Orchestration and Reconfiguration Control Architecture

Experimental analysis of CSI based anti-sensing techniques

CSI-MURDER

Revision: v.1.0

Call identifier	ORCA-OC3-EXP-EXC
Date of report	14/08/2020
Organization	University of Brescia (Department of Information Engineering – DII)
Submission date	14/08/2020
Authors	Francesco Gringoli, Marco Cominelli, Renato Lo Cigno
Coordinator[1]	Francesco Gringoli
Authors	Francesco Gringoli, Marco Cominelli, Renato Lo Cigno
Patron	imec

[1] Coordinator is the main (technical and administrative) contact person for this experiment or extension.

SUMMARY

One of the hottest topics related to advanced IEEE 802.11 (Wi-Fi) PHY layers is the ability to perform environment sensing by exploiting auxiliary information present in the Wi-Fi signals, **which ultimately can lead to unauthorized surveillance**. Recent publications show that the Channel State Information (CSI) embedded in Wi-Fi frames carries enough information about the propagation environment to enable fine localization of people, making it possible to control people movement and activity.

The goal of the CSI-MURDER project is

1. **to evaluate the severity of this threat, and**
2. **to propose novel countermeasures to prevent unauthorized surveillance by leveraging the environment and the functionalities offered by the ORCA testbed.**

An extensive experimental campaign has been performed in two different surveillance scenarios, both reasonable in real-life scenarios: in the first case (passive attack), an attacker exploits the CSI transmitted by standard devices (e.g., an 802.11 AP into a room) and only controls a receiver where he collects the CSI data for further processing a device; in the second one (active attack), the attacker actively transmits packets to locate the victim. In both cases the victim does not need to carry a device, but she/he is localized just because the CSI information is modified in a unique way as a function of where the victim is standing. It is clear that the **unauthorized surveillance unveiled by these two scenarios is particularly distressing**, as it seems that the **victim has no way to protect herself,** but to renounce to Wi-Fi communications, indeed, in the second scenario the victim should even actively jam the electromagnetic spectrum, because the attacker transmit his own packets to collect the relative information.

CSI-MURDER has, for the first time, devised and shown that countermeasures can be taken and they are effective, and indeed that at least for the case of passive attacks these countermeasures can be embedded in standard devices, so as to prevent even the possibility of attack.

The countermeasure for the passive attack is based on actively modifying the CSI information at the transmitter side, so that every packet still resembles a valid packet transmitted in a realistic environment, but it is forged introducing random characteristics that prevent learning the electromagnetic "fingerprint" that identifies the victim location. The countermeasure for the active attack is instead more complex, because it entails a very fast reaction of a device that intercept all traffic and inject appropriate random disturbance synchronous with the packet preamble. Details change, but the disturbance is crafted in such a way to have the same effect as the modification of the preamble at the transmitter side in case of passive attacks.

Results of all the experiments done show that even in a challenging environment it is possible to perform unauthorized localization and determine the position of a person up to a certain extent, and we can imagine that as technology evolves, these localization systems will become more precise and cheaper. **These same results show that the countermeasures devised by the CSI-MURDER team are very effective,** and pave the road for further research and development to improve the methodology, find theoretical results that support the heuristics so far implemented, and include these methodologies in devices and communication systems that include privacy by design and whose privacy-preserving properties are based on sound theoretical results.

The methodology developed on the ORCA testbed has been reproduced in a local setting to evaluate the significance of the environment in performing such measurements. In all the considered scenarios, the proposed countermeasure has been proven effective in **neutralizing unauthorized surveillance while preserving the communication between the nodes of the wireless network.**

TABLE OF CONTENTS

1 TECHNICAL CONTRIBUTION

1.1 Concept and objectives

CSI-MURDER takes the move from one of the most flourishing research topics in wireless networks: the ability to exploit communication signals to sense the surrounding environment and hence to localize people and possibly objects [Adib2013, Ma2019, Sanam2018, Wu2018]. However, the objective of the Experiment is not to propose yet another localization system, **but to explore the possibility of preventing unauthorized localization without hampering the communication performance**. The experiment focuses on the latest IEEE 802.11 (Wi-Fi) PHY layers because localization research based on Wi-Fi is thriving, laying the ground for novel—more accurate—localization techniques, but also raising serious questions about users' privacy, with emphasis on unauthorized surveillance.

Wi-Fi frames embed a field in their header that enables the derivation of the Channel State Information (CSI), which is a compact description of the properties of the communication channel. From a communication perspective, the CSI is an estimation of the current channel condition that allows to adapt transmission and reception (the equalizer in particular) parameters accordingly, greatly enhancing performance. Recently, researchers have studied innovative ways to collect, process and analyse time series of CSI measurements to understand how wireless signals propagate in the environment, revealing features of the physical structure: for instance, measuring how CSI amplitude changes over time it is possible to detect human presence or motion, recognize different types of activity, even detect specific gestures or measure human breathing, albeit these fine-grained detection has proven hard to replicate. Research focuses mostly on sensing and localization, forgetting the privacy implications that they pose; in particular, it seems that almost nobody is tackling the problem of neutralizing un-authorized surveillance while preserving the communication performance.

CSI-MURDER is thus one of the first efforts to **understand if "obfuscation" techniques can be found that render localization inaccurate or unreliable, thus preserving users' privacy**. The methodology is strictly experimental, aiming at presenting a feasibility proof and collect insight to tackle the theoretical analysis in the future. The experiment has two objectives defined as follows (taken from the proposal):

- **Anti-sensing technique A against passive sensing (Objective 1).** In this case, attackers exploit the transmissions originating from victim's devices. As victims have full control of their (at home) hardware, they can configure it to artificially shape the preamble of the transmitted signals so that they are still decodable at the intended receiver (e.g., victim's smart tv), but the CSI extracted by attacker's sniffer will appear completely random (i.e., keeps changing over time).
- **Anti-sensing technique B against active sensing (Objective 2)**. In this other case attackers use their own transmitting equipment and study how CSI of frames that they transmit change over time. To circumvent sensing, the victim must deploy a system that detects transmissions coming from the attacker and react by over-transmitting signals that can modify the CSI observed by the attacker in a random fashion, but without jamming the packets that can still be received.

These two objectives require two different solutions, but the anti-sensing techniques developed in the two cases stem from the very same premise: the ability to tamper with the CSI in order to make them appear "random" at a receiver owned by a malicious user and prevent further analysis.

To study the problem in an effective manner and to design a solid countermeasure, we leverage the inner flexibility of the ORCA testbed. In particular, we rely mostly on the following features:

- A full-stack SDR-based implementation of the IEEE 802.11 protocol compatible with Linux;
- An automated framework for repeating measurements over long time spans with high accuracy.

Given the complexity of the experiments and their cutting-edge nature, CSI-MURDER also exploits a laboratory at the University of Brescia and equipment available in Brescia (compatible with equipment available at **w.iLab.t**) to prepare the experiments in ORCA, but also to validate the results obtained in **w.iLab.t** with additional data that provides further insight, especially in light of publishing the results in high-impact venues, where replicability and validation are fundamental requirements.

1.1.1 Localization Obfuscation Concepts

Localization methodologies are based on the analysis of the CSI, thus any such methodology, and in particular those based on Machine Learning, Neural Networks and so forth must be based on the information present in the CSI itself. Figure 1.1.1 reports the amplitude and phase of the OFDM signal collected on 70 packets at the receiver with a person in two different locations. The NN of the localization system 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.

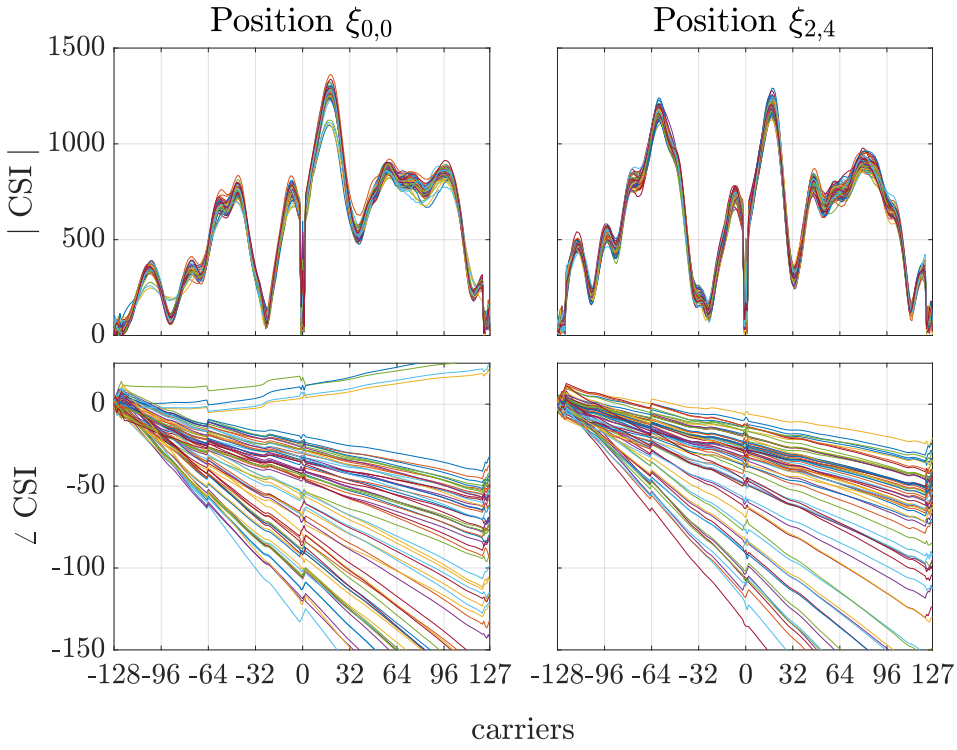


Figure 1.1.1: Amplitude and phase of the CSI for several packets collected with a person in two different positions in a laboratory

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 by nearly all positions), so that they are probably useless in position 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.

We can try to disrupt the NN ability to learn the position of a person from CSI information by "fiddling around" with 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.

As it is explained in detail in Sections 1.2.1.2 and 1.2.1.3 describing the implementation, early experiments have shown that the peaks are the features that, once modified, obfuscate the location more efficiently, thus the obfuscation systems we have developed for this Experiment, both in case of passive attacks and of active attacks, are based on the modification of these peaks.

1.2 Technical results and lessons learned

The results of the Experiments are extremely promising, **showing that it is possible to obfuscate the location while maintaining the communications.** At the same time, our initial implementation, has the clear disadvantage to carry the signature of the obfuscation, meaning that it is immediately evident if a frame preamble has been tampered with, so that an attacked could potentially study counter-countermeasures.

It must however be considered that the challenges we faced in the implementation of the obfuscation system and in the deployment of the experiments are daunting, and in general they could not be foreseen in the design phase, being implementation and environment related. Thus, a proof of feasibility, even if clearly identifiable by an external observer is a clear success. In the following sections we describe the work carried out in the project, with special emphasis on the most promising results and on how we overcame the difficulties and achieved the intended technical results thanks, also, to the ORCA environment.

1.2.1 Localization and Obfuscation Implementation

The obfuscation system we devised (see Section 1.2.1.2) should in principle work with any localization system, because it operates directly on the signal, ideally making it impossible to derive proper information on the location of a person simply because the position-dependent information carried by the original signal is destroyed by tampering with the signal at the transmitter. Thus, we should be able to pick any existing localization implementation and use it. Unfortunately, most of the localization system proposed are not endowed with a public implementation of the system, thus we had to also implement a credible localization system. Our implementation is derived from [Cai2018, Kosterhon2020] and is described in Section 1.2.1.1. Finally, the selective jammer implementation required to achieve Objective 2 is described in Section 1.2.1.3.

All the implementations described here, together with instructions on how to use the software, pointers to the GitHub open repositories for download and extension and so forth can be found on the Experiment web site at https://ans.unibs.it/projects/csi-murder

For the sake of readability we keep the description compact without any unessential detail, and we refer the reader interested in details to the web page of the Experiment.

