IEEE 802.11 CSI randomization to Preserve Location Privacy: An Empirical Evaluation in Different Scenarios★

Marco Cominelli, Felix Kosterhon, Francesco Gringoli, Renato Lo Cigno, Arash Asadi

Abstract

Passive device-free localization of a person exploiting the Channel State Information (CSI) from Wi-Fi signals is quickly becoming a reality. While this capability would enable new applications and services, it also raises concerns about citizens' privacy. In this work, we propose a carefully-crafted obfuscating technique against one of such CSI-based localization methods. In particular, we modify the transmitted I/Q samples by leveraging an irreversible randomized sequence. I/Q symbol manipulation at the transmitter distorts the *location-specific* information in the CSI while preserving communication, so that an attacker can no longer derive information on user's location. We test this technique against a Neural Network (NN)-based localization system and show that the randomization of the CSI makes undesired localization practically unfeasible. Both the localization system and the CSI randomization are implemented on real devices. The experimental results obtained in our laboratory show that the considered localization method works smoothly regardless of the environment, and that adding random information to the CSI prevents the localization, thus providing the community with a system that preserve location privacy and communication performance at the same time.

Keywords: localization, privacy, channel state information, neural networks, Wi-Fi, randomization, experiments and measures

1. Introduction

The theme of precise localization of devices or people has long been of great interest from both a research and an industrial perspective, particularly indoor positioning, as GPS-based systems cannot work. One specific field of indoor positioning is the localization of "bodies"—human beings, but also other physical objects in the ambient—without these bodies being fitted with an active or passive communication device. Clearly, camera-based localization or anti-intrusion systems (e.g., radar- or lidar-based) are part of this latter field, but they are outside the scope of this work. We focus instead on systems based on Wi-Fi, as wireless communications signals are less detectable by users. In addition, such systems can take advantage of the widely deployed Wi-Fi infrastructure, thus paving the road to widespread surveillance systems, including illegal ones, as the presence of a standard Wi-Fi system is today almost ubiquitous.

In this respect, localization based on Channel State Information (CSI), whose variations can be correlated to changes in the physical environment, is extremely interesting [2, 3]. The authors in [4] pioneered this field by proposing to use advanced MIMO technologies combined with signal processing typical of radar systems with just three antennas and a Software Defined Radio (SDR) module to reveal the position, and even gestures, of a person behind a wall. Clearly such a system poses huge privacy concerns, as the localization can be obtained with proper devices even outside the room or building the person is in: A person moving within the room changes the signal propagation in the environment, which is in turn reflected in the CSI. This exposes the person to the risk of being tracked without having the possibility to avoid it, or even be aware of it. A possible countermeasure against Wi-Fi sensing attacks has been implemented in [5] to prevent gesture recognition; however, the proposed system relies on an additional component acting as a relay that must be placed in the environment. Moreover, obfuscation performance strongly depends on the position of such device. If no countermeasures are available, the only solution to stop such attacks consists in jamming or disabling all Wi-Fi communication in the vicinity, which is not desirable.

Thus, we asked ourselves a clear question: is it possible to explicitly alter the CSI to preserve privacy without hampering communications? This paper gives an initial and positive answer to this question. The key contribution of this work are the following: *i)* a multi-site measurement-based study that confirms the feasibility of passive localization of people based on CSI analysis beyond any reasonable doubt; and *ii)* introducing a CSI randomization technique that prevent unauthorized localization while maintaining good communication performance. A closely related work [6] was also published recently, which manipulates the CSI with the goal of avoiding radiometric fingerprinting of Wi-Fi chipsets and helping in the prevention of impersonation attacks. This latter work starts from the observation that hardware imperfections in devices lead to unique non-linear phase errors that can be used as a device fingerprint, and operate on the phase of signals to prevent fingerprinting.

In this article, we analyze the entire CSI characteristics finding the peak amplitude to be the characteristic carrying most information on location, and thus works on these amplitude peaks

★The initial results of this paper was partially presented at the 14th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization [1].

to obfuscate localization.

2. CSI-based Localization

Localization using Wi-Fi signals has a long literature story, starting from Received Signal Strength Indicator (RSSI) fingerprinting techniques, to mixed and fusion methods, and we refer the interested reader to a recent survey [7] for an overview. We are interested in techniques based on the analysis of CSI that, starting a decade ago, have emerged as the most powerful technique for Wi-Fi-based indoor localization [8, 9]. In particular we concentrate on methodologies based on Neural Network (NN) and Machine Learning (ML)/Deep Learning (DL) in general [2, 3, 10, 11, 12].

We exploit the CSI collector and extraction technique presented in [13, 14], which work on many different devices, like the Asus 4x4 Access Points. Once extracted, the CSI is fed to a NN-based system as described in detail [15] and summarized briefly in Section 2.2. Section 2.1 discusses the essentials of OFDM that are needed to understand both the NN design and the randomization technique discussed in Section 3.

2.1. OFDM Transmissions

Without any limitation of the proposed methodology, we focus the description on a 80 MHz OFDM transmission with a single spatial stream, i.e., transmitted by a single antenna. This type of modulation is described in the Very-High-Throughput (VHT) part of the standard; the corresponding physical level is called VHT-PHY: while being an extension of the legacy 20 MHz physical level, called OFDM-PHY, it adds many features including both Single-User (SU) and Multi-User (MU) MIMO capabilities, enhanced modulation rates and up to 8 spatial streams. We report here only the details useful to understand the randomization procedure presented in Section 3 and the rationale of the NN design, and we refer the interested readers to the standard [16] and classic literature as [17].

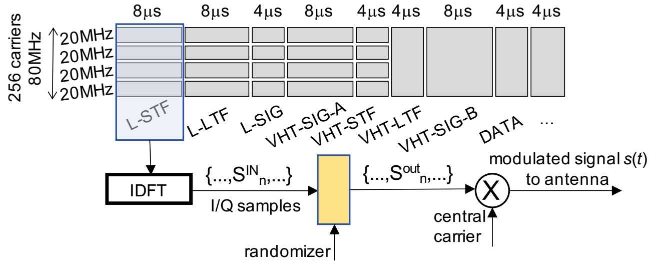


Figure 1: Format of an OFDM frame: initial symbols are known and some are used to infer the CSI at the receiver; the orange block is where randomization is introduced in Section 3.

OFDM divides the transmitted data over 256 equally spaced subchannels and keeps carriers orthogonal by construction: it builds the signal in the frequency domain mapping data to each carrier with the appropriate modulation, and then generates an OFDM symbol with the corresponding time-domain I/Q samples S_n with an Inverse Discrete Fourier Transform (IDFT), as shown in Fig. 1. This operation is repeated until the entire frame is transmitted: in the figure this corresponds to the IDFT block that takes as input each symbol (the light blue rectangle) as it move left to right, producing a train of OFDM symbols $s(t)$ after multiplication by the carrier that defines the Wi-Fi channel. To help the receiver to decode the signal, the first symbols carry constant (known) content and information about the encoding process. Figure 1 reports these symbols, showing also the first three legacy symbols (L-) that can be decoded by OFDM-PHY receivers located in the corresponding 20 MHz channels that build up the 80 MHz VHT channel. They are used for setting Automatic Gain Control (AGC) and estimating time and frequency offsets (Legacy Short Training Field, L-STF); for fine frequency tuning and for collecting CSI measures of the corresponding 20 MHz channel (Legacy Long Training Field, L-LTF); and for *understanding* the duration of the remaining part of the frame (Legacy Signal, L-SIG), so that legacy receivers can set the channel as busy until then. After the L- symbols there are the VHT headers: the VHT Signal A (VHT-SIG-A) carries information required to interpret VHT data (it is still transmitted as 20 MHz symbols); the VHT-STF is used to improve AGC estimation; the VHT-LTF is used to collect the 80 MHz CSI data; and the VHT-SIG-B is only used for MU-MIMO transmission, but it is always present. After the preambles DATA symbols are transmitted. The orange block is where we apply the randomization to protect users' privacy, and clearly in standard devices this block is not present.

At the receiver, operations are executed in the opposite order: I/Q samples are collected from the incoming signal and transposed in the frequency domain with a Discrete Fourier Transform (DFT). Information extracted from header symbols are used to recover the clock, reduce the carrier frequency offset (L-STF, L-LTF), and equalize the DATA symbols before decoding their content (L-LTF and VHT-LTF with the help of the extracted CSI). CSI estimation is fundamental to enable the high throughput of modern Wi-Fi allowing fast and precise equalization, but it can also be used to infer what happened to the signal during the propagation: i.e., it is possible to identify a precise *condition* of the environment, and based on this, for instance, to derive the position of a person in a room with a classification technique.

2.2. A NN-based Solution

Among the different classes of neural networks, Convolutional Neural Networks (CNNs) have been widely used in many applications thanks to their superior performance when dealing with pattern recognition tasks. Authors of [18] suggested that also a passive (device-free) localization system can benefit from the use of a CNN to learn the nonlinear relationship between the target's locations and the CSI fingerprints; to this end, they proposed a CNN-based localization system named PILC which outperforms other fingerprinting techniques. In this subsection we briefly present the main features of a localization system that was inspired by the work in [18]; further details about the design and the implementation of such system are found in [15].

