ACCIO: How to Make Location Privacy Experimentation Open and Easy

Vincent Primault, Mohamed Maouche, Antoine Boutet, Sonia Ben Mokhtar, Sara Bouchenak, Lionel Brunie

ACCIO: How to Make Location Privacy Experimentation Open and Easy

Vincent Primault*, Mohamed Maouche†, Antoine Boutet‡, Sonia Ben Mokhtar†, Sara Bouchenak†, Lionel Brunie†
*University College London, United Kingdom
v.primault@ucl.ac.uk
†Univ Lyon, INSA Lyon, LIRIS, UMR5205, CNRS, F-69621 Villeurbanne, France
{mohamed.maouche,sonia.benmokhtar,sara.bouchenak,lionel.brunie}@insa-lyon.fr
‡Univ Lyon, INSA Lyon, Inria, CITI, F-69621 Villeurbanne, France
antoine.boutet@insa-lyon.fr

***Abstract*—The advent of mobile applications collecting and exploiting the location of users opens a number of privacy threats. To mitigate these privacy issues, several protection mechanisms have been proposed this last decade to protect users' location privacy. However, these protection mechanisms are usually implemented and evaluated in monolithic way, with heterogeneous tools and languages. Moreover, they are evaluated using different methodologies, metrics and datasets. This lack of standard makes the task of evaluating and comparing protection mechanisms particularly hard.**

In this paper, we present ACCIO, a unified framework to ease the design and evaluation of protection mechanisms. Thanks to its Domain Specific Language, ACCIO allows researchers and practitioners to define and deploy experiments in an intuitive way, as well as to easily collect and analyse the results. ACCIO already comes with several state-of-the-art protection mechanisms and a toolbox to manipulate mobility data. Finally, ACCIO is open and easily extensible with new evaluation metrics and protection mechanisms. This openness, combined with a description of experiments through a user-friendly DSL, makes ACCIO an appealing tool to reproduce and disseminate research results easier. In this paper, we present ACCIO's motivation and architecture, and demonstrate its capabilities through several use cases involving multiples metrics, state-of-the-art protection mechanisms, and two real-life mobility datasets collected in Beijing and in the San Francisco area.

I. INTRODUCTION

The advent of smart and always-connected handheld devices opens the way to a large variety of online services, such as location-based services (LBSs for short), which provide users with contextual and personalised answers. Location data is extensively collected by mobile applications for commercial or advertisement purposes [1].

However, analysing mobility data can reveal extensive information about the associated owners, such as where they live [2], their daily activities [3] or more sensitive information (e.g., their religion [4]). To mitigate these privacy issues, many location privacy protection mechanisms (LPPMs for short) have been proposed this last decade (e.g., [5]–[9]). These LPPMs are typically used to protect (or sanitize) mobility datasets gathered by an LBS before publishing, exchanging or selling them to a third party (e.g., a marketing company or an advertiser). Moreover, this protection process has recently become compulsory for all companies and institutions handling personal data of European citizens with the new European General Data Protection Regulation [10], which has come into effect in May 2018. The goal of LPPMs is two-fold: *enhancing privacy* to protect users while *preserving utility* in the resulting sanitized data. Indeed, the goal is not to prevent the collection or usage of mobility data, but to do it while preserving the privacy of users. However, due to the conflicting nature of these two dimensions, there is a clear trade-off between privacy and utility. For instance, a marketing company buying such a protected mobility dataset should not be able to re-identify the users or infer personally sensitive information about the users (e.g., where they live, what is their occupation) but it should still be able to extract some relevant signals it is interested in (e.g., the activity patterns around a specific shop).

The offered privacy and utility guarantees are generally evaluated by LPPMs designers, either theoretically by proving a set of theoretical properties (e.g., k-anonymity [11] or differential privacy [12]), or experimentally by relying on ad-hoc metrics, which largely vary from one LPPM to another. Moreover, evaluating an LPPM is often performed by a monolithic code, i.e., designed only towards this purpose, and which is not always made available by their authors. To illustrate this issue, we analysed nine state-of-the-art and representative LPPMs (summarised in Table I and further discussed in Section III) and pointed out that no LPPMs are evaluated using the same metrics and the same datasets.

To deal with the issue of evaluating LPPMs, few solutions have been proposed in the literature. For instance, Location Privacy Meter [25] is a framework designed to quantify location privacy. However, it has a strict underlying probabilistic model that does not accommodate the large variety of LPPMs and metrics that exist in the literature. GEPETO [26], in turn, aims at visualising the impact of LPPMs and attacks through a graphical user interface. However this tool only focuses on re-identification attacks and does not allow the automation of experiments. Moreover, both tools can only be executed on a single machine, and therefore cannot exploit any clusters of computing resources to speed up execution of long-running experiments. From these observations, it clearly appears that

TABLE I
EXAMPLES OF EXISTING LPPMS, EVALUATED USING HETEROGENEOUS METRICS AND DATASETS.

		Datasets								**Privacy**		**Utility**		**Performance**	
LPPM category	**LPPM**	Cabspotting [13]	MDC [14]	Geolife [15]	D4D [16]	CENSUS [17]	Gowalla [18]	Proprietary	Synthetic	Attack correctness	Data distortion	Data distortion	Query distortion	Execution time	Scalability
k-**anonymity**	*Never Walk Alone* [19]							✓	✓		✓		✓		✓
	$\mathcal{W}4\mathcal{M}$ [20]							✓	✓		✓	✓	✓	✓	✓
	GLOVE [5]				✓							✓		✓	
Differential privacy	GEO-I [6]					✓							✓		
	Differentially private grids [7]					✓	✓						✓		
	Jiang et al. [21]							✓				✓			
Other approaches	Path confusion [22]								✓	✓		✓			
	Promesse [23]	✓	✓	✓						✓		✓	✓	✓	
	Plausible synthetic traces [24]		✓							✓			✓		

there is a lack of a common platform to evaluate existing and new LPPMs, and to perform comparative studies using novel datasets and evaluation metrics. This poses real and important limitations for the progresses and the reproduction of results in this research area .