1.2.1.1 Localization system

The main idea at the core of CSI-based localization is the assumption that a distinctive *electromagnetic fingerprint* can be associated to one specific configuration of the environment. The relative position of the transceivers, the geometry of the room, the type and the position of obstacles—and ultimately the location of a person—all affect the way waves are propagating in the environment and consequently the CSI measured. No accurate and implemented analytical solution is known to this problem, and most of the proposals available in the literature (e.g., [Abbas2019, Sanam2018, Wu2018]) are based on Machine Learning (ML) and Artificial Intelligence (AI) technologies.

Neural Networks (NN) and Deep Learning (DL) have proven to be powerful and viable solutions, even though they need a training phase. In particular, Convolutional Neural Networks (CNN) have been effective to tackle the CSI-based localization problem. Many architectures have been proposed for this scope, but they are all equivalent from the perspective of our Experiment as there are no theoretical

reasons for one architecture to be more robust than others in face of obfuscation. Our implementation inspired by the one presented in [Cai2018] is an extension of the one developed in [Kosterhon2020].

The architecture of the CNN we implemented is shown in Figure 1.2.1. The software can work with any IEEE 801.11 channelization (10, 20, 40, 80, and 160 MHz); clearly the larger the bandwidth the more accurate is the localization and the more computational power is needed to run it.

The description that follows refers to 20 MHz channels because the hardware available in **w.iLab.t** does not allow handling larger channels. The localization system comprises two parts: the CSI extraction and the CNN. The CSI extraction works on-line at the Nexus 6P mobile phone receiver that embeds a Broadcom chipset for which our group previously developed a CSI extraction tool based on the nexmon firmware [Schulz2017]. The CNN part works off-line and is implemented using TensorFlow, a well-known framework developed by Google widely adopted by the research community. The localization system that we used at our lab is similar, i.e., still based on the same CSI extraction tool and NN, but works on 80MHz channels: instead of Nexus 6P as receivers we used Asus RT-AC86U Access Points that embed a chipset similar to that of the phone.

The input of the network consists of an array of 52 complex values obtained by removing from the normalized CSI the unused subcarriers; the output of the network is a choice among a set of N classes which corresponds to the estimated location of the person. The first two layers of the network are convolutional layers of size 30 and 50 respectively, while the kernels in both layers have 5 taps. The output of the last convolutional layer is concatenated into a 1D vector that is then processed through three fully-connected layers with different dimensions. The last layer has size N, equal to the number of classes that the network is asked to discriminate. The activation function of all the layers but the last is the common Rectified Linear Unit (ReLU) function. The last layer of the network uses a softmax activation function for selecting the most probable class (i.e., the person's location).

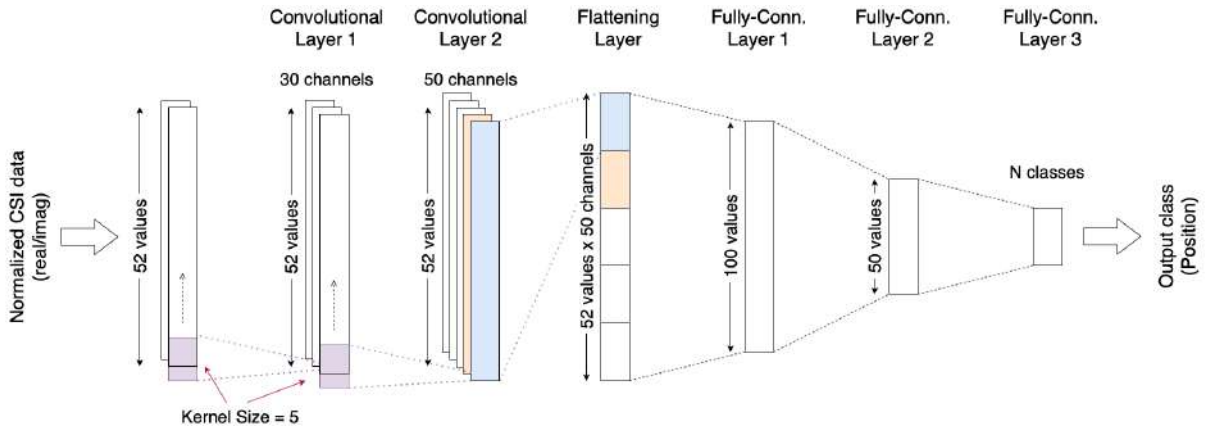


Figure 1.2.1: Architecture of the Neural Network used for localization.

1.2.1.2 Obfuscation system (Objective 1)

During the project, we used two different implementations of the obfuscation system. The first version, which we developed in Matlab, was used for understanding which type of modification worked better. Frames generated and obfuscated by Matlab were then transmitted using a SDR radio and finally received on nodes running our CSI extraction tool: from there we were feeding the NN and analysing the performance of the obfuscation. The second system was implemented by the patron inside the OpenWiFi stack that is available on **w.iLab.t** testbed. After having understood which obfuscation system to use, we then customised the OpenWiFi transmitter for adopting the same approach.

In both cases, we were manipulating (tampering) the signal so that it looks like it has been transmitted in a channel with a response different from the actual one; we call this "*fake channel response*" for the sake of simplicity. The fake response is applied to the baseband IQ samples at the transmitter as shown in Figure 1.2.2. On a 20 MHz channel, each OFDM symbol comprises 64 subcarriers, as shown in Figure 1.2.3, equally spaced by 312.5 kHz. Not all the subcarriers are used to transmit data: 11 are used as guard-bands, 4 as pilots and also the subcarrier at 0 is not used. On an 80 MHz channel, things are slightly more complex for the case of single spatial streams, but the principle is the same: each OFDM symbol comprises of 256 subcarriers, again spaced by 312.5 kHz. Similarly to the 20 MHz case, 11 subcarriers are used as guard-bands, 8 are pilots and 3 subcarriers around 0 are suppressed. In addition, as we show in Figure 1.2.2, the first part of the physical preamble of an 80 MHz frame is built as if there are four neighbouring 20 MHz frames: this is fundamental to let the 80 MHz transmissions coexist with all those nodes that are configured as 20 MHz only, behind the considered 80 MHz spectral extension. Thanks to this approach, each 20 MHz node can detect the start of an 80 MHz transmission.

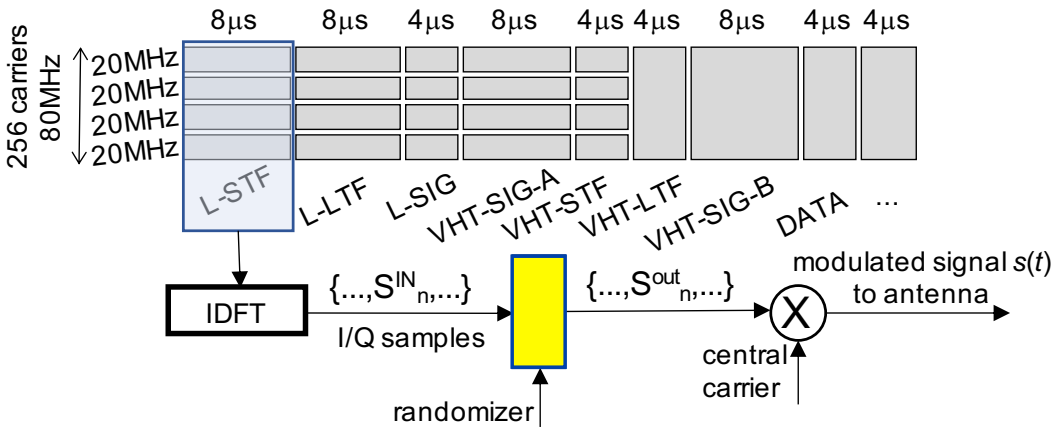


Figure 1.2.2: Format of an 80 MHz OFDM frame with the obfuscation processing in the lower part: initial symbols are known and are used to infer the CSI at the receiver; specifically, a proper randomization is injected through the yellow block after the IDFT mimicking a fake channel response.

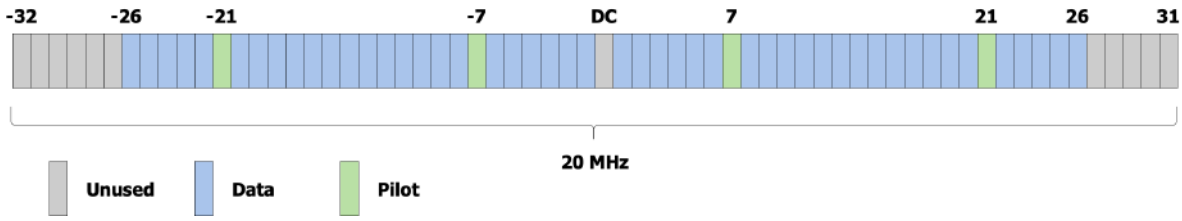


Figure 1.2.3: Wi-Fi OFDM uses 52 of the available 64 subcarriers to carry information the other 12 are used as guard-bands and pilots.

The fake channel response must be properly randomized, so that localization systems cannot derive sufficient information to infer anything meaningful about the position of a person (or an object) in a room or laboratory. So far, we have not addressed the problem of localization obfuscation in open air, but to the best of our knowledge, no CSI-based (nor RSSI-based) passive localization system works outdoors.

The fake channel response can be seen as a random distortion to the transmitted CSI, so that a receiver can still equalize the channel (i.e., not destroying the communication) while localization efforts based on CSI characteristics are hampered.

The position of the obfuscation block in the transmitter depends on the version. For instance, in our Matlab implementation we decided to apply the appropriate distortion at the output of the IDFT block, right before the Digital to Analog Converter (DAC) represented by the SDR transmitter, in the yellow block where randomization is injected in Figure 1.2.2. 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 could also be implemented outside chipsets in a real device, allowing the realization of specialized, privacy-preserving devices without the need to develop a new chipset from scratch. The fake channel response consists in the manipulation of the transmitted signal so that additional peaks, notches or phase jumps appear randomly in the CSI. This is done by taking the I/Q samples, moving to the frequency domain by running a DFT, applying an obfuscation complex "mask" on each carrier, and finally going back to the time domain (with an IDFT transformation) where for each symbol recompute the Cyclic Prefix. In the OpenWiFi stack the details of the implementation were decided by the patron but, as we will see next, the spectral mask can be assigned by using specific configuration functions of the OpenWiFi stack.

This disturbance must not be "white," looking like a memoryless random process, because otherwise the CNN would very simply filter it out, selecting only the location-dependent features. The optimization of the process that changes the fake channel response is outside the scope of this Experiment.

As mentioned before, the fake channel response can modify the peaks, notches or phase jumps of the signal, thus it can be seen as a filtering function where a proper mask is multiplied with the signal to be transmitted. Figure 1.2.4 shows the three possibilities, where S(·) is the Wi-Fi signal generated by the PHY before IDFT and mask(f) is the mask that we apply to achieve the obfuscation.

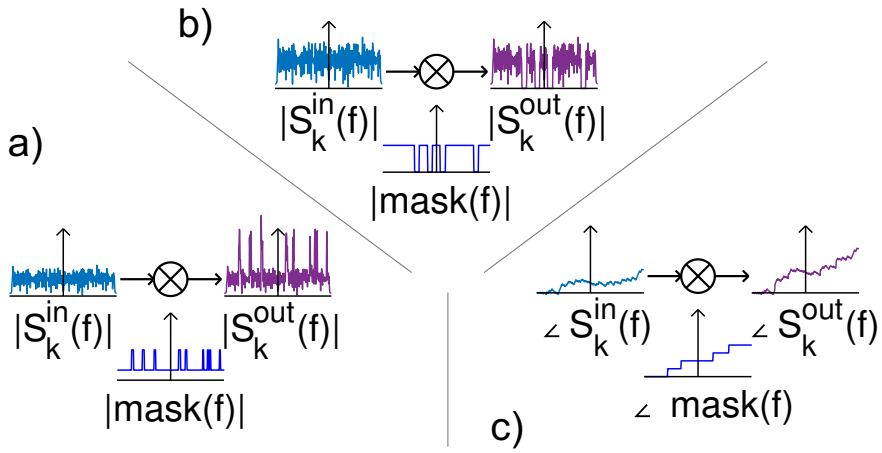


Figure 1.2.4: The tree different manipulations experimented to randomize the position: a) adding random peaks; b) introducing random notches; c) introducing random phase jumps.

