A Low-Complexity Trajectory Privacy Preservation Approach for Indoor Fingerprinting Positioning Systems

Amir Mahdi Sazdar[a], Seyed Ali Ghorashi[a,b,*], Vahideh Moghtadaiee[c], Ahmad Khonsari[d,e], David Windridge[f]

[a] *Cognitive Telecommunication Research Group, Department of Electrical Engineering, Shahid Beheshti University G. C., Tehran 1983963113, Iran.*
[b] *School of Architecture, Computing and Engineering, University of East London, London, UK.*
[c] *Cyberspace Research Institute, Shahid Beheshti University G. C. Tehran 1983963113, Iran.*
[d] *Dept. of ECE, College of Engineering, University of Tehran, Tehran 1417466191, Iran.*
[e] *School of CS, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran.*
[f] *Dept. of Computer Science Middlesex University, London, UK.*

Abstract

Location fingerprinting is a technique employed when Global Positioning System (GPS) positioning breaks down within indoor environments. Since Location Service Providers (LSPs) would implicitly have access to such information, preserving user privacy has become a challenging issue in location estimation systems. This paper proposes a low-complexity k-anonymity approach for preserving the privacy of user location and trajectory, in which real location/trajectory data is hidden within k fake locations/trajectories held by the LSP, without degrading overall localization accuracy. To this end, three novel location privacy preserving methods and a trajectory privacy preserving algorithm are outlined. The fake trajectories are generated so as to exhibit characteristics of the user's real trajectory. In the proposed method, no initial knowledge of the environment or location of the Access Points (APs) is required in order for the user to generate the fake location/trajectory. Moreover, the LSP is able to preserve privacy of the fingerprinting database from the users. The proposed approaches are evaluated in both simulation and experimental testing, with the proposed methods

*Corresponding author
Email address: a_ghorashi@sbu.ac.ir (Seyed Ali Ghorashi)

outperforming other well-known k-anonymity methods. The method further exhibits a lower implementation complexity and higher movement similarity (of up to 88%) between the real and fake trajectories.

Keywords: `Location Privacy-Preserving, Trajectory Privacy-Preserving, Fingerprinting Positioning, k-anonymity.`

1. Introduction

Indoor Positioning Systems (IPSs) are used for navigation inside of buildings, for example, to localize customers in commercial stores, or to rescue people in emergency situations. Various IPS services are designed for positioning purposes based on geographic information in conjunction with the Received Signal Strength Intensity (RSSI) from Wi-Fi Access Points (APs). The information measured at a single location, constuting the "Fingerprint", is saved by the Location Service Provider (LSP) as part of the fingerprinting radio map. This radio map consists of Reference Point (RP) coordinates and the measured RSSI [1]. In fingerprinting positioning, users are localized by comparing their RSSI values with the RP information stored by the LSP. However, this positioning technology potentially threatens user privacy due to the user's locations and trajectories being held by the LSP.

User trajectory tracking without explicit legal consent is generally considered a privacy violation as it potentially also exposes places of interest, social lives, even psychological aspects of the individual [2]. In this paper, it is assumed that LSPs are adversarial with respect to the user, and may provide the user's spatial information to unauthorized third parties or other LSPs. For instance, consumer interest in specific products in a store may be important for competitor stores and advertising agencies. Similarly, monitoring of patient location in a healthcare-related context may lead to insurance-relevent information being exposed. Therefore, it is critical to preserve patient/consumer location and trajectory information [3, 4].

In this paper, privacy preservation of users with respect to their locations

and trajectories during fingerprint positioning is addressed. In the proposed privacy preservation method, k-1 fake locations and trajectories employing a signal path-loss model are generated in order to incorporate k-anonymity in the trajectories. The contributions of the paper are summarized as follows:

1) Three different algorithms are proposed for preserving user location privacy by generating fake fingerprints for fake locations, making it possible to control the distance between the real and fake user location.
2) A low-complexity algorithm is introduced for preserving user trajectory privacy by generating fake trajectories with high movement similarity between the real and fake trajectories. No encryption, hash functions, or clustering are required.
3) The proposed methods are able to preserve the privacy of the LSP's fingerprint database by preventing any additional information leakage concerning the fingerprinting radio map, including measured fingerprints and location of RPs or APs.
4) The proposed methods are also able to maintain the localization accuracy identical to that prior to applying the privacy preservation algorithm to the fingerprinting network (more complicated methods tend to degrade positioning accuracy).

The structure of the paper is follows: in Section 2, the basics of fingerprinting positioning and privacy protection methods are outlined. Section 3 describes the proposed privacy preservation method. The proposed method is assessed via simulation and experimental testing in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

In this section we give a review of previous studies concerning fingerprint positioning and related privacy-preservion methods.

2.1. Fingerprint Positioning Technique

There is substantial extent research in the localization field, the most common technology being Global Positioning System-based (GPS-based) [5]. However, this typically fails within buildings due to the sharp drop in the intensity of satellite signals [6]. Alternative methods available for positioning such as ground-based navigation systems (e.g., LOng RAnge Navigation (LORAN) and SHOrt RAnge Navigation (SHORAN) System), Inertial Navigation System (INS), infrared localization, audio/video analytical localization [7, 8] typically require additional infrastructure such as transmitters, antennas and specific radars, all of which need to be deployed and adjusted before the actual localization procedure.

In fingerprinting positioning, however, existing Wi-Fi APs are employed as signal transmitters with the receivers being the users' smartphones [9]. Fingerprinting positioning consists of two phases: training (offline) and localization (online). The offline process collects RSSI from all existing APs at the known RPs and stores them along with the (x, y) coordinates of the RPs in the fingerprinting database [10]. The recorded RSSI matrix has a $N \times M$ dimension, where N, M are the number of RPs and APs, respectively. The fingerprint at a location (x_i, y_i) is given as $\boldsymbol{F}_i = [RSSI_{i1}, RSSI_{i2}, ..., RSSI_{iM}]$, where $RSSI_{ij}$ is the RSSI of AP_j at the i^{th} RP. In the localization phase, the user creates a vector of RSSIs from the fingerprints of its location and sends it to the LSP. The LSP then calculates the best match location (x, y) by comparing the RSSI of the user with the RSSI of all RPs formerly stored in the database. There are many matching algorithms that can be used in fingerprinting localization. The most common algorithms fall into one of two classes; probabilistic (e.g., Bayesian algorithms) [1] and deterministic (e.g., Nearest Neighbor (NN), K-Nearest Neighbor (KNN), and weighted KNN (WKNN)) [1, 9]. The procedure for fingerprinting localization in both the offline and online phases is shown in Figure 1.

