Un protocole basé sur des mobiles sécurisés pour une collecte participative de données spatiales en mobilité réellement anonyme

Dai Hai Ton That, Iulian Sandu-Popa, Karine Zeitouni, Cristian Borcea

Un protocole basé sur des mobiles sécurisés pour une collecte participative de données spatiales en mobilité réellement anonyme

Dai Hai Ton That[1], Iulian Sandu Popa[1, 2], Karine Zeitouni[1], and Cristian Borcea[3]

[1]Laboratoire DAVID - Université de Versailles Saint-Quentin , 45, Avenue des Etats-Unis, 78035 Versailles, France
[1]INRIA Saclay-Ile-de-France , 1 Rue Honoré d'Estienne d'Orves, 91120 Palaiseau, France
[1]Department of Computer Science, New Jersey Institute of Technology, Newark, New Jersey, USA
{dai-hai.ton-that2,iulian.sandu-popa,karine.zeitouni}@uvsq.fr, borcea@njit.edu

12 septembre 2017

Résumé

La collecte mobile d'information géographique volontaire (communément appelée VGI) se développe rapidement, transformant les citoyens en capteurs. Elle s'applique à l'observation de phénomènes spatiaux, comme le trafic routier, la pollution atmosphérique ou le bruit. Cependant, la géolocalisation des participants soulève des craintes justifiées de violation de vie privée. Le succès du déploiement d'une telle collecte à large échelle dépend donc de la capacité à protéger les données personnelles des participants tout en permettant de traiter en temps réel de flux continus de données géolocalisées. Cet article présente PAMPAS, un système distribué de VGI respectueux de la vie privée et offrant l'agrégation efficaces des données. PAMPAS se base sur des mobiles équipés d'une sécurité matérielle, appelés ici sondes sécurisées, qui calculent des requêtes distribuées tout en empêchant les utilisateurs d'accéder aux données des autres utilisateurs. Les échanges de données entre sondes sécurisées sont chiffrés et passent par un serveur dit de support. Nous proposons deux protocoles distribués : l'un pour le calcul d'agrégats spatiaux et l'autre pour le partitionnement spatial adaptatif utilisé notamment pour paralléliser le calcul d'agrégats. Les résultats des expérimentations et l'analyse de sécurité montrent que ces protocoles permettent de collecter, agréger et partager des statistiques ou des données dérivées an temps réel sans divulguer aucune donnée personnelle.

1 Introduction

L'information géographique volontaire (Volunteered Geographic Information - VGI) [Goo07] suscite un intérêt croissant dans l'observation de phénomènes urbains. Dans ce domaine, la collecte participative de données en mobilité (mobile participatory sensing ou texitMPS) est en plein essor dans des applications communautaires relatives au trafic routier ou à la pollution, offrant une alternative à l'utilisation d'équipements de l'infrastructure, qui sont très coûteux en installation et maintenance avec une couverture spatiale limitée. De nombreuses applications mobiles se basent déjà sur des données communautaires sur le trafic routier, permettant la navigation dynamique, par exemple, INTRIX [1]. D'autres s'en servent pour faciliter la recherche de places de parking ou pour la cartographie des nids de poules (par exemple streetbump.org à Boston) ou du bruit [DEAS12]. De plus, l'émergence de capteurs légers et à bas coût amène à un changement de paradigme dans les observations environnementales et sanitaires d'après l'agence de protection de dl'environnement américaire [SWS+13]. De nombreux projets de ce type ont été menés récemment dans ce domaine : CitSense à Oslo, CamMobSens à Cambridge, MetroSense à Dartmouth, ou encore OpenSense en Suisse [Pen14].

Le point commun de ces scénarios est que chaque participant joue le rôle de sonde mobile du phénomène observé. Ces données sont ensuite agrégées pour produire des cartes collaboratives du phénomène en question, lesquelles sont exploitées via des services ou partagées telles quelles avec la communauté. L'agrégation des observations se rapporte à des unités de temps et d'espace. Elles peuvent être de différents types : simple comptage, moyenne, médiane, ou des fonctions géostatistiques complexes faisant intervenir l'interpolation spatiale [NWL12]. La plupart des systèmes de MPS nécessitent l'envoi de la géolocalisation des participants en continu à un système central de collecte, géré par le fournisseur du service. Seulement, de plus en plus de citoyens sont réticents à accorder leur confiance ces organismes, car ils craignent une exploitation inappropriée de leurs données personnelles et des risques de violation de leur vie privée. En effet, la localisation peut révéler leurs activités, leurs habitudes et bien d'autres informations personnelles. De telles craintes sont justifiées, car il a été démontré que quatre localisations suffisent à ré-identifier 95% des personnes anonymes dans un jeux de données [dMHVB13]. Par conséquent, cette méfiance freine la diffusion et l'adoption à large échelle des systèmes collaboratifs.

Ce problème de privacité a fait l'objet de nombreux travaux, dont ceux de [BOT13], [DSJ13], [DEAS12], [QLM+11], [TRL+09]. Cependant, la plupart des approches utilisent un serveur d'anonymisation (proxy) et requièrent de lui faire confiance [DSJ13], [TRL+09], quand d'autres, basés sur la cryptographie, sont trop coûteux [BOT13], [DEAS12] , ou encore sacrifient la précision des résultats, comme [QLM+11] qui applique le masquage par du bruit. Un système de MPS performant offrant la protection des données personnelles reste un challenge aujourd'hui.

Récemment, l'émergence de dispositifs personnels sécurisés a ouvert de nouvelles perspectives dans la protection des données personnelles. Qu'il s'agisse de token portable sécurisé [ANP14], [TNP14], communicant avec le mobile de l'uti-

1. INRIX : http ://inrix.com, Waze : http ://www.waze.com and Navigon : http ://navigon.com

lisateur ou directement intégré à celui-ci (par exemple, Google Vault [2]), de matériel sécurisé inviolable embarqué dans le véhicule d'un conducteur [FNSA12] ou encore la zone de confiance, TrustZone CPU [ARM09] équipant la plus part des mobiles, tous ces dispositifs sécurisés offrent des garanties tangibles de sécurité basée sur un matériel sécurisé. Leur capacité de traitement de données combinée à ces garanties de sécurité peuvent être exploitées pour concevoir une architecture distribuée intégrant la protection native des données ou « privacy-by-design ». En se basant sur ces dispositifs personnels, cette architecture est dite centrée utilisateur, ce qui offre une alternative à l'architecture traditionnelle centrée serveur.

Cet article [3] présente un système de MPS respectueux de la vie privée PAMPAS (pour Privacy-Aware Mobile PArticipatory Sensing). Il s'agit d'un système distribué associé à un protocole sécurisé pour le calcul et le partage d'agrégats spatiaux. Le protocole et l'architecture se basent sur des mobiles sécurisés et offrent au système un haut niveau de garantie d'anonymat des participants. Il permet en outre une exécution en temps réel des agrégats spatiaux par la minimisation des coûts de communication et de calculs, et ce malgré les contraintes de ressources inhérentes aux dispositifs sécurisés. Ces propriétés sont susceptibles de rassurer les participants et par conséquent de les inciter à contribuer à la collecte [GLW+15].

Dans PAMPAS, tous les participants sont supposés avoir un dispositif mobile sécurisé matériellement, que l'on appelle « sonde sécurisée » ou SP pour Secure Probe. Un tel dispositif peut utiliser une large gamme de technologies parmi les dispositifs sécurisés cités précédemment. Notons qu'en plus de contribuer à la production de données, certaines sondes sécurisées jouent le rôle de nœud de calcul pour traiter les requêtes d'agrégation distribuées. Les SP sont également destinataires des résultats de ces requêtes. Le matériel sécurisé protège l'accès par d'autres utilisateurs des données échangées lors du calcul distribué. Tous les échanges de données sont chiffrés et s'opèrent à l'aide d'un serveur de support appelé SSI pour Supporting Server Infrastructure. Afin de fournir des résultats en temps réel et en continu, une exécution en parallèle de l'agrégation spatiale a été mise en œuvre dans PAMPAS, en répartissant le calcul selon un partitionnement spatial des SP. La construction et la maintenance de ce partitionnement visent d'une part à répartir et à équilibrer la charge dans les SP utilisées pour les calculs, et d'autre part, à garder l'anonymat dans ces parties.

PAMPAS a été implémenté et validé en utilisant une plateforme représentatives de dispositifs sécurisés. L'expérimentation utilise deux jeux de données synthétiques générés par micro-simulation du trafic routier dans des réseaux routiers de deux villes de petite et de moyenne tailles. Nous avons comparé PAMPAS avec une méthode de l'état de l'art décrite dans [TNP14] et montrée son efficacité et son passage à l'échelle. Notre analyse de la sécurité de notre protocole montre son efficacité dans la protection des données personnelles tout au long du processus de collecte participative.