To overcome these limitations, we propose in this paper ACCIO, an open and extensible location privacy experimentation platform enabling researchers to quickly design, launch and disseminate reproducible location privacy experiments. An ACCIO experiment consists of a workflow described using a Domain Specific Language (DSL) we designed to facilitate the configuration and deployment of sophisticated experiments. More precisely, a workflow is a composition of parameterisable operators which can be any mobility data manipulation routine, whether it is an LPPM, a privacy or utility evaluation metric (at the time of writing, 28 such operators are implemented in ACCIO, and new ones can be easily added). These workflows are then executed using on the ACCIO platform, which can scale out from a single machine to a distributed deployment on a cluster of machines or a custom cloud infrastructure. Finally, our framework allows to quickly analyse results through a Web interface, as well as to export them in simple text files for a later analysis with custom tools (e.g., Matlab or Python)

In this paper, we demonstrate the capabilities of ACCIO through three different use cases. We evaluate three very different state-of-the-art LPPMs (GEO-I [6], $\mathcal{W}4\mathcal{M}$ [20] and PROMESSE [23]) using various metrics and datasets. We show that those algorithms can be unified under concepts provided by our framework, and expressed elegantly with a few lines of our DSL. The openness of ACCIO combined with its capability to describe experiments using user-friendly DSL makes it an interesting tool for practitioners and researchers to disseminate reproducible results. Finally, ACCIO is available for download: *https://privamov.github.io/accio/*

The remaining of this paper is organised as follows. We start by presenting the problem statement in Section II before reviewing related works in Section III. We then describe ACCIO and its architecture in Section IV. We present our experimental setup in Section V, and the three use cases considered to demonstrate the capability of ACCIO in Sections VI-A to VI-C. We finally conclude in Section VII.

II. PROBLEM STATEMENT

Location privacy protection mechanisms attempt to protect users against privacy threats coming from their mobility data. LPPMs are algorithms that transform raw mobility datasets into sanitized mobility datasets. Many LPPMs have been proposed these last years. LPPMs can be classified according to the guarantees they offer to the users. The two most adopted privacy guarantees are k-anonymity [11] and ϵ-differential privacy [12]. While the former hides a user within cloaking areas containing at least $k-1$ other users (e.g., GLOVE [5]), the latter disturbs mobility data in such a way that it theoretically bounds by a factor ϵ the impact of one record (i.e., his presence or absence) on the result of a specific processing (e.g., GEO-I [6]). There are also LPPMs that do not fall into the above two categories (e.g., [23], [24]) but focus on enforcing practical guarantees such as protecting specific sensitive data.

LPPMs are usually evaluated either theoretically or practically using real or synthetic mobility datasets through a set of metrics. Only a few publicly available mobility datasets exist, the others are unfortunately proprietary and not shared. Evaluation metrics include privacy, utility and performance metrics. While privacy metrics quantify the effectiveness of the mechanism to protect user privacy, utility metrics quantify its ability to leave the data useful from the point of view of a data analyst. Privacy and utility are highly conflicting goals, which makes finding an acceptable trade-off between them a difficult task. Finally, performance metrics quantify the running effectiveness of a protection mechanism.

Table I shows a set of nine representative state-of-the-art LPPMs, along with the metrics and datasets that were used by their authors to evaluate them. The objective of this

table is not to provide an exhaustive survey of state-of-the-art LPPMs [1], but rather to demonstrate the heterogeneity problem we are addressing in this paper. Indeed, a quick bird's-eye view shows that metrics and datasets largely vary from one LPPM to another. This heterogeneity makes it difficult to fairly compare LPPMs. Indeed, a given LPPM can perform very well under a particular combination of metrics and datasets while being less effective in a different setup. For instance, this aspect is demonstrated in [28] where GEO-I is evaluated with privacy metrics (namely "Attack correctness" and "Data distortion") different from those used by their authors. This study eventually showed that efficiently protecting privacy, with respect to those particular metrics, was only achievable by setting a very strong privacy parameter, which came at the cost of a very degraded utility.

To summarise, different metrics and datasets are used to evaluate LPPMs, thus making their comparison very difficult. It becomes even worse when metrics are not clearly defined, or when using proprietary datasets that are not publicly available. The challenge addressed by ACCIO is hence three-fold. First, we want to achieve *reproducibility* of published results. Second, we want to achieve *reusability*. Indeed, researchers should not have to reimplement again and again the same algorithms, and should instead reuse existing and interoperable building blocks. Third, we want to achieve *extensibility*, by allowing to easily integrate a new algorithm and compare it to the state-of-the-art solutions.

III. RELATED WORKS

In this section, we review existing works addressing the challenge of easing the evaluation of LPPMs.

Location privacy frameworks. Shokri et al. proposed a fully-fledged framework designed specifically to evaluate location privacy [25]. Performance of an LPPM is quantified by comparing the outcome of a privacy attack performed on both the raw mobility trace and on its protected counterpart. The whole evaluation process is split in five steps: 1) reading data, 2) simulating an application, 3) applying an LPPM, 4) executing an attack, and 5) evaluating its efficiency with a metric. Each step can be replicated to compare different datasets, LPPMs, attacks or evaluation metrics. Towards this purpose, they propose several new attacks and formally define three evaluation metrics: accuracy, certainty and correctness. They actually implemented their framework as a tool and released it under an open source license, along with its documentation [29]. However, this solution only works for probabilistic LPPMs and is not adapted to more generic mechanisms such as $\mathcal{W}4\mathcal{M}$ or PROMESSE (i.e., limited *reusability*). Furthermore, it only considers privacy when evaluating an LPPM and does not consider utility nor the associated privacy/utility trade-off. *GEPETO* [26] is a tool to study location privacy proposed by Gambs et al. This tool allows to apply several LPPMs on mobility datasets, and launch privacy attacks. It focuses on visualisation by providing a graphical user interface to display on a map results of algorithms. However, this tool does not provide clear extension points to integrate new pieces of code (i.e., limited *extensibility*), and as a GUI application does not allow to easily script experiments (i.e., limited *reproducibility*). It is worth noting that GEPETO's authors later extended their tool to scale out to large datasets by leveraging MapReduce [30]. Conversely, the workflow-oriented structure of ACCIO combined with its new DSL achieves together reproducibility, reusability and extensibility.

Scientific workflow tools. Although our goal is to specifically support location privacy research, there are some similarities with generic scientific workflow management systems. The latter are used to model experiments, launch them on distributed architectures (e.g., a grid or a cloud), and provide access to the results. They are more often used in disciplines such as bioinformatics and astronomy. Pegasus [31] reads workflows from XML files, in addition to providing programmatic APIs for generating these files. This tool also comes with a Web interface to monitor and debug executions. Swift [32] provides a language roughly similar to C to describe computations. It is then compiled and automatically parallelised when possible. Kepler [33] comes with a desktop application to create and execute workflows. It allows to visually connect operations and see how they interact. However, all these tools come with a set of operations targeted towards astronomy or chemistry, and not for spatio-temporal datasets and location privacy. The survey of Liu et al. [34] gives and extensive view about scientific workflow management systems.

IV. ACCIO DESIGN

In this section, we present the design of ACCIO. As shown in Figure 1, ACCIO is composed of several components. Users interact with the server through a REST API which in turn controls the experiment and its deployment as well as the results. ACCIO is implemented and publicly available under the GNU GPL v3 licence, and comes with a documentation website [35]. It is made of almost 20,000 lines of code written mainly in Scala (a language running in the Java Virtual Machine) for the server, and about 3,000 additional lines of Javascript for the Web interface.