A first question arises on the most appropriate manipulation to obfuscate location information. To solve this issue, we ran a preliminary experimental campaign in the laboratory at the University of Brescia, to avoid wasting precious experiment time at **w.iLab.t.** Figure 1.2.5 shows one exemplary outcome of these experiments, where we qualitatively compare how effective it is to obfuscate the location of a person standing in a specific position in the laboratory (the black circle in the figure). It is clear that adding random spike destroys the capability of the localization system to properly locate the person, while adding notches of phase jumps is almost ineffective. This was verified in many different experiments, with different persons and also in different rooms. Also adding peaks, notches and phase jumps all together does not improve on the case of adding peaks only. Thus, we decided to implement and use a mask that adds random peaks in the CSI, changing their pattern periodically.

The obfuscation system works by assigning one scalar value to each subcarrier (also to the unused ones, for ease of implementation). Each value indicates a scale factor that reduces the amplitude of the corresponding subcarrier, i.e., the set of all the 64 values forms a mask representing the profile of the fake channel response. Since each value is coded with 3 bits, scale factors are discretized over 8 levels as indicated in Table 1.2.1. The same type of mask can be applied in both versions, i.e., our own developed in Matlab and the OpenWiFi stack.

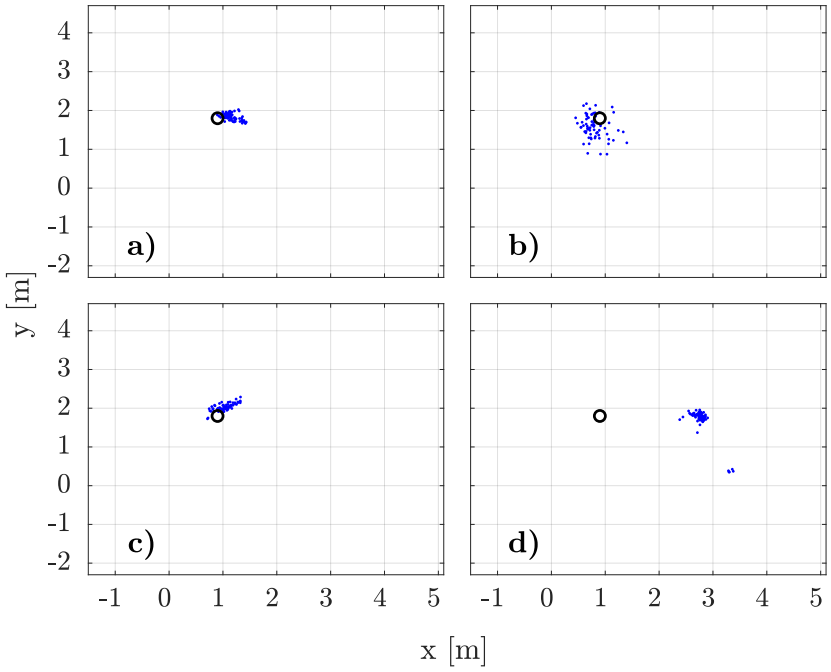


Figure 1.2.5: Preliminary results with a single localization position showing that introducing peaks is sufficient to obfuscate the location, while notches and phase jumps seems to have a lesser impact. a) without CSI modification; b) with selective phase shifting; c) with randomly-placed notch filters; d) with randomly-placed spikes.

Table 1.2.1: Mapping between values and scale factors.

Value	Scale Factor
000	0/7 = 0.00
001	1/7 = 0.14
010	2/7 = 0.29
011	3/7 = 0.42
100	4/7 = 0.57
101	5/7 = 0.71
110	6/7 = 0.86
111	7/7 = 1.00

For the sake of completeness, we have implemented also the filter with notches. Thus, our code can realize:

- fake channels with randomly placed notches: a random number of subcarriers is selected (ranging from 3 to 6 according to a uniform distribution) and their scale factor is set to 0, while all the other channels are assigned scale factor 1;

- fake channels with randomly placed peaks: a random number of subcarriers is selected (ranging from 3 to 6 according to a uniform distribution) and their scale factor is set to 1, while all the other channels are assigned scale factor 1/7—this corresponds to amplify the selected subcarriers by a factor of 7;

although only the second version has been actually used during the Experiments.

The following code snippet shows how carriers are selected for applying notches/peaks. It is important to notice that in the implementation of the obfuscation system, OFDM subcarriers are numbered from 0 to 63, with the DC component starting at index 0 and the negative subcarriers being mapped at indices from 32 to 63.

```
[…]
const int nCarriers = 64;
const int nPoints = floor(3 + 4 * drand48()); // number between 3 and 6

/* Initialize mask to all 1's when applying peaks
   or initialize it to all 7's when applying notches */
unsigned int mask[nCarriers];
for (int carrier = 0; carrier < nCarriers; carrier++) {
    mask[carrier] = 1;
}

for (int k = 0; k < nPoints; k++) {
    int carrier = floor(1 + 52 * drand48());
    if (carrier > 26) carrier += 11;
    mask[carrier] = 7; // or set it to 0 if applying notches
}
[…]
```

The purpose of the obfuscation system is to show a fake channel response which is varying over time, so we denote by *T* the time spent using one given configuration. The two limit cases are obtained for *T* going to infinity, in which the fake channel response never changes and obviously becomes quite useless for our scope, and for *T* going to 0, in which the fake channel response changes for every transmitted packet. The main drawback of this latter approach is that if we let the mask change too fast, the randomly placed peaks/notches can be treated as noise by the neural network and get easily discarded in the classification problem. This insight shows that the problem of selecting the optimal *T* is crucial for the effectiveness of the obfuscation system, but it is also deeply connected to the scenario considered, e.g., how much time does the training phase takes or how many different locations are considered. In general, an attacker setting up an unauthorized localization system to monitor a victim should not be able to recover the value of *T*, hence *T* should be made random.

As we mentioned already the optimal process that drive *T* is outside the scope of this project, and we run experiments either with fixed *T* or drawing *T* from a truncated exponential distribution with mean μ = 2.5 s and maximum 2μ. The following code snippet shows how *T* is selected by our obfuscation system:

```
[…]
const double avg = 2.5;
double x, y;

do {
    x = drand48();
    y = -avg * log(1-x); // y is draw from a single-side exp. distr.
} while (y > 2 * avg);

unsigned int T = (unsigned int) (y * 1000000); // T is in us
[…]
```

It is clear that the obfuscation we devised, although simple, is not tailored or designed specifically for the CNN technique we have implemented for localization. Albeit we did not have the resources to implement other localization techniques, we are convinced that the obfuscation works with any CSI-based localization system, and probably also with RSSI-based ones, thus the results obtained are general in nature. We are also aware that by knowing the obfuscation methodology, it is probably possible to devise a localization technique that circumvent it, but this Experiment is a proof of concept and a feasibility study that pave the road for further research whose goal should be to find an optimal obfuscation system that preserve privacy by design without hampering communication capabilities.

1.2.1.3 Obfuscation Against Active Attacks (Objective 2)

When the attack is **active, i.e., both receiver and the transmitter are controlled by the attacker, obfuscation through filtering is not possible** (the legitimate user does not have access to the transmitter), thus the only possible countermeasure is to tamper with the packets of the attacker. We do not want, as done e.g., in [Gezici2016], to jam all the traffic, because this would destroy communications, but to interfere in such a way that both the physical header from which the CSI is extracted and the following symbols are modified in the same way: this would make the entire frame content decodable at receiver but it would also alter the CSI.

Before describing the implementation we devised, let us recall that **this technique has never been implemented before, and that our goal is to provide a proof of concept,** without having the possibility of implementing the system in hardware or FPGA, which would allow a real-time reaction to signals, so that some tricks are due for the implementation. The main idea is to keep the mechanism as simple as possible to make it anyways easy to implement in an FPGA without having to use complex techniques such as full-duplex blocks. In principle, the FPGA should just listen to the channel and decide when to start transmitting the interfering signal, after switching its radio from reception to transmission. In our proof of concept, we hence skip the “reactive” part, where we are aware that synchronization may be non-trivial, and we just focus on the interfering signal: we use two SDR radios for transmitting the attacking signal and the interfering signal at the same time, synchronising them by properly generating the corresponding I/Q samples on the controlling host, as we show in Figure 1.2.6. It is clear that in this configuration synchronization is ideal, and also the position of the two transmitting antennas is very close. The study of the impact of partial de-synchronization and how to compensate for the different distance of the antennas from the receiver is very interesting and part of future work given the success of this initial proof of concept.

Figure 1.2.6: Reference scenario for demonstrating the active attack.

Since we do not use a full-duplex system, the content of the interfering signal cannot be based on any information from the attacking signal: we craft it by generating simple sinusoidal tones at the same frequencies of the OFDM sub-carriers composing the Wi-Fi signal. The number of tones and their frequencies is decided in a very similar way as in the “Obfuscation System” of Objective 1 explained above. To test the effectiveness of the technique, we first simulated it with Matlab (we provide the code in Appendix B). Since this is a preliminary idea, we did not try to model analytically the phenomena behind the proposed technique yet, and we consider the Matlab simulation enough to prove its validity at the moment. Figure 1.2.7 shows the CSI of a 20 MHz signal that propagates through

multiple randomly generated AWGN channels with same SNR. In this test the energy of the Wi-Fi signal is four times that of the interfering signal, that is composed of 7 neighbouring carriers.

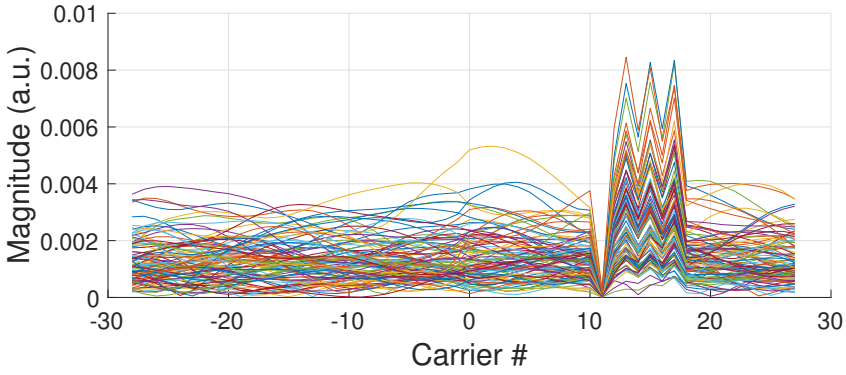


Figure 1.2.7: Matlab simulation of the proposed active attack.

Still the effect on the collected CSI is extremely clear, and in line with the principle of the passive technique. We only plot the CSI of signals that were correctly decoded by Matlab. For the sake of clarity, the interfering signal extends only from the central sample of the L-STF symbol and the VHT-SIG-B symbol. This is extremely important as 1) it gives the possibility to react at the reception of the L-STF, and 2) it does not need to know the length of the frame, as it ends before the payload starts.

We then implemented the SDR-based system by sending to a B210 radio the sequence of I/Q samples representing the Wi-Fi frame that we generated with Matlab. At the same time, we are sending to the other B210 radio a different sequence of samples, generated with Matlab by summing together a number sinusoidal tones with random frequencies, selected using the same approach as in the passive technique. The pattern of the sinusoidal tones in the obfuscating signals is also changed every *T*, as discussed in the previous sections. In the **w.iLab.2** testbed, where a single B210 radio was up and running, we used the two chains of a single B210 radio, while for the testbed in Brescia we were using two separate B210 radios synchronised with an external clock.

While we did not port the active system to OpenWiFi, this work should be relatively easy: the receiver should be modified in order to react at the first detection of the L-STF by switching from RX to TX and transmitting the interfering sequence of fixed length. As the signal is composed of sinusoidal tones of known frequency (yet randomly selected from a pool of possible tones), they can be precomputed so that the transmitter should only sum the precomputed tone samples and avoid computing them for each transmitted frame.