Le reste de cet article est organisé comme suit. La section 2 donne une vue d'ensemble du système PAMPAS avec son architecture, le modèle de menaces, puis les exigences du protocole. La phase d'agrégation de ce protocole est pré-

2. http ://www.cnet.com/news/googles-project-vault-is-a-security-chip-disguised-as-an-micro-sd-card/

3. Ce travail a été en partie financé par le projet ANR-11-INSE-005 KISS - Koffre-fort d'Informations personnelles Sûr et Sécurisé.

sentée dans la section 3, puis le partitionnement dans la section suivante. La section 5 donne une analyse de la sécurité de PAMPAS. L'évaluation expérimentale est détaillée dans la section 6. La section suivante compare PAMPAS à l'état de l'art. Pour finir, nous récapitulons les contributions et donnons quelques perspectives en conclusion de cet article.

2 Vue d'ensemble de PAMPAS

Cette section présente l'architecture du système PAMPAS, le modèle de menace ainsi que le modèle de données et de traitement considérés dans notre contexte. Partant de ces éléments, nous déclinons les besoins précis du protocole mis en place dans notre système.

2.1 Architecture du système

PAMPAS repose sur une architecture hybride combinant des éléments sécurisés du côté de l'utilisateur (les sondes sécurisées SP) et un serveur de support (le SSI) permettant l'échange sécurisé de messages entre les utilisateurs mobiles. Les SP et le SSI exécutent conjointement deux protocoles pour échanger les observations captées, pour calculer des requêtes continues d'agrégats spatiaux, et enfin, pour générer périodiquement une partition des SP selon leur localisation. Cette architecture assure une protection totale de la privacité des utilisateurs vis-à-vis du serveur de support SSI.

Comparée à une architecture décentralisée en pair-à-pair (P2P), cette architecture hybride n'entraîne aucun surcoût du côté des participants pour maintenir la couche P2P, ce qui est un avantage important étant donné les faibles ressources des SP des participants et leur dynamicité. De plus, l'architecture hybride permet l'échange de messages entre SP en $O(1)$, alors que ce coût est en $O(logN)$ dans les système P2P.

Sonde sécurisé (SP). Chaque participant dans notre système est muni d'un dispositif sécurisé inviolable, appelé sonde sécurisée ou SP dans notre contexte. Une SP peut être de n'importe quel type de ceux florissant sur le marché. Peu importe son nom commercial et son facteur de forme, ce type de dispositif intègre à minima un microcontrôleur sécurisé ou MCU (par exemple, une carte à puce) pour le calcul, connectée à une mémoire Flash NAND, typiquement sur carte SD pour le stockage de données.

Dans le contexte de PAMPAS, une SP joue trois rôles : sonde mobile (ou producteur de données), nœud de calcul et enfin, consommateur du résultat des requêtes. En effet, il envoie les données spatiotemporelles observées, sous forme chiffrée au SSI, participe au calcul distribué des agrégats et reçoit le résultat final des autres SP via le SSI. Le haut niveau de sécurité des SP ont font des composants de confiance dans cette architecture. Toutefois, cette confiance a un prix qui est dû aux limitations des ressources dans les microcontrôleurs sécurisés : les MCU ont généralement un processeur peu puissant et une très faible capacité en mémoire RAM (de l'ordre de quelques dizaines de Kilo-octets). Dans certains cas, les SP sont alimentées par une batterie dont la consommation doit être prise en compte. De plus, les SP ont une disponibilité limitée, dépendant des connexions/déconnexions de leur utilisateur. Par conséquent, il est nécessaire d'optimiser tous les traitements et toutes les communications au niveau

des SP.

Serveur de support (SSI). A la différence de l'architecture typique centrée serveur, le SSI dans PAMPAS agit uniquement comme coordinateur pour échanger les messages entre les SP et pour des stockages temporaires. Comme le SSI n'est pas de confiance, les données qui y transitent sont chiffrées en utilisant un chiffrement non déterministe, c'est à dire que le code crypté n'est pas identique d'une exécution à l'autre, de manière à contrer la cryptanalyse.

En conclusion, la sécurité et la privacité dans PAMPAS résulte de la combinaison de matériel sécurisé et une distribution de l'ensemble des données et des traitements sur des SP. Dès lors, le défi est d'être capable de calculer en continu tout type de fonctions agrégats en temps réel dans une telle architecture centrée utilisateur, et ce malgré les faibles ressources des SP.

2.2 Modèle de menace

Une des principales préoccupations dans PAMPAS est d'assurer une pleine protection des données personnelles des participants. L'autre est d'empêcher tout effort de changer ou de biaiser les résultats d'agrégation. Le système repose sur les hypothèses suivantes.

La confiance dans les SP est assurée par le haut-niveau de garanties contre les attaques physiques apportées par les MCU. Par ailleurs, toutes les données stockées dans la mémoire Flash sont protégées par chiffrement. A noter que faire confiance dans les SP ne signifie pas de faire confiance aux utilisateurs eux-mêmes qui n'ont accès qu'aux résultats des requêtes.

Par ailleurs, nous considérons le modèle « honnête mais curieux », ce qui veut dire que le SSI obéit au protocole qu'il est supposé suivre, mais qu'il pourrait tenter d'inférer tout ce qu'il peut des données et des comportements qu'il observe. Dans notre cas, il y aurait peu d'intérêt à considérer le modèle malicieux, car si le SSI tente de falsifier le protocole (s'il supprime des messages par exemple) pour inférer plus d'informations, il serait détecté : les SP effectuant l'agrégation pourraient vérifier la présence de leurs données échangées via le SSI. Le SSI s'exposerait ainsi à des conséquences juridiques et financières.

Pour finir, nous supposons la communication entre les SP et le SSI anonymes, comme par l'usage de proxy ou d'un réseau d'anonymisation tel que Tor, de manière à ce que la privacité ne puisse pas être compromise par un serveur malicieux qui tente d'identifier l'origine des messages.

2.3 Modèles de données et de requêtes

Modèle de données. PAMPAS a été conçu pour répondre à différents types de calculs dans le contexte d'application du domaine du MPS. Typiquement, ces applications nécessitent l'agrégation de mesures de capteurs géo-localisées et datées [PBBL11]. Ces données peuvent être décrites par un triplet *(localisation, temps, valeur)*. Elles sont chiffrées puis envoyées au SSI. PAMPAS n'impose pas de restriction sur la fréquence de génération et d'envoi de ces données par l'application. Cependant, le système doit être efficace et doit passer à l'échelle pour une collecte de grande envergure et à une fréquence élevée. Par ailleurs, la garantie de privacité des participants doit être indépendante du nombre de mesures collectées et de leur distribution spatio-temporelles. Concernant la représentation de la localisation, PAMPAS considère deux modèles selon le type

de mobilité des participants : (i) la localisation absolue définie par les coordonnées géographiques *(x, y)* et qui est adaptée au mouvement libre dans un espace 2D ; (ii) la localisation par référence à un réseau, définie par le couple *(rid, pos)* où *rid* désigne un identifiant de route et *pos* une position relative sur cette route, qui correspond à la mobilité contrainte dans un réseau connu d'avance (ici le réseau routier). La valeur mesurée peut correspondre à différentes mesures, comme le volume de trafic, la vitesse ou le niveau de bruit.

Modèle de requêtes. PAMPAS supporte des requêtes continues d'agrégation spatio-temporelle sur un flux de données selon une ou des fonctions agrégats. Ces requêtes peuvent être exprimées en langage CQL [JMS+08] comme dans la figure 1. A noter que dans le cas de PAMPAS, le flux de données est multi-source étant donné que les observations proviennent de plusieurs SP en parallèle. Les résultats sont générés selon une fenêtre temporelle glissante et selon un découpage en unité spatiales et permet ainsi un suivi de l'évolution temporelle des statistiques spatiales d'intérêt pour l'application. Par exemple, cela servirait à produire des cartes dynamiques du bruit, du temps de parcours moyen ou de la vitesse sur un réseau routier.

```
SELECT SpatialAggregate(value), [COUNT(*)]
FROM ParticipatorySensingStream
        [WINDOW x seconds SLIDE x seconds]
GROUP BY spatialUnit
HAVING predicateOnSpatialAggregate
```

FIGURE 1 – Requête agrégat spatio-temporelle dans PAMPAS

Unités spatiales. Comme dans le modèle de requêtes ci-dessus, les agrégats sont basés sur un ensemble fini d'unités spatiales formant un espace de référence discret. Sans perte de généralité, nous considérons dans la suite deux types d'espaces de référence représentatifs de deux types de mobilité. Dans le cas où les participants se déplacent librement dans l'espace, nous considérons un découpage régulier selon une grille où chaque cellule constitue une unité spatiale. La taille de la cellule est déterminée par les besoins de l'application. Lorsque la mobilité est canalisée dans un réseau connu d'avance, comme le réseau routier, les unités spatiales correspondent à des sections du réseau, comme par exemple, des tronçons de routes entre deux intersections successives. Dans les deux cas, le nombre d'unités spatiales peut être grand et se mesurer en centaines ou milliers d'unités. A noter que, comme pour les fenêtres temporelles d'observation, PAMPAS n'impose pas de restrictions sur le découpage en unités spatiales. La génération de l'espace de référence est en dehors du périmètre de cet article. Le comptage correspond au calcul de distribution spatiale des participants. Comme expliqué dans la section 4, il sert à vérifier l'équilibrage du partitionnement introduit pour optimiser notre processus.

