'efficacité de *Deloc*, mais ne représentent pas la relation entre les membres du mécanisme collaboratif. Pour cette fin, nous proposons δ*-fuzziness* (chapitre 6), une métrique qui vise à évaluer les mécanismes collaboratifs, en incluant notre mécanisme basé sur la délégation.

Deloc est capable de fournir des garanties rigoureuses en termes de protection de la vie privée dans les LBS. Cependant, le fait qu'il soit basé sur la collaboration de plu-

sieurs utilisateurs nécessite des mesures additionnelles. Autrement dit, *Deloc* dépend du réseau de foule de délégation, ce qui implique des risques potentiels engendrés par les membres du réseau (collaborateurs *curieux* et *malicieux*), et éventuellement des violations possibles de ses exigences de base [16]. Pour contrer cela, nous proposons dans le chapitre suivant deux modèles de mesure et quantification qui aident à évaluer les membres du réseau de délégation, et à estimer leur confiance ainsi que les risques qu'ils peuvent engendrer.

CHAPITRE 5

ESTIMATION DES RISQUES ET CONFIANCE

L'estimation de risques et l'évaluation du niveau de confiance assurée dans un environnement donné sont deux fonctions nécessaires pour l'efficacité d'un processus de préservation de la vie privée. Pendant que le niveau de confiance décrit à quel point un environnement est fiable, l'estimation de risques est la fonction faisant partie de la perception globale de confiance et servant à estimer l'effet de divulgation de données. Les deux définitions suivantes expliquent et résument les concepts de risques et de confiance dans notre contexte.

Définition 5.1 (Risque). *Le risque représente l'éventualité d'une divulgation causée par l'inclusion de données sensibles dans un processus de délégation.*

Définition 5.2 (Confiance). *La confiance dans l'environnement de délégation représente l'assurance de déroulement des processus au sein de la plateforme sans divulgation causée par un collaborateur curieux ou malicieux.*

Dans le contexte de cette recherche, le partage préservant la confidentialité *(privacy-preserving sharing)* est un des concepts ayant les objectifs les plus similaires aux objectifs d'un LPPM. Le principe de partage préservant la confidentialité est simple, un système qui donne accès aux données de ses utilisateurs doit assurer un maximum d'opacité tout en gardant l'utilité de l'information. Certains des travaux qui abordent ce sujet traitent le problème du point de vue de la gestion de la confidentialité, soit en revenant toujours aux mécanismes de protection de confidentialité adoptés par l'utilisateur et la politique de confidentialité du système [64, 67], soit en évaluant la perte des données privées et son impact sur la vie privée de l'utilisateur [9, 10, 69]. Les données sont jugées sensibles selon les paramètres de confidentialité adoptés par l'utilisateur.

Quant à la quantification de risques, un des premiers travaux dans le contexte de vie

privée remonte à 2010, dans lequel les auteurs présentent une approche pour quantifier le risque d'un profil en fonction des paramètres de confidentialité [75]. Un autre travail présente une plateforme qui regroupe deux mécanismes [57]. Le premier se charge d'évaluer la portée des données d'un utilisateur en analysant son profil. Le deuxième s'occupe de risques qu'un contact peut occasionner en analysant le contenu textuel de son profil et en essayant de définir les principaux traits de sa personnalité. Les auteurs de ce travail utilisent une analyse textuelle sur les données partagées sur Facebook.

Dans le contexte de cette recherche, nous définissons deux modèles d'évaluation : à savoir de risques et de confiance. Nous évaluons l'impact d'une requête sur la détermination de la position réelle d'un utilisateur, et cela en mesurant les risques et la confiance liés à l'utilisation et à la divulgation au sein d'un LBS donné. Même en utilisant un mécanisme collaboratif efficace, la présence d'un collaborateur malicieux dans le processus peut mener à un bris de la vie privée de l'utilisateur en question.

En reprenant l'architecture de *Deloc* (figure 3.2), notons que la quantification des risques et la mesure de la confiance constituent une partie indépendante dans *Deloc*, avec elle, interagissent les composants du processus de délégation d'un côté (chapitre 4) et les composants de la métrique de confidentialité d'un autre côté (chapitre 6). Nous illustrons ainsi la partie de la plateforme qui gère la quantification de risques et la mesure de confiance dans la figure 5.1.

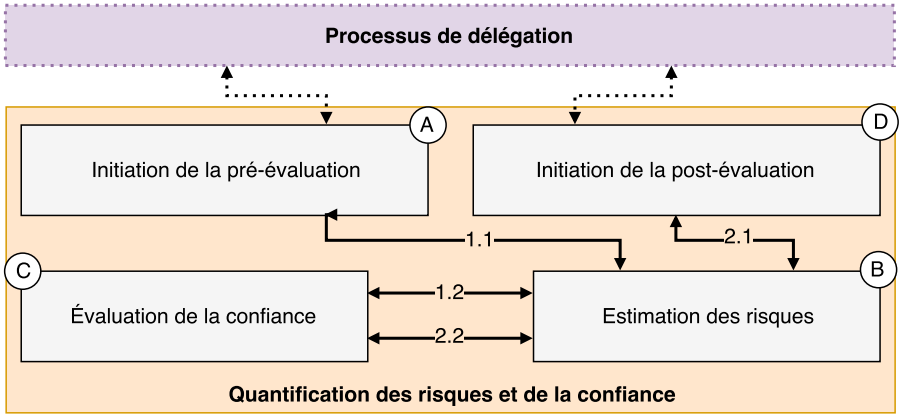


Figure 5.1 – Architecture des composants de risques et confiance

Comme illustré dans la figure 5.1, la première invocation de la procédure d'estimation de risques et confiance initiée dans (A), se fait durant le processus de délégation, cette estimation inclue la quantification de risques dans l'étape (B) ainsi que l'évaluation de confiance dans l'étape (C). La deuxième invocation, initiée dans l'étape (D), se fait après la récupération des résultats géodépendants, et sert à ré-estimer les risques et la confiance après l'opération de délégation.

En nous basant sur l'architecture définie ci-dessus, nous discutons dans la suite de ce chapitre des deux modèles de quantification de risques et mesure de confiance. Les deux modèles sont basés sur *la théorie de l'information* mutuelle et sur *les chaînes de Markov* respectivement. Nous abordons également leur applicabilité au contexte des LBS, ainsi que leurs fondements théoriques et pratiques. Enfin, nous évaluons les deux modèles en nous basant sur l'environnement de simulation décrit dans le chapitre précédent (4.2) en discutant leur validation conceptuelle et empirique.

5.1 Quantification des risques

La perception de la vie privée change d'un individu à l'autre, ce qui implique la nécessité d'un modèle de mesure indépendant pour chacun. Cela revient, dans notre contexte, à estimer le risque occasionné par la divulgation de chaque information dans le risque de divulgation totale de l'ensemble de données, l'ensemble est exprimé par le profil d'utilisateur associé à sa requête géodépendante. Pour illustrer cela, supposons qu'un utilisateur ait un profil contenant les informations suivantes : {courriel : MarshallH-Jones@jourrapide.com, nom d'utilisateur : Puppere, Date de naissance : 30 mars 1988}. Nous cherchons alors à estimer le risque occasionné par la divulgation de chacune de ces données par rapport au risque de divulgation total du profil.

Afin de pouvoir arriver à une évaluation précise, nous adoptons deux notions connues dans le contexte du partage préservant la confidentialité, la première est celle de *l'information mutuelle* [105], et la deuxième de *l'information spécifique* [32]. Les deux notions

fonctionnent ensemble afin de vérifier si d'une part les données contenues dans un profil ne posent pas un risque, et d'autre part estimer le risque occasionné par la divulgation de chaque donnée indépendamment par rapport au risque total de divulgation.

La notion de l'information mutuelle est étroitement liée à celle de l'entropie d'une variable aléatoire, une notion fondamentale en théorie de l'information, qui définit la quantité d'information contenue dans une variable aléatoire. Par ailleurs, l'information mutuelle est également une fonction mathématique utilisée dans la théorie des probabilités et dans la théorie de l'information, elle sert à quantifier l'information moyenne ou le gain d'informations sur une variable aléatoire X en observant une autre variable aléatoire Y.

Autrement dit, l'information mutuelle de deux variables aléatoires X et Y est la mesure de la dépendance mutuelle entre ces deux variables. Plus précisément, elle sert à mesurer la quantité d'information obtenue à propos de la variable aléatoire X à travers l'autre variable aléatoire Y [26]. La quantité d'information est souvent exprimée en *bits*, ou en d'autres unités qui dépendent du contexte. Plus formellement, l'information mutuelle de deux variables aléatoires discrètes X et Y est définie par :

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{5.1}$$

Où $p(x,y)$ est la fonction de probabilité conjointe de X et Y, et les deux $p(x)$ et $p(y)$ représentent la fonction marginale de distribution de probabilité de X et Y respectivement.

Afin de mesurer le risque occasionné par un seul symbole (une donnée dans notre cas), nous utilisons des mesures basées sur *la décomposition de l'information*. C'est là où la notion de l'information spécifique entre en jeu. En résumé, l'information spécifique est le nom donné à la famille de mesures dépendantes de l'état qui convergent vers l'information mutuelle [15].

Afin de mieux comprendre les deux concepts, considérons deux variables aléatoires

X et Y avec des valeurs dans $\{x_1, x_2, \ldots, x_n\}$ et $\{y_1, y_2, \ldots, y_m\}$ respectivement, telle que $p_i = P[X = x_i]\ \forall i \in [1,n]$, avec p_i la probabilité d'obtenir une valeur x_i et $q_j = P[Y = y_j]$ $\forall j \in [1,m]$, avec q_j la probabilité d'obtenir une valeur y_j. Notons par p^*_{ij} la probabilité conjointe entre X et Y, et $p^+_{i|j}$ la probabilité conditionnelle X en sachant Y. L'information mutuelle peut être décomposée pour quantifier la contribution d'une seule information dans l'information moyenne comme suit [14] :

$$\begin{aligned} I(X;Y) = &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} p^*_{ij} \log\left(\frac{p^+_{i|j}}{p_i}\right) \end{aligned} \tag{5.2}$$

Cette équation mesure l'incertitude entre la connaissance à priori de X, définie par l'entropie $H(X)$, et la connaissance de X étant donné Y notée par $H(X|Y)$. La mesure de l'incertitude entre les deux variables permet d'évaluer l'information apportée par les données divulguées par rapport aux données déjà connues par l'adversaire.

Pour illustrer l'application de ce concept dans notre contexte, prenons l'exemple d'un utilisateur u_1 qui souhaite envoyer une requête géodépendante pour récupérer la liste des théâtres à proximité. Comme mentionné auparavant, la requête contient sa position actuelle l_1 et un ensemble de données P qui contient son profil. En suivant la technique de délégation, la requête reçue par le LBS contient la position l_1 et le profil de l'utilisateur final, dénoté u_{n+1}, tel que n est le nombre des opérations de délégation dans le processus en question. Nous cherchons alors à estimer la différence entre la connaissance du LBS sur l'origine de la requête, et sa connaissance en observant la requête précédente ou le flux sortant d'un réseau de délégation.

Autrement dit, il s'agit d'estimer l'incertitude entre le fait que les données divulguées soient celles de u_1 et le fait qu'elles appartiennent peut être à un autre utilisateur, u_{n+1} par exemple. En termes de variables aléatoires, le processus de délégation peut être

représenté par une fonction de X à Y, tel que $X = \{x_1, x_2, ..., x_n\}$ est la variable aléatoire qui représente les connaissances antérieures de l'adversaire, et $Y = \{y_1, y_2, ..., y_m\}$ qui représente les données divulguées. Le gain d'information qu'un adversaire peut avoir en observant une requête géodépendante peut donc être quantifié comme suit :

$$I(X;Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} p(X = x_i \cap Y = y_j) \log \left(\frac{p(X = x_i | Y = y_j)}{\sum_{i=1}^{n} p(X = x_i \cap Y = y_j)} \right) \tag{5.3}$$

Dans notre contexte, nous distinguons **deux cas principaux** : le premier est celui du processus de délégation avec les risques que le processus peut apporter à l'utilisateur, le deuxième cas est celui du risque occasionné par le LBS. Dans la suite, nous discutons les propriétés et les définitions formelles de la quantification des risques dans chacun des deux cas.

5.1.1 Risques liés au processus de délégation

Un utilisateur peut commencer le processus de délégation, ou faire partie du chemin du routage. Dans les deux cas, l'estimation des risques sert à évaluer l'impact de l'opération de délégation sur la divulgation des informations confidentielles.

Quand un utilisateur instaure un processus de délégation, il transmet sa requête géodépendante avec des informations qui servent à retracer le chemin du retour des résultats. Ces informations sont représentées par l'adresse réseau qui sert à établir une connexion avec le nœud considéré. Les risques que le demandeur initial peut avoir sont donc liés au premier nœud dans le chemin du routage, et peuvent se manifester quand ce premier nœud est malveillant.

Nous considérons un utilisateur participant au processus du routage comme étant malveillant s'il essaye de définir l'identité du demandeur initial. La nature probabiliste des mécanismes utilisés par un nœud malveillant (adversaire) implique l'impossibilité de connaître l'identité exacte du demandeur initial. En fait, ce qu'un adversaire cherche est de pouvoir extraire le maximum d'information à propos de l'identité du demandeur

en observant les informations divulguées [108].

Les risques dans ce cas peuvent se produire quand le demandeur initial utilise successivement le même nœud d'entrée dans le chemin du routage, ou quand des liens existent entre eux bien avant le processus de délégation (*par ex.* s'il fait partie de ces contacts dans un LBS).

Le deuxième cas des risques liés au processus de délégation est celui des utilisateurs faisant partie du processus. Autrement dit, ce cas traite les risques que les utilisateurs participant au processus de délégation, autre que le demandeur initial, peuvent avoir. Selon l'architecture de *Deloc*, les utilisateurs intermédiaires accèdent seulement aux informations réseaux des autres et à la requête géodépendante du demandeur initial. En d'autres termes, ils ne peuvent pas faire de liaison entre la requête déléguée et la source adjacente, car la requête contient l'information d'un autre utilisateur, qui est le demandeur initial.

Cette particularité assure que les utilisateurs membres du chemin du routage sont protégés contre toute divulgation d'information. Par conséquent, le seul cas que nous devons considérer est celui du demandeur initial.

Formellement, nous définissons le gain d'information qu'un adversaire faisant partie du processus de délégation peut avoir en observant une requête géodépendante en se basant sur l'équation 5.3. Cependant, la seule différence est que l'adversaire doit définir d'abord le rôle de l'utilisateur en question, ce dernier peut être un demandeur initial ou un collaborateur dans le processus de délégation. Le gain d'information est alors défini comme suit :

$$I(X;Y|Z) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l} \frac{p(X = x_i \cap Y = y_j)}{p(Z = z_k)} \log\left(\frac{p(X = x_i, Y = y_j|Z = z_k)}{\sum_{j=1}^{m} p(X = x_i \cap Y = y_j)}\right) \tag{5.4}$$

Telle que $Z = \{z_1, z_2, ..., z_l\}$ est la variable aléatoire représentant l'ensemble des membres participants au processus de délégation. Nous supposons qu'un adversaire peut accéder à la liste de tous les utilisateurs participants à un processus de délégation donné.

Cela représente le pire cas, et assure que la valeur retournée par 5.4 reflète la valeur la plus élevée du gain d'information possible.

5.1.2 Risques liés aux LBS

L'estimation des risques qu'un LBS peut occasionner à un utilisateur consiste à évaluer la possibilité de son identification en n'accédant qu'aux données résultantes du processus de délégation. Autrement dit, un LBS ne peut identifier les participants au processus, mais il a accès à l'historique des requêtes, ce dernier constitue l'ensemble de ses connaissances antérieures. De plus, un LBS possède les données de profil de ses utilisateurs, et en tant qu'adversaire, il peut les utiliser pour deviner l'identité réelle derrière une requête géodépendante.

Nous supposons dans ce cas que toute observation d'une requête ne diminue pas les connaissances antérieures d'un LBS, et que ce dernier peut à tout temps revenir à tout l'historique des requêtes à sa disposition. Par conséquent, afin d'estimer le risque associé à une requête géodépendante nous reprenons l'équation 5.3 telle quelle sans aucune modification.

Dans le contexte de *Deloc*, **la quantification des risques** est une opération qui se fait en deux étapes, la première est faite avant la décision de déléguer une requête à un participant donné, et la deuxième se fait après la récupération des résultats géodépendants, où les risques seront évalués et comparés avec les estimations initiales.

La quantification des risques seule est insuffisante pour garantir un bon rapport confidentialité/utilité dans *Deloc*. Par exemple, la violation de la vie privée d'un utilisateur peut se produire quand l'un des participants au processus de délégation est un adversaire (collaborateur malicieux) qui essaye d'identifier les autres participants. Ainsi, nous proposons des mécanismes d'estimation de confiance qui servent à l'évaluation globale de l'environnement d'utilisation. Dans notre cas, l'environnent contenant les utilisateurs, les adversaires et le LBS. Nous discutons dans la section suivante les bases théoriques et fondements du modèle de mesure de confiance au sein de *Deloc*.

5.2 Mesure de confiance

Plusieurs travaux basés sur le concept de la confiance (*Trust*) ont été proposés et sont plus ou moins efficaces. Par exemple, le système *PRISM* (PRIvacy Sensitive Messaging) fournit de la protection aux utilisateurs de messagerie Internet selon différents niveaux de communication [92]. Il donne à l'utilisateur la possibilité de choisir le niveau de communication pour chacun de ses contacts en facilitant la prise de décisions concernant la confidentialité des données [92]. Un autre système nommé PViz utilise la solidité de liens entre deux contacts afin de pouvoir proposer à l'utilisateur des politiques de confidentialité adaptées [81]. Cette dernière méthode manque de précision du fait que les liens sociaux entre deux contacts en ligne peuvent être induits en erreur afin de forcer une politique de confidentialité flexible [42].

D'autres systèmes ont été conçus pour faciliter la gestion de l'audience. Le système *Audience View Interface* qui permet à un utilisateur de visualiser comment son profil est perçu par un ami ou par un groupe d'amis, et d'ajuster par conséquent les paramètres de confidentialités de son profil, est l'un des premiers systèmes de gestion de l'audience [74]. Cette approche est disponible aujourd'hui dans la majorité des réseaux sociaux, et permet de faciliter la configuration de paramètres de confidentialité d'un profil utilisateur selon des niveaux prédéfinis. Cependant, elle reste plus ou moins efficace, car elle nécessite l'intervention explicite de l'utilisateur, et peut parfois mener à des décisions de divulgation indésirables [113]. Par exemple, une divulgation due à la sous-estimation de la valeur des données.

Dans cette recherche, nous nous inspirons d'un travail plus large présenté dans le contexte des réseaux ad hoc mobiles par Chang *et al.*, l'idée des auteurs était de pouvoir garantir une confidentialité et une sécurité optimale dans un réseau bâti par ses membres [18].

Pour atteindre notre objectif de mesurer la confiance, nous modélisons les dispositifs participants au processus de délégation ainsi que le LBS en nous basant sur la théorie

des graphes. Ainsi, le graphe $G = (V, E)$ se compose de l'ensemble G qui contient les nœuds dans le processus de délégation (dispositifs ou LBS), et E qui représente les liens entre ces nœuds. Chaque nœud possède deux propriétés principales : son type s'il est un participant au processus ou bien un LBS, et sa valeur de confiance mise à jour après chaque processus de délégation accompli. Notons ici que le demandeur initial dans un processus donné est considéré aussi comme un dispositif participant. La figure 5.2 illustre le graphe qui représente *Deloc* et ses entités.

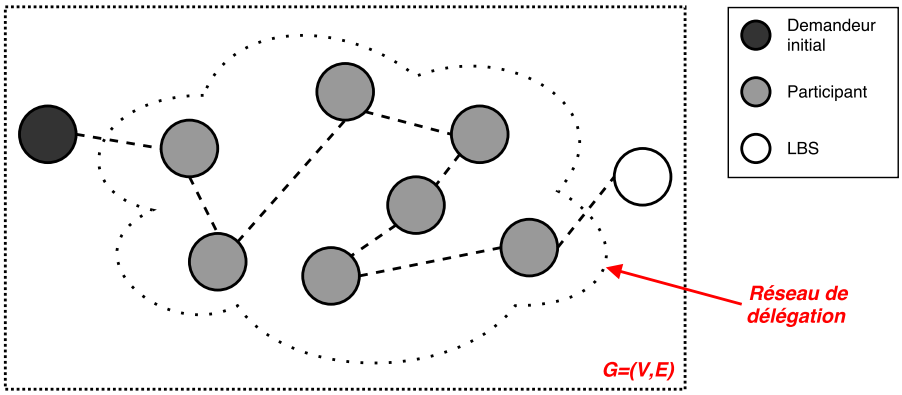


Figure 5.2 – Représentation du réseau de délégation sous forme de graphe

Nous définissons **la confiance** dans *Deloc* en étant un état qui passe d'un nœud à un autre nœud en fonction de la valeur de confiance actuelle. En d'autres termes, le nœud en possession de l'information (la requête géodépendante) possède la propriété d'être **digne de confiance** pour pouvoir l'acquérir et la garder, et il ne la transmet pas au prochain nœud sauf si ce dernier est lui aussi digne de confiance. La propriété d'être digne de confiance est évaluée par le niveau de confiance d'un nœud donné. Autrement dit, pour qu'un nœud soit digne de confiance, son niveau de confiance doit être supérieur à un seuil minimal prédéfini par la plateforme sinon il est considéré comme indigne de confiance, et par conséquent, non-sélectionné pour un processus de délégation.

La transmission de l'état de confiance ainsi que l'information se répète donc jusqu'à ce que l'information arrive au service géodépendant.

D'une façon formelle, nous considérons la transmission d'état de confiance entre les

nœuds comme étant un processus stochastique à temps discret, avec la particularité qu'un seul nœud possède l'état de confiance actuel à un instant donné t. La transition de l'état se fait uniquement d'un nœud à un autre nœud (absence des transitions en boucle). L'évaluation de confiance en utilisant modélisation est atteinte en se basant sur *les chaînes de Markov*. Dans la prochaine section sont présentées les principales propriétés des chaînes de Markov, et leur utilité, ainsi que la justification de leurs sélections dans le contexte de cette recherche.

5.2.1 Chaînes de Markov dans le contexte de la confiance

Une chaîne de Markov est un modèle stochastique décrivant une séquence d'événements possibles dans laquelle la probabilité de chaque événement dépend uniquement de l'état atteint dans l'événement précédent. Autrement dit, pour qu'un modèle stochastique possède la propriété d'être une chaîne de Markov, sa distribution de probabilité conditionnelle des états futurs doit dépendre uniquement de l'état présent, et non pas de la séquence des événements qui l'ont précédé. De façon plus formelle, pour qu'un processus soit une chaîne de Markov il doit respecter la condition suivante :

$$p(X_t = j | X_0 = i_0, X_1 = i_1, \cdots, X_{t-1} = i_{t-1}) = p(X_t = j | X_{t-1} = i_{t-1}) \tag{5.5}$$

Un exemple d'une chaîne de Markov peut s'illustrer dans le modèle de *marche aléatoire*. En résumé, le modèle décrit un système possédant une dynamique discrète composée d'une succession de pas aléatoires et totalement décorrélés. L'évolution du taux de change d'une certaine devise (*par ex.* taux du Dollar Canadien face à l'Euro) représente un modèle de marche aléatoire et peut être éventuellement modélisée par les chaînes de Markov.

L'utilisation des chaînes de Markov dans notre contexte s'explique par notre volonté d'assurer les trois points suivants :

- La dépendance de l'état actuel de confiance de l'état précédent seulement.

- L'indépendance des états de confiance par rapport au temps.
- La possibilité d'estimer le niveau de confiance à n'importe quel instant t_n.

Ainsi, une modélisation basée sur les chaînes de Markov assure une estimation de confiance basée sur l'état actuel du processus. Autrement dit, pour qu'un nœud puisse transmettre l'état de confiance, il doit le posséder d'abord, et cette possession assure qu'il est digne de confiance pour pouvoir choisir le prochain nœud. Par conséquent, nous nous limitons au nœud précédent seulement. À chaque étape, nous choisissons le nœud le plus sûr (avec le niveau de confiance la plus élevée). Cette propriété assure la protection pendant tout le processus de délégation et elle représente la raison principale de notre choix des chaînes de Markov pour la modélisation de confiance dans *Deloc*.

Une autre propriété des chaînes de Markov qui satisfait les exigences de notre modélisation est *l'égalité des chances* relatives au passage de confiance. Autrement dit, à un instant donné t_n la probabilité de choisir un nœud ne dépend pas de la valeur de t_n, et les possibilités d'être choisi sont similaires pour tout ensemble de nœuds possédant les mêmes valeurs de confiance. Nous détaillons le modèle de confiance proposé et le rôle des chaînes de Markov dans la section suivante.

5.2.2 Modèle d'estimation de confiance

Comme mentionné auparavant, chaque nœud dans la représentation du graphe possède deux propriétés principales, la première décrit son type, tandis que la deuxième représente son niveau de confiance. Cette dernière propriété est à la base du modèle de confiance, qui à son tour représente la base sur laquelle les chaînes de Markov sont utilisées pour estimer la confiance pendant un processus de délégation.

Le modèle d'estimation de confiance est composé de trois étapes principales : l'étape de *l'établissement de la relation de confiance*, l'étape de *définition des événements de confiance*, et celle de *l'attribution du niveau de confiance*. Chacune de ces étapes interagit avec les autres afin d'atteindre l'objectif d'attribuer un niveau de confiance à chaque

nœud.

Les étapes se déclenchent après l'occurrence de l'événement initial, représenté par la connexion d'un dispositif au réseau de délégation. La toute première étape réside dans l'établissement de la relation de confiance, cette étape est initiée suite à la connexion d'un dispositif au réseau de délégation et sert à faire une évaluation préliminaire de la confiance du nœud en question. La deuxième étape consiste à définir les événements responsables de l'évaluation de confiance selon les résultats de l'étape précédente.

La dernière étape est l'attribution du niveau de confiance. Ces étapes se répètent pour chaque nœud qui rejoint le réseau de délégation, et sont exécutées dans cet ordre à chaque fois qu'il se connecte. Autrement dit, le niveau de confiance attribué à un nœud est réinitialisé et évalué à chaque fois que ce nœud se connecte au réseau.

Nous discutons dans les sections suivantes chacune des étapes du modèle et les différentes interactions qui aident à obtenir une estimation précise. Notons que nous utilisons les deux termes nœud et dispositif de façon interchangeable pour décrire un dispositif mobile, le choix du terme est donc fait selon le contexte d'utilisation.

5.2.2.1 Établissement de la relation de confiance

Le niveau de confiance d'un nœud est affecté principalement par son comportement dans le réseau de délégation. De ce fait, un nœud avec "un bon comportement" tel que l'absence des déconnexions imprévues ou la possession d'une bande passante suffisante obtient un niveau de confiance élevé. Certaines caractéristiques du comportement d'un nœud, tel que le lien de connexion, sont attribuées au moment de sa connexion au réseau de délégation, ce qui permet l'établissement de la relation de confiance.

Un exemple justifiant notre approche basée sur le comportement pour la définition du modèle de confiance peut s'illustrer par les mécanismes de détection des intrus dans un système d'information. En raison de la difficulté du suivi et de la surveillance des adversaires, l'utilisation des modèles basés sur le comportement représente une approche efficace [106].

Dans notre cas, les événements définissant un comportement digne de confiance permettent l'élimination des utilisateurs suspects. Par exemple, s'assurer que le collaborateur est un utilisateur réel et non un robot en analysant sa façon de connexion au réseau. Par ailleurs, les événements utilisés permettent l'assurance de la sincérité de collaborateurs en partant de la supposition que l'objectif d'un collaborateur malicieux n'est pas la collaboration, assurer donc qu'un utilisateur vise cette dernière mitige les menaces de divulgation.