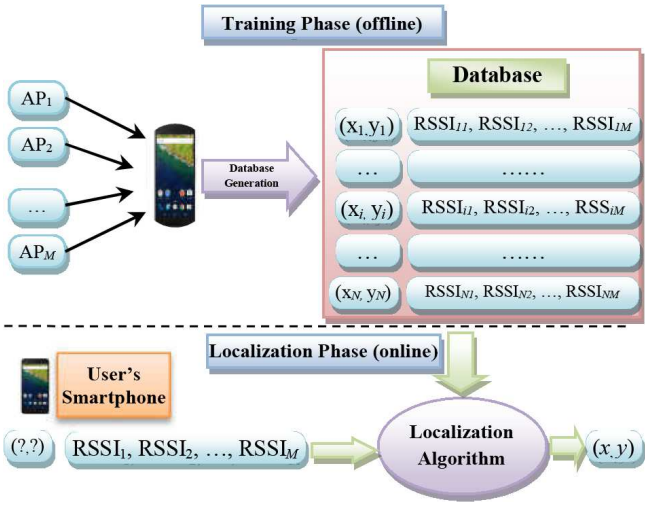


Figure 1: Fingerprinting localization technique.

2.2. Privacy-Preserving Methods

In this section, related privacy-preserving concerns are addressed. In the privacy challenge, insider and outsider attackers are considered the two main threats. Insider attackers are malicious or curious parts of the network, while outsider attackers are third parties or other out-of-network operators [11]. In fingerprinting positioning, LSPs are treated as trusted parties and user location privacy is revealed for them. However, this is a serious privacy risk when facing curious LSPs. Therefore, in this section, after addressing the main privacy-preserving concepts, related studies on the user's privacy preserving against curious LSPs are reviewed.

One of the most important methods to protect users' privacy is using a pseudo-identifier [12]. The Pseudo-identifier is considered as an interface between the data and certain characteristics of the user, such as birth, gender, postal code, and so on. Modifying these attributes results in a quasi-identification. The Pseudo-identifier tries to change these features, so that users cannot be identified and tracked [13]. In location privacy, these mechanisms are divided into three general categories: ambiguity, anonymity, and cryptography. In ambiguity mechanisms, the spatial resolution of the user is decreased to pre-

serve the exact level of their presence. In some cases, a set of fake locations is produced for the user to protect their privacy [14]. Points Of Interest (PoI) and adding random data are two examples of ambiguity methods [15]. The main idea behind anonymity mechanisms is to hide the user amongst k-1 different users, known as k-anonymity [16]. In this method, the probability of identifying the user is $1/k$ with a guarantee of hiding the user among k-1 users [15, 16]. Other methods are based on cryptography; these generally impose a significant computational overhead on the system [17]. A combination of the above primary methods can be also used to increase users' privacy levels. For instance, [18] proposes a new model for preserving the user privacy by combining PoI in the populated areas with double encryptions in sparse blocks.

Most of the aforementioned methods assume an outside privacy attacker and are not able to hide the user from the LSP. In order to address this challenge with respect to a curious LSP, authors in [19] reduced LSP knowledge about user location by using a k-anonymity Bloom (k-AB) filter and sending a Partial Radio Map (PRM) to the user. The user determines the $1/k$ relevent probabilities for his/her security and conceals themself between k other users, finally estimating their location by using the PRM, such that the LSP can only estimates user location with a $1/k$ probability. However, sending PRM to the users still reveals the privacy of the fingerprinting database. Authors in [20] proposed a privacy protection method utilizing homomorphic encryption and Paillier cryptosystem and claim they can protect the user's location privacy and the LSP's database at the same time. However, authors in [21] showed that the technique in [20] cannot preserve LSP's database information, because homomorphic encryption methods send the nearest distance vector of the fingerprint database to the user and expose the LSB's fingerprint database. Therefore, a malicious user can exploit the fingerprinting database by sending some specific fingerprint (e.g. zero fingerprint) to the LSP [21]. Authors in [21] introduced four methods to prevent this threat and they explained a model in [22] for Wi-Fi based indoor localization using Paillier encryption or Semi-Trusted Third Parties (STTPs) which preserve the user location privacy and LSP's fingerprint database.

Authors in [23] improved their previous method in [21] and provided a method for protecting users' privacy adopting the secret sharing encryption and the use of two STTPs with secure channel communication. They mapped the 8-bit RSSI to 4-bit values to reduce the network transmission traffic and to make faster localization process. This method works faster than other introduced methods due to the use of 4-bit values, however, the average positioning error increases about 0.1-5% compared to the case that 8-bit values are used.

Authors in [24] suggested scanning various active 802.11 Wi-Fi signals to preserve the user's privacy in indoor fingerprinting positioning in both the online and offline phases, regarding active Wi-Fi scanning as a kind of privacy leakage. In their method, a mobile device passively listens to the beacons' signal and determines the valid RSSI fingerprints, and does not send any signal out. They further improved their solution in [25] by introducing an obfuscating approach in fingerprinting positioning via a passive scanning procedure. In [25], all Wi-Fi signals are scanned offline by the frequency hopping technique in different time intervals, and in the online phase real RSSIs are scanned and the localization is carried out by deterministic and probabilistic approaches. Wang et al. [26] addressed the privacy problem in terms of client location and LSB database. They proposed a Differential Privacy (DP)-based privacy-preserving for IPS, which includes four phases, AP fuzzification, location retrieval, DP-based finger clustering and finger permutation. The two first phases run on the user's device and the two last phases run in the LSP. Finally, the authors in [27] introduced a method for producing a forged trajectory using a dummy signal strength. In this method, the user is required to estimate the location of APs. Therefore, the user needs to have a basic knowledge of the environment and to know RSSI values at several locations in the area in order to estimate the APs' location [27].