Fonctions d'agrégation. PAMPAS peut théoriquement calculer tout type d'agrégations pour des applications de MPS. Le support de ces fonctions en temps réel pourrait être limité par les faibles capacités de calcul et de mémoire RAM des SP. En pratique, notre expérimentation (cf. section 6.2) a montré que même les fonctions complexes, comme la médiane et l'interpolation spatiale, s'exécutent en quasi temps réel. Nous considérons les fonctions suivantes classées en trois catégories : (i) les fonctions algébriques et distributives : comptage,

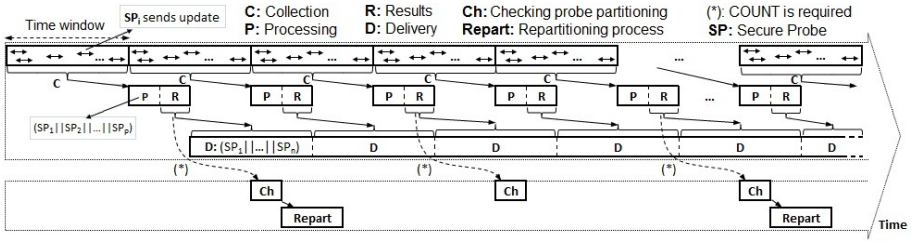

FIGURE 2 – Schéma global du protocole dans PAMPAS

somme, moyenne, écart-type, etc., et qui sont les plus couramment utilisées dans l'état de l'art pour calculer le volume du trafic ou la vitesse pratiquée dans un réseau de transport [BOT13], [DEAS12], [PBBL11] ; (ii) des fonctions holistiques : médiane, centile, top-k, également utilisées dans les analyses statistiques ; (iii) les fonctions spécifiques : principalement de géostatistiques dont l'interpolation spatiale, par exemple par la pondération inverse à la distance - inverse distance weighting ou IDW - [NWL12].

La particularité des deux dernières catégories de fonctions agrégats est qu'il est difficile de les exécuter de manière incrémentale, ce qui ne permet pas de les traiter par sous-ensembles. Ces fonctions ne sont pas supportées dans les approches cryptographiques (cf. section 7) car celles-ci utilisent les propriétés d'additivité des agrégats de la première catégorie. De même, les fonctions holistiques ne peuvent être calculées efficacement dans l'architecture distribuée proposée dans [TNP14] comme le montre la section 6.

2.4 Exigences du protocole

A la lumière des éléments d'architecture et de modèle décrits ci-dessus, le protocole à mettre en place pour PAMPAS doit adresser les défis suivants :

(i) Privacité : le protocole doit garder toutes les données personnelles dans les SP et ne révéler aucune information sur les participants. Contrairement aux architectures centrée serveur, l'architecture centrée-utilisateur permet de satisfaire ce critère.

(ii) Généralité : le protocole doit être en mesure de calculer tout type de fonction d'agrégation sur les observations collectées par les participants et couvrant échelle très large, comme une ville. Ceci le différencie des approches cryptographiques dans lesquelles seules les calculs simples tels que la somme ou la moyenne sont supportés et seulement en nombre limité de groupes.

(iii) Performances : le protocole doit être suffisamment efficace pour calculer de manière continue des agrégats simples ou complexes avec des ressources limitées des SP et retourner les résultats en quasi temps réel [4]. Par conséquent, il est indispensable de minimiser les coûts de communication et de traitement des données dans le protocole.

(iv) Le passage à l'échelle : le protocole doit permettre un déploiement du système PAMPAS sur un territoire très large et supporter la charge de millions d'utilisateurs, et pour un échantillonnage des mesures à haute fréquence.

4. A noter que les limitations des ressources dans les SP sont dues à la sécurité embarquée et à un impératif économique dans le contexte d'un déploiement grand public

(v) Précision : PAMPAS doit refléter avec fidélité les mesures calculées. Autrement dit, la protection des données personnelles ne devrait pas impacter la précision du résultat généré par le protocole.

3 Protocole global d'agrégation

Le protocole sécurisé global dans PAMPAS se compose de trois phases successives exécutées en pipeline (voir Figure 2). Dans la première phase, le SSI collecte les mesures chiffrées envoyées par les SP dans une période correspondant à une fenêtre temporelle glissante - appelée période de collecte -. Le chiffrement utilisé pour les mesures est non déterministe de manière à ne révéler aucune information au SSI lui permettant de lier des mises à jour successives. Toutes les SP partagent une clé secrète [5] de chiffrement / déchiffrement des données. Notons que malgré cette capacité de déchiffrement par la SP des données d'autres participants, son détenteur n'y a pas accès grâce à leur propriété d'inviolabilité matérielle. Ainsi, l'utilisateur n'a accès qu'au résultat final des requêtes. Les données de base peuvent donc transiter et être manipulées par différentes SP en toute sécurité. A la fin de chaque période de collecte, le SSI déclenche le calcul d'agrégation sur la fenêtre glissante. Dans cette phase, seulement une petite partie des SP, choisies aléatoirement par le SSI, est impliquée dans le calcul et ces SP changent à chaque exécution de la requête agrégat.

Pour cela, le SSI partitionne l'échantillon de mesures collecté de manière à répartir les traitements sur quelques SP de calcul et les données sur leur mémoire RAM, de manière à éviter le recours plus couteux à la mémoire secondaire. Ensuite, chaque partie est envoyée à la SP qui calcule un résultat d'agrégation pour cette partie, puis renvoie le résultat chiffré au SSI. Enfin, la troisième phase du protocole consiste en la période de diffusion du résultat aux participants. Leurs SP déchiffrent ces résultats et le cas échéant les fusionnent localement (lorsque leur requête couvre plusieurs zones de la partition).

Les algorithmes 1 et 2 donnent une description détaillée des opérations exécutées respectivement par le SSI et les SP. Dans ce qui suit, nous notons E_k l'opération de chiffrement déterministe avec une clé k et et nE_k celle de chiffrement non déterministe avec cette même clé. De même, nous utiliserons les notations E_k^{-1} et nE_k^{-1} pour les opérations inverses de déchiffrement déterministe et non déterministe correspondant. Afin d'améliorer les performances d'exécution, le protocole proposé groupe les données des participants par zone du découpage spatial, ce qui, contrairement aux protocoles existants comme celui de [TNP14] (voir la section 7), permet d'agréger l'ensemble de l'échantillon d'un groupe par une SP unique. Pour cela, en plus des valeurs cryptées de manière non déterministe, chaque sonde SP envoie le code crypté de manière déterministe de la zone correspondant. Ainsi le corps du message est de la forme : $message = (E_k(groupID), nE_k(sample))$ (cf. Algorithme 1, ligne 4 et Algorithme 2, ligne 3). De cette manière, le SSI peut grouper les messages en se basant sur le code crypté de la zone, puis envoyer les messages correspondants à une SP différente par zone pour générer le résultat de l'agrégat (lignes 7 à 9 de l'algorithme 1). Les avantages sont multiples. En premier lieu, la période d'agrégation est garantie de se terminer en une seule itération, étant donné que

5. Pour augmenter encore plus la sécurité, la clé secrète est renouvelée périodiquement. Elle est générée de manière aléatoire par une SP choisi au hasard.

Algorithme 1 : Protocole de PAMPAS du côté du SSI

collection_period() **:**
  /* Receive encrypted updates from SPs */
  while $(true)$ **do**
    $message = (E_k(G_i), nE_k(sample))$
    $store(message) \rightarrow list[E_k(G_i)][currentTimeWindow]$

processing_period() **:**
  foreach *i in* $\{E_k(G_i)\}$ **do**
    /* feed in parallel the randomly selected SPs */
    randomly select $SP_i \in E_k(G_i)$
    while $message \leftarrow list[E_k(G_i)][lastTimeWindow]$ **do**
      $send(message, SP_i)$
  foreach *i in* $\{E_k(G_i)\}$ **do**
    /*Receive the final results from worker SPs*/
    $enc\_result_i^{final} = (E_k(G_i), nE_k(result))$

delivery_period() **:**
  foreach *i in* $\{E_k(G_i)\}$ **do**
    /*Push $result_i$ to all requesting SPs*/
    $multicast(enc\_result_i^{final}, \{SP_k\})$

Algorithme 2 : Protocole de PAMPAS du côté d'une SP

collection_period() **: /* for all SPs */**
  /* Generate and send the sensing update : $update(G_i, sample)$ */
  $message = (E_k(G_i), nE_k(sample))$ $send(message, SSI)$
  /* Send a fake sample to the SSI with probability $P_{G_i}^{fake}$ */
  if $rand(0, 100) >= P_{G_i}^{fake}$ **then**
    $fake\_message = (E_k(G_i), nE_k(fake\_sample))$
    $send(fake\_message, SSI)$

processing_period() **: /* only for the selected SPs, one for each** G_i ***/**
  while *message = receive(SSI)* **do**
    $sample = nE_k^{-1}(message)$
    $result = result \oplus sample$
  $enc\_result_i^{final} = (E_k(G_i), nE_k(result))$
  $send(enc\_result_i^{final}, SSI)$

delivery_period() **: /* for all SPs */**
  /* Pull the results for required $\{G_i\}$ from the SSI */
  foreach *i in* $\{E_k(G_i)\}$ **do**
    $send\_request(E_k(G_i), SSI)$
    $result_i^{final} = nE_k^{-1}(receive(SSI))$

chaque SP impliquée calcule l'agrégat complet par unité spatiale dans la zone dont il a la charge. Ceci permet une diminution drastique des coûts à la fois du calcul et de communication. Le second avantage est que la consommation de la mémoire de la SP est minimisée du fait de la répartition des données et des unités spatiales, ce qui évite le stockage en mémoire secondaire et réduit également les temps de traitements au niveau des SP. Enfin, le résultat final est également partitionné et la SP destinataire peut être intéressée par une zone limitée de la carte, ce qui offre une amélioration supplémentaire du coût de communication.

