A privacy-preserving protocol for indoor Wi-Fi localization

Samuel N. Eshun
University College Cork
Cork, T12 YN60, Ireland
s.eshun@cs.ucc.ie

Paolo Palmieri
University College Cork
Cork, T12 YN60, Ireland
p.palmieri@cs.ucc.ie

ABSTRACT

Location-aware applications have witnessed massive worldwide growth in recent years due to the introduction and advancement of smartphones. Most of these applications rely on the Global Positioning System (GPS) which is not available in indoor environments. As a result, Wi-Fi fingerprinting is becoming increasingly popular as an alternative as it allows localizing users in indoor environments, has lower power consumption, and is also more economical as it does not require a dedicated sensor other than a Wi-Fi card. The technique allows a service provider (SP) to construct a Wi-Fi database (called radio map) that can be used as a reference point to localize a user. However, this process does not preserve the user privacy, as the location can only be computed interactively with the SP. The service provider may also reveal sensitive information on the indoor space (*e.g.* the building map) to the user. Thus, we need an indoor localization protocol that addresses the privacy of both parties. In this paper, we present a privacy-preserving cryptographic protocol for indoor Wi-Fi localization, that prevents the SP from learning the exact location of the user outside of certain pre-defined sensitive areas, while keeping the SP's database secure. Thus, both parties cannot learn anything about each other's input beyond the implicit output revealed.

KEYWORDS

location privacy, cryptographic protocols, Bloom filter

ACM Reference Format:
Samuel N. Eshun and Paolo Palmieri. 2019. A privacy-preserving protocol for indoor Wi-Fi localization. In *Proceedings of Proceedings of the 16th conference on Computing Frontiers (CF '19).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Location-aware applications in recent years have witnessed a massive growth worldwide. Most of these applications rely on Global Navigation Satellite Systems (GNSS) for users of such applications to obtain their location. An example of a popularly used GNSS is the Global Positioning System (GPS). However, GPS perform better when used in an open environment (outdoors) and is mostly unavailable or has very poor signals if used indoors. Instead, the Wi-Fi-fingerprinting-based localization approach [1, 4, 19] is currently regarded as the most promising technique for providing accurate information for indoor areas. In particular, a major advantage of this method is that the existing Wi-Fi-infrastructure, based on the IEEE 802.11 standard [8], makes it easier and cheaper to deploy with no additional hardware cost.

In order to infer a user's location, Wi-Fi fingerprinting requires a location fingerprinting database known as a *radio map*, built by recording Received Signal Strength (RSS) from various access points (APs) in different locations [9, 11]. Therefore, fingerprinting localization has two phases: the offline and the online phase. In the offline phase, the Wi-Fi RSS from all the various APs of known positions in the building or desired indoor area are recorded, in order to build a database (radio map) saved on the service provider's server. The online phase, where the user interacts with the service, comes with two sub-phases. First, the user measures the location fingerprint (RSS) from all the APs in range at its current location (a grid point). The location fingerprint is then sent to the central server to make a computation which compares the RSS received to the reference radio map built in the offline phase, with the help of a trained decision, in order to determine the location of the user in the building.

The computation of the location can be done, among others, using Euclidean distance, likelihood estimator, or support vector machine for regression [3]. Although Wi-Fi-fingerprinting localization is promising, unlike GNSS, it can violate user's location privacy or leak user's location information [13, 14, 17]. This is due to the fact that the user computes the location interactively with the server, therefore revealing their location to the service provider. One of the proposed solutions is to allow the client to have full access to the radio map to compute the position locally [12, 19]. However, this approach burdens not only the already constrained computational resources of the user (especially in mobile settings) but can also lead to a serious privacy breach of the service providers's database. In particular, this could leak the entire map of a particular building. If we consider sensitive settings such as an airport, this is clearly not desirable. To solve this problem, most of the literature focuses on the privacy of the user neglecting the Service Provider (SP) [12, 19], thereby exposing the SP's database. Another area that has not been fully addressed so far, is how can the SP have real-time access to the user's location without at the same time violating the user's privacy. For example, an employer may wish to monitor his employees in areas deemed sensitive, without violating their privacy elsewhere in the building; and can this be achieved indoor with a secured database that cannot be used for malicious activities by users.