ACCIO experiments are described as workflows linking together basic building block called operators. We first introduce the concepts of operator (Section IV-A) before to present the workflow (Section IV-B). We then explain the architecture of ACCIO and the deployment of an experiment in Section IV-C. Finally, we present how ACCIO can be extended in Section IV-D.

A. Operators

An *operator* is the basic building block of ACCIO. It acts as a function in a program: given some inputs, it produces some outputs. Each operator comes with a very clearly defined interface: it defines the inputs it consumes and the outputs it produces beforehand. Because inputs and outputs are strongly typed, values are checked for correctness before actually executing operators. Inputs and outputs have a name, a type

[1] Review of state-of-the-art LPPMs is addressed in surveys such as [27].

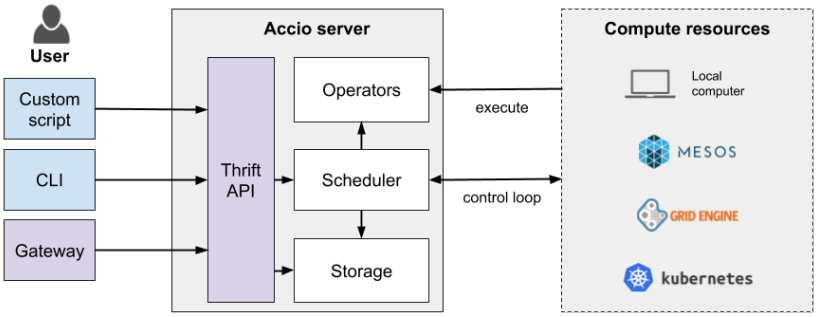


Fig. 1. Overview of ACCIO's architecture: users use a client application to interact with the server, which in turn interacts with a library of ACCIO operators deployed on some computing resources.

and possibly a default value (for inputs only). An operator then consumes zero or several inputs and produces one or several outputs. The outputs generated by the execution of an operator are automatically collected and ingested back into ACCIO. An example of operator is reported in Figure 2.

- **Name:** GEO-I
- **Input:** [White] Epsilon, a strictly positive floating number, which controls the privacy level
- **Input:** [Black] A single mobility dataset to protect
- **Output:** [Grey] A single protected mobility dataset

Fig. 2. Example of operator implementing GEO-I LPPM.

Operators are the basic execution units of ACCIO and are always executed on a single machine. They define the computing resources they need (e.g., the number of CPU cores or the quantity of RAM) to execute properly. Operators are stateless, should be side-effect-free and are assumed to be deterministic. These constraints enforce the reproducibility. It means that given some inputs, an operator should produce the exact same outputs at each execution. Hopefully, we support injecting some randomness through the notion of unstable operators. The latter are given an initial seed, that can be used later to initialise a pseudo-random number generator in a predictable state, and hence use some kind of controlled randomness.

Because of their simple interface, operators support a large variety of algorithms, including LPPMs and evaluation metrics. ACCIO can accommodate LPPMs working with datasets of various sizes and shapes, as well as metrics evaluating different privacy (from the classical k-anonymity and differential privacy guarantees to metrics relying on elaborate attacks) and utility (from simple information theoretical quantifications to metrics relying on sophisticated data mining tasks) aspects.

Operators need to be implemented by developers following a simple API. We provide more details about how to extend ACCIO with new operators in Section IV-D.

B. Workflows

Experiments in ACCIO are described as *workflows*. A workflow is a list of steps, where each step corresponds to an instance of an operator with a set of inputs (specified either directly or as a dependency to another step). These steps then form a directed acyclic graph, where the edges represent the data flowing between the steps. An example of workflow used to evaluate the GEO-I LPPM is depicted in Figure 3, the considered operators are briefly presented in Section V-C. There is a single root step (i.e., with no upstream dependencies to another step), `DatasetReader` whose goal is to read a dataset stored somewhere (e.g., local disk or Amazon S3), and convert it into a standardised format that all the operators will understand. This dataset is then pre-processed by the `TemporalSampling` and `TemporalGapSplitting` steps, in a sequential fashion. Then, the `GeoI` step eventually produces a protected dataset from the pre-processed dataset. Finally, the evaluation is done in two steps: `PoisRetrieval` (privacy metric) and `SpatialDistortion` (utility metric). Each of those steps needs two inputs: a baseline (i.e., unprotected) dataset and a protected dataset to be compared with that baseline. Because of there is no dependencies between these two evaluation steps, they can be executed in parallel. Lastly, the

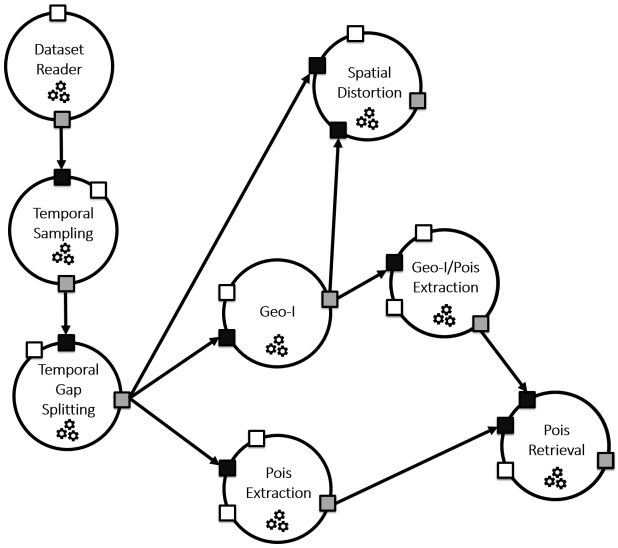


Fig. 3. Example of an ACCIO workflow that evaluates GEO-I LPPM with two metrics. It describes experiments by linking operators together.

`PoisExtraction` and `GeoI/PoisExtraction` steps are intermediate steps whose goal is to extract points of interest (i.e., all places where users stopped and spent some time [3]), which will be used by the privacy metric.

When specifying a workflow, one essentially defines a list of steps and how to connect them together, where each step is a particular instance of an operator. To provide reusability, the same workflow can be launched multiple times with different parameters. While operators need to be implemented by developers, workflows are represented either in JSON or in our DSL, thus allowing anyone with basic scripting knowledge to write such a workflow. In addition, reproducibility of results are made easier by only disseminating ACCIO workflows. Lastly, workflows are instantiated through *jobs*, where a job is a single execution of a workflow with a given set of parameters. We also support launching a batch made of several jobs at once, thus allowing to test tens or hundreds different combinations of parameters.