Cependant, en dépit de tous ces bénéfices, l'approche décrite ci-dessus a une limitation en raison de la variabilité de distribution spatiales des participants et de leur mobilité. En effet, bien que la localisation exacte des participants, ainsi que l'unité spatiale soient masquées, connaissant le nombre de participants par zone, le SSI pourrait utiliser des informations externes (par exemple, des statistiques sur la densité de trafic) pour inférer une localisation approximative des participants et compromette, dans une certaine mesure, leur privacité.

4 Protocole de partitionnement des sondes sécurisées

Pour contrer les possibles attaques précitées dues à une répartition spatiale inégale des participants, PAMPAS partitionne les unités spatiales de la requête de telle sorte que le nombre de SP dans les parties soit approximativement le même et leur nombre soit suffisamment élevé pour garder l'anonymat des participants. Comme pour le protocole d'agrégation de données, le partitionnement est exécuté par une SP choisie aléatoirement. Les parties forment une distribution spatiale, autrement dit, zones Z_i constituées chacune d'unités spatiales adjacentes. Etant donné que les utilisateurs sont mobiles, leur distribution dans l'espace change au fil du temps. Par conséquent, le partitionnement doit être recalculé périodiquement afin de garder les parties équilibrées modulo un écart très faible. Le défi alors consiste à mettre en œuvre un algorithme de partitionnement pouvant être exécuté périodiquement par une SP et de manière efficace. En effet, les algorithmes classiques de partitionnement spatial sont beaucoup trop coûteux pour être considérés dans le contexte de ressources très limitées des SP.

Notre algorithme est basé sur l'idée suivante. Nous utilisons une courbe de remplissage de l'espace afin d'indexer les unités spatiales définies dans la requête de l'application. La propriété de ces courbes est de transformer un espace multidimensionnel en espace à une dimension en conservant la localité des données d'origine. Ensuite, nous trions les unités spatiales sur la valeur d'index de la courbe de remplissage. Une fois triées, on calcule ou on vérifie un partitionnement équilibré des SP. Le coût algorithmique est de $O(P)$ en stockage et $O(N)$ en calcul, où P est la taille de la partition et N le nombre total d'unités spatiales.

Indexation des unités spatiales. Dans notre système, nous utilisons les courbes de Hilbert, qui sont largement employées, pour indexer les unités spatiales considérées par les applications de collecte participative, mais d'autres courbes de remplissage de l'espace peuvent être également utilisées (par exemple, les courbes Z). Dans le cas d'un mouvement libre des SP, l'indexation est simple puisque l'espace est déjà partitionné avec une grille uniforme (voir Figure 3

Algorithme 3 : Vérification de l'équilibre des partitions (sur une SP)

```
check_probe_partitioning() :/* one randomly selected SP */
    /* Pull all the results from the SSI */
    foreach i in {E_k(G_i)} do
        send_request(E_k(G_i), SSI)
        enc_result_i^final = receive(SSI)
        allCounts[G_i][] ← E_k^{-1}(enc_result_i^final)
        update localy stored counts for spatial units /* also required to
        compute the probability to generate fake samples */
        compute weights[G_i] = SUM(allCounts[G_i][])
    compute standard_deviation(weights)
    if standard_deviation(weights) < threshold then
        send_for_broadcast(nE_k(allCounts), SSI)
    else
        execute probes_repartitioning()
```

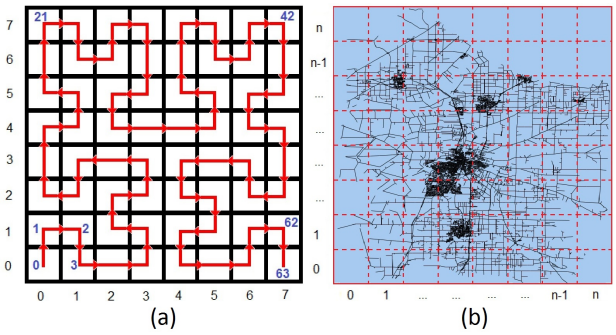

FIGURE 3 – Indexation des unités spatiales avec la courbe de Hilbert

gauche). Dans le cas d'un mouvement contraint (par un réseau de transport), l'indexation nécessite deux étapes. Tout d'abord, nous couvrons le réseau de transport par une grille uniforme (voir Figure 3 droite). La granularité de la grille est ajustée aux segments du réseau de telle sorte que la plupart des cellules couvrent approximativement un segment. Ensuite, les cellules de la grille sont indexées selon une courbe de Hilbert et chaque segment du réseau est marqué de l'indice de Hilbert de la cellule contenant le centre du segment. Dans le cas où plusieurs segments se retrouvent dans une cellule, les segments sont ordonnés selon les coordonnées x et y de leurs centres et étiquetés en conséquence. Une fois les unités spatiales indexées, elles sont triées selon la valeur de l'index et pondérées par le nombre de SP et la liste de couples (unité spatiale, poids) est stockée dans un vecteur pondéré, diffusé à tous les participants pour être utilisé dans la phase de partitionnement des sondes.

Vérification de la distribution et repartitionnement. Périodiquement, PAMPAS vérifie l'équilibre du partitionnement. En effet, la mobilité des participants fait qu'ils peuvent parfois changer de zone, qu'ils se connectent ou se déconnectent, ce qui peut entrainer un déséquilibre et le besoin de repartition-

ner l'espace. Quand la vérification du partitionnement est déclenchée, le système calcule un agrégat de comptage (count) en plus de la fonction d'agrégat de l'application (voir Requête de la figure 1). Le résultat global est alors envoyé à une SP choisie au hasard par le SSI. La SP chargée de la vérification décrypte les données et met à jour les poids du vecteur d'unités spatiales (lignes 4-7 de l'algorithme 3). Cette opération est en $O(N)$ où N étant le nombre d'unités spatiales. Dans le même temps, la SP calcule en mémoire le nombre d'utilisateurs par zone de la partition (les zones sont envoyées par le SSI, ligne 8 de l'algorithme 3) et l'écart-type qui capture le déséquilibre éventuel de la partition courante. Si ce déséquilibre se situe dans des limites prédéfinies, la SP envoie le résultat de comptage (sous forme cryptées via le SSI) aux autres SP afin d'actualiser les poids de leur vecteur d'unités spatiales. Dans le cas contraire, la SP chargée de la vérification lance un nouveau partitionnement.

L'algorithme de partitionnement des sondes est en $O(N)$ de temps d'exécution et en $O(P)$ en stockage (voir Algorithme 4). Pour définir les nouvelles frontières de zones, nous utilisons un algorithme glouton qui génère les zones à partir du vecteur pondéré en y insérant des unités spatiales (lignes 12-16 de l'algorithme 4) jusqu'à ce que le poids cumulé pour la zone courante soit dans la moyenne estimée du nombre de sondes par zone (ligne 10 de l'algorithme 4). Le résultat du partitionnement est une liste de p jalons indiquant les frontières des parties dans le vecteur pondéré des unités spatiales (ligne 16 de l'algorithme 4). Le résultat est ensuite crypté et envoyé via le SSI à toutes les sondes, qui mettent à jour leurs données de partitionnement et les répercutent sur les nouveaux échantillons de mesures échangés. Du fait de sa faible complexité, l'algorithme de partitionnement proposé peut être exécuté assez fréquemment (par exemple toutes les 30 secondes), et ce malgré les ressources limitées d'une SP. Cependant, il peut arriver que des déséquilibres persistes malgré tout, par exemple, lorsque certaines unités spatiales sont fortement déséquilibrées. Or, l'équilibre des zones est nécessaire pour éviter toute fuite d'information sur la localisation approximative des utilisateurs. Pour pallier ce problème, les SP génèrent, quand cela est nécessaire, des échantillons factices dans toutes les zone ayant le nombre de sondes est inférieur au nombre maximum toute zone confondue. Par conséquent, dans la période de collecte, une SP envoie avec une certaine probabilité un échantillon factice en plus de l'échantillon réel. La probabilité d'envoyer un échantillon factice est inversement proportionnelle au nombre de participants de la zone. La même approche est utilisée pour masquer le nombre d'unités spatiales dans chaque groupe. En effet, à la fin de la phase d'agrégation, chaque SP choisie pour calculer l'agrégation ajoute au résultat de sa zone des valeurs fictives en nombre égal à la différence entre le nombre maximum d'unités toutes zones confondue et le nombre d'unités de la zone considérée. De cette façon, tous les résultats partiels reçus par le SSI ont la même taille et le SSI ne peut donc déduire aucune information sur l'emplacement des zones à partir du nombre d'unités spatiales qu'elles contiennent.