1.1 Contribution

In this paper, we present an indoor location privacy-preserving protocol that allows a service provider to query the location of a user without violating their location privacy. The protocol protects

CF '19, April 30-May 2, 2019, Alghero, Italy

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

both the user and the SP's privacy but still delivers the service based on the user's location. Thus, both the SP and the client are mutually distrusting, hence a typical problem of secure multi-party computation. The protocol builds on the data structure presented by Palmieri *et al.* [5, 6, 16] to protect users' privacy. The protocol also uses homomorphic encryption and the (spatial) Bloom filter (SBF) [5, 6, 16], a modification of the classical Bloom filter [2], to build trust among both parties.

In particular, we summarize the main contributions of this paper below:

- To the best of our knowledge, the proposed scheme is the first to use a probabilistic data structure (SBF, a variant of the original Bloom filter) to address privacy of indoor localization based on Wi-Fi-fingerprinting. While other privacy-preserving protocols for indoor localization exist in the literature (such as [13], later expanded in [18]), this paper does not focus on how the user can obscure the location from the service provider, but on a broader perspective: how a service provider in a private manner can have real-time access to users location without violating their privacy.
- The proposed construction includes an efficient decision algorithm over additive homomorphic encryption that can locate the user with a high probability without leaking any location information, achieving private indoor localization over Wi-Fi fingerprinting.
- Further to this, the partial homomorphic computation (based on Paillier's cryptosystem [15] or any other scheme providing additive homomorphism) used to encrypt location information enables privacy-preserving location-based applications based on the private indoor localization. In particular, it guarantees the privacy of both the service provider and the user, such that when the user is not at a sensitive area the SP learns nothing about the location.
- Finally, the proposed scheme only utilizes the existing Wi-Fi infrastructure, and doesn't require new devices for the indoor environment or additional resources.

2 PRIVATE INDOOR WI-FI LOCALIZATION

In this section, we give a general overview of Wi-Fi-fingerprinting. The process can be divided into two-phases: offline (performed by the service provider), and online (which involves both parties, service provider and user).

In the offline phase, the SP records the Wi-Fi-fingerprinting known as received signal strength in an area assumed to be a rectangular grid of points of interest (x_i, y_i). Each location point (x_i, y_i) where $i = 1 \cdots l$ falls within a range of an Access Points (AP), and the pair (x,y) is the location coordinate of the fingerprint. The RSS M-dimensional vector $V_i = \{r_1, r_2, \cdots, r_M\}$ where M is the total number of access points within the range of a location (x_i, y_i) , and $M \leq N$. In reality, M will always be less than N, because no location (x_i, y_i) can be within the range of all access points N. A database (radio map) is created as a reference point to localize a user during the online phase. An example of the database is shown in Table 2. In practice, the RSS signal quality measured in decibels is considered exceptional when it's below -40, very good between -40 to -55 and very low beyond -80. We assign -100 to indicate an access point is unavailable.

Table 1: NOTATIONS AND MEANINGS

Symbol	Meaning
RSS	Received signal strength
r_j	RSS value of an access point j
V_u	RSS values of user(u)
V	RSS values for all N AP
C	Encrypted RSS values
$\mathbf{B} = \{\triangle_i\}_{i=1}^{s}$	The set of area of interest
SBF	The spatial Bloom filter
$B^{\#}$	SBF over the set of areas B
$\triangle_i$	An area of interest
$\mathbf{Enc}(B^{\#})$	Encryption of $B^{\#}$ using pk_s
pk_u	Public key of user U
sk_u	Secrete key of user U
pk_s	Public key of server
sk_s	Secrete key of server
$\mathbf{Enc}(\mathbf{pk}_u, V_u)$	Encryption of V_u using pk_u
$\{(\theta_i)\}_{i=1}^{k}$	The k nearest neighbours
$\{\mathsf{Enc}(\theta_i)\}_{i=1}^{k}$	Encryption of KNN
t_u	The exact location of user U
B_u	SBF build over t_u
$B_u^{\#}$	Output of $\mathsf{Enc}(B^{\#})$ and B_u

In the online phase, the user using a positioning algorithm gathers RSS values $V_{u_i} = \{r_1, r_2, \cdots, r_M\}$ at a location (x_i, y_i) . These RSS values will be compared with RSS tuples $V_i = \langle\{r_{ij}\}_{j=1}^{M}\rangle_{i=1}^{l}$ for l locations and N access points in the database of the SP.

