Privacy-Preserving Indoor Localization on Smartphones

Andreas Konstantinidis, *Member, IEEE*, Georgios Chatzimilioudis, Demetrios Zeinalipour-Yazti, *Member, IEEE*, Paschalis Mpeis, Nikos Pelekis, and Yannis Theodoridis, *Member, IEEE*

Abstract—Indoor Positioning Systems (IPS) have recently received considerable attention, mainly because GPS is unavailable in indoor spaces and consumes considerable energy. On the other hand, predominant Smartphone OS localization subsystems currently rely on server-side localization processes, allowing the service provider to know the location of a user at all times. In this paper, we propose an innovative algorithm for protecting users from location tracking by the localization service, without hindering the provisioning of fine-grained location updates on a continuous basis. Our proposed *Temporal Vector Map (TVM)* algorithm, allows a user to accurately localize by exploiting a *k-Anonymity Bloom (kAB)* filter and a *bestNeighbors* generator of camouflaged localization requests, both of which are shown to be resilient to a variety of privacy attacks. We have evaluated our framework using a real prototype developed in Android and Hadoop HBase as well as realistic Wi-Fi traces scaling-up to several GBs. Our analytical evaluation and experimental study reveal that *TVM* is not vulnerable to attacks that traditionally compromise k-anonymity protection and indicate that *TVM* can offer fine-grained localization in approximately four orders of magnitude less energy and number of messages than competitive approaches.

Index Terms—Indoor, Localization, Smartphones, Fingerprinting, Radiomap, Privacy, K-Anonymity.

1 Introduction

People spend 80-90% of their time in indoor environments[1], including shopping malls, libraries, airports or university campuses. The omni-present availability of sensor-rich mobiles has boosted the interest for a variety of indoor location-based services, such as, in-building guidance and navigation, inventory management, marketing and elderly support through Ambient and Assisted Living [9], [20].

To enable such indoor applications in an energy-efficient manner and without expensive additional hardware, modern smartphones rely on cloud-based *Indoor Positioning Services (IPS)*, which provide the accurate location (position) of a user upon request. There are numerous IPS, including *Skyhook, Google, Indoo.rs, Wifarer, Navizon, IndoorAtlas, ByteLight* and our open in-house *Anyplace* [27] system[2]. These systems rely on geolocation databases (DB) containing wireless, magnetic and light signals, upon which users can localize.

Particularly, IPS geolocation DB entries act as reference points for requested localization tasks, as explained thoroughly in Section 2. In summary, a smartphone can determine its location at a coarse granularity (i.e., km or hundreds of meters) up to a fine granularity (i.e., 1-2 meters), by comparing against the reference points, either on the service or on the smartphone itself. One fundamental drawback of IPS is that these receive information about the location of a user while servicing them, generating a variety of location privacy concerns (e.g., surveillance or data for unsolicited advertising)[3]. These concerns don't exist with the satellite-based *Global Positioning System (GPS)*, used in outdoor environments, as GPS performs the localization directly on the phone with no location-sensitive information downloaded from any type of service. Although in this work we are mainly concerned with fine-grained Wi-Fi localization scenarios in indoor spaces, our discussion is equally applicable to other types of indoor fingerprints (e.g., magnetic, light, sound) and outdoor scenarios (e.g., cellular).

Location tracking is unethical in many respects and can even be illegal if it is carried out without the explicit consent of a user. It can reveal the stores and products of interest in a mall we've visited, doctors we saw at a hospital, book shelves of interest in a library, artifacts observed in a museum and generally anything else that might publicize our preferences, beliefs and habits. Somebody might claim that telecoms and governments are already tracking smartphone users outdoors, on the premise of public and national safety[4], thus there is no need to care about indoor location privacy either. Clearly, there is a lot of controversy on whether this is right or wrong, which has to do with different cultural, religious, legal and socio-economic dimensions.

We feel that location tracking by IPS poses a serious imminent privacy threat, which will have a much greater impact than other existing forms of location tracking discussed in Section 2 (i.e., outdoor GPS tracking or Browser-based location tracking). This holds as IPS can track users at very fine granularity over an extended period of time (i.e., recall that people spend considerable time indoors). Moreover, IPS are

D. Zeinalipour-Yazti (Corresponding Author), Department of Computer Science, University of Cyprus, Email: dzeina@cs.ucy.ac.cy, Tel: +357-22-892755, Fax: +357-22-892701, Address: 1 University Avenue, P.O. Box 20537, CY-2109, Nicosia, Cyprus; A. Konstantinidis, G. Chatzimilioudis, P. Mpeis, University of Cyprus, 1678 Nicosia, Cyprus; N. Pelekis, Y. Theodoridis, University of Piraeus, 18534 Piraeus, Greece.

1. US Environmental Protection Agency, http://epa.gov/iaq/
2. Available at: http://anyplace.cs.ucy.ac.cy/
3. Nov. 30, 2012: Forbes Magazine, http://goo.gl/MjcMR
4. Dec. 4, 2013: The Washington Post, http://goo.gl/0jJcrL

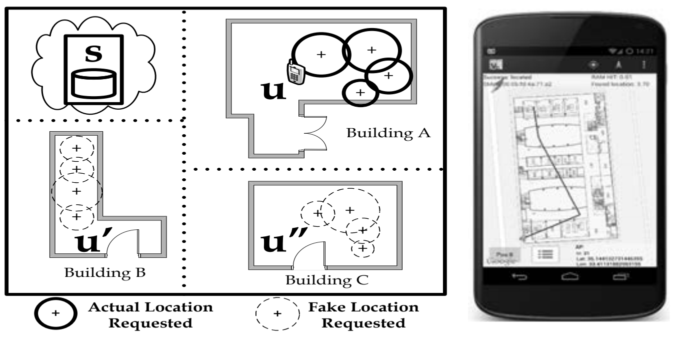


Fig. 1. (Left) Indoor localization of user u using the cloud-based IPS s. During the localization, u requests $k-1$ camouflaged locations using the *TVM* algorithm, such that s can know the location of u only with probability $1/k$. (Right) Our *TVM* prototype implemented in Android OS.

private enterprises that are less controlled, thus they might be tempted to exploit the "big" location data of their customers, by either selling it to advertising companies or by linking it to other sensitive data sources. Additionally, a user cannot know where IPS host and operate their data and whether these conform or not to latest legislative efforts and reforms (e.g., *EU Data Protection Directive, the US White House Consumer Privacy Bill of Rights, U.S.-EU Safe Harbor guidelines, US Do-Not-Track Online Act, etc.*) Finally, IPS are attractive targets for hackers, aiming to steal location data and carry out illegal acts (e.g., breaking into houses[5]).

In this paper, we consider that IPS are *fundamentally untrusted entities* and, as such, develop hybrid techniques that on the one hand exploit the IPS utility, but on the other hand also offer controllable location privacy to the user. Particularly, we tackle the technical challenge of *enabling a user u to localize through an IPS s, without allowing s to know where u is.* We devise the *Temporal Vector Map (TVM)* algorithm[6], which guarantees that s can not identify u's location with a probability higher than a user-defined preference p_u. In *TVM*, a user u camouflages its location from s, by requesting a subset of k entries from s, where k is a user-defined constant.

To understand the operation of *TVM*, at a high level, consider the illustration of Figure 1 (left). An arbitrary user u moves inside building A, using the *TVM* smartphone application shown in Figure 1 (right). While u requests reference locations from s pertinent to building A, it also requests reference locations related to arbitrary other buildings B and C. Particularly, u uses a hashing scheme that makes sure that for a given user-preference $k = 3$, s will not be able to distinguish u's request from requests made by $k-1$ arbitrary other users u' and u''. Under reasonable assumptions about the scope of IPS, we show that s can know u's location only within p_u, even while u is moving. Particularly, the *TVM* algorithm operates in two phases outlined next.

In *Phase 1* of *TVM*, u computes a *k-Anonymity Bloom (*kAB*)* filter structure, which provides location privacy for *snapshot* localization tasks using a bloom filter [4]. When u needs *continuous* localization (e.g., as u moves), the *kAB* of Phase 1 itself is not adequate to preserve the privacy of u, since by issuing k independent requests, s can realize by exclusion that there are $k-1$ invalid requests (as one of the requests will always relate to the real building A). This allows s to deterministically derive u's real location.

To circumvent the above problem, in *Phase 2* of *TVM*, u uses the *bestNeighbors* algorithm to issue a set of camouflaged localization requests that follow a similar natural movement pattern to that of u (i.e., dotted circles in Figure 1, left). This provides the illusion to s that there are k other users moving in space, thus camouflaging u among k other users. Since our *TVM* algorithm transfers only a partial state of the database from s to u, it requires less network traffic and smartphone-side energy than current approaches that transfer the complete database to u prior the localization task.

This paper builds on our previous work in [14], where we presented *TVM*'s predecessor algorithm BMA, which handles snapshot localization tasks only. This work presents a complete framework that covers cases where a user is moving, captures the performance of our approach through analysis and experimentation on a real prototype implemented in Apache HBase and Android. Finally, this work also investigates a range of possible privacy attacks and the provided theoretical guarantees. Overall, our contributions in this work are summarized as follows:

- We devise *TVM*, a complete algorithmic framework for enabling a user to localize without letting the service know where the user is. Our algorithm encapsulates an innovative algorithm for snapshot localization, coined *createkAB*, as well as a counterpart algorithm for continuous localization, coined *bestNeighbors*.
- We provide an analytical study for both the performance and the privacy guarantees provided by our approach. We particularly developed analytical models that enable us to qualitatively derive the properties of our framework.
- We present a real prototype system consisting of a big-data back-end and a smartphone front-end. Using our system, we provide an extensive experimental evaluation with four different realistic datasets on our SmartLab cluster [17] comprising of over 40 real smartphones.

The remainder of the paper is organized as follows. Section 2 provides the related work on indoor localization and privacy-preserving data management. Section 3 provides our desiderata, system model and assumptions. Section 4 presents the *TVM* algorithm, its internal structures and procedures. In Section 5, we provide a performance and privacy analysis of our algorithm. Subsequently, in Section 6 we describe our *TVM* prototype, which is evaluated in Section 7 using different realistic datasets and experimental parameters. Finally, Section 8 concludes this paper.

2 Background and Related Work

In this section, we provide background and related work on indoor localization and privacy-preserving data management, upon which our presented techniques are founded.