3. Proposed Method

In this section, the proposed methods for preserving location and trajectory privacy from curious LSPs are described. All of the proposed methods, can be considered as an instance of the k-anonymity method, as the user hides their real location/trajectory between k-1 other locations/trajectories, so the LSP or attackers can only estimate the location of the user with a $1/k$ probability, in which k ($k > 1$) is pre-specified by the user.

The major differences of the proposed methods with the previous research described in Section 2 are as follows: here, we propose a low-complexity privacy preserving algorithm in which no encryption methods or hash functions are required to preserve users' location/trajectory privacy; therefore, no preparation steps or key exchange phases are required. With no encryption procedure, the proposed method can also work faster than those using encryption methods. Moreover, utilizing the suggested algorithm in a location fingerprinting system does not increase the localization error and maintains the same level of accuracy as before. In addition, the full radio map or PRM is not accessible to users. Hence, it preserves the LSP's fingerprint database at the same time. Finally, unlike [27], no knowledge about AP locations and measured fingerprints in the indoor environment are necessary in the proposed methods.

3.1. Location Privacy-preservation

To protect the location privacy of the user, first we need to preserve the actual fingerprints of the user. Therefore, the proposed method initially generates k-1 fake fingerprint vectors, all of which are sent to the LSP along with the real fingerprint of the user, $\mathbf{F}_u$. All of these k elements are placed in an arbitrary order in a vector, denoted by **FP**, as $\mathbf{FP}=[\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_k]$. In other words, **FP** includes k **F** vectors, only one of which is the true fingerprint of the user, $\mathbf{F}_u$. The LSP sends back k locations in a vector as $\mathbf{Loc} = [Loc_1, Loc_2, ..., Loc_k]$, without knowing which are the true or fake **F** vectors. However, the user knows the index of the true **F** in the **FP** vector, so it is able to choose the corresponding location in the **Loc** vector received from the LSP. From the LSP's point

of view, the location of the user can be any of those k locations. Therefore, the probability of identifying the true location of the user is $1/k$. These fake fingerprints are made with three different algorithms explained in subsection 3.1.1. The protocol for preserving user location privacy from the LSP is shown in Algorithm 1.

Algorithm 1 User Side Location Privacy-Preserving Protocol.

Output: k Locations (one real and k-1 fake locations)
1: Scan the RSSI and create $\mathbf{F}_u$
2: Select a random positive number k ($k > 1$)
3: Generate k-1 fake $\mathbf{F}$ vectors ($\mathbf{F}_1$, $\mathbf{F}_2$, ..., $\mathbf{F}_{k-1}$)
4: Send $\mathbf{FP}$ vector,[$\mathbf{F}_1$, $\mathbf{F}_2$, ..., $\mathbf{F}_k$], to the LSP (knowing $\mathbf{F}_i$ is $\mathbf{F}_u$)
5: Get $[Loc_1, Loc_2, ..., Loc_k]$ from LSP
6: Select $\mathbf{Loc}_i$ as the real location

For example, assume that the user RSSI vector from five existing APs is $\mathbf{F}_u$= (-75, -65, -60, -80, -70) dBm. Selecting $k = 3$, the user produces two fake $\mathbf{F}$ vectors and sends $\mathbf{FP} = [\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3]$ to the LSP in an arbitrary order, knowing which element is the true index. The LSP then sends back three locations corresponding to these $\mathbf{F}$ vectors and the user can recognize its true location. In this case, the LSP can only estimates the true location of the user with a 1/3 probability. Algorithm (2) shows the implementation of this protocol in the server.

Algorithm 2 Server Side Location Privacy-Preservation Protocol.

Input: k $\mathbf{F}$ vectors ([$\mathbf{F}_1$, $\mathbf{F}_2$, ..., $\mathbf{F}_k$])
Output: k Locations ($[Loc_1, Loc_2, ..., Loc_k]$)
1: Receive $\mathbf{FP}$ vector($[\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_k]$) from the user $\mathbf{u}$
2: For each $\mathbf{F}_i$ in $\mathbf{FP}$ vector Calculate Loc_i
3: Send $[Loc_1, Loc_2, ..., Loc_k]$ back to the user $\mathbf{u}$

3.1.1. Methods for Generating Fake Fingerprints

For generating the fake fingerprint vectors, three methods are herein outlined: random fingerprints, random permutations, and smart permutations.

Random fingerprints: Fake **F** vectors are generated by selecting values in the range of [-24, -110] dBm, which is the typical RSSI range for Wi-Fi signals. The maximum value of -24 dBm is chosen as the nearest distance to the AP (note that this value is obtained in our actual measurements). -110 dBm is considered to represent the absence of signals from the AP. Here, fake locations are completely random, and the user cannot change them.

Random permutation: This method utilizes a random permutation of the RSSIs in the real **FP** vector. In order to generate the fake **F** vectors, users can change the order of elements in the real $\mathbf{F}_u$. If there are M APs in the area, there are $M!$ permutations for $\mathbf{F}_u$ elements to generate the fake **F** vectors. Therefore, there are $\begin{pmatrix} M! \\ k-1 \end{pmatrix}$ fake **F** vectors. In order to create the **FP** vector then, the **F** vectors can have $k!$ orders. Hence, if the user sends $\boldsymbol{r}$ requests for localization to the LSP, there are $\Pr = r \times k! \times \begin{pmatrix} M! \\ k-1 \end{pmatrix}$ possible locations for the user. By increasing k, the location privacy of the user is increasingly preserved.

Smart Permutation: Here, the fake **F** vectors are chosen via a smart method such that the user can control the generated fake locations, producing them according to specific distances to the real location. This method is based on the principle that the user generally receives a strong RSSI value from the nearest AP and a weak RSSI value from the farther one. In the smart permutation method, users select an arbitrary percentage number (Per) and change ***Per%*** of the elements in the $\mathbf{F}_u$ vector. The user can select the value of ***Per*** based on the level of protection needed. The higher ***Per*** values in the smart permutation method result in a larger average distance between the real and fake locations, which provides higher level of protection. In order to obtain the furthest fake location possible, users carry out replacement of the maximum elements of the $\mathbf{F}_u$ vector with the minimum ones and vice versa until ***Per%*** of all elements have been replaced altogether. In this smart method, if the intensities of the two APs with the highest and lowest values are displaced together, the distance between the real and the fake locations also increases. However, if the RSSI

values of two APs are roughly the same, the intensity of the signals in the fake fingerprint, obtained via the $\mathbf{F}_u$, is approximately the same and the real and fake locations will be also close to each other. Algorithm (3) shows the proposed smart permutation of the real $\mathbf{F}_u$ vector. In this method, the user can change ***Per%*** of $\mathbf{F}_u$ elements. If ***Per*** is 100% all of $\mathbf{F}_u$ elements are changed.