Table 2: WI-FI-FINGERPRINT REFERENCE DATABASE

l_i	ap_1	ap_2	ap_3	ap_4	$\cdots$	ap_n
1	$r_{1,1}$	$r_{1,2}$	-100	-100	$\cdots$	$r_{1,n}$
2	-100	$r_{2,2}$	$r_{2,3}$	$r_{2,4}$	$\cdots$	-100
3	-100	-100	$r_{3,3}$	$r_{3,4}$	$\cdots$	$r_{3,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
l	$r_{l,1}$	$r_{l,2}$	$r_{l,3}$	-100	$\cdots$	-100

The Euclidean distance θ_i, which is a measure between V and V_u is then computed to localize the position of the user. There are various distance algorithm (Manhattan distance, Minkowski distance, *etc.*) that can be used but in this paper, we use the Euclidean distance :

$$\theta_i = \|V_i - V_{u_i}\| = \sqrt{\sum_{j=1}^{N}(V_i - V_{u_i})^2} \tag{1}$$

Due to difficulty in making computation involving square root of encrypted values, the squared Euclidean distance between V and V_u is rather used in our algorithm as it achieves the same result:

$$\begin{aligned}\theta_i = \|V_i - V_{u_i}\|^2 &= \sum_{j=1}^{N}(r_{ij} - r_{u_i})^2 \\ &= \sum_{j=1}^{N} r_{ij}^2 + \sum_{j=1}^{N} r_{u_i}^2 + \sum_{j=1}^{N} - 2r_{ij} \cdot r_{u_i}\end{aligned} \tag{2}$$

The k nearest neighbours $\{\theta_i\}_{i=1}^{k}$ are then selected and the average computed to localize the position of the user.

2.1 The two-party protocol

The description of the Wi-Fi-indoor location algorithm didn't factor any privacy concern and the computational limitations at the user's side. In this paper we present a novel algorithm where all these concerns are taken into consideration. In addition, unlike other location algorithms, where applications are based on the user's exact location, here the SP tries to learn the location of the user without violating their privacy: in particular, the protocol will never reveal the user's exact location to the service provider, but only whether the user is within any area of interest. Areas of interest are pre-defined by the SP (before execution of the protocol). If the user is outside such an area, the provider learns nothing.

The two-party computation involves the user and the SP, and the communication happens over an authenticated channel. That is, both sides are sure the messages they are receiving are coming from the right source. With this protocol the SP is interested in learning the location of the user but doesn't want the user to have access to the database used in computing the location, whereas the user doesn't want the SP to learn the exact location but only presence in a predefined area of interest. This kind of problem is called secure two-party computation where both sides cannot learn anything about their respective inputs, beyond the explicit output.

Protocol 1 Privacy preserving protocol for indoor Wi-Fi

User		**Server**
$\mathsf{Enc}(B^{\#})$		$\mathsf{Enc}(B^{\#}),\ V$
$V_u \leftarrow\$ \{0,1\}$		
$C \leftarrow\$ \mathsf{Enc}(\mathsf{pk}_u, V_u)$		$\mathsf{Enc}(B^{\#}) \leftarrow\$ \mathsf{Enc}(\mathsf{pk}, B^{\#})$
	$\xrightarrow{C = \{c_j\}_{j=1}^{M}}$	
		$\mathsf{Enc}(\theta_i) \leftarrow\$ \mathsf{Enc}(V_u) \odot V$
	$\xleftarrow{\{\mathsf{Enc}(\theta_i)\}_{i=1}^{k}}$	
$\theta_i \leftarrow\$ \mathsf{Dec}(\mathsf{sk}_u, \mathsf{Enc}(\theta_i))$		
$t_u = (x,y) = loc\ \{\theta_i\}_{i=1}^{k}$		
$B_u = create(t_u)$		
$B_u^{\#} = \mathsf{Enc}(B^{\#}) \odot B_u$		
	$\xrightarrow{B_u^{\#}}$	
		$\mathsf{Dec}(\mathsf{sk}, B_u^{\#})$
		$i \leftarrow\$ \{\triangle_i\}_{i=1}^{s}$
		$0 \leftarrow\$ Otherwise$
finalize		

The proposed Protocol 1 addresses the concerns of both parties by revealing only the area of the user's location but nothing else to the SP. The nature of the SBF filter, a variant of Bloom filter [2] only makes it possible for binary input. That is, given a vector of n -bits initially set to zeroes, k-independent hash functions are used to map locations (elements) unto the filter by setting their vector corresponding positions to 1. The user then shuffles these positions to prevent the SP from detecting the exact location of the user. On the flip-side, the SP's database is also protected from malicious users.