Our localization system extracts features (I/Q or Amplitude and Phase) of the CSI and feeds them to a CNN that has been previously trained with labeled data. In general, this approach is characterized by two distinct phases: *i)* an offline stage (training) in which so-called *fingerprints* (data associated with specific locations) are collected and stored in a database, and *ii)* an online stage (operation) when the CSI samples (amplitude and phase or I/Q) are fed to the NN that returns the estimated location.

Before feeding the CSI measurements into the neural network, different pre-processing steps are necessary in order to extract meaningful features from the raw data. PILC developers used *CSI Tool* to extract raw CSI data from devices equipped with the Intel 5300 NIC. This tool allows for the extraction of only 30 OFDM subcarrier groups on 20 MHz Wi-Fi channels (about one value for every couple of subcarriers). In our work instead, we retrieve the CSI data from the device chipset using the *Nexmon CSI* tool [14], which gives access to all 256 OFDM subcarriers of 80 MHz Wi-Fi channels. The most important operations performed by our system are the following.

Selection of Subcarriers: Not all subcarriers are used for localization. As we are focusing on VHT-PHY transmissions, we discard all pilot subcarriers as they are used by the chipset for producing the CSI itself: should we receive a multi-stream transmission, the amplitude of the pilots would be random (as reported in [14]) so we decided to discard them. We also remove subcarriers that are not used for transmission according to the 802.11ac standard.

Normalization: Using both the amplitude and the phase information, there are several issues pointed out in [10]. The lack of synchronization regarding the central frequency leads to a Carrier-Frequency Offset (CFO), while the mismatch in the time-domain introduces a Sampling-Frequency Offset (SFO) generated by the Analog-to-Digital Converter (ADC). All these errors are caused by imperfections of the hardware. After applying further processing steps as detailed in [15], the network can be trained with the CSI measurements.

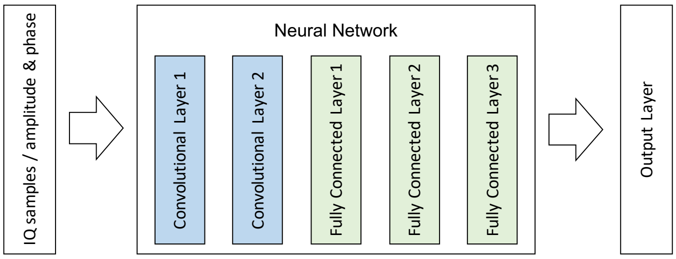


Figure 2: Architecture of the chosen NN

The network design, visualized in Fig. 2, is the following. First, we use two convolutional layers to extract meaningful features from the CSI data; the two layers build more complex descriptors that are essential for the localization process. Moreover, we exploit the spatial closeness of adjacent frequencies, as a convolutional layer considers multiple values at the same time. In cascade to the convolutional layers there are three fully-connected layers that combine the extracted features to obtain an estimate for the position of the target. In this work the output layer can take two different forms, depending on the localization task of interest: in one case, we consider the traditional approach in which the number of output neurons matches the number of target positions (each neuron outputs a likelihood for the corresponding position); on the other hand, we verify that with a sufficiently dense and regular grid, the output layer can consist on only two neurons whose output can be directly associated to Cartesian coordinates x-y. The output layer type must be selected *a* priori, which means that we are actually considering two different network architectures; therefore, the offline (or training) phase has to be performed separately for the two cases. When presenting our results, we will state clearly which of the two output layers we use in each experiment.

The number of layers, as well as the hidden neurons for each layer, were originally obtained heuristically with experiments starting with only one layer and increasing the number of layers as well as the number of hidden neurons until the network was able to describe the relationship between the CSI and the corresponding position of the user. The loss function uses the Euclidean distance to rate the performance of the network, as it provides an intuitive understanding of the error. We choose the common Rectified Linear Unit (ReLU) as activation function and the Adaptive Momentum Estimation (ADAM) [19] algorithm to adjust the weights based on the corresponding labels. When dealing with the classification task, the output layer is using the softmax activation function.

2.3. Location Learning Insight

The hardware used, the number of antennas at the transmitter and receiver and many other details may influence the precision of the localization; however, in this paper we want to focus on the elementary reasons that allow the localization and the minimal countermeasures that prevent localization. For this reason we limit the discussion to the use of one single TX-RX chain.

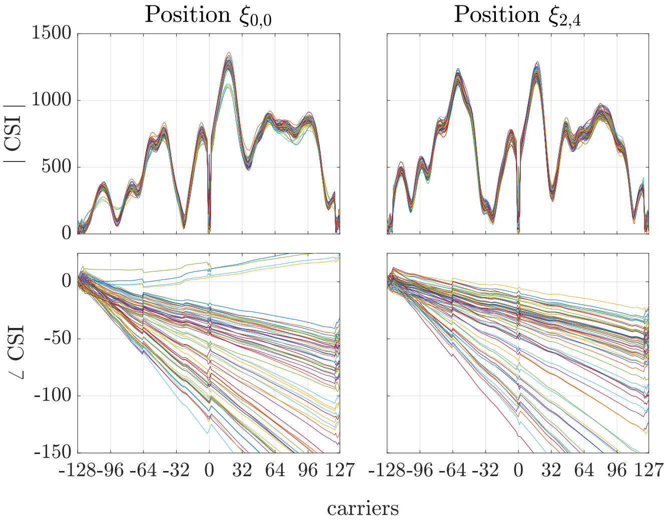


Figure 3: Plots of the amplitude (upper row, Broadcom 4365 802.11ac chipset units) and phase (lower row, radians) versus the carrier number with the person in two different training spots ξ (see Section 5).

Figure 3 reports the amplitude and phase of the OFDM signal collected on 70 packets at the receiver with a person in two different locations. The details of the experiments are described in Section 5; what is important for our discussion is that the NN is fed exactly with these quantities, so that whatever the NN learns it must be present here. It is clear that the characteristics are remarkably constant across different packets for the same position, so that learning is feasible. Interesting features that are visible are the peaks and notches in the amplitude, and the phase jumps; however, the notch around carrier 0 and the phase jumps seem to be independent from the position of the person (result confirmed at nearly all positions), making them ineffective for location estimation. The other peaks and notches, instead, have positions in the spectrum that clearly depend on the person's location, which is probably what the NN learns.

The NN ability to learn the position of a person from CSI information can be disrupted "manipulating" the CSI, so that it does not reflect exactly the electromagnetic fingerprint of the environment (thus revealing the position of the person in a deterministic way), but it also contains "deceit features" that confuse the learning and decision process of the NN and we conjecture also any other CSI-based positioning system. The next section is dedicated to discuss how we achieve this goal; we point out straight away, that when the randomization is active, the NN is trained anew, thus ensuring that the obfuscation is really due to CSI manipulation, and not to a bad experiment where the NN is trained without randomization, while testing is done with the randomization active.

3. CSI Randomization

To achieve location privacy, i.e., prevent a localization system to identify the position of a person, we can apply a time varying random distortion to the transmitted preambles in such a way that a receiver can still equalize the channel—i.e., not interrupting the communication—while any localization effort based on CSI characteristics (as discussed in Section 2) is invalidated.

With reference to Fig. 1, we decided to apply the appropriate distortion at the output of the IDFT block, right before the DAC and the front end, in the orange block that injects a "randomizer" modifier in the figure. This position may be sub-optimal w.r.t. other positions earlier in the transmission chain, but it makes the technique easily understandable and it can also be implemented outside chipsets, allowing the realization of specialized, privacy-preserving devices without the need to develop a new chipset from scratch. The introduction of location privacy protection in commercial devices can be done at different levels, from a pure software implementation as hinted above, to full integration at the physical layer, definitely a faster and more performing implementation. It can also be envisaged to develop a sublayer of the standard protocols empowering many functions, including a negotiation between transmitter and receiver, that would thus be facilitated in the frame decoding.

With "randomization" in our context we refer to a manipulation of the transmitted signal so that additional peaks, notches or phase jumps appear randomly in the CSI. The disturbance must obviously change over time, but preliminary evaluations show that the manipulation should neither change too quickly, e.g., appearing "white", as generated by a memoryless random process nor too slowly, e.g., like a constant distortion superimposed to the CSI. In fact, in both cases the NN would be able to easily filter out artifacts and identify again location-dependent features. While some theoretical work is required to devise randomization techniques whose effect cannot be learned at all by the NN, our goal for now is to propose a proof-of-concept for which also simple techniques introducing changes in the CSI periodically (with a fixed, empirically-defined period) can effectively work against the proposed localization framework. Furthermore, the disturbance must not be too distorting to avoid hampering the communication performance. The actual manipulation is different depending on the disturbance that we introduce as shown in Fig. 4, where $\mathrm{S}^{IN}(\cdot)$ is the signal at the output of the Inverse Fast Fourier Transform (IFFT), and mask$(\cdot)$ identify the type of distortion.