Algorithm 3 Create Fake **FP** by Smart Permutation.

Input: $\mathbf{F}_u$ and Permutation Percent (***Per***)
Output: Fake **F**

```
Fake F = F_u
repeat
    while Max and Min RSSI values are changed do
        Select next Max and Min RSSI
    end while
    Swap(Max and Min RSSI in Fake F)
until (Per% elements of the Fake F are changed )
return Fake F
```

3.2. Trajectory Privacy-Preserving

If the user is moving, the generated fake **F** vectors might produce some locations irrelevant to the previous location of the user. Therefore, it is possible for the LSP to monitor and analyze all estimated locations of the user and look for the most related and rational ones based on the spatial distances and request times, as well as the average speed of the person in question. Hence, we need to protect the trajectory of the user at the same time.

In the proposed method, fake **F** vectors have been modified according to real changes in $\mathbf{F}_u$ such that unauthorized observers or LSPs would not be able to distinguish the difference between the real and fake trajectories. For this purpose, a distance-calculation method using a path-loss model is used to calculate the variations of user's real location and change the fake **F** vectors accordingly. In many studies, signal propagation inside buildings is modeled by

a known log-distance path-loss model as in the case below [28, 29]

$$\log\, d = \frac{1}{10n}(P_{TX} - P_{RX} + G_{TX} + G_{RX} - X_{\alpha} + 20\log\lambda - 20\log(4\pi)) \quad (1)$$

where N is the path-loss exponent (**PE**), d is the estimated distance between the transmitter and the receiver, and P_{TX} and $P_{RX}(dBm)$ are the transmitted and received powers, respectively. G_{TX} and $G_{RX}(dBi)$ are the antenna gains of the transmitter and receiver. $\lambda(m)$ denotes the wavelength of the signal and X_{α} models the path-loss variation at one point due to the shadowing caused by obstacles in propagation assumed to be a zero mean Gaussian random variable with a standard deviation given by α. The standard deviation of X_{α} is in the range of 3 to 20 dB, depending on the construction of the building and the number of partitions the signal passes through [28]. The distance d can be computed using equation (2) as follows.

$$d = 10^{\left((P_{TX} - P_{RX} + G_{TX} + G_{RX} - X_{\alpha} + 20\log\lambda - 20\log(4\pi))/10n\right)} \quad (2)$$

3.2.1. Proposed Method for Trajectory Privacy Preservation

The initial points for the fake trajectories are generated using the methods proposed in subsection 3.1.1. As the user relocates and receives new signal strengths from the APs, the fake **F** vectors should change accordingly. Suppose that in the previous example, after the user is translated to the new location, $\mathbf{F}_u$ has been changed to $\mathbf{F}_u$ = (-85,-59,-65,-82,-68) dBm. The difference between these two vectors is $\mathbf{DF}_u$ = (-10,+6,-5,-2,+2). The changes indicate that the user moves away from the first, third and fourth APs and closer to the second and fifth APs.

To preserve the trajectory, the **F** vector of the other k-1 fake locations will also need to change accordingly. By utilizing $\mathbf{DF}_u$ and Equation (2), we are able to calculate the appropriate distances that the user should move in relation to different APs. According to these distance calculations, the RSSI values of the new fake locations are incremented or decremented with respect to the

RSSI values of their previous locations. There is, however, the possibility that changing the fake RSSI values generates an out-of-range ([-24,-110] dBm) value. In this case, we reverse the movement direction in the fake trajectory with regard to that particular AP.

To improve trajectory preservation in the proposed method, the specific **PE** of the area of interest is used in (2) instead of generalised values. Hence, we suppose that LSP has the **PE** values corresponding to each AP and sends a M-element Path-loss Exponent Vector (**PEV**) along with the estimated **Loc** vector to the user. Then, the user employs **PEV** to create the next fake location. The proposed method for preserving the user's trajectory is presented in Algorithm 4). As the user creates k-1 fake trajectories alongside the real one, the probability of the trajectory correctly tracking is $1/k$.

4. Performance Evaluation

The proposed methods are verified via simulation and experimental testing; the results are presented in this section.

4.1. Numerical Results on Simulated Data

In the simulation of the localization process and production of the fake trajectories, we employ an area of dimensions $100\,m \times 100\,m$ without obstacles. There are 9 APs and 100 RPs in the simulated environment. In the simulation process, the power of the APs and receivers are assumed to be $20dBm$ with frequency $2.4GHz$; the antennas' gain is $5dBi$, and λ is $0.12m$. Signal variations are assumed to have a Gaussian distribution, $N(0, 5)$. The localization process's requests are simulated by a Poisson random variable and the users' trajectory is simulated via a Random Walk simulation model [30].

Figure 2 shows an example implemention of the proposed method on the simulated data for 20 localization requests. In Figure 2, the location of APs and RPs are shown with red crosses and black dots. The real user trajectory is plotted as a green line and the four fake trajectories via four dashed lines. In a

Algorithm 4 User Trajectory Privacy-Preservation Protocol.