1.2.2 Experiments setup

To carry out the Experiment, we have used two different locations and several setups to guarantee the soundness and reproducibility of results. Compared to other Experiments, CSI-MURDER is characterized by the need of "using" human beings during the experiment as the main goal is exploring the possibility of obfuscating the localization to preserve privacy. This implies that remote experiments are extremely complex, and we are very grateful to our patron and in particular to **Vincent Sercu** for the support provided and patience to act as a localization victim in **w-iLab.2**. However, to optimize the experimental time in **w-iLab.2**, we ran many preliminary experiments both in the **w-iLab.2** testbed using the robots extensively and in our laboratory in Brescia. Furthermore, Objective 1 (O1) and Objective 2 (O2) require different setup. Overall, we have **7** experimental setups (**4** in Brescia and **3** in **w-iLab.2**) listed below and each one described in detail in the following subsections.

B1. Passive attack with humans: Localization precision dilution in Brescia (O1)

B2. Passive Attack with objects: Comparison with humans (O1)
B3. Passive Attack with humans: Work position obfuscation (O1)
B4. Active Attack with humans: Localization precision dilution in Brescia (O2)
W1. Passive attack with humans: Localization precision dilution in **w-iLab.2** (O1)
W2. Passive attack with objects: Localization precision dilution in **w-iLab.2** (O1)
W3. Active Attack with robots: Localization precision dilution in **w-iLab.2** (O2)

In each experiment we evaluate the performance of the localization system **both under normal operating conditions and when our obfuscation system is deployed** with the goal of showing that our obfuscation system significantly affects the performance of a localization system that otherwise would work fairly well in all the considered scenarios. In every experiment both the training of the neural network and the testing of its precision are done either with or without the obfuscation, i.e., if no obfuscation is used, then the CNN is trained without obfuscation and the testing too, while if the obfuscation is used, then both the training and the testing are done with the filter active.

We have **always collected data in two different days or at least several hours apart,** to verify that the localization system is able to estimate the position not only "contextually" with the training, but also at another point in time, when the micro characteristics of the environment (temperature, humidity, etc.) have slightly changed, and also the person being tracked can be a different one, he has changed dressing, or simply his own physiological parameters (sweat, hearth rate, etc.) are not identical.

As a final note, we have **not run experiments where the CNN is trained without the obfuscation and the testing is done with the obfuscation active**, simply because a simple analysis shows that in this scenario the CNN cannot work properly even from a theoretical point of view, because the process it observes during the two phases is neither stationary nor ergodic, thus the theoretical foundations of the learning process are violated.

1.2.2.1 B1: Passive attack with humans: Localization precision dilution (Brescia)

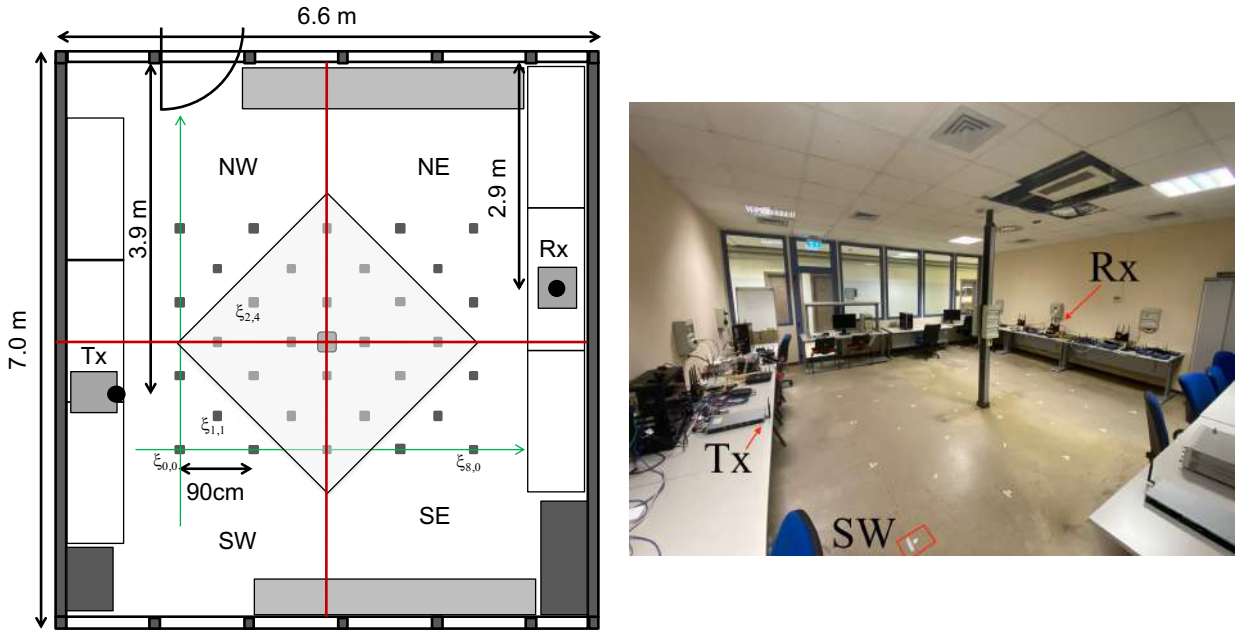


Figure 1.2.8: On the left the map of the laboratory in Brescia with the points used for training the CNN ($\zeta_{i,j}$) and the position of the transmitter and the central receiver. The space is divided into four quadrants SW–NE for the sake of clarity. On the right the picture of the laboratory: the origin point of the training grid ($\zeta_{0,0}$ - SW) is highlighted together with the transmitter and the central receiver reported in the map; the other two receivers are also visible.

The goal of this scenario is to explore how much the simple obfuscation methodology devised is efficient in diluting the precision of the localization. Figure 1.2.8 shows the layout of the experiment on the left-hand side and a picture of the laboratory with the important features on the right-hand side. The position of the transmitter and one of the receivers we used in the experiments are highlighted in the map, while important features (as well as the transmitter and the three receivers) are highlighted in the picture. The picture also shows a pillar at the centre of the room supporting electrical outlets, making the environment already complex (although not as complex as the one of **w-iLab.2** as we shall see in the relative subsection.

In this scenario, the localization system is trained using 700 packets for each $\zeta_{i,j}$ point, and the last stage of the NN (the fully connected Layer 3 in Figure 1.2.2) is configured to output the estimated cartesian coordinates rather than a classification on the $\zeta_{i,j}$ points. In other words, one of the experimenters stand in each $\zeta_{i,j}$ point for a couple of minutes, allowing the collection of enough packets to do the training and also a "contextual testing," i.e., collect at least another 70 packets that are used to verify if the localization system can actually localize the person and with which precision. After several hours (normally the day after) the same experimenter (or another one) repeats the entire experiment to test if the localization system is still able to estimate the position with enough accuracy.

As we shall discuss presenting the results in Section 1.3, the notion of position (and precision dilution) of a human body is ambiguous: Is it the projection of the barycentre on the floor? Is it the middle point between the feet? Or should we consider a 3D space? This latter hypothesis, probably the most accurate and scientifically valid is never considered in the literature to the best of our knowledge, and the CNN-based methodology we use for localization is indeed designed and trained for 2D spaces. For this reason, in general, we do not consider a position as a single point but as a 2D circle of radius ρ centred approximately on the projection of the person barycentre. If not stated otherwise, we use ρ=0.25m.

Some of the results obtained in this scenario are included in [Cominelli2020].

1.2.2.2 B2: Passive Attack with objects: Localization precision dilution (Brescia)

Most of the literature on CSI-based localization focuses on people; however, in many cases it might turn useful to adapt the same technology to localize objects. Moreover, in our particular case we are conducting extensive experimentation with robots (rather than people) in the Orca testbed since the related experiments are much quicker to schedule and easier to automate with scripts.

Robots have a much smaller volume than a person, and they expose a metallic—highly reflective—frame on top of them, so it is not trivial to foresee how the electromagnetic characteristics of channel will be modified by their movements. This experiment, conducted in our laboratory in Brescia, aims at filling the gap between human localization and small object localization and it is useful in providing a reference for localization performance of robots in the **w-iLab.2** facility.

Figure 1.2.9 shows the two objects we have chosen to run these experiments: a stove pipe and a "fake robot" (left picture). Both objects are handcrafted to mimic some appropriate characteristics. First of all they are both high enough (around 1 m) to be comparable with a human being, as some quick experiments highlighted that any object (stools, boxes, etc.) that remains well below the line of communication between the transmitter and the receiver is not able to modify the propagation environment in a significative way. The stove pipe is an almost perfect reflector, though being round scatter the electromagnetic waves rather than coherently reflecting them. The "fake robot" at first sight does not look like a robot at all, but we have crafted it in such a way that we think it may have some electromagnetic properties of a robot. First of all it has a "body" and "legs"; the legs are metallic, while that body is a sandwich of cardboard, aluminium foil, and other soft, padding materials, thus it is partially reflective, partially absorbing and highly asymmetrical, so that moving and turning it as shown in the right picture may have a differentiated influence on the propagation characteristics.

Figure 1.2.9: The "fake robot" (tripod with reflective panel) and the stove pipe used for localization experiments with objects.

For this setup we are considering only 8 target positions, represented in Figure 1.2.10. All the other characteristics of this setup are the same as setup B2 described in subsection 1.2.2.1.

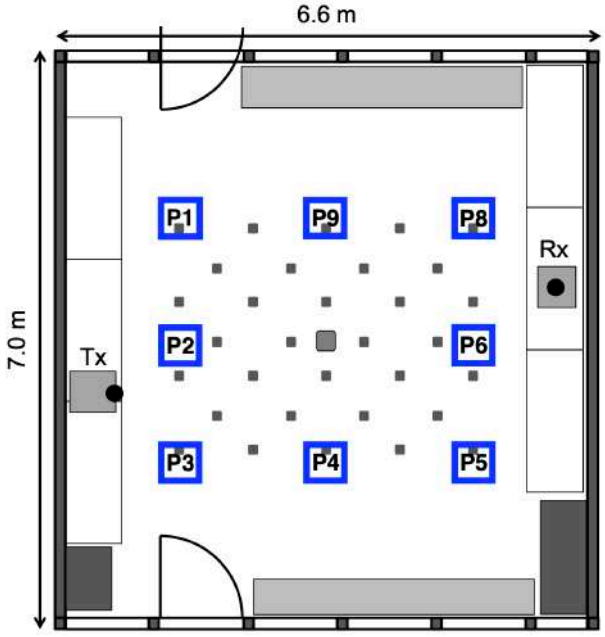


Figure 1.2.10: Representation of the 8 positions considered for object localization in our lab.

1.2.2.3 B3: Passive Attack with humans: Work position obfuscation (Brescia)

This setup inherits most of the characteristics of B1 in subsection 1.2.2.1, but it is tailored to represent a realistic threat to people privacy. As depicted in the left-hand side of Figure 1.2.11, 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 CNN-based localization system. In our specific setup, visualized in the right-hand side of Figure 1.2.11, the receiver RX and the transmitter TX are on the opposite side of the room, in the same positions of scenario B1 (Figure 1.2.8).

We make the following assumptions regarding the attacker model:

1. The attacker is able to train the localization system, which only requires collecting some measurements of reference positions; and
2. The attacker can only access the receiver and retrieve CSI from it. It's clear that this attack can be easily replicated in hotels and multi-room environments as well as in private homes.

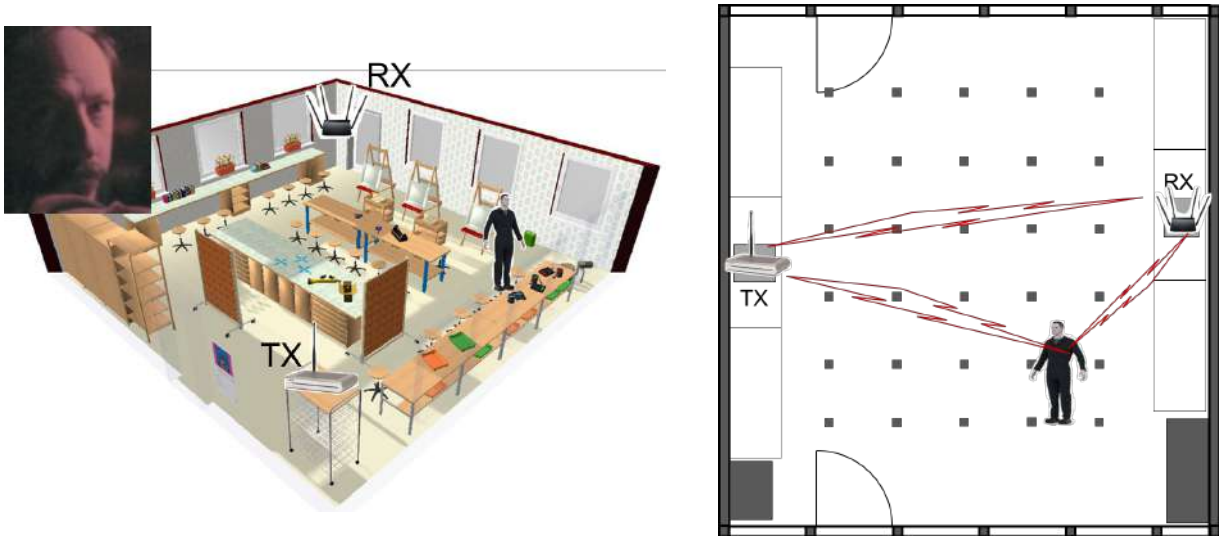


Figure 1.2.11: The "privacy attack scenario" we emulate in this setup.