C. Architecture of ACCIO

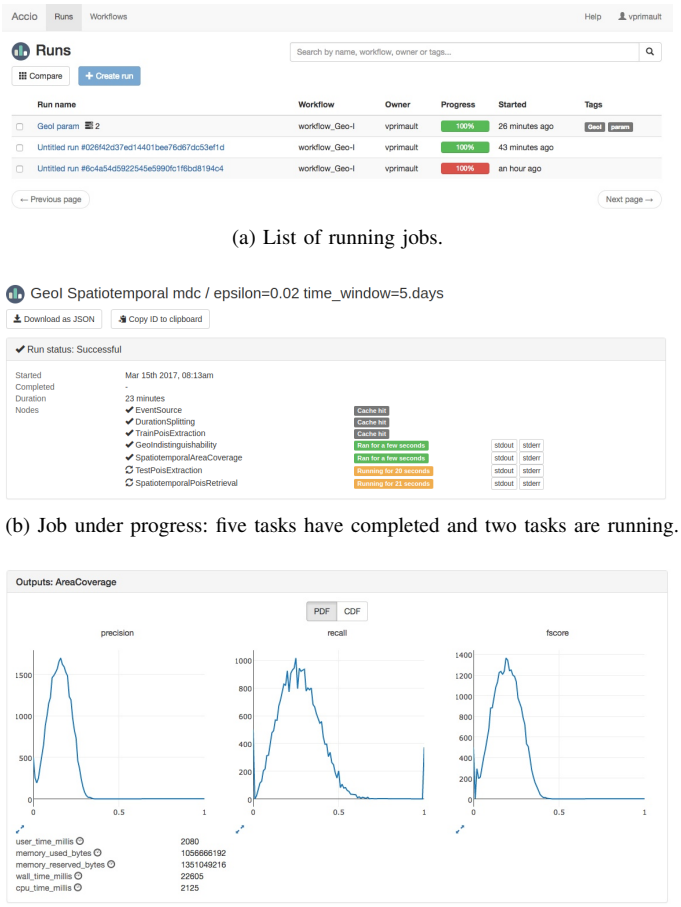
As depicted in Figure 1, the core component of ACCIO is the server which exposes a Thrift API[2] to users. These users exploit this API to create their experiments (i.e., workflows and jobs), monitor progress and download results once they are completed. We also provide a command-line application as well as a Web interface to facilitate management. The ACCIO server consists of the following main components:

CLI & Web interfaces. ACCIO ships both with a command-line client and a Web application. While the web interface is more targeted towards visually list experiments (Figure 4a), monitoring the progress of jobs (Figure 4b), and previewing results (Figure 4c), the command-line client is more suited for exporting the whole results as CSV for a more detailed analysis. ACCIO does not intend to be a full-fledged analysis framework; we prefer to let the users have the control over the tools they want to leverage, whether it is Excel, Python or R.

Operators. A library of operators provide a definition and an implementation of a set of operators. Built-in operators include pre-processing routines (e.g., `TemporalSampling`), LPPMs (e.g., `GEO-I`) and evaluation metrics (e.g., `SpatialDistortion`). At the time of writing, ACCIO comes with 28 such operators, which correspond to 3 LPPMs, 10 metrics, 14 pre-processing routines and one dataset reader. A library of operators is a simple binary that can be discovered and loaded by the server. The server uses them to validate the correctness of workflows, while the computing cluster needs these libraries to actually execute the operators.

Scheduler. A scheduler is an interface to interact with computing resources. A job is typically split into a list of tasks, each task corresponding to a step inside the workflow, and hence an instance of an operator. Because each operator declares its required resources, several tasks can be executed in parallel, as long as computational resources are available in the targeted resources (e.g., cluster). In practice, tasks belonging to the same jobs without any dependencies between them, as well as tasks from different jobs can be all executed in parallel, thus resulting in maximising the resource usage. As developing a new scheduler is outside the scope of this work, the scheduler can be implemented by relying on well-known resource managers such as Mesos [37] or HTCondor [38][3]. ACCIO also comes with a simple local scheduler, where tasks are ran as sub-processes of the main server process.

Storage. Finally, the storage is responsible for persisting jobs and tasks. We currently provide an in-memory storage (mainly used for testing) and a MySQL storage.



(a) List of running jobs.

(b) Job under progress: five tasks have completed and two tasks are running.

(c) Previewing the results of a job.

Fig. 4. ACCIO comes with a Web interface allowing the user to monitor the progress of jobs.

D. Extending ACCIO

ACCIO has been designed to be extensible, which means that several components have pluggable implementations: storage, scheduler and operators. In this section, we give a hint about how custom operators can be implemented.

At its heart, an operator is just a binary obeying to a simple protocol: when executed, it receives as command-line

[2]Thrift is an Interface Description Language allowing to describe and generate code for RPC servers and clients [36]. This allows us to build a type-safe and efficient binary communication protocol.

[3]Those are not actually implemented in ACCIO, but we plan to support external computing clusters in the future.

argument the Thrift-seralised context of execution. The latter includes the values of inputs, the seed allowing to inject controlled randomness, and where to write the outputs. It should then take care of returning a 0 exit code once successfully completed, or anything else if the execution failed in some way. Operators can be liberally used the standard output and error streams, which may be captured and made available to the users. Because these requirements correspond to the standard Unix way of doing things, operators can virtually be implemented in any language. In practice, we provide a Software Development Kit to implement operators in Scala through a lightweight interface.

V. Experimental setup

Before to demonstrate the capacity of Accio through different use cases in next sections, we present in this section the considered experimental setup. We present the datasets we use (Section V-A), the runtime environment (Section V-B) and the relevant operators (Section V-C). This experimental setup will then be used throughout the three different use cases summarised in Table II.

TABLE II
Demonstration of the soundness of Accio through three use cases.

Use case	#LPPMs	#Datasets	#Metrics
1: Metric diversity	1	1	5
2: Dataset diversity	1	2	5
3: LPPM diversity	3	1	5

A. Datasets

A dataset of mobility data is a list of spatio-temporal records where each record is the location of a given user at a given time. These records are usually partitioned into mobility traces, where each trace is the list of records belonging to a given user. Real-life or synthetic datasets can be used to evaluate LPPMs. While the former ensure to exhibit realistic behaviours but are usually small and can be sparse, the latter can be generated with any size. We use two real-life datasets to evaluate Accio: Geolife and Cabspotting. The Cabspotting dataset [13] contains GPS traces of taxi cabs in San Francisco (USA), while the Geolife dataset [15] gathers GPS trajectories of people during they daily life in Beijing (China) and around. Table III highlights interesting features of those datasets.

TABLE III
Features of the datasets used to evaluate Accio.