Choix de la taille de la partition. Le coût du protocole d'agrégation est composé des coûts de calcul côté SP et de communication entre le SSI et les SP. La taille de la partition a un impact à la fois sur le coût de calcul et sur le coût de communication. Plus précisément, le coût de calcul diminue avec son augmentation et atteint la valeur minimale lorsque cette taille est égale au nombre d'unités spatiales, à savoir, lorsqu'une SP est utilisée pour agréger les échantillons d'une seule unité spatiale. Seulement, augmenter le nombre de zones

Algorithme 4 : Processus de repartitionnement (sur une SP)

```
1  PROBE_REPARTITIONING() :/* one randomly selected SP */
2      compute QI_comp and QI_comm for current P
3      while true do
4          /* adjust the partition size P */
5          if QI_comp > QI_comm then
6              tP = 2 * P
7          else
8              tP = P/2
9          /* repartition for tP */
10         avgGroupWeight = SUM(allCounts[])/tP
11         currentGroupWeight = 0
12         for i = 0 to NUM_SPATIAL_UNITS − 1 do
13             currentGroupWeight+ = allCounts[i]
14             if currentGroupWeight ≈ avgGroupWeight then
15                 newPartitionMilestones[].add(i)
16                 currentGroupWeight = 0
17         /* check if the new partitioning for tP is better than for P*/
18         compute tQI_comp and tQI_comm for tP
19         if tQI_comp + tQI_comm < QI_comp + QI_comm then
20             P = tP ; QI_comp = tQI_comp
21             QI_comm = tQI_comm
22         else
23             break
24     message = allCounts[]||newPartitionMilestones[]
25     send_for_broadcast(nE_k(message), SSI)
```

conduit à plus de déséquilibre, ce qui nécessite l'injection de plus d'échantillons factices et augmente le coût de communication. Donc, la modification du nombre de zones a un effet opposé sur le coût de calcul et le coût de communication.

$$QI_{comp} = Max_{i=1,P}[Comp\_time_i] - Max_{j=1,\#spatialUnits}[Comp\_time_j] \quad (1)$$

$$QI_{comm} = \frac{size(sample)}{bandwidth} \sum_{i=1}^{P} \{Max_{j=1,G}[Count_j(probes)] - Count_i(probes)\} +$$

$$\frac{size(sample)}{bandwidth} \sum_{i=1}^{G} \{Max_{j=1,P}[Count_j(spatialUnits)] - Count_i(spatialUnits)\} \quad (2)$$

Afin de parvenir à une optimisation globale, PAMPAS calcule deux indicateurs pour mesurer l'impact de la taille de la partition sur les coûts de calcul et de communication : QI_{comp} et QI_{pers} définis par les formules (1) et (2). QI_{comp} estime la dégradation du temps de calcul des SP résultant du fait que plusieurs unités spatiales sont déléguées à une SP au lieu d'utiliser une SP pour chaque unité spatiale. Estimer le temps de calcul est assez simple, car le temps est

globalement linéaire selon le nombre d'échantillons à traiter par la SP. QI_{pers} estime la dégradation du coût de communication causée par le déséquilibre des zones. Le premier terme indique le surcoût provoqué par les échantillons factices introduits pour équilibrer les zones, tandis que le second terme mesure le surcoût dû aux résultats factices générés pour équilibrer les unités spatiales par zone.

Ces indicateurs sont calculés lors du partitionnement (ligne 2 de l'algorithme 3). Si $QI_{comp} > QI_{comm}$, la SP multiplie par deux la taille de la partition par rapport à la précédente avant de recalculer la partition. Inversement, si $QI_{comp} > QI_{pers}$, la SP divise par deux le nombre de zones dans la partition à calculer. La SP continue d'ajuster la taille de la partition jusqu'à atteindre la valeur minimale de $QI_{comp} + QI_{pers}$ (lignes 19-24 de l'algorithme 4 et obtenir ainsi un coût global d'agrégation quasi-optimal.

5 Analyse de la sécurité

Les utilisateurs ne peuvent pas lire les données brutes des autres utilisateurs puisque les données en mémoire vive sont protégées par l'inviolabilité du MCU (à savoir, la mémoire RAM se trouve à l'intérieur du MCU) et les données stockées en mémoire Flash NAND sont protégées par du chiffrement.

Le SSI n'ayant pas la clé de chiffrement, il ne peut donc accéder aux données qui lui sont envoyées. En outre, le chiffrement non-déterministe des échantillons protège les données contre les attaques basées sur l'analyse de fréquence. Le SSI pourrait aussi essayer d'acheter une SP et passer pour un utilisateur afin d'accéder à la clé de chiffrement. Mais ce serait inutile puisque l'inviolabilité d'une SP protège la clé de chiffrement. Le SSI pourrait s'associer avec une application utilisant le protocole, mais il n'aurait accès qu'au résultat de l'agrégation. Enfin, étant donné que les échantillons sont communiqués au SSI de manière anonyme, celui-ci ne peut pas identifier les expéditeurs ou relier les messages consécutifs d'un même utilisateur.

On pourrait supposer que le SSI tente de déduire par analyse de fréquence des informations à partir des identifiants des zones, chiffrés de manière déterministe. Néanmoins, les zones étant équilibrées, il reçoit à peu près le même nombre de messages par zone. Par conséquent, le SSI n'a aucun moyen d'associer le participant à une zone ni à la configuration topologique des zones. Ainsi, la seule information que le SSI acquiert est le nombre de zones et de son évolution au fil du temps, ce qui ne présente aucun risque pour la privacité des participants. A noter que même si le SSI pouvait associer un participant à un identifiant chiffré de zone, il serait impossible de le distinguer des autres participants. Rappelons que les messages sont envoyés de manière anonyme afin qu'il soit difficile au SSI de les relier par provenance.

Par conséquent, les protocoles de PAMPAS sont sécurisés et protègent pleinement la vie privée des participants au système.

6 Evaluation expérimentale

Notre évaluation expérimentale vise deux objectifs principaux. Le premier est de mesurer le temps d'exécution et le passage à l'échelle du protocole d'agré-

gation de PAMPAS en prenant comme référence un protocole de l'état de l'art présenté dans [TNP14]. Le second objectif est de quantifier les performances du protocole de partitionnement. Nous avons implémenté et validé PAMPAS sur des dispositifs sécurisés dont la configuration matérielle est représentative des plates-formes matérielles sécurisées. Comme cas d'usage pour nos expériences, nous avons utilisé l'information communautaire sur le trafic routier et l'observation du bruit par des véhicules sondeurs. Nous avons utilisé à la fois des jeux de données synthétiques et réels de traces de localisations de participants véhiculés. Une démonstration de notre prototype a été présentée dans [TTSPZ15] illustrant le scénario de surveillance du trafic. La figure 4 illustre l'interface graphique type de ces applications. Elle montre la carte d'agrégation par interpolation du bruit et la carte du trafic dans un réseau routier. Une démonstration de notre prototype a été présenté dans [TTSPZ15] à l'aide d'un scénario de contrôle du trafic.

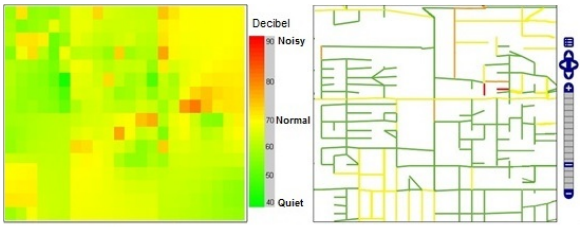

FIGURE 4 – Cartographies des agrégats pour deux applications : carte du bruit (à gauche) et carte du trafic (à droite)

6.1 Cadre expérimental

Plate-forme matérielle. Dans notre évaluation expérimentale, le SSI est hébergé sur un PC (3.1 GHz i5-2400, 8 Go de RAM, Windows 7) qui affiche également les résultats globaux sous forme graphique à des fins de validation. Les SP sont émulées par des dispositifs matériels sécurisés représentatifs (voir Figure 5). Un tel dispositif comprend un microcontrôleur avec un CPU 32-bit RISC à 120 MHz, un coprocesseur cryptographique capable d'exécuter AES et SHA, 128KB de RAM et 1MB de mémoire Flash NOR pour stocker la pile logicielle, une carte à puce hébergeant le matériel cryptographique (par exemple, les clés de chiffrement) et un lecteur de carte SD avec une grande capacité de stockage. Nous avons utilisé une carte SD classique (SDHC Samsung Essential Classe 10 de 32 Go) pour un stockage secondaire en mémoire Flash. Le SSI dans notre système de test utilise une connexion Ethernet multicanal avec une bande passante globale de 100 Mbps. A souligner que notre implémentation limite la consommation de RAM du côté des SP à seulement 30 KB et la bande passante maximum de communication avec le SSI à 200 Kbps afin de valider les protocoles proposés même sous des dispositifs sécurisés moins puissants. Pour émuler un très grand nombre de SP, nous avons exécuté de manière séquentielle sur une seule SP toutes les tâches d'agrégation et de communication avec le SSI, tandis que le temps d'exécution "parallèle" est estimé par le temps maximum au niveau

d'une SP. Les résultats expérimentaux ont également été validés [TTSPZ15] sur des sondes sécurisés connectées en parallèle au serveur.