La façon dont une relation de confiance est établie dépend d'une liste de contrôle prédéfinie qui contient le strict minimum pour considérer un nœud digne de confiance ou fiable. Selon les exigences du mécanisme, les éléments qui constituent la liste de contrôle sont flexibles, et peuvent être adaptés selon le contexte. La vérification de la liste de contrôle permet d'établir la relation de confiance avec le nouveau nœud en lui donnant un niveau de confiance initial allant de 0 à 1.

Afin de bien expliquer l'étape de l'établissement de la confiance, prenons l'exemple d'un dispositif, nommé A, qui rejoint le réseau de délégation, représenté par le graphe $G = (V, E)$. Nous supposons dans cet exemple que la liste de contrôle CL illustrée par le tableau 5.I est celle utilisée au moment de l'établissement de la relation de confiance avec le nœud A.

Tableau 5.I – Exemple d'une liste de contrôle pour l'établissement de la confiance

Élément	**Méthode d'estimation**
Débit de la bande passante	Mesurer la vitesse de transmission d'un ensemble des requêtes
Niveau d'énergie disponible	Récupérer cette information au moment de la connexion du client
Lien de connexion	Déterminer si la connexion est directe ou via Proxy ou VPN
Type de connexion	Déterminer si le dispositif utilise une connexion via WiFi ou mobile
Durée de la dernière session	Vérifier la durée de la dernière session continue

Lorsque le dispositif A rejoint le réseau de délégation, le mécanisme l'évalue en fonction de chaque élément de CL afin de lui attribuer le niveau de confiance initial. Plus

formellement, le niveau de confiance T_A du dispositif A issu à la fin de cette étape est défini par l'équation suivante :

$$\forall c_i \in CL, T_A = \begin{cases} T_{A^*} + (1/|CL|) & \text{, si } c_i \text{ est satisfait} \\ T_{A^*} & \text{, sinon} \end{cases} \tag{5.6}$$

Tel que T_{A^*} est le niveau de confiance précédent égal à 0 si le mécanisme vérifie le premier élément dans CL, ou égal au dernier niveau calculé dans le cas des autres éléments dans CL. La signification de l'expression "satisfaite" ne veut pas dire "vrai", ce qui est vrai est le résultat de l'évaluation d'un élément de CL. Par exemple, si nous prenons l'élément concernant le niveau d'énergie disponible, et si le dispositif possède au moins 1% d'énergie cette question sera évaluée comme vraie, par contre elle n'est pas satisfaite si le mécanisme exige 50% d'énergie disponible. Autrement dit, un élément satisfait est forcément vrai, mais pas le contraire.

Revenons à notre exemple, nous supposons que le dispositif A possède les valeurs listées dans le tableau 5.II. Selon l'équation 5.6, le niveau de confiance initial attribué à A au moment de sa connexion au réseau de délégation est égal à 0.8. Le mécanisme peut exiger un seuil minimum du niveau de confiance initial pour permettre à un nœud de joindre le réseau de délégation. Une fois la valeur du seuil respectée, le nœud est considéré comme "suffisamment digne de confiance", et pourra éventuellement participer aux opérations de délégation.

Tableau 5.II – Exemple de calcul du niveau de confiance initial

Propriété	**Seuil**	**Valeur de** A	**Satisfaction**
Débit de la bande passante	512Kbps	15Mbps	Oui
Niveau d'énergie disponible	50%	62%	Oui
Lien de connexion	Directe	Proxy	Non
Type de connexion	WiFi	WiFi	Oui
Durée de la dernière session	>15min.	30min.	Oui

Il est possible également de pondérer chaque élément de *CL* selon le contexte. Par exemple dans les applications de télédétection en masse qui collectent souvent des données de l'environnement et qui utilisent des requêtes géodépendantes sporadiques, il est plus important que le dispositif possède les capteurs requis qu'avoir une haute vitesse de connexion. En fait la définition de la liste de contrôle est flexible et le système peut l'adapter selon le contexte de l'application en question.

Après l'obtention du niveau de confiance initial, le mécanisme continue l'évaluation du nœud selon les événements qu'il produit. Pour atteindre cet objectif, nous nous basons sur la notion des chaînes de Markov et sur une table d'événements de confiance. Nous discutons dans la prochaine section les événements de confiance et leur impact sur le niveau de confiance d'un nœud.

5.2.2.2 Définition des événements de confiance

Le niveau de confiance d'un nœud dépend de son comportement au sein du réseau de délégation, et pour évaluer ce comportement nous supposons que l'état de confiance de chaque nœud est un modèle de chaîne de Markov. Nous définissons ainsi l'ensemble des événements qui peuvent augmenter ou diminuer la valeur de confiance. Nous regroupons les événements produits par un nœud selon deux groupes, ceux qui augmentent la valeur de confiance, appelés de "bons événements", et ceux qui la diminuent, dits de "mauvais événements". Comme dans le cas de l'établissement de la relation de confiance, l'ensemble des événements dans cette étape dépend du contexte, et peut être différent d'un LBS à un autre. En outre, chaque événement peut également être pondéré selon son importance dans le contexte en question. Le tableau 5.III liste un exemple d'événements possibles utilisés pour l'évaluation de la confiance d'un nœud donné.

Tableau 5.III – Exemple des événements de confiance

Bon événement	**Mauvais événement**
Connexion normale	Connexion anormale
Déconnexion normale	Déconnexion anormale
Énergie suffisante	Énergie insuffisante
Vitesse connexion suffisante	Vitesse connexion insuffisante
Requête géodépendante complète	Requête géodépendante incomplète
	Requête géodépendante altérée

Comme illustré dans le tableau 5.III, les événements représentent des actions reliées à l'activité d'un nœud et aident à refléter les aspects reliés à l'exécution de son rôle au sein du réseau de délégation. Chacun des événements peut prendre deux états (positif ou négatif) à l'exception de l'événement d'altération des requêtes géodépendantes. Nous expliquons dans la liste suivante chaque événement et la façon de son évaluation.

- **Connexion.** Cet événement est représenté par la façon qu'un nœud utilise pour rejoindre le réseau. Quand un nœud envoie une demande de connexion et attend l'approbation d'une façon normale, la connexion est jugée comme étant normale. Si le nœud envoie plusieurs demandes de connexion consécutives, sa connexion est jugée comme anormale. L'événement de connexion à ce stade est différent de celui survenu au moment de l'établissement de la relation de confiance, du fait que les nœuds ne sont pas connectés au réseau de façon continue.

- **Déconnexion.** Une déconnexion est jugée normale quand le nœud envoie une demande pour quitter le réseau et attend son approbation. Dans le cas contraire, où le nœud quitte le réseau sans aucun préavis, la déconnexion est jugée anormale.

- **Énergie.** Le niveau d'énergie disponible pour un nœud est souvent utile dans le cas des dispositifs mobiles, où l'épuisement de la batterie engendre l'indisponibi-

lité du nœud. Selon le contexte, le mécanisme exige un seuil minimal d'énergie ou chaque valeur au-dessus de ce seuil est jugée comme suffisante.

- **Vitesse connexion.** La vitesse de connexion requise d'un nœud peut varier selon le contexte. Comme dans le cas de l'énergie disponible, le mécanisme peut exiger un seuil minimal de vitesse de connexion qu'il utilise afin d'évaluer la suffisance ou non de la vitesse de connexion du nœud en question.

- **Requête géodépendante.** Cet événement se déclenche à la fin d'un processus de délégation complété dont le nœud à évaluer a déjà fait partie de son chemin de routage. Cela revient aussi à évaluer le nombre des processus de délégation complétés passés par un nœud. La particularité de cet événement est que l'exécution complète d'un processus de délégation déclenche l'événement "requête géodépendante complète" pour tous les nœuds participants. Une exécution incomplète, par contre, ne déclenche l'événement "requête géodépendante incomplète" que pour le nœud où la requête a été perdue. Cela assure que chaque nœud s'évalue indépendamment des autres nœuds participant au même processus.

- **Altération des données.** Le cas de cet événement peut se résumer dans la perte d'une requête au niveau d'un nœud qui ne manifeste aucun comportement anormal. Autrement dit, cet événement se déclenche quand un nœud connecté a suffisamment d'énergie et de vitesse de connexion, mais il ne délègue pas les requêtes qu'il reçoit. Cela revient en quelque sorte à l'événement de "requête géodépendante incomplète", sauf que celui-ci représente une altération successive des requêtes. Le nombre nécessaire des requêtes incomplètes pour assigner cet événement à un nœud dépend du contexte et peut être prédéfini par le mécanisme.

Les événements listés dans le tableau 5.III représentent un exemple de comportement affectant le niveau de confiance d'un nœud, et l'occurrence de chaque événement implique une transition de la valeur de confiance selon le modèle de chaîne de Markov

adopté. Par ailleurs, les événements autres que ceux mentionnés n'engendrent aucune transition et n'affectent pas le niveau de confiance. Ainsi, pour un ensemble d'événements $T_e = \{e_0, ..., e_n\}$ contenant les deux sous-ensembles T_{e^+} et T_{e^-} représentant les bons et les mauvais événements respectivement, la transition d'un état S_i à un autre état S_j dans la chaîne de Markov en fonction d'un événement e_i se reproduit de la façon suivante :

$$S_i \rightarrow \begin{cases} S_{i+1} & \text{, si } e_i \in T_{e^+} \\ S_{i-1} & \text{, si } e_i \in T_{e^-} \\ S_i & \text{, sinon} \end{cases} \tag{5.7}$$

La transition 5.7 assure que chaque événement produit engendre un impact sur l'état actuel du niveau de confiance d'un nœud, ce qui permet une évaluation continue qui ne se déclenche qu'après l'activité du nœud en question. La figure 5.3 illustre la représentation de la transition de l'état de confiance en fonction des événements selon un modèle de chaîne de Markov avec un taux d'arrivée de $\lambda_{i,i+1}$ et un taux de départ de $\mu_{i,i-1}$.

Le taux d'arrivée d'un état donné représente le taux d'augmentation de la confiance de l'état, le taux de départ représente la diminution de la confiance. Ainsi, l'état 0 est le niveau de confiance le plus bas, et l'état n est le niveau le plus élevé. Un nœud donné est considéré comme "indigne de confiance" si son niveau de confiance est plus bas que 0. Cette dernière règle est justifiée par le fait qu'avoir un niveau de confiance inférieur à 0 nécessite un nombre de mauvais événements supérieur à celui des bons événements.

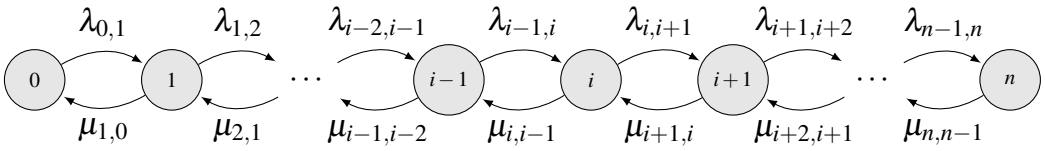


Figure 5.3 – Chaîne de Markov de la transition de confiance

Tel qu'illustré dans la figure 5.3, les transitions d'état de confiance se déroulent d'un

état à l'état suivant ou précédent, et le dernier état atteint est le niveau de confiance actuel d'un nœud donné. Une alternative peut se présenter par l'attribution des poids à chaque événement où les transitions dépendront du poids, ce qui influence les taux de départ et d'arrivée d'un état, et éventuellement, l'évolution du niveau de confiance d'un nœud. Le cas d'utilisation des poids peut être utile dans certains contextes où un événement est plus important qu'un autre. Nous limitons le contexte de cette recherche à des événements non pondérés en supposant que tous les événements ont le même niveau d'importance en ce qui concerne l'évaluation de confiance.

Après avoir défini l'établissement de la relation de confiance ainsi que les événements qui affectent le niveau de confiance, il est important de définir les méthodes de calcul et d'attribution de niveau de confiance à un nœud donné, ce qui constitue le contenu dans la section suivante.

5.2.2.3 Attribution du niveau de confiance

Le point de départ de cette étape est le niveau de confiance obtenu dans la dernière connexion du nœud. Après détermination du nouveau niveau de confiance, il peut être changé en fonction des événements dans la deuxième étape et selon le modèle de chaîne de Markov défini. L'évolution du niveau de confiance d'un nœud est liée à son comportement au sein du réseau, un comportement exprimé par les différents événements émis par le nœud en question.

Pour bien comprendre le fonctionnement de l'attribution d'un niveau de confiance, prenons l'exemple ou nous supposons que la dernière valeur du niveau de confiance obtenu par un nœud donné A est égale à 2. Ce dernier participe ensuite à un processus de délégation complet et par conséquent son état de confiance selon le modèle de chaîne de Markov passe de l'état i à l'état $i+1$ et sa nouvelle valeur de niveau de confiance passe de 2 à 3. Ainsi, la probabilité de transition de l'état de confiance d'un état i à un autre état j pour un membre X est formulée de la façon suivante :

$$p_{ij}(t) = P[X_{t+s} = j | X_s = i] = \frac{n_{ij}}{\sum_{i,j \in S} n_{ij}}, \forall i, j \in S, i \neq j \tag{5.8}$$

Tel que n_{ij} est le nombre de transitions de l'état i à l'état j, et S l'espace de niveau de confiance. La probabilité de transition de l'état de confiance est obtenue selon le modèle de chaîne de Markov définie précédemment. De plus, le modèle de transition de l'état de confiance est considéré comme un *processus de Markov à temps continu*, une propriété qui n'est pas valide sauf si le modèle est homogène, irréductible, et apériodique [18]. Autrement dit, si le modèle de chaîne de Markov respecte ces trois propriétés, le niveau de confiance de chaque nœud peut être calculé en obtenant la *probabilité stationnaire* de la chaîne de Markov en question.

À titre de rappel, la probabilité stationnaire d'une chaîne de Markov représente la période du temps passé dans chaque état de la chaîne de Markov en question. Autrement dit, quelle que soit la distribution de probabilité initiale, après un nombre suffisamment grand de mesures, la distribution de probabilité est approximativement égale à une valeur qui peut être calculée. De façon plus formelle, une probabilité π sur un l'espace E est dit stationnaire pour une chaîne de Markov $(X_n))n \geq 0$ si pour tout $n \geq 0$:

$$(\forall i \in S, P(S_n = i) = \pi(i)) \Rightarrow (\forall i \in S, P(S_{n+1} = i) = \pi(i)) \tag{5.9}$$

Ou, de manière équivalente :

$$\forall j \in S, \sum_{i \in S} \pi(i) p(i, j) = \pi(j) \tag{5.10}$$

Par conséquent, une mesure stationnaire π est appelée probabilité stationnaire si elle respecte les deux conditions suivantes :

$$\forall i \in S, \pi_i \geq 0 \quad \text{et} \quad \sum_{x \in E} \pi_i = 1 \tag{5.11}$$

L'utilisation de la probabilité stationnaire n'est utile que si le modèle de chaîne de

Markov en question respecte les trois conditions d'être homogène, irréductible et apériodique. Nous expliquerons dans la suite l'application de ces trois conditions dans notre contexte.

Homogénéité. Le modèle de confiance proposé a la propriété d'être sans mémoire, ce qui implique que l'état de confiance actuel ne dépend que du dernier état. Par conséquent, le modèle de confiance a la propriété d'homogénéité. Ainsi, pour une période du temps $[e, f]$, la probabilité de transition en temps continu de l'état i à l'état j peut être représentée par :

$$p_{ij}(e, f) = P[X_f = j | X_e = i] \tag{5.12}$$

Tel que $e \leq f$. Un autre point utile pour atteindre le calcul de la probabilité stationnaire peut être achevé en appliquant la loi de Chapman-Kolmogorov [47]. À titre de rappel, la loi de Chapman-Kolmogorov est une équation qui vise à relier les distributions de probabilités conjointes dans les processus stochastiques, notamment les processus stochastiques markoviens. Selon les auteurs de [55], nous pouvons exprimer les transitions d'états d'une chaîne de Markov homogène en appliquant la loi de Chapman-Kolmogorov afin d'obtenir l'équation suivante :

$$p_{ij}(e, f) = \sum_{k \in S} p_{ik}(e, g).p_{kj}(g, f) \tag{5.13}$$

Tel que $0 \leq e \leq g < f$. La représentation de la probabilité de transition selon l'équation (5.13) permet l'application de la valeur de l'état initial égal à π_i quand le temps tend vers l'infini à condition que le modèle soit aussi irréductible.

Irréductibilité. Le changement des états de confiance peut augmenter ou diminuer selon les événements du nœud (Tableau 5.III). Par conséquent, tous les états de confiance $i \in S$ dans le modèle de confiance peuvent être atteints à partir de n'importe quel autre état $j \in S$, et aucun de ses états n'est absorbant. En résultat, nous constatons que le modèle de confiance est irréductible, et nous pouvons appliquer la règle disant que l'état

initial d'un modèle homogène est irréductible quand le temps tend vers l'infini est égal à π_i sur l'équation (5.13) pour obtenir l'équation suivante :

$$\pi_j = \sum_{i \in S} p_{ij}(e,f).\pi(e) \tag{5.14}$$

Le vecteur de probabilité d'état $\pi = [\pi_0, \pi_1, \cdots]$ à n'importe quel moment f peut être représenté par l'équation suivante :

$$\pi(f) = \pi(e).P(e,f) \tag{5.15}$$

Tel que $P(e,f)$ est la matrice de probabilité de transition pour n'importe quels deux états de confiance i et j et à n'importe quel intervalle du temps $[e,f], e \leq f$. Notons ici que $\sum_j \pi_j = 1$ pour tout vecteur de probabilité π qui concerne la transition des états de confiance.

La dernière propriété que le modèle de chaîne de Markov doit assurer pour que nous puissions appliquer la loi de probabilité stationnaire est l'apériodicité, une propriété que nous discutons dans la suite.

Apériodicité. Un déclenchement d'événement est nécessaire pour que le niveau de confiance d'un nœud change d'un état i à un état j. L'ensemble des variables définissant les événements change à travers le temps sans règle précise (la vitesse de connexion par exemple peut augmenter ou diminuer dans la même session), ce qui empêche le processus de transition d'être périodique. En résumé, un processus périodique est un processus dont les événements changent d'impact en fonction des périodes de temps prédéfinies, ce qui n'est pas le cas dans le modèle de confiance proposé. En outre, si un état i est apériodique et fait partie d'un modèle homogène irréductible, tous les autres états $j \in S$ sont forcément apériodiques.

Par conséquent, le modèle de confiance proposé est une chaîne de Markov à temps continu, ce qui implique la possibilité d'estimer la probabilité de l'état de confiance d'un nœud en se basant sur son vecteur unique de probabilité stationnaire. En nous basant sur

les trois propriétés du modèle et sur chacune des équations (5.13) (5.14), et (5.15), nous calculons la probabilité de l'état stationnaire par l'équation suivante :

$$\forall i, j \in S, \quad \pi_j.q_j = \sum_{j \neq i} \pi_i.q_{ij} \tag{5.16}$$

Tel que q_j est le taux des transitions sortantes de l'état de confiance j, et q_{ij} est le taux de transition de i vers j. Notons ici que $\sum_j \pi = 1$ représente la somme des probabilités des états de confiance dans le vecteur respectif à l'état j.

L'équation (5.16) permet l'estimation de l'état de confiance d'un nœud à tout moment en évaluant son taux des transitions. Cependant, vu que chaque nœud garde son état actuel de confiance en mémoire et le reporte au début de chaque processus de délégation, il est plus précis de directement vérifier son état en supposant que toute valeur supérieure à 0 signifie un nœud digne de confiance. Nous discutons en détail des interactions des deux modèles de risques et de confiance au sein du réseau de délégation tout en montrant leur position vis-à-vis les exigences de *Deloc*.

5.3 Fondements applicatifs et validation

En revenant à l'architecture des mécanismes de quantification de risques et de mesure de confiance (Figure 5.1), nous remarquons que l'estimation des risques s'initialise pendant et après le processus de délégation. L'initiation se fait une première fois quand la requête est transmise, puis une autre fois quand les résultats géodépendants sont retournés au demandeur. Nous discutons dans la suite l'applicabilité des deux modèles proposés (section 5.1 et 5.2) dans les deux cas de pré-évaluation (estimation pendant le processus) et de post-évaluation (estimation après le processus), ainsi que le mécanisme de l'estimation de la protection utilisé en conjonction avec les modèles des risques et confiance.

Un déclenchement d'événement est nécessaire pour que le niveau de confiance d'un

nœud change d'un état i à un état j. L'ensemble des variables définissant les événements changent à travers le temps sans règles précises (la vitesse de connexion par exemple peut augmenter ou diminuer dans la même session), ce qui empêche le processus de transition d'être périodique.

5.3.1 Estimation de risques de divulgation de l'information

La première évaluation se fait quand le demandeur initial envoie sa requête au réseau de délégation. Le mécanisme commence par l'estimation de risques en se basant sur le contenu de la requête, puis évalue le niveau de confiance de chaque participant au processus de délégation avant la transmission de la requête.

Selon le modèle proposé dans la section 5.1, la quantification de risques se fait par la comparaison de l'effet du rajout de certaines informations sur le risque d'identification engendré par l'envoi de la requête géodépendante. Notons ici que l'objectif d'un adversaire est d'identifier le possesseur de la requête géodépendante en cours de délégation, et d'essayer de minimiser l'incertitude concernant l'appartenance de la requête en question à un utilisateur donné.

En revenant au modèle de quantification de risques, l'équation (5.3) estime le gain qu'un adversaire peut procurer à propos de l'utilisateur cible en observant ses requêtes géodépendantes et se basant sur les connaissances antérieures qu'il a sur lui. À titre indicatif, l'équation (5.3) est formulée comme suit :

$$I(X;Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} p(X = x_i \cap Y = y_j) \log \left(\frac{p(X = x_i | Y = y_j)}{\sum_{i=1}^{n} p(X = x_i \cap Y = y_j)} \right)$$

Tel que $X = \{x_1, x_2, ..., x_n\}$ est la variable aléatoire qui représente les connaissances antérieures de l'adversaire, et $Y = \{y_1, y_2, ..., y_m\}$ représente les données divulguées.

Pour bien comprendre le déroulement de la procédure de l'estimation des risques, prenons l'exemple d'un utilisateur u_1 qui envoie une requête géodépendante req_1 contenant ses coordonnées géographiques ainsi que le contenu de sa requête tel qu'illustré

dans le listing 5.1.

Listing 5.1 – Exemple d'une requête géodépendante brute

```
{
 "IP": "132.204.26.75",
 "location": "45.4966297,-73.6171364",
 "content":
 {
  "language":"fr"
  "radius": 50
  "type": "restaurant"
  "keyword": "burger"
  "opennow": true
 }
}
```

Tel qu'illustré dans le listing 5.2, une requête géodépendante ne contient que les informations nécessaires pour la récupération des résultats géodépendants. La valeur de la variable *IP* représente l'adresse réseau du dernier dispositif en possession de la requête. Autrement dit, chaque dispositif qui reçoit la requête obtient dans cette valeur l'adresse IP du dispositif adjacent. Cette valeur est mise à jour pendant le processus de délégation en suivant le chemin de la requête jusqu'à son arrivée au service géodépendant.

Les autres données de la requête représentent les informations destinées à l'utilisation par les services géodépendants. Les valeurs qui appartiennent au *contenu* sont mises à titre indicatif, le demandeur initial peut rajouter d'autres variables telles qu'une exigence sur le prix minimal, un numéro de téléphone ou même une adresse physique du point d'intérêt recherché.

Une constatation que nous pouvons relever en observant la requête telle qu'issue de la part du demandeur initial est lorsqu'un adversaire peut corréler l'information de l'adresse IP avec les coordonnées géographiques, il peut augmenter la probabilité d'identification du demandeur.

Autrement dit, si l'adresse IP renvoie l'information disant que l'utilisateur est dans la ville de Montréal, l'adversaire peut être sûr que la requête appartient à cet utilisateur si les coordonnées géographiques dans la requête représentent un endroit à Montréal aussi. Un problème qui peut être résolu en remplaçant la valeur de l'adresse IP par un identifiant qui sert à retracer le chemin vers un utilisateur par le biais d'un serveur, dans notre cas le répertoire inconscient (section 4.1.4).

À titre indicatif, le répertoire inconscient est le composant de *Deloc* responsable de la génération et de l'envoi de la liste des dispositifs actifs aux utilisateurs autorisés. Il représente également la seule entité connectée en tout temps et accessible par tous les utilisateurs de *Deloc*, ce qui justifie son applicabilité dans ce contexte.

De façon plus formelle, pour chaque attribut contenu dans la requête géodépendante, le mécanisme de l'estimation de risques calcule la valeur $I(X;Y)$ correspondante au gain qu'un adversaire peut procurer de cet attribut. Dans notre cas, les deux attributs qui peuvent identifier la position d'un utilisateur sont l'adresse IP et la taille du rayon de recherche avec une valeur de $I(X;Y)$ égale à 1 et 0.9 respectivement. Ces valeurs sont calculées en se basant sur les deux probabilités de connaître la position la ville de l'utilisateur en possédant son adresse IP, et identifier sa position en connaissant sa zone de de recherche de la taille contenue dans la requête (50m). Pour des raisons de simplification de calcul, nous supposons sans le second cas que la zone de recherche soit divisée en 10 zones plus petites dont l'utilisateur est forcément dans une d'elles.

Le listing 5.2 représente la même requête géodépendante avec la variable *sid* qui représente l'enregistrement contenant l'adresse IP d'un dispositif au sein du répertoire inconscient. La valeur de *sid* est une valeur générée à chaque début du processus de délégation et est liée à des données volatiles que le serveur détruit à la fin du processus. Un autre ajustement de la requête concerne le rayon de recherche, *Deloc* essaye de rapporter une valeur de rayon qui est assez large pour imposer un risque au demandeur initial. La valeur de rayon peut être redéfinie manuellement par l'utilisateur, ou automatiquement par *Deloc*.

Listing 5.2 – Exemple d'une requête géodépendante anonymisée

```
{
 "sid": "e09b67a34",
 "location": "45.4966297,-73.6171364",
 "content":
 {
  "language":"fr",
  "radius": 500,
  "type": "restaurant",
  "keyword": "burger",
  "opennow": true
 }
}
```

La notion du risque ici s'explique par le fait que lorsqu'un adversaire en possession de cette requête (listing 5.2) essaye d'identifier, de tracer, ou de trouver une liaison entre le contenu de la requête avec le demandeur initial. En d'autres termes, un adversaire utilise ses connaissances antérieures et les données de la requête afin de connaître si le dernier dispositif qui a envoyé la requête est le demandeur initial. En conséquence, l'estimation des risques se fait par le calcul $I(X;Y)$ tel que X est l'information déjà en possession de l'adversaire, et Y les données divulguées dans la requête géodépendante.

En appliquant les équations (5.3) et (5.4) dans le cas de la requête illustrée dans le listing 5.2, le mécanisme estime le risque en fonction de la requête géodépendante, des participants au processus de délégation, et des connaissances antérieures de l'adversaire. Notons ici que l'estimation des risques au cours du processus de délégation se base sur l'équation 5.4, tandis que celle dans le dispositif final se calcule par l'équation (5.3).

La deuxième évaluation qui doit être faite après la récupération des résultats géodépendants est aussi basée sur la notion de l'information mutuelle (section 5.1). Une fois récupéré, le contenu des résultats est comparé avec la requête initiale pour vérifier si un adversaire peut faire une corrélation, et éventuellement augmenter la probabilité

d'identifier le demandeur initial. La vérification se fait par une analyse du contenu des résultats, qui ne doit contenir que la réponse à la requête géodépendante, avec chaque variable de la requête. Si les résultats contiennent une information identificatrice, cette dernière sera retirée avant que la requête ne retourne au demandeur initial.