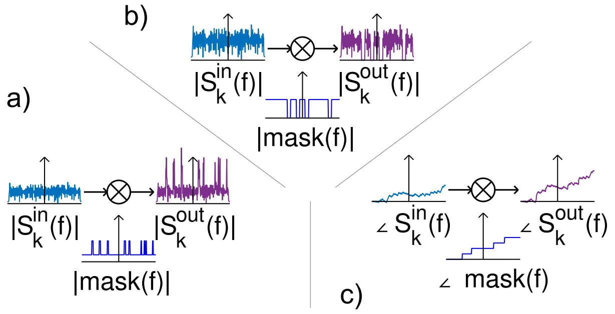


Figure 4: The tree different manipulations we experiment to randomize the position: **a)** adding random peaks in the channel response; **b)** introducing random notches; **c)** introducing random phase jumps.

We apply the disturbance by filtering the sequence of I/Q samples in the frequency domain. Since in this case the modulation format is fixed to VHT-PHY at 80 MHz with Long Guard Interval we can easily extract the samples forming each OFDM symbol and, knowing the structure of each symbol, we can: *i)* invert the encoding process, going back to the coefficients assigned to each of the 256 subcarriers; *ii)* multiply them by values of the filter corresponding to their frequency; and *iii)* recreate the "tampered" OFDM symbols.

We tested several different types of filters to explore the effect on localization given by basic filtering functions: introducing peaks, notches of phase jumps, where the number and positions of the peaks, notches and phase jumps are random functions. With reference to Fig. 4, we tested the randomization technique applying a random number of: a) peaks (between 5 and 10), each of uniform random width between 2 and 6 frequency samples; b) notches: similar to the previous case, but nulling instead of amplifying the selected points; c) phase jumps: all points after randomly selected positions accumulate a phase delay of $\pi/2$, with the number of positions selected randomly from 2 to 6. The number of phase jumps and peaks/notches is selected empirically to maintain the packet delivery ratio at acceptable levels; the theoretical analysis of the optimal randomized filtering is left for future investigation.

From a preliminary performance analysis it turned out that the effect of notches and phase jumps on the randomization of the estimated position is negligible, while interesting results were obtained for the filter with random peaks. Localization results for one sample point under different types of randomization are shown in Fig. 5. It is clear from this example that even if the localization precision is affected by applying random phase shifts to the CSI (Fig. 5b), an attacker can still make a sensible guess about the user's location. Only the introduction of randomly-placed peaks in the CSI (Fig. 5d) effectively degraded localization performance; therefore, we will use only this filter in the remaining of the paper.

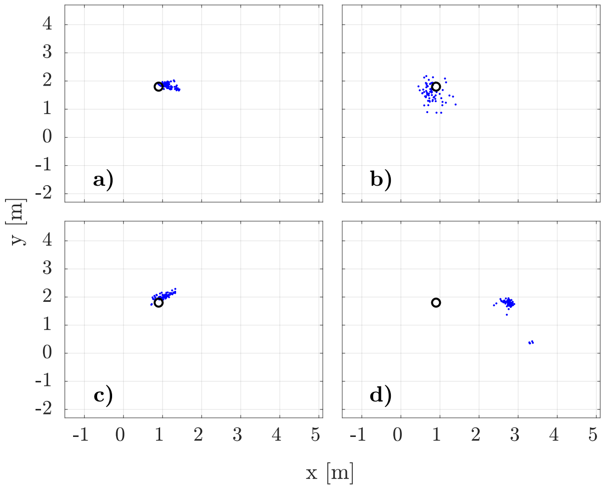


Figure 5: Localization results for one target point (black circle) under different conditions: a) without CSI modification; b) with selective phase shifting; c) with randomly-placed notch filters; d) with randomly-placed spikes.

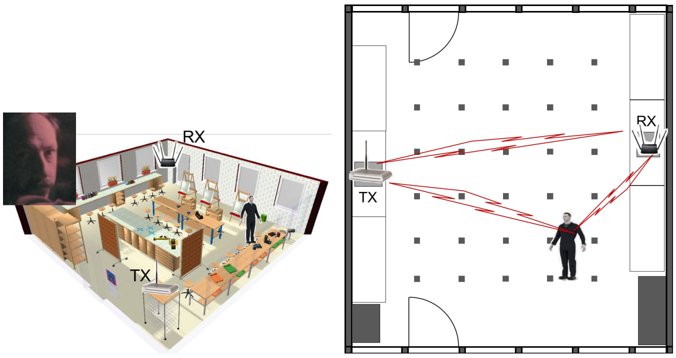
4. Location Privacy Violation

Tracking a person without her/his consent is illegal in many countries, and the location and whereabouts of a person may reveal a lot on her/his behavior and also on her/his activities. Even if localization is not extremely precise, as we discuss in the following, the position estimation may be used to control people at work, home, school. We first present the model of the attacker we consider in this paper and then three possible metrics that give different insights on the information disclosed by the position estimation.

4.1. Attacker Model

As depicted in Fig. 6 (a), an attacker wants to infer the location of a person in a room, e.g., an employee being kept under surveillance in a laboratory. We assume the presence of a common Wi-Fi AP providing Internet access in the laboratory. The attacker (e.g., the employer) has positioned a hidden Wi-Fi receiver—in our case a second AP, but in general any device capable of extracting CSI—in the laboratory and uses the NN-based localization system described before. In our specific setup, visualized in Fig. 6 (b), the receiver RX and the transmitter TX are on the opposite side of the room.

We make only the following assumptions regarding the attacker model: first, the attacker is able to train the localization system, which only requires collecting some measurements of reference positions; second, the attacker can only access the receiver and retrieve CSI from it. For instance, this attack can be easily replicated in hotels and multi-room environments as well as in private homes.



(a) Scenario

(b) Schematic setup

Figure 6: The location attack scenario (a) mapped to our laboratory setup (b).

4.2. Obfuscation Metrics

Measuring the performance of an obfuscation method, specially when its goal is protecting human privacy, can be complex. Indeed, a method may confuse the localization technique, diluting the precision, yet the attacker might still retrieve enough information to reveal important details on the person location, for instance if she/he is on one side or the other of the room. To gain the best possible insight in the performance of both the localization technique described in Section 2 and the obfuscation method introduced in Section 3, we use three different metrics, which are suited in different context and situations.

The first metric we consider is the classification accuracy:

$$A_c = \frac{N_c}{N_l} \tag{1}$$

where N_c is the number of samples correctly classified, and N_l the total number of samples taken. The overall accuracy is usually accompanied by other metrics, which help interpreting the results, such as the per-class accuracy, precision, and recall. When convenient, confusion matrices, i.e., how samples are assigned to classes, are very useful to present the results because they summarize all the accuracy metrics in a natural way. However, in the specific case of localization, this approach has some limitations. First, when the number of classes (target locations in our case) is large, confusion matrices become very large and difficult to read and understand. Moreover, they blur the sense of spatial proximity between different target points: not all localization errors are equally bad, since guessing a position adjacent to the correct one is not as wrong as guessing a position at the opposite side of the room. In Section 5.3 we introduce a concept of spatial proximity that holds for the particular case considered and helps the interpretation of the results.

The second metric is a Euclidean distance measure to verify and validate the methodologies under analysis. The classical mean square error of the distance is not appropriate for our goals. Recall that the NN can be trained as a classifier, but it can also be configured to output a position in a Cartesian space if it is trained on a regular grid with dense enough training spots ξ. However, the NN outputs a (x, y) position in a plane (2D), while a human body occupies a fairly vast space in 3D, so that it is indeed not possible to define the distance between the body and the (x, y) estimate. Call ρ a radius around the point estimate (x, y) of the NN, so that the circle of radius ρ and center (x, y) can be considered the estimated projection of the human body on the 2D plane. ρ is a parameter that indicate how accurate the location estimation is desired; if not otherwise stated, in this paper we can consider $\rho = 0.25$ m, which correspond to a high accuracy, since the projection of a human body is hardly smaller than a circle with such a radius. $\rho = 1.0$ m corresponds instead to a fairly loose requirement in localization precision.

Given the coordinates (x, y) of the estimate e as computed by the NN, and the coordinates (x_c, y_c) of the actual position p_c where the person stands, we construct a localization reliability $\mathcal{L}_{\text{R}}$ index as follows

$$\mathcal{L}_{\text{R}} = \frac{1}{N_l}\sum_{i=1}^{N_l} \mathcal{I}_d(i) \; ; \; \mathcal{I}_d(i) = \begin{cases} 1 & \text{if } d_i < \rho \\ 0.5 & \text{if } \rho \leq d_i < 2\rho \\ 0.25 & \text{if } 2\rho \leq d_i < 3\rho \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

where $d_i = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ is the Euclidean distance between the position estimate e_i and the coordinates of the position where the person is when the i-th sample is taken. Estimates must be taken when the person is standing still or moving slowly, so that enough estimates can be collected assuming the person rest in the same position p_c.

Clearly $\mathcal{L}_{\text{R}} \in [0, 1]$ and converges to one when all position estimates (one every received packet) are within ρ from the true position and converges to zero when all estimates are three times ρ from the true position. Averaging position estimates would not help an attacker for two reasons. First, the target can move, thus averaging estimated positions would simply result in estimating a sort of baricenter of the movements, and this also without the obfuscation. Second, while the obfuscator is on, ideally each position estimate is a random point in the room (tough with some memory to avoid filtering out the randomization), so that the result of averaging over many frames would always return as most likely position the center of the room, which is useless for the attacker.