We now give a detailed description of the protocol. The protocol assumes a planar of rectangular grid covering finite points of interest (PoI) with a space of 1 m between two grid points. Each point of interest is within the range of M access points of a total of N access points in the building or system.

In the offline phase, SP records the Wi-Fi-fingerprints multiple times at each location for each access point and the average is computed. Thus r_i will be the average of r_{ij}. Thus if $j = 3$, then r_i will be the average of measuring three times for each access point at position (x_i, y_i). Like Table 2, the database is constructed for all points of interest.

Algorithm 2 Spatial Bloom filter algorithm

Input : $\{\triangle_i\}_{i=1}^{s},\ H \leftarrow\$ \{h_j\}_{j=1}^{q}$
Output : $B^{\#}$

```
for i ←$ 1 ··· s do
    for j ←$ 1 ··· q do
        foreach t ∈ △_i do
            B#[h(t)] ←$ i
return B#
```

The SP constructs a spatial Bloom filter (SBF) [5, 6, 16], a variant of the Bloom filter [2] and a data structure that allows spatial representation. Like the original Bloom filter, values are inserted using a one-way function that supports probabilistic membership queries, that is whether a member belongs the set or not. The SP constructs an SBF over only areas of interest, then encrypts the filter $\mathsf{Enc}(B^{\#})$ and sends to the user.

The online phase is divided into four sub-phases: measuring, computation, private localization, and the sever localization.
In the measuring phase, a user u, using a positioning algorithm obtains a fingerprint V_u and uses Pailler's cryptosystem to generate a pair of keys $< pk_u, sk_u >$. The key generation can be executed once and offline, and the secret key can be kept until it expires (or is compromised).

When the user receives a request from the SP, the user queries the local dataset V_u and uses the pk_u to encrypt each r_{ij} and sends it to the server as outlined in algorithm 3.

$$\begin{aligned} C &= \mathsf{Enc}(\mathsf{pk}_u, V_{u_j}) \\ &= \{c_1, c_2, \cdots, c_M\} \end{aligned} \tag{3}$$

$$c_j = \{\mathsf{Enc}(pk_u, -2r_j),\ \mathsf{Enc}(pk_u, r_j^2)\} \tag{4}$$

Algorithm 3 Private localization of user

Input : $V_u = \{r_j\}_j^M \longleftarrow_\$ \{0,1\}^n$
Output : $t_u = (x, y)$

USER OUTSOURCED LOCATION

1 : **for** $j \longleftarrow_\$ 1 \cdots M$ **do**
2 : $\quad \mathsf{Enc}(V_{u_j}) \longleftarrow_\$ \mathsf{Enc}(\mathsf{pk}, r_j)$
3 : $post\ \mathsf{Enc}(V_{u_j})$

USER PRIVATE LOCATION

4 : **get** $\{\mathsf{Enc}(\theta_i)\}_{i=1}^k\ from$ **S**
5 : **for** $i \longleftarrow_\$ 1 \cdots k$ **do**
6 : $\quad \theta_i \longleftarrow_\$ \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\theta_i))$
7 : $\quad t_u = (x, y) \longleftarrow_\$ mean\{\theta_i\}$
8 : $B_u = create(t_u)$
9 : $B_u^\# = \mathsf{Enc}(B^\#) \odot B_u$
10 : **post** $B_u^\#$ *to* **Server**

The computation phase: after the localization server had received the encrypted fingerprint $\{c_i\}_{i=1}^M$, it uses the Pailler's homomorphic additive properties to compute the Euclidean distance θ_i using a set of V_i from the reference database to obtain the encrypted k-nearest neighbours. As a further protection to the database against a malicious user, the server permutes the distance ciphertext and add a generated random number to it. The computation under encryption is similar to Equation 2. Basically, the server encrypts the squares of the selected training fingerprint r_{ij}^2 to output:

$$\mathsf{Enc}(\theta_i) = \prod_{j=1}^{N} \mathsf{Enc}(r_{ij}^2) \cdot \mathsf{Enc}(r_{u_j}^2) \cdot \mathsf{Enc}(-2r_{ij} \cdot r_{u_j}) \quad (5)$$
$$where \quad \mathsf{Enc}(-2r_{ij} \cdot r_{u_j}) = \mathsf{Enc}(-2r_j)^{r_{ij}}$$

Private location retrieval phase: the user receives the encrypted k-neartest neigbours $\mathsf{Enc}(\theta_i)$ line 4 of algorithm 3, decrypts it with the private key sk_u to obtain the K nearest neighbours θ_i same as Equation 1. The user then estimate the private location t_u by computing the geometric center of the k-nearest neighbours.