Dataset	Time span	#Users	#Records	Area
Cabspotting	1 month	536	11 million	San Francisco area
Geolife	5.5 years	178	25 million	Beijing

B. Environment

Experiments related to the use cases presented in next sections were executed using the local scheduler (i.e., which executes operators as subprocesses) on a single virtual machine running Ubuntu 14.04, having access to 16 cores and 50 Gb of memory.

C. Operators

We use the following subset of operators in our experiments organised in three categories: pre-processing, LPPM, and privacy and utility metrics.

Data pre-processing operators. The purpose of the pre-processing is either to clean a dataset to remove outliers (e.g., remove too short traces), to enforce some features for a fair comparison (e.g., sampling rate, duration), or to simulate an applicative use case (e.g., sending data by batches of six hours). We use two such operators in our experiments.

- `Temporal Sampling`: samples traces to ensure a minimum duration between two consecutive points.
- `Temporal Gap Splitting`: splits traces into two new traces each time the temporal gap between two consecutive points is greater than a specified duration. Resulting traces are assigned to two different (virtual) users.

In our case, the goal of the temporal sampling operator is to reduce the size of the datasets to handle, as well as to uniformise the sampling rate between different datasets. The temporal gap splitting operator is then used to simulate a crowd-sourcing application sending traces to a server when the user is inactive.

LPPMs. These operators implement state-of-the-art protection mechanisms. We use three such operators in our experiments.

- `GeoI`: implements Geo-I [6], which is an LPPM ensuring differential privacy. This LPPM add a calibrated noise controlled by an ϵ parameter (the smaller ϵ, the higher the amount of noise added to the raw data). This LPPM was reimplemented from the methodology described by the authors.
- `W4M`: implements $\mathcal{W}4\mathcal{M}$ [20], which ensures k-anonymity. More precisely, this mechanism enforces that at least k users move inside a cylindrical volume of diameter $delta$. We reused the binary that was made available by the authors [39] and simply implement converters between our dataset format and theirs.
- `Promesse`: implements Promesse [23]. This mechanism aims to hide the POIs of users by using a speed smoothing technique enforcing a constant distance $alpha$ between consecutive records. We wrapped the original code of the LPPM to create an operator.

These LPPMs are all implemented in a different manner: by reimplemented the state-of-the-art, by using original work or by wrapping an existing publicly available binary. This demonstrates the highly flexibility of Accio when it comes to implementing operators for LPPMs.

Privacy and utility metrics. These operators evaluate either the privacy gain or the utility loss between an original dataset and its protected version from a LPPM. We use five such operators in our experiments.

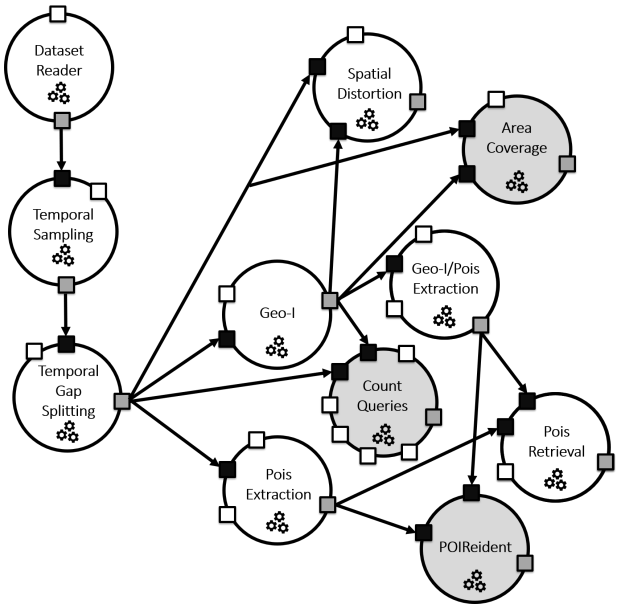


Fig. 5. Workflow to evaluate GEO-I LPPM with five different metrics.

```
uri = param("/path/to/cabspotting")
epsilon = param_set(0.0001, 0.001, 0.01, 0.1, 1)

def PreProcess(data):
  d = TemporalSampling(data, "5.minutes").data
  d = TemporalGapSplitting(d, "6.hours").data
  d = EnforceSize(d, minSize="15.minutes").data
  return d

def POIs(data):
  return PoisExtraction(data, diameter="200.meters",
    duration="15.minutes")

def Metrics(train, test):
  PoisRetrieval(POIs(train), POIs(test),
    threshold="100.meters")
  PoisReident(POIs(train), POIs(test),
    diameter="200.meters", duration="15.minutes")
  CountQueries(train, test, n="1000",
    minSize="500.meters", maxSize="5000.meters",
    minDuration="2.hours", maxDuration="8.hours")
  SpatialDistortion(train, test)
  AreaCoverage(train, test, level=13)

d = PreProcess(DatasetReader(uri))
Metrics(d, GeoI(d, epsilon).data)
```

Fig. 6. DSL to evaluate GEO-I with five different metrics.

- `PoisRetrieval`: implements the points of interest retrieval metric [23], which quantifies how many points of interest (POI, meaningful places where user spends a certain amount of time such as home or work places) can still be retrieved from a protected dataset compared to the original dataset.
- `CountQueries`: implements the count query distortion metric [20], which evaluates the correctness of randomly generated count queries on a protected dataset. These count queries operator aims at counting individual users present inside a given area during a given time window.
- `AreaCoverage`: implements the area coverage metric [20], [23], which assesses whether the mobility of users in the protected dataset covers the same area than the original dataset.
- `SpatialDistortion`: implements the spatial distortion metric [23], which quantifies the spatial modification of mobility of users between a protected dataset and the original dataset.
- `PoisReident`: implements a re-identification metric [28], which evaluates whether users can still be re-identified from a protected dataset. The underlying re-identification attack leverages the POIs of users to compute the similarity between two mobility traces and thus to perform the mapping of users.

Moreover, we need an additional operator to extract the POIs from a dataset.

- `PoisExtraction`: extracts points of interest from a dataset, using the algorithm described in [28].

While this last operator could theoretically be directly embedded into the relevant operators (e.g., `PoisRetrieval` and `PoisReident`), splitting this routine into a separate operator favours the reusability.

VI. Evaluation through Use cases

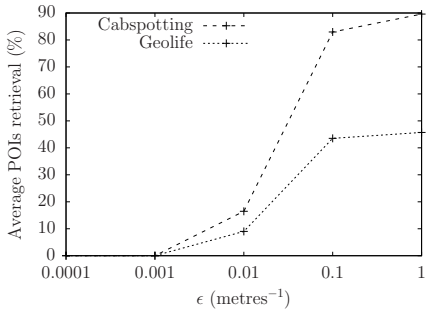
To demonstrate the capacity of ACCIO, we present experiments through different use cases. The objective of these use cases is to highlight iteratively several important aspects of location privacy experiments. We first expose an experiment evaluating an LPPM with different metrics (Section VI-A). We then evaluate this LPPM with different datasets (Section VI-B), and finally we compare it against other LPPMs (Section VI-C). For each use case, we present the workflow and the associated DSL as well as the results of the targeted evaluations.