Fig. 7 describes the metric, the reasoning is that an attacker is interested to understand where a person is with some level of accuracy described by ρ. Once the position is known almost surely (all samples lie within this accuracy), then the attacker is satisfied and $\mathcal{L}_{\text{R}} = 1$; while if the estimated position is not only always outside the circle with radius ρ, but it lies so far away that the two circles of radius ρ centered in the estimate $e = (x, y)$ and in the true position $p_c = (x_c, y_c)$ are separated by ρ, then the estimate is useless, hence $\mathcal{L}_{\text{R}} = 0$. The other two cases lie in between.

Indeed, also the true position of a person cannot be measured with absolute precision, but this issue is outside the scope of this paper, and we assume that when a person stand on (covers with his feet) a specific location (x_c, y_c), that is his exact position.

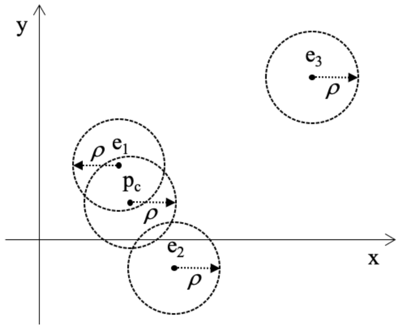


Figure 7: Examples of the $\mathcal{L}_{\text{R}}$ metric: for estimate $e_1 \rightarrow \mathcal{L}_{\text{R}} = 1$, for $e_2 \rightarrow \mathcal{L}_{\text{R}} = 0.5$ and for $e_3 \rightarrow \mathcal{L}_{\text{R}} = 0$, thus if the position is estimated from these tree samples $\mathcal{L}_{\text{R}} = \frac{1.5}{3} = 0.5$.

As a useful comparison to understand the reliability of localization we can use the metric in Eq. (2) assuming the location is simply a random point in the portion of the room where a person can reasonably stay, i.e., the room area minus the area where tables and furniture are. If we call this useful area in the room A_u, then the $\mathcal{L}_{\text{R}}$ for a random guess is

$$\mathcal{L}_{\text{R}}^{\text{rand}} = \min\left(1, \frac{15\pi\rho^2}{4A_u}\right) \qquad (3)$$

as a function of ρ, neglecting the border effect[1].

Finally, the third metric is focused on a privacy breaching scenario in which the attacker is not interested in determining the exact position of the victim, but rather in which area of the room it is located. This metric measures the capability of the system to localize a person with high reliability, but with relaxed precision. The subject of interest does not stand in a specific location, but stays, possibly moving slowly, in one of the four quarters (NW, North-East (NE), SE, South-West (SW)) of the lab. Equation (4) defines the accuracy as a-posteriori probability, where P_l is the empirical probability that the attacker successfully infer the position of the person in the lab quarter where he actually is, N_l is the number of position estimations collected, including those that infer the position in the shaded area in Fig. 8 that are considered "wrong" ($\mathcal{I}_l(i) = 0$), and $\mathcal{I}_l(i)$ is an indication function that tells if the i-th location estimation is correct ($\mathcal{I}_l(i) = 1$) or not ($\mathcal{I}_l(i) = 0$).

$$P_l = \frac{1}{N_l}\sum_{i=1}^{N_l} \mathcal{I}_l(i) \qquad (4)$$

It is immediate to notice that if the location samples i are taken at fixed intervals, i.e., sending packets at constant intervals, and P_l is close to one, then the time spent by the person in the different lab corners is revealed to the attacker.

[1] The border effect, i.e., areas with a positive weight in Eq. (2) that are outside A_u, is marginal for small ρ and underestimate the location reliability as ρ increases, because it counts as a valid position also portions of the lab that are outside A_u, thus $\mathcal{L}_{\text{R}}^{\text{rand}}$ is an actual lower bound for the location reliability, giving a good reference for the location reliability reduction provided by the randomization techniques we propose.

5. Setup and Results in Brescia

The initial experimentation is carried out in a Laboratory of the ANS[2] group at the University of Brescia. We describe the main characteristic of the testbed and some implementation details respectively in Section 5.1 and Section 5.2. Next, we present the results obtained using the CSI randomization technique against the NN localization described in Section 2.2: in Section 5.3 the NN has been trained to choose among a fixed set of target locations (classification task); the corresponding results obtained when the NN is trained to produce x-y estimates of the target position are presented in Section 5.4. Finally, a brief discussion about the impact of our system on the performance of Wi-Fi connections is carried out in Section 5.5.

5.1. Testbed Setup

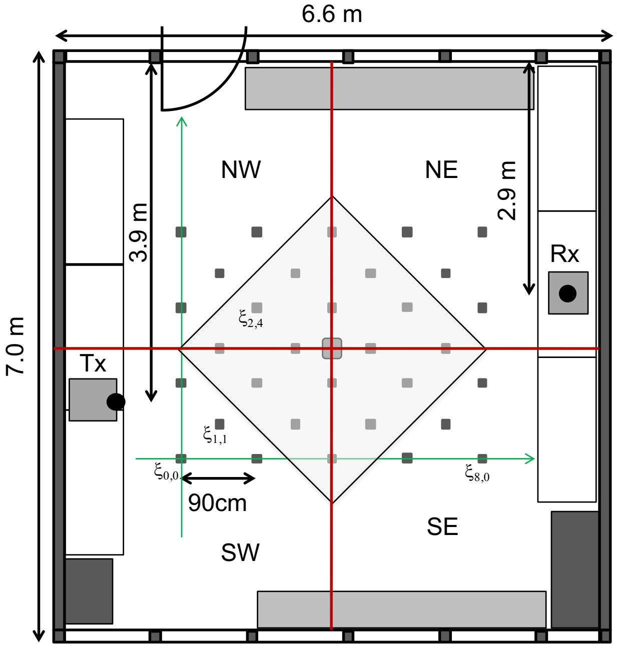


Figure 8: Layout of the experimental setup at the University of Brescia, the square dots on the floor are the training spots (ξ) for the neural network; the space is divided into four quadrants `SW`–`NE` for the sake of clarity.

Figure 8 reports a plan of the Laboratory with the position of the Tx and Rx devices and the places where a person stood to train the NN. The goal of the experiment is twofold: *i)* to validate the results presented in [15] to guarantee that they can be reproduced with an independent implementation; *ii)* to measure the actual capacity of the randomization technique presented in Section 3 to obfuscate the actual position of a person.

The Euclidean coordinate system origin is set on the SW corner of the grid as shown by the thin green axes; to make explanations easy we assume the lab is oriented with the north to the top. The training spots $\xi_{x,y}$ are numbered starting from the axis origin, so that $\xi_{0,0}$ is in the origin and $\xi_{8,6}$ is the one in the NE corner.

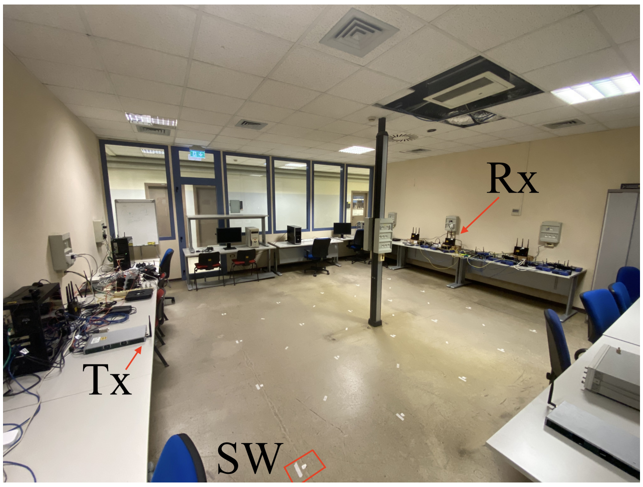


Figure 9: Photo of the laboratory described in Fig. 8, the pole in the middle with electrical outlets clearly creates a complex electromagnetic environment; the transmitter is labeled with '`Tx`' on the west side while the receiver used in the one in the middle on the east side of the lab. Some of the ξ spots are visible on the floor (white dots) with the coordinate origin one marked '`SW`'.

Besides classification and Euclidean distance, we also consider the scenario where the goal of the attacker is to identify which working areas (e.g., desk, workbench) are used by the staff at the laboratory, for instance to determine the fraction of the work time dedicated to different tasks, an act contrary to labor legislation in many countries. To this end, the map in Fig. 8 is divided into four sectors: NW, NE, SE, SW, as shown with the red thick lines, while Fig. 9 presents the actual setup of the laboratory. The shaded square of 2 m edge at the center of the room is not considered for the localization purposes, as it is an area where a person would not normally stay, but simply transit moving between the quadrants of the lab. As a side note, consider how simple it is to setup such an attack: the presence of an AP in a laboratory is very likely, a small sniffing device can be hidden easily, the training can be done when nobody else is present; given all this, then the attacker can very easily tell how much time the person spends in which part of the lab.