Une fois les risques estimés, le mécanisme passe à l'évaluation de la confiance des dispositifs qui participent au processus de délégation. Cela assure que la requête ne passe que par des dispositifs sûrs et fiables. Nous discutons dans la section suivante de l'applicabilité du modèle de confiance proposé dans le contexte de *Deloc*.

5.3.2 Évaluation de confiance de collaborateurs

L'évaluation de confiance durant le processus de délégation est une étape préliminaire pour choisir le prochain dispositif à recevoir la requête. Revenons à l'architecture de base de *Deloc*, le choix d'un dispositif se fait après la réception d'une proposition de la part du répertoire inconscient (section 4.1.4). Ce dernier est responsable de la vérification des estimations et des scores de confiance.

Quand un des événements qui affectent le niveau de confiance se produit, une nouvelle valeur de confiance est donc attribuée au dispositif concerné, cette valeur est directement communiquée au serveur du répertoire inconscient qui joue le rôle d'un chef d'orchestre de gestion des niveaux de confiance. Cela implique que chaque dispositif enregistré dans le répertoire inconscient contient la valeur de son niveau de confiance, et la sélection se fait selon ce niveau. En reprenant la figure 4.5 qui décrit la sélection des dispositifs actifs dans le répertoire inconscient (section 4.1.4), la propriété de "l'usage" est donc remplacée par une nouvelle propriété du *niveau actuel de confiance*. La figure 5.4 illustre un exemple de la nouvelle structure de modèle des enregistrements des dispositifs au niveau du répertoire inconscient basé sur la figure 4.5 .

Tel qu'illustré dans la figure 5.4, la structure proposée des enregistrements dans la section 4.1.4 doit être changée vers une structure qui prend en considération le niveau de confiance. Une alternative pour la gestion des niveaux de confiance peut être l'enregis-

ID	Usage
device_1	25
device_2	13
device_3	7
device_4	43
device_5	5

ID	Niveau de confiance
device_1	2
device_2	1
device_3	6
device_4	7
device_5	3

Figure 5.4 – Nouvelle structure des enregistrements dans le répertoire inconscient

trement du niveau de confiance d'un dispositif dans tous les dispositifs adjacents.

Dans une solution décentralisée, l'utilisation d'un tel modèle nécessite une augmentation du temps d'exécution (le temps de trouver le niveau de confiance d'un dispositif non adjacent), et une communication continue avec les dispositifs adjacents pour rapporter le nouveau score de confiance après chaque événement. Dans notre contexte, l'utilisation du répertoire inconscient est parfaitement convenable en raison de la transparence et de la flexibilité que nous estimons avoir en utilisant *Deloc*.

L'évaluation de la post-confiance à son tour se fait pour chaque dispositif dans le chemin de retour. Rappelons que dans le concept du réseau de délégation, chaque dispositif qui délègue une requête communique avec deux dispositifs adjacents où l'un des deux prend la requête en charge, et l'autre sert comme un dispositif de sauvegarde (section 4.1.5).

Dans le modèle de confiance proposé, nous distinguons deux cas, le premier est quand le dispositif garde ou augmente son niveau de confiance, c'est le cas d'une requête qui poursuit son chemin normalement.

Le deuxième cas concerne le niveau de confiance du dispositif qui diminue, c'est là où la requête passe par le dispositif de la sauvegarde qui doit, à son tour, posséder un niveau de confiance supérieur au seuil prédéfini. Dans le cas où les deux dispositifs échoueraient à l'évaluation du niveau de confiance, la requête est donc perdue et le demandeur initial est notifié pour un renvoi de sa requête.

L'applicabilité des deux modèles de l'évaluation des risques et confiance dans le contexte de *Deloc* est une nécessité. Nous venons de voir que la définition et les fondements théoriques des deux modèles conviennent parfaitement au contexte d'un mécanisme collaboratif tel que *Deloc*. Cependant, ce qui est aussi primordial est l'impact de l'adoption des deux modèles sur le fonctionnement global du mécanisme en termes de performances et efficience.

Nous avons vu dans le chapitre précédent que l'utilisation de *Deloc* peut passer inaperçue pour un utilisateur (section 4.2) et que le mécanisme ne représente aucune surcharge remarquable en termes de performance et d'efficience. Par conséquent, nous discutons dans la suite de ce chapitre de l'impact de l'application des deux modèles de l'évaluation de risques et de confiance sur le fonctionnement global de *Deloc*, ainsi que des indices de performances et efficience associée à leur utilisation.

5.4 Tests et simulation

En nous basant sur l'environnement simulé décrit dans la section 4.2, nous évaluons les indicateurs d'efficience et d'efficacité de *Deloc* dans la présence des deux modèles de quantification des risques et mesure de confiance. Ce que nous rajoutons à l'environnement simulé seront les fonctions d'estimation du risque calculées dans le dispositif concerné, et les mécanismes d'évaluation de confiance dans le serveur du répertoire inconscient.

Préparer l'environnement de simulation

Afin de simuler les événements qui influencent la confiance au sein du réseau, nous utilisons un minuteur qui déclenche aléatoirement un des événements dans des périodes non uniformes. L'objectif est de pouvoir répliquer le comportement d'un dispositif réel. La figure 5.5 illustre un exemple d'une série des événements déclenchés par le minuteur.

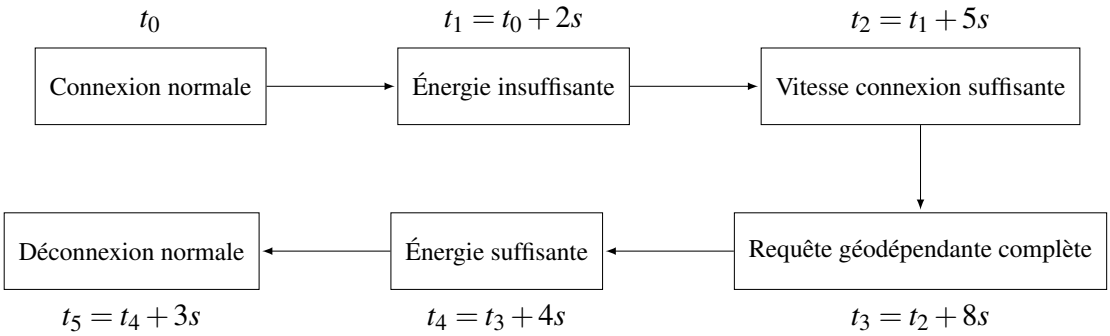


Figure 5.5 – Exemple d'une série des événements dans l'environnement simulé

Le minuteur renvoie des événements sélectionnés aléatoirement à partir de deux ensembles des bons et des mauvais événements tels que décrits dans le tableau 5.III. L'utilisation d'une telle approche assure une dispersion non uniforme similaire au comportement d'un dispositif réel. Notons ici que l'événement "requête géodépendante complète" ne fait pas partie des événements aléatoires, et n'est envoyé que dans le cas où une requête s'exécute réellement en passant par le dispositif simulé en question.

La quantification des risques se fait dans le dispositif de l'utilisateur concerné une fois que l'utilisateur se sert de *Deloc*, ce dernier effectue la quantification nécessaire des risques avant de transmettre la requête au réseau de délégation. La méthode utilisée pour quantifier les risques est celle décrite dans la section 5.1, et les équations implémentées sont (5.3) et (5.4).

En revanche, l'évaluation de confiance se fait à son tour au niveau du serveur du répertoire inconscient qui se charge de garder trace de tous les événements des dispositifs connectés à un instant donné. À la déconnexion d'un dispositif, le répertoire inconscient enregistre le dernier niveau de confiance atteint par le dispositif pour des utilisations ultérieures, et supprime par la suite le reste des événements. L'utilisation d'une telle approche permet la garantie de confidentialité dans le cas où le répertoire inconscient est compromis vu que seul le dispositif qui a généré les événements peut les garder.

L'autre particularité liée à l'intégration du modèle de confiance est l'utilisation d'une structure de données locale qui tient les transitions des états de confiance. Chaque dis-

positif possède un tableau enregistré au niveau local qui garde l'évolution de son niveau de confiance.

L'utilité d'enregistrement local des événements réside dans la permission au dispositif de définir son propre niveau de confiance, l'utilisation de ce dernier pour la revérification au niveau du serveur, et l'identification possible des dispositifs malveillants qui tentent de biaiser l'échange des événements.

Simulation des modèles de risques et de confiance

La procédure de simulation est ensuite exécutée avec les mêmes donnés utilisées dans la section 4.2 contenants 9963 dispositifs réels et 3956113 ($\approx 4M$). L'exécution de la procédure de simulation a généré 55621 requêtes géodépendantes complétées. Nous discutons dans la suite de cette section des résultats de la simulation en les comparant au résultat initial obtenu avant l'incorporation des modèles de quantification des risques et d'évaluation de confiance.

En reprenant les mêmes mesures discutées dans la section 4.2, nous élaborons une comparaison entre les indicateurs d'efficience obtenus avant et après l'intégration des modèles de quantification et évaluation.

L'intégration du modèle des risques, qui est faite au niveau local, implique un calcul supplémentaire léger qui sert à identifier les risques de divulgation de certaines données comprise dans la requête géodépendante, une procédure qui ne nécessite pas une grande capacité de calcul.

L'intégration du modèle de confiance à son tour, qui est fait au niveau du répertoire inconscient, exige le suivi et l'enregistrement des événements des dispositifs, des événements qui se produisent même en absence du modèle, ce dernier qui n'implique qu'à les suivre et les utiliser dans l'estimation de confiance.

La figure 5.6 illustre les principaux indicateurs d'efficience utilisés pour l'évaluation de *Deloc*. Nous remettons les résultats de la section 4.2 en parallèle avec les nouveaux résultats obtenus après l'intégration des deux modèles afin de pouvoir observer la diffé-

rence entre les deux cas.

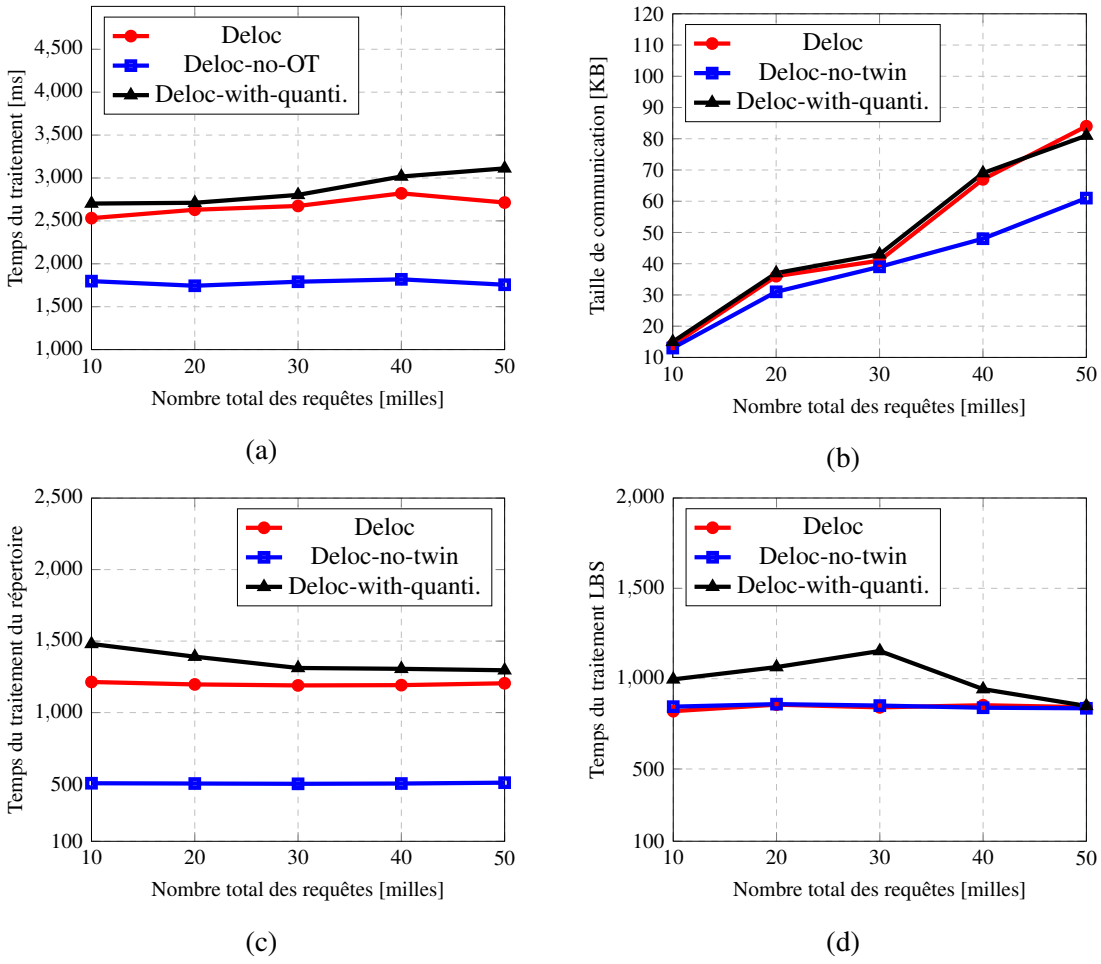


Figure 5.6 – Indicateurs d'efficience de *Deloc* après l'intégration des modèles de quantification

Tel que nous pouvons constater à partir de la figure 5.6, la surcharge imposée par l'intégration des modèles de quantification des risques et d'estimation de confiance ne représente pas une contrainte pour l'utilisation de *Deloc*. Le temps moyen de traitement (figure 5.6a) est légèrement plus long avec l'utilisation des deux modèles.

Cependant, la différence du temps est en millisecondes ($\approx 1500ms$) et peut passer inaperçue dans la plupart des cas. La figure 5.6b qui représente la taille moyenne de communication en utilisant *Deloc* montre l'indifférence entre le déroulement avec ou

sans les modèles de quantification et estimation.

Quant au répertoire inconscient qui se charge de l'évaluation de confiance (figure 5.6c), le temps du traitement engendré par l'intégration du modèle n'impose aucune surcharge significative, et son exécution ne rajoute pas de délais au processus de sélection des dispositifs actifs. Finalement, le temps du traitement d'une requête géodépendante par le LBS (figure 5.6d) n'est pas affecté par l'intégration des deux modèles vu que le LBS traite les requêtes indépendamment du *Deloc*.

Les résultats de la simulation après intégration des modèles de quantification des risques et évaluation de confiance confirment l'efficacité de *Deloc* face aux architectures et types des services géodépendants actuels. En outre, la surcharge minimale qu'il implique, ainsi que les garanties rigoureuses de confidentialité et d'utilité lui confèrent un avantage par rapport aux autres approches et mécanismes.

5.5 Conclusion

La gestion d'utilisateurs d'un réseau représente un défi délicat dans les mécanismes collaboratifs. Contrairement au cas des mécanismes centralisés, l'identification des anomalies dans les mécanismes collaboratifs nécessite des mesures fiables et exactes.

Pour atteindre nos objectifs, nous avons rajouté au mécanisme de délégation deux modèles de quantification des risques et mesure de confiance, qui interagissent afin de permettre au mécanisme de garder le contrôle. Le contrôle est nécessaire car les utilisateurs peuvent représenter des menaces aux autres usagers.

Nous avons proposé un modèle de quantification de risques basé sur la théorie de l'information mutuelle, et un modèle d'évaluation de confiance basé sur les chaînes de Markov. Nous avons discuté des fondements théoriques des deux modèles, et démontré leur efficacité. Nous avons démontré également leur surcharge négligeable dans l'exécution du processus de délégation, ainsi que leur nécessité et leur rôle dans la réalisation d'un LPPM optimal.

Revenons aux particularités des LPPM collaboratifs, leur nature dynamique implique que l'utilisation d'une métrique classique pour estimer leur efficacité en matière de protection de la confidentialité peut mener à des estimations imprécises. Nous justifions cette affirmation par la nature de l'échange des requêtes dans un mécanisme collaboratif où le contenu est souvent altéré, combiné, ou changé. Par conséquent, nous proposons dans le chapitre suivant une métrique de confidentialité conçue pour estimer le niveau de protection fournie par un mécanisme collaboratif. Nous appelons la métrique δ-*fuzziness*, inspirée des théories de la logique floue [65].

Troisième partie

Validation et Applications

CHAPITRE 6

MÉTRIQUE FLOUE DE CONFIDENTIALITÉ

La métrique de confidentialité représente l'outil utilisé pour valider le respect d'un LPPM à un ensemble des exigences de confidentialité. Connue également, sous le nom de mesure de confidentialité, une métrique de confidentialité décrit l'ensemble des critères formels qui servent à quantifier le niveau de protection assurée par un mécanisme donné, et cela vis-à-vis des objectifs de protection prédéfinis [27]. Le choix et la conception d'une métrique dépendent essentiellement du paradigme utilisé dans un LPPM, ainsi que les objectifs qu'il doit atteindre.

Dans le cas de *Deloc*, l'utilisation d'une métrique conçue pour des mécanismes basés sur la transformation de coordonnées géographiques (*par ex.* substitution ou confusion) risque de donner des évaluations non précises. De plus, vu que le contenu à protéger dans le cas des mécanismes collaboratifs inclut des données de plusieurs individus dans un même ensemble, il est nécessaire d'utiliser une métrique adaptée aux mécanismes collaboratifs pour bien estimer le niveau de confidentialité. Dans le contexte de cette recherche, nous proposons une métrique inspirée de la théorie des ensembles flous que nous nommons δ-*fuzziness*. La métrique proposée prend en considération les particularités des mécanismes collaboratifs, notamment de *Deloc*.

6.1 Métriques de confidentialité existantes

Nous distinguons deux catégories majeures : les métriques qui se basent sur des méthodes computationnelles, et celles basées sur la théorie des probabilités. Nous détaillerons chaque catégorie dans la suite de cette section.

6.1.1 Métriques computationnelles

Les métriques de cette catégorie utilisent des méthodes et des règles de calcul bien définies afin d'évaluer la confidentialité assurée par un LPPM. Elles sont le plus souvent utilisées dans des mécanismes basés sur *l'obscurcissement*, et ne prennent pas en considération l'adversaire ou l'utilité souhaitée.

Une métrique computationnelle parmi les plus connues dans ce contexte est l'adaptation de k-anonymity aux services géodépendants [114]. Le concept a été initialement proposé pour la publication confidentielle des données (*Private Data Publishing*), et a été rapidement adopté dans plusieurs domaines, incluant les services géodépendants. Le niveau de confidentialité est mesuré en fonction de la valeur k, qui indique si un adversaire peut ré-identifier un utilisateur dans un groupe de k utilisateurs. Ainsi, nous disons qu'un mécanisme est k-anonyme s'il peut assurer que l'utilisateur n'est pas identifiable parmi au moins $k-1$ autres utilisateurs.

Dans le contexte des services géodépendants, les critères de k-anonymity sont atteints différemment. Par conséquent, la confidentialité dans les LPPM est exprimée par des zones géographiques. De ce fait, l'adaptation de k-anonymity est liée aux mécanismes qui se basent sur l'obscurcissement, qui remplace les coordonnées géographiques d'un utilisateur par une zone plus large dans le but de le protéger. Ainsi, on dit qu'une zone géographique est k-anonyme si elle est indiscernable parmi $k-1$ autres zones à proximité.

Plus formellement, une zone est k-anonyme si la probabilité de ré-identification de la position d'un utilisateur est égale à $1/k$. L'utilisation d'une métrique basée sur k-anonymity se fait souvent avant la génération de la zone protégée, vu que la valeur de k est un fondement pour calculer la zone [27]. La figure 6.1 illustre un exemple de k-anonymity ou la valeur de $k = 3$, cela veut dire que la métrique assure l'impossibilité d'identifier les positions exactes d'au moins 3 utilisateurs dans la zone 3-anonyme.

Des métriques alternatives à k-anonymity ont été proposées, notamment l-diversity

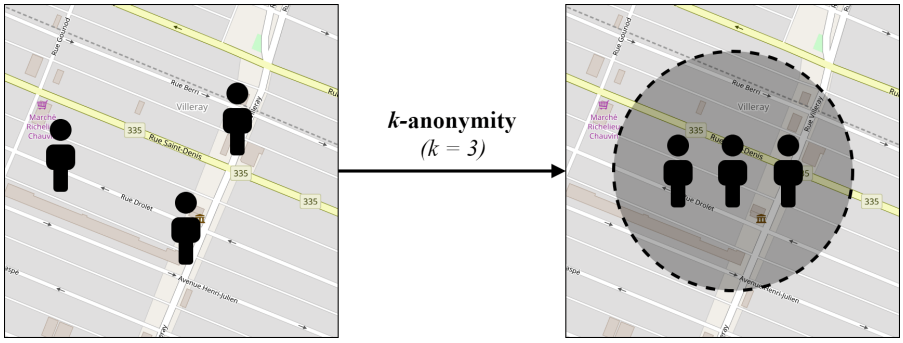


Figure 6.1 – Exemple de k-anonymity ($k = 3$)

[80] et t-closeness [72], afin de remédier aux faiblesses de k-anonymity, telles que l'échec face aux attaques utilisant les données contextuelles. L'objectif de ces métriques est la prise en considération des informations contextuelles et sémantiques liées à une position. Autrement dit, ces deux métriques évaluent l'efficacité d'un mécanisme dans des endroits ou la ré-identification serait triviale si nous utilisions k-anonymity (*par ex.* endroits publics). Pendant que l-diversity assure qu'un endroit est indiscernable parmi $l-1$ autres endroits différents en le regroupant selon leurs données contextuelles, t-closeness garantit qu'un endroit visité par un utilisateur apparaît au même nombre de fois pour tous les utilisateurs dans une zone donnée.

Les méthodes computationnelles échouent face aux modèles d'attaques actuels [78, 110]. Un adversaire possède souvent une solide connaissance antérieure concernant un utilisateur, et peut corréler différentes requêtes géodépendantes afin de soutirer de l'information. Ainsi, les recherches ont été rapidement orientées vers des modèles plus solides, basés sur la théorie de probabilités.

6.1.2 Métriques probabilistes

Les métriques probabilistes sont des modèles d'évaluation de confidentialité qui sont basés sur le calcul de *la probabilité de ré-identification* d'un utilisateur. Plusieurs métriques ont été proposées dans cette catégorie, elles offrent des garanties plus rigoureuses que celles des métriques computationnelles, et ne dépendent pas du paradigme utilisé

dans un mécanisme (chapitre 2 section 2.3.1).

D'un point de vue général, les modèles probabilistes décrivent l'adversaire en termes de probabilités, en prenant en considération qu'un adversaire pourrait posséder des connaissances antérieures concernant l'utilisateur, et que ces dernières pourraient être utilisées en les corrélant avec les coordonnées géographiques récupérées. Les connaissances antérieures sont souvent exprimées par des variables aléatoires, et les capacités d'inférence par des modèles statistiques d'inférence.

Plusieurs sous-classes des modèles probabilistes ont été proposées. Alors que certains mesurent l'efficacité d'un LPPM en fonction de la protection de la position réelle, d'autres expriment cette protection par la sensibilité de la position révélée (*par ex.* révélation de la position du domicile face à celle du café préféré).

Métriques basées sur les entropies

L'entropie est une mesure du degré moyen d'incertitude associé à un ensemble d'événements. Plus formellement, l'entropie associée à un ensemble contenant N éléments est définie par :

$$h = -\sum_{1}^{n} p_i \log p_i$$

Tel que p_i représente la probabilité d'occurrence de l'événement i. Dans le contexte des LBS, les entropies ont été utilisées comme base à plusieurs métriques d'évaluation de la confidentialité. En résumé l'adaptation des entropies implique le calcul d'incertitude liée à l'application d'un mécanisme donné.

Le premier mécanisme qui adopte le principe des entropies pour évaluer la confidentialité a fait son apparition en 2004, ou les auteurs du mécanisme nommé *Mix Zones* [12] utilisent des zones géographiques prédéfinies dans lesquelles les requêtes géodépendantes ne sont pas envoyées au LBS. En d'autres termes, le mécanisme protège les utilisateurs par *l'interdiction* des requêtes dans une zone prédéfinie, et *la confusion* des

pseudonymes des utilisateurs au moment de la quitter. La figure 6.2 illustre un exemple d'une zone de confusion ou les pseudonymes de 3 utilisateurs sont changés au moment de quitter la zone.

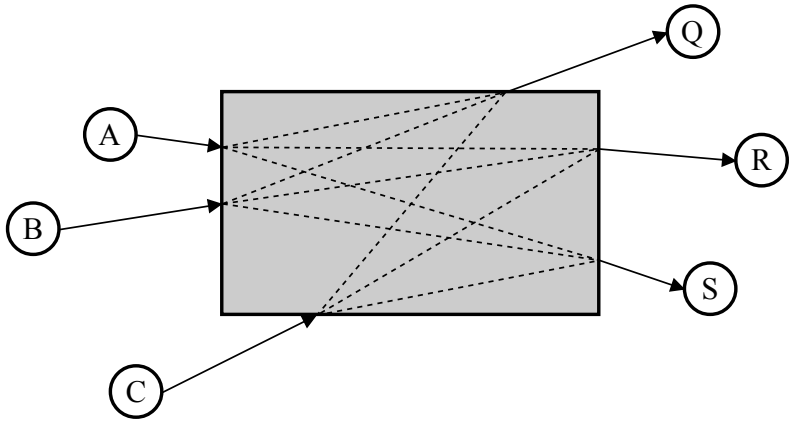


Figure 6.2 – Exemple de *Mix Zones*

L'évaluation de ce mécanisme se fait par la mesure de l'incertitude concernant le pseudonyme qu'un utilisateur a eu avant d'entrer dans la zone, et celui qu'il a obtenu après l'avoir quitté. De façon plus formelle, considérons un ensemble d'utilisateurs N qui entrent une zone d'interdiction (*mix zone*) X avec la liste des pseudonymes $P = p_1, p_2, ..p_n$ ou chaque pseudonyme est associé à un utilisateur. Le même groupe d'utilisateurs N quittera la zone avec d'autres pseudonymes représentés par l'ensemble $Q = q_1, q_2, ..q_n$. Le rôle des entropies dans un tel mécanisme est de mesurer l'incertitude de lier un élément de P avec un autre élément de Q, et qui est formulée avec la probabilité de trouver un ensemble M qui contient toutes les similarités entre P et Q :

$$h = -\sum_i Pr(m_i|M) \log Pr(m_i|M)$$

D'autres mécanismes qui utilisent les entropies pour leurs évaluations ont été proposés [1, 121]. La plupart d'entre eux ont été basés sur la technique des *mix zones* avec des optimisations concernant la sélection de la zone d'interdiction et sa sémantique par rapport aux utilisateurs. En résumé, ils adoptent les entropies afin de mesurer l'incer-

titude liée à un changement d'identité qui se fait à la présence de certaines conditions (zone prédéfinie, nombre de requêtes, etc.). Par exemple, interdire les requêtes géodépendantes quand l'utilisateur arrive à son domicile, et les reprendre quand il le quitte après l'attribution d'un nouveau pseudonyme.

Métriques basées sur la marge d'erreur

Utilisées souvent pour mesurer la protection d'une trajectoire, les métriques basées sur la marge d'erreur évaluent la distance entre la position réelle et la position résultante d'un mécanisme. L'objectif est d'assurer que l'adversaire ne puisse pas identifier la position exacte en maximisant la distance.