Input: $\mathbf{F}_u$ Vector
Output: Fake **F** Related to the Real Trajectory

$\star\downarrow\star$ **User Side (Location Privacy)** $\star\downarrow\star$

Scan the location RSSI and create the $\mathbf{F}_u$ vector
Select a random, positive integer k ($k > 1$)
Create k-1 Fake **F** vectors base on different ***Per%*** values
Send **FP** vector (**FP**=[$\mathbf{F}_1$, $\mathbf{F}_2$, ..., $\mathbf{F}_k$]) to the LSP

$\star\downarrow\star$ **Server Side (Localization Process)** $\star\downarrow\star$

Estimate k locations
Generate **PEV**
Send $[Loc_1, Loc_2, , Loc_k]$ and **PEV** to the user

$\star\downarrow\star$ **User Side (Trajectory Privacy)** $\star\downarrow\star$

User selects its real location and stores the other k-1 fake locations for creating k-1 fake trajectories
Move to the new location
Scan RSSI and create new $\mathbf{F}_u$ vector
Calculate $\mathbf{DF}_u$
Calculate the distance (d) from each AP with equation (2)
Changing the RSSI values of Fake **F** vectors using (d) and **PEV**
for each invalid RSSIj value in the Fake **F** vector **do**
    Reverse Trajectory Direction from AP_j
    Change $RSSI_j$ value using new direction
end for
Create the real and Fake **F** vectors
Send **FP**s to the LSP and Start Server Side Section

new localization request, the RSSI values of the fake **F** vectors change according to the difference in the new and previous $\mathbf{F}_u$ vector. As it can be seen in Figure 2, there are four created trajectories in addition to the real one. We see that the progression of the real and fake trajectories are practically identical, and the probability of discovering the real trajectory is 1/5.

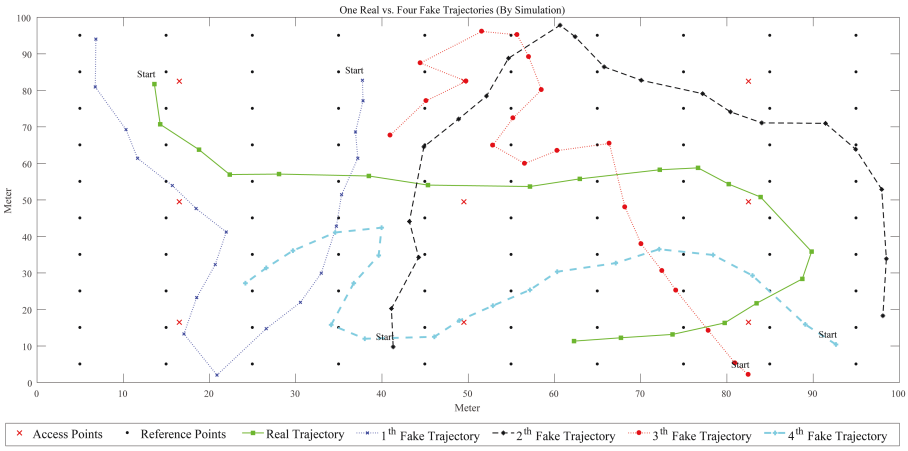


Figure 2: An example of the proposed method on the simulated data when k=5 and four fake trajectories are generated alongside the real one.

4.2. Experimental Test and Results

In order to evaluate the proposed method in realistic situations, we setup a real-world experiment (we shall first detail the experimental testing; the result of location privacy preservation using the suggested three methods for the fake fingerprints generation are then discussed, and finally the proposed method for trajectory privacy preserving is analyzed).

4.2.1. Experimental Setup

The RSS fingerprint samples are recorded by a Samsung Galaxy SM-J500H smartphone over two days using two developed Android-based applications, the first for data-acquisition from existing APs, in which RSSI values and Media Access Control (MAC) addresses of sensed APs are simultaneously sent to the server and recorded (Wampserver software is utilized for the server side); the second is for the localization process. The testbed is located in the Cyberspace Research Institute at Shahid Beheshti University with an approximate dimension of $17\,m \times 50\,m$. In this experiment, there are nine Wi-Fi APs located in the ceilings. To build the fingerprinting database, the RSSI at 354 RPs are measured in four directions (North, South, West and East, each direction 100 samples). The size of the fingerprinting database is $6.4MB$. We also calculate

the **PEV** including the path-loss exponent of all nine APs, **PEV**=[3.64, 3.51, 4.28, 3.21, 4.27, 4.26, 4.27, 5.33, 4.91]. The average value of **PE** is 3.7 for the entire area. Figure 3 displays the map of the surveyed environment. In this figure, red-cross and black dotted are represented the APs and RPs, respectively. The location algorithm is implemented via the KNN method when $K = 4$, as this has demonstrated the best positioning accuracy.

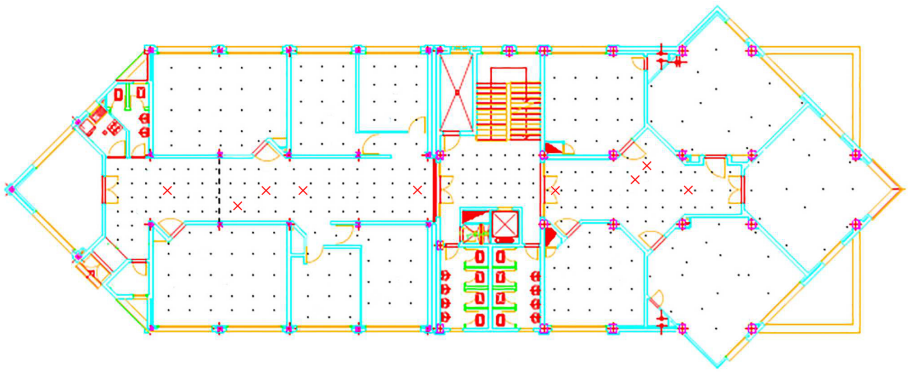


Figure 3: Testbed layout showing the locations of APs (red-crosses), and RPs (black dots).

4.2.2. Results on Generated Fake Fingerprints

Here, three proposed algorithms for generating the fake locations are compared. Table 1 represents the implementation of suggested methods for producing fake locations, with each method executed 100 times. In Table 1, the first row shows the localization error for the estimated locations of all RPs using the KNN leave-one-out method. In this case, the average distance error from the user's real location is $1.11m$. In the second and third rows, random fingerprints and random permutations of the real fingerprints are shown. They provide an average distance of $7.38m$ and $6.25m$ between the real and fake locations, respectively. However as indicated before, the user cannot control the distance in these two algorithms. The last row of Table 1 represents the smart permutation method, in which for **Per** = 100%, we can achieve an average distance of $11.45m$ from the true location. However, changing 25% of elements only gives an average distance of $1.56m$. Therefore, higher **Per** values in the smart permutation method result in a larger average distance between the real and fake locations.