5.2. Implementation Details

The localization system is implemented on Commercial Off-The-Shelf (COTS) devices: in particular, we use the *ASUS RT-AC86U* router. For the transmitter side, instead, we use an Ettus N300 SDR radio and we use the MATLAB Wi-Fi toolbox to generate the frame and apply the randomization procedure. Note that no modification is required at the receiver.

At the transmitter, an infinite MATLAB loop generates a Wi-Fi frame at each iteration with random payload, and randomizes the corresponding raw signal as explained in Section 3. To this end, the software converts the packet into a vector containing I/Q samples according to the VHT-PHY modulation with one spatial stream and 80 MHz bandwidth. Then, it parses the vector and separates the VHT-PHY symbols. For each of them the software applies a specific procedure to isolate 256 I/Q samples

[2] The Advanced Networking Systems group `https://ans.unibs.it/` is one of the research groups in telecommunications at the University of Brescia

from the symbol and apply the DFT to get the OFDM coefficients assigned to each carrier. Once in the frequency domain, the software multiplies each coefficient of the OFDM spectrum by the value of the randomization filter at the same frequency. Finally, it generates a new sequence of 256 I/Q samples by inverting the OFDM spectrum applying a IDFT: it also recovers the structure of the symbol by either adding the GI or by completing the missing part taking into account the periodicity.

At receivers we replace the firmware that controls the Wi-Fi card with the nexmon-csi [20]. With this software we have access to all the 256 subcarriers of each TX-RX stream when using channels with 80 MHz bandwidth. The received packets are saved to a capture file together with the corresponding CSI data that are conveyed to user-space as UDP datagrams. In post-processing we extract CSI data from such packets and we feed the NN.

5.3. Localization Results with Position Fingerprints

As we described in Section 4.2, measuring the overall accuracy of a classification algorithms in localization tasks has many shortcomings. In Fig. 10 we define a concept of spatial proximity in terms of "near" locations that can be used to further distinguish the severity of classification errors when trying to localize the victim. Border points can have less than four neighboring locations. For example, according to the definition in Fig. 10, a point on the border of the grid has only two neighbors, and the four vertices of the measurement grid ($\xi_{0,0}$, $\xi_{0,6}$, $\xi_{8,6}$ and $\xi_{8,0}$) have only one neighbor.

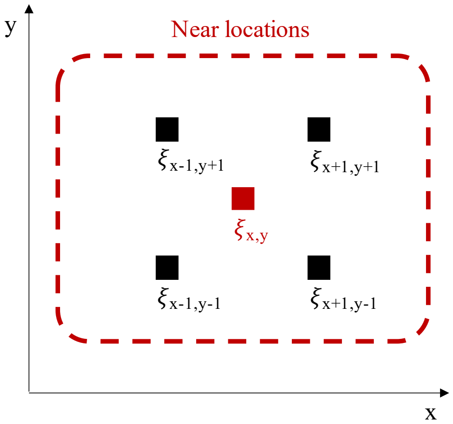


Figure 10: Visualization of the "near" locations for a generic point $\xi_{x,y}$ on the localization grid defined in Fig. 8. All the other points on the grid are considered "far" from the point $\xi_{x,y}$

Localization results with and without the application of our CSI randomization technique are summarized in Table 1. The average accuracy over the 32 target positions of the chosen localization system is 78%. When our randomization technique is applied, the accuracy drops below 5%, effectively disrupting CSI-based localization. Results are not uniform across target positions, but we deem this due to random effects rather than intrinsic properties of the positions. It is remarkable that randomization makes the localization procedure to classify position mostly far from the true one, thus not allowing even approximated estimates.

5.4. Localization Results with x-y Estimates

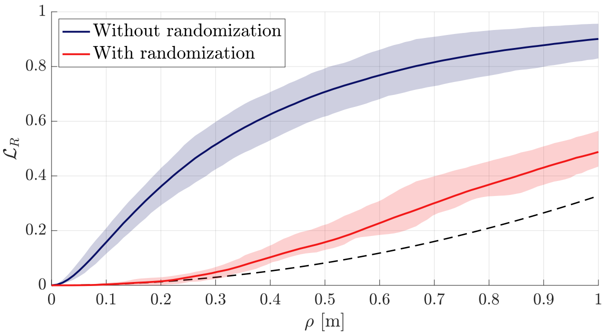


Figure 11: Localization performance according to metric $\mathcal{L}_R$ for ρ ranging from 0 to 1. Solid lines report the average result, while the shaded areas are the envelope of all measures including using different antennas and positions of the receiver. The dashed line represents the theoretical result for uniformly distributed random guesses.

The first insight in this case is focused on evaluating the localization capabilities of an attack using the metric described by Eq. (2) when the output layer produces an estimate in terms of a x-y coordinates. Once again we compare the performance obtained with and without the randomization procedure. In both cases, we train the NN with 700 packets for each one of the 32 positions highlighted in Fig. 8 and we test the localization on a different set of measures consisting of 150 packets per position collected at a different time. We capture CSI data from each of the four antennas available at the three receivers in the lab (visible in the picture of Fig. 9) for a total of 12 CSI feeds. Interestingly, results for the three receivers are similar and it also turns out that training the NN with data from one single antenna or any combination of the four antennas for each of the three receivers does not have any significant impact on the results. Figure 11 reports the average results obtained considering the metric $\mathcal{L}_R$. The solid line is the average for the 32 positions computed by considering all the CSI (average over 12 antennas). We also show the shaded regions between the worst and best performing antenna, obtained again by averaging over the 32 positions. The localization reliability increases with ρ, but the most interesting cases for in-room localization are the ones with small ρ, i.e., values between 0.3 and 0.6. In particular, for $\rho = 0.3$, the average $\mathcal{L}_R$ score is above 0.5 for the localization system, but drops below 0.05 when CSI randomization is active. The benefit of using our randomization system is evident from the fact that the curves obtained using randomized CSI are much closer to the black dashed line corresponding to the calculus for uniformly distributed random guesses.

With respect to the last metric defined in Section 4.2, which represents the probability of an attacker locating the victim at least in the correct area of the lab, we subdivided the lab into four quadrants and we measured the performance of our system by comparing the value of P_l when the randomization is active or not and when the user is moving outside the shaded area in Fig. 8. Without CSI randomization, the NN predicts positions

Table 1: Classification accuracy over the 32 target positions, with and without applying our CSI randomization technique. Classification errors for each position are divided between "near" locations (as defined in Fig. 10) and "far" locations (all the remaining points).

	Without CSI randomization															
	$\xi_{0,0}$	$\xi_{2,0}$	$\xi_{4,0}$	$\xi_{6,0}$	$\xi_{8,0}$	$\xi_{1,1}$	$\xi_{3,1}$	$\xi_{5,1}$	$\xi_{7,1}$	$\xi_{0,2}$	$\xi_{2,2}$	$\xi_{4,2}$	$\xi_{6,2}$	$\xi_{8,2}$	$\xi_{1,3}$	$\xi_{3,3}$
Correct	100	92.9	5.7	90.0	0.0	100	97.1	100	98.6	58.6	100	60.0	57.1	95.7	100	75.7
Near	0.0	0.0	1.4	0.0	0.0	0.0	2.9	0.0	0.0	0.0	0.0	0.0	42.9	4.3	0.0	0.0
Far	0.0	7.1	92.9	10.0	100	0.0	0.0	0.0	1.4	41.4	0.0	40.0	0.0	0.0	0.0	24.3
	$\xi_{5,3}$	$\xi_{7,3}$	$\xi_{0,4}$	$\xi_{2,4}$	$\xi_{4,4}$	$\xi_{6,4}$	$\xi_{8,4}$	$\xi_{1,5}$	$\xi_{3,5}$	$\xi_{5,5}$	$\xi_{7,5}$	$\xi_{0,6}$	$\xi_{2,6}$	$\xi_{4,6}$	$\xi_{6,6}$	$\xi_{8,6}$
Correct	92.9	100	22.8	94.3	100	100	25.7	100	100	51.4	55.7	70.0	88.6	100	98.6	78.6
Near	1.4	0.0	52.9	1.4	0.0	0.0	2.9	0.0	0.0	7.1	0.0	0.0	11.4	0.0	0.0	0.0
Far	5.7	0.0	24.3	4.2	0.0	0.0	71.4	0.0	0.0	4.3	44.3	0.0	0.0	0.0	1.4	21.4
	With CSI randomization															
	$\xi_{0,0}$	$\xi_{2,0}$	$\xi_{4,0}$	$\xi_{6,0}$	$\xi_{8,0}$	$\xi_{1,1}$	$\xi_{3,1}$	$\xi_{5,1}$	$\xi_{7,1}$	$\xi_{0,2}$	$\xi_{2,2}$	$\xi_{4,2}$	$\xi_{6,2}$	$\xi_{8,2}$	$\xi_{1,3}$	$\xi_{3,3}$
Correct	0.0	0.0	100	0.0	0.0	0.0	0.0	0.0	0.0	2.9	0.0	0.0	0.0	0.0	40.0	0.0
Near	0.0	0.0	0.0	0.0	0.0	0.0	47.1	77.1	0.0	0.0	0.0	0.0	0.0	85.7	0.0	0.0
Far	100	100	0.0	100	100	100	52.9	22.9	100	87.1	100	8.6	100	14.3	60.0	100
	$\xi_{5,3}$	$\xi_{7,3}$	$\xi_{0,4}$	$\xi_{2,4}$	$\xi_{4,4}$	$\xi_{6,4}$	$\xi_{8,4}$	$\xi_{1,5}$	$\xi_{3,5}$	$\xi_{5,5}$	$\xi_{7,5}$	$\xi_{0,6}$	$\xi_{2,6}$	$\xi_{4,6}$	$\xi_{6,6}$	$\xi_{8,6}$
Correct	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Near	22.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	48.6	0.0	0.0
Far	77.1	100	100	100	100	100	100	100	100	100	100	100	100	51.4	100	100

that fall in the correct quadrant with probability 0.66. Despite the result appearing quite low, from Fig. 12 we can clearly identify clusters of points in the correct quadrant that would help the attacker making a more sensible guess. However, we can see from the same figure that this analysis is not useful when randomization is active: in this case, in fact, the probability that the estimated position falls in the right quarter of the lab drops to 0.30, just slightly above the random guess of 0.25.