The goal in this scenario is not to measure the precision dilution of the obfuscation system, but to actually measure its ability to prevent a real attack on people's privacy. To focus ideas, suppose that the goal of the attacker, i.e., the person who is trying to illegally track the position of someone, is to know in front of which working desk somebody working in the laboratory is passing his time, for instance to determine the fraction of his work time dedicated to different tasks, an act contrary to labour legislation in most countries, at least in Europe. To this end, the sectors (NW, NE, SE, SW, separated by the red lines) we divided the laboratory in Figure 1.2.8 comes handy. The shaded square of 2 m edge at the centre of the room is not considered for the localization purposes, as it is clearly 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. The goal of the attacker is thus to understand in which sector the person is passing his time, and can do this by either estimating the cartesian coordinates of the person as in scenario B1, or by training the CNN-based localization system to classify the position into one of the sectors.

Some of the results obtained in this scenario are included in [Cominelli2020].

1.2.2.4 B4: Active Attack with humans: (Brescia)

The goal of Objective 2 is to provide a countermeasure against active attackers that are not exploiting existing Wi-Fi signals but are bringing their own equipment (transmitter and receiver) on the field to perform CSI measurements. Obviously, the technique developed for Objective 1 cannot work in this scenario, but we can apply the same core idea of tampering with the packet headers to enforce again users' location privacy.

This setup provides only a proof-of-concept of the proposed anonymization method. We assume that an attacker is able to transmit and receive Wi-Fi packets using two devices that he fully controls. We also assume that the victim controls another device that can detect all packets on a given channel (included the ones transmitted by the attacker) and react immediately by injecting some "noise" in the channel, which ultimately lead to corrupted CSI reception but still preserves Wi-Fi communications. In order to emulate this, we adopted the setup described in Section 1.2.1.3. In this setup, we measure the effectiveness of CSI obfuscation on the 8 positions represented in Figure 1.2.10.

1.2.2.5 W1: Passive attack with humans: Localization precision dilution (w-iLab.2)

The laboratory in Ghent is very different from the one in Brescia, and accessing it with humans is complex and requires the cooperation of a local patron, but thanks to the experience accumulated in Brescia we designed a set of reasonable setups. Figure 1.2.12 and 1.2.14 report the map of the entire laboratory and the portion of it we used for the experiments with the position of the transmitter, the receivers and the positions of the person to be localized. The maps are not perfectly in scale, but the entire laboratory measures approximately 55 x 18 m. Figure 1.2.13 reports instead a picture of the portion of the laboratory we used for the Experiment. The presence of the ventilation pipes is evident and the devices used as transmitter and receivers are also visible (sdr).

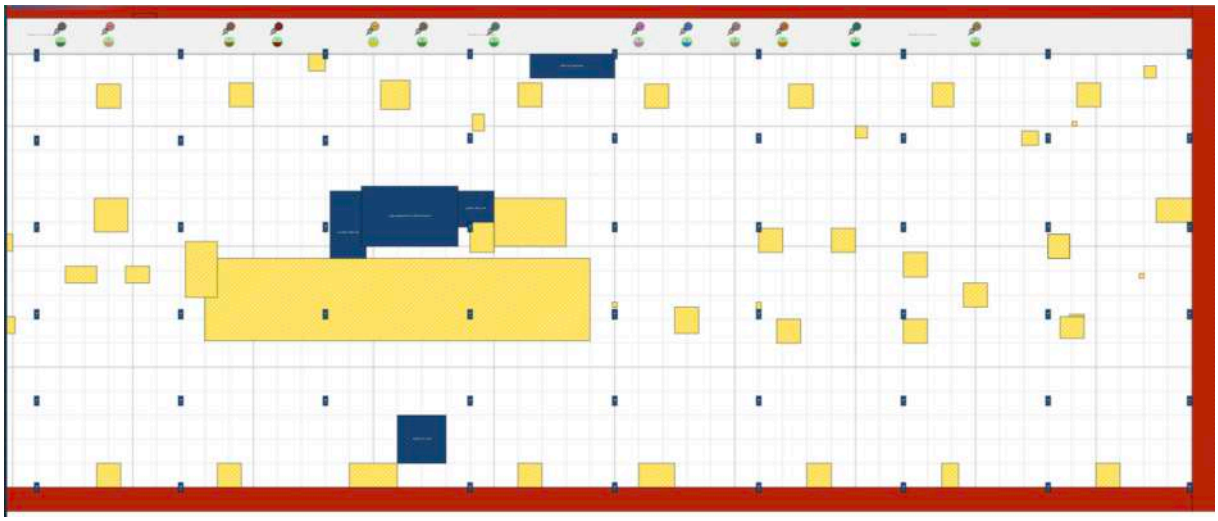


Figure 1.2.12: Complete map of the w-iLab.2, the yellow and blue squares are floor to ceiling obstacles.

The yellow and blue squares in the maps of Figures 1.2.12 and 1.2.14 are metallic obstacles that are more or less floor to ceiling, and additional obstacles (ventilation pipes) are also present horizontally (not represented in the maps, but some of them are visible in the picture of Figure 1.2.13) only the right part of the laboratory has been used for the Experiment. As in setup B1 the positions are considered to be a circle with 0.25m radius, here are shown as larger squares just for the sake of

readability. In any case, since we cannot use a regular grid, localization is based on classification and not on the extrapolation of Euclidean coordinates.

Figure 1.2.13: Picture of the portion of w-iLab.2 we used for the Experiment, the ventilation pipes are well visible as well as some of the devices available in the laboratory (sdr1,...,4) are also visible, as well as a device in the foreground dangling from the ceiling; sdr1 is the device we use as transmitter, while the receivers are realized with the robots as shown in Figure 1.2.15; this picture, with reference to the map in Figure 1.2.14 is taken from left to right roughly in the position P7.

1.2.2.6 W2: Passive Attack with objects: Localization precision dilution (w-iLab.2)

This setup is fairly identical to what we already described in the previous subsection for setup W1.

We have started our experiments on the **w-iLab.2** testbed using the available mobile nodes. In this way, we could test the functionality of the localization system and the obfuscation countermeasure with automated experiments without the need of having one person from our patron to repeatedly go into the **w-iLab.2** testbed for localization experiments. Moreover, the knowledge acquired during these preliminary trials was fundamental in designing scripts that optimize the time required to perform full CSI measurements on the testbed in setup W1.

Ultimately, we have repeated an experiment analogous to the one reported for setup W1, except for the fact that the target “victim” is a mobile node deployed in the very same positions P1,…,P10. In this document we report results only for this last experiment. For completeness, we report in Figure 1.2.16 the setup of these experiments in jFed. The target Wi-Fi link is established between nodes **sdr1** (Xilinx ZC706 Zynq running OpenWiFi and controlled by **server16**) and **apuT1**. The three mobile receivers Rx1, Rx2 and Rx3 are identified by nodes **mobile8**, **mobile7**, and **mobile5** respectively. The node **apuT1** is configured as an AP for a control network that we use to drive the robots and to issue control commands when robots are undocked.

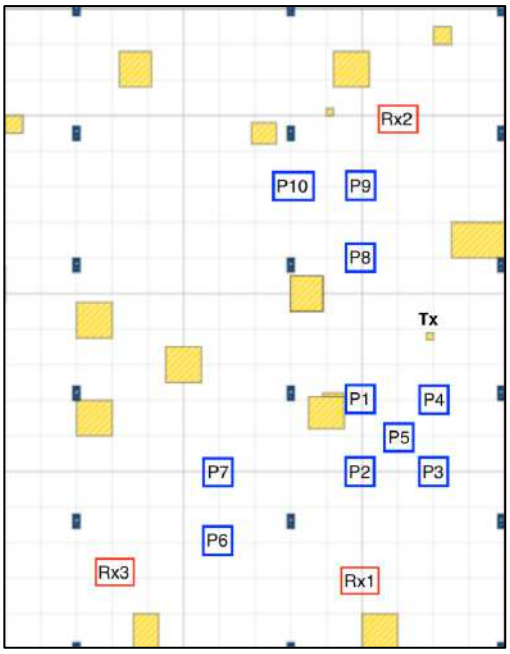
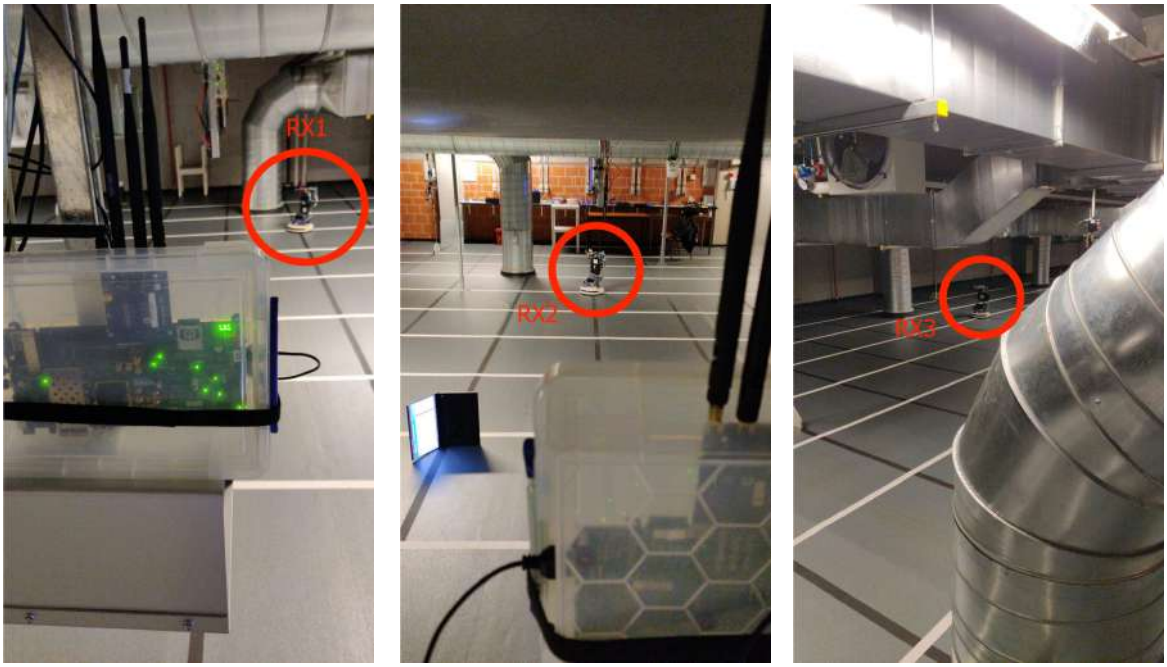


Figure 1.2.14: Portion of w-iLab.2 we used for the experiments with the position of the transmitter (Tx) the three receivers (Rx1,2,3) and the positions of the person (P1,...,10) used both for training and testing the localization system; the squares of the grid are 1m.

Figure 1.2.15: Three pictures with the robot-receivers placed in the positions indicated in Figure



1.2.14 (Rx1,2,3).