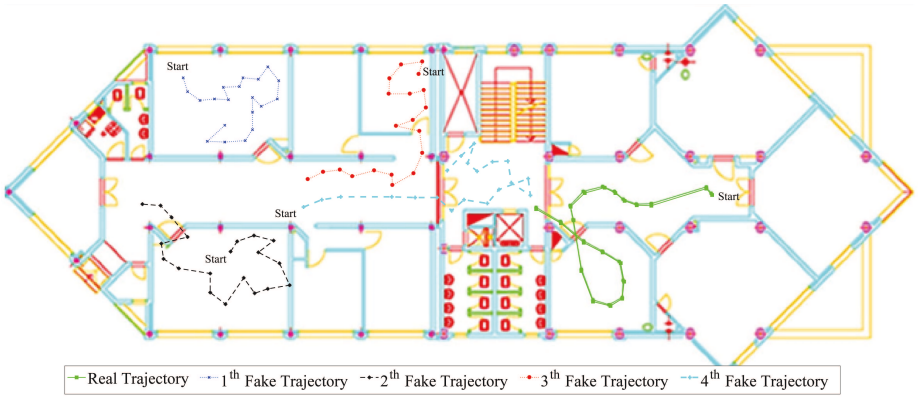
Table 1: Distance between the real and fake user locations after 100 interations of fake location generation.

Generated Fake Method	Parameters	Distance from User Real Location (m)		
		Avg.	Max.	Min.
Only Localization Process	KNN , K=4	1.11	6.31	0.02
Random Fingerprints	Random[-110,-15]	7.38	21.50	0.40
Random Permutation	Permutation	6.25	21.28	0.37
Smart Permutation	25%	1.56	6.11	0.15
	60%	3.67	18.81	0.21
	100%	11.45	19.74	0.76

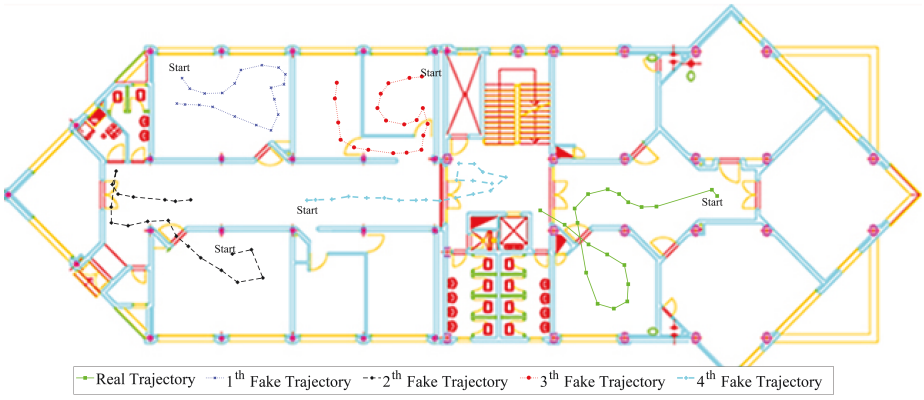
4.2.3. Results on Generated Fake Trajectories

The proposed fake trajectories are analyzed and compared with the real one here. Figure 4 illustrates the results of the proposed method using the actual measured data for $k = 5$ and 20 localization requests. Figure 4-(a) utilizes the average **PE** of $n = 3.7$ for all APs whereas Figure 4-(b) uses the corresponding **PE**s for each APs. In these figures, a green solid line represents the real trajectory and the other four fake lines are shown with blue, black, red and turquoise dashed lines. As it can be seen in Figure 4, the moving speed of the user is kept almost similar in all fake trajectories and there is no sharp jump for them. Therefore, the proposed method is able to hide the real trajectory between four other fake trajectories.

In Table 2, the average movements of the user in the proposed method are shown for 100 localization requests when the average **PE** and **PEV** are used. The average user's movement between two positioning requests in the real trajectory is $1.78m$ and in four fake trajectories is $1.58m$ employing **PEV**. The results indicate that the proposed method can maintain the similarity between the real and fake trajectories. Furthermore, Table 2 shows the movement similarity between the real and fake trajectories is 88% when **PEV** is utilized, whereas using the average **PE** decreases the similarity value to 77%. Due to the achieved high similarity, it would be difficult to distinguish the real user trajectory based on the user's movements.



(a) The average **PE** ($n = 3.7$) for all APs.



(b) The specific **PEV** vector for each AP.

Figure 4: Results of the proposed method using the actual measured data for k = 5 and 20 localization requests (green line is real and other line are four fake trajectories)

Table 2: Average movement of the user for 100 positioning requests in the real and fake trajectories.

Trajectory		Average Movement (m)	
		$n = 3.7$	**PEV**
Real Trajectory		1.12	1.78
Fake Trajectories	1^{th}	0.77	1.47
	2^{th}	1.07	1.68
	3^{th}	0.80	1.70
	4^{th}	0.81	1.47
	Average($1^{th} \sim 4^{th}$)	0.86	1.58
Difference Between Real and Average ($1^{th} \sim 4^{th}$) Fake Trajectories		**0.26**	**0.20**

Figure 5 is the screenshot of our mobile application in a real situation for trajectory privacy preservation with $k = 3$ and four positioning requests. Green squares show the real trajectory of the user and the black/blue signs are two fake trajectories. The average user's movement between two positioning requests in the real and two fake trajectories are $2.95m$, $2.84m$ and $2.96m$, respectively, which shows highly correlating movements between them. When positioning requests are increased from 4 to 100, the average difference of movements between the real and fake trajectories is $0.25m$.