A. Use case 1: Metric diversity

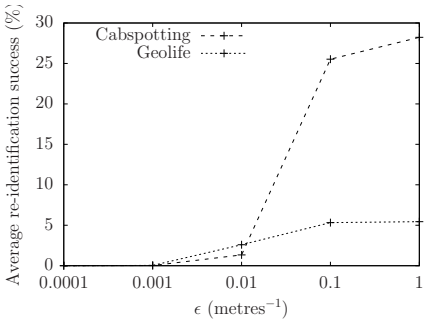
In this first experiment, we evaluate the GEO-I LPPM under five different metrics. We build on the example workflow given in Section IV-B, Figure 3 and add three more metrics to it. The final workflow is depicted in Figure 5 where the new metrics are in grey.

Overall, the whole workflow is expressed in 21 single lines of DSL (or as little as 17 single lines of DSL, when excluding blank lines), which is shown in Figure 6. This use case shows that ACCIO can accommodate new metrics very easily. Consequently, ACCIO makes it easier the experiment of LPPMs with evaluation metrics that were not specifically used their authors. This advantage makes the evaluation of an LPPM more objective.

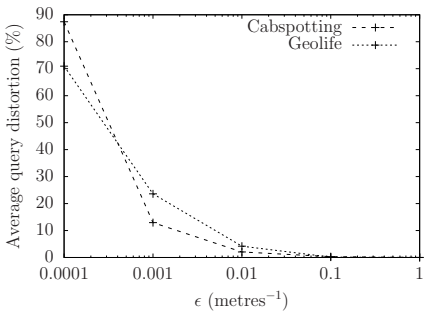
As depicted Figure 6, we launch this workflow to experiment GEO-I with multiple configurations, with $\epsilon \in \{0.0001, 0.001, 0.01, 0.1, 1\}$, and on the Cabspotting dataset. Each operator of the DSL is parameterised with specific values (e.g., for the POIs extraction a POI is defined as a place where users stay in a diameter of 200 metres during at least 15 minutes). Configuration details of each operator are available on the documentation website of ACCIO [35]. Results of this



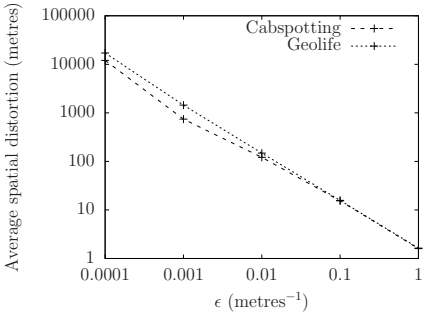
(a) Privacy – POIs retrieval (lower is better)



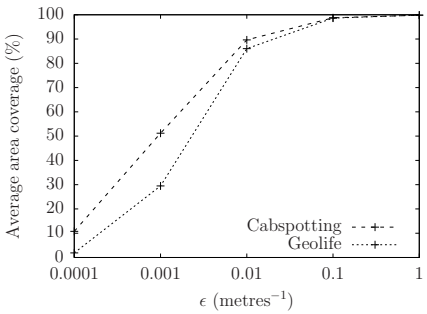
(b) Privacy – Re-identification (lower is better)



(c) Utility – Count queries (lower is better)



(d) Utility – Spatial distortion (lower is better)



(e) Utility – Area coverage (higher is better)

Fig. 7. Use case 2: Privacy and utility trade-off of GEO-I LPPM with five metrics and two mobility datasets.

experiment are shown in Figure 7. These results represent the privacy and utility evaluation of GEO-I with five different evaluation metrics. We remind that the lower ϵ, the strongest the theoretical guarantee provided by GEO-I. Conversely, a high value of ϵ means a very relaxed theoretical privacy guarantee. Results clearly show the trade-off between privacy and utility. Until $\epsilon = 0.001$, privacy is perfectly preserved, with respect to chosen privacy metric, at the cost of a degraded utility (e.g., from 87 % to 12 % for Count queries with ϵ at 0.0001 and 0.001 respectively). Increasing ϵ results in weakening privacy while improving utility.

B. Use case 2: Dataset diversity

To highlight the flexibility and the reusability of ACCIO workflows, in this second experiment we evaluate GEO-I with the same metrics as in the first use case (Section VI-A), but with an additional mobility dataset, specifically Geolife. Consequently, the workflow DSL (Figure 6) remains the same but we simply add a dataset in the `uri` parameter as follows (Figure 8).

```
1  uri = param_set("/path/to/cabspotting",
       "/path/to/geolife")
```

Fig. 8. Change in the DSL of the first use case (Fig. 6) to evaluate the GEO-I LPPM with two datasets.

Results for all privacy and utility metrics under both datasets are shown in Figure 7. An interesting observation can be done from these results. Curves exhibit similar behaviours for both datasets on all metrics, except for the re-identification privacy metric. Indeed, Figure 7b shows a much lower re-identification success rate when $\epsilon > 0.01$ for Cabspotting dataset compared to Geolife dataset. As for evaluation metrics, ACCIO also makes it possible to integrate additional datasets very easily. Our observation with the re-identification success rate shows how important it is to cross-validate results with multiple datasets.

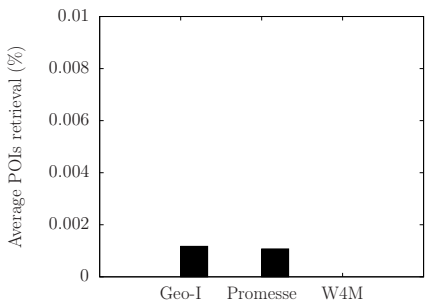
C. Use case 3: LPPM diversity

In this third and last experiment, we compare GEO-I against two other LPPMs, namely $\mathcal{W}4\mathcal{M}$ and PROMESSE. In this experiment, we will use the very same metrics to compare three extremely different LPPMs, enforcing different privacy models. We use the same privacy and utility metrics than in the first and second use cases (only with the Cabspotting dataset). The changes compared to the DSL of the first use case are reported in Figure 10 (we essentially add two new lines at the end).

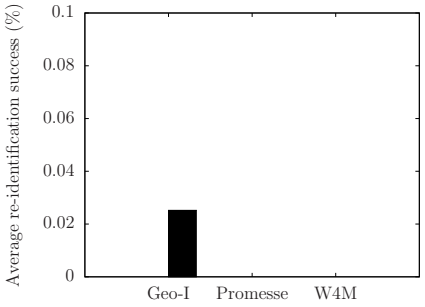
```
21  Metrics(d, GeoI(d, epsilon=0.001).data)
22  Metrics(d, Promesse(d, alpha="200.meters").data)
23  Metrics(d, W4M(d, delta="600.meters").data)
```

Fig. 10. Changes in the DSL of the first use case (Fig. 6) to compare three different LPPMs.