Sever localization phase: in the localization, the user first uses algorithm 2 to construct a SPB over the exact location t_u.The user who has access to SP's encrypted SBF $\mathsf{Enc}(B^\#)$ over the area of interest (AoI) will perform entry-wise homomorphic product of the encrypted SBF $\mathsf{Enc}(B^\#)$ and B_u to obtain $B_u^\#$ as illustrated in algorithm 3.The user then randomizes the order of values in the filter by shuffling $\mathsf{Enc}(B^\#)$, the output of the encrypted filter and sends to the server.

When the server receives the encrypted filter $\mathsf{Enc}(B^\#)$, it uses the private key $\boldsymbol{sk_s}$ to decrypt the filter and performs location verification using algorithm 5. Should any of the hash function $h \in H$ return 0, then certainly the user is not in an area of interest.

Algorithm 4 Localizing user's location by Server

Input : $\{\mathsf{Enc}(V_u)\}_{j=1}^M,\ \{V\}_{i=1}^N$
Output : $\triangle_i$

COMPUTING KNN IN ENCRYPTED FORMAT

1 : **for** $j \longleftarrow_\$ 1 \cdots N$ **do**
2 : $\quad$ **for** $j \longleftarrow_\$ 1 \cdots M$ **do**
3 : $\quad\quad \theta[i] \longleftarrow_\$ \mathsf{Enc}(V_{u_j}) \odot V_i$
4 : $post \quad \{\theta_i\}_{i=1}^k$

IMPORTING USER LOCATION

5 : **get** $\mathsf{Enc}(B_u^\#)\ from\ User$
6 : **for** $i \longleftarrow_\$ 1 \cdots S$ **do**
7 : $\quad$ **if** $\triangle_i = \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(B_u^\#))$
8 : $\quad\quad$ **return** $\triangle_i$
9 : $\quad$ **else** :
10 : $\quad\quad$ **return** 0

But if the output bits is greater than 0 for all hash functions, and the smallest output value is i, then the user's position is $t_u \in \triangle_i$ minus the false probability of the filter.

Algorithm 5 Verification of Spatial Bloom filter algorithm

Input : $B_u^\#,\ H \longleftarrow_\$ \{h_j\}_{j=1}^q, \mathbf{B} = \{\triangle_i\}_{i=1}^s$
Output : $\triangle_i$

1 : **for** $i \longleftarrow_\$ 1 \cdots s$ **do**
2 : $\quad$ **for** $j \longleftarrow_\$ 1 \cdots q$ **do**
3 : $\quad\quad$ **if** $B^\#[h(t_u)] = 0$:
4 : $\quad\quad\quad$ **return** 0
5 : $\quad\quad$ **else** $B^\#[h(t_u)] \longleftarrow_\$ i$:
6 : **return** $\triangle_i$

3 SECURITY

In this section, we present a security definitions for the two-party computation protocol, based on the *honest but curious* model[7]. Thus, the parties follow the instructions of the protocol but may try to learn more information than allowed by the protocol.

The two-party protocol is secure if at the end of execution, the exact location of the user is privacy preserved and the server learns only the predefined area of interest of the user's location but not the exact location. If the user is outside the predefined area of interest, then the server learns nothing about the user's location. The protocol preserves the service provider's privacy if, at the end of the protocol execution, the user is unable to obtain the list of the reference Wi-Fi-fingerprint (radio map), and the privacy of the predefined areas encoded in the filter is preserved.