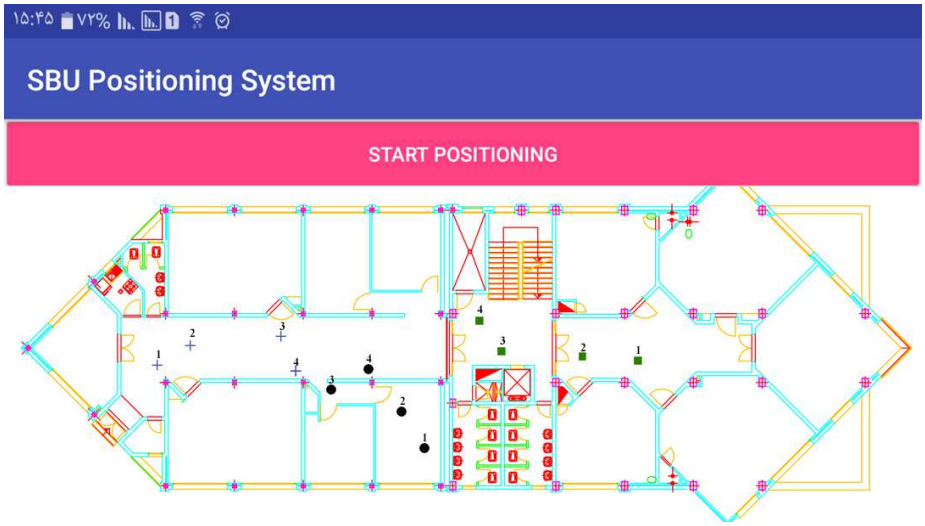


Figure 5: A screenshot of our mobile application in a real situation of trajectory privacy preservation for four positioning requests (Green squares are the real trajectory. Black circles and blue pluses show the two fake trajectories).

Table 3 compares the proposed method with the methods presented in [19], [20], [23], [26] and [27] for 100 localization requests and $k = 5$. The Table shows that localization errors do not change in the proposed method and [27] after employing privacy preserving algorithms since they utilize the LSP fingerprint database information and KNN method for localization. The algorithm in [27] needs to know different real locations and their measured fingerprints to estimate APs' location, whereas the suggested method here does not need such information. Using PRM and k-AB filter in [19], homomorphic cryptography in [20], reducing the required RSSI bits in [23] and clustering methods in [26] degrade the localization accuracy.

Furthermore, the computational overhead in the proposed method is much lower than [20], [23] and [26] and more straightforward than [19] and [27] as it does not employ any cryptographic or clustering process or hash function. Moreover, the proposed method does not require a secure channel as it is needed in [23] to communicate with the STTPs. The proposed method can also preserve the LSP's fingerprinting database similar to [23], [26] and [27], whereas users have access to PRM in [19]. In addition, user motion similarity with respect to the real and fake trajectories of the proposed method is the highest overall (88%), implying the movements in the fake trajectories are very similar to those of the real trajectory although the proposed privacy preserving algorithm is intrinsically less complicated, and no information about the fingerprints and AP location is required.

Table 3: Comparison of the proposed method with [19], [20], [23], [26] and [27].

Comparison Type		**Methods**					
		[19]	[20]	[23]	[26]	[27]	Proposed
Average positioning error (Meter)		1.28	1.83	1.97	1.62	1.11	1.11
Computational overhead	LSP	Finding related RPs and create PRM	Homomorphic encryption and decryption ([23] in two STTP)		Clustering and finding related RPs in each cluster	k times positioning	k times positioning
	User's device	hash functions of Bloom filter and location estimation by PRM	Homomorphic encryption and decryption ([23] share generation)		Fingerprint permutation and finding location from clustered radio maps	Identify the approximate location of the APs and dummy generated fingerprints	Fake fingerprints generation
Network's traffic		Fingerprint, bloom filter information and PRM	Encrypted fingerprint and encrypted matrix of nearest distance ([23] at least 40% less than [20])		Clustered radio maps	k times positioning	k times positioning
Initial knowledge		no	Key exchange phase		no	N different known locations	no
Privacy-preserving approach		k-Anonymity	Encryption/ Differential Privacy		k-Anonymity/ Differential Privacy	k-Anonymity	k-Anonymity
Preserving the radio map		no	no	yes	yes	yes	yes
Movement similarity		87%	80% $\sim$ 85%	78% $\sim$ 83%	85%	86%	88%
Similarity to error ratio		0.68	0.55	0.53	0.62	0.77	0.79
Average running time (Sec)		1.82	2.43 Paillier(mod 512)	0.78 [0.54 $\sim$ 0.98]	2.16	1.72	1.63

The movement similarity to average localization error ratio is shown in the eighth row of Table 3. Higher values of this ratio for a given method indicate the relative benefits of that method over the others. For the the same movements, the method that degrades the positioning accuracy the most receives the lowest ratio value. Likewise, between two methods with the same localization errors, the one with a lower movement similarity provides a lower ratio value. This ratio for the suggested method is 0.79, the highest overall among the tested methods, demonstrating the advantage of the proposed algorithm in preserving trajectory privacy. Finally, the last row of Table 3 shows the maximum running time occurred for the algorithms specified in [20], [26], and [19] as they used cryptographical concepts, clustering methods, and hash functions along with user side activities on PRMs, respectively. Although the proposed method is slower than the method introduced in [23] (because [23] uses 4-bit RSSI values), our proposed method provides a higher localization accuracy. The running times given in [27] and that of the proposed methods are the same; the proposed method is slightly faster, however.

5. Conclusion

In this paper, a class of low-complexity privacy preservation methods are proposed for user location/trajectory without employing hash functions or encryption algorithms. In the location privacy preserving method, the user is hidden among $k-1$ fake locations by using smart permutation of fingerprints to produce fake locations. In the suggested trajectory privacy preserving method, fake trajectories are generated by the user without knowing any information about the environment or the measured fingerprints. Employing the proposed algorithms in the location fingerprinting system does not decrease the positioning accuracy. In the fake trajectories approach, the user movement characteristics are kept almost similar to the real trajectory. By utilizing the proposed method, we have demonstrated that it is possible to achieve up to 88% movement similarity between the real and fake trajectories. Therefore, the possibility of

trajectory detection by curious LSPs is greatly reduced. The proposed method is not only is able to preserve the location and trajectory of the user but also protect the fingerprinting database at the same time, as the LSP does not send any additional information or PRM to the user. For future work, we will seek to use the proposed method for outdoor localization in order to demonstrate its versatility. We also propose to improve fake trajectories by identifying obstacles and walls in order to avoid sudden trajectory transitions. Finally, we will try to decrease the running time with methods such as the RSSI bit mapping introduced in [23].