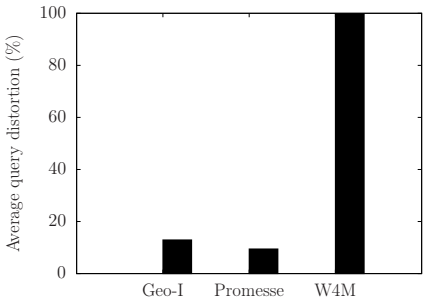
For readability reasons, each LPPM is parametrised in such a way that it offers a relevant trade-off between privacy and utility. GEO-I is configured with $\epsilon = 0.001$ by analysing results



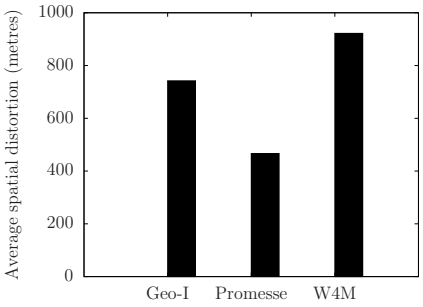
(a) Privacy – POIs retrieval (lower is better)



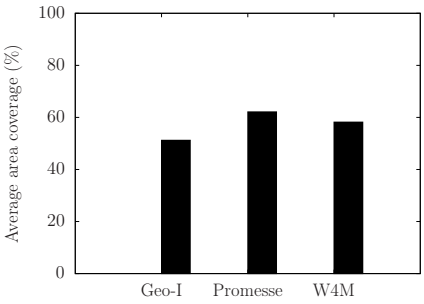
(b) Privacy – Re-identification (lower is better)



(c) Utility – Count queries (lower is better)



(d) Utility – Spatial distortion (lower is better)



(e) Utility – Area coverage (higher is better)

Fig. 9. Use case 3: Results of the comparison of three different LPPMs.

of previous use cases. Indeed, this value of ϵ gave an almost perfect privacy level with a minimal utility loss (compared to $\epsilon = 0.0001$). $\mathcal{W}4\mathcal{M}$, in turn, is configured with $k = 10$ and $\delta = 600$ metres. These values are chosen from analysing the associated research paper, some experimental results, and tips given by authors for reasonable parameters. We configure it with $k = 10$ and $\delta = 600$ metres. Finally, PROMESSE is configured with $\alpha = 200$ metres, which should hide properly POIs extracted with a radius of 200 metres [23]. Compared to the first use case, the DSL of this use case takes only two additional lines to add the two new LPPMs.

Figure 7 exposes the results of this experiment. Due to the relevant configuration of LPPMs, they all feature good privacy levels. More precisely, a perfect privacy is achieved for $\mathcal{W}4\mathcal{M}$ with both metrics and almost perfect privacy for GEO-I and PROMESSE (with an average POIs retrieval below 0.001 % and an average re-identification success rate below 0.003 %). However, large differences appear when evaluating utility. They all show similar area coverage (i.e., between 50 % and 60 %) but behave significantly differently in terms of count query distortion and spatial distortion, where $\mathcal{W}4\mathcal{M}$ obtains the worst results.

VII. DISCUSSION & CONCLUSION

In this paper we presented ACCIO, a framework designed to enhance location privacy studies. This tool makes the evaluation and the comparison of LPPMs easier by proposing a unified and extensible framework including state-of-the-art LPPMs, evaluation metrics and a library for spatio-temporal data manipulation and pre-processing. ACCIO comes with a user-friendly and expressive DSL to describe and easily reproduce complex experiments on location privacy. In addition, the user interacts with ACCIO via a Web or command-line interface. Finally, this tool works as well on a single machine as on a distributed system parallelising the execution on multiple nodes. Through different uses cases, we demonstrate how easy it is to launch experiments with ACCIO going from the evaluation of one LPPM with few evaluation metrics to more advanced experiments comparing multiple LPPMs together.

Compared to previous state-of-the-art frameworks, ACCIO brings reusability (which Location Privacy Meter [25] does not provide easily, because of its strict C++ interface), extensibility (GEPETO [26] claims to be extensible, but extension points are not clearly documented) and reproducibility (GEPETO is not easily scriptable). Reusability is provided by the concept of a workflow that is a composition of operators. Because the operators are simple binary executables, they are portable and may be integrated with other workflow management systems or called from other code, thus ensuring interoperability. Moreover, the operator's interface is generic enough to accommodate otherwise very different algorithms. Extensibility comes from well-defined interfaces and extension points, thus allowing to easily integrate new operators into our platform.

Extending Location Privacy Meter or GEPETO, e.g., to add a new LPPM, requires writing C++/Java code, recompiling the framework and then executing it. Conversely, new operators in ACCIO can potentially be developed in multiple languages,

and the operators' lifecycle is managed separately from the rest of framework. Moreover, writing experiments using existing operators only involves writing a few lines of DSL (no compilation is required). ACCIO has already been successfully used by non programmers to experiment with location privacy. Finally, this DSL allows to easily replay past experiments and reproduce their results.

TABLE IV
SIZE OF WORKFLOWS EXPRESSED USING OUR DSL.

Use case	Lines of DSL
1: Metric diversity	21
2: Dataset diversity	21
3: LPPM diversity	23

This goal of this paper was not to evaluate once again state-of-the-art LPPMs but to highlight the flexibility and the simplicity of ACCIO to design, deploy and reproduce location privacy experiments. Table IV summarises the number of lines it takes to describe the experiment associated to each use case considered in this paper. For example, an experiment involving an LPPM, two datasets and five metrics (i.e., the second use case) needs only 21 lines of DSL to be written by a researcher. For comparison, only the code of the operators used in this experiment represents about 1,050 single lines of code in Scala, which does not take into account the code that would be needed to orchestrate them properly outside ACCIO, nor the code dealing with reading and writing datasets (which is a library integrated into ACCIO). Moreover, as showcased during these case studies, ACCIO allows to very easily alter an experiment (e.g., adding a new metric, a new LPPM, changing a parameter), without having to recompile anything. The DSL is currently oriented around easily expressing parameter sweeps (i.e., testing several values of a set of parameters) or variations of operators sharing a similar interface (i.e., testing different LPPMs), in order to identify the best performing configuration. We leave as future work the task to formally evaluate the expressiveness of our DSL.