3.1 Security Discussion

To begin with, the user's fingerprint sent to the server is encrypted using the semantically secure Paillier's cryptosystem. Hence the

server cannot determine the closest neighbours to the location of the user without the user's private key sk_u. In addition, the exact location of the user is protected from the server. This is achieved by using algorithm 3 line 9, by encoding the exact location of the user using (spatial) Bloom filter to produce an encrypted filter $B_u^{\#}$. Before the user sends $B_u^{\#}$ to the server, it permutes the filter by shuffling it to change the order of the values corresponding to the user's location. After decryption by the server, it will be impossible for the server to reconstruct the user's filter based on the order. Also, the security properties of the Paillier's encryption algorithm (or any other additively homomorphic encryption scheme) ensure that the server learns only limited values of $B^{\#}$ corresponding to the non-zero values of B_u [10], making it impossible to identify the user's exact location. The server will learn only the area of interest i if the number of non-zeroes corresponds to b and all the values take i the area of interest. If the number of non-zero values corresponds to b, but some of the values are $> i$, due to the collision of the filter then the server learns only the area of interest $\triangle_i$ and some patterns of values but not the exact location. However, if the number of non-zero values are $< b$, after the server decrypts $B_u^{\#}$ using algorithm 4 line 7 then certainly the server learns nothing about the user's location making it privacy preserved.

We argue the protocol also preserves privacy of the SP data. In that, anytime the user queries for the K nearest neighbours, the server blinds the result with a random number that introduces enough noise to prevent the user from knowing the real distances. Also, the permutation prevents the user from knowing the indexes of the closest neighbours though some information may be leaked after computing the centroid, this is insignificant. To a larger extent, the database is protected so far as both parties follow the instructions of the protocol. The privacy of the predefined areas are preserved thanks to the homomorphic encryption of the filter. From algorithm 3 line 9, the user can perform multiplication without knowing the cleartext of the predefined areas of interest inserted in the filter by the server.

We do acknowledge that a malicious user may attempt to build its own database by recording enough RSSI values over the same field. However, as we can assume the area is controlled, this will be practically unfeasible. Moreover, the malicious user needs almost the same amount of RSS values to make accurate predictions.

4 CONCLUSION AND FUTURE WORKS

In this paper, we propose a privacy-preserving protocol for indoor Wi-Fi-localization, that addresses the privacy of the user and the service provider. In particular, our scheme uses partial homomorphic encryption (Paillier's cryptosystem) which guarantees the location privacy of the user by performing the localization computation in the encrypted domain. This, in turn, ensures that most of the computational overhead at the user side is delegated to the server while hiding the exact location from the server. The use of the homomorphic encryption also preserves the data privacy of the service provider (SP). We use spatial Bloom filters in addition to homomorphic encryption enabling the SP to use location-based service to learn the predefined areas of the user, but not the exact location while preventing the user from learning these predefined areas. When the user is outside these predefined areas in the building, the service provider learns nothing about the user's location.

We finally discuss the security of the protocol considering semi-honest parties, *i.e.* they follow the instructions of the protocol but try to gain additional knowledge. For future work, we plan to extend the security features of the protocol to defend against malicious adversaries. A three-party model can be researched, to outsource most of the computation at the user side, which is useful in constrained embedded systems especially, defending against malicious adversaries. In a three-party scenario, further security guarantees for both the user and the server in the presence of third-party would need to be researched.

REFERENCES

[1] Paramvir Bahl and Venkata N. Padmanabhan. 2000. RADAR: An In-Building RF-Based User Location and Tracking System. In *Proceedings IEEE INFOCOM 2000, The Conference on Computer Communications, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Reaching the Promised Land of Communications, Tel Aviv, Israel, March 26-30, 2000*. IEEE, 775–784. https://doi.org/10.1109/INFCOM.2000.832252

[2] Burton H. Bloom. 1970. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Commun. ACM* 13, 7 (1970), 422–426. https://doi.org/10.1145/362686.362692

[3] Mauro Brunato and Roberto Battiti. 2005. Statistical learning theory for location fingerprinting in wireless LANs. *Computer Networks* 47, 6 (2005), 825–845. https://doi.org/10.1016/j.comnet.2004.09.004

[4] Luca Calderoni, Matteo Ferrara, Annalisa Franco, and Dario Maio. 2015. Indoor localization in a hospital environment using Random Forest classifiers. *Expert Syst. Appl.* 42, 1 (2015), 125–134. https://doi.org/10.1016/j.eswa.2014.07.042

[5] Luca Calderoni, Paolo Palmieri, and Dario Maio. 2015. Location privacy without mutual trust: The spatial Bloom filter. *Computer Communications* 68 (2015), 4–16. https://doi.org/10.1016/j.comcom.2015.06.011

[6] Luca Calderoni, Paolo Palmieri, and Dario Maio. 2018. Probabilistic Properties of the Spatial Bloom Filters and Their Relevance to Cryptographic Protocols. *IEEE Trans. Information Forensics and Security* 13, 7 (2018), 1710–1721. https://doi.org/10.1109/TIFS.2018.2799486

[7] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. 2015. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press.