5. Apr. 19, 2010: The Huffington Post, http://goo.gl/8aoQ
6. Available at: http://tvm.cs.ucy.ac.cy/

TABLE 1
Localization Technologies for Smartphones

Technology	Runs on	Target	Localization	Location Tracking	Energy (User)	Messaging
GPS	user	outdoor	≈1m (super fine)	No	Bad	-
Cell_ID DB	server	indoor, outdoor	≈1000m (coarse)	Yes	Good	Good
Wi-Fi_ID DB	server	indoor, outdoor	≈200m (coarse)	Yes	Good	Good
Server-Side (SS) Wi-Fi RadioMap	server	indoor, outdoor	≈1.6-10m (fine)	Yes	Good	Good
Client-Side (CS) Wi-Fi RadioMap	user, server	indoor, outdoor	≈1.6-10m (fine)	No	Bad	Bad
Temporal Vector Map (TVM)	**user, server**	**indoor, outdoor**	**≈1.6-10m (fine)**	**No**	**Good**	**Good**

2.1 Background on Indoor Localization

The localization literature is very broad and diverse as it exploits several technologies. GPS is obviously ubiquitously available but has an expensive energy tag and is also negatively affected from the environment (e.g., cloudy days, forests, downtown areas, etc.). Besides GPS, the localization community [9] proposed numerous proprietary solutions including: *Infrared, Bluetooth, visual or acoustic analysis, laser and LiFi, RFID, Inertial Measurement Units, Ultra-Wide-Band, Sensor Networks, etc.*; including their combinations into hybrid systems. Most of these technologies deliver a high level of positioning accuracy, however they require the deployment and calibration of expensive equipment, such as custom transmitters, antennas or beacons, which are dedicated to positioning. This is time consuming and implies high installation costs, while the approaches we discuss operate off-the-shelf on conventional smartphones and Wireless LANs already deployed in most buildings.

Currently, we find the following off-the-shelf positioning systems for modern smartphones (summarized in Table 1):

i) Global PS (GPS): uses radio signals from satellites to offer super fine accuracy often less than 1 meter. The localization is carried out on the handheld, thus we consider that there are no privacy concerns with this approach. However, GPS drains considerable energy and also is unavailable or significantly degraded inside buildings, due to the blockage or attenuation of signal strength [9]. Consequently, GPS cannot be used for indoor localization and is even becoming a secondary choice for outdoor urban spaces, due to its high energy consumption.

ii) Cell DB, Wi-Fi DB or Hybrid Cell/Wi-Fi DB: use radio signals from mobile Cell Towers, Wi-Fi Access Points (APs), or their combination, to offer coarse accuracy that is often less than 1000 meters and 200 meters, respectively. The given databases have been constructed offline by contributors (e.g., an Android phone by default forwards Wi-Fi AP and Cell Tower data to Google). Subsequently, users can obtain their current location using a query / response to the cloud-based localization service. Figure 2, shows a typical example of such a query and response to Google's hybrid Cell/Wi-Fi DB. Particularly, a user u transmits data about the identity (i.e., MAC address or Cell-Id) and signal intensity of its surrounding Wi-Fi APs and Cell Towers. The service s then returns the location of u with an estimated accuracy. The accuracy is a function of how much the service knows about data encapsulated in the query. For this category, the localization is carried out on the server, thus we consider that the service fundamentally violates a user's location privacy.

REQUEST from u to s
```
{
  "homeMobileCountryCode": 310,
  "homeMobileNetworkCode": 410,
  "radioType": "gsm",
  "carrier": "Vodafone",
  "cellTowers": [
    {
      "cellId": 42,
      "locationAreaCode": 415,
      "mobileCountryCode": 310,
      "mobileNetworkCode": 410,
      "age": 0,
      "signalStrength": -60,
      "timingAdvance": 5555
    }
    { ... more CellTowers ... }
  ],
  "wifiAccessPoints": [ { ...
        "macAddress": "01:23:45:67:89:AB",
        "signalStrength": -65,
        "age": 0,
        "channel": 11,
        "signalToNoiseRatio": 40
      }
      { ... more AP... }
    ]
}
```

RESPONSE from s to u
```
{
    "location": {
    "latitude": 51.0,
    "longitude": -0.1,
    },
    "accuracy": 1200.4,
}
```

Fig. 2. Request/Response to Google Cell_ID/Wi-Fi_ID DB

iii) Wi-Fi RadioMaps: is similar to (ii), which stores radio signals from Wi-Fi APs in a database, but at a much higher density. For example, our Anyplace [27] and open-source Airplace [16] systems, use a technology that achieved the second highest known accuracy [21], with an average error of 1.96 meters that works as follows: in an offline phase, a logging application records the so called *Wi-Fi Fingerprints*, which comprise of *Received Signal Strength RSS* values of Wi-Fi AP at certain locations (x,y) pin-pointed on a building floor map (e.g., every few meters). Subsequently, in a second offline phase, the Wi-Fi Fingerprints are joint into a NxM matrix, coined the *Wi-Fi RadioMap*, where N is the number of unique (x,y) fingerprints and M the total number of APs. Finally, a user can compare its currently observed RSS fingerprint against the RadioMap in order to find the best match, using known algorithms such as KNN and WKNN [18].

Particularly, the *K-Nearest-Neighbor (KNN)* approach calculates the Euclidean distance d_i between the user u's currently observed fingerprint V_u against all fingerprints V_i in the RadioMap, i.e., $d_i = ||V_i - V_u||, \forall V_i \in RM$. Then the K nearest fingerprints around the user's device are selected and the user is positioned using convex combination of those K locations. However, by considering that all K nearest neighbor fingerprints are of equal importance (i.e., assigned an equal weight equal to $w_i = 1/K$) may decrease the localization accuracy, since fingerprints that are far away may also be included in the calculation. Therefore, a more effective way of weighting the K nearest fingerprints is required. In the *Weighted-KNN (WKNN)* approach, the K nearest neighbors, calculated as in KNN, are assigned a weight equal to:

$$w_i \propto \frac{1}{||V_i - V_u||}$$

Finally, the user's location is calculated again using a convex combination of those K locations, where in this case the farther locations affect less the calculation than the closer locations.

Discussion: For the final RSS fingerprint comparison step, we differentiate between the following two cases: a) Wi-Fi RadioMap *Server-Side (SS)*, where the localization is taken place on the IPS; and b) Wi-Fi RadioMap *Client-Side (CS)*, where the RadioMap is downloaded to the smartphone prior the localization. In *SS*, localization can be achieved with little network messaging and minimal energy consumption, as the bulk of operation is taken place on the IPS that has an unlimited energy and processing budget. Unfortunately, since the localization in *SS* is carried out by the IPS, this approach is fundamentally violating our location privacy objective. *CS* on the other hand, meets our location privacy objective, but unfortunately requires the download of the RadioMap. As RadioMaps can potentially be very large (e.g., WiGLE.net had 2.8 billion unique records by June, 2015), the *CS* approach leads to the waste of precious and limited smartphone battery and bandwidth. Our analytical and experimental evaluation in Sections 5 and 7 validate this argument.

2.2 Privacy-Preserving Data Management

In relational databases, *k-anonymity [31]* has been a long studied problem with roots to privacy-preserving medical record data sharing and the advent of Hippocratic databases by IBM in 2002 [3]. *Location Privacy* typically refers to the scenario where a data owner wants to publish data or allow spatial querying in its moving object database. To achieve privacy-preservation, the data owner must first "sanitize" the given dataset, such that no one can associate a particular record with the corresponding data subject or infer the sensitive information of any data subject.

Privacy-preserving techniques for location services are based on some of the following concepts: (i) *sanitized locations*; (ii) *spatial cloaking*; (iii) *space transformations*; and (iv) *k-anonymity*. When using sanitized locations a set of fake locations (sanitized) per user are reported to protect location privacy [12], [34]. The main idea in concept (ii) is to blur a user's exact location into a cloaked area that satisfies the user's privacy requirements [5], [8], [13]. In (iii), the locations of users are transformed into another space in which their exact [7], [33] or approximate [11] spatial relationships are maintained.

As for (iv), k-anonymity guarantees that a querying user u is indistinguishable among at least $k-1$ others [30], [31]. In user location privacy k-spatial anonymity is achieved by obfuscating the location of a querying user so that it cannot be identified with a probability higher than $1/k$. This can straightforwardly be achieved using k sanitized locations, assuming that the locations of u has uniform probability over the space. Similarly in trajectory anonymity, k-anonymity approaches guarantee that at least k user trajectories will be indistinguishable among others.

The state-of-the-art k-anonymity [1], [2], [25] approaches mainly rely on historical data and derive sanitized trajectories from a set of real trajectories. Similarly, following an in-house strategy, [26] proposes an integrated platform for applying data mining and privacy-preserving querying over mobility data. The above studies relate to privacy-preserving *data sharing* as opposed to *online localization* described in the present work, where the sanitized trajectories are deterministically derived in real-time and do not rely on any real trajectories.

TABLE 2
Notation used throughout this work

Notation	Description
A, (x,y)	Geographic area, location inside A
ap_i, AP, M, a	Access Point i, set of ap_i, $\|AP\|$, coverage of ap_i
s, u, U	Positioning service, user, set of all u
RM, N, pRM	RadioMap matrix (on s), RM rows, partial RM
V_u	Fingerprint of u (MAC and RSS of its covering AP)
p_u	Privacy preference threshold of u
B_u	*kAB* filter of u (generated using V_u, p_u)
C_u	Candidate set of AP MAC identifiers
$\mathcal{E}_u$	Energy consumed by u for localization

3 System Overview

This section formalizes our system model, assumptions and desiderata. Our main symbols are summarized in Table 2.