Parmi les travaux les plus représentatifs qui se basent sur cette métrique, nous citons celui de Hoh *et al.* où les auteurs maximisent la distance estimée en mixant les trajectoires des utilisateurs [58]. De façon plus formelle, la trajectoire d'un utilisateur est remplacée par celle d'un autre utilisateur si les deux se croisent dans l'objectif de tromper l'adversaire. Pour deux ensembles N contenant un groupe d'utilisateurs et M un ensemble des horodatages de leurs positions, la marge d'erreur pour une trajectoire d'un utilisateur $u \in N$ est définie comme suit [49] :

$$E(u) = -\frac{1}{NM} \sum_{i=1}^{M} \sum_{j=1}^{I_i} p_j(i) d_j(i)$$

Tel que I_i est le nombre total de possibilités de présence de l'utilisateur u à l'instant i, p_j représente la probabilité associée à l'hypothèse j à l'instant i, et $d_j(i)$ la distance entre la position réelle de l'utilisateur u à l'instant i et la position estimée.

L'utilisation de la marge d'erreur comme métrique de confidentialité impose l'utilisation des mécanismes qui maximisent la distance entre la position réelle d'un utilisateur et la position calculée. De ce fait, son impact sur l'utilité de l'information remet en question son efficacité dans des applications actuelles, ou encore dans les LBS dédiés à la navigation pour lesquels la métrique a été initialement proposée.

Confidentialité différentielle

Initialement proposée dans le domaine des bases de données statistiques, la confidentialité différentielle est utilisée dans plusieurs domaines pour assurer la confidentialité pendant l'analyse des données, notamment lors de l'exploration de données statistiques et de l'indexation géospatiale [36, 37] . Elle se démarque des métriques précédentes par une perspective différente de la confidentialité. Plutôt que de tenter de regrouper les individus, minimiser l'incertitude ou maximiser la marge d'erreur, la confidentialité différentielle tente de limiter le risque supplémentaire lié à l'utilisation d'un service où ses données se retrouvent dans l'ensemble en question.

Nous dirons qu'une opération sur un ensemble de données $D(A)$ est différentiellement confidentielle si nous pouvons inférer environ la même quantité d'information sur un individu u, qu'il soit présent ou non dans $D(A)$. Plus formellement, nous définissons la confidentialité différentielle comme suit :

Définition 6.1. ***ε-confidentialité différentielle***. *Un mécanisme de randomisation A qui est ε-différentiellement confidentiel si, pour tous les ensembles de données D_1 et D_2 différents d'au plus par un seul tuple de données :*

$$\frac{Pr[A(D_1,q)=r]}{Pr[A(D_2,q)=r]} < e_{\varepsilon}$$

Telle que r est la réponse retournée par A en prenant les ensembles de données D_1 et D_2 et la requête r comme entrée.

Dans le contexte de géolocalisation, la confidentialité différentielle a été adoptée dans plusieurs travaux, où les auteurs proposent un meilleur mécanisme qui aide à atteindre les critères de confidentialité différentielle [7, 19, 39, 126]. Son adaptation nécessite des paramètres que le partage confidentiel conventionnel ne prend pas en considération telle que les coordonnées géographiques et l'horodatage des données. Par exemple, un adversaire peut identifier la relation entre deux positions en analysant la différence

entre leurs horodatages.

Andrés *et al.* tentent d'atteindre la confidentialité différentielle en proposant une adaptation nommée *Geo-indistinguishability*, qui se base sur l'ajout de bruit randomisé à une position en se basant sur la loi de *Laplace* [7]. ElSalamouny *et al.* proposent une variante plus rigoureuse en matière de protection de vie privée, appelée *(D-ε)-Location Privacy* où ils rajoutent le paramètre *D* qui représente la distance minimale pour qu'un mécanisme respecte la confidentialité différentielle [39].

Malgré ses fortes garanties concernant la vie privée des utilisateurs, la confidentialité différentielle implique quelques faiblesses. Toujours dans le contexte des LBS, l'utilisation d'un mécanisme ε-différentiellement confidentiel est inutile dans le cas des services qui utilisent des requêtes continues (les applications de navigation telles que *Waze* ou *Google Maps*). En outre, l'accumulation du bruit rajouté à chaque mise à jour de coordonnées finira par rendre l'information transmise complètement erronée et par conséquent nuire à l'utilité du service. De plus, un mécanisme optimal en termes d'efficience est encore requis pour que l'adoption de la confidentialité différentielle dans les LBS atteigne les mêmes exigences rigoureuses initialement proposées pour les bases de données statistiques.

Le choix d'une métrique de confidentialité dépend essentiellement des objectifs de protection envisagés par un LPPM. Ces objectifs sont mis en fonction de la classe LBS visée. Autrement dit, chaque métrique de confidentialité est conçue et peut garantir une évaluation précise pour un sous-ensemble de classes seulement. De ce fait, le choix de la métrique représente une étape primordiale lors de la mise en œuvre d'un LPPM.

6.2 Contexte théorique

Comme nous venons de mentionner, les métriques existantes sont principalement conçues pour des mécanismes qui effectuent des opérations de transformation sur les

coordonnées géographiques d'un utilisateur. Autrement dit, elles mesurent le niveau de confidentialité, l'utilité de l'information résultante, et la portée d'une attaque potentielle dans le cas où les coordonnées résultantes sont différentes des originales.

Notre métrique δ-*fuzziness* est une métrique probabiliste qui vise à estimer les risques d'identification associés au partage d'une position. L'estimation se fait par rapport aux autres positions formant la connaissance d'un adversaire.

Afin de pouvoir définir δ-*fuzziness* de façon formelle, nous définissons les bases de notre métrique selon trois modèles principaux : le modèle de *l'adversaire*, le modèle de *la confidentialité*, et celui de *l'utilité*. Nous discutons ces modèles et leurs implications dans la métrique proposée dans les sections qui suivent.

6.2.1 Modèle d'adversaire

La représentation d'un adversaire dépend de deux facteurs principaux : *les connaissances antérieures* qu'il peut collecter à propos de ses victimes, et *les attaques* qu'il peut réaliser ainsi que la probabilité de leur succès.

6.2.1.1 Connaissances antérieures

Selon le travail de Wernke *et al.* [123], les connaissances antérieures d'un attaquant peuvent être vues sous deux dimensions, à savoir *l'information temporelle* et *l'information contextuelle*. La dimension temporelle décrit si un attaquant peut avoir accès à une seule paire ou un ensemble de coordonnées géographiques dans une période donnée de temps. En pratique, la plupart des mécanismes existants ne considèrent que le cas où l'adversaire n'accède qu'à une seule position. Cependant, le cas où un adversaire accède à un ensemble des positions, parfois des trajectoires, est le cas qui reflète la majorité des LBS actuels, ce qui atteste l'insuffisance de ces mécanismes. En fait, les services géodépendants enregistrent toutes les requêtes reçues, qui forment en accumulant une connaissance antérieure contenant des positions, des trajectoires, et des requêtes géodépendantes.

En ce qui concerne la dimension contextuelle, l'information prise en considération est toute information supplémentaire au-delà des coordonnées spatiotemporelles. L'attaquant peut utiliser ces informations en les corrélant avec des positions connues de l'utilisateur. Par exemple, un attaquant pourrait diminuer la taille de la zone d'obscurcissement d'un utilisateur en utilisant une carte du réseau routier pour déterminer où l'utilisateur peut se déplacer.

Plus formellement, les connaissances antérieures qu'un adversaire peut collecter sur un utilisateur u peuvent être représentées par l'ensemble B_u contenant les événements et les informations contextuelles liés à la position de l'utilisateur. Pour un utilisateur u, l'ensemble des connaissances antérieures liées à sa position l_u est représenté par l'ensemble $B_u = \{R_u, C_u\}$, telle que R_u est une matrice qui représente les événements reliés à l'utilisateur u et C_u un ensemble des vecteurs représentant les informations contextuelles de chaque position l_u.

Un exemple d'informations contextuelles peut s'illustrer par les contraintes de mouvements des utilisateurs au sein d'une région donnée. Par exemple, il se peut qu'il ne soit pas possible de se déplacer entre deux régions lointaines dans une période courte de temps, ou de se déplacer entre deux régions en raison d'obstacles physiques.

En nous basant sur la modélisation de connaissances antérieures décrite dans un travail de Shokri *et al.* [108], nous assumons que les événements collectés par l'adversaire sont représentés par une matrice de transition. La raison d'utilisation des matrices est de pouvoir exprimer les événements de manière mathématique, et de pouvoir éventuellement estimer l'effet des connaissances antérieures sur le modèle d'adversaire. En d'autres termes, la matrice de transition R_u contient les informations sur la présence d'un utilisateur dans une position donnée et représente l'ensemble des positions observées par l'adversaire.

La matrice de transition reflète la présence de l'utilisateur dans des sous-régions d'une région géographique donnée, et peut être exprimée par une chaine de Markov. Ainsi, un profil utilisateur P_u est une matrice de transition de la chaine de Markov cor-

respondante à l'utilisateur u tel que chaque entrée dans le profil représente la probabilité décrivant le mouvement de l'utilisateur u vers la sous-région r_j sachant qu'il est dans la sous-région r_i.

Une représentation précise de C_u peut être achevée par l'utilisation d'un ensemble de vecteurs, où chaque paire (i, j) dans R_u est reliée à un vecteur contant les informations contextuelles respectives. En d'autres termes, un vecteur v_{ij} utilisé par l'adversaire pour représenter les informations contextuelles liées à (i, j), peut être représenté par $v_{ij} = (w_1, w_2, \cdots, w_n)$, tel que chaque w_n de v_{ij} représente une valeur fonction de la présence d'une information contextuelle.

L'objectif final d'un adversaire dans le processus de formation des connaissances antérieures est d'inférer le profil P_u d'un utilisateur en connaissant les positions observées représentées par la matrice R_u et les informations contextuelles représentées par C_u. Cela revient à estimer :

$$Pr(P_u|C_u, R_u) \tag{6.1}$$

L'efficacité d'un modèle d'adversaire ne dépend pas uniquement de l'information observée ou de la connaissance antérieure, mais aussi de l'efficacité des techniques et mécanismes d'attaques utilisée. Nous discutons dans la suite les principales techniques d'attaques et leur applicabilité dans notre contexte.

6.2.1.2 Techniques d'attaques

Une attaque géodépendante vise toujours à définir la position réelle de l'utilisateur. Par conséquent, un adversaire tente de déterminer cette position par l'utilisation de la corrélation des coordonnées géographiques précédentes, ou celle des données contextuelles. Dans les deux cas, l'efficacité de l'attaque dépend de la disponibilité et de la quantité des informations antérieures. En nous basant sur le modèle de Shokri *et al.*, nous définissons deux catégories d'attaques qui regroupent la majorité des attaques.

Attaques de divulgation de présence

Dans ce type d'attaques, l'adversaire analyse une ou plusieurs requêtes géodépendantes afin de déterminer la position réelle d'un utilisateur. Les attaques effectuées en analysant une seule position sont souvent exécutées face aux mécanismes de transformation évalués par des métriques computationnelles (*par ex.* k-anonymity). D'ailleurs, parmi les faiblesses majeures liées à l'utilisation de ce type de mécanismes nous citons *l'homogénéité des clusters*, et *la distribution des positions révélées*.

Les attaques utilisant l'homogénéité des clusters cherchent à identifier les positions qui ne sont pas suffisamment éparpillées dans une zone géographique donnée. Autrement dit, cette vulnérabilité se produit quand l'utilisateur et les autres utilisateurs désignés pour cacher sa position réelle sont dans une petite zone (*par ex.* des collègues qui se trouvent dans le même immeuble). Par conséquent, l'adversaire pourra identifier la position de tous les membres, y compris l'utilisateur en question.

Quant aux attaques basées sur la distribution des positions révélées, l'attaquant tente d'observer si le mécanisme suit un modèle de distribution prévisible, par exemple, il essaye de définir si tous les clusters sont dans la même région, ou s'ils sont distribués de façon observable.

Un adversaire vise à identifier la position réelle de l'utilisateur en calculant la probabilité suivante :

$$Pr\{l_u(t) \in c | o_u, P_u\} \tag{6.2}$$

Telle que $l_u(t)$ est la position réelle de l'utilisateur u à l'instant t, c est le cluster représentant la zone géographique la plus petite utilisée par l'adversaire, o_u est la position observée, et P_u est le profil utilisateur estimé durant la formation des connaissances antérieures.

Attaques de divulgation de rencontres

Ce type d'attaques profite des informations contextuelles incluses dans la requête géodépendante (*par ex.* adresse postale), ou des informations disponibles publiquement (*par ex.* cartes géographiques, annuaires téléphoniques). L'adversaire utilise les données contextuelles qu'il possède pour augmenter l'estimation de la probabilité de présence d'un utilisateur à une position donnée et pouvoir éventuellement la relier aux positions d'autres utilisateurs. L'objectif de l'adversaire dans ce cas est de corréler les données contextuelles afin de trouver une relation entre les utilisateurs dans le but d'inférer leurs positions communes.

Un exemple de l'utilisation des données contextuelles peut être l'utilisation des données d'une carte géographique. Face à un mécanisme d'obscurcissement, l'utilisation de telles données peut aider à inférer la position exacte d'un utilisateur. La figure 6.3 illustre deux exemples d'utilisation des données contextuelles issues d'une carte afin de limiter les zones ou un utilisateur pourra se trouver.

(a) (b)

Figure 6.3 – Exemples de la corrélation des données contextuelles

Comme nous pouvons le voir sur la figure 6.3, un mécanisme d'obscurcissement qui révèle la zone illustrée en cercle noir permet à un adversaire de limiter la probabilité de la position réelle d'un utilisateur au centre hospitalier sur la carte (figure 6.3a). De même,

la zone révélée dans la figure (6.3b) limite la probabilité au bout de la route obtenue par la corrélation avec des données cartographiques.

En termes de probabilités, l'adversaire tente d'estimer la probabilité suivante afin de révéler la position de l'utilisateur :

$$Pr\{l_u(t), l_v(t) \in c | o_u, o_v, P_u\} \tag{6.3}$$

Telles que $l_u(t)$ et $l_{uv}(t)$ sont les positions réelles des utilisateurs u et v à l'instant t, c est le cluster représentant la zone géographique la plus petite utilisée par l'adversaire, o_u, et o_v les positions observées des utilisateurs u et v, et P_u est le profil utilisateur estimé durant la formation des connaissances antérieures.

En prenant en considération les deux catégories d'attaques définies précédemment, et la possibilité que l'adversaire puisse être le LBS même, notre métrique doit supposer que toute information échangée est susceptible d'être exploitée par un adversaire. Par conséquent, une métrique efficace doit assurer que les attaques mentionnées ne peuvent révéler la position réelle d'un utilisateur ni son comportement géodépendant.

6.2.2 Modèle de confidentialité

Un modèle de confidentialité dépend initialement de ses objectifs et du mécanisme utilisé pour les atteindre. Les objectifs de protection concernent les données échangées, notamment l'identité et la position, et le comportement. Pendant que l'identité représente toute information sensible qui pourra aider à inférer d'une façon unique l'identité réelle d'un utilisateur, la position et le comportement sont principalement liés aux coordonnées de l'utilisateur. La position reflète les coordonnées géographiques d'un utilisateur à un instant donné, et le comportement représente l'ensemble de ces coordonnées et l'information identifiable qu'un adversaire pourra tirer (*par ex.* trajectoire de l'utilisateur allant du point A au point B).

Protection de l'identité

L'objectif est de protéger la vie privée des utilisateurs. Par conséquent, même les LBS peuvent tirer profit de la confidentialité assurée pour chacun de leurs utilisateurs. Ainsi, si nous assurons que les utilisateurs sont complètement anonymes, le LBS ne sera pas obligé par la loi d'obtenir le consentement préalable des utilisateurs ou de limiter l'utilisation des données. Cela est particulièrement pertinent lorsque les données sont obtenues à partir d'une population que leurs données sont potentiellement utilisées à des fins difficiles à anticiper.

Cependant, un défi majeur d'assurer l'anonymat réside dans le fait que le simple remplacement de l'identifiant de l'utilisateur par un pseudonyme n'est pas suffisant parce que la position peut être facilement associée à une connaissance externe, dans le cas le plus simple d'une application mobile, par exemple l'accès de l'application aux contacts d'un utilisateur peut dévoiler l'identité d'un utilisateur. La protection de la position est donc nécessaire pour fournir des garanties d'anonymat rigoureuses.

Protection de la position

L'objectif est de protéger la position réelle de l'utilisateur, indépendamment de toute autre considération, car la position elle-même est considérée comme une information sensible. En fait, c'est l'objectif de la majorité des mécanismes visant la protection des utilisateurs.

Le défi est de fournir des solutions plus simples, sans compromettre l'utilité d'une application. Un exemple qui représente la nécessité de telles solutions peut s'illustrer par l'utilisation d'une application de navigation dans laquelle l'utilisateur a toujours besoin de communiquer sa position exacte pour avoir des itinéraires précis.

Protection du comportement

Cet objectif concerne la protection des trajectoires représentées par un ensemble d'événements inférés par l'observation du mouvement de l'utilisateur. Une trajectoire est sensible quand il révèle un comportement qui par la loi ou selon les préférences personnelles de l'utilisateur devrait être maintenu privé.

Notons que le comportement de l'utilisateur peut-être déduit par observation d'une variété de facteurs contextuels, par exemple le temps, le contexte géographique (*par ex.* réseau routier), la fréquence des visites et la présence d'autres individus proches (*par ex.* endroit fréquenté par plusieurs utilisateurs). Par conséquent, la protection de l'identité et de la position doivent prendre en considération la sémantique d'une position, ou au moins fournir des alternatives qui empêchent la corrélation entre un utilisateur et un lieu ou activité donnée.

L'objectif de déterminer un modèle de confidentialité est donc de pouvoir proposer une métrique efficace qui doit trouver un équilibre entre protéger l'utilisateur et garder l'utilité souhaitée. Nous discuterons dans la prochaine section, le modèle d'utilité que nous adoptons afin de mettre en œuvre notre métrique de confidentialité.

6.2.3 Modèle d'utilité

L'utilité représente l'efficacité des coordonnées résultantes d'un mécanisme dans le processus d'attribution des données géodépendantes précises. La quantification de l'utilité implique la mesure de l'exactitude de l'information obtenue, une propriété estimée selon la distance géographique entre la position exacte et la position résultante du mécanisme [108]. Dans notre cas, l'utilité est mesurée par cette distance, et exprimée comme une marge d'erreur associée aux coordonnées calculées. Nous définissons l'utilité d'un mécanisme $U(M)$ pour une position réelle l et une position observée o du même mécanisme M par :

$$U(M) = ||o - l|| \tag{6.4}$$

En termes de probabilités, cela revient tout simplement à estimer $Pr(l|o)$ vu que la valeur calculée dans cette probabilité reflète la ressemblance entre la position réelle l et la position observée o.

Afin d'assurer le maximum d'utilité, un mécanisme doit minimiser la distance entre les deux positions. Trouver donc l'équilibre confidentialité/utilité est un des majeurs défis pendant la mise en œuvre d'un LPPM.

La définition de chacun des modèles d'attaquant, de confidentialité, et d'utilité sert à définir les bases de notre métrique de confidentialité. Ainsi, leurs propriétés servent à attester l'efficacité de la métrique. Nous discutons dans la section suivante les principaux composants et définitions de δ-*fuzziness*.

6.3 Métrique floue de confidentialité

En nous inspirant de l'outil d'évaluation défini par Shokri *et al.* [108], nous définissions les méthodes d'évaluation pour δ-*fuzziness*. En partant des modèles définis dans la section précédente, nous définissons δ-*fuzziness* en fonction du modèle d'attaquant et en respectant le modèle d'utilité. Nous considérons le modèle d'attaquant, car c'est l'entité visée par un LPPM.

Le modèle d'attaquant comme point de départ

Nous supposons que l'adversaire connait les fonctions d'anonymisation utilisées par un LPPM. Il peut également avoir accès à certaines données de localisation (bruyantes ou incomplètes) des utilisateurs, et à d'autres informations publiques sur les lieux visités par chaque utilisateur, tels que son domicile et son lieu de travail. À partir de ces informations, l'adversaire peut former un *profil de mobilité*, dénoté P_u pour chaque utilisateur

$u \in U$. L'ensemble U décrit l'ensemble des utilisateurs d'un LBS donné.

Tel que discuté dans la section 6.2.1, le modèle de l'adversaire décrit deux aspects majeurs qui sont les connaissances antérieures et les types d'attaques utilisés. Ainsi, δ-*fuzziness* assure que le mécanisme résiste devant les attaques d'un adversaire, peu importe ses connaissances antérieures. En outre, δ-*fuzziness* vise également à évaluer la complaisance d'un LPPM aux exigences de confidentialité, et aux garanties concernant l'utilité. Nous discutons dans la suite les deux méthodes d'évaluation des connaissances antérieures et des attaques d'un adversaire, ainsi que leur relation avec les deux modèles de confidentialité et d'utilité.

La protection de la localisation implique la minimisation de la probabilité de distinction estimée en rendant la position réelle indiscernable parmi les autres positions dans une zone géographique prédéfinie. Afin d'y parvenir, nous définissons la métrique de confidentialité δ-*fuzziness* où la confidentialité est mesurée en se basant sur l'hypothèse suivante : le risque d'identification associé à une position doit être égal ou inférieur à une valeur prédéfinie δ.

En raison du concept collaboratif de *Deloc* nous n'adoptons pas une métrique de confidentialité existante (*par ex.* l'anonymat, la confidentialité différentielle, etc.). En d'autres termes, les métriques existantes sont conçues pour mesurer la confidentialité dans des LPPM qui transforment les coordonnées géographiques par des opérations computationnelles. Dans notre cas, δ-*fuzziness* assure une meilleure quantification dans les mécanismes collaboratifs, ainsi que dans les mécanismes computationnels existants.

Algèbre floue comme base de δ-*fuzziness*

Nous supposons que toute requête géodépendante contient les coordonnées géographiques, le profil d'utilisateur, et le contenu de la requête (*par ex.* itinéraire du point A au point B). Ainsi, nous définissons un ensemble flou $(\mathscr{L}, m)$, où $\mathscr{L}$ contient les requêtes observées, et $m : \mathscr{L} \rightarrow [0,1]$ la fonction d'appartenance correspondante. La fonction d'appartenance définit le niveau de risque associé à l'inclusion de l'élément l dans $\mathscr{L}$.

Le niveau de risque dans ce contexte reflète la probabilité que l'adversaire soit capable d'identifier un élément $l_u \in \mathscr{L}$ en tant que position réelle l_u de l'utilisateur u à l'instant t. Ainsi, en partant des équations (6.1), (6.2), et (6.3) nous définissons cette valeur par le calcul de la probabilité suivante :

$$Pr\{l_u(t)|o_u, C_u, R_u\} \tag{6.5}$$

Tels que o_u représente la position observée de l'utilisateur u, C_u est l'ensemble des positions observées constituant la connaissance antérieure, et R_u représente les informations contextuelles relatives à la position observée o_u.

En partant de cette équation (6.5), nous définissons la fonction d'appartenance d'un élément $l \in \mathscr{L}$, elle sert à déterminer le risque d'identification pour chaque position observée. Tous les éléments dans $\mathscr{L}$ ont au plus le degré δ qui indique la valeur de risque maximum autorisé dans l'ensemble $\mathscr{L}$. Autrement dit, nous cherchons à identifier l'effet de la divulgation d'une requête donnée sur la confidentialité du demandeur, une valeur qui dépend essentiellement de la similarité qu'un adversaire peut percevoir entre la position observée o_u et la position réelle l_u en possédant C_u et R_u.

La raison derrière notre choix de l'algèbre flou réside dans deux points principaux, à savoir :

- La considération des connaissances antérieures de l'adversaire nécessite un mécanisme de comparaison qui prend en compte toutes les requêtes précédentes de l'utilisateur.
- Le concept de délégation implique la comparaison du contenu de la requête de l'utilisateur avec celles des collaborateurs.

De ce fait, la fonction d'appartenance de l'algèbre floue est utilisée pour exprimer ces deux notions de comparaison de la probabilité d'identification. Nous discutons dans les sections qui suivent les étapes de calcul de la fonction d'appartenance en partant de la notion de *similarité* entre les éléments d'un ensemble flou.

Évaluation de la similarité inter-utilisateur

Nous définissons une fonction de similarité qui estime la ressemblance entre deux profils utilisateurs. Dans notre cas, ce qui nous intéresse est la similarité en termes de données identificatrices, nous évaluons la ressemblance exacte entre les variables d'un profil utilisateur en la représentant sous forme de vecteurs. La comparaison se fait une à une de sorte que chaque profil divulgué est comparé uniquement avec le profil du demandeur. Par conséquent, pour un ensemble de variables X représentant le profil P du demandeur, nous définissons une fonction d'encodage $Enc(x,x')$ d'une variable x' appartenant à l'ensemble de variables X' du profil P' en cours de comparaison de la façon la suivante :

$$Enc(x,x') = \begin{cases} 1, & \text{si } x' \equiv x \\ 0, & \text{sinon.} \end{cases} \tag{6.6}$$

Chaque valeur $x' \in X'$ représente chaque profil sous forme de vecteur. À la fin de l'encodage , chaque profil est représenté par une projection $e : P \to Z_+^k$ tel que $e(P) = (p_1, ..., p_k)$. La similarité entre les deux profils P et P' est calculée selon le nombre d'occurrences des variables similaires. L'importance d'une variable dans le calcul de la similarité dépend essentiellement du son type, et de ce nous voulons protéger dans un profil. Nous regroupons les variables identificateurs ou sensibles en nous basant sur 4 groupes :

- **Les identificateurs explicites (*Explicit identifiers*) :** regroupe les attributs permettant d'identifier directement une personne (*par ex.* nom, numéro de téléphone, courriel, numéro d'assurance sociale).

- **Les quasi-identificateurs (*Quasi-identifiers*) :** regroupe les attributs permettant d'obtenir des informations précises sur une personne sans pouvoir la reconnaître directement (*par ex.* âge, origine). Cependant, la corrélation des données de ce

type peut mener à une identification précise de l'individu.

- **Les attributs sensibles (*Sensitive attributes*) :** regroupe les informations sensibles sur une personne (*par ex.* état de santé, informations bancaires).
- **Les attributs non-sensibles (*Non-sensitive attributes*) :** regroupe l'ensemble des autres attributs (*par ex.* pays de résidence, marque du téléphone).

La définition de ces groupes permet la flexibilité de la fonction d'encodage en fonction de la robustesse souhaitée d'un LPPM, ainsi que de l'efficacité de l'encodage. Par exemple, il se peut que deux individus aient les mêmes ensembles de quasi-identificateurs et d'attributs sensibles où une fonction de similarité qui ne prend pas en considération le regroupement des identificateurs peut donner une similarité élevée.

L'utilisation des groupes est utile également dans le cas où la similarité est élevée dans le groupe des identificateurs explicites, ce qui suggère l'appartenance du profil au même individu (*par ex.* cas de plusieurs comptes Google vérifiés avec le même numéro de téléphone).

En partant de ces 4 groupes, la fonction de similarité prend l'encodage résultant de la fonction 6.6, et estime le niveau de ressemblance pondérée entre deux profils. En d'autres termes, la fonction associe à chaque attribut faisant partie d'un groupe donné un coefficient (1 à 4), où 1 représente les attributs non-sensibles, et 4 les identificateurs explicites. L'estimation se fait par le calcul de la somme des éléments du vecteur représentant un profil D' et la division sur le nombre d'attributs comparés. Plus formellement, pour deux profils P et P', et leur encodage $e(P) = (p_1, ..., p_k)$ et $e(P') = (p'_1, ..., p'_k)$, la similarité entre P et P' est calculée par la fonction de similarité suivante :

$$Sim(P, P') = \frac{\sum_{n=1}^{k} p'_k c_k}{\sum_{n=1}^{k} c_k} \tag{6.7}$$

Tel que k est le nombre d'attributs identificateurs encodés en fonction de la ressemblance pair-à-pair, et de c_k le coefficient associé à l'attribut représenté par p'_k.