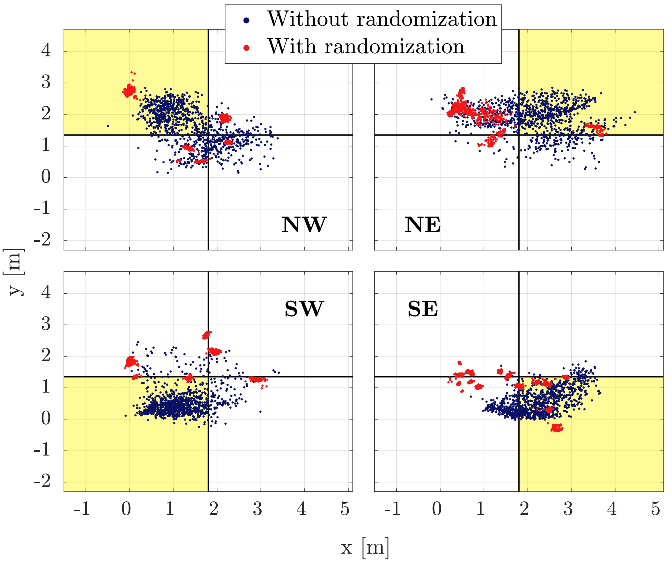


Figure 12: Positioning results considering different quarters of the lab. Each dot represents the position estimated by the attacker for each packet received. We report the estimates performed with and without CSI randomization using orange and purple dots respectively.

To better assess the performance of the system with and without randomization, we ran a second experiment where we collected CSI data when the user was sitting at four different desks located at each corner of the lab as indicated in Fig. 13. In this experiment the user can slightly move, i.e., by rotating over the chair vertical axis, or by moving arms and hands on the desk. The only constraint is to stay within the circles reported in the figure. It is clear that while without randomization the NN predicts the positions with very high accuracy (they almost always fall within their circle), when randomization is applied the predicted position is almost always wrong.

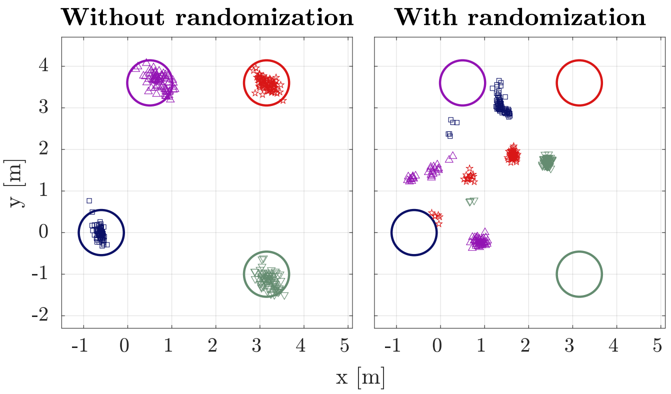


Figure 13: Localization results when the user is working at four different desks placed in the corners of the lab and not moving.

5.5. *Impact on Performance*

So far we have discussed only CSI-based localization and the possibility of obfuscating the location information with a random manipulation of the OFDM symbols prior to transmission. In the experiments, we transmitted with the lowest-order modulation and coding scheme (MCS) (i.e., `MCS0`), which uses BPSK. However, it is important to investigate the communication performance for higher-order MCSs because they are more susceptible to channel errors. We hence computed the

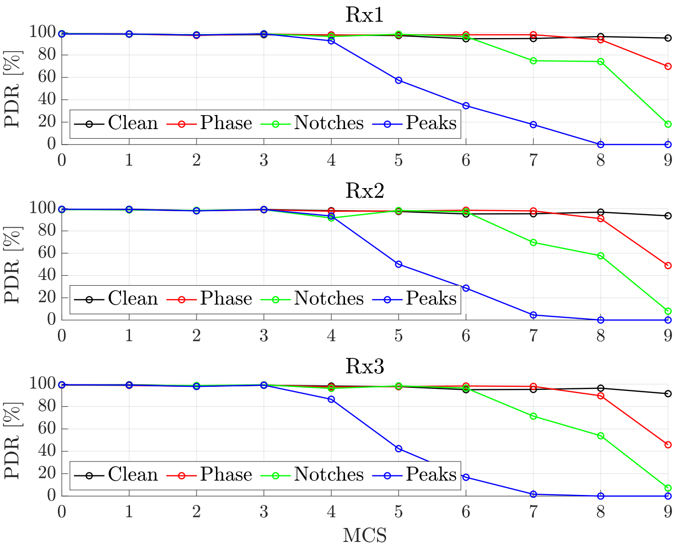


Figure 14: Impact of different CSI randomization techniques on the PDR at the three receivers (they are visible in Fig. 9, the one used for localization at the center indicated with the red arrow and the other two on the left and right). The PDR is plotted as a function of the MCS, which directly relates to the throughput.

Packet Delivery Ratio for all VHT-PHY MCS transmitted with 80 MHz bandwidth and a single spatial stream: we report in Fig. 14 the PDR for the three receivers when randomization is off (Clean) and on (Phase, Notches, Peaks). For completeness, we measure the impact of phase- and notch- based manipulation even if they have not been used for obfuscation.

It appears from the figure that the three receivers display very good performance for all MCS without randomization. As easily predictable, only robust MCSs retain acceptable Packet Delivery Rate (PDR) when the randomizing filter is applied. In particular, when the modulation used is sensitive to distortion (i.e., 64- and 256-QAM modulations) the systematic errors introduced by the filter prevent correct decoding of the frame at one receiver (Rx3) and stops reception at another one (Rx1). This further deteriorates when increasing the MCS: MCS8 and MCS9 cannot be received at almost any position when the distortion is based on peaks. Instead, when the manipulation is based on phase of notches, it seems that the performance is more easily retained. These are however only preliminary results to show feasibility and not a full analysis of achievable performance.

Let us analyze the reasons and discuss how this problem can be addressed, as it is clear that a localization obfuscation method should not disrupt communication capabilities. Eq. (5) describes the mathematical model of the signal at the receiver, where $S(F)$ is the signal spectrum at the output of the IDFT of the 802.11ac transmitter, $M(f)$ is the filtering function used for obfuscation (the 'mask' in Fig. 4), and $H(f)$ is the channel response, including attenuation, distortion and multipath fading.

$$R(f) = S(f) \times M(f) \times H(f) \tag{5}$$

Equation (5) is an equivalent model of the system we propose, but it is not how we actually perform the obfuscation, as it is done in the digital domain rather than in the analog one as described by Eq. (5). $R(f)$ is converted back in the digital domain and passed through the dynamic equalizer driven by the CSI information before it is fed to the digital receiver implementing demodulation and error correction. From Eq. (5) it is clear that to preserve the communication performance $M(f) \times H(f)$ should maintain the properties of an equivalent, physically realizable and admissible (for 802.11ac) channel response $H'(f)$. The theoretical analysis of the properties of a randomizing filter that preserve this property and also obfuscate location is part of the future work of our research teams. The manipulation should be energy-agnostic to avoid changing the transmission power of the frame, while it should increase the uncertainty of the CSI, as localization is based on the stability and determinism of the channel distortion.

6. Setup and Results in Ghent

This part of the measurement campaign was enabled by the w-iLab.2 testbed hosted by IMEC[3] near Ghent, Belgium. In this testbed, due to the particular configuration of the environment described in Section 6.1, we have run experiments with the NN that performs only classification-based localization. Furthermore, it is impossible in this lab to select localization points with full freedom, thus the selected points must be considered as a typical real world situation, where a person in a lab can stand only in positions that are determined by the lab layout itself. Whether such restrictions can help the attacker to circumvent our obfuscation method requires further investigation. The implementation details are briefly presented in Section 6.2. Despite the great difference with respect to the laboratory in Brescia, we obtain results confirming the effectiveness of the obfuscation system also in this case, as described in Section 6.3.

6.1. Testbed Setup

The entire facility measures approximately 55 x 18 m; however, we use only a portion of the available area to run experiments, as shown in Fig. 15. The testbed has almost no external radio interference, but represents a very complex environment from an electromagnetic standpoint: the ventilation pipes and many metal obstacles offer very little line-of-sight between different locations of the room, while generating at the same time a lot of multipath components.