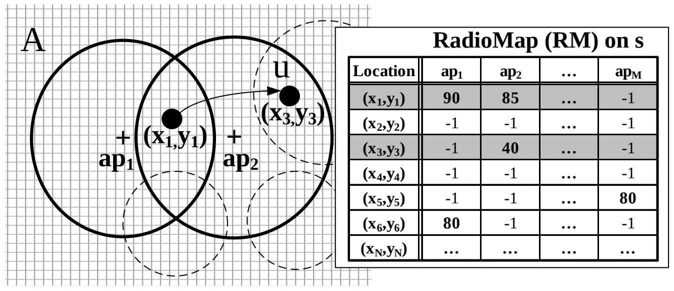
3.1 System Model

Research Goal. *Provide continuous localization to a mobile user u that can measure the signal intensity of its surrounding APs, with minimum energy consumption on u, such that a static cloud-based server s can not identify u's location with a probability higher than a user-defined preference p_u.*

We assume a planar area A containing a finite set of (x,y) points (see Figure 3). We also assume that A is covered by a set of Wi-Fi Access Points $\{ap_1, ap_2, \cdots, ap_M\}$, each covering a planar points. Area A is not necessarily continuous and can be considered as the joint area of all $ap_i \in AP$ (i.e., global coverage). Each ap_i has a unique ID (i.e., MAC address) that is publicly broadcasted and passively received by anyone moving in the a points of ap_i. The signal intensity at which the ID of ap_i is received at location (x,y), is termed the *Received Signal Strength (RSS)* of ap_i at (x,y), having for ease-of-exposition a value in the range $[0..100]$.

Let a static (cloud-based) positioning service s have constructed beforehand an $N \times M$ table, coined *RadioMap* (RM), which records the RSS of the $ap_i \in AP$ broadcasts at specified $(x,y) \in A$ locations. When an ap_i is not seen at a certain (x,y) the RM records "-1" in its respective cell. Any subset of RM rows will be denoted as *partial RadioMap (pRM)*. A user u localizes through the indoor positioning service s, using the ID and RSS broadcasts of surrounding $ap_i \in AP$ while moving. This information is termed, hereafter, *RSS Vector* or *Fingerprint* (V_u) of u, which changes from location to location and over time. Contrary to RM rows having M attributes, V_u has only $M' << M$ attributes.

We assume s to be a static (cloud-based) server of infinite resources, similar to popular positioning and mapping services (e.g., Google Maps), where the user can only communicate with s over the web. Given that s is fundamentally untrusted, we are interested in enabling a user u to localize through a server s without allowing s to know where u is. Therefore, u has a privacy preference threshold p_u, defined as follows:



RadioMap (RM) on s

Location	ap_1	ap_2	...	ap_M
(x_1,y_1)	**90**	**85**	...	-1
(x_2,y_2)	-1	-1	...	-1
(x_3,y_3)	-1	**40**	...	-1
(x_4,y_4)	-1	-1	...	-1
(x_5,y_5)	-1	-1	...	**80**
(x_6,y_6)	**80**	-1	...	-1
(x_N,y_N)	...	...	...	...

Fig. 3. System Model: i) user u moving in area A covered by access point set AP, requests localization from s; and ii) a RadioMap RM of N locations and M access points.

Privacy preference (p_u) *of a user u means that s can not identify the location of u with a probability higher than p_u.*

The location privacy of a user u in our *TVM* algorithm, is provided by a *kAB* filter B_u, which is generated using V_u and p_u. The given filter B_u shall be used by the localization service s, to derive from its RM a set of candidate access points C_u, which lead to a pRM that can be utilized by u to localize.

We are interested in providing this service with minimum energy $\mathcal{E}_u$ overhead on u (i.e., smartphone-perceived energy costs). The reason for neglecting the server-perceived energy costs is that the cloud service s can be "infinitely" powerful with an "infinite" power source (i.e., compared to power-limited smartphones) providing negligible additional energy cost at the smartphone device. For ease of exposition, our analysis uses the notation $\mathcal{E}^{TX}$, $\mathcal{E}^{RX}$ and $\mathcal{E}^{CPU}$ to denote the energy cost needed by u for transmitting, receiving and processing a single database entry from s.

3.2 When it Works

We shall next describe, similarly to [29], under what assumptions our propositions are sound. We consider a service that is fundamentally *untrusted*. As such, the service is operating in one of the following modes: i) it is compromised by the adversary owner of the service; or ii) it is compromised by some adversary third party (e.g., hacker). In both cases, the adversary can operate in the following two modes: i) an *active attacker* mode, in which the adversary attempts to alter system resources or actively combine background knowledge in order to infer where the users are; and ii) a *passive attacker* mode, during which the adversary attempts to learn from whatever data is available on the system (e.g., log files, wiretapping network sockets, etc.) without necessarily having additional information about the users. The *TVM* algorithm presented in this work, is sound under a *passive attacker* model for which the following high-level characteristics apply:

No Low-Level Attacks: We assume cryptographic *Transport Layer Security* and no *Man-in-the-Middle* attacks (e.g., communication *eavesdropping* and *tampering*). These attacks, potentially carried out by governmental or other agencies, could possibly reveal information about $u's$ location regardless of what is safely communicated with our TVM algorithm from u to s. In summary, our work protects u from the untrusted service s without worrying about third parties that might intervene in the communication process.

No Modified Responses: We assume that s is not actively modifying responses in an attempt to manipulate u's behavior and identify its location. More specifically, we assume that s is using a deterministic automaton for its responses (i.e., returning the same answer to a given input). Such a behavior by s could easily be validated by an external auditor, which could periodically certify that s responds consistently (otherwise users have the option to stop using the service).

No Access to User Identifier: One popular technique for location tracking of users on the WWW are user identifiers [28]. This includes: i) the user's *Internet Protocol* address (e.g., browser-based HTML5 or Ip2Geo lookup services), ii) the smartphone's Wi-Fi *MAC* address, its *International Mobile Equipment Identity (IMEI)* and its *Mobile Equipment IDentifier (MEID)*; and iii) the unique identifiers used by services for personalized advertising purposes (e.g., *Google's PREF cookie*). In this work, we assume that these identifiers are not available to s. Particularly, i) Internet Protocol routing can be carried out through a peer-to-peer anonymization network (e.g., the I2P Anonymous Network); ii) MAC/IMEI/MEID identifiers are hidden or periodically modified by the user; and iii) a user doesn't accept cookies.

No Background Knowledge: Any background information attained by s can breach any privacy guarantee [23]. As such, we assume that server s makes no effort to attain any background information and receives only information from the users. This also excludes any statistical background knowledge about buildings (e.g., number of users in a building or area) or user movement, and any validation methods of user requests against geographical maps or building map.

3.3 Baseline Approaches

There are two extreme scenarios of using an IPS, one guaranteeing maximum location privacy with maximum energy consumption, and the other without any privacy guarantees but with minimum energy consumption.

i) *Client side (CS)*: u sends a request to s and receives the whole database upon which it localizes. In this scenario, no information about u is sent to s and therefore it guarantees privacy with the minimum possible probability p of s finding the location of u, i.e., $p{=}1/|A|$, where $|A|$ are all the locations within area A. Regarding energy consumption, though, u receives all N database entries and performs the full computation locally, spending the maximum possible energy, i.e., $\mathcal{E}_u{=}\mathcal{E}^{TX}{+}N{\cdot}M{\cdot}\mathcal{E}^{RX}{+}N{\cdot}\mathcal{E}^{CPU}$. Although this is a one-time cost, which might seem bearable for continuous localization, it can still be prohibitive, as in real world scenarios the database can become extremely large with respect to number of rows N and size of rows M.

ii) *Server side (SS)*: V_u is transmitted to server s, where the location of u is computed. In this scenario, u only sends V_u and receives (x, y), without performing any further computation. In this case, the untrusted server knows with certainty

Algorithm 1 . *Temporal Vector Map (TVM)*

```
Input: $V_u$ is the current fingerprint of $u$; $p_u$ is $u$'s privacy preference; $RM$ is the RadioMap on $s$
Output: $(x, y)$ is the location of $u$
     ▷ Phase 1: Initial Localization (of u through s)
     ———— User-side (u): ————
 1: $B_u = createkAB(V_u, p_u)$                    ▷ kAB filter in Algorithm 2
 2: send $B_u$ to $s$
     ———— Server-side (s): ————
 3: $C_u = kABtoAP(B_u)$               ▷ Set of Candidate AP MAC identifiers
 4: $pRM = filter(RM, C_u)$             ▷ Set of $RM$ rows filtered by $C_u$
 5: send $pRM$ to $u$
     ———— User-side (u): ————
 6: $(x, y) = localize(V_u, pRM)$       ▷ using WKNN, RBF or SNAP [16]
     ▷ Phase 2: Subsequent Localization (of u through s)
     ———— User-side (u): ————
 7: if $(canNotBeServed(V_u, pRM))$ then
 8:     $C_u = bestNeighbors(V_u, pRM)$    ▷ Set of APs in Algorithm 3
 9:     send $C_u$ to $s$
         ———— Server-side (s): ————
10:     $pRM = filter(RM, C_u)$         ▷ Set of $RM$ rows filtered by $C_u$
11:     send $pRM$ to $u$
12: end if
     ———— User-side (u): ————
13: $(x, y) = localize(V_u, pRM)$       ▷ using WKNN, RBF or SNAP [16]
```

u's location, i.e., p=1 regardless of the user preference p_u, therefore, no privacy is achieved. On the other hand, the minimum possible energy is spent, i.e., $\mathcal{E}_u$=$\mathcal{E}^{TX}$+$\mathcal{E}^{RX}$.

4 The TVM Algorithm

In this section, we detail the internal phases of the *Temporal Vector Map (TVM)* algorithm, its correctness properties, an example of its operation and further optimizations.

4.1 Outline

Algorithm 1 outlines the high-level steps of our proposed *TVM* algorithm for answering initial and subsequent localization queries of some user u through the service s. In phase 1, u generates a *k-Anonymity Bloom (kAB)* filter B_u using the *createkAB* routine in Line 1, presented in Algorithm 2. The given filter B_u, sent to s, guarantees that s can not identify u's location with a probability higher than p_u. Upon reception, s uses B_u in Line 3, to find the set of possible matching AP identifiers C_u. In Line 4, s uses C_u to identify a *partial RadioMap (pRM)*, which is sent to u. Using pRM, u is able to localize with known fingerprint-based algorithms such as WKNN, RBF or SNAP [16] in Line 6. In phase 2, for the subsequent localization tasks, u identifies whether it can be served from its prior pRM state in Line 7 (e.g., if a user only moved by a few meters). If this is not the case, u initiates the *bestNeighbor* routine in Line 8, presented in Algorithm 3. This routine generates a new set C_u, which maintains the privacy guarantees when sent to s. Upon reception, s uses the new C_u to identify the corresponding pRM in Line 10 and send it to u to complete localization.