Similarité comme attribut de sensibilité

Rappelons que chaque élément dans $\mathscr{L}$ est une requête géodépendante qui contient un profil d'utilisateur ainsi que ses coordonnées géographiques. Nous définissons ainsi la fonction qui évalue la réussite de la première attaque ayant pour but d'identifier la position réelle d'un utilisateur à un instant donné.

$$Sens(U,U') = Sim(P,P') \times Dis(loc,loc') \tag{6.8}$$

$Sens(U,U')$ est la fonction qui estime la sensibilité associée à l'observation de l'élément U' au lieu de U, $Sim(P,P')$ représente la similarité entre le profil observé P' et le vrai profil P, et $Dis(loc,loc')$ est le coefficient de la distance physique entre les coordonnées géographiques dans les deux profils. Ce dernier coefficient est calculé selon le seuil maximum de distance autorisée entre deux positions. Autrement dit, nous supposons une distance maximale, notée σ, et nous calculons le coefficient de la distance en fonction de la distance physique réelle d. Par conséquent, la fonction $Dis(loc,loc')$ est définie comme suit :

$$Dis(loc,loc') = \begin{cases} 0, & \text{si } d(loc,loc') \geq \sigma \\ 1-(d(loc,loc')/\sigma) & \text{sinon.} \end{cases} \tag{6.9}$$

Notons ici que le coefficient retourné par $Dis(loc,loc')$ augmente quand on se rapproche de la position réelle et diminue quand on s'éloigne. La valeur de σ est prédéfinie soit de façon dynamique par le mécanisme de protection, soit par l'utilisateur lui-même. Dans les deux cas, la valeur de σ définie la zone dans laquelle les autres positions sont considérées comme à proximité de la position réelle.

Cette valeur de sensibilité représente la valeur de la probabilité de l'équation 6.5, et est utilisée pour évaluer le degré d'appartenance d'un élément l à $\mathscr{L}$.

Calcul du degré d'appartenance

Pour bien comprendre le principe de la fonction de similarité et de sensibilité, prenons l'exemple de trois utilisateurs u_1, u_2 et u_3, ayant pour profils p_1, p_2 et p_3 respectivement. Chaque profil contient un ensemble de données identificatrices, quasi-identificatrices, et sensibles. Nous calculons la similarité entre u_1 et les deux utilisateurs u_2 et u_3, ainsi que la sensibilité estimée quand u_2 ou u_3 remplacent u_1 dans une requête géodépendante. Le tableau 6.I liste les données associées à chaque utilisateur.

Tableau 6.I – Exemple des attributs contenus dans un profil d'utilisateur

	u_1	u_2	u_3
Position	46.327525, -79.466208	41.816385, -87.623027	46.327444, -79.467265
Attributs identificateurs			
Nom	Marshall H. Jones	Ruth G. Patton	Christopher J. Johnson
Téléphone	769-701-897	773-272-6585	769-701-897
Courriel	MarshallHJones@jourrapide.com	RuthGPatton@dayrep.com	ChristopherJJohnson@teleworm.us
Nom d'utilisateur	Puppere	Wremn1998	tae6yohLai
Attributs quasi-identificateurs			
Date de naissance	30 Mars 1988	01 Mai 1998	01 Janvier 1988
Attributs sensibles			
Mot de passe	Ahdee1och7	Ejahyee5B	Ahdee1och7

Commençons tout d'abord par l'encodage de chacun des profils u_2 et u_3 par rapport à u_1 (équation 6.6). Nous rappelons ici que l'encodage d'un profil se fait en comparant chacun de ses attributs aux attributs d'un profil de référence, dans notre cas u_1. Par conséquent, nous obtenons ces deux vecteurs qui représentent l'encodage des attributs de u_2 et u_3 respectivement.

$$e(P_2) = [0,0,0,0,0,0]$$
$$e(P_3) = [0,1,0,0,0,1]$$

Tel que P_2 et P_3 sont les profils associés aux utilisateurs u_2 et u_3 respectivement. Les valeurs des vecteurs $e(P_2)$ et $e(P_2)$ dépendent de ressemblance avec P_1. Par exemple, les profils des deux utilisateurs u_1 et u_3 contiennent le même numéro de téléphone, cela est donc représenté par la valeur 1 dans la deuxième position du vecteur $e(P_3)$. Nous mentionnons ici que l'encodage de la position géographique ne se fait pas comme dans les autres attributs, mais plutôt par la fonction $Dis(loc, loc')$ définie dans (6.9).

Après l'encodage nous passons au calcul de similarité (équation 6.7), le résultat est une moyenne pondérée qui représente la similarité entre les attributs de u_2 et u_3 avec u_1 en fonction des coefficients de sensibilité. Dans cet exemple, les profils contiennent des attributs identificateurs, quasi-identificateur, et sensibles, avec les coefficients 4, 3, et 2 respectivement. Le résultat de la fonction de similarité pour les deux profils P_2 et P_3 est le suivant :

$$Sim(P_1, P_2) = ((0 \times 4) + (0 \times 4) + (0 \times 4) + (0 \times 4) + (0 \times 3) + (0 \times 2))/21 = 0$$
$$Sim(P_1, P_3) = ((0 \times 4) + (1 \times 4) + (0 \times 4) + (0 \times 4) + (0 \times 3) + (1 \times 2))/21 = 0.286$$

L'étape suivante consiste à estimer la réussite de l'attaque en observant un des utilisateurs u_2 ou u_3 au lieu de u_1. Avant de pouvoir l'estimer, nous calculons le coefficient de la distance pour un seuil de protection σ égal à 500m (équation 6.9). Nous obtenons les valeurs suivantes :

$$\begin{aligned} Dis(loc_{u_1}, loc_{u_2}) &= 0 \\ Dis(loc_{u_1}, loc_{u_3}) &= 1 - (d(loc_{u_1}), (loc_{u_3})/\sigma) \\ &= 1 - (81/500) \\ &= 0.838 \end{aligned}$$

Par conséquent, les valeurs de sensibilité associées à chacun des utilisateurs sont :

$$Sens(P_1, P_2) = 0$$

$$Sens(P_1, P_3) = 0.286 \times 0.838 = 0.2396$$

En conclusion, la notion des ensembles flous définit la relation entre les positions contenues dans plusieurs requêtes, ou celles contenues dans la connaissance antérieure que l'adversaire possède. Par définition, un ensemble flou exige une fonction d'appartenance qui estime le niveau d'inclusion de chaque élément. Dans notre cas, la fonction d'appartenance détermine le niveau de risque associé à la divulgation d'un élément, étant donné la condition d'inclusion qui suppose qu'un élément ayant une valeur d'appartenance 1 signifie qu'il est extrêmement risqué de le divulguer. L'estimation de la valeur d'appartenance est basée sur la probabilité de distinction d'un élément par l'adversaire. Cette dernière probabilité est calculée en fonction de la sensibilité d'un élément $l \in \mathscr{L}$.

6.4 Utilisation et validation

Nous expliquons dans cette section les cas d'utilisation ainsi que la validation de δ-*fuzziness* en discutant de son efficacité, et en la comparant aux deux des principales métriques existantes, notamment *k-anonymity* [114] et *differential privacy* [36, 37] (confidentialité différentielle).

La métrique δ-*fuzziness* est une mesure de confidentialité adaptée au contexte des mécanismes collaboratifs. Dans le cas de *Deloc*, elles servent à attester son efficacité et son utilité.

Nous exécutons deux modèles d'attaque face aux résultats jugés confidentiels selon δ-*fuzziness*. L'utilisation d'une telle approche permet de mesurer l'efficacité de la métrique. En partant des modèles d'attaques définies dans la section (6.2.1.2), nous définissons deux attaques selon les deux scénarios suivants :

- Pour un utilisateur u et un instant t, quelle est la position réelle de u à l'instant t ?
- Pour deux utilisateurs u et v, quel est le nombre d'instants où u et v se trouvent dans la même position (ou région) ? Cela revient au nombre de rencontres de u et v.

L'efficacité de la première attaque, dénotée *LOCA*, est évaluée en fonction de la distance entre la position estimée par l'attaque et la position réelle de l'utilisateur. Ceci dit, un attaquant doit aussi identifier la relation entre une position et un utilisateur. Autrement dit, l'attaque commence par identifier les positions susceptibles d'appartenir au même utilisateur, puis tente de cibler celles qui peuvent être sa position actuelle. L'efficacité de l'attaque LP_{LOCA} est mesurée par l'équation suivante :

$$LP_{LOCA}(u,t) = Pr\{l_u(t) = o_u(t)|O_u, P_u\} \tag{6.10}$$

Telle que o_u est la position observée de l'utilisateur, O_u l'ensemble de toutes les positions observées dans les instants $\{t-1, t-2, \cdots, t-n\}$, et P_u le profil de l'utilisateur u observé par l'adversaire.

Quant à la deuxième attaque, dénotée *RENA*, son efficacité est évaluée en fonction de rencontres entre deux utilisateurs u et v. Cela revient à estimer la distance entre les positions des deux utilisateurs dans un instant donnée t. Ainsi, la mesure de l'efficacité de LP_{RENA} est mesurée par l'équation suivante :

$$LP_{RENA}(u,v,t) = Pr\{l_u(t) = l_v(t)|O_u, O_v, P_u, P_v\} \tag{6.11}$$

Telles que o_u et o_v sont les positions observées des utilisateurs u et v, O_u et O_v les deux ensembles de toutes les positions observées dans les instants $\{t-1, t-2, \cdots, t-n\}$, et P_u et P_v les profils observés des utilisateurs u et v.

Évaluation de δ-*fuzziness* par attaque

Pendant l'évaluation de δ-*fuzziness*, nous évaluons la réussite de l'attaque sur l'ensemble des données collectées par l'exécution de requêtes simulées. L'environnement d'application de δ-*fuzziness* dans *Deloc* est un environnement simulé avec les mêmes paramètres utilisés dans la section (4.2). En résumé, nous utilisons l'ensemble de données de BrightKite [20] qui contient 58228 nœuds avec 4491143 ($\approx 4,5M$) coordonnées géographiques. Chaque tuple de cet ensemble est composé de l'identifiant de l'utilisateur, des coordonnées de sa position et de l'horodatage respectif. Nous préparons par la suite le nouveau jeu de données en limitant l'ensemble original aux nœuds ayant plus de 70 tuples, de façon à ce que notre nouvel ensemble de données soit composé de 9963 nœuds avec 3956113 ($\approx 4M$) coordonnées géographiques.

Le déroulement de la simulation se fait de la même manière que dans (4.2) avec la particularité de garder trace de toute requête ou information échangée pendant le processus de délégation afin de pouvoir mesurer l'efficacité de *Deloc* en utilisant δ-*fuzziness*. En exécutant la même simulation décrite dans (4.2) pour une durée plus longue, nous avons réussi à obtenir 122647 requêtes géodépendantes avec leurs chemins de routage et leurs utilisateurs (représenté par une unité d'exécution dans l'environnement simulé) participants.

Les mesures utilisées pour évaluer la réussite des attaques *LOCA* et *RENA* prennent en considération les points suivants :

- La valeur δ dénotant le seuil de niveau d'appartenance d'un élément (c'est à dire, la sensibilité du profil de participant).
- Le nombre des délégations effectuées.
- La taille σ de la zone sensible.
- Le nombre de requêtes échangées à partir de la même position

L'environnement de simulation permet un réglage fin de ses paramètres, ce qui nous aide à capturer des mesures précises. Les figures 6.4 et 6.5 illustrent les "boîtes à moustaches"[1] correspondantes aux moyennes de probabilité de réussite de chacune des attaques *LOCA* et *RENA* en fonction de type d'attaque et des paramètres que nous venons de mentionner.

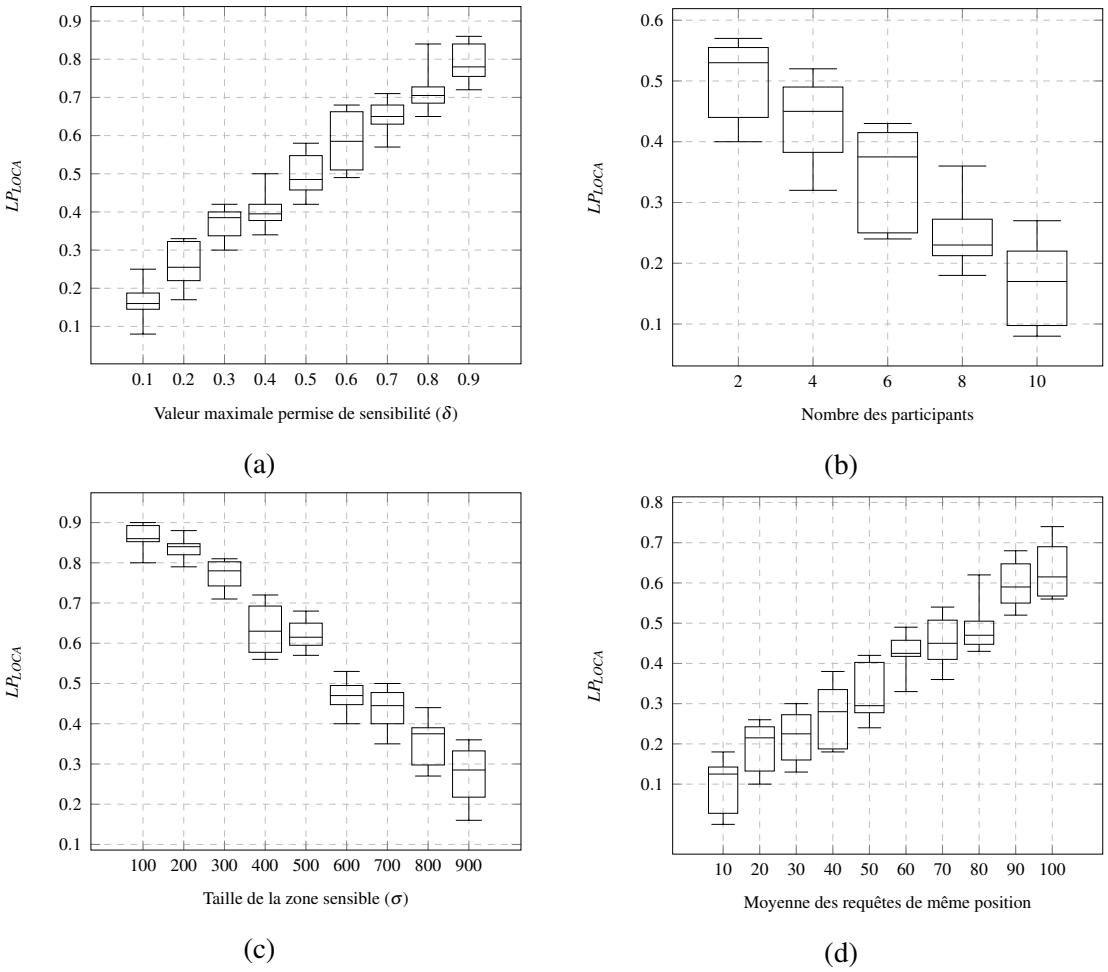


Figure 6.4 – Probabilité de réussite de l'attaque *LOCA*

La figure 6.4 illustre la probabilité de réussite de l'attaque *LOCA*, exprimée par la valeur LP_{LOCA}. Chaque boîte représente les valeurs correspondantes aux nombres totaux

[1]Une boîte à moustache est un type de représentations graphiques de données statistiques qui résume des caractéristiques (minimum, quartile inférieur, médian, quartile supérieur, maximum) d'une population dans une seule représentation

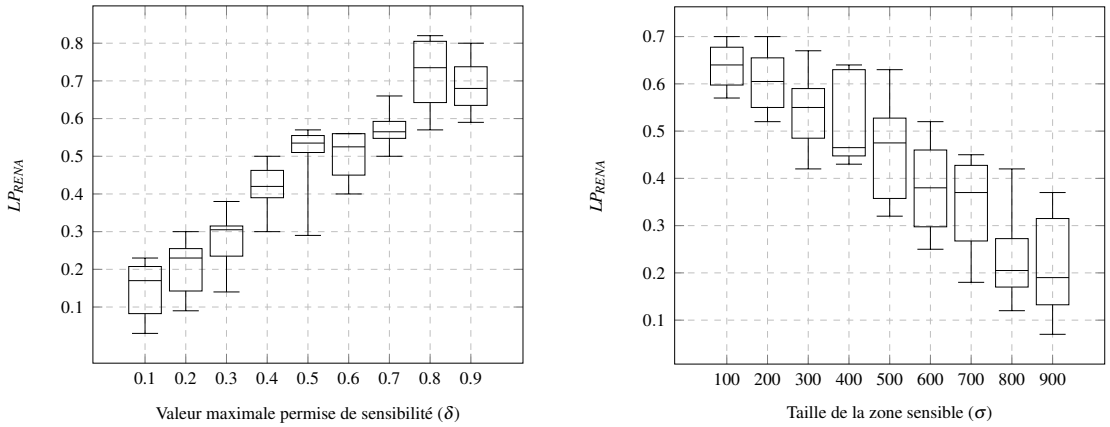
de requêtes obtenues par simulation. Nous remarquons pour la figure 6.4a que la valeur LP_{LOCA} augmente en fonction de la valeur permise de sensibilité. Cela atteste que le fait d'exiger une valeur δ plus petite implique plus de protection contre les attaques de détermination de positions.

Pour les figures 6.4b, 6.4c, et 6.4d, nous fixons la valeur δ à 0.5, et nous explorons l'effet des autres paramètres sur la réussite de l'attaque *LOCA*. Ainsi, nous constatons que le nombre de participants ait un effet sur la valeur LP_{LOCA} qui diminue quand le nombre de participants au processus de délégation augmente. Cela peut être justifié par le nombre de profils différents observés par l'adversaire, chaque profil rajouté implique un bruit supplémentaire à l'information de base, ce qui diminue la probabilité de la distinction de cette dernière.

La figure 6.4c illustre l'effet de la taille de la zone sensible σ sur la valeur LP_{LOCA}. À titre de rappel, la valeur σ est la taille de la région utilisée pour la définition de la similarité entre deux profils pendant la composition de l'ensemble flou. Une valeur σ de 300 veut dire que pour un utilisateur donné u, les utilisateurs à moins de 300m de proximité sont considérés similaires (selon leurs distances de u). Par conséquent, nous constatons que l'augmentation de la taille de cette zone influence négativement l'efficacité de l'attaque *LOCA*. Par exemple, la médiane de la valeur LP_{LOCA} passe de 0.86 quand la taille de la zone est de 100m à 0.285 lorsqu'elle est de 900m.

Pour la figure 6.4d, nous illustrons l'effet engendré par l'envoi de plusieurs requêtes géodépendantes de la même position sur la valeur LP_{LOCA}. Nous remarquons que l'utilisation du même LBS à partir de même endroit consécutivement augmente la probabilité de deviner la position réelle même en utilisant la délégation. L'envoi de 10 requêtes à partir de la même position s'accorde avec une médiane de la valeur LP_{LOCA} égale à 0.125 contre 0.615 pour l'envoi de 100 requêtes. Cela peut être justifié par le filtrage de bruit ou de données répétitives, une technique qui identifie les vraies données si ces dernières se représentent dans plusieurs ensembles bruités [73]. Autrement dit, l'adversaire essaye d'identifier la valeur inchangée (demandeur initial) dans plusieurs ensembles de

variables (données bruitées).



(a) (b)

Figure 6.5 – Probabilité de réussite de l'attaque *RENA*

Pour l'évaluation de la réussite de l'attaque *RENA*, nous mesurons l'effet de la valeur permise de sensibilité (figure 6.5a) et l'effet de la zone sensible (figure 6.5b). Nous omettons le nombre de participants et le nombre des requêtes de la même position vu que nous mesurons la probabilité de réussite en fonction de rencontres des utilisateurs pair par pair. La figure 6.5a illustre la valeur LP_{RENA} pour deux utilisateurs qui ont les deux une valeur maximale permise de sensibilité égale à une valeur donnée. Nous remarquons que la valeur de LP_{RENA} augmente si la valeur δ augmente. Nous constatons à partir de la figure 6.5b que LP_{RENA} décrémente avec l'incrémentation de σ.

Dans la deuxième étape de la validation de δ-*fuzziness* nous exécutons les mêmes attaques *LOCA* et *RENA* dans le cas de requêtes géodépendantes protégées par des mécanismes respectant *k-anonymity* et *differential privacy*. Pour le cas de *k-anonymity*, nous utilisons la méthode décrite dans un travail de Niu *et al.* dans laquelle les auteurs utilisent du bruit sous forme de positions fictives afin d'assurer une protection *k-anonyme* [87]. Concernant la confidentialité différentielle, nous utilisons la méthode présentée dans [7], une méthode largement utilisée pour garantir la confidentialité différentielle dans le contexte des LBS.

Nous exécutons chacun des mécanismes décrits dans [87] et [7] dans le même environnement de simulation et avec le même jeu de données utilisé pendant la validation de δ-*fuzziness*. Ainsi, pour une exécution de la même période du temps, nous avons obtenu 121473 requêtes pour le mécanisme basé sur *k-anonymity*, et 123374 requêtes pour le mécanisme basé sur la confidentialité différentielle. La figure 6.6 illustre la probabilité de réussite des attaques *LOCA* et *RENA* dans le cas de chaque mécanisme.

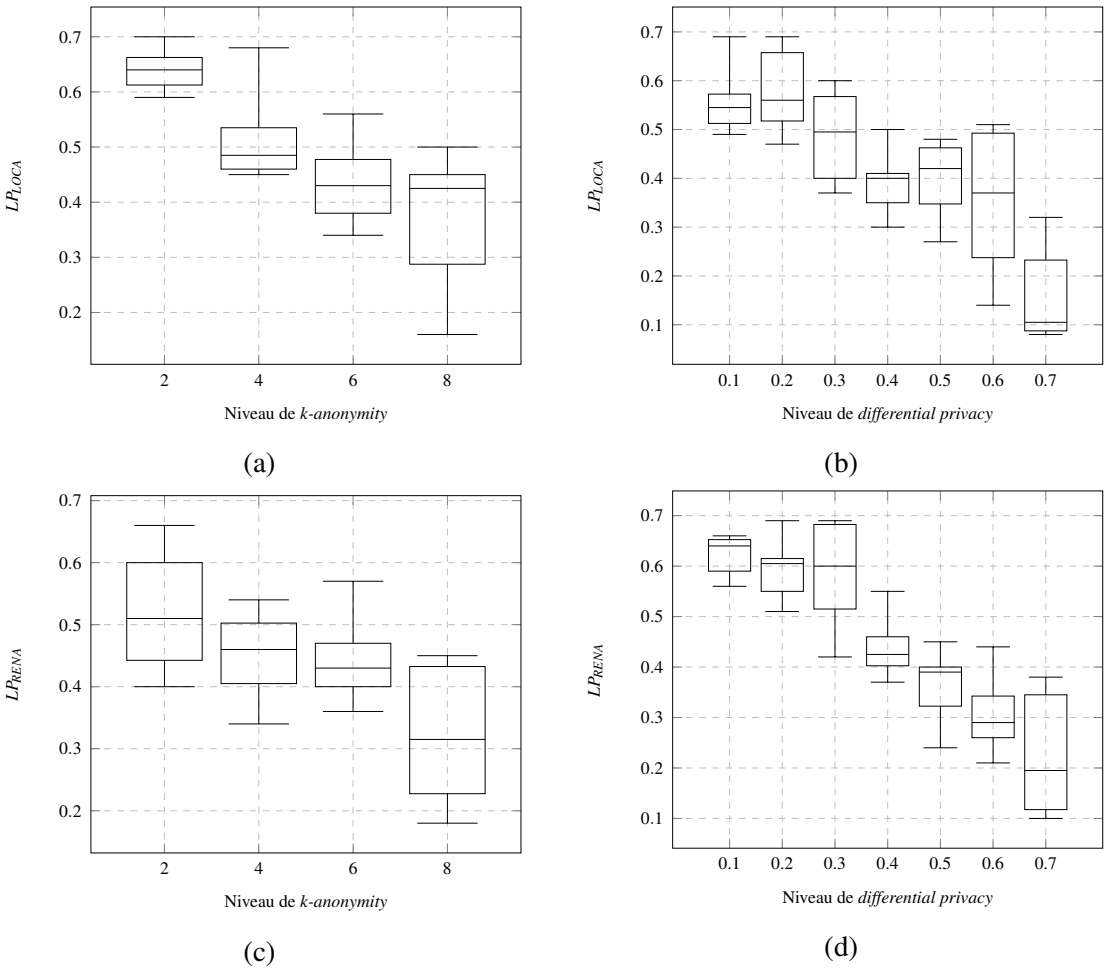


Figure 6.6 – Comparaison de *k-anonymity* et *differential privacy*

Tel que nous pouvons observer sur la figure 6.6, le niveau de protection, exprimée par la probabilité de réussite des *LOCA* et *RENA*, augmente selon le niveau de protection du mécanisme en question. Dans la figure 6.6a la réussite de l'attaque *LOCA* dépend du

nombre de clusters utilisé pour *k-anonymity* (section 6.1). Cependant, même en exigeant une valeur $k = 8$ la probabilité de réussite de *LOCA* est significative. Pour la figure 6.6b qui illustre la réussite de *LOCA* face à la confidentialité différentielle, nous remarquons que la probabilité de réussite diminue si la valeur de confidentialité augmente. La probabilité de réussite ne rapproche pas à 0.1 sauf si la valeur de confidentialité est égale à 0.7. Pour atteindre cette dernière valeur, le mécanisme nécessite des coûts computationnels élevés [39], une condition qui représente un obstacle dans un environnement mobile.

Dans le cas de l'attaque *RENA* nous remarquons que la relation entre sa réussite et le niveau de confidentialité est similaire à celle de l'attaque *LOCA*. La différence dans les valeurs de probabilités peut être justifiée par la complexité de *RENA* par rapport à *LOCA*.

Les métriques existantes ne peuvent pas donner une estimation précise dans le cas des mécanismes collaboratifs. En effet, un mécanisme collaboratif qui satisfait les exigences d'une métrique qui n'est pas conçue pour son type de mécanisme ne peut assurer les exigences estimées de protection. De plus, même si ces exigences sont atteintes, le coût à payer en matière d'utilité est souvent élevé, comme le cas de la confidentialité différentielle qui rajoute du bruit à la vraie position pour la protéger. L'autre point est la limite de la capacité computationnelle dans les environnements mobiles, ainsi que la limite du temps allouée à l'exécution de chaque requête. Les environnements mobiles exigent des mécanismes rapides et efficaces, et si ces mécanismes suivent des règles de métriques coûteuses, leur influence sur le flux d'exécutions sera rapidement perceptible, et les exigences de base seront déjouées.

En partant du travail de Mendes *et al.* [83], nous listons les principaux avantages et inconvénients des métriques existantes ainsi que la position de δ-*fuzziness* face à leurs caractéristiques dans le tableau 6.II.