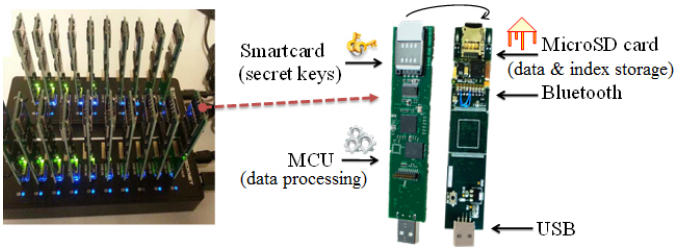

FIGURE 5 – Sondes sécurisées (SP)

Méthode de référence. Afin de souligner l'importance des protocoles de PAMPAS, nous avons implémenté le protocole sécurisé proposé dans [TNP14] et pris cette méthode comme référence. Ce protocole peut être appliqué sans modification pour agréger les échantillons prélevés dans chaque fenêtre temporelle. Notez que dans les travaux de [TNP14] , deux autres protocoles sont également proposés. Mais ces protocoles sont encore plus couteux à exécuter dans notre contexte que le protocole sécurisé.

Jeux de données et fonctions d'agrégation. Nous utilisons à la fois des données synthétiques et des données réelles afin de tester l'efficacité et le passage à l'échelle de PAMPAS. Nous avons utilisé le générateur de Brinkhoff [Bri02] pour générer des traces de mobilité sur deux réseaux réelle de la route des villes de Oldenburg en Allemagne et Stockton dans le comté de San Joaquin en Calofornie. Le réseaux d'Oldenburg est de petite taille et compte 7035 segments de route, tandis que Stockton comprend un réseau routier de taille moyenne comportant 24123 segments. En fonction de la taille du réseau, nous générons des traces de mobilité correspondant à un nombre moyen et grand d'utilisateurs. Pour Oldenburg, les jeux de données de taille moyenne et grande contiennent 47000 et 270000 utilisateurs respectivement. Pour Stockton, les jeux de données de taille moyenne et grande contiennent 200000 et 1,35 millions d'utilisateurs respectivement. La répartition spatiale des traces suit la densité spatiale de réseau. Par rapport aux jeux de données réels existants, les jeux de données synthétiques ont l'avantage d'avoir une excellente couverture spatiale et temporelle. Cependant, il est également important de valider les protocoles proposés avec de jeux de données réels. Pour ce faire, nous avons utilisé le jeu de données de trajectoires de taxi T-Drive [YZZ+10]. Cette base de données contient environ 15 millions d'unités de trajectoire collectées par 10357 taxis sur la période du 2 au 8 février 2008 à Pékin. Comme la densité de taxis est trop faible par rapport à l'ensemble de données synthétiques, nous avons extrait et fusionné une période d'une heure dans nos tests, afin de générer un jeu de données comparable, contenant 191000 unités couvrant 32800 segments de route.

Pour montrer la généricité de PAMPAS, nous avons sélectionné trois fonctions d'agrégation, à savoir, la moyenne, la médiane et IDW [NWL12]. Elles correspondent aux trois types d'agrégats décrits dans la Section 2.3. Nous associons les agrégats moyenne et médiane avec l'application de surveillance du trafic, à savoir, le calcul du temps moyen de déplacement et la vitesse médiane par segment de route. Par conséquent, ces deux scénarios considèrent le mouve-

ment contraint des sondes mobiles. L'agrégat IDW est associé à l'application de surveillance du niveau de bruit et considère un mouvement libre. Dans ce cas d'usage, nous avons exploité les mêmes traces de mobilité que précédemment, mais les avons considéré dans l'espace 2D dans un but d'expérimentation. En outre, nous utilisons une grille de 64x64 pour diviser l'espace 2D en un nombre limité de cellules (à savoir, 4096 cellules ou unités spatiales) pour le scénario du mouvement libre. Les valeurs d'échantillonnage de la vitesse sont générés par le générateur d'objets mobiles, tandis que les valeurs du niveau de bruit, nous les estimons de manière proportionnelle au nombre de sondes par cellule de la grille 2D.

6.2 Évaluation des performances

Temps d'exécution. La figure 6 indique le temps d'agrégation pour les trois fonctions pour les deux protocoles avec 200.000 sondes à Stockton. Le temps d'agrégation est mondiale, à savoir, il comprend à la fois le temps de calcul et de communication. Les résultats indiquent que PAMPAS est très efficace car il ne nécessite que quelques secondes pour calculer les résultats globaux pour toutes les fonctions testées. D'autre part, le protocole de base est beaucoup plus coûteux (en particulier pour les fonctions d'agrégation complexes) conduisant à des temps d'agrégation jusqu'à trois ordres de grandeur plus élevés que PAMPAS. Ces résultats démontrent que PAMPAS atteint l'objectif de travailler en temps réel (e.g., les conducteurs peuvent voir les conditions de circulation en temps réel).

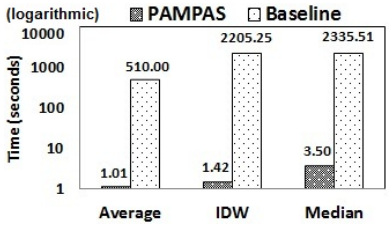

FIGURE 6 – Temps d'exécution de l'agrégation

Passage à l'échelle. Nous avons poussé les tests de passage à l'échelle des protocoles en variant le nombre de sondes, d'unités spatiales et de fonctions d'agrégat. La figure 7 montre le temps d'agrégation pour les deux protocoles pour les fonctions moyenne (graphique du haut) et médiane (graphique du bas) avec un nombre d'utilisateurs moyen et grand et sur deux réseaux routiers (Oldenbourg et Stockton). Les résultats confirment que seul PAMPAS passe à l'échelle dans toutes les configurations de paramètres d'entrée. Dans le pire des cas, le temps de calcul atteint 14 secondes pour la vitesse médiane sur 1,3 millions d'échantillons couvrant 24123 unités spatiales. Le protocole de base ne passe pas à l'échelle avec le nombre d'utilisateurs et encore moins avec le nombre d'unités spatiales. En pratique, le protocole de base ne peut agréger les échantillons en temps réel que pour un petit nombre d'unités spatiales (cas du réseau d'Oldenburg) et des fonctions d'agrégats pouvant être calculées de manière incrémentale (par exemple, la moyenne). La taille très limitées de RAM des SP et l'impossibilité de paralléliser le calcul d'agrégats rendent le protocole

de base inefficace pour les besoins des applications de collecte participative.

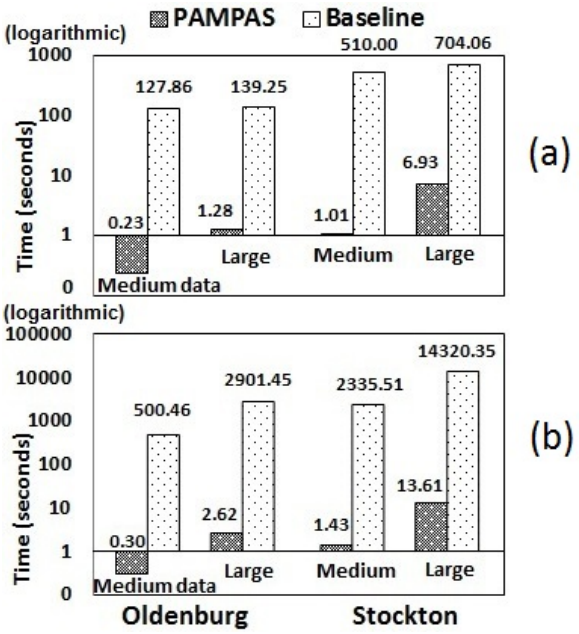

FIGURE 7 – Passage à l'échelle de PAMPAS et de la méthode de référence avec la fonction Moyenne (en haut) et la fonction Médiane (en bas)

Performances et passage à l'échelle du protocole de partitionnement. La figure 8 (haut) présente le temps de calcul du partitionnement dans les réseaux d'Oldenburg et de Stockton. Un nouveau partitionnement des sondes peut être calculé en quelques secondes par une SP. Cela signifie que la vérification et le repartitionnement des sondes peuvent être exécutés fréquemment, ce qui permet d'adapter le protocole d'agrégation de PAMPAS et maintenir son efficacité, même pour des changements rapides de distribution spatiale des sondes. La majeure partie du coût de partitionnement réside dans la lecture et l'écriture des données de partitionnement en mémoire Flash. Ceci explique également l'augmentation du temps de partitionnement avec le nombre de partitions, puisque dans ce cas, les opérations d'entrée/sortie sont exécutées à une granularité plus petite, ce qui est plus coûteux.