4.2 Phase 1: Initial Localization

We start out with background on Bloom filters, underlying the operation of the *kAB* filter, used in the first phase of *TVM*. Bloom filters [4] are space-efficient probabilistic data structures that are used to answer *set-membership queries* efficiently. The idea is to first allocate a vector of b bit positions, initially all set to 0, and then use h independent hash functions to hash an element to one of the b positions in the vector with a uniform random distribution. To test whether an element e is a member of a set S, we can construct one Bloom filter for e and one Bloom filter for all elements in S. If a single non-zero position in the former is a zero position in the latter, then e certainly does not exist in S. If all non-zero positions match, then e *might* be a member of S. Therefore, Bloom filters do not prevent *false positives*. The most significant feature of Bloom filters, is that given h optimal hash functions, there is a clear relationship between the size b of the filter and the probability *fpr* of a false positive:

Algorithm 2 . *createkAB*

```
Input: $V_u$ is the fingerprint of $u$; $p_u$ is $u$'s privacy preference
Output: $B_u$ kAB filter for $u$
1: Constants: $h$, $M$, $a$ ▷ # of hash functions, $|AP|$, access point coverage
2: $ap_i$ randomly chosen from $V_u$  ▷ Candidate needed for this localization
3: $k = \frac{1}{a \cdot p_u}$                                  ▷ Equation (2)
4: $b = \left\lfloor \frac{-h}{ln(1-\sqrt[h]{k/M})} \right\rfloor$              ▷ Equation (4)
5: for all $h$ hash functions do
6:     $B[hash(ap_i) \bmod b]$ = 1
7: end for
```

$$fpr \approx (1 - e^{-h/b})^h \tag{1}$$

k-Anonymity Bloom (kAB) Filter: In our case, we use Bloom filters for *1-to-k matching queries*, exploiting the inherent characteristic of the controllable fpr. That is, a *kAB* filter B_u is constructed at user u, which guarantees at least k positive matches on the server side, camouflaging the location of u. The value of k is determined by the user-defined parameter p_u as $k \propto 1/p_u$ and the element used to create the *kAB* filter B_u is the MAC address of an access point ap_i accessible by u. The *kAB* filter guarantees that s can not identify u's location with a probability higher than p_u. Particularly, given that ap_i covers a locations, s can not distinguish the location of u among at least $k \cdot a$ locations if:

$$k = \frac{1}{a \cdot p_u} \tag{2}$$

Algorithm 2 presents the internal steps of the *kAB* filter generation. For analysis purposes, we assume the following system constants known to both to u and s: (i) h predefined hash functions; (ii) the number M of access points on s; and (iii) the non-overlapping coverage of each access point a. Initially, u chooses a random ap_i within its vicinity from its RSS vector V_u (Line 2), and creates a *kAB* filter B_u by applying the h hash functions on ap_i (Line 6). Given that we want s to match at least k out of M access points, we set the false positive ratio to:

$$fpr = k/M \tag{3}$$

Based on Equations (1)-(3) the number of bits b to be used for B_u has to be (Line 4):

$$b = \left\lfloor \frac{-h}{ln(1 - \sqrt[h]{k/M})} \right\rfloor \tag{4}$$

Lemma 1 *Phase 1 guarantees that s cannot identify the location of u with a probability higher than p_u.*

Proof. Server s receives *kAB* filter B_u and computes set C_u of all $ap_i \in AP$ that match B_u. Given that the false positive ratio fpr of B_u was chosen according to Equation (3) and s has M registered access points, then $|C_u| = fpr \cdot M = k$. Therefore, s is able to identify the location of u with a probability of at most $p = \frac{1}{k \cdot a} = p_u$ □

4.3 Phase 2: Continuous Localization

Phase 2 provides continuous localization (when u moves in space to a new fingerprint V_u), while continuing to guarantee that s cannot identify the location of u with a probability higher than p_u. The *kAB* filter of *TVM* Phase 1 itself, is not adequate to preserve u's privacy. Particularly, by issuing k independent B_u requests, s is able to realize by exclusion that there are $k-1$ invalid request. This happens as one of the requested access points will have a natural movement pattern, while the rest $k-1$ candidates will have a random movement pattern. Consequently, the C_u set of candidate APs generated by consecutive executions of phase 1 is problematic.

In Algorithm 3, u addresses the aforementioned problem by evolving the candidate set C_u using a strategy that provides the illusion to s that there are k other natural movement patterns. To achieve this objective, u exploits its prior C'_u set (encapsulated in the prior pRM) to generate a new set C_u.

Particularly, in Line 3, u randomly chooses an ap_i from its current fingerprint V_u and adds it to the candidate set of AP MAC addresses C_u in Line 4. As a next step, u is challenged with the generation of the camouflage candidates, which will hide ap_i in a way that p_u is guaranteed. In Line 5, u computes its movement pattern vector Δ_u for moving from ap'_i to ap_i, e.g., by computing the vector defined by the centroids of ap'_i and ap_i. Then the new camouflage candidates will be derived, whose movement pattern vector is closest to Δ_u.

In Line 7, u iterates over all C'_u candidates of the prior localization round. For each candidate, a movement pattern comparison takes place in Line 8, with the identified camouflage candidate being added to the C_u result. Upon completion, C_u is returned to Algorithm 1 to complete the localization.

Lemma 2 *Phase 2 guarantees that s cannot identify the location of u with a probability higher than p_u.*

Proof. Server s receives C_u of size $|C_u| = |C'_u| = k$. Given that every $ap_j \in C_u$ follows similar movement pattern as $ap_i \in V_u$, s can not eliminate any ap_j from being a member of V_u. Therefore, s is able to identify the location of u with a probability of at most $p = \frac{1}{k \cdot a} = p_u$ □

4.4 Limitations

One basic limitation of our approach becomes evident when some access points in AP have a limited number of neighbors in space (see Figure 4, where u'' has no neighboring access points). In this case the $bestNeighbors$ routine in Lines 7-10 of Algorithm 3, might end up including camouflage candidates that have very different movement patterns than that of u. This could allow s to identify those camouflage candidates and thus locate u with a probability higher than p_u.

Algorithm 3 . *bestNeighbors*

```
Input: $V_u$ is the fingerprint of $u$; $pRM$ is the local partial RM on $u$
Output: $C_u$ set of candidate APs to be sent to $s$ as new query
 1: Constants: $h$, $M$, $a$ ▷ # of hash functions, $|AP|$, access point coverage
 2: Static: $ap'_i$                         ▷ Candidate used in prior localization
    ————— Generate Necessary Candidate: —————
 3: $ap_i$ randomly chosen from $V_u$  ▷ Candidate needed for this localization
 4: $C_u = \{ap_i\}$                                     ▷ Add to results
    ————— Generate Camouflage Candidates: —————
 5: $\Delta_u = \delta(ap'_i, ap_i)$     ▷ Compute movement vector (from $ap'_i$ to $ap_i$)
 6: $C'_u = extract(pRM)$        ▷ Set of candidates from prior localization
 7: for all $ap'_j \in C'_u$ do                  ▷ Evolve camouflage candidates
 8:     $ap_j = argmin_{ap_z \in pRM}(|\Delta_u - \delta(ap'_j, ap_z)|)$ ▷ Find new candidate
 9:     $C_u = C_u \cup \{ap_j\}$                         ▷ Add to results
10: end for
```

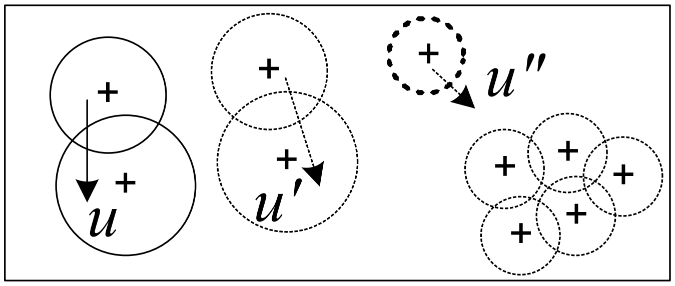


Fig. 4. No Neighboring Access Points: A camouflage candidate u'' has no neighboring access points. In this case *TVM* gives the real user u the option to either abort or to continue with a decreased privacy guarantee p_u.

Fortunately, *TVM* can identify this case on the user-side during Algorithm 3 with minor modifications. Particularly, when the movement vector of the camouflage candidate ap_j in Line 8 differs from u's movement vector by more than a given threshold, the given function disqualifies ap_j from being part of the C_u set. If this results in $|C_u| < k$, *TVM* gives the option to u to either abort or continue with a decreased privacy guarantee that corresponds to $p_u = \frac{1}{a \cdot |C_u|}$.

Another limitation of our framework is that it can uphold its privacy guarantees only if the server remains a passive attacker. There is no privacy guarantee against an active attacker s, since any background information attained by s can breach privacy guarantee [23]. In the worst-case, an attacker might become aware of u's location through an external entity. As such, we claim that our framework guarantees privacy only as long as s makes no effort to attain any background information (passive attacker mode) and receives only information from the users. This also excludes any statistical background knowledge about buildings (e.g., number of users in a building or area) or user movement, and any validation methods of user requests against geographical maps or building map.

4.5 Example

Consider the scenario in Figure 5, where a user u aims to localize at some arbitrary location with fingerprint V_u=$\{ap_1$:90, ap_2:90$\}$. Our scenario assumes $M = 100$ access points, $h = 3$ predefined hash functions, access point coverage

RadioMap (RM)

Location	ap_1	ap_2	ap_3	ap_4	ap_5	…	ap_{99}	ap_{100}
(x_1,y_1)	**90**	85	-1	-1	-1	…	-1	-1
(x_2,y_2)	-1	-1	-1	-1	-1	…	-1	-1
(x_3,y_3)	-1	40	-1	15	30	…	**40**	-1
(x_4,y_4)	-1	-1	-1	-1	-1	…	-1	-1
(x_5,y_5)	-1	-1	-1	40	-1	…	-1	**80**
(x_6,y_6)	**80**	-1	**85**	-1	-1	…	-1	-1
…	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**

Scenario Constants: **M**=100; **h**=3; **a**=1;

Phase 1 **[Input:** **V_u** = {ap_1:90, ap_2:90}; **p_u** = 1/3;]
ap_i = ap_2; **b**=8; **k**=3; **B_u** = {0,0,0,1,0,1,0,0}; **C_u** = {ap_2, ap_4, ap_5}

Phase 2 **[Input:** **V_u** = {ap_1:80, ap_3:75}; **p_u** = 1/3; **C'_u** = {ap_2, ap_4, ap_5}]
ap_i = ap_3; **C_u** = {ap_3, ap_{99}, ap_{100}}

Fig. 5. Example execution of the Temporal Vector Map (*TVM*) algorithm for two consecutive localization queries.