Tableau 6.II – Comparaison des métriques existantes avec δ-*fuzziness*

Métrique	Avantages	Inconvénients
Métriques computationnelles		
k-anonymity	- Simplicité de la mise en œuvre. - Nombre d'implémentation existante.	- Suppose que chaque utilisateur possède seulement un seule tuple de données. - Ne considère pas les attributs sensibles.
l-diversity	- Simplicité de la mise en œuvre. - Considère les attributs sensibles.	- Ne considère pas la distribution des attributs sensibles.
t-closeness	- Simplicité de la mise en œuvre. - Considère la distribution des attributs sensibles.	- La corrélation entre les types d'attributs diminue quand le niveau de confidentialité augmente.
Métriques probabilistes		
Basée sur les entropies	- Simplicité de la mise en œuvre.	- Limite l'utilité de service, car elle nécessite des mécanismes qui interdisent l'accès aux services en ligne.
Basée sur la marge d'erreur	- Simplicité de la mise en œuvre.	- Limite la précision, car elle nécessite des mécanismes qui maximisent la différence entre les données
Confidentialité Différentielle	- Assure une protection rigoureuse. - Adaptable aux plusieurs contextes.	- Absence de recommandation sur la valeur optimale du seuil de protection. - Augmenter le seuil engendre une perte de précision à cause de la quantité du bruit rajouté.
δ-*fuzziness*	**- Assure une protection rigoureuse. - Conçu pour des mécanismes collaboratifs qui ne nécessitent pas de transformations géographiques.**	**- Vise les services géodépendants seulement. - Son adaptation aux autres contextes nécessite une validation supplémentaire.**

6.5 Conclusion

La métrique de confidentialité est le seul moyen d'attester l'efficacité d'un LPPM. Elle sert à quantifier des valeurs, des fois abstraites, pour permettre l'estimation de la protection fournie. Dans le contexte de cette recherche, nous passons par les principales métriques existantes dans la littérature qui semblent insuffisantes dans le cas des mécanismes collaboratifs. À titre de rappel, un mécanisme collaboratif implique la collaboration de plusieurs utilisateurs dans l'objectif de fournir une protection pour chacun d'eux, et sa particularité réside dans la non-nécessité de transformation géographique.

Les métriques existantes estiment le niveau de confidentialité d'un mécanisme en comparant les données originales et celles issues du mécanisme, souvent altérées ou remplacées (section 2.3.1). Dans notre cas, nous cherchons à mettre en oeuvre une métrique qui peut estimer la confidentialité même s'il n'existe aucune transformation géographique.

Nous avons proposé δ-*fuzziness*, une métrique inspirée de l'algèbre floue et basée sur la théorie des probabilités, mesurant la protection fournie par un mécanisme collaboratif. L'idée est d'évaluer la confidentialité en termes de fiabilité de collaborateurs au lieu de se concentrer sur les coordonnées géographiques. Nous avons également démontré l'efficacité de δ-*fuzziness* en l'analysant de façon empirique, et en la comparant à deux des métriques existantes les plus utilisées, notamment *k-anonymity* et *differential privacy*. Nous avons discuté l'insuffisance de ces métriques, qui représentent les deux catégories de métriques computationnelles et probabilistes jusqu'au jour.

Notre métrique est efficace et est adaptée pour les mécanismes collaboratifs, pouvant être utilisée dans différents contextes tels que le partage confidentiel de données ou autres contextes exigeant une protection rigoureuse de la confidentialité. L'autre particularité de δ-*fuzziness* réside dans la manière d'évaluer un mécanisme de protection. Les données issues ou utilisées par un mécanisme sont décortiquées avant d'être utilisées pour l'évaluation. En effet, dans le cas des LPPM, δ-*fuzziness* sépare les données de géolocalisation des autres données, ces derniers sont ensuite divisés en sous-ensembles selon leur sensibilité. Cette pratique permet à δ-*fuzziness* d'assurer une évaluation précise, que les autres métriques échouent à offrir dans le contexte de mécanismes collaboratifs.

CHAPITRE 7

MISE EN ŒUVRE ET APPLICATIONS

L'implémentation de *Deloc* comprend plusieurs étapes et chacune d'elle représente un composant dans la plateforme. L'étude de cas que nous effectuons concerne l'utilisation d'une application sur un dispositif mobile Android où l'utilisateur exploite tous les services d'un LBS en passant par *Deloc*. Nous décrivons dans la suite de ce chapitre les différentes étapes de l'implémentation, ainsi que l'efficacité de cette dernière en matière d'utilité et confidentialité.

7.1 Plateforme de virtualisation

Afin de permettre à la plateforme de gérer le flux de données dans un dispositif mobile, il est nécessaire de contourner certaines restrictions imposées par les systèmes mobiles actuels. Par mesure de sécurité, un dispositif mobile ne permet pas l'altération d'un flux de données entrant ou sortant, ce qui n'autorise pas *Deloc* à intercepter une requête, à la modifier, ou éventuellement à la déléguer. Pour remédier à ce problème sans devoir modifier le comportement du dispositif mobile, nous utilisons une plateforme de virtualisation qui permet d'avoir un espace dédié dans le disque du dispositif mobile et qui est utilisé pour installer et gérer des applications.

La plateforme de virtualisation que nous réalisons est inspirée du concept de clonage des applications mobiles. La plateforme est conçue pour Android, elle permet de créer un espace virtuel pour pouvoir installer et lancer des applications indépendamment des applications installées directement sur le dispositif mobile. Cela peut être atteint sans avoir besoin d'un accès super-utilisateur ou de violer les conditions d'utilisation d'Android.

La plateforme permet d'intercepter le flux de données en utilisant des couches de liaisons personnalisées, ce qui permet de gérer les données des applications virtualisées.

La géolocalisation est gérée dans un dispositif mobile Android par les services de

gestion de géolocalisation du système. Dans le cas de virtualisation, une couche supplémentaire est rajoutée pour permettre l'utilisation des services du système ou l'implémentation de ses propres services. La figure 7.1 illustre le principe du fonctionnement de l'espace virtuel dans le cas de services de géolocalisation. Une application qui demande l'utilisation de la géolocalisation ne peut accéder directement aux services du système, car elle doit passer par le biais des services virtuels qui se chargent de la récupérer.

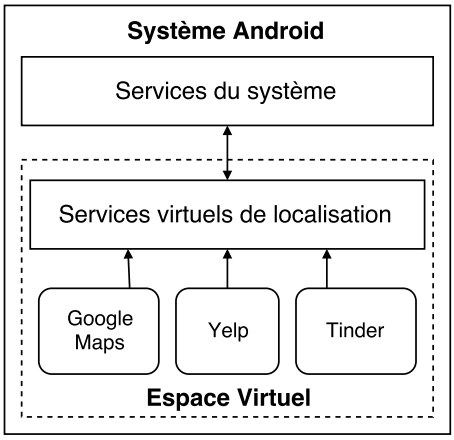


Figure 7.1 – Exemple de conception générale de l'espace virtuel

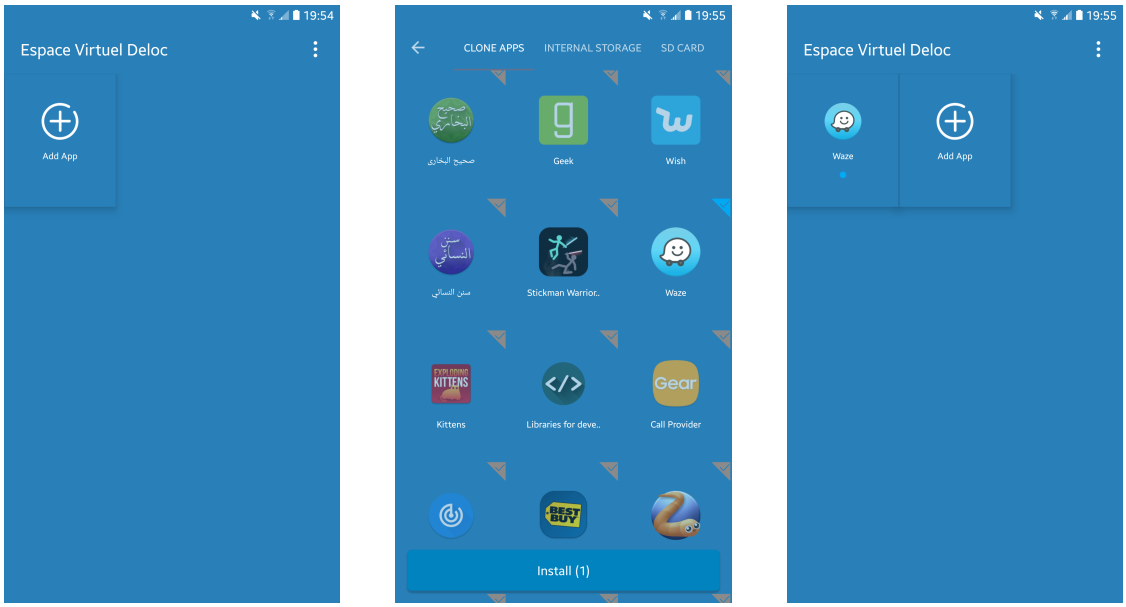
Dans le cas de *Deloc*, deux caractéristiques principales de l'espace virtuel sont nécessaires ; la première concerne la façon dont les coordonnées géographiques sont gérées pour forcer la transformation au moment de la requête, tandis que la deuxième concerne la destination de flux de données pour permettre la délégation des requêtes géodépendantes. Une fois que l'application récupère les coordonnées et prépare la requête à envoyer, *Deloc* intercepte cette dernière et la transmet au réseau de délégation. Ces deux opérations assurent le contrôle de flux de données tout en étant invisibles à l'utilisateur. Le listing 7.1 montre une des classes de l'espace virtuel qui gère les services de géolocalisation, représentée par un observateur du *changement de position* dans la plateforme de virtualisation.

Listing 7.1 – Un observateur du changement de position dans la plateforme de virtualisation

```
package mirror.android.location;

import android.location.Location;
import android.os.IBinder;
import android.os.IInterface;

import mirror.MethodParams;
import mirror.RefClass;
import mirror.RefMethod;
import mirror.RefStaticMethod;

public class ILocationListener {
    public static Class<?> TYPE =
        RefClass.load(ILocationListener.class,
        "android.location.ILocationListener");
    public static class Stub {
        public static Class<?> TYPE = RefClass.load(Stub.class,
            "android.location.ILocationListener.Stub");
        @MethodParams({IBinder.class})
        public static RefStaticMethod<IInterface> asInterface;
    }
    @MethodParams(Location.class)
    public static RefMethod<Void> onLocationChanged;
}
```

Comme illustré dans le listing 7.1, la plateforme de virtualisation implémente un observateur qui existe dans les services par défaut d'Android. L'objectif est de pouvoir manipuler les requêtes géodépendantes et les coordonnées de l'utilisateur. Notons ici que l'utilisation d'une plateforme de virtualisation ne détourne aucune des règles d'uti-

lisation d'Android, et ne présente ni une vulnérabilité ni un problème de sécurité. Le principe est de créer un espace indépendant au sein du dispositif mobile concerné sans avoir l'intention de modifier le système. La figure 7.2 illustre un exemple d'installation de l'application *Waze* dans la plateforme de virtualisation.



(a) (b) (c)

Figure 7.2 – Exemple d'installation d'application dans la plateforme de virtualisation

Tel qu'illustré, le principe est de cloner une application existante (7.2b) afin de créer une nouvelle copie (7.2c) qui peut être manipulée par la plateforme de virtualisation. Les applications installées dans l'espace virtuel respectent les règles d'utilisation du système Android ; ainsi, aucune d'elles ne peut recevoir des privilèges de super-utilisateur, ou contrôler les données échangées par d'autres applications.

Une autre particularité de la plateforme de virtualisation est de pouvoir générer pour chaque espace virtualisé une *liste de contacts*, un *historique d'appels*, ou un *espace de stockage*. La plateforme renvoie les données fictives générées, ou accorde directement l'accès au contenu du dispositif mobile. L'utilité de cette opération réside dans la possibilité de restreindre tout accès indésirable pour un ensemble de données, et pas seulement aux données de géolocalisation.

La plateforme de virtualisation permet de fournir une base sur laquelle tous les composants de *Deloc* sont implémentés. Les composants de *Deloc* fonctionnent en parallèle et échangent de l'information avec la plateforme de virtualisation afin de permettre la génération de positions jumelles, la délégation des requêtes géodépendantes, et l'évaluation de risques et de confiance. Nous discutons dans la suite l'implémentation proposée de chacun des composants de *Deloc*, ainsi que leurs interactions avec la plateforme de virtualisation.

7.2 Implémentation des composants

Selon l'architecture de *Deloc*, chacun des composants est conçu de façon à interagir avec au moins un autre composant qui fait partie ou pas du même processus. Le terme processus réfère dans ce contexte aux deux opérations majeures qui composent *Deloc*, qui sont la délégation et la quantification.

7.2.1 Processus de délégation

La partie de *Deloc* qui compose les fonctionnalités de ce processus est celle représentée par les composants illustrés dans la figure 4.1. À titre de rappel, les composants principaux sont la position jumelle, le répertoire inconscient, le chemin du routage binœud et le cache géodépendant.

7.2.1.1 Position jumelle

L'implémentation du mécanisme de la position jumelle implique des algorithmes de génération effectués au niveau du dispositif. Le déroulement des opérations se fait selon les étapes suivantes :

1. L'utilisateur lance l'application à partir de l'espace virtuel pour envoyer une requête géodépendante.

2. La requête est ensuite interceptée par la plateforme de virtualisation pour remplacer la position réelle (issue des services du système d'exploitation) par la position jumelle.

3. La plateforme de virtualisation fait appel aux composants nécessaires pour le calcul de la position jumelle, et transmet la requête aux composants responsables de la délégation après substitution de la position.

Notons ici que la plateforme de virtualisation a un rôle principal dans le calcul de la position jumelle. Si nous n'interceptons pas la requête, nous ne serons pas en mesure de modifier son contenu.

Comme décrit précédemment (section 4.1.2), le rôle principal des positions jumelles est de fournir une première protection à la position réelle de l'utilisateur sans diminuer l'utilité ni menacer la confidentialité de ce dernier. C'est pourquoi le calcul des positions jumelles utilise des propriétés contextuelles uniques pour générer des **substituts uniques**. Dans notre cas, les propriétés que nous utilisons sont la *position réelle de l'utilisateur*, le *LBS en cours d'utilisation* et le *contexte de localisation*. Plus précisément, la position jumelle est la position résultante de l'utilisation de la fonction $twin(user\_loc, attr[])$, qui prend comme arguments la position réelle de l'utilisateur et l'ensemble des propriétés contextuelles. La façon que la fonction utilise pour calculer la position jumelle dépend de sa mise en œuvre et des exigences de la zone de confidentialité de l'utilisateur. Une zone de confidentialité réfère à la région géographique la plus grande autour d'un utilisateur, incluant la position jumelle.

Le calcul de la position jumelle compte tenu de la position réelle de l'utilisateur peut être extrait en calculant l'angle de rotation θ_i et la distance len_i. La valeur de rotation définit l'angle ayant l'axe horizontal pour rayon et la position réelle pour sommet. Ensuite, la position jumelle est placée sur le deuxième rayon en fonction de la valeur de la distance de la position réelle. L'algorithme 1 décrit notre implémentation du calcul des deux positions.

Algorithm 1 Calculer la position jumelle

Require: $loc[]$ ▷ La position réelle de l'utilisateur
Require: $attr[]$ ▷ Les attributs contextuels de calcul
Ensure: $twin[]$

1: **if** $loc[] \neq null$ and $attr[] \neq null$ **then**
2: $hexaLat \leftarrow hash.crc32(loc[latitude])$
3: $hexaLon \leftarrow hash.crc32(oc[longitude])$
4: $hexaLbs \leftarrow hash.crc32(attr[id_{lbs}])$
5: $hexaUsr \leftarrow hash.crc32(attr[id_{user}])$

6: $decLat \leftarrow hexaToDecimal(hexaLat)$
7: $decLon \leftarrow hexaToDecimal(hexaLon)$
8: $decLbs \leftarrow hexaToDecimal(hexaLbs)$
9: $decUsr \leftarrow hexaToDecimal(hexaUsr)$

10: $distanceRatio \leftarrow |\cos(decLbs) \times \cos(decUsr)|$
11: $\theta_i \leftarrow \log(decLat) \times \log(decLon)$
12: $len_i \leftarrow attr[privacyZone] \times distanceRatio$
13: **return** $[\theta_i, len_i]$
14: **else**
15: **error** Arguments invalides
16: **end if**

L'algorithme 1 prend les coordonnées de la position réelle de l'utilisateur et les attributs contextuels comme entrées et calcule une valeur de hachage unique pour chaque attribut (lignes 2 à 5). La valeur de hachage est une valeur hexadécimale de 32 bits, calculée à l'aide de la méthode de vérification de redondance cyclique (*Cyclic Redundancy Check* - CRC). Chacun des nombres hexadécimaux est ensuite converti en valeur décimale (lignes 6 à 9), un nombre entier positif ayant une longueur de 8 à 10 chiffres. Le rapport de distance, calculé dans la ligne 10, est délimité par 0 et 1 selon les intervalles *cos* et utilisé pour donner une distance unique len_i située entre la position réelle et les limites de la zone de confidentialité (ligne 12). L'angle de rotation θ_i est limité par $\approx 93^\circ$ dans notre cas, puisque la valeur du logarithme est délimitée par $log(n_{max})$, où n_{max} est la valeur décimale du plus grand nombre hexadécimal de 32 bits (FFFF FFFF). Néanmoins, pour obtenir un angle de rotation limité par 360°, nous pouvons simplement remplacer

la fonction *hash* dans les lignes 2 à 5 par une fonction qui renvoie un hexadécimal de 64 bits. Enfin, l'algorithme renvoie un tableau contenant à la fois θ_i et len_i définissant la position jumelle.

Notre implémentation de l'algorithme proposé est faite en Java sous la plateforme Android. Nous utilisons les méthodes de hachage, de conversion et de calcul mathématiques fournies par l'interface de programmation d'applications Android dans sa version 25 (*Android API level 25*). Notons ici que même si l'interface de programmation d'applications utilisée est celle de la version 25, les méthodes utilisées (hachage CRC32, logarithme, etc.) sont également disponibles dans des interfaces plus anciennes telles que l'interface de la version 9. Cela assure que l'implémentation proposée garantit une compatibilité totale avec les différentes versions existantes du système Android.

Une fois la position réelle remplacée par la position générée, la plateforme de virtualisation retourne la requête à *Deloc* qui initie le processus de délégation en faisant appel au répertoire inconscient pour récupérer les dispositifs actifs. Ces derniers sont ensuite utilisés pour transmettre la requête géodépendante dans le réseau de délégation.

7.2.1.2 Répertoire inconscient

Le répertoire inconscient utilise ses propres mécanismes pour gérer les éléments enregistrés, dont l'addition, la sélection et la suppression. Bien que l'addition d'un élément soit effectuée en insérant simplement les informations correspondantes à l'élément dans la base de données, la suppression est déclenchée par un planificateur qui vérifie en permanence la disponibilité des dispositifs. La sélection à son tour se fait par plusieurs étapes (voir la figure 4.6), dont la plus importante est le calcul des facteurs entiers correspondants à la taille du répertoire. Afin d'atteindre cet objectif, l'algorithme utilisé est basé sur la méthode de crible quadratique auto-initialisant (*self initializing quadratic sieve*)[24], connue pour être l'algorithme le plus efficace pour la factorisation de nombres entiers de moins de 10^{100} [96]. La méthode considère un nombre $a = \gcd(n, x - y)$ comme facteur entier de n si les deux entiers $x, y \in \mathbb{Z}/n\mathbb{Z}$ assurent les deux équations

suivantes :

$$x \equiv \pm y \pmod{n} \quad \text{ET} \quad x^2 \not\equiv y^2 \pmod{n}$$

La méthode que nous utilisons est une version légèrement modifiée de l'ensemble des algorithmes utilisés pour construire la factorisation du crible quadratique décrit dans [8]. L'implémentation est basée sur l'idée des entiers *B-lisse*, et est divisée en 6 étapes. Un entier est *B*-lisse s'il peut être factorisé en utilisant seulement les nombres premiers inférieurs à *B*. Les 6 étapes sont implémentées chacune dans un algorithme indépendant, et la liste suivante résume le rôle de chacun :

1. Choisir le seuil de la valeur *B* qui sera utilisée comme taille de base pour la factorisation.

2. Générer la liste *X* des racines qui sont *B*-lisse.

3. Factoriser les éléments de la liste *X* pour générer les vecteurs correspondants.

4. Résoudre les équations définissant une dépendance linéaire entre les vecteurs générés dans l'étape 3.

5. Sélectionner $x^2 \equiv y^2 \bmod n$ tel que x et y sont deux valeurs choisies aléatoirement.

6. Calculer $\gcd(n, x-y)$, si le facteur est trivial une autre dépendance linéaire de l'étape 4 est résolue avec deux autres valeurs de x et y, sinon l'algorithme est terminé.

En résultat, l'ensemble des algorithmes de notre implémentation Java peut récupérer tous les facteurs d'un répertoire contenant toute la population actuelle du globe (7,55 milliards [31]) en ≈ 11 secondes (résultats sur un ordinateur portable équipé d'un processeur Intel à double cœur *Core i7-5600u*).

La prochaine étape consiste à transmettre la demande à la paire de dispositifs actifs et à lancer la délégation. Le point clé de cette étape est d'assurer la transmission des

requêtes dans le réseau de foule de délégation. À cette fin, chaque requête contient les adresses des dispositifs impliqués dans la transmission, ainsi que l'adresse du dispositif du support. Une délégation réussie se termine par la récupération des résultats géodépendants à partir d'un LBS.

7.2.1.3 Chemin du routage bi-nœud

Le chemin du routage bi-nœud est la route prise par une requête géodépendante pour arriver à un LBS, il se compose des différents dispositifs connectés et il se change à l'initialisation de chaque nouveau processus de délégation. Le transfert des requêtes entre les dispositifs se fait par le biais d'une connexion Internet, impliquant l'utilisation du protocole HTTP. Chaque requête est cryptée avant sa transmission avec une clé que seul le destinataire peut utiliser. L'idée est d'utiliser un mécanisme efficace d'un chiffrement de bout en bout qui protège le contenu d'une requête dans le cas où elle serait interceptée avant son arrivée à destination.

Il existe plusieurs travaux qui proposent des techniques de chiffrement de bout à bout. Dans notre cas, nous utilisons une méthode basée sur l'échange de clés *Diffie-Hellman* [17]. En résumé, un chiffrement de bout à bout basé sur la méthode Diffie-Hellman assure qu'aucune partie ne peut accéder à la clé autre que les deux parties engagées dans la communication, et cela même si la partie tierce observe les échanges d'informations entre les deux parties engagées. Nous justifions notre choix par la nécessité d'un mécanisme de chiffrement léger qui peut s'exécuter sur des dispositifs mobiles sans imposer des coûts computationnels supplémentaires. La figure 7.3 illustre un exemple sur le principe de fonctionnement d'un échange d'une clé de chiffrement entre deux utilisateurs Alice et Bob.

Comme illustré dans la figure 7.3, même si une partie tierce observe les informations échangées entre Alice et Bob, elle ne peut pas identifier la clé de chiffrement utilisée.

Le calcul des clés se fait entre chaque deux dispositifs qui vont s'échanger une requête géodépendante. Cela assure l'intégrité de la requête, et la génération d'une nou-

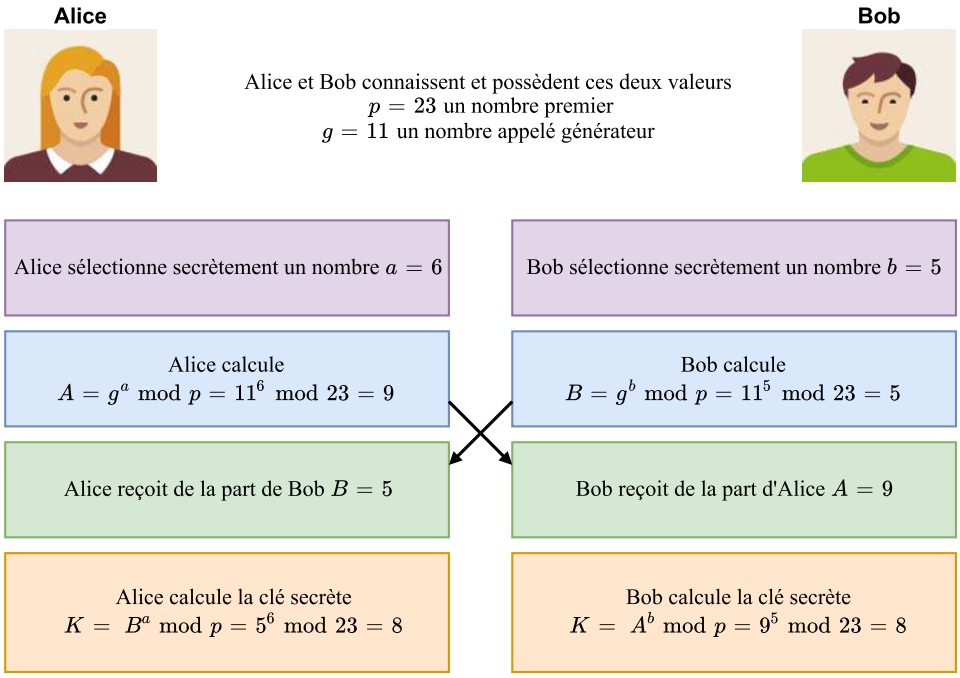


Figure 7.3 – Exemple de fonctionnement d'un échange de clés Diffie-Hellman

velle clé à chaque échange. Par ailleurs, le fonctionnement de la méthode Diffie-Hellman passe inaperçu et n'affecte pas le temps de transmission d'une requête géodépendante.

Un routage réussi se finalise par la récupération des résultats géodépendants à partir d'un LBS. Ces résultats sont enregistrés dans le cache localisé du dispositif final du chemin de routage. Nous discutons dans la suite l'implémentation proposée pour le cache géodépendant.

7.2.1.4 Cache géodépendant

Le rôle du cache géodépendant est de réutiliser les résultats obtenus d'un LBS pour d'autres requêtes similaires issues du même réseau de délégation. Notre implémentation implique le calcul d'une similarité entre la requête en termes de temps et de coordonnées géographiques. Deux requêtes sont similaires si elles contiennent le même contenu, des coordonnées géographiques à proximité et qui sont issues de deux périodes de temps

proches.

L'implémentation proposée vise à comparer le contenu de la requête déléguée avec le contenu du cache de chaque dispositif participant. Le cache est implémenté en tant que structure de donnée simple (une table dans une petite base de données locale) qui sert à enregistrer les requêtes géodépendantes passées par le dispositif en question, ainsi que les résultats géodépendants respectifs.

Le mécanisme de calcul de similarité est implémenté selon deux méthodes. La première se charge de calculer la distance physique entre deux positions géographiques et de vérifier cette distance par rapport au seuil de similarité. La deuxième s'occupe de calculer l'intervalle de temps entre les deux requêtes et de le vérifier par rapport à un seuil de période du temps. L'intervalle de temps est calculé en fonction de l'horodatage de la requête sans prendre en considération la date (jour).

La figure 7.4 illustre l'organigramme de la détermination des résultats à partir du cache géodépendant après réception d'une requête.

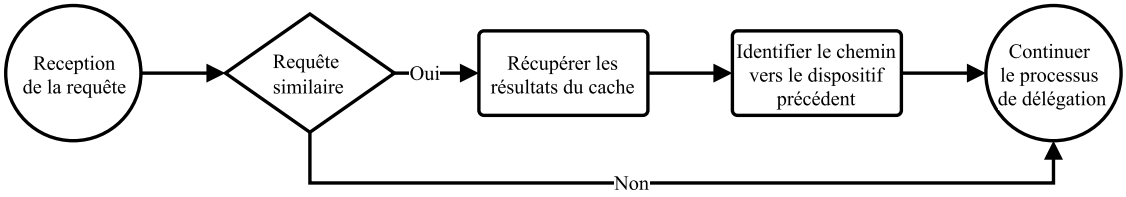


Figure 7.4 – Organigramme de fonctionnement du cache géodépendant

Le cache géodépendant est également doté d'un mécanisme d'expiration qui sert, selon une valeur prédéfinie, à supprimer les résultats relatifs aux anciennes requêtes. Cela assure que les résultats enregistrés en cache apportent un maximum d'utilité. Le délai d'expiration du cache peut être fixé par le mécanisme, ou associé à des règles. Dans l'implémentation proposée, nous supposons que les résultats qui datent de plus de 3 mois doivent être supprimés. Cela assure la minimisation de l'espace disque utilisé pour le cache, ainsi que la rapidité de la vérification à l'arrivée de chaque requête.