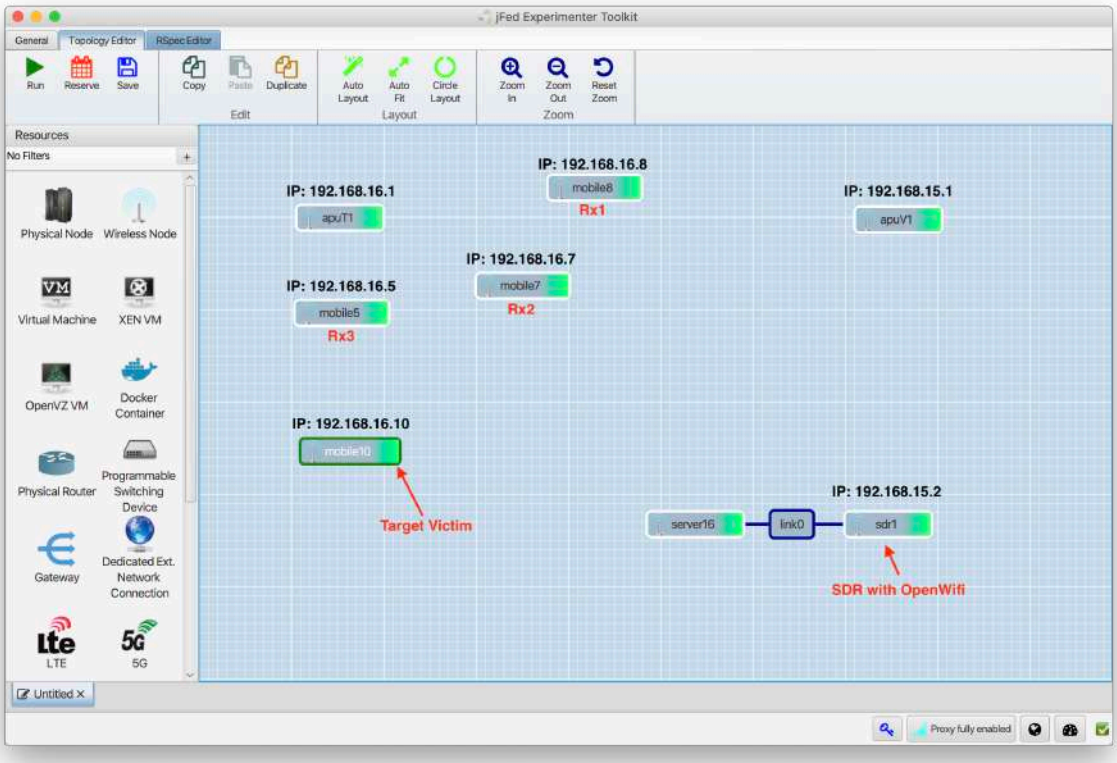


Figure 1.2.16: Configuration of the experiment in jFed for setup W1 and W2. For setup W2, the node mobile10 is also controlled through apuT1 and used as victim instead of a human target.

1.2.2.7 W3: Active Attack with robots: (w-iLab.2)

In this setup we test the proof-of-concept CSI obfuscation against active attackers, extending to w-iLab.2 the setup B4. In fact, also in this case we are emulating two distinct transmitters—the equipment brought by the attacker and the active device which provides the desired obfuscation—using two distinct channels of a USRP B210 SDR. In Figure 1.2.17 we show the setup of this experiment, with the 8 target positions of the victim robot placed around a square in a way similar to what already described for setup B4.

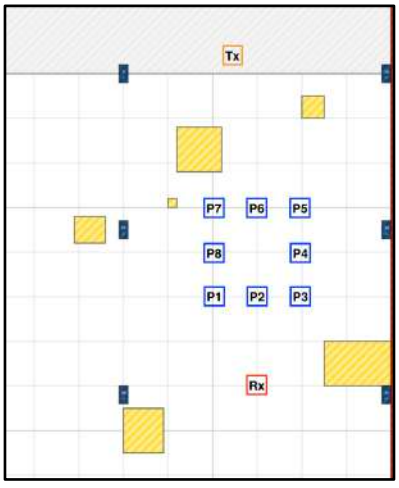


Figure 1.2.17: Map of the w-iLab.2 testbed with the 8 positions for which the active attack was evaluated; the Tx node is a USRP B210.

1.2.3 Results and analysis

It is now time to present the results obtained in the seven setups described in Section 1.2, but before presenting them we need to analyse and discuss the meaning of the measurements and the metrics that can be used to evaluate the results. The concept of "precision dilution" very often used in positioning systems derives, as well known, from the deliberate randomization done by the U.S. Department of Defence in the early days of GPS. The concept is based on the reliability of the position estimate within a Euclidean space, which is perfectly legitimate for GPS, estimating the position of a small antenna that can be considered as a point in space. The reliability is based on the simple Euclidean distance between the estimate and the true position. In our case, where we want to estimate the position of a human body, which occupies a fairly large volume in space, this is not appropriate. Thus as precision dilution we consider two different metrics depending on how the CNN-based localization system is used, whether it estimates a Euclidean coordinate or if it is tuned to perform as a classifier, i.e., it always assign an estimation point to one of the possible locations of the body/object to be identified. Estimating a Euclidean coordinate is feasible only if the training has been done on a regular and quite fine grid as we did in setup B1 in Brescia, while it is impossible if the training points are few, sparse and/or not regular; therefore, in all other setups the CNN is trained to classify one of the possible locations (e.g. P1,…,10 for scenario W1).

The first metric we use is a Euclidean distance measure to verify and validate the methodologies under analysis. As discussed above, the classical mean square error of the distance is not appropriate for our goals. 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 centre (*x,y*) can be considered the projection of the human body on the 2D plane

Given the coordinate estimate (x, y) as computed by the NN, and the co- ordinates (x_c,y_c) of the training point ξ where the person stands (see Figure 1.2.8), we construct a localization reliability index L_R as follows

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

where d_i is the Euclidean distance between the position estimate and the coordinates of the ξ where the person was when the i-th sample is taken and N_l is the total number of position samples (packets) taken to localize the person.

Clearly $L_R \in [0, 1]$ and converges to one when all position estimates are within ρ from the true position and converges to zero when all estimates are more than three times ρ from the true position. As a useful comparison to understand the reliability of localization we can use the metric above assuming the location is simply a random point in the portion of the laboratory where a person can reasonably stay, i.e., the lab minus the border where tables and furniture are. If we exclude 0.8 m around the wall, then the useful area of the lab is A_u = (6.6 − 0.8) × (7.0 − 0.8) ≃ 36 m^2, thus randomly placing the location of a person inside this area yields

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

as a function of ρ, giving a good reference to compare the localization quality and the obfuscation effectiveness.

The second metric we define and use in all the cases when the NN is used as a classifier and does not output Euclidean coordinates measures the capability of the system to actually localize a person with high reliability, but with relaxed precision. In other works, this metric simply measures the probability (or percentage) that an estimation is correct of not:

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

where N_l is the total number of position samples (packets) taken to localize the person or object as above and $I_l(i)$ is an indication function that tells if the location is correctly classified ($I_l(i)=1$) or not ($I_l(i)=0$). P_l is a simpler metric compared to L_R, and gives less insight in the obfuscation process and efficiency, but it is meaningful also in scenarios and setup where the localization system tries only to infer a "rough position" previously classified, and not a specific location based on the interpolation of Euclidean coordinates. Sometimes in the literature, especially works based on ML/AI classification techniques, instead of a simple probability metric as P_l, results are presented through "confusion matrices", which are the representation of the entire classification procedure in terms of conditional probability, i.e., where the person/object position has been estimated given the actual position of the person/object. Confusion matrices can sometimes give more insight in the results, but they are very dispersive, so we only present a few examples in Appendix A.

1.2.3.1 B1: Passive attack with humans: Localization precision dilution (Brescia)

The B1 setup is focused on evaluating the localization dilution that is achievable with the obfuscation method of CSI-MURDER. Thus, we compare the performance obtained with and without the obfuscation. For both cases, we train the NN with 700 packets for each one of the 32 positions highlighted in Figure 1.2.8 and then we test the localization performance 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 Figure 1.2.8) 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 1.2.18 reports the average results obtained considering the metric L_R. The solid line is the average for the 32 positions computed by considering all the CSI (averaged over 12 antennas). The shaded regions around the solid lines, show the area between the worst and best performing antenna, obtained again by averaging over the 32 positions. All the lines increase with ρ as expected, the most interesting cases for localization are the ones with small ρ, e.g. for values between 0.3 and 0.6. In particular, for ρ = 0.3, the average LR 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 uniformly distributed random guesses L_R^{rand}.

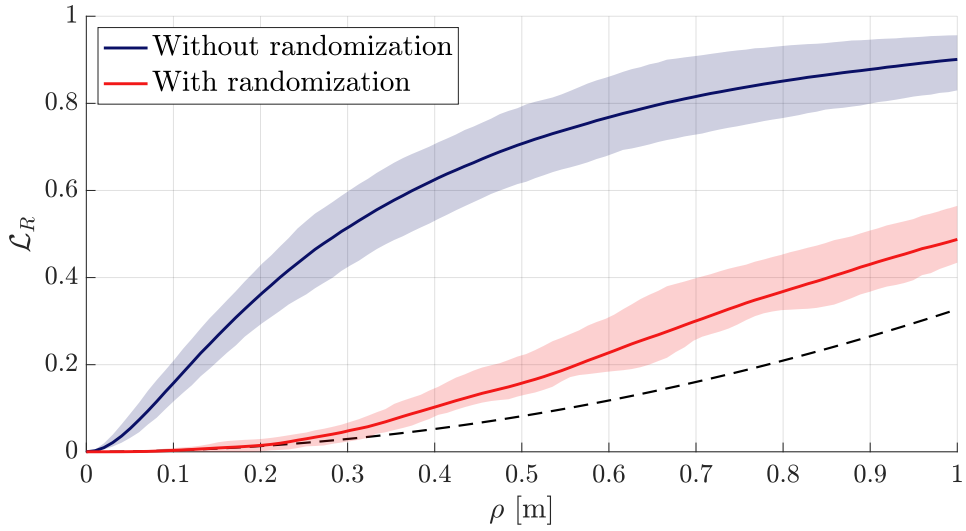


Figure 1.2.18: Localization performance according to metric L_R for ρ ranging from 0 to 1. Solid lines report the average result, while the shaded areas are the envelope of all measures including using different antennas and positions of the receiver. The dashed line is the theoretical result for uniformly distributed random guesses.

Figure 1.2.18 gives a clear representation of the feasibility and power of localization obfuscation. Unfortunately, in its simplicity and clarity it requires to train the localization system on a regular grid of points, so that it is able to infer Euclidean coordinates, and it requires a very long experimentation time involving humans, so it is not possible to replicate these results for the setup W1, where a regular grid of points is not feasible and we cannot ask our patrons to spend hours and hours in standing in specific points in the lab. For this reason, we also present the same results as accuracy matrices in Tables 1.2.2 and 1.2.3, so that results can be compared across different setups.

On the same data, we reconfigure the localization system to classify the location, i.e., for each and every packet received the system assigns the position of the person to one of the possible position that is has learned during the training phase, rather than trying to extrapolate Euclidean coordinates. We present two set of results: Table 1.2.2 refer to the performance of the localizer immediately after the training (the dataset is the same, just running the testing on the last 70 packets for each position) and Table 1.2.3 to the localization performance obtained in another moment, normally after one day. Table 1.2.3 correspond to the same scenario of Figure 1.2.18, while Table 1.2.2 determines a sort of upper bound for the localization system, because the testing is done on the same dataset as the as training, which is not a credible scenario in localization: the position of a person is inferred when she/he has not moved at all, so the attacker knows the position, and the position errors are due only to the localization system errors themselves. Both tables present the results in the case of *clean* CSI and in the case of *obfuscated* CSI.

Using the same dataset (Table 1.2.2) the localization system is extremely accurate with clean CSI, while with the obfuscated CSI the classification estimation still corresponds to a decent "educated guess" as the position is correctly classified in nearly 50% of the cases. When instead the localization is attempted in another moment (Table 1.2.3) the obfuscation technique completely prevents the localization system to have even an educated guess of the actual position. It is interesting to notice that the overall accuracy (3%) corresponds nearly perfectly to a random guess over 32 positions, even if the actual distribution of the guesses is not uniform, but favours two of the possible positions. We do not have an explanation for this behaviour. To be completely transparent in our presentation, it

must be mentioned that the localization system is not very accurate if the testing is done in a different moment than the training, but still it correctly guesses the position about 50% of the times, which is much more than the uniform probability of a random guess (1/32).