References

[1] A. Pérez-Navarro, J. Torres-Sospedra, R. Montoliu, J. Conesa, R. Berkvens, G. Caso, C. Costa, N. Dorigatti, N. Hernández, S. Knauth, et al., Challenges of fingerprinting in indoor positioning and navigation, in: Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation, Elsevier, 2019, pp. 1–20.

[2] N.-D. T. Tieu, H. H. Nguyen, H.-Q. Nguyen-Son, J. Yamagishi, I. Echizen, Spatio-temporal generative adversarial network for gait anonymization, Journal of Information Security and Applications 46 (2019) 307–319.

[3] G. Wang, R. Lu, C. Huang, Y. L. Guan, An efficient and privacy-preserving pre-clinical guide scheme for mobile eHealthcare, Journal of Information Security and Applications 46 (2019) 271–280.

[4] R. H. Khokhar, R. Chen, B. C. Fung, S. M. Lui, Quantifying the costs and benefits of privacy-preserving health data publishing, Journal of biomedical informatics 50 (2014) 107–121.

[5] Z. Zhuang, K.-H. Kim, J. P. Singh, Improving energy efficiency of location sensing on smartphones, in: Proceedings of the 8th international conference on Mobile systems, applications, and services, ACM, 2010, pp. 315–330.

[6] Y. Gu, A. Lo, I. Niemegeers, A survey of indoor positioning systems for wireless personal networks, IEEE Communications Surveys & Tutorials, 11 (1), 2009.

[7] S. He, S.-H. G. Chan, Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons, IEEE Communications Surveys & Tutorials 18 (1) (2016) 466–490.

[8] V. Moghtadaiee, A. G. Dempster, Indoor location fingerprinting using FM radio signals, IEEE Transactions on Broadcasting 60 (2) (2014) 336–346.

[9] Q. D. Vo, P. De, A survey of fingerprint-based outdoor localization, IEEE Communications Surveys and Tutorials 18 (1) (2016) 491–506.

[10] Z. E. Khatab, A. Hajihoseini, S. A. Ghorashi, A fingerprint method for indoor localization using autoencoder based deep extreme learning machine, IEEE sensors letters 2 (1) (2017) 1–4.

[11] A. Alzubaidi, J. Kalita, Authentication of smartphone users using behavioral biometrics., IEEE Communications Surveys and Tutorials 18 (3) (2016) 1998–2026.

[12] P. Kalnis, G. Ghinita, K. Mouratidis, D. Papadias, Preventing location-based identity inference in anonymous spatial queries, IEEE transactions on knowledge and data engineering 19 (12) (2007) 1719–1733.

[13] D. Smith, Secure pseudonymisation for privacy-preserving probabilistic record linkage, Journal of Information Security and Applications 34 (2017) 271–279.

[14] A. S. Saxena, D. Bera, V. Goyal, Modeling location obfuscation for continuous query, Journal of information security and applications 44 (2019) 130–143.

[15] I. J. Vergara-Laurens, L. G. Jaimes, M. A. Labrador, Privacy-preserving mechanisms for crowdsensing: Survey and research challenges, IEEE Internet of Things Journal 4 (4) (2017) 855–869.

[16] L. Sweeney, k-anonymity: A model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (05) (2002) 557–570.

[17] S. Zakhary, A. Benslimane, On location-privacy in opportunistic mobile networks , a survey, Journal of Network and Computer Applications 103 (2018) 157–170.

[18] I. J. Vergara-Laurens, D. Mendez, L. G. Jaimes, M. Labrador, A-PIE: An algorithm for preserving privacy, quality of information, and energy consumption in participatory sensing systems, Pervasive and Mobile Computing 32 (2016) 93–112.

[19] A. Konstantinidis, G. Chatzimilioudis, D. Zeinalipour-Yazti, P. Mpeis, N. Pelekis, Y. Theodoridis, Privacy-preserving indoor localization on smartphones, IEEE Transactions on Knowledge and Data Engineering 27 (11) (2015) 3042–3055.

[20] H. Li, L. Sun, H. Zhu, X. Lu, X. Cheng, Achieving privacy preservation in Wi-Fi fingerprint-based localization, in: IEEE Conference on Computer Communications (INFOCOM), IEEE, 2014, pp. 2337–2345.

[21] Z. Yang, K. Järvinen, The death and rebirth of privacy-preserving Wi-Fi fingerprint localization with paillier encryption, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 1223–1231.

[22] Z. Yang, K. Järvinen, Modeling privacy in Wi-Fi fingerprinting indoor localization, in: International Conference on Provable Security, Springer, 2018, pp. 329–346.

[23] K. Järvinen, H. Leppäkoski, E.-S. Lohan, P. Richter, T. Schneider, O. Tkachenko, Z. Yang, PILOT: Practical privacy-preserving indoor localization using outsourcing, in: 2019 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2019, pp. 448–463.

[24] L. Schauer, F. Dorfmeister, F. Wirth, Analyzing passive Wi-Fi fingerprinting for privacy-preserving indoor-positioning, in: International Conference on Localization and GNSS (ICL-GNSS), IEEE, 2016, pp. 1–6.

[25] L. Schauer, Wi-Fi tracking threatens users' privacy in fingerprinting techniques, in: Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation, Elsevier, 2019, pp. 21–43.

[26] Y. Wang, M. Huang, Q. Jin, J. Ma, DP3: A differential privacy-based privacy-preserving indoor localization mechanism, IEEE Communications Letters 22 (12) (2018) 2547–2550.

[27] H. Li, Y. He, X. Cheng, L. Sun, A lightweight location privacy-preserving scheme for Wi-Fi fingerprint-based localization, in: International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI), IEEE, 2016, pp. 525–529.

[28] A. Bose, C. H. Foh, A practical path loss model for indoor Wi-Fi positioning enhancement, in: 6th International Conference on Information, Communications & Signal Processing, IEEE, 2007, pp. 1–5.

[29] M. Hossain, Y. Jin, W.-S. Soh, H. N. Van, SSD : A robust RF location fingerprint addressing mobile devices' heterogeneity, IEEE Transactions on Mobile Computing 12 (1) (2013) 65–77.

[30] Z. Zhu, G. Cao, Toward privacy preserving and collusion resistance in a location proof updating system, IEEE Transactions on Mobile Computing 12 (1) (2013) 51–64.