Nous discutons dans la section suivante notre implémentation du processus de quan-

tification de risques et de mesure de confiance.

7.2.2 Processus de quantification et de mesure

Les deux principaux composants du processus de quantification et de mesure sont ceux qui se chargent de quantifier les risques, et de mesurer la confiance. Les deux composants fonctionnent en parallèle (chapitre 5) et ont pour objectif de fournir des mesures concernant les risques engendrés par l'inclusion de contenu sensible dans les requêtes géodépendantes, ainsi que le niveau de confiance au sein du réseau de délégation.

Comme mentionné auparavant, le déclenchement des deux composants se fait en deux fois selon la direction de la requête. La première fois quand la requête est en chemin vers le LBS, et la deuxième au retour de la requête contenant les résultats vers le demandeur initial. À chaque déclenchement, le premier composant à interagir est celui de la quantification de risques, qui appelle par la suite le composant de la mesure de confiance.

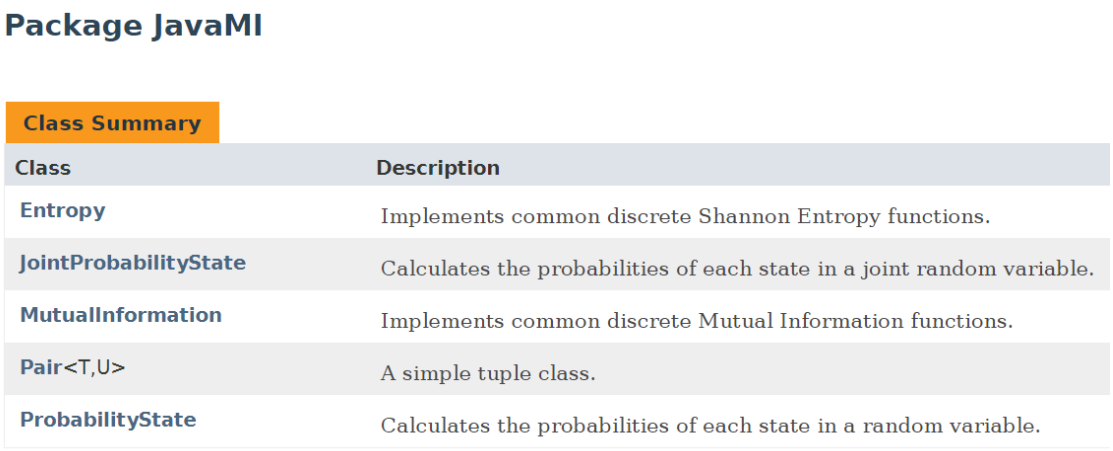
7.2.2.1 Quantification de risques

Dans le cas de la quantification des risques, nous basons notre implémentation sur un projet à code source libre avec une licence publique générale limitée GNU[1]. Le projet appelé *JavaMI*[2], représente une bibliothèque qui contient une série de fonctions pour travailler avec les concepts de la théorie de l'information. La bibliothèque contient également des fonctions de manipulation de variables qui servent au prétraitement des variables discrètes afin de générer les valeurs théoriques associées.

Notre choix d'une bibliothèque à code source libre se justifie par notre objectif de vouloir maximiser l'utilisation des solutions existantes proposées par la communauté informatique. Cela nous permet par ailleurs d'utiliser des outils qui sont largement supportés par la communauté, ce qui assure un gain majeur en termes de temps de réalisation

[1] https ://www.gnu.org/licenses/lgpl.html
[2] https ://github.com/Craigacp/JavaMI

et d'efficacité de l'outil en question. La figure 7.5 illustre les principaux composants de la bibliothèque *JavaMI*. Notons que les parties utilisées dans notre implémentation sont *"Entropy"* et *"MutualInformation"*.



Package JavaMI

Class Summary

Class	Description
Entropy	Implements common discrete Shannon Entropy functions.
JointProbabilityState	Calculates the probabilities of each state in a joint random variable.
MutualInformation	Implements common discrete Mutual Information functions.
Pair<T,U>	A simple tuple class.
ProbabilityState	Calculates the probabilities of each state in a random variable.

Figure 7.5 – Composants de la bibliothèque *JavaMI*

L'intégration de la bibliothèque *JavaMI* se fait dans le composant de la quantification des risques ; ce dernier se charge de la détermination et la normalisation des données destinées à la quantification. Le composant appelle par la suite les fonctionnalités de calcul d'entropie et d'information mutuelle de *JavaMI* afin d'obtenir les résultats de quantification. La figure 7.6 illustre la relation entre le composant de quantification implémenté et la bibliothèque *JavaMI*.

Comme illustré sur la figure 7.6, la bibliothèque à source code libre *JavaMI* n'est utilisée que pour le calcul des formules de la quantification de risques. Les formules doivent donc être formalisées selon le contenu de la requête en question, et conformément aux exigences de protection définies par le mécanisme.

Le composant de quantification nécessite l'altération de flux de données échangées entre l'utilisateur et un LBS afin d'identifier les éléments clés de la requête. Cette procédure est donc atteinte en passant par la plateforme de virtualisation.

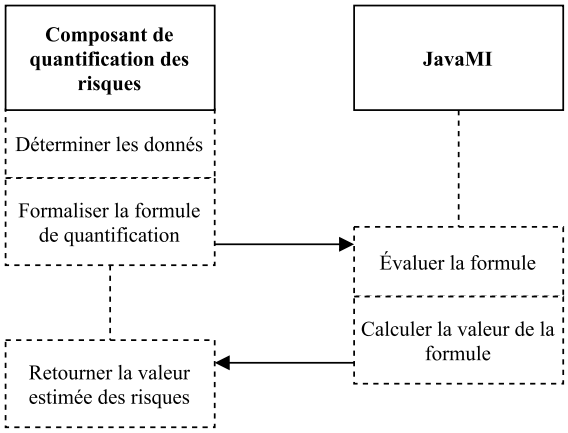


Figure 7.6 – Interaction entre le composant de quantification et la bibliothèque *JavaMI*

7.2.2.2 Mesure de la confiance

Quant à la mesure de confiance, notre implémentation rajoute des méthodes de mesure au sein du serveur du répertoire inconscient. Un exemple pour bien comprendre le fonctionnement de ce composant peut être illustré par les deux événements représentant "la vitesse de connexion" et "le succès de la délégation d'une requête géodépendante".

Dans le premier cas, l'occurrence de l'événement est évaluée en mesurant le temps que met un paquet de données à arriver au serveur du répertoire conscient et vice-versa. Pour le deuxième cas, représenté par le succès de la délégation d'une requête, l'événement ne peut être évalué sauf si le composant accède aux requêtes passées par la plateforme de virtualisation afin de définir lesquelles sont complétées.

Pour le deuxième cas, nous rajoutons un mécanisme de notification à l'espace virtuel qui sert à notifier le composant de mesure de confiance à la fin de chaque requête. Le contenu de notification se fait au début et à la fin d'un processus de délégation. Après la réception des résultats par le demandeur initial, ce dernier envoie l'événement correspondant au composant de mesure de la confiance. Le composant revérifie les participants dont il a reçu les événements de type "passage réussi de la requête" et attribue à chacun d'eux le score de confiance correspondant.

La mesure de niveau de confiance se fait d'une façon continue, indépendamment des

processus des délégations. Cela assure que la mesure de confiance n'influence pas les délais accordés à l'exécution d'un processus de délégation. La figure 7.7 illustre une partie du *journal d'événements* utilisé par le répertoire inconscient pour garder traces de tous les événements au sein du réseau de délégation. Ces événements sont ensuite utilisés pour évaluer le niveau de confiance selon les règles décrites dans la section 5.2.

```
TIMESTAMP              ORIGIN      TYPE
[29/04/2018 23:04:21]  e9b66557    connection established
[29/04/2018 23:04:24]  66d3c930    connection established
[29/04/2018 23:04:25]  e9b66557    request sent
[29/04/2018 23:04:25]  66d3c930    request received
[29/04/2018 23:04:26]  b459be2e    connection established
[29/04/2018 23:04:26]  b459be2e    request received
[29/04/2018 23:04:26]  66d3c930    request waiting for delegation
[29/04/2018 23:04:26]  e9b66557    request sent
[29/04/2018 23:04:26]  66d3c930    request delegated
[29/04/2018 23:04:27]  387d884b    connection established
[29/04/2018 23:04:27]  66d3c930    request received
[29/04/2018 23:04:27]  387d884b    request received
[29/04/2018 23:04:27]  66d3c930    request waiting for delegation
```

Figure 7.7 – Exemple des événements enregistrés dans le répertoire inconscient

7.2.3 Démarrage à froid

Le démarrage à froid représente le cas de la première utilisation de *Deloc*, c'est-à-dire le cas où le nombre de participants est insuffisant pour atteindre les objectifs de confidentialité estimés de *Deloc*. Tant que *Deloc* a besoin d'au moins deux dispositifs actifs pour initier un processus de délégation, le démarrage à froid peut présenter un problème. Une solution pourrait être la simulation d'appareils, de sorte que si aucun dispositif n'est actif, *Deloc* s'occupe des requêtes en leur nom. Toutefois, un tel processus nécessite la collecte des données des utilisateurs et peut atteindre la confidentialité de ces derniers, et cela contredit les objectifs de *Deloc*, ce qui n'est pas souhaitable.

Par conséquent, nous utilisons un déclencheur d'activité qui met *Deloc* en veille s'il existe moins de deux dispositifs actifs. En outre, l'utilisation de la position jumelle, qui peut être utilisée même sans dispositifs actifs, peut fournir une *protection temporaire*

lorsque le processus de délégation n'est pas accessible. L'utilisation d'une telle approche peut s'avérer insuffisante face à nos exigences de base, même si elle s'appuie sur le paradigme de substitution de position.

7.2.4 Complexité algorithmique

La complexité dans le contexte de ce travail considère le débit de calcul et la latence. Le débit est la mesure de capacité qui représente le nombre de requêtes traitées dans un laps de temps. La latence du temps de réponse correspond au temps nécessaire pour expédier une requête.

Nous considérons une évaluation empirique due à l'absence de références en matière de complexité dans le contexte des LPPM ne permettant pas de les comparer avec *Deloc*. Ainsi, ce que nous présentons dans l'évaluation empirique atteste l'efficacité, la performance, ainsi que les coûts computationnels minimaux assurés par *Deloc*. Nous pensons que ces indicateurs sont suffisants dans notre recherche qui a pour objectif de proposer un mécanisme de protection de confidentialité.

7.3 Scénarios d'exécutions

Nous exécutons l'implémentation de *Deloc* en fonction de trois scénarios qui reflètent des classes d'applications LBS du monde réel. Le *premier scénario* concerne l'utilisation de *Deloc* dans une application de navigation (par exemple, Google Maps), le *deuxième scénario* présente une application de météo (par exemple, AccuWeather), et le *troisième scénario* traite celui des applications de télédétection de masse, de l'anglais *Mobile crowd-sensing* (par exemple, MyShake). Notre choix pour ces classes se justifie par leur popularité et leurs couvertures des différents cas d'utilisation expliqués dans la section 2.1.

Pour chacun des scénarios examinés, nous définissons le type du contenu échangé, la fréquence et la direction des requêtes, ainsi qu'un scénario d'une interaction typique

de l'utilisateur. Le tableau 7.I liste les différences entre les trois classes d'application en fonction de leurs propriétés.

Tableau 7.I – Différences entre les classes LBS de l'étude de cas

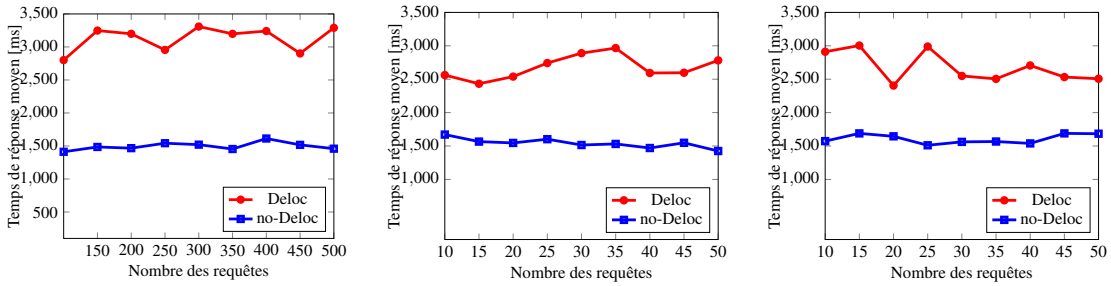
Type d'application	**Informations requises**	**Fréquence**	**Direction**
Navigation	Position exacte Points d'intérêts	Continue	Interrogation
Météo	Position exacte	Sporadique	Interrogation
Télédétection de masse	Position exacte Données des capteurs	Sporadique	Transaction

Tel que nous pouvons constater à partir du tableau 7.I, la différence principale entre les trois classes réside dans la fréquence et la direction des requêtes géodépendantes. Ainsi, une requête à fréquence continue implique l'envoi de plusieurs requêtes dans une période définie de temps, et avec un intervalle de temps négligeable (entre 1 et 2 secondes) entre l'envoi de chacune des requêtes. Le cas de requêtes sporadiques implique l'envoi des requêtes sur demande et avec un large (dépendant de l'utilisateur) intervalle de temps. La direction détermine si une requête est initiée par un utilisateur qui attend des résultats géodépendants (interrogation), ou une requête transmise à la demande d'un LBS et qui ne nécessite pas une réponse de la part de ce dernier (transaction).

Afin d'évaluer l'efficacité de *Deloc* dans ces cas, nous déroulons une application pour chaque classe LBS mentionnée. Notre choix de l'application à évaluer est basé sur le nombre d'utilisateurs ; nous choisissons pour chaque classe, l'application la plus utilisée, et nous la déroulons sous *Deloc* pour évaluer l'impact de ce dernier sur le flux d'exécution normal de l'application en question.

Comme mentionné auparavant, pour la première classe nous utilisons l'application de navigation de cartes *Google Maps*, pour la deuxième nous optons pour l'application de météo *AccuWeather*, et pour la troisième nous choisissons celle de collecte des données de tremblements de terre *MyShake*. La figure 7.8 illustre la différence en termes de

temps de réponse pour chacune des classes dans les deux cas de présence ou de non-présence du *Deloc*.



(a) Google Maps (b) AccuWeather (c) MyShake

Figure 7.8 – Distance moyenne et intervalle du temps moyen dans le mécanisme de délégation

Le temps de réponse décrit l'intervalle requis pour une requête afin qu'elle se termine, une valeur représentée par la différence de temps entre l'envoi d'une requête et la réception d'une réponse. Tel qu'illustré dans la figure 7.8, l'utilisation de *Deloc* n'occasionne pas un coût significatif en termes du temps de réponse. La différence dans le cas des 3 applications est en moyenne de 1283ms, avec la moyenne la plus élevée de 1630ms pour les applications de navigation.

Dans le cas des applications de navigation, *Deloc* lance plusieurs processus de délégation simultanément. Chaque processus est responsable d'une requête donnée. L'ensemble des résultats récupérés est construit en fonction du temps d'émission des requêtes, et constitue l'information utilisée pour la navigation (la trajectoire).

Pour les applications de météo, les requêtes géodépendantes s'exécutent sans aucune contrainte supplémentaire. *Deloc* lance un processus de délégation pour chaque requête et récupère les résultats quand ils sont disponibles.

Quant aux applications de télédétection de masse, c'est le serveur LBS qui initie la communication, ce qui implique un scénario légèrement différent. Quand un LBS demande des données de ses utilisateurs, la demande est reçue normalement sans passer par aucun mécanisme de protection. Cependant, une fois que l'application commence à

rendre les données géodépendantes, *Deloc* intercepte la requête et initie le processus de délégation. Cela assure la confidentialité et l'utilité estimée d'un service de télédétection en masse.

En conclusion, nous remarquons que *Deloc* s'intègre parfaitement dans les flux d'exécution de différentes classes d'application LBS, qu'il ne rajoute pas des coûts supplémentaires significatifs en termes du temps de réponse (1.6 seconde est tolérable dans la majorité des classes LBS), et qu'il assure l'utilité estimée de l'utilisation d'un LBS. En outre, l'architecture de *Deloc* assure son extension vers la majorité des classes existantes des LBS.

7.4 Conclusion

Nous avons décrit dans ce chapitre l'implémentation de *Deloc*, une procédure effectuée selon plusieurs étapes, chacune d'elle représentant un composant dans la plateforme. Les composants implémentés reflètent les fonctionnalités souhaitées de *Deloc* et assurent le respect de ses exigences.

Nous avons également discuté notre proposition pour permettre l'interception de flux de données sur les dispositifs mobiles Android, une proposition qui se résume dans la plateforme de virtualisation. L'importance d'une telle plateforme réside dans la nécessité d'un moyen qui permet non seulement l'altération de données d'une requête, mais aussi la transparence en termes d'interception de requêtes, et qui garantit la possibilité d'exécuter tout type de requête géodépendante.

En outre, nous avons détaillé le fonctionnement ainsi que l'interaction des composants de *Deloc*, tout en spécifiant le rôle de la plateforme de virtualisation dans chacun d'eux. Nous avons également montré l'interaction entre les deux processus majeurs de *Deloc*, notamment le processus de délégation et le processus de quantification.

Concernant le processus de délégation, nous avons discuté notre façon de l'implémenter, ainsi que sa faisabilité dans les applications actuelles. Quant au processus de

quantification, nous avons décrit une manière efficace qui garantit l'obtention des estimations précises.

Enfin, nous avons discuté aussi l'applicabilité de l'implémentation proposée dans trois des classes LBS les plus utilisées. Notre évaluation a porté sur des applications réelles (Google Maps, AccuWeather, et MyShake) pour un nombre défini des requêtes. Les résultats attestent que *Deloc* ne rajoute pas de coûts supplémentaires significatifs en termes du temps de traitement ($\approx$ 1.6 seconde).

En conclusion, nous avons mis en œuvre une implémentation efficace et applicable dans la majorité des cas des applications actuelles, sans imposer des coûts supplémentaires et sans contredire les exigences de protection et d'utilité souhaitées.

CHAPITRE 8

CONCLUSION

Le fait que les LBS envahissent nos vies quotidiennement ne peut pas être négligé, c'est grâce à leur facilité d'utilisation et à leur commodité que le nombre de leurs utilisateurs augmente exponentiellement. Cela dit, cette adoption conduit à des risques sérieux concernant la vie privée des utilisateurs. En fait, l'agrégation et l'analyse des données de géolocalisation sont devenues facilement accessibles par les techniques d'apprentissage et d'exploration de données [56, 66, 99], et peuvent certainement être raffinées lorsque l'historique des positions est également disponible.

Cependant, l'utilité et la commodité offertes par les LBS représentent la raison majeure derrière cette adoption. Dans la majorité des cas, les utilisateurs adoptent un LBS parce qu'ils sentent le besoin de l'utiliser. Par conséquent, une solution radicale de type "abandonner les LBS" n'est pas applicable dans ce contexte. Les utilisateurs ont besoin d'utiliser les LBS, mais ils ont également besoin de protéger leurs données personnelles.

L'objectif de cette thèse est de proposer et de mettre en œuvre une plateforme holistique de protection qui assure la confidentialité des utilisateurs tout en maintenant l'utilité du service géodépendant. Partant de cet objectif, nous avons défini une liste d'exigences basées sur des scénarios et des cas d'utilisation où chaque exigence identifie une caractéristique de la plateforme proposée. À titre de rappel, les exigences d'une plateforme de protection efficace peuvent se résumer dans la garantie des points suivants :

- **La confidentialité** des coordonnées géographiques de la position exacte de l'utilisateur.
- **L'anonymat** de l'identité réelle de l'utilisateur, et de toute information qui pourrait le relier à sa véritable identité.

- **L'utilité maximale** des données géodépendantes récupérées à partir d'un LBS donné.
- **La séparation** de la position de l'utilisateur du contenu de ses requêtes de manière à ce qu'un adversaire ne doive pas être capable de relier une ou plusieurs requêtes géodépendantes à un utilisateur donné.
- **L'indétectabilité** des données de géolocalisation et des autres données identificatrices pour qu'un adversaire ne puisse pas être capable de relier une ou plusieurs requêtes géodépendantes à tout autre type de données.
- **La flexibilité** et l'extensibilité de la plateforme de protection.
- **L'efficience** en matière de coût computationnel.

En nous basant sur ces exigences, nous avons proposé une plateforme holistique de préservation de la confidentialité de géolocalisation qui représente, avec tous ses composants, l'ensemble de nos contributions.

8.1 Contributions

Dans une remise en question du besoin de considérer la confidentialité et l'utilité estimée dans les LBS, nous proposons une plateforme visant à gérer la protection de la confidentialité tout au long de l'interaction entre l'utilisateur et le service géodépendant. Ainsi, nos contributions sont représentées selon les trois aspects principaux suivants.

• **Première Contribution.** Nous avons proposé un mécanisme collaboratif de préservation de la confidentialité de géolocalisation. Ce mécanisme se base sur la notion de délégation qui assure la transmission d'une requête géodépendante sans avoir besoin d'identifier l'utilisateur qui l'a émise. Pour ce faire, nous avons proposé et mis en œuvre des notions et modèles qui représentent les composants sur lesquels le modèle de délégation est fondé.

La première notion introduite est celle de la **position jumelle**, ainsi nous avons proposé une alternative aux techniques de substitution de position géographique pour atteindre une protection optimale face à certains types d'attaques populaires. En d'autres termes, la position jumelle est une position de substitution unique qui a comme avantage principal la prévention contre les attaques sur les multi-requêtes et celles qui tentent de définir les limites maximales de mouvement (section 4.1.2).

La deuxième notion réside dans l'architecture du mécanisme de délégation lui-même. Nous avons élaboré un ensemble de protocoles de communication entre les différents participants du processus qui assure la confidentialité de chacun des utilisateurs, et l'efficacité de transfert de requêtes géodépendantes. Les protocoles proposés font partie de l'architecture du *chemin bi-nœud de délégation* et celle de *transfert inconscient* des informations de routage (section 4.1.4 et 4.1.5).

La troisième notion est celle du *cache géodépendant* (section 4.1.6). En résumé, l'objectif du cache est de réduire les coûts de communication et de minimiser la latence au sein du réseau en minimisant le nombre des requêtes envoyées au LBS et le nombre d'opérations de délégation.

Nous avons mis en œuvre ces trois notions, et nous avons permis une protection de la vie privée dans les LBS sans devoir transformer les coordonnées géographiques de l'utilisateur. De même, les résultats de la validation ont démontré des coûts computationnels non significatifs, ce qui contourne les limites computationnelles des dispositifs mobiles.

• **Deuxième Contribution.** Nous avons mis en œuvre deux modèles d'évaluation de **risques** et de **confiance** dans le contexte des mécanismes collaboratifs.

Le premier modèle concerne la **quantification de risques** (section 5.1) associés à tout échange d'informations. L'objectif est donc d'assurer une protection maximale des utilisateurs. Le modèle de quantification de risques basé sur les théories d'information mutuelle et d'information spécifique. Nous avons réussi d'une part à mesurer la quantité d'information qu'un adversaire peut obtenir en observant une requête géodépendante, et à démontrer d'autre part l'efficacité de notre adaptation des théories mentionnées précé-

demment dans le contexte des LBS.

Le deuxième modèle concerne la **mesure de confiance** (section 5.2) au sein de l'environnement de délégation. L'idée était de proposer un mécanisme de mesure qui aide à identifier les participants malveillants dans un processus de délégation d'un côté, et à estimer le niveau de confiance globale dans des environnements collaboratifs de protection de l'autre côté. Ainsi, le modèle d'estimation proposé repose sur les chaines de Markov. Ceci est justifié par l'idée de base qui suppose que le niveau de confiance d'un participant est affecté par un ensemble de transitions. Autrement dit, nous avons identifié les actions possibles d'un participant lors d'un processus de délégation, et nous les avons considérés comme étant des événements. De même, nous avons modélisé une suite de ces événements par une chaine de Markov. En résultat, l'état de la chaine de Markov à un instant donné reflète le niveau de confiance du participant.

La validation des deux modèles atteste leur efficacité et leurs coûts supplémentaires non significatifs en termes de temps d'exécution. En moyenne, la différence entre une requête sans *Deloc* et une autre avec *Deloc* (avec tous ces composants) est de ≈ 1.3 seconde. Une valeur négligeable dans le contexte des LBS.

• **Troisième Contribution.** Nous avons élaboré une métrique probabiliste de confidentialité adaptée aux mécanismes collaboratifs. En s'inspirant des théories de l'algèbre floue, la métrique proposée, nommée δ-*fuzziness*, est une métrique probabiliste qui vise à estimer la probabilité d'identification associée au partage d'une position géographique par rapport aux autres positions formant la connaissance d'un adversaire. Nous avons proposé trois modèles qui représentent les fondements théoriques de δ-*fuzziness*, à savoir le modèle d'attaquant, le modèle de confidentialité et le modèle d'utilité.

Par ailleurs, nous avons défini les notions de *similarité* et de *sensibilité* qui visent à évaluer la ressemblance entre deux utilisateurs, et à estimer les risques que chacun peut occasionner à l'autre. Les deux notions ont été utilisées ensemble afin de mesurer la probabilité d'identification dans δ-*fuzziness*. De même, nous avons pu mettre en œuvre une métrique qui peut fournir une estimation exacte sur l'efficacité d'un mécanisme

collaboratif en utilisant une combinaison des ensembles flous et des probabilités. Une estimation qui ne peut être atteinte en utilisant les métriques existantes traditionnelles.

En outre, nous avons effectué une validation empirique de δ-*fuzziness* en la testant avec deux modèles d'attaque parmi ceux les plus utilisés. Nous l'avons également comparée avec deux métriques représentatives, à savoir *k-anonymity* pour la catégorie des métriques computationnelles, et *differential privacy* pour celle des métriques probabilistes. Les résultats attestent que notre métrique peut assurer des garanties plus rigoureuses tout en gardant l'utilité souhaitée d'un LBS.

En définitive, nous avons présenté une plateforme holistique de protection de vie privée au sein des LBS, qui ne sacrifie pas l'utilité estimée d'un service géodépendant. La plateforme peut être utilisée dans la majorité des classes actuelles en fournissant une protection évaluée par δ-*fuzziness*.

8.2 Limitations et travaux futurs

Bien que ce travail soit prometteur, il contient tout de même certaines limitations que nous aimerions résoudre dans un futur proche.

Une première réflexion concerne l'amélioration de la zone minimale de confidentialité. L'utilisation d'une zone adaptable qui change de taille en fonction de la position actuelle peut augmenter l'efficacité de *Deloc*. Cette adaptation doit tout de même être validée afin de mesurer son effet sur les garanties de protection fournies par *Deloc*. Une zone de confidentialité adaptable peut être achevée par l'exploitation de données contextuelles telles que la sensibilité de l'endroit où l'utilisateur se trouve, ou le nombre des individus dans le même endroit. En outre, la réalisation d'une telle zone nécessite une modélisation précise des propriétés contextuelles à considérer, une étude de la perception de ces propriétés pour chaque utilisateur, et un modèle de classification des données contextuelles. Il faut définir un mécanisme qui mesure la sensibilité d'une zone géogra-

phique donnée par rapport à l'utilisateur en question.