Classification accuracy [%] – clean CSI, 1 run									
j = 6	92.6	-	100	-	71.4	-	100	-	100
j = 5	-	100	-	100	-	100	-	100	-
j = 4	100	-	82.9	-	100	-	40.0	-	100
j = 3	-	95.7	-	100	-	100	-	100	-
j = 2	100	-	100	-	81.4	-	75.7	-	100
j = 1	-	97.1	-	81.4	-	90.0	-	100	-
j = 0	100	-	90.0	-	100	-	100	-	100
$\zeta_{i,j}$	i = 0	i = 1	i = 2	i = 3	i = 4	i = 5	i = 6	i = 7	i = 8
Overall accuracy = 93.7 %									
Classification accuracy [%] – obfuscated CSI, 1 run									
j = 6	98.6	-	75.7	-	38.6	-	100	-	22.9
j = 5	-	71.4	-	4.3	-	14.3	-	92.9	-
j = 4	37.1	-	7.1	-	100	-	20.0	-	48.6
j = 3	-	84.3	-	71.4	-	44.3	-	15.7	-
j = 2	61.4	-	100	-	35.7	-	21.42	-	100
j = 1	-	0.0	-	10.0	-	20.0	-	84.3	-
j = 0	7.1	-	34.3	-	45.7	-	70.0	-	27.1
$\zeta_{i,j}$	i = 0	i = 1	i = 2	i = 3	i = 4	i = 5	i = 6	i = 7	i = 8
Overall accuracy = 48.9 %									

Table 1.2.2: Localization accuracy over the 32 $\zeta_{i,j}$ positions with *clean* and *obfuscated* CSI; training and testing samples are extracted from the same run.

Classification accuracy [%] – clean CSI, 2 run									
j = 6	100	-	57.1	-	71.4	-	70.0	-	100
j = 5	-	92.9	-	100	-	90.0	-	1.4	-
j = 4	0.0	-	100	-	100	-	0.0	-	100
j = 3	-	0.0	-	90.0	-	92.9	-	4.3	-
j = 2	58.6	-	100	-	1.4	-	0.0	-	4.3
j = 1	-	0.0	-	0.0	-	0.0	-	30.0	-
j = 0	52.9	-	71.4	-	0.0	-	100	-	1.4
$\zeta_{i,j}$	i = 0	i = 1	i = 2	i = 3	i = 4	i = 5	i = 6	i = 7	i = 8
Overall accuracy = 49.7 %									
Classification accuracy [%] – obfuscated CSI, 2 run									
j = 6	0.0	-	0.0	-	0.0	-	0.0	-	0.0
j = 5	-	0.0	-	0.0	-	0.0	-	0.0	-
j = 4	0.0	-	0.0	-	0.0	-	0.0	-	0.0
j = 3	-	75.7	-	0.0	-	0.0	-	0.0	-
j = 2	0.0	-	21.4	-	0.0	-	0.0	-	0.0
j = 1	-	0.0	-	0.0	-	0.0	-	0.0	-
j = 0	0.0	-	0.0	-	0.0	-	0.0	-	0.0
$\zeta_{i,j}$	i = 0	i = 1	i = 2	i = 3	i = 4	i = 5	i = 6	i = 7	i = 8
Overall accuracy = 3.0 %									

Table 1.2.3: Localization accuracy over the 32 $\zeta_{i,j}$ positions with *clean* and *obfuscated* CSI; training and testing samples are extracted from two different runs.

In this setup, given the availability of 802.11ac hardware, we also run experiments to understand if and how the CSI obfuscation affect the communication performance. In experiments dedicated to localization, only packets with the lowest-order modulation and coding scheme (MCS) (i.e., MCS0) that uses BPSK are transmitted for the sake of efficiency, as the preambles that allow the computation of the CSI always use MCS0. However, it is important to investigate the communication performance for higher-order MCSs because they are more susceptible to channel errors. Hence, we computed the Packet Delivery Ratio for all VHT-PHY MCS transmitted with 80 MHz bandwidth and a single spatial stream: Table 1.2.4 reports the Packet Delivery Ratio (PDR) for the three receivers when randomization is off (w/o) and on (w).

It appears from the table that the positions of the three receivers enable acceptable performance for all MCS: only one receiver (Rx 1) suffers a bit with MCS9 without randomization. As easily predictable, only robust MCSs retain acceptable 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 (Rx 3) and kills reception at another one (Rx 1). Things get worse when further increasing the MCS: MCS8 and 9 cannot be received at almost any position.

MCS index	Mbit/s	Rx 1 w/o %	Rx 1 w %	Rx 2 w/o %	Rx 2 w %	Rx 3 w/o %	Rx 3 w %
0-BPSK	29.3	95.5	94.8	96.4	95.0	95.7	94.9
1-QPSK	58.5	90.7	92.8	91.7	93.8	91.5	93.3
2-QPSK	87.8	95.6	94.6	96.1	94.6	96.2	94.9
3-16-QAM	117.0	92.4	93.2	92.7	94.0	92.8	94.0
4-16-QAM	175.5	92.2	91.8	92.4	94.3	92.3	94.3
5-64-QAM	234.0	93.2	11.9	94.4	91.2	93.8	80.2
6-64-QAM	263.3	94.0	4.7	93.3	93.4	93.4	65.8
7-64-QAM	292.5	94.3	1.1	95.4	79.8	95.5	40.9
8-256-QAM	351.0	92.4	0.0	93.6	12.0	93.6	0.1
9-256-QAM	390.0	71.0	0.0	94.9	0.2	94.3	0.0

Table 1.2.4: Packet Delivery Ratio as a function of the Modulation and Coding Scheme, with and without the CSI obfuscation block.

We think that these results are extremely encouraging, as we did not designed the obfuscation method having the transmission performance as a constraint, rather, we focused on the feasibility of localization obfuscation. The design of an obfuscation methodology that takes as constraint the transmission performance can be tackled, not that it is clear that localization can be obfuscated, with a theoretical approach that is inherently outside the scope of an ORCA Experiment. Given the absence of 802.11ac hardware in **w-iLab.2** testbed, we have not pursued the analysis of throughput further.

1.2.3.2 B2: Passive Attack with objects: Comparison with humans (Brescia)

In this setup we are comparing the performance of the localization system when our target is not a human body but a relatively small object made of reflective material. The results described in this section are useful to compare the localization performance in the two cases and move from the idea of replicating in our lab in Brescia the experiments conducted with robots on the **w-iLab.2** testbed.

We see from Table 1.2.5 that when training and testing samples are drawn from the same CSI collection run, the CNN is always able to output the correct result. However, when we are drawing samples from two different runs (Table 1.2.6), the performance of the localization system begins to vary more

consistently when applied to locate different objects. It appears that human presence has a stronger impact on the electromagnetic characterization of the channel, leading to recurrent patterns that are "more recognizable" by the CNN, despite some natural variability in the CSI. These results ultimately lead us to conduct localization experiments in the w-iLab.2 testbed not only using the mobile nodes but also with human targets, with the kind support of our patron in Ghent.

Stove pipe									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	100	100	100	100	100	100	100	100	**100**
Tripod with reflective panel									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	100	100	100	100	100	100	100	100	**100**
Human									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	97.1	100	100	100	100	100	100	100	**99.6**

Table 1.2.5: Accuracy of the localization models learned on *clean* CSI with different targets; training and testing samples are selected from the same run.

Stove pipe									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	0.0	100	100	100	0.0	8.6	24.3	100	**54.1**
Tripod with reflective panel									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	92.9	100	0.0	100	100	100	100	0.0	**74.1**
Human									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	100	90.0	100	24.3	40.0	65.7	100	100	**77.5**

Table 1.2.6: Accuracy of the localization models learned on *clean* CSI with different targets; training and testing samples are selected from two different runs.

1.2.3.3 B3: Passive Attack with humans: Work position obfuscation (Brescia)

In this scenario we reduce the possible target positions, as we imagine that the attacker is not interested in deriving the precise location of the victim but rather having a rough estimate about which area of the room the victim is occupying. Training data consists of CSI collected when the victim is slowly moving in each one of the four corners of the room, labelled SW, NW, NE and SE respectively (see Figure 1.2.8). It is expected that in this case the classification carried out by the CNN is much more accurate with respect to the previous case with a finer grid, as shown both in Table 1.2.7 and Table 1.2.8. Also in this case, our approach proves effective against passive localization since localization performance is sensibly lower when the CSI are obfuscated.

In this setup it seems that the localization system is not much affected by the fact that testing is done on the same run of data as the training or not. Possibly this is due to the fact that a "loose localization" in terms of room sectors rather than precise positions present the learning system with a wide set of possible CSI so that the NN does not learn peculiarities of CSI for a specific point in space and time, but rather learns the generic properties of the CSI when a human being is in one of the sectors and these

properties are less sensitive to time-based changes of the environment. When the CSI is obfuscated it seems that the NE sector is unaffected by the obfuscation. We could not analyse the reasons for this, but we think that it will be possible in the future to design more efficient obfuscation filters.

Clean CSI					
Pos.	SW	NW	NE	SE	**Overall**
Acc. [%]	100	100	100	100	**100**
Obfuscated CSI					
Pos.	SW	NW	NE	SE	**Overall**
Acc. [%]	52.0	47.7	100	8.0	**51.7**

Table 1.2.7: Accuracy of the localization models learned on *clean* and *obfuscated* CSI with *human* target; training and testing samples are selected from the same run.

Clean CSI					
Pos.	SW	NW	NE	SE	**Overall**
Acc. [%]	100	100	100	100	**100**
Obfuscated CSI					
Pos.	SW	NW	NE	SE	**Overall**
Acc. [%]	14.7	76.0	93.3	36.0	**55.0**

Table 1.2.8: Accuracy of the localization models learned on *clean* and *obfuscated* CSI with *human* target; training and testing samples are selected from two different runs.

Figure 1.2.19 presents results for a similar case, but were the CNN is trained to output Euclidean coordinates, (x,y) estimates, of the victim's position, and the training is done with the victim sitting on a chair as if working normally. The "normal" position is represented by circles with a 60 cm. radius, a reasonable representation of how a person would normally sit and slightly move on an office chair. The data visualization clearly shows that also in this case a passive attacker can be easily tricked by our obfuscation mechanism in determining positions that are completely useless to reliably track a person.

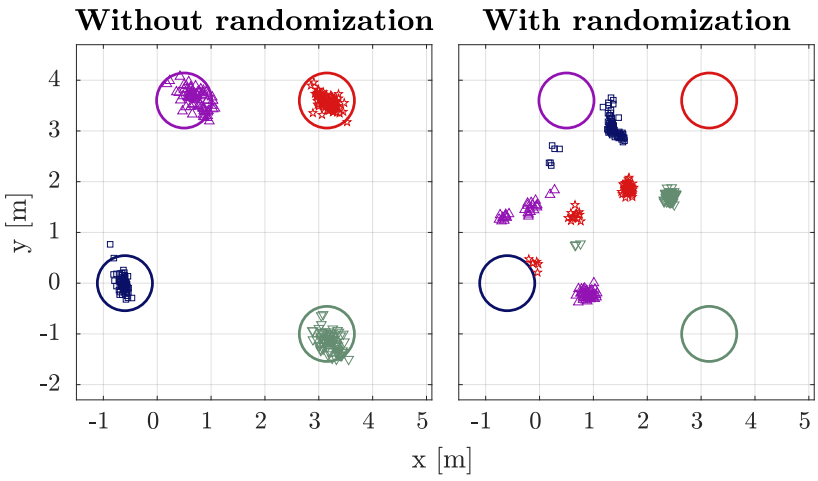


Figure 1.2.19: Localization estimates when the CNN is trained to output (x,y) coordinates. The four circles represent the location of the training/testing positions.