La figure 8 (bas) indique que le facteur de déséquilibre du partitionnement, à savoir, le rapport entre la taille maximum et la taille moyenne des parties, augmente avec le nombre de zones. Le facteur de déséquilibre est un indicateur important dans PAMPAS puisqu'un déséquilibre important implique une augmentation du nombre d'échantillons factices injectés par les SP et, par conséquent, augmente le coût de communication.

La figure 9 montre l'impact du nombre de partitions sur le temps global d'agrégation ainsi que le détail correspondant des temps de calcul et de communication. Le temps de calcul diminue avec le nombre de zones dans la partition car la quantité de travail effectué par les SP chargées de l'agrégation d'une zone diminue également. A l'inverse, le temps de communication augmente avec la taille de la partition, du fait que plusieurs échantillons factices sont injec-

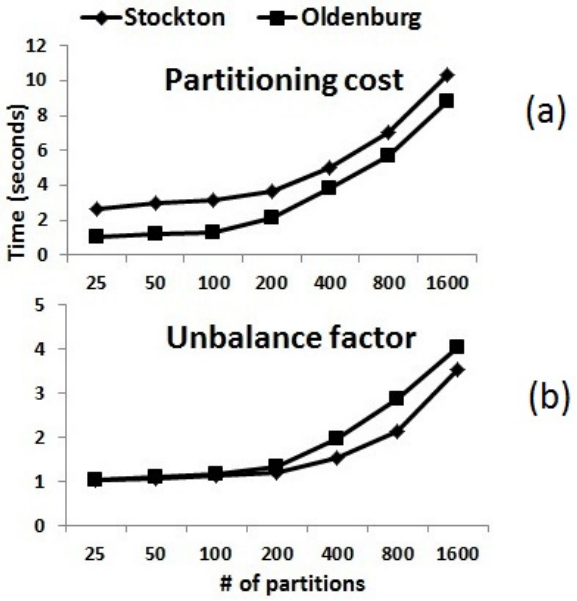

FIGURE 8 – Coût du partitionnement (en haut) et degré de déséquilibre des partitions (en bas) pour différentes tailles de partitions

tés dans le système, comme cela a été expliqué précédemment. Globalement, le temps d'agrégation quasi-optimal est obtenu avec le choix d'un partitionnement qui minimise la dégradation cumulée des coûts de calcul et de communication (voir la Section V). Nous avons obtenu des résultats similaires avec le jeu de données réelles pour lequel le nombre optimal de partitions est de 100 (le partitionnement du réseau a été calculé en seulement 2 secondes).

Discussion. Il est à noter que le temps d'agrégation peut être considérablement amélioré en augmentant la puissance de calcul et la bande passante de communication de le SSI. Par exemple, l'augmentation de la bande passante du serveur de 100 Mbps à 1 Gbps, fait baisser le temps d'agrégation maximum (utilisant la médiane sur le plus grand jeu de données simulé avec le réseau de Stockton) de 14 secondes à moins de 7 secondes. Par ailleurs, dans certains scénarios, pousser les calculs dans les dispositifs de l'utilisateur peut être problématique (par exemple, lorsque les dispositifs alimentés par batterie, ou si d'autres applications sont exécutées en parallèle dans le dispositif). Cependant, PAMPAS minimise ce type de problème grâce à sa grande efficacité. Par exemple, dans nos tests, un utilisateur participant au système pendant une heure a une probabilité comprise entre 3,5% et 8,7% de participer une fois à un calcul d'agrégat en supposant que les résultats d'agrégation sont produits toutes les 30 secondes, et une probabilité entre 0,004% et 0,12% d'effectuer un partitionnement en supposant que le partitionnement des sondes est vérifié chaque minute. Dans tous les cas, la calcul se fait en quelques secondes au maximum et se suffit de ressources modestes.

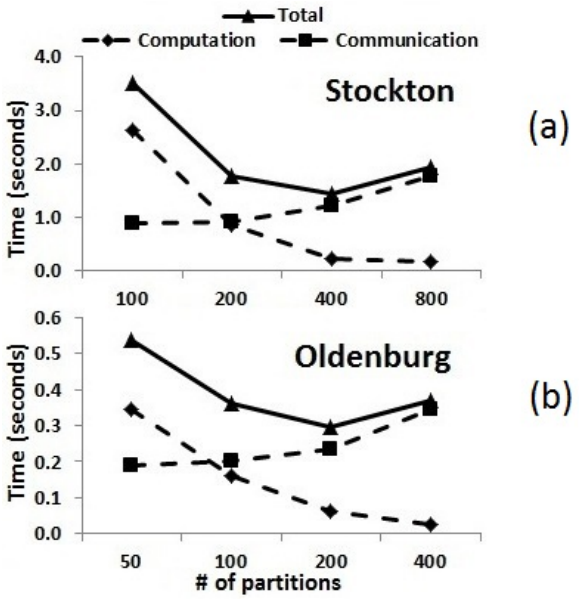

FIGURE 9 – Coût de communication et de calcul pour différentes tailles de partitions

7 Travaux connexes

Les architectures traditionnelles employées dans les systèmes de MPS reposent sur un serveur central puissant pour enregistrer les données collectées par les participants à travers un appareil mobile standard, pour les traiter, puis pour publier les résultats [DSJ13], [TRL+09]. Ce modèle est simple à développer et à déployer. Cependant, il soulève de sérieuses craintes et empêche une large adoption des collecte participatives, étant donné que la localisation d'un utilisateur permet facilement à un attaquant d'identifier les participants avec précision et d'inférer des connaissances sur leurs habitudes et leur mode de vie [dMHVB13]. Les travaux en rapport avec cette problématique sont de trois catégories : les architectures centrées serveur ; les protocoles cryptographiques ; et plus récemment, les dispositifs sécurisés mis en œuvre dans d'autres contextes applicatifs.

Approches centrées serveur. Dans cette catégorie, on cite la proposition de [HIJ+12] dans le contexte du trafic routier qui utilise une technique de masquage spatio-temporel selon un découpage virtuel du réseau routier par des lignes de comptage fictives, appelées Virtual Trip Line (VTL). Les véhicules sondeurs n'envoient les informations qu'au passage de ces VTL avec un délai suffisant pour les masquer. Afin d'éviter de relier l'identité du participant à sa localisation, ces deux informations sont séparées dans des serveurs différents. De même, le calcul et la maintenance des emplacements des VTL est effectué sur un troisième serveur. Les limitations de cette approche sont de trois types : (i) il n'est pas évident de générer des VTL qui garantissent l'anonymat et la précision (les auteurs se limitent au portions d'autoroutes excluant les abords des échangeurs) ; (ii) le système ne protège pas contre les risques de collusion entre

les serveurs qui le composent ; (iii) la construction de cette architecture est complexe et onéreuses. Un autre système est SpotMe [QLM+11] basé sur l'ajout de bruit. Le principe est que chaque participant envoie, en plus de sa localisation, un nombre fixe de localisations fictives. Plus ce nombre augmente, plus la localisation est anonyme. Le serveur central estime le nombre d'utilisateurs par zone en appliquant un modèle de probabilité. Cependant, cette approche souffre des problèmes suivants : (i) elle tend à sacrifier de manière significative la précision de l'agrégat au fur et à mesure qu'on augmente le bruit ; (ii) les données fictives augmentent les coûts de communication ; (iii) elle entraine surtout le risque de liage des données entre des envois successifs, ce qui limite son usage au cas de services par localisation sporadique, ce qui n'est pas adapté à notre contexte.

En somme, ces architectures tentent de répondre au problème de confiance dans un serveur central, mais ce faisant, elles se confrontent à de nombreuses limitations. En utilisant une architecture totalement décentralisée, PAMPAS évite ces problèmes. De plus, la confiance est assurée par un dispositif hautement sécurisé, inviolable et à faible coût.

Approches cryptographiques. Une autre manière de protéger les données des utilisateurs est d'appliquer des protocoles cryptographiques spécifiques. Les solutions de l'état de l'art [BOT13], [DEAS12], [PBBL11] reposent sur la propriété additive du chiffrement homomorphe permettant au composant - le serveur central dans [PBBL11] ou le mobile de l'utilisateur dans [DEAS12] - qui reçoit les données chiffrées de calculer des agrégats sans les décrypter. Toutefois, ces solutions présentent trois inconvénients majeurs. Le premier est qu'elles ne sont pas suffisamment génériques en terme de fonctions agrégats, limitant les agrégats aux fonctions simples comme le comptage, la somme ou la moyenne (les fonctions holistiques requièrent des mécanismes plus complexes et sont trop couteux à calculer avec les technologies actuelles). Le deuxième problème est le coût élevé de cryptages / décryptage, de calcul et de communication [BOT13], [PBBL11]. Ainsi, de telles solutions ne satisfont pas les critères de passage à l'échelle et de temps réel des systèmes de MPS. Enfin, le résultat d'agrégation peut perdre en précision du fait que les calculs ne s'appliquent que sur un espace très limité de valeurs dans la plupart des protocoles cryptographiques [BOT13], [PBBL11].

A la différence de ces travaux, PAMPAS est plus rapide tout en étant aussi sécurisé grâce à la sécurité matérielle employée.