ACCIO can be easily enriched with new features and operators (e.g., LPPM, evaluation metrics). As future work, additionally to enrich the operators library with state-of-the-art LPPMs and metrics, we plan to support more distributed execution platforms, which are nowadays widely used by researchers to run their experiments quickly. Another interesting contribution would be to support streaming operators, i.e., working on continuous streams of data. Moreover, from our experience, visualising preliminary results can be of a great help to debug issues or improve operators. Consequently, improving visualisation tools to preview results could be also an interesting feature to develop.

REFERENCES

[1] H. Almuhimedi, F. Schaub, N. Sadeh, I. Adjerid, A. Acquisti, J. Gluck, L. F. Cranor, and Y. Agarwal, "Your location has been shared 5,398 times!: A field study on mobile app privacy nudging," in *CHI*, 2015, pp. 787–796.

[2] J. Krumm, "Inference Attacks on Location Tracks," in *PERVASIVE*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 127–143.

[3] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "Show Me How You Move and I Will Tell You Who You Are," *Transactions on Data Privacy*, vol. 4, no. 2, pp. 103–126, Aug. 2011.

[4] L. Franceschi-Bicchierai, "Redditor cracks anonymous data trove to pinpoint muslim cab drivers," Available online at http://mashable.com/2015/01/28/redditor-muslim-cab-drivers/, Jan. 2015.

[5] M. Gramaglia and M. Fiore, "Hiding Mobile Traffic Fingerprints with GLOVE," in *CoNEXT*, 2015.

[6] M. E. Andrès, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential Privacy for Location-based Systems," in *CCS*, 2013, pp. 901–914.

[7] W. Qardaji, W. Yang, and N. Li, "Differentially private grids for geospatial data," in *ICDE*, Apr. 2013, pp. 757–768.

[8] Y. Xiao, L. Xiong, S. Zhang, and Y. Cao, "Loclok: Location cloaking with differential privacy via hidden markov model," *Proc. VLDB Endow.*, vol. 10, no. 12, pp. 1901–1904, Aug. 2017.

[9] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *INFOCOM*, 2014, pp. 754–762.

[10] "General data protection regulation - european commission," Available online at https://www.eugdpr.org.

[11] L. Sweeney, "k-Anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[12] C. Dwork, "Differential Privacy," in *Automata, Languages and Programming*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, vol. 4052, pp. 1–12.

[13] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD data set epfl/mobility (v. 2009-02-24)," Available online at http://crawdad.cs.dartmouth.edu/epfl/mobility, 2009.

[14] N. Kiukkonen, B. J., O. Dousse, D. Gatica-Perez, and L. J., "Towards rich mobile phone datasets: Lausanne data collection campaign," in *ICPS*, 2010.

[15] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining Interesting Locations and Travel Sequences from GPS Trajectories," in *WWW*, 2009, pp. 791–800.

[16] Y. de Montjoye, Z. Smoreda, R. Trinquart, C. Ziemlicki, and V. D. Blondel, "D4D-Senegal: The Second Mobile Phone Data for Development Challenge," *CoRR*, vol. abs/1407.4885, 2014. [Online]. Available: http://arxiv.org/abs/1407.4885

[17] "United states census bureau," Available online at https://www.census.gov.

[18] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," Available online at http://snap.stanford.edu/data, Jun. 2014.

[19] O. Abul, F. Bonchi, and M. Nanni, "Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases," in *ICDE*, 2008, pp. 376–385.

[20] ——, "Anonymization of moving objects databases by clustering and perturbation," *Information Systems*, vol. 35, no. 8, pp. 884–910, 2010.

[21] K. Jiang, D. Shao, S. Bressan, T. Kister, and K.-L. Tan, "Publishing Trajectories with Differential Privacy Guarantees," in *SSDBM*, 2013, pp. 12:1–12:12.

[22] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in *SECURECOMM*, 2005, pp. 194–205.

[23] V. Primault, S. Ben Mokhtar, C. Lauradoux, and L. Brunie, "Time Distortion Anonymization for the Publication of Mobility Data with High Utility," in *TrustCom*, 2015, pp. 539–546.

[24] V. Bindschaedler and R. Shokri, "Synthesizing Plausible Privacy-Preserving Location Traces," in *CCS*, 2016.

[25] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *S&P*, 2011, pp. 247–262.

[26] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "GEPETO: A GEoPrivacy-Enhancing TOolkit," in *AINA*, 2010, pp. 1071–1076.

[27] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A Classification of Location Privacy Attacks and Approaches," *Personal Ubiquitous Computing*, vol. 18, no. 1, pp. 163–175, Jan. 2014.

[28] V. Primault, S. Ben Mokhtar, C. Lauradoux, and L. Brunie, "Differentially Private Location Privacy in Practice," in *MOST*, 2014.

[29] R. Shokri, "Location-privacy meter tool," Available online at http://icapeople.epfl.ch/rshokri/lpm/doc/.

[30] S. Gambs, M. O. Killijian, I. Moise, and M. N. del Prado Cortez, "Mapreducing gepeto or towards conducting a privacy analysis on millions of mobility traces," in *IPDPS, Workshops and Phd Forum*, May 2013, pp. 1937–1946.

[31] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. Ferreira da Silva, M. Livny, and K. Wenger, "Pegasus, a workflow management system for science automation," *Future Generation Computer Systems*, vol. 46, no. C, pp. 17–35, 2015.

[32] M. Wilde, M. Hategan, J. M. Wozniak, B. Clifford, D. S. Katz, and I. Foster, "Swift: A language for distributed parallel scripting," *Parallel Computing*, vol. 37, no. 9, pp. 633–652, Sep. 2011.

[33] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock, "Kepler: An extensible system for design and execution of scientific workflows," in *SSDBM*, 2004, pp. 423–.

[34] J. Liu, E. Pacitti, P. Valduriez, and M. Mattoso, "A Survey of Data-Intensive Scientific Workflow Management," *Journal of Grid Computing*, vol. 13, no. 4, pp. 457–493, 2015.

[35] "Accio documentation," Available online at https://privamov.github.io/accio/.

[36] M. Slee, A. Agarwal, and M. Kwiatkowski, "Thrift: Scalable Cross-Language Services Implementation," Available online at https://thrift.apache.org/static/files/thrift-20070401.pdf, 2007.

[37] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. D. Joseph, R. Katz, S. Shenker, and I. Stoica, "Mesos: A platform for fine-grained resource sharing in the data center," in *NSDI*, 2011, pp. 295–308.

[38] D. Thain, T. Tannenbaum, and M. Livny, "Distributed computing in practice: the condor experience." *Concurrency - Practice and Experience*, vol. 17, no. 2-4, pp. 323–356, 2005.

[39] O. Abul, F. Bonchi, and M. Nanni, "Wait 4 Me: Time-tolerant Anonymization of Moving Objects Databases – Executable," Available online at http://kdd.isti.cnr.it/W4M/.