$a = 1$ and a user privacy preference threshold $p_u = \frac{1}{3}$. User u first runs Algorithm 2, which uses V_u and p_u to create a *kAB* filter. Particularly, it chooses in Line 2 an arbitrary $ap_i = ap_2$ and subsequently calculates $k = \frac{1}{a \cdot p_u} = 3$ and $b = \lfloor \frac{-3}{ln(1-\sqrt[3]{3/100})} \rfloor = 8$. Without loss of generality, let the above result in *kAB* filter $B_u = \{0, 0, 0, 1, 0, 1, 0, 0\}$, which is transmitted from u to s.

Upon reception of B_u, s derives the candidate set $C_u = \{ap_2, ap_4, ap_5\}$. At this point the server s knows that u needs RadioMap (RM) rows pertinent to the identifiers in C_u. To satisfy this query, s retrieves any row that has at least one positive value for any $ap_i \in C_u$ (see rows with lightly-shaded cells in Figure 5). In our example, these rows are $\{1, 3, 5\}$ that comprise the pRM, which is sent back to u for localization.

Let us now assume that u moves to a new location with fingerprint V_u=$\{ap_1$:80, ap_3:75$\}$. The prior pRM of u neither includes all RM rows related to ap_1 nor includes all RM rows related to ap_3 (e.g., (x_6, y_6) is of interest to u, but not in pRM). Consequently, u uses Algorithm 3 that starts out by choosing an arbitrary $ap_i = ap_3$ from V_u.

Subsequently, it goes through its prior *pRM* aiming to identify camouflage candidates that maintain a similar movement pattern to the user's actual movement pattern (i.e., from ap_2 to ap_3). In our example, we assume that these candidates are ap_{99} and ap_{100} (see rows with darkly-shaded cells in Figure 5), thus $C_u = \{ap_3, ap_{99}, ap_{100}\}$. The resulting pRM now contains rows $\{3, 5, 6\}$ and is, finally, shipped to u to complete localization.

4.6 Optimization Using Caching

In order to further optimize the performance of *TVM*, a cache on the smartphone's internal storage (e.g., sdcard, flash memory) can be used to keep previous partial RadioMaps. When this optimization process is utilized, the user checks if any of the locally cached RadioMaps can serve its localization request (Algorithm 1, Line 7). This reduces the occasions where Phase 2 is initiated and, thus, network resources are conserved with the tradeoff of higher memory utilization on the smartphone.

4.6.1 *Intermittent Connectivity and Disconnections:*

In mobile networks, the communication between smartphone users and IPS often suffers by intermittent connectivity that refers to the frequent disconnection of a mobile node in random time intervals. This often occurs due to the following two reasons [32]: i) there is a gap between the coverage of two Access Points (APs) and thus the connectivity experienced by mobile users passing by will likely to be intermittent; and ii) because of physical obstacles as well as high mobility patterns of the mobile users. In the literature, there are several propositions for dealing with intermittently connected networks with one of the most popular being *prefetching*. Prefetching predicts what data an application will request in the future and speculatively retrieves and caches that data in anticipation of those future needs [10]. Therefore, our already cached partial Radiomaps can be further optimized and used to deal with disconnections, even though this is not the major focus of this work [15].

5 Analysis and Attacks

In this section, we analyze the performance and privacy characteristics of the *TVM* algorithmic framework.

5.1 Performance Analysis

We analytically derive the performance of *TVM* with respect to the energy $\mathcal{E}_u$ consumed on user u. We adopt a worst case analysis as it provides a bound for all input. Our experimental evaluation in Section 7, shows that under realistic and real datasets our approach performs more efficiently than the projected worst case. The analysis is based on our system model and ignores any energy not directly associated with the localization process, e.g., LCD, Bluetooth, etc.

Lemma 3. *Our TVM approach has an energy complexity of* $O(\mathcal{E}^{TX} + n \cdot M \cdot \mathcal{E}^{RX} + n \cdot \mathcal{E}^{CPU})$*, where n is the number of entries retrieved from s.*

Proof. During initial localization in Phase 1, u creates a *kAB* filter B_u by hashing a single ap_i, consuming $\mathcal{E}^{CPU}$ energy, and forwards B_u of size b to s, consuming $b \cdot \mathcal{E}^{TX}$ energy. On the other hand, during a subsequent localization in Phase 2, u first evolves the candidate set C_u of size k, consuming $k \cdot \mathcal{E}^{CPU}$ energy, and then forwards C_u to s, consuming $k \cdot \mathcal{E}^{TX}$ energy. Then, s responds with $n << N$ database entries to u, where each entry has $M + 2$ values, therefore u consumes asymptotically $n \cdot M \cdot \mathcal{E}^{RX}$ energy. Finally, u localizes itself using the n entries, thus consumes in the worst case $n \cdot \mathcal{E}^{CPU}$ energy. We can safely assume that $k < b < n << N$, therefore, adding all consumptions in an asymptotic manner yields $O(\mathcal{E}^{TX} + n \cdot M \cdot \mathcal{E}^{RX} + n \cdot \mathcal{E}^{CPU}))$ □

Lemma 4. *Our TVM approach has a message cost of $O(n)$, where n is the number entries retrieved from s.*

Proof. Derived from analysis of previous proof □

There is a clear trade-off on the user preference p_u, since with smaller p_u stronger privacy is achieved on the one hand, but on the other more energy is consumed since $k \to M$ and $n \to N$. If user sets maximum $p_u = 1/M$ (i.e., hiding within all existing APs), then it will be $n = N$ and u will receive the whole database. In this case, *TVM* is the same as the client-side approach *CS* described in Subsection 3.3.

5.2 Privacy Attacks

In this subsection, we discuss how *TVM* is resilient to a variety of known privacy attacks.

Linking Attack: Sweeney [31] showed that after removing uniquely identifying attributes from a person's record, people can still be identified by the so-called quasi-identifiers (non-sensitive and non-unique data) that can be linked to break anonymization and compromise privacy. In our case, the uniquely identifying attribute is the fingerprint of a user location. In fact, this is also the only attribute sent by the user to the server, therefore there are no other attributes that could link to the user's fingerprint value. Nevertheless, *TVM* camouflages the user's fingerprint, guaranteeing that the server can not identify the user's location with a probability higher than a user-defined preference p_u.

Homogeneity Attack: Over the past few years several researchers have shown that k-anonymity does not guarantee privacy and it is sometimes vulnerable to attacks. For example, grouping the k-anonymous set may leak information due to lack of diversity in the sensitive attributes giving rise to the so-called Homogeneity attack [22]. This attack is often tackled by l-diversity [22] that guarantees a diversity l within the sensitive data of the k-anonymous set, or the T-closeness [19] that ensures that the difference in the distribution of the sensitive data in the resulting set and the data in the whole set is smaller than a threshold. In our case, however, there is an inherent diversity in the resulting k-anonymous set of *TVM*, since it uses hashing to generate a set of unique access point MAC values that has a uniform distribution over all values.

6 TVM Prototype System

In this section, we describe our *TVM* framework and prototype system used in our evaluation with the SmartLab cluster of Smartphones[7] [17], which is built on top of the ubiquitous Android OS and a back-end that runs over the open-source *Apache Hadoop/HBase* project [8] making our solution *big data* ready. We provide an overview of the three layers that compose *TVM* framework, followed by a description of our client's side Graphical User Interface (GUI).

6.1 TVM System Architecture

The user-side is built on top of the ubiquitous Android OS, and its installation package (i.e., APK) has a size of $1,28MB$. It is composed of the *RSS Logger* and the *Find Me* applications of our Airplace Indoor Localization system [16]. The *RSS Logger* application is developed around the Android RSS API for scanning and recording data samples in specific locations at predefined intervals. These samples contain the MAC addresses and RSS levels of all neighboring Wi-Fi AP, as well as the coordinates of the location where the user initiated the recording. The *Find Me* application is a client that runs *TVM* on Android smartphones, connects to the server in order to download the partial RadioMap and enables the user to self-locate independently thereafter.

The server-side is composed of the privacy sub-layer that hosts *TVM* and the storage sublayer. The latter sublayer is suitable for managing and processing large datasets across clusters of computers. Our storage sublayer utilizes most of the Hadoop modules like HDFS, MapReduce and HBase (a NoSQL column-store database) to ensure scalability and reliability. Using a column-store allows us to capture the fact that the RadioMap has an extremely high number of columns M and rows N (i.e., up to 2^{48} columns to capture all possible MAC addresses and an unspecified number of rows).

Finally, the web-based communication layer is responsible for the interaction between the client and server sides through an Oracle Glass Fish server, which is an open-source platform for delivering server-side Java applications and web-services. The communication between the two sides is based on a JSON protocol/WEB2.0 API. Our code is written in JAVA and consists of approximately 38,600 Lines Of Code (LOC). In particular, our server-side code uses $\sim$9,300 LOC and runs over JDK 7.3 and Ubuntu Linux, while our smartphone code uses $\sim$29,300 LOC plus $\sim$906 Lines of XML elements for the Manifest file (settings) and the UI XML descriptions.

6.2 TVM Android

Our prototype GUI, built using our in-house Anyplace project, provides all the functionalities for a user to utilize *TVM*. The GUI is divided into a visualization interface and a settings interface. The visualization interface uses the Android Google MAP API and our proprietary Wi-Fi AP format, which captures multi-dimensional signal strength values collected from nearby AP (i.e., each AP is identified by its network MAC address and its signal strength is measured in dBm). This allows a user to visualize its location/trace as well as the camouflaged locations/traces in both indoor and outdoor environments. At a high level, our settings interface enables a user to (i) keep a record of fingerprints on local storage and crowdsource them to the server, (ii) configure various privacy, e.g., p_u, and performance preferences, e.g., enable caching, (iii) connect to the positioning service and localize using various *TVM*, *CS* or *SS* methods and (iv) switch between online and offline mode to change between experimentation and real operation.