In the testbed we have access to an FPGA running OpenWiFi [21], a full-stack implementation of the IEEE 802.11 standard based on SDR. Moreover, the w-iLab.2 testbed is provided with a dozen mobile nodes ("robots") that can be driven around the testbed and programmed to automate complex operations.

In this testbed, we have run localization experiments using both robot and human targets. Due to travel limitation at the time of the experiments, the results with a human target are unfortunately limited.

[3] IMEC is a joint research center and our patron in the ORCA project, see `https://doc.ilabt.imec.be/ilabt/wilab/index.html` for further details on the lab.

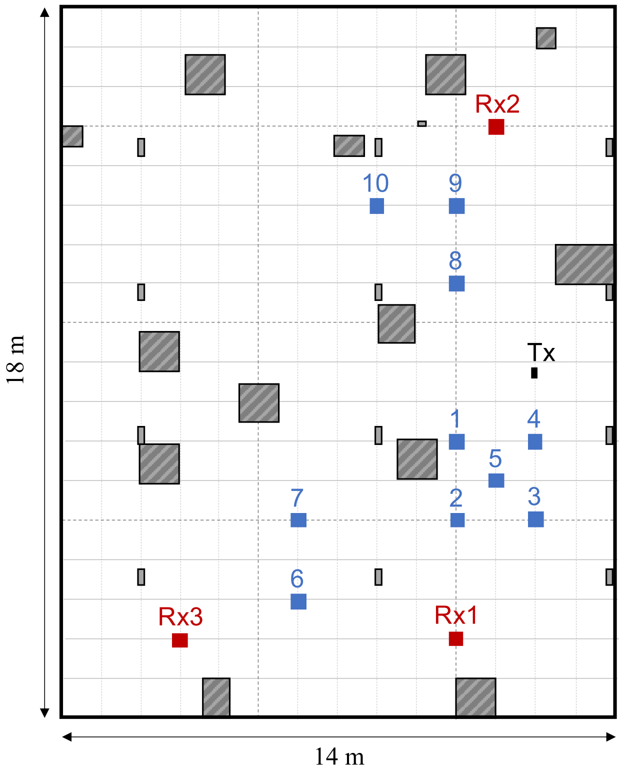


Figure 15: Map of the w-iLab.2 testbed. Blue and red squares identify target and receiver locations respectively; gray squares are obstacles preventing line-of-sight.

6.2. *Implementation Details*

The localization system is implemented on Nexus 6P smartphones attached to the robots available in the testbed. The transmitter is a Xilinx ZC706 board, configured to boot a GNU/Linux operating system and to run a modified version of OpenWiFi implementing the IEEE 802.11 protocol stack and providing support for our CSI randomization operations. This specific version of OpenWiFi allows the manipulation of the CSI field of transmitted packets on a per-packet basis, similar to what was shown in Section 3. The details of the implementation are a little different in this case, but the result is the same: we can change the relative amplitude of each transmitted subcarrier. To reproduce the same randomization effect given by peak insertion on a single frame, we can scale to 1/7 of the original amplitude all the subcarriers except the ones corresponding to the peaks (the value 1/7 is actually fixed by the specific implementation and is not our choice). Phase information cannot be arbitrarily edited in this setup, but we have already motivated in Section 3 the choice to proceed with CSI amplitude randomization only. We create an `iperf` session between the OpenWiFi node and another wireless node in the testbed, using a set of scripts that also take care of changing the CSI profile periodically. However, OpenWiFi currently supports 20 MHz channels only, so that we have limited bandwidth in this scenario with respect to the setup we have used in Section 5 and only 64 OFDM subcarriers.

Figure 16 shows the CSI collected when the target robot is in two different positions. We notice that despite the two positions being very far apart (see map in Fig. 15) the small size of the robot is not enough to cause variations in the CSI that can be appreciated by the human eye. The presence of two "groups" of CSI amplitudes profiles is an artifact due to the specific Automatic Gain Control (AGC) algorithm used at the receiver, which is designed to scale the amplitude of the received signal to match the available range of values and only has a few quantized amplification factors.

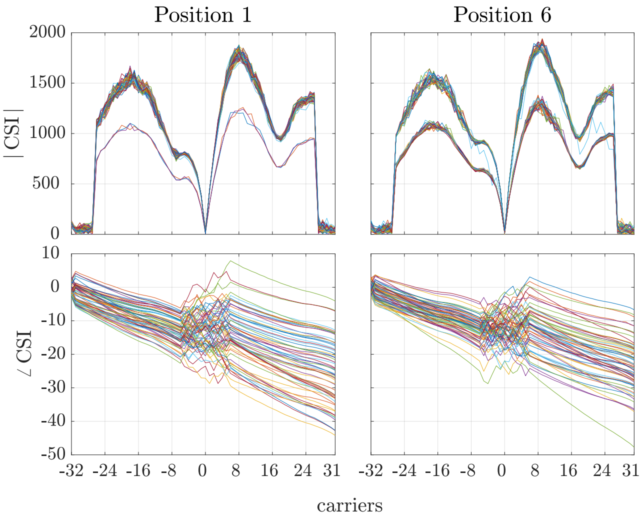


Figure 16: Plots of the amplitude (upper row, Broadcom 4358 chipset units) and phase (lower row, radians) versus the carrier number with the **robot** target in two different spots (see also Fig. 15).

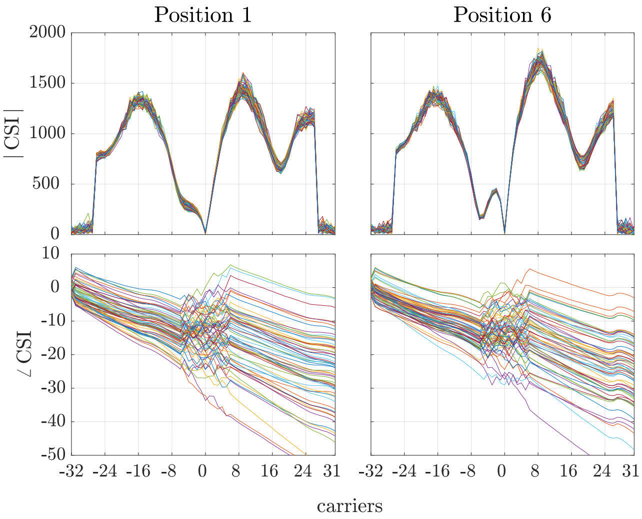


Figure 17: Plots of the amplitude (upper row, Broadcom 4358 chipset units) and phase (lower row, radians) versus the carrier number with the **human** target in two different spots (see also Fig. 15).

Figure 17 presents the comparison between the CSI collected when a human target is standing in the same two positions. While it is possible to better discriminate the two CSI in this case, the profile of the CSI amplitude is overall very similar. This suggests that there exist some environments in which this type of localization technique can encounter problems. In fact,

Table 2: Classification accuracy in the w-iLab.2 for the considered scenarios.

Target, Scenario	Rx1 %	Rx2 %	Rx3 %
Human, clean CSI	69.0	57.7	45.6
Human, randomized CSI	18.9	5.9	9.6
Robot, clean CSI	26.9	34.4	29.3
Robot, randomized CSI	12.3	10.7	8.1

we have verified that the accuracy of the localization system is lower than in the other testbed in Brescia, most probably due to the presence of many reflectors that affects the CSI much more than a robot or a human body.

6.3. Localization Results with Position Fingerprints

Given the complexity of the environment and the foreseen impairment of the NN based localization technique, it is interesting to investigate if randomization of the CSI still has the same impact. The answer is positive and it is summarized in Table 2 that reports the overall classification accuracy as defined in Eq. (1). Independently from the accuracy of the NN with clean CSI, which is particularly low for robots as they have a small electromagnetic footprint, when randomization is applied the localization accuracy falls around the value of a random guess (there are 10 possible positions, so 10% is a random guess), with the exception of Rx1 with the human target where the accuracy is close to 19%.

Figure 18 reports the confusion matrices obtained at receiver Rx1 with the human target, with and without applying CSI randomization, and helps getting some insight in how the environment influences the entire problem. First of all, it is clear that P1 is so close to a reflecting surface that a target there is irrelevant for a CSI point of view for Rx1, to the point that none of the samples are classified correctly even without the randomizer. Next, the apparent better performance of the classifier in this case is indeed biased by positions P4 and P10, that have a much higher accuracy than average, however, this is hardly a claim of victory for an attacker, since the improved accuracy requires that the victim stands in the positions that yield good results.

Indeed, the analysis of these, and others, confusion matrices, indicates that the obfuscation technique tends to force the NN to make wrong decisions, but these are not uniformly spread on all possible position, rather, they are grouped in specific locations. We believe that this is due to the time-correlation structure of the random sequences generated by the obfuscator. Overall the gain, i.e., the reduction of estimate precision lies between a factor 2 and a factor 10, with this latter indicating indeed perfect obfuscation, but specific positions may have different performance. The ideal obfuscator should spread all estimates uniformly on every possible position, while it is clear that even if wrong, the estimates cluster in other specific positions (see the result in Section 5 too), thus suggesting that there may be some information left that a better localization system may exploit.