Approches centrées utilisateurs et intégrant du matériel sécurisé. Des travaux récents ont proposé l'usage de dispositifs offrant une sécurité matérielle du côté de l'utilisateur pour protéger ses données personnelles, tout en contribuant à des traitements statistiques [ANP14], [TNP14]. Il s'agit d'une architecture distribuée où la confiance découle de deux techniques : (i) la décentralisation des calculs sur les dispositifs sécurisés des utilisateurs ou PDS , évitant de placer sa confiance dans un serveur central exposé aux malveillances ; (ii) le PDS est inviolable et protège des attaques physiques, y compris vis-à-vis de leur propriétaire. Dans [ANP14], les auteurs proposent METAP, un protocole générique pour la publication des données préservant la privacité, dans le contexte d'une architecture, dite asymétrique, où plusieurs dispositifs sécurisés mais peu puissants communiquent avec un serveur puissant mais qui n'est pas de confiance. METAP a été conçu pour la publication des données et non leur agrégation. Il ne tient pas compte du caractère spatio-temporel des observations et des besoins spécifiques du MPS. Quant aux travaux de To et al. [TNP14]. ba-

sés sur une architecture similaire, ils considèrent l'exécution de requêtes SQL sur une base de données distribuée sans révéler d'information individuelle au serveur central. Le protocole d'agrégation proposé s'applique pour différents types de calculs au sein d'une architecture centrée utilisateur [6]. Néanmoins, dans notre contexte où les données sont collectées en flux et en continu, cette solution engendre des coûts de traitement et de communication élevés, qui augmentent avec le nombre de participants et/ou de groupes d'agrégation. Il existe trois raisons à cela : (i) pour agréger les données d'un même groupe, plusieurs itérations sont nécessaires pour les agréger au sein d'un même PDS ; (ii) le nombre de groupes étant élevé (par exemple, égal au nombre de sections de routes d'une grande agglomération), leur traitement lors d'un cycle de calcul risque fort de dépasser la taille en mémoire RAM des PDS ; (iii) dans le cas d'agrégats holistiques ou complexes, comme l'agrégation n'est pas incrémentale, il est nécessaire de réunir tout l'échantillon par groupe avant de le traiter, ce qui affecte encore plus le passage à l'échelle du système. PAMPAS partage la même idée que les systèmes précédents quant à l'utilisation d'une architecture centrée utilisateur. Toutefois, le protocole d'agrégation est adapté aux besoins et spécificités du contexte MPS, à savoir l'aspect fortement dynamique des données et la mobilité des participants, les contraintes temps-réel pour répondre à des requêtes continues, et la complexité des agrégats et leur nature spatio-temporelle. Il existe également des solutions centralisées offrant une sécurité matérielle visant à protéger l'exécution des applications sur un serveur standard. Par exemple, Haven [BPH14] étend le niveau de protection matériel fourni par l'architecture Intel SGX, à l'origine sur des morceaux de codes, au système d'exploitation entier. Mais cette solution ralentit le calcul considérablement et ne supporte pas certaines fonctionnalités du système utiles au développement d'applications, comme la création de processus. Par ailleurs, la sécurité n'est pas tout à fait garantie, car elle dépend de la capacité du fabricant de la puce sécurisée à protéger les clés secrètes.

8 Conclusion

Cet article propose PAMPAS, un système basé sur une architecture distribuée et des mobiles sécurisés pour une collecte participative de données spatiales en mobilité réellement anonyme. Cette combinaison permet à PAMPAS d'atteindre le même niveau de confidentialité que les solutions cryptographiques sans avoir à sacrifier la généricité, la précision et le passage à l'échelle. La solution d'agrégation proposée est, à notre connaissance, la première proposition d'un protocole distribué qui soit sécurisé, efficace et performante et qui corresponde à la fois aux contraintes matérielles strictes de dispositifs personnels sécurisés et aux contraintes de temps-réel des applications de collecte participative. L'évaluation expérimentale basée sur des dispositifs matériels sécurisés représentatifs pour les plates-formes sécurisées valide la solution proposée.

6. Etant donné que tous les traitements sont effectués du côté utilisateur, le serveur ne sert que de support pour ce protocole

Références

[ANP14] Tristan Allard, Benjamin Nguyen, and Philippe Pucheral. Metap : revisiting privacy-preserving data publishing using secure devices. *Distributed and Parallel Databases*, 32(2) :191–244, 2014.

[ARM09] ARM. *ARM Security Technology - Building a Secure System using TrustZone Technology.* ARM Technical White Paper, 2009.

[BOT13] Joshua W. S. Brown, Olga Ohrimenko, and Roberto Tamassia. Haze : privacy-preserving real-time traffic statistics. In *ACM SIGSPATIAL*, pages 540–543, 2013.

[BPH14] Andrew Baumann, Marcus Peinado, and Galen Hunt. Shielding applications from an untrusted cloud with haven. In *OSDI*, pages 267–283, 2014.

[Bri02] Thomas Brinkhoff. A framework for generating network-based moving objects. *GeoInformatica*, 6(2) :153–180, June 2002.

[DEAS12] George Drosatos, Pavlos S. Efraimidis, Ioannis N. Athanasiadis, and Matthias Stevens. A privacy-preserving cloud computing system for creating participatory noise maps. In *COMPSAC*, pages 581–586, 2012.

[dMHVB13] Yves-Alexandre de Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd : The privacy bounds of human mobility. *Scientific reports*, 3, 2013.

[DSJ13] Ellie D'Hondta, Matthias Stevensb, and An Jacobs. Participatory noise mapping works ! an evaluation of participatory sensing as an alternative to standard techniques for environmental monitoring. *Pervasive and Mobile Computing*, 9(5) :681–694, October 2013.

[FNSA12] Miad Faezipour, Mehrdad Nourani, Adnan Saeed, and Sateesh Addepalli. Progress and challenges in intelligent vehicle area networks. *Magazine Communications of the ACM*, 55(2) :90–100, 2012.

[GLW+15] Hui Gao, Chi Harold Liu, Wendong Wang, Jianxin Zhao, Zheng Song, Xin Su, Jon Crowcroft, and Kin K. Leung. A survey of incentive mechanisms for participatory sensing. *IEEE Comm. Surveys and Tutorials*, 17(2) :918–943, 2015.

[Goo07] Michael F Goodchild. Citizens as sensors : the world of volunteered geography. *GeoJournal*, 69(4) :211–221, 2007.

[HIJ+12] Baik Hoh, Toch Iwuchukwu, Quinn Jacobson, Daniel Work, Alexandre M. Bayen, Ryan Herring, Juan Carlos Herrera, Marco Gruteser, Murali Annavaram, and Jeff Ban. Enhancing privacy and accuracy in probe vehicle-based traffic monitoring via virtual trip lines. *IEEE Tran. on Mobile Computing*, 11(5) :849–864, 2012.

[JMS+08] Namit Jain, Shailendra Mishra, Anand Srinivasan, Johannes Gehrke, Jennifer Widom, Hari Balakrishnan, Ugur Çetintemel, Mitch Cherniack, Richard Tibbetts, and Stanley B. Zdonik. Towards a streaming sql standard. In *PVLDB 1(2)*, pages 1379–1390, 2008.

[NWL12] Silvia Nittel, J. C. Whittier, and Qinghan Liang. Real-time spatial interpolation of continuous phenomena using mobile sensor data streams. In *ACM SIGSPATIAL*, pages 530–533, 2012.

[PBBL11] Raluca Ada Popa, Andrew J. Blumberg, Hari Balakrishnan, and Frank H. Li. Privacy and accountability for location-based aggregate statistics. In *CCS*, pages 653–666, 2011.

[Pen14] Michele Penza. Cost action td1105 : New sensing technologies for environmental sustainability in smart cities. In *IEEE SENSORS*, 2014.

[QLM+11] Daniele Quercia, Ilias Leontiadis, Liam Mcnamara, Cecilia Mascolo, and Jon Crowcroft. Spotme if you can : Randomized responses for location obfuscation on mobile phones. In *ICDCS*, pages 363–372, 2011.

[SWS+13] Emily G Snyder, Timothy H Watkins, Paul A Solomon, Eben D Thoma, Ronald W Williams, Gayle SW Hagler, David Shelow, David A Hindin, Vasu J Kilaru, and Peter W Preuss. The changing paradigm of air pollution monitoring. *Environmental science & technology*, 47(20) :11369–11377, 2013.

[TNP14] Quoc-Cuong To, Benjamin Nguyen, and Philippe Pucheral. Privacy-preserving query execution using a decentralized architecture and tamper resistant hardware. In *EDBT*, pages 487–498, 2014.

[TRL+09] Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Samuel Madden, Hari Balakrishnan, Sivan Toledo, and Jakob Eriksson. Vtrack : accurate, energy-aware road traffic delay estimation using mobile phones. In *ACM SenSys*, pages 85–98, 2009.

[TTSPZ15] Dai-Hai Ton-That, Iulian Sandu-Popa, and Karine Zeitouni. PPTM : Privacy-aware participatory traffic monitoring using mobile secure probes. In *IEEE MDM*, 2015. Demo paper.

[YZZ+10] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive : driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*, pages 99–108. ACM, 2010.