7 Experimental Evaluation

In this section, we describe the details of our experimental methodology: our datasets and our evaluation metrics. We then present the results of our evaluation using five experimental series.

7.1 Datasets

As a foundation for generating large-scale realistic RadioMaps to carry out our trace-driven experimentation, we used the following real data:

7. Available at: http://smartlab.cs.ucy.ac.cy/
8. Apache Hadoop http://hadoop.apache.org/

CSUCY Data: Data is collected in a typical building at the Computer Science (CS) department of the University of Cyprus using three Android devices. In particular, it consists of 45,000 reference fingerprints taken from $\sim$120 Wi-Fi APs installed in the four floors of the *CS* and neighboring buildings. On average, 10.6 APs are detected per location. We collected our data by walking over a path that consists of 2,900 locations. The CSUCY data has a size of $\sim$2.6 MBs.

KIOSUCY Data: Data is collected inside a typical office environment at the KIOS Research Center, University of Cyprus using three different Android devices. In particular, it consists of 105 fingerprints from $\sim$10 Wi-Fi APs. The KIOSUCY data has a size of $\sim$0.14 MBs.

Crawdad[9] **Data:** Data obtained from the Crawdad online archive that include fingerprints from four areas in the United States: the University of Dartmouth, a building in Kirkland Washington DC, and two buildings in Seattle. In particular, it consists of fingerprints from 6,807 distinct locations from $\sim$1,293 APs. The Crawdad data has a size of $\sim$17 MBs.

To evaluate the scalability of our propositions for regions of various scale, we have generated four large realistic RadioMaps by replicating the above datasets onto various locations of real towns around the world obtained manually from TimeGenie[10]. The resulting RadioMaps are the following:

(i) Campus Dataset: A Campus-scale dataset generated by combining the real datasets. It has a size of $\sim$20 MBs.

(ii) Town Dataset: A Town-scale dataset generated by replicating the real datasets onto various areas around a town. It has a size of $\sim$100 MBs.

(iii) City Dataset: A City-scale dataset generated by replicating the real datasets onto various areas around a city. It has a size of $\sim$1 GB.

(iv) Country Dataset: A Country-scale dataset generated by replicating the real datasets onto various areas around a country. It has a size of $\sim$20 GBs.

7.2 Evaluation Metrics

The performance of our *TVM* approach is evaluated in terms of energy (in Joules) consumed by the smartphone device and messaging cost (i.e., number of RM rows) during the localization process. Similarly to [35], we neglect the server-perceived energy costs as the cloud service can be "infinitely" powerful and with an "infinite" power source (i.e., compared to power-limited smartphones). The energy consumed on a smartphone is measured with the help of PowerTutor[11], which according to [6] is 86% and 82% accurate for smartphones in low and high frequencies, respectively. PowerTutor provides a log file showing the energy (in mJ) consumed by the smartphone's major components such as CPU, network interface, GPS, etc., within pre-defined time intervals (e.g., per second timestamps). We measure the total energy consumed by the *TVM* approach on a Samsung Google Nexus S smartphone, isolating the log entries pertinent to the *TVM* process and summing its energy consumed by the CPU and the Wi-Fi antenna modules.

9. Crawdad, http://crawdad.cs.dartmouth.edu/
10. Time Genie, http://www.timegenie.com/
11. PowerTutor, http://powertutor.org/

Note, that the caching optimization described in Subsection 4.6 is used by default for the *TVM* algorithm. For ease of exposition and without loss of generality, the default value of an access point coverage is set to $a = 1$, throughout the experiments. For measuring the performance of consecutive localizations we have defined a fixed route for each dataset, where a user localizes itself every 30 seconds for a total of 300 consecutive localizations. In our experiments we measure the cumulative cost of the whole route.

7.3 Series 1 - Performance Evaluation

In the first experimental series, we evaluate the performance and scalability of our *TVM* approach with respect to the alternative approaches detailed in Table 1 and Subsection 3.3:

- *Server-Side (SS)* solutions (i.e., Cell_ID, WiFi_ID or Server-side RadioMap), which are privacy-invasive, but consume minimal energy.
- *Client-Side (CS)* solution offering optimal privacy guarantees, but consuming the maximum possible energy.

We are not comparing against GPS as this technology is not appropriate for the indoor scenarios we consider in this work. This series uses all four datasets and evaluates the algorithms for both snapshot and continuous localization scenarios.

For snapshot localization, Figure 6 shows that *TVM* performs around one to four orders of magnitude better than the *CS* approach, both in energy and messaging cost, as the dataset size increases. This is due to the fact that *CS* downloads the whole RadioMap (RM) and performs localization at the smartphone. These results are in line with the theoretical cost of *CS*, $\mathcal{E}_u{=}\mathcal{E}^{TX}{+}N{\cdot}M{\cdot}\mathcal{E}^{RX}{+}N{\cdot}\mathcal{E}^{CPU}$, as shown in Subsection 3.3. Comparing to the theoretical cost of *TVM* $\mathcal{E}_u{=}\mathcal{E}^{TX}{+}n{\cdot}M{\cdot}\mathcal{E}^{RX}{+}n{\cdot}\mathcal{E}^{CPU}$ (see Lemma 3), it is apparent that the difference in energy cost is directly related to the difference between the size N of RM utilized by *CS* and the size n of the partial RM (pRM) utilized by *TVM*. It is worth noting that N grows proportional to the dataset size, whereas n stays approximately the same as it is determined solely by user preference p_u and parameter a that are constant in this experiment. This justifies the experimental findings that show a constant messaging cost for *TVM* for all datasets. Furthermore, the energy cost of *TVM* is not constant for all datasets, due the fact that larger datasets have a higher number M of access points, and therefore the required energy cost per message slightly increases. The *SS* approach consumes almost zero energy on the smartphone, $\mathcal{E}_u{=}\mathcal{E}^{TX}{+}\mathcal{E}^{RX}$, since it lets the server perform all computations upon reception of V_u.

For continuous localization, Figure 6 shows that *TVM* performs around one-and-a-half to five orders of magnitude better than *CS*, in terms of energy consumption, for the same reasons as above. The energy measured in this experiment is the sum over 300 consecutive localizations, and this is the reason why the energy measured for all algorithms is much higher than in the experiments for snapshot localization. The energy consumed by the *CS* algorithm includes downloading the RM once and consecutively localizing using the whole RM on the smartphone, rather than utilizing the much smaller pRM. This is the reason why the energy consumption of the

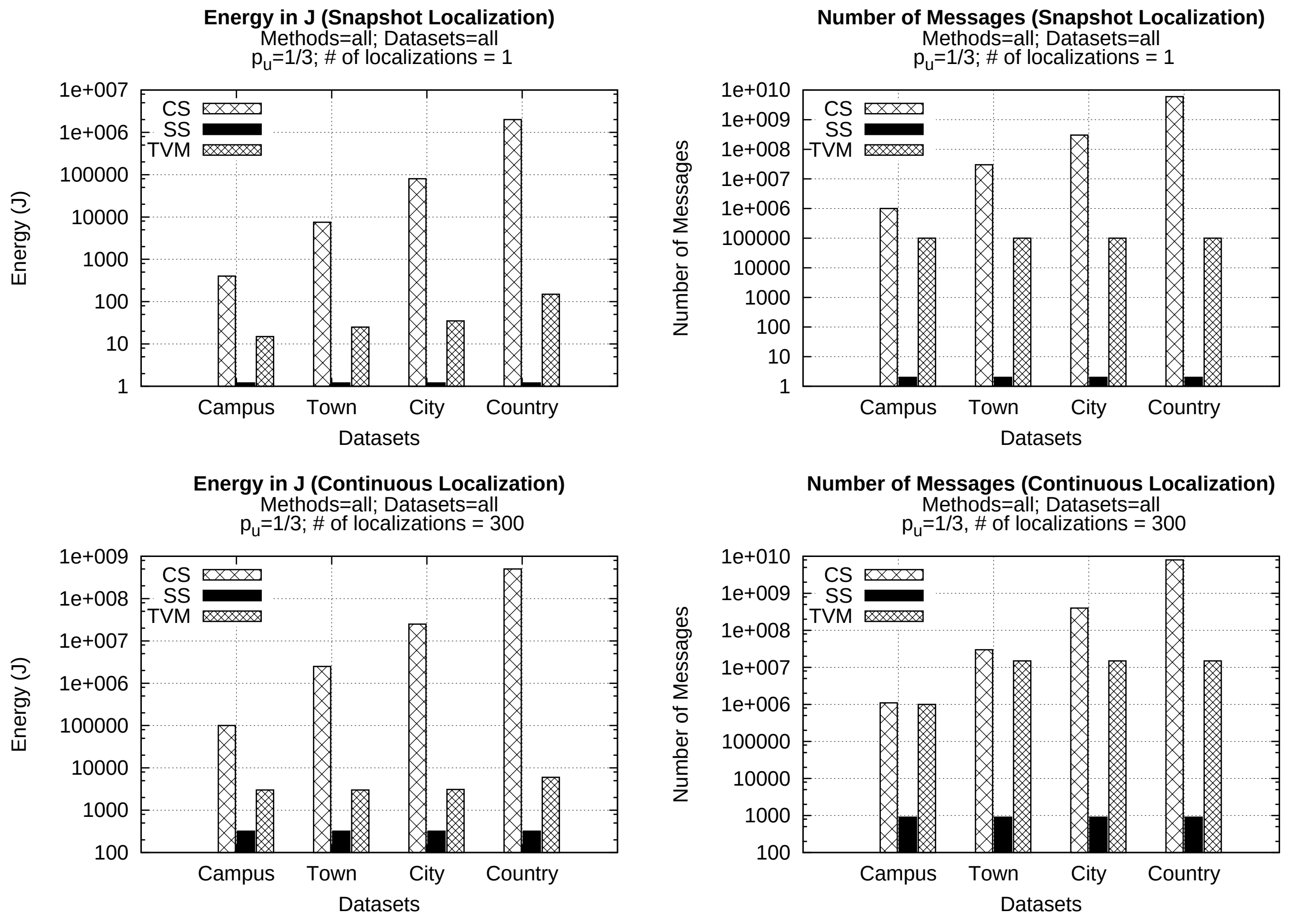


Fig. 6. **Series 1 - Performance Evaluation:** snapshot (top) & continuous (bottom) localization scenario in terms of energy (left) and number of messages (right), varying the dataset size.