7. Discussion and Future Work

New technologies cannot come at the price of reducing people's rights. High performance Wi-Fi communications, which use advanced signal processing techniques to compensate the distortions of the electromagnetic environment, enable tracking the location of people, even if they do not carry any Wi-Fi device with them.

In this paper we have proposed a novel technique that, introducing carefully crafted random distortion of the Wi-Fi signal spectrum at the transmitter, prevents inferring the position of a person in a room exploiting the CSI at the receiver. We have shown with a real implementation that the technique is feasible and works as intended. At the same time it does not destroy communications as, for instance, jamming would do, and this is fundamental to have location-obfuscation techniques widely adopted.

We think that this methodology can be widely adopted, even finding its way into future standards, so that citizens can use Wi-Fi without even bothering that their precise location can be tracked by an adversary. The path to achieve this goal, however, is still quite long, and it includes having a full, formal understanding of the signal manipulations that allow achieving location obfuscation without hampering communications *at all*; extensive experimental campaigns to verify that the technique works with different localization methodologies and different channel bandwidths; analysis of more sophisticated attacks based on the joint analysis of many MIMO channels, and so forth. We would also like to investigate if this same methodology can be applied to obfuscate the position of a device, rather than the position of a person who does not hold or wear a communication device. Finally, we remark that this paper addressed passive attacks, where the attacker controls only a receiver, but exploits the normal Wi-Fi traffic. In this case, the only useful traffic for the attacker comes from transmitters that are perfectly fixed and whose position is well known and stable, so that the NN can be trained in advance, thus the obfuscator needs to be installed only in APs or similar 'infrastructure' devices. Active attacks, where the attacker controls both the transmitter and the receiver are another very interesting research area, where, however, privacy protection cannot be based on randomization at the transmitter.

Acknowledgements

We would like to thank our patron in IMEC and in particular Vincent Sercu for the time he dedicated to the experiments, standing patiently in the designates spots in w-iLab.2 for training and testing purposes. This work has been partially funded at the University of Brescia by the European Commission under the Horizon 2020 Orchestration and Reconfiguration Control Architecture – ORCA project (grant no. 732174) Open Call 3 "Experimental analysis of CSI based anti-sensing techniques - CSI-MURDER" experiment; at the University of Darmstadt in the context of the DFG-funded project SenShield (Project ID: 447586980).

Target Position

Predicted Position	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	
P1	0	0	0	0	0	0	0	0	0	0	
P2	11	70	0	0	0	0	0	0	21	1	
P3	20	0	70	0	0	0	9	0	0	0	
P4	3	0	0	70	0	0	17	0	0	0	
P5	0	0	0	0	58	0	0	0	0	0	
P6	5	0	0	0	12	45	3	0	0	11	
P7	1	0	0	0	0	0	10	0	0	0	
P8	5	0	0	0	0	0	0	54	1	0	
P9	25	0	0	0	0	0	16	16	48	0	
P10	0	0	0	0	0	25	15	0	0	58	
%	0.0	100	100	100	82.9	64.3	14.3	77.1	68.6	82.9	**69.0**

(a) Without CSI randomization.

Target Position

Predicted Position	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	
P1	0	0	0	0	0	0	0	0	0	0	
P2	0	2	0	2	0	1	0	0	0	0	
P3	0	0	0	0	0	0	0	0	27	0	
P4	0	0	0	68	0	0	0	0	0	0	
P5	20	0	17	0	0	0	2	0	0	0	
P6	0	18	1	0	0	3	0	13	0	0	
P7	0	0	0	0	0	0	10	57	0	0	
P8	0	37	0	0	0	21	3	0	0	22	
P9	0	2	52	0	70	45	34	0	1	0	
P10	50	11	0	0	0	0	21	0	42	48	
%	0.0	2.9	0.0	97.1	0.0	4.3	14.3	0.0	1.4	68.6	**18.9**

(b) With CSI randomization.

Figure 18: Confusion matrices for receiver Rx1 and a human target. The datasets used for computing these results contain 70 CSI for each target position (700 CSI in total). Each cell of the matrix contain the number of CSI. The bottom row indicates per-class accuracy; the number in bold face in the corner indicates the overall accuracy.

References

[1] Cominelli, Marco and Kosterhon, Felix and Gringoli, Francesco and Lo Cigno, Renato and Asadi, Arash, An Experimental Study of CSI Management to Preserve Location Privacy, in: 14th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH), London, UK, 2020, pp. 1–8.

[2] G.-S. Wu, P.-H. Tseng, A Deep Neural Network-Based Indoor Positioning Method using Channel State Information, in: Int. Conf. on Computing, Networking and Communications (ICNC), IEEE, Maui, HI, USA, Mar. 2018, 2018, pp. 290–294.

[3] T. F. Sanam, H. Godrich, An Improved CSI Based Device Free Indoor Localization Using Machine Learning Based Classification Approach, in: 26th European Signal Processing Conf. (EUSIPCO), IEEE, Rome, Italy, Sept. 2018, 2018, pp. 2390–2394.

[4] F. Adib, D. Katabi, See through walls with WiFi!, in: Conf. of the Special Interest Group on Data Communication (SIGCOMM), ACM, Hong Kong, Aug. 2013, 2013, pp. 75–86.

[5] Y. Qiao, O. Zhang, W. Zhou, K. Srinivasan, A. Arora, PhyCloak: Obfuscating Sensing from Communication Signals, in: 13th Conf. on Networked Systems Design and Implementation, USENIX Association, Santa Clara, CA, USA, Mar. 2016, 2016, p. 685–699.

[6] Abanto-Leon, Luis F. and Bäuml, Andreas and Sim, Gek Hong (Allyson) and Hollick, Matthias and Asadi, Arash, Stay Connected, Leave no Trace: Enhancing Security and Privacy in WiFi via Obfuscating Radiometric Fingerprints, Proc. ACM Meas. Anal. Comput. Syst. 4 (2020) 44:1–44:31.

[7] X. Guo, N. Ansari, F. Hu, Y. Shao, N. Elikplim, L. Li, A Survey on Fusion-Based Indoor Positioning, Comm. Surveys & Tutorials 22 (1) (2020) 566–593.

[8] Z. Yang, Z. Zhou, Y. Liu, From RSSI to CSI: Indoor Localization via Channel Response, ACM Comput. Surv. 46 (2) (2013) 25:1–25:32.

[9] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, L. Ni, CSI-Based Indoor Localization, Trans. Parallel Distrib. Syst. 24 (7) (2013) 1300–1309.

[10] X. Wang, L. Gao, S. Mao, CSI Phase Fingerprinting for Indoor Localization with a Deep Learning Approach, Internet of Things Journal 3 (6) (2016) 1113–1123.

[11] Schmidt, Erik and Inupakutika, Devasena and Mundlamuri, Rahul and Akopian, David, Sdr-fi: Deep-learning-based indoor positioning via software-defined radio, IEEE Access 7 (2019) 145784–145797.

[12] M. Abbas, M. Elhamshary, H. Rizk, M. Torki, M. Youssef, WiDeep: WiFi-based Accurate and Robust Indoor Localization System using Deep Learning, in: Int. Conf. on Pervasive Computing and Communications (PerCom), IEEE, Kyoto, Japan, Mar. 2019, 2019.

[13] M. Schulz, F. Gringoli, J. Link, M. Hollick, Shadow Wi-Fi: Teaching smartphones to transmit raw signals and to extract channel state information to implement practical covert channels over Wi-Fi, in: Int. Conf. on Mobile Systems, Applications, and Services (MobiSys'18), ACM, Munich, Germany, June 2018, 2018, pp. 256–268.

[14] F. Gringoli, M. Schulz, J. Link, M. Hollick, Free your CSI: A channel state information extraction platform for modern Wi-Fi chipsets, in: 13th Int. Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH '19), ACM, Los Cabos, Mexico, Oct. 2019, 2019, pp. 21–28.

[15] F. Kosterhon, Device-Free Indoor Localization: A User-Privacy Perspective, Master's thesis, Technische Universität Darmstadt, Secure Mobile Networking Lab, Department of Computer Science (April 2020). doi:https://www.doi.org/10.13140/RG.2.2.25468.56965.

[16] I. S. for Information technology, 802.11-2016 - 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).

[17] R. Prasad, OFDM for Wireless Communications Systems, Artech House, London, UK, 2004.

[18] C. Cai, L. Deng, M. Zheng, S. Li, Pilc: Passive indoor localization based on convolutional neural networks, in: 2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS), IEEE, Wuhan, China, 2018.

[19] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization (2014). arXiv:1412.6980.

[20] M. Schulz, D. Wegemer, M. Hollick, Nexmon: The C-based Firmware Patching Framework (May 2017).
URL `https://nexmon.org`

[21] X. Jiao, W. Liu, M. Mehari, M. Aslam, I. Moerman, openwifi: a free and open-source ieee802.11 sdr implementation on soc, in: 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020, pp. 1–2. doi:10.1109/VTC2020-Spring48590.2020.9128614.