[8] IEEE. 2016. IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. In *Proceedings IEEE INFOCOM IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2016*. https://doi.org/10.1109/INFCOM.2004.1356988

[9] Kamol Kaemarungsi and Prashant Krishnamurthy. 2004. Modeling of Indoor Positioning Systems Based on Location Fingerprinting. In *Proceedings IEEE INFOCOM 2004, The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, Hong Kong, China, March 7-11, 2004*. IEEE, 1012–1022. https://doi.org/10.1109/INFCOM.2004.1356988

[10] Hiroaki Kikuchi and Jun Sakuma. 2014. Bloom Filter Bootstrap: Privacy-Preserving Estimation of the Size of an Intersection. *JIP* 22, 2 (2014), 388–400. https://doi.org/10.2197/ipsjjip.22.388

[11] Elina Laitinen, Jukka Talvitie, and Prashant Krishnamurthy. 2011. Comparison of Positioning Accuracy of Grid and Path Loss-Based Mobile Positioning Methods Using Receiver Signal Strengths. In *Proceedings SPAMEC 2011, Signal Processing and Applied Mathematics for Electronics and Communications, August 26-28, 2011*. 1–4.

[12] Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian E. Smith, James Scott, Timothy Sohn, James Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello, and Bill N. Schilit. 2005. Place Lab: Device Positioning Using Radio Beacons in the Wild. In *Pervasive Computing, Third International Conference, PERVASIVE 2005, Munich, Germany, May 8-13, 2005, Proceedings (Lecture Notes in Computer Science)*, Hans-Werner Gellersen, Roy Want, and Albrecht Schmidt (Eds.), Vol. 3468. Springer, 116–133. https://doi.org/10.1007/11428572_8

[13] Hong Li, Limin Sun, Haojin Zhu, Xiang Lu, and Xiuzhen Cheng. 2014. Achieving privacy preservation in WiFi fingerprint-based localization. In *2014 IEEE Conference on Computer Communications, INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014*. IEEE, 2337–2345. https://doi.org/10.1109/INFOCOM.2014.6848178

[14] Liran Ma, Amin Y. Teymorian, and Xiuzhen Cheng. 2008. A Hybrid Rogue Access Point Protection Framework for Commodity Wi-Fi Networks. In *INFOCOM 2008. 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 13-18 April 2008, Phoenix, AZ, USA*. IEEE, 1220–1228. https://doi.org/10.1109/INFOCOM.2008.178

[15] Pascal Paillier. 1999. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding (Lecture Notes in Computer Science)*, Jacques Stern (Ed.), Vol. 1592. Springer, 223–238. https://doi.org/10.1007/3-540-48910-X_16

[16] Paolo Palmieri, Luca Calderoni, and Dario Maio. 2014. Spatial Bloom Filters: Enabling Privacy in Location-Aware Applications. In *Information Security and Cryptology - 10th International Conference, Inscrypt 2014, Beijing, China, December 13-15, 2014, Revised Selected Papers (Lecture Notes in Computer Science)*, Dongdai Lin, Moti Yung, and Jianying Zhou (Eds.), Vol. 8957. Springer, 16–36. https://doi.org/10.1007/978-3-319-16745-9_2

[17] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. 2011. Quantifying Location Privacy. In *32nd IEEE Symposium on Security and Privacy, S&P 2011, 22-25 May 2011, Berkeley, California, USA*. IEEE Computer Society, 247–262. https://doi.org/10.1109/SP.2011.18

[18] Zheng Yang and Kimmo Järvinen. 2018. The Death and Rebirth of Privacy-Preserving WiFi Fingerprint Localization with Paillier Encryption. In *2018 IEEE Conference on Computer Communications, INFOCOM 2018, Honolulu, HI, USA, April 16-19, 2018*. IEEE, 1223–1231. https://doi.org/10.1109/INFOCOM.2018.8486221

[19] Moustafa Youssef and Ashok K. Agrawala. 2005. The Horus WLAN location determination system. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, MobiSys 2005, Seattle, Washington, USA, June 6-8, 2005*, Kang G. Shin, David Kotz, and Brian D. Noble (Eds.). ACM, 205–218. https://doi.org/10.1145/1067170.1067193