CS is higher than *TVM*, in spite of the fact that their messaging cost is the same for the Campus dataset. *TVM* and *CS* have a similar messaging cost for small datasets (e.g., Campus and Town) due to the fact that *TVM* may end up downloading the whole RM during the 300 localization efforts, just like *CS*. Notice, that the messaging cost is upper bound by the total size of each dataset, which is equal to the messaging cost of *CS*. For the large Country dataset, *TVM* outperforms *CS* in messaging cost by around two-and-a-half orders of magnitude.

7.4 Series 2 - Privacy Guarantees

In the second experimental series, our objective is to investigate the probability p with which a server s can identify the exact location of a user u for different user preferences p_u, similarly to the spatial size metric of [24]. In particular, p is equal to one over the (partial) radiomap size or the cloaking region that u uses to localize itself. We also show the same p for the baseline approaches *CS* and *SS* that provide a probability $p = 1/A = \frac{1}{M \cdot a}$ and $p = 1$, respectively. As explained in Subsection 3.3, these represent lower and upper bounds on the probability p with which a server s can identify the exact location of a user u. We calculate p in snapshot localization scenarios, but the same probability holds for continuous localization scenarios as described in Subsection 4.3. We use four different values for the user preference $p_u = \{1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}\}$ in *TVM* to further examine how the privacy guarantees are influenced.

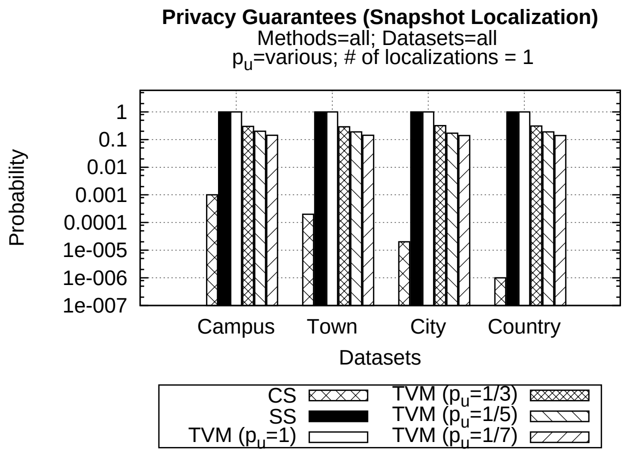


Fig. 7. **Series 2 - Privacy Guarantees:** the probability with which a server can identify the location of a user.

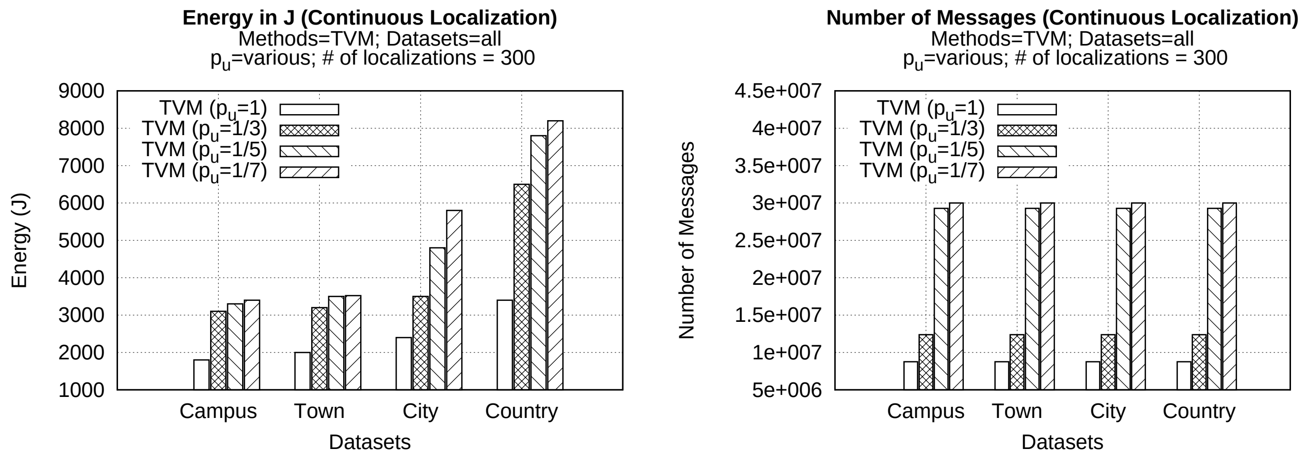


Fig. 8. **Series 3 - Varying Privacy Preference p_u:** the effect on the performance of *TVM* in a continuous localization scenario in terms of energy in Joules (left) and number of messages (right).

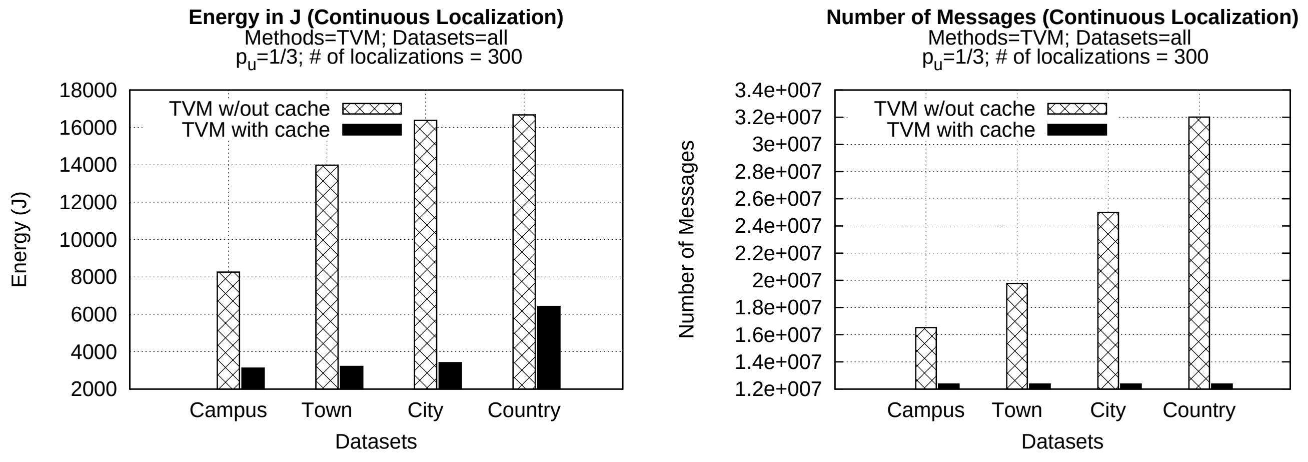


Fig. 9. **Series 4 - Optimizing *TVM* with Caching:** the effect on the performance of *TVM* in a continuous localization scenarios in terms of energy in Joules (left) and number of messages (right).

In Figure 7, the lower bound provided by *CS* is the maximum possible privacy guarantees since it does not send any information about u to s and therefore s only knows that a user's location is within the whole area A covered by the radiomap of size $\frac{1}{M \cdot a}$. The privacy guarantees of *CS* become stronger as the radiomap size increases, since the probability is directly related to the size of the dataset, i.e., the number of access points M. The upper bound provided by the *SS* approach shows no privacy guarantees, as it allows s to know the best possible location of u with probability 1, since u forwards to s its fingerprint and s finds the best approximation to the radiomap entries using the WKNN approach. The proposed *TVM* approach provides steady privacy guarantees in all datasets independently from the size of the dataset as shown by Lemmata 1 and 2 of Section 6. The privacy guarantees of *TVM* are set by the user through the privacy preference threshold p_u, therefore, the probability achieved is always less and approximately equal to p_u.

7.5 Series 3 - Varying Privacy Preference p_u

In this experimental series, we evaluate the trade-off between the privacy guarantees and the energy consumed in continuous localization scenarios. In particular, we investigate how the user preference p_u affects the performance of *TVM* both in terms of energy and messaging cost. We use four different values for $p_u = \{1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}\}$.

Figure 8 shows that the energy consumption and the messaging cost increase for lower values of p_u. This is due to the fact that lower values of p_u result in a higher number of camouflage candidates to be used, and thus in a larger number n of rows to be collected in pRM. On the other hand, the privacy guarantee in *TVM* is determined by the user preference p_u. The probability that a server knows the exact location of a user u decreases with p_u. These observations are in line with our analytical results in Section 5 and the experimental results in Subsection 7.4. They also support our argument that there is a clear trade-off between the performance (energy) and the privacy guarantees that the *TVM* provides to the user. That is, lower values of user preference p_u give stronger privacy guarantees, but require more resources for the localization.

7.6 Series 4 - Optimizing TVM with Caching

In this experimental series, we evaluate the improvement on the performance of *TVM* in continuous localization scenarios when caching is used. In particular, we compare our *TVM* approach with and without the caching optimization described in Subsection 4.6. Figure 9 shows that the proposed optimization technique reduces the requests for new pRMs. This results in an improvement of $\sim$66%-120% and $\sim$70%-250% for the energy and messaging cost, respectively.

7.7 Series 5 - Device Diversity

In this experimental series, we expose the firmness and stability of *TVM* approach on a variety of popular Android devices (i.e., Samsung Galaxy Tab 2.0, Samsung Google Nexus S, Asus TF700T and HTC Desire) under various user preference thresholds p_u and in continuous localization scenarios. For brevity, we will only present the results for the town dataset, since the respective results of the other datasets look very similar. The results in Figure 10 show that the *TVM* approach performs best for the Samsung Galaxy tablet and performs worst for the HTC Desire smartphone. The decrease in response time as the preference threshold p_u increases, is almost linear and with relative small standard deviation. Overall, the behavior of *TVM* is consistent for all values of p_u under various devices, showing that TVM is a stable technique.

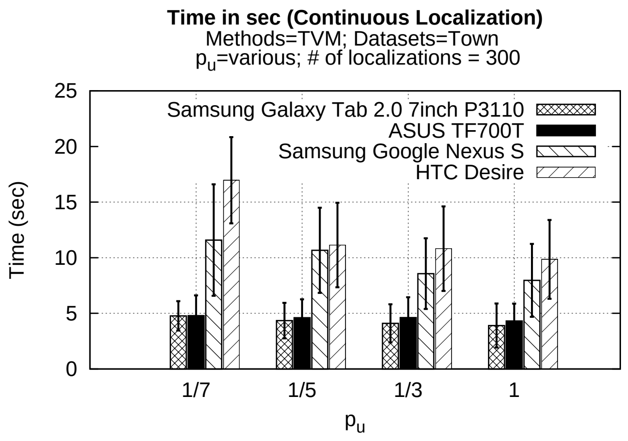


Fig. 10. **Series 5 - Android device diversity:** Firmness and stability of *TVM* on different Android smartphones.