L'autre point qui peut être considéré en tant qu'amélioration est le rajout de l'exécution simultanée des processus de délégation. Utile dans les LBS à requêtes continues (*par ex.* applications de navigation), l'exécution simultanée peut réduire significativement l'empreinte de *Deloc* lors de l'échange de requêtes géodépendantes. Le rajout d'une telle caractéristique nécessite une modélisation exhaustive des dispositifs mobiles actuels. Étant donné la limite que ces derniers imposent, une gestion précise de l'exécution des requêtes est donc nécessaire.

La troisième limitation de ce travail se représente dans la validation dans un environnement réel. Bien que notre simulation utilise des données issues des utilisateurs réels, la validation dans des cas d'utilisation réelle est néanmoins nécessaire. Notre objectif est de lancer une procédure de validation à long terme où nous pourrions observer le comportement et l'efficacité de *Deloc* dans des dispositifs appartenant à des utilisateurs réels. L'objectif est de confirmer les résultats suggérés par la simulation d'un côté, et d'améliorer *Deloc* de l'autre côté.

BIBLIOGRAPHIE

[1] Osman Abul, Francesco Bonchi et Mirco Nanni. Never walk alone : Uncertainty for anonymity in moving objects databases. Dans *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 376–385. Ieee, 2008. ISBN 1424418364.

[2] Hazim Almuhimedi, Florian Schaub, Norman Sadeh, Idris Adjerid, Alessandro Acquisti, Joshua Gluck, Lorrie Faith Cranor et Yuvraj Agarwal. Your location has been shared 5,398 times ! : A field study on mobile app privacy nudging. Dans *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 787–796. ACM, 2015. ISBN 1450331459.

[3] Esma Aïmeur. Online privacy : risks, challenges, and new trends. Dans *International Conference on Risks and Security of Internet and Systems*, pages 263–266. Springer, 2014.

[4] Esma Aïmeur, Gilles Brassard et Paul Molins. Reconstructing profiles from information disseminated on the internet. Dans *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 875–883. IEEE, 2012. ISBN 1467356387.

[5] Esma Aïmeur, Gilles Brassard et Jonathan Rioux. Data privacy : An end-user perspective. *International Journal of Computer Networks and Communications Security*, 1(6):237–250, 2013.

[6] Esma Aïmeur, Oluwa Lawani et Kimiz Dalkir. When changing the look of privacy policies affects user trust : An experimental study. *Computers in Human Behavior*, 58:368–379, 2016. ISSN 0747-5632.

[7] Miguel E Andrés, Nicolàs E Bordenabe, Konstantinos Chatzikokolakis et Catuscia Palamidessi. Geo-indistinguishability : Differential privacy for location-based

systems. Dans *Proceedings of the 2013 ACM SIGSAC conference on Computer and communications security*, pages 901–914. ACM, 2013.

[8] Damian A Ball et Brandon Morton. The quadratic sieve-an implementation, 2010.

[9] Sebastian Banescu, Milan Petković et Nicola Zannone. *Measuring privacy compliance using fitness metrics*, pages 114–119. Springer, 2012. ISBN 3642328849.

[10] Justin Lee Becker. *Measuring privacy risk in online social networks*. University of California, Davis, 2009.

[11] Alastair R Beresford et Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive computing*, 2(1):46–55, 2003. ISSN 1536-1268.

[12] Alastair R Beresford et Frank Stajano. Mix zones : User privacy in location-aware services. Dans *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 127–131. IEEE, 2004. ISBN 0769521061.

[13] Claudio Bettini, X Sean Wang et Sushil Jajodia. Protecting privacy against location-based personal identification. Dans *Workshop on Secure Data Management*, pages 185–199. Springer, 2005.

[14] Michele Bezzi. An information theoretic approach for privacy metrics. *Trans. Data Privacy*, 3(3):199–215, 2010.

[15] Michele Bezzi. Anonymity measuring device, 2013.

[16] Antoine Boutet, Davide Frey, Rachid Guerraoui, Arnaud Jégou et Anne-Marie Kermarrec. Privacy-preserving distributed collaborative filtering. *Computing*, 98 (8):827–846, 2016. ISSN 0010-485X.

[17] Emmanuel Bresson, Olivier Chevassut, David Pointcheval et Jean-Jacques Quisquater. Provably authenticated group diffie-hellman key exchange. Dans *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 255–264. ACM, 2001. ISBN 1581133855.

[18] Ben-Jye Chang et Szu-Liang Kuo. Markov chain trust model for trust-value analysis and key management in distributed multicast manets. *IEEE Transactions on Vehicular Technology*, 58(4):1846–1863, 2009. ISSN 0018-9545.

[19] Konstantinos Chatzikokolakis, Catuscia Palamidessi et Marco Stronati. A predictive differentially-private mechanism for mobility traces. Dans *International Symposium on Privacy Enhancing Technologies Symposium*, pages 21–41. Springer, 2014.

[20] Eunjoon Cho, Seth A Myers et Jure Leskovec. Friendship and mobility : user movement in location-based social networks. Dans *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. ACM, 2011. ISBN 1450308139.

[21] Nathan Clarke, Jane Symes, Hataichanok Saevanee et Steve Furnell. Awareness of mobile device security : A survey of user's attitudes. *International Journal of Mobile Computing and Multimedia Communications (IJMCMC)*, 7(1):15–31, 2016.

[22] Common Criteria Portal. Common criteria portal. http://www.commoncriteriaportal.org/, 2017. Retrieved : November 02, 2017.

[23] Sunny Consolvo, Ian E Smith, Tara Matthews, Anthony LaMarca, Jason Tabert et Pauline Powledge. Location disclosure to social relations : why, when, and what people want to share. Dans *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 81–90. ACM, 2005. ISBN 1581139985.

[24] Scott Patrick Contini. Factoring integers with the self-initializing quadratic sieve, 1997.

[25] Caitlin D Cottrill. Location privacy preferences : A survey-based analysis of consumer awareness, trade-off and decision-making. *Transportation Research Part C : Emerging Technologies*, 56:132–148, 2015. ISSN 0968-090X.

[26] Thomas M Cover et Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012. ISBN 1118585771.

[27] Maria Luisa Damiani. Location privacy models in mobile applications : conceptual view and research directions. *GeoInformatica*, 18(4):819–842, 2014. ISSN 1384-6175.

[28] Maria Luisa Damiani, Elisa Bertino et Claudio Silvestri. The probe framework for the personalized cloaking of private locations. *Trans. Data Privacy*, 3(2):123–148, 2010.

[29] Biswajit Das et Jyoti Shankar Sahoo. Social networking sites-a critical analysis of its impact on personal and social life. *International Journal of Business and Social Science*, 2(14), 2011. ISSN 2219-1933.

[30] Yves-Alexandre de Montjoye, César A Hidalgo, Michel Verleysen et Vincent D Blondel. Unique in the crowd : The privacy bounds of human mobility. *Scientific reports*, 3, 2013.

[31] U N DeSA. World population prospects : the 2017 revision. *Population division of the department of economic and social affairs of the United Nations Secretariat, New York*, 2017.

[32] Michael R DeWeese et Markus Meister. How to measure the information gained from one symbol. *Network : Computation in Neural Systems*, 10(4):325–340, 1999. ISSN 0954-898X.

[33] Roger Dingledine, Nick Mathewson et Paul Syverson. Tor : The second-generation onion router. Report, DTIC Document, 2004.

[34] Josep Domingo-Ferrer, Sergio Martínez, David Sánchez et Jordi Soria-Comas. Co-utility : Self-enforcing protocols for the mutual benefit of participants. *Engineering Applications of Artificial Intelligence*, 59:148–158, 2017. ISSN 0952-1976.

[35] Matt Duckham et Lars Kulik. A formal model of obfuscation and negotiation for location privacy. Dans *International Conference on Pervasive Computing*, pages 152–170. Springer, 2005. ISBN 3540260080.

[36] Cynthia Dwork. *Differential Privacy*, pages 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-35908-1. URL https://doi.org/10.1007/11787006_1.

[37] Cynthia Dwork. Differential privacy : A survey of results. Dans *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[38] Cheng Tien Ee et Ruzena Bajcsy. Congestion control and fairness for many-to-one routing in sensor networks. Dans *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 148–161. ACM, 2004. ISBN 1581138792.

[39] Ehab ElSalamouny et Sébastien Gambs. Differential privacy models for location-based services. *Transactions on Data Privacy*, 9(1):15–48, 2016.

[40] EGNSSA European GNSS Agency. Gnss market report : Location-based services. http://www.gsa.europa.eu/sites/default/files/LBS_0.pdf, March 2015. Retrieved : November 02, 2017.

[41] Drew Fisher, Leah Dorner et David Wagner. Short paper : location privacy : user behavior in the field. Dans *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, pages 51–56. ACM, 2012. ISBN 1450316662.

[42] Ricard Fogues, Jose M Such, Agustin Espinosa et Ana Garcia-Fornes. Open challenges in relationship-based privacy mechanisms for social network services. *International Journal of Human-Computer Interaction*, 31(5):350–370, 2015. ISSN 1044-7318.

[43] Huiqing Fu, Yulong Yang, Nileema Shingte, Janne Lindqvist et Marco Gruteser. A field study of run-time location access disclosures on android smartphones. *Proc. USEC*, 14, 2014.

[44] Sarah Jean Fusco, Katina Michael, Anas Aloudat et Roba Abbas. Monitoring people using location-based social networking and its negative impact on trust. Dans *Technology and Society (ISTAS), 2011 IEEE International Symposium on*, pages 1–11. IEEE, 2011. ISBN 1424491495.

[45] Sébastien Gambs, Marc-Olivier Killijian et Miguel Núñez del Prado Cortez. Show me how you move and i will tell you who you are. Dans *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, pages 34–41. ACM, 2010. ISBN 1450304354.

[46] Sébastien Gambs, Marc-Olivier Killijian et Miguel Núñez del Prado Cortez. De-anonymization attack on geolocated data. *Journal of Computer and System Sciences*, 80(8):1597–1614, 2014. ISSN 0022-0000.

[47] Crispin Gardiner. Stochastic methods. *Springer Series in Synergetics (Springer-Verlag, Berlin, 2009)*, 1985.

[48] Mark N Gasson, Eleni Kosta, Denis Royer, Martin Meints et Kevin Warwick. Normality mining : Privacy implications of behavioral profiles drawn from gps enabled mobile phones. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(2):251–261, 2011. ISSN 1094-6977.

[49] Gabriel Ghinita. Privacy for location-based services. *Synthesis Lectures on Information Security, Privacy, and Trust*, 4(1):1–85, 2013. ISSN 1945-9742.

[50] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi et Kian-Lee Tan. Private queries in location based services : anonymizers are not necessary. Dans *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 121–132. ACM, 2008. ISBN 160558102X.

[51] Oded Goldreich. *Foundations of cryptography : volume 2, basic applications*. Cambridge university press, 2009. ISBN 1107393973.

[52] Philippe Golle et Kurt Partridge. On the anonymity of home/work location pairs. *Pervasive computing*, pages 390–397, 2009.

[53] Ralph Gross et Alessandro Acquisti. Information revelation and privacy in online social networks. Dans *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80. ACM, 2005. ISBN 1595932283.

[54] Marco Gruteser et Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. Dans *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM, 2003.

[55] Gary D Hachtel, Enrico Macii, Abelardo Pardo et Fabio Somenzi. Markovian analysis of large finite state machines. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(12):1479–1493, 1996. ISSN 0278-0070.

[56] Bo Han, Paul Cook et Timothy Baldwin. Geolocation prediction in social media data by finding location indicative words. *Proceedings of COLING 2012*, pages 1045–1062, 2012.

[57] C Helou, A Guandouz et E Aïmeur. A privacy awareness system for facebook users. *J. Inf. Secur. Res*, 31:15–29, 2012.

[58] Baik Hoh et Marco Gruteser. Protecting location privacy through path confusion. Dans *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 194–205. IEEE, 2005. ISBN 0769523692.

[59] Baik Hoh, Marco Gruteser, Hui Xiong et Ansaf Alrabady. Enhancing security and privacy in traffic-monitoring systems. *IEEE Pervasive Computing*, 5(4):38–46, 2006. ISSN 1536-1268.

[60] Nicholas Hopper, Eugene Y Vasserman et Eric Chan-Tin. How much anonymity does network latency leak ? *ACM Transactions on Information and System Security (TISSEC)*, 13(2):13, 2010. ISSN 1094-9224.

[61] HuffingtonPost. Please rob me ! https://www.huffingtonpost.com/2010/02/17/please-rob-me-site-tells_n_465966.html, January 2010. Retrieved : November 02, 2017.

[62] Panos Kalnis, Gabriel Ghinita, Kyriakos Mouratidis et Dimitris Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE transactions on knowledge and data engineering*, 19(12):1719–1733, 2007. ISSN 1041-4347.

[63] Hidetoshi Kido, Yutaka Yanagisawa et Tetsuji Satoh. An anonymous communication technique using dummies for location-based services. Dans *Pervasive*

Services, 2005. ICPS'05. Proceedings. International Conference on, pages 88–97. IEEE, 2005. ISBN 0780390326.

[64] Shinsaku Kiyomoto, Toru Nakamura, Haruo Takasaki, Ryu Watanabe et Yutaka Miyake. *PPM : Privacy policy manager for personalized services*, pages 377–392. Springer, 2013. ISBN 3642405878.

[65] George Klir et Bo Yuan. *Fuzzy sets and fuzzy logic*, volume 4. Prentice hall New Jersey, 1995.

[66] Yuki Kondo, Masatsugu Hangyo, Mitsuo Yoshida et Kyoji Umemura. Home location estimation using weather observation data. Dans *Advanced Informatics, Concepts, Theory, and Applications (ICAICTA), 2017 International Conference on*, pages 1–6. IEEE, 2017. ISBN 153863001X.

[67] Tracy Ann Kosa, K EI-Khatib et Stephen Marsh. Measuring privacy. *Journal of Internet Services and Information Security (JISIS)*, 1(4):60–73, 2011.

[68] Axel Küpper. Location-based services. *Fundamental and operation, John Willey and Sons, Ltd*, 2005.

[69] Balachander Krishnamurthy, Delfina Malandrino et Craig E Wills. Measuring privacy loss and the impact of privacy protection in web browsing. Dans *Proceedings of the 3rd symposium on Usable privacy and security*, pages 52–63. ACM, 2007. ISBN 159593801X.

[70] John Krumm. Inference attacks on location tracks. *Pervasive computing*, pages 127–143, 2007.

[71] Daniel Le Métayer et Guillaume Piolle. Droits et obligations à l'ère numérique : protection de la vie privée, 2010.

[72] Ninghui Li, Tiancheng Li et Suresh Venkatasubramanian. t-closeness : Privacy beyond k-anonymity and l-diversity. Dans *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115. IEEE, 2007. ISBN 1424408024.

[73] Chi Lin, Yi Wang, Shuang Wei, Danyang He et Jie Wang. Compromising location privacies for vehicles cloud computing. *International Journal of Web and Grid Services*, 14(1):88–105, 2018. ISSN 1741-1106.

[74] Heather Richter Lipford, Andrew Besmer et Jason Watson. Understanding privacy settings in facebook with an audience view. *UPSEC*, 8:1–8, 2008.

[75] Kun Liu et Evimaria Terzi. A framework for computing the privacy scores of users in online social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(1):6, 2010. ISSN 1556-4681.

[76] Yan Liu, Kun Wang, Huang Guo, Qing Lu et Yanfei Sun. Social-aware computing based congestion control in delay tolerant networks. *Mobile Networks and Applications*, 22(2):174–185, 2017. ISSN 1383-469X.

[77] Xavier Lopez. The future of gis : real-time, mission critical, location services. Dans *Proceedings of Cambridge Conference*, pages 713–725, 2003.

[78] Chris YT Ma, David KY Yau, Nung Kwan Yip et Nageswara SV Rao. Privacy vulnerability of published anonymous mobility traces. *IEEE/ACM Transactions on Networking (TON)*, 21(3):720–733, 2013. ISSN 1063-6692.

[79] Chunguang Ma, Lei Zhang, Songtao Yang et Xiaodong Zheng. Hiding yourself behind collaborative users when using continuous location-based services. *Journal of Circuits, Systems and Computers*, 26(07):1750119, 2017. ISSN 0218-1266.

[80] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke et Muthuramakrishnan Venkitasubramaniam. l-diversity : Privacy beyond k-anonymity. *ACM Transac-*

tions on Knowledge Discovery from Data (TKDD), 1(1):3, 2007. ISSN 1556-4681.

[81] Alessandra Mazzia, Kristen LeFevre et Eytan Adar. The pviz comprehension tool for social network privacy settings. Dans *Proceedings of the Eighth Symposium on Usable Privacy and Security*, page 13. ACM, 2012. ISBN 1450315321.

[82] Imran Memon, Qasim Ali Arain, Muhammad Hammad Memon, Farman Ali Mangi et Rizwan Akhtar. Search me if you can : Multiple mix zones with location privacy protection for mapping services. *International Journal of Communication Systems*, 2017. ISSN 1099-1131.

[83] Ricardo Mendes et João P Vilela. Privacy-preserving data mining : Methods, metrics, and applications. *IEEE Access*, 5:10562–10582, 2017. ISSN 2169-3536.

[84] Katina Michael et MG Michael. The social and behavioural implications of location-based services, 2011.

[85] Stephanie Mitchell. Harvard on foursquare. https://news.harvard.edu/gazette/story/2010/01/harvard-and-foursquare/, January 2010. Retrieved : November 02, 2017.

[86] CTV News. Paul bernardo's day parole hearing pushed to october. http://www.ctvnews.ca/canada/paul-bernardo-s-day-parole-hearing-pushed-to-october-1.3510836, July 2017. Retrieved : November 02, 2017.

[87] Ben Niu, Qinghua Li, Xiaoyan Zhu, Guohong Cao et Hui Li. Achieving k-anonymity in privacy-aware location-based services. Dans *INFOCOM, 2014 Proceedings IEEE*, pages 754–762. IEEE, 2014. ISBN 1479933600.

[88] Paul Ohm. Broken promises of privacy : Responding to the surprising failure of anonymization. *UCLA Law Review*, 57:1701, 2009.

[89] Balaji Palanisamy et Ling Liu. Mobimix : Protecting location privacy with mix-zones over road networks. Dans *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pages 494–505. IEEE, 2011. ISBN 1424489601.

[90] Balaji Palanisamy et Ling Liu. Attack-resilient mix-zones over road networks : architecture and algorithms. *IEEE Transactions on Mobile Computing*, 14(3): 495–508, 2015. ISSN 1536-1233.

[91] Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady Andrienko, Natalia Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, Jose Macedo et Nikos Pelekis. Semantic trajectories modeling and analysis. *ACM Computing Surveys (CSUR)*, 45(4):42, 2013. ISSN 0360-0300.

[92] Sameer Patil et Alfred Kobsa. Enhancing privacy management support in instant messaging. *Interacting with Computers*, 22(3):206–217, 2009. ISSN 1873-7951.

[93] Tao Peng, Qin Liu, Dacheng Meng et Guojun Wang. Collaborative trajectory privacy preserving scheme in location-based services. *Information Sciences*, 387: 165–179, 2017. ISSN 0020-0255.

[94] Laura Perusco et Katina Michael. Control, trust, privacy, and security : evaluating location-based services. *IEEE Technology and society magazine*, 26(1):4–16, 2007. ISSN 0278-0097.

[95] Andreas Pfitzmann et Marit Hansen. A terminology for talking about privacy by data minimization : Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf, août 2010. v0.34. Retrieved : November 02, 2017.

[96] Carl Pomerance. A tale of two sieves. *Biscuits of Number Theory*, 85:175, 2008.

[97] Alejandro Quintero et Samuel Pierre. Assigning cells to switches in cellular mobile networks : a comparative study. *Computer Communications*, 26(9):950–960, 2003. ISSN 0140-3664.

[98] Michael O Rabin. How to exchange secrets with oblivious transfer. *IACR Cryptology ePrint Archive*, 2005:187, 2005.

[99] Afshin Rahimi, Duy Vu, Trevor Cohn et Timothy Baldwin. Exploiting text and network context for geolocation of social media users. *arXiv preprint arXiv :1506.04803*, 2015.

[100] David Rebollo-Monedero, Jordi Forné, Agusti Solanas et Antoni Martínez-Ballesté. Private location-based information retrieval through user collaboration. *Computer Communications*, 33(6):762–774, 2010.

[101] Carmen Ruiz-Vicente, Dario Freni, Claudio Bettini et Christian S Jensen. Location-related privacy in geo-social networks. *Internet Computing, IEEE*, 15 (3):20–27, 2011. ISSN 1089-7801.

[102] Zakaria Sahnoune, Cheu Yien Yep et Esma Aïmeur. *Geolocation Hazards in Geosocial Networks*, pages 71–88. Springer, 2015. ISBN 331917956X.

[103] Muhammad N Sakib et Chin-Tser Huang. Privacy preserving proximity testing using elliptic curves. Dans *Telecommunication Networks and Applications Conference (ITNAC), 2016 26th International*, pages 121–126. IEEE, 2016. ISBN 1509009191.

[104] Nadine Schuessler et Kay W Axhausen. Identifying trips and activities and their characteristics from gps raw data without further information. *ETH, Eidgenössische Technische Hochschule Zürich, IVT*, 2008.

[105] Claude E Shannon. A note on the concept of entropy. *Bell System Tech. J*, 27(3): 379–423, 1948.

[106] Hassan Sharghi et Kamran Sartipi. A user behavior-based approach to detect the insider threat in distributed diagnostic imaging systems. Dans *Computer-Based Medical Systems (CBMS), 2016 IEEE 29th International Symposium on*, pages 300–305. IEEE. ISBN 1467390364.

[107] Reza Shokri, Panos Papadimitratos, George Theodorakopoulos et Jean-Pierre Hubaux. Collaborative location privacy. Dans *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*, pages 500–509. IEEE, 2011. ISBN 1457713454.

[108] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec et Jean-Pierre Hubaux. Quantifying location privacy. Dans *Security and privacy (sp), 2011 ieee symposium on*, pages 247–262. IEEE, 2011. ISBN 0769544029.

[109] Reza Shokri, George Theodorakopoulos, Panos Papadimitratos, Ehsan Kazemi et Jean-Pierre Hubaux. Hiding in the mobile crowd : Locationprivacy through collaboration. *IEEE Transactions on Dependable and Secure Computing*, 11(3): 266–279, 2014. ISSN 1545-5971.

[110] Reza Shokri, Carmela Troncoso, Claudia Diaz, Julien Freudiger et Jean-Pierre Hubaux. Unraveling an old cloak : k-anonymity for location privacy. Dans *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*, pages 115–118. ACM, 2010. ISBN 1450300960.

[111] Daniel J Solove. I've got nothing to hide and other misunderstandings of privacy. *San Diego L. Rev.*, 44:745, 2007.

[112] Eric Stewart. Live-tweeting a terrorist attack : How the public's posts can help in an emergency. http://www.rcmp-grc.gc.ca/en/gazette/live-tweeting-a-terrorist-attack/, April 2016. Retrieved : November 02, 2017.

[113] Fred Stutzman, Ralph Gross et Alessandro Acquisti. Silent listeners : The evolution of privacy and disclosure on facebook. *Journal of privacy and confidentiality*, 4(2):2, 2013.

[114] Latanya Sweeney. k-anonymity : A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002. ISSN 0218-4885.

[115] Monika Taddicken. The privacy paradox in the social web : The impact of privacy concerns, individual characteristics, and the perceived social relevance on different forms of self-disclosure. *Journal of Computer-Mediated Communication*, 19(2):248–273, 2014. ISSN 1083-6101.

[116] Akamai Technologies. State of the internet - q3 2015 report. Online, nov 2015.

[117] Paola Tubaro, Antonio A Casilli et Yasaman Sarabi. *Against the hypothesis of the end of privacy : an agent-based modelling approach to social media*. Springer Science & Business Media, 2013. ISBN 3319024566.

[118] Catherine E Tucker. Social networks, personalized advertising, and privacy controls. *Journal of Marketing Research*, 51(5):546–562, 2014. ISSN 0022-2429.

[119] Efraim Turban, Jon Outland, David King, Jae Kyu Lee, Ting-Peng Liang et Deborrah C Turban. *E-Commerce : Regulatory, Ethical, and Social Environments*, pages 573–612. Springer, 2018.

[120] Tracy L Tuten et Michael R Solomon. *Social media marketing*. Sage, 2017. ISBN 1526424541.

[121] Dong Wang, Deshu Li, Xiaohong Li et Zhu Xiao. An analysis of anonymity on capacity finite social spots based pseudonym changing for location privacy in vanets. Dans *Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on*, pages 763–767. IEEE, 2015. ISBN 1467376825.

[122] Samuel D Warren et Louis D Brandeis. The right to privacy. *Harvard law review*, pages 193–220, 1890. ISSN 0017-811X.

[123] Marius Wernke, Pavel Skvortsov, Frank Dürr et Kurt Rothermel. A classification of location privacy attacks and approaches. *Personal and Ubiquitous Computing*, 18(1):163–175, 2014. ISSN 1617-4909.

[124] Alan F Westin. Privacy and freedom. *Washington and Lee Law Review*, 25(1): 166, 1968. ISSN 0043-0463.

[125] Pedro Wightman, Winston Coronell, Daladier Jabba, Miguel Jimeno et Miguel Labrador. Evaluation of location obfuscation techniques for privacy in location based information systems. Dans *Communications (LATINCOM), 2011 IEEE Latin-American Conference on*, pages 1–6. IEEE, 2011. ISBN 1467302791.

[126] Yonghui Xiao et Li Xiong. Protecting locations with differential privacy under temporal correlations. Dans *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1298–1309. ACM, 2015. ISBN 1450338321.

[127] Qiuwei Yang et Pan Kong. Rulecache : A mobility pattern based multi-level cache approach for location privacy protection. Dans *Parallel and Distributed Systems (ICPADS), 2016 IEEE 22nd International Conference on*, pages 448–455. IEEE, 2016. ISBN 1509044574.

[128] Man Lung Yiu, Christian S Jensen, Xuegang Huang et Hua Lu. Spacetwist : Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. Dans *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 366–375. IEEE, 2008. ISBN 1424418364.

[129] Lichen Zhang, Yingshu Li, Liang Wang, Junling Lu, Peng Li et Xiaoming Wang.

An efficient context-aware privacy preserving approach for smartphones. *Security and Communication Networks*, 2017, 2017. ISSN 1939-0114.

[130] Yu Zheng. *Location-based social networks : Users*, pages 243–276. Springer, 2011. ISBN 1461416280.

[131] Kathryn Zickuhr. Location-based services. http://www.pewinternet.org/2013/09/12/location-based-services/, 2013. Retrieved : November 02, 2017.

PUBLICATIONS

Journaux

- **Sahnoune, Z.**, Aïmeur, E. (2018). Deloc : Delegation-Based Mechanism for Location Privacy. Journal of Information Privacy and Security. Submitted after revisions.

- Aïmeur, E., **Sahnoune, Z.** (2018). Privacy, Trust, and Manipulation in Online Relationships. Journal of Technology in Human Services. Submitted after revisions.

Conférences

- **Sahnoune, Z.**, Aïmeur, E., El Haddad, G., & Sokoudjou, R. (2015). Watch your mobile payment : an empirical study of privacy disclosure. In Trustcom/BigDataSE/ISPA, 2015 IEEE (Vol. 1, pp. 934-941). IEEE.

- **Sahnoune, Z.**, Yep, C. Y., & Aïmeur, E. (2015). Geolocation hazards in geosocial networks. In International Conference on E-Technologies (pp. 71-88). Springer.

- **Sahnoune, Z.**, Yep, C. Y., & Aïmeur, E. (2014). Privacy issues in geosocial networks. In International Conference on Risks and Security of Internet and Systems (pp. 67-82). Springer.