8 Conclusions

In this paper, we propose a complete algorithmic framework, coined *Temporal Vector Map (TVM)*, for enabling a user to localize without letting the service know where the user is. Our algorithm encapsulates a number of innovative internal components for snapshot and continuous localization. We provide an analytical study for both the performance and the privacy guarantees provided by our approach and present a real prototype system consisting of a big-data back-end and a smartphone front-end. Using our system, we provide an extensive experimental evaluation with four different realistic datasets on our SmartLab smartphone cluster. Our results indicate that *TVM* can offer fine-grained localization in approximately four orders of magnitude less energy and number of messages than competitive approaches. In the future, we aim to carry out a field study, investigate server-side optimizations that will further boost the performance of *TVM*, and also investigate the applicability of the *TVM* framework to more generalized sensor measurements.

Acknowledgments: We would like to thank the anonymous reviewers for their valuable comments and suggestions, as well as Giannis Evagorou and Silouanos Nicolaou (University of Cyprus) for their contributions to the evaluation of TVM. This work was supported in part by the third author's Startup Grant funded by the University of Cyprus, COST Action IC903 (MOVE) *"Knowledge Discovery for Moving Objects"*, EU's FP7 projects (MODAP) *"Mobility, Data Mining, and Privacy"* and (DATASIM) *"DATA science for SIMulating the era of electric vehicles"*, and an industrial award by MTN Cyprus.

References

[1] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proceedings of the 24th IEEE Intl. Conference on Data Engineering*, ICDE '08, pages 376–385, 2008.

[2] O. Abul, F. Bonchi, and M. Nanni. Anonymization of moving objects databases by clustering and perturbation. *Inf. Syst.*, 35(8):884–910, 2010.

[3] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *Proceedings of the 28th Intl. Conference on Very Large Data Bases*, VLDB '02, pages 143–154, 2002.

[4] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, July 1970.

[5] C.-Y. Chow, M. F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *Proceedings of the 14th Annual ACM Intl. Symposium on Advances in Geographic Information Systems*, GIS '06, pages 171–178, 2006.

[6] M. Dong and L. Zhong. Self-constructive high-rate system energy modeling for battery-powered mobile systems. In *Proceedings of the 9th Intl. Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 335–348, 2011.

[7] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan. Private queries in location based services: Anonymizers are not necessary. In *Proceedings of the ACM SIGMOD Intl. Conference on Management of Data*, SIGMOD '08, pages 121–132, 2008.

[8] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st Intl. Conference on Mobile Systems, Applications and Services*, MobiSys '03, pages 31–42, 2003.

[9] Y. Gu, A. Lo, and I. Niemegeers. A survey of indoor positioning systems for wireless personal networks. *Communications Surveys Tutorials, IEEE*, 11(1):13–32, 2009.

[10] B. D. Higgins, J. Flinn, T. J. Giuli, B. Noble, C. Peplin, and D. Watson. Informed mobile prefetching. In *Proceedings of the 10th Intl. Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 155–168. ACM, 2012.

[11] A. Khoshgozaran and C. Shahabi. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In *Proceedings of the 10th Intl. Conference on Advances in Spatial and Temporal Databases*, SSTD'07, pages 239–257, 2007.

[12] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *Proceedings of the Intl. Conference on Pervasive Services*, ICPS '05, pages 88–97, 2005.

[13] J.-S. Kim, Y. Han, and K.-J. Li. K-anonymity in indoor spaces through hierarchical graphs. In *Proceedings of the Fourth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, ISA '12, pages 21–28, New York, NY, USA, 2012. ACM.

[14] A. Konstantinidis, G. Chatzimilioudis, C. Laoudias, S. Nicolaou, and D. Zeinalipour-Yazti. Towards planet-scale localization on smartphones with a partial radiomap. In *Proceedings of the 4th ACM Intl. Workshop on Hot Topics in Planet-scale Measurement*, HotPlanet '12, pages 9–14, 2012.

[15] A. Konstantinidis, G. Nikolaides, G. Chatzimilioudis, G. Evagorou, D. Zeinalipour-Yazti, and P. K. Chrysanthis. Radiomap prefetching for indoor navigation in intermittently connected wi-fi networks. In *Proceedings of the 16th IEEE International Conference on Mobile Data Management*, MDM '15, pages 34–43, Pittsburgh, PA, USA, 2015. IEEE Press.

[16] C. Laoudias, G. Constantinou, M. Constantinides, S. Nicolaou, D. Zeinalipour-Yazti, and C. G. Panayiotou. The airplace indoor positioning platform for android smartphones. In *Proceedings of the 13th IEEE Intl. Conference on Mobile Data Management*, MDM '12, pages 312–315, 2012.

[17] G. Larkou, C. Costa, P. G. Andreou, A. Konstantinidis, and D. Zeinalipour-Yazti. Managing smartphone testbeds with smartlab. In *Proceedings of the 27th Intl. Conference on Large Installation System Administration*, LISA '13, pages 115–132, 2013.

[18] B. Li, J. Salter, A. G. Dempster, and C. Rizos. Indoor positioning techniques based on wireless lan. In *Proceedings of the 1st Intl. Conference on Wireless Broadband and Ultra Wideband Communications*, pages 13–16, June 2006.

[19] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of the 23rd IEEE Intl. Conference on Data Engineering*, ICDE '07, pages 106–115, April 2007.

[20] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1067–1080, Nov 2007.

[21] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen. A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, IPSN '15, pages 178–189, New York, NY, USA, 2015. ACM.

[22] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1):25–36, Mar. 2007.

[23] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern. Worst-case background knowledge for privacy-preserving data publishing. In *Proceedings of the 23rd IEEE Intl. Conference on Data Engineering*, ICDE '07, pages 126–135. IEEE Computer Society, 2007.

[24] S. Mascetti, L. Bertolaja, and C. Bettini. Safebox: adaptable spatio-temporal generalization for location privacy protection. *Trans. on Data Privacy*, 7(2):131–163, 2014.

[25] M. E. Nergiz, M. Atzori, and Y. Saygin. Towards trajectory anonymization: A generalization-based approach. In *Proceedings of the SIGSPATIAL ACM GIS 2008 Intl. Workshop on Security and Privacy in GIS and LBS*, SPRINGL '08, pages 52–61, 2008.

[26] N. Pelekis, A. Gkoulalas-Divanis, M. Vodas, A. Plemenos, D. Kopanaki, and Y. Theodoridis. Private-hermes: A benchmark framework for privacy-preserving mobility data querying and mining methods. In *Proceedings of the 15th Intl. Conference on Extending Database Technology*, EDBT '12, pages 598–601, 2012.

[27] L. Petrou, G. Larkou, C. Laoudias, D. Zeinalipour-Yazti, and C. G. Panayiotou. Crowdsourced indoor localization and navigation with anyplace. In *Proceedings of the 13th Intl. Symposium on Information Processing in Sensor Networks*, IPSN '14, pages 331–332, 2014.

[28] F. Qiu and J. Cho. Automatic identification of user interest for personalized search. In *Proceedings of the 15th Intl. Conference on World Wide Web*, WWW '06, pages 727–736, 2006.

[29] D. Quercia, I. Leontiadis, L. McNamara, C. Mascolo, and J. Crowcroft. Spotme if you can: Randomized responses for location obfuscation on mobile phones. In *Proceedings of the 31st Intl. Conference on Distributed Computing Systems*, ICDCS '11, pages 363–372, June 2011.

[30] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. on Knowl. and Data Eng.*, 13(6):1010–1027, Nov. 2001.

[31] L. Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, Oct. 2002.

[32] Y. Xia and C. K. Yeo. Mobile internet access over intermittent network connectivity. *Journal of Network and Computer Applications*, 40(0):126 – 138, 2014.

[33] M. L. Yiu, G. Ghinita, C. S. Jensen, and P. Kalnis. Outsourcing search services on private spatial data. In *Proceedings of the 25th IEEE Intl. Conference on Data Engineering*, ICDE '09, pages 1140–1143, 2009.

[34] M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu. Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In *Proceedings of the 24th IEEE Intl. Conference on Data Engineering*, ICDE '08, pages 366–375, 2008.

[35] D. Zeinalipour-Yazti, C. Laoudias, C. Costa, M. Vlachos, M. I. Andreou, and D. Gunopulos. Crowdsourced trace similarity with smartphones. *IEEE Trans. on Knowl. and Data Eng.*, 25(6):1240–1253, 2013.

Andreas Konstantinidis (Ph.D., University of Essex, UK, 2009) is currently a Visiting Lecturer with the Department of Computer Science and Engineering, Frederick University and a Post-Doctoral Researcher with the Department of Computer Science, University of Cyprus. His current research interests include mobile and sensor systems as well as the application of intelligent techniques to these environments.

Georgios Chatzimilioudis (Ph.D., University of California - Riverside, USA, 2010) is currently a Visiting Lecturer at the Department of Computer Science, University of Cyprus. His current research interests include mobile and sensor data management, distributed query optimization and crowdsourcing with smartphones.

Demetrios Zeinalipour-Yazti (Ph.D., University of California - Riverside, USA, 2005) is an Assistant Professor at the Dept. of Computer Science, University of Cyprus. He has also held positions at the the Open University of Cyprus and Akamai Technologies (MA, USA). His research interests include data management in systems and networks.

Paschalis Mpeis (BSc., University of Cyprus, Cyprus, 2012) is a Graduate Student at the School of Informatics of the University of Edinburgh. His research interests include mobile systems and distributed databases.

Nikos Pelekis (Ph.D., UMIST, UK, 2002) is a Lecturer at the Dept. of Statistics and Actuarial Science, University of Piraeus. His research interests include data mining, spatiotemporal databases, management of location-based services, machine learning and geographical information systems.

Yannis Theodoridis (Ph.D., National Technical University of Athens, Greece, 1996) is a Professor at the Dept. of Informatics, University of Piraeus. His research interests include database management, in particular mobility data management and exploration.