1.2.3.4 B4: Active Attack with humans: (Brescia)

With respect to Objective 2 of our experiment, we have conducted both in Brescia and in Ghent some preliminary experiments concerning the location privacy protection against active attacks, i.e. the ones in which the attacker owns and controls both the transmitter and the receiver. We start showing the results obtained in our local setup; in section W3 we present the results obtained in the w-iLab.2 testbed, performing some comparisons between the two setups and drawing some conclusions.

Preliminary results show that an obfuscation system preventing such types of attack require to react as fast as possible when detecting a new Wi-Fi packet. A set of pure tones, randomly placed on the wide-band spectrum of the signal, must be transmitted as soon as the packet is detected and for the entire duration of the header in order to obtain the same CSI obfuscation effect that we already described for passive setups.

The 8 positions P1,…,8 considered in this experiment are the same positions used in scenario B2 and are illustrated in Figure 1.2.10. The target of the localization is a person standing in the corresponding positions. We show both in Table 1.2.9 and in Table 1.2.10 that also the new "active" countermeasure works well at anonymizing user's location when it is able to detect transmitted packets early and react immediately to generate spurious CSI information for the attacker.

Clean CSI									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	100	100	100	100	100	100	100	100	**100**
Obfuscated CSI									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	28.6	0.0	4.3	7.1	81.4	0.0	98.6	94.3	**39.3**

Table 1.2.9: Accuracy of the localization models learned on *clean* and *obfuscated* CSI with *human* target; training and testing samples are selected from the same experiment.

Clean CSI									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	100	100	91.4	60.0	100	0.0	55.7	100	**75.9**
Obfuscated CSI									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	88.6	0.0	0.0	4.3	5.7	28.6	74.3	67.1	**33.6**

Table 1.2.10: Accuracy of the localization models learned on *clean* and *obfuscated* CSI with *human* target; training and testing samples are selected from two different experiment.

1.2.3.5 W1: Passive attack with humans: Localization precision dilution (w-iLab.2)

Also for this setup in which we want to localize a human victim, we measure the localization performance 1) when the training and testing samples are drawn from the same experiment (the testing data is at the end of the experiment), and 2) when the training and testing samples are drawn respectively from two different repetitions of the experiment.

For scenario 1) we show in Table 1.2.11 that the localization system is able to classify almost perfectly the ten target positions; however, even in this overly optimistic scenario (for the attacker perspective), the proposed obfuscation framework has a severe impact on classification accuracy, as shown in the same Table with *obfuscated* CSI.

Localization performance quickly degrades in scenario 2) in a way similar to what we already observed in our previous experiments. Nevertheless, the results in Table 1.2.12 show that the localization system still works quite well with clean CSI, but it is doomed to complete failure if the CSI obfuscation system is in place.

The discussion is similar to what we have verified in the previous experiments: also in the challenging environment of the **w-iLab.2** facility—rich of reflective objects and obstacles that strongly interfere with electromagnetic propagation—even the presence of a human body can modify the propagation environment in a meaningful and recognizable way, since the localization system can correctly locate the person with a significative accuracy at least if not much time passes between the training of the localization system and the testing on the victim, while after some time the localization precision drops to about 50% (still quit high compare with 0.1 of a random guess over 10 possible positions).

Rx1, clean CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	100	100	100	92.9	100	100	100	98.6	100	100	**99.1**
Rx2, clean CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	100	100	100	100	98.6	84.3	98.6	100	100	8.6	**89.0**
Rx3, clean CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	100	100	100	100	100	100	100	97.1	100	100	**99.7**
Rx1, obfuscated CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	78.6	0.0	100	0.0	0.0	0.0	100	100	0.0	77.1	**45.6**
Rx2, obfuscated CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	100	48.6	0.0	0.0	0.0	0.0	97.1	4.3	0.0	0.0	**25.0**
Rx3, obfuscated CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	15.7	14.3	0.0	100	45.7	100	100	30.0	0.0	15.7	**42.1**

Table 1.2.11: Accuracy of the localization models learned by the three Rx for P1–10 on *clean* and *obfuscated* CSI with *human* target; training and testing samples are selected from the same run.

In any case, obfuscation is very effective, reducing localization precision below 40% during the same experiment, and down to a uniform random guess if testing is done in a different experiment than training. It is interesting to notice that in all results it looks like there are locations that are much easier to "pin" for the localization system and they are also more difficult to obfuscate, but indeed, this is not a property of the position, rather of the position, the receiver, time and possibly many other parameters we do not control. This seems to be a "property" of **w-iLab.2** and we have not observed it in Brescia.

Rx1, clean CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	0.0	100	100	100	82.9	64.3	14.3	77.1	68.6	82.9	**69.0**
Rx2, clean CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	50.0	100.0	52.9	0.0	80.0	52.9	87.1	0.0	87.1	67.1	**57.7**
Rx3, clean CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	0.0	47.1	28.6	100	12.9	7.1	68.6	0.0	91.4	100	**45.6**
Rx1, obfuscated CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	0.0	2.9	0.0	97.1	0.0	4.3	14.3	0.0	1.4	68.6	**18.9**
Rx2, obfuscated CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	0.0	8.6	0.0	0.0	0.0	0.0	0.0	0.0	50.0	0.0	**5.9**
Rx2, obfuscated CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	22.9	0.0	0.0	0.0	0.0	0.0	1.4	0.0	71.4	0.0	**9.6**

Table 1.2.12: Accuracy of the localization models learned by the three Rx for P1–10 on *clean* and *obfuscated* CSI with *human* target, training and testing samples are drawn from two different experiments.

A confirmation that in the complex (and indeed unusual) environment of **w-iLab.2** the overall combination of devices and objects is dominant can be seen in Figure 1.2.20 where we show the CSI collected for four different positions of the victim by the same receiver. The CSIs looks almost identical to the human eye, yet we know that the localization system can find and distinguish appropriately the slight differences present. A further confirmation of this comes from the CSIs presented in Figure 1.2.21 collected for the same position of the victim at the three receivers: they are completely different!

The CSI obfuscation system introduces time-varying features and these are learned by the neural network so that localization performance is compromised, but clearly if a combination of the receiver, transmitter and all the other features is so characteristic that remains distinguishable even in face of the random artificial features, then this specific combination remains difficult to obfuscate, an one may conclude that localization is still possible, while more precisely one should say that that specific overall combination is difficult to obfuscate.

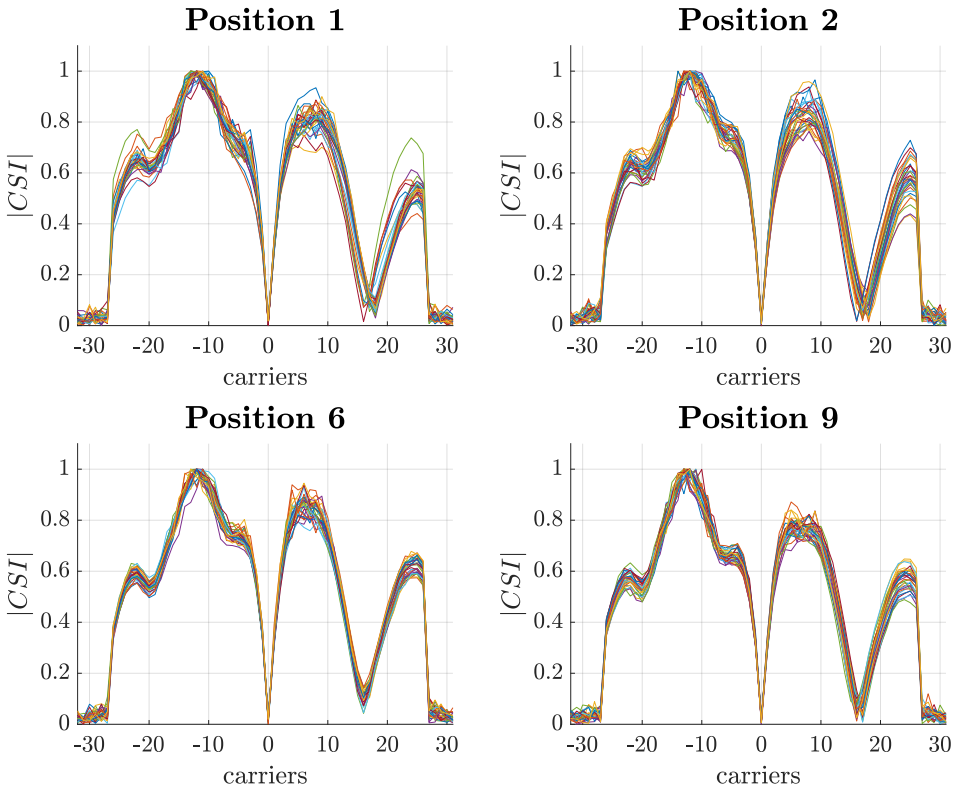


Figure 1.2.20: Magnitude (normalized) of the *clean* CSI collected by Rx1 for different positions of the target person.

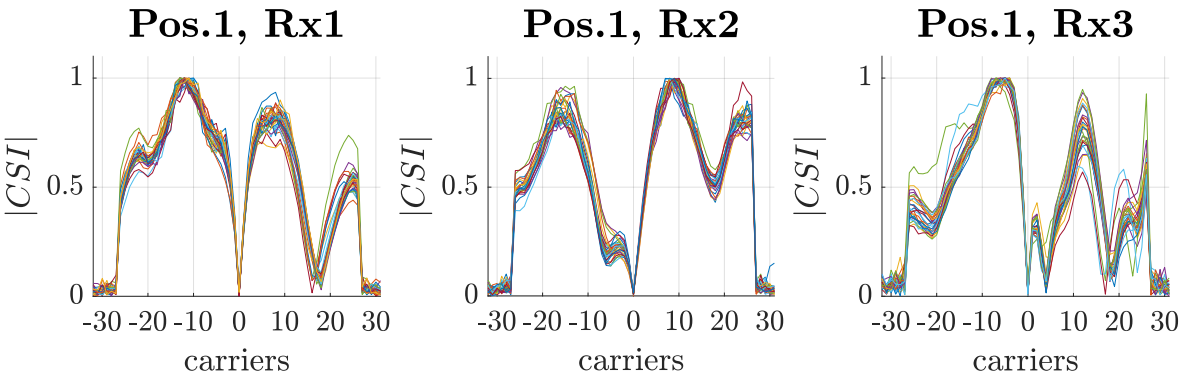


Figure 1.2.21: Comparison among CSI received by the three Rx for the same position of the victim.

1.2.3.6 W2: Passive Attack with objects: Localization precision dilution (w-iLab.2)

The results discussed for scenario W1—captured with a human victim in the **w-iLab.2** testbed—are fundamental for CSI-MURDER perspective, but they are obviously too expensive (from the perspective of the person(s) training the localization system first and then playing the role of the victim) to collect extensive results. Thus, from the very design of the Experiment, we planned to use the robots available in the testbed for a large measurement campaign, albeit, as already discussed, the actual impact of robots on the CSI was to be verified. We did this verification running an extensive evaluation campaign in this setup, before running the actual experiments on obfuscation. During this part we also did all the tunings necessary to work with robots rather than humans.

The possibility of scripting the movements of the mobile nodes (robots) and programming in advance the execution of the experiments (both on mobile nodes and on the SDR platform with OpenWiFi),

enabled to plan complex trajectories of the robots in the testbed that ultimately led to many data collection campaigns in a fully automated manner.

Data collected in this way were analysed and turned out very useful to determine the overall performance of our prototypes, as well as to spot pitfalls in our approach and correct our localization/obfuscation methods. Here we report results on passive robot localization that mainly confirm what we have already seen for scenarios B2 and W1. By drawing (without reinsertion) training samples and testing samples from the same dataset, the localization system works very well on *clean* CSI, while when training and testing samples are taken from two different runs of the same experiment (collected at different moments in time), then the localization performance in *clean* condition drops significantly, in this case to a mere 30% of accuracy. These results hints to the fact that objects (at least small objects) localization with Wi-Fi may be more difficult than human localization, but also that it is in any case feasible and countermeasures must be studied.

Our obfuscation proposal in any case works correctly, reducing the accuracy of localization, once more to a uniform random guess when training and testing are done on different runs, and to an average 30% when they are taken in the same run.

Tables 1.2.13 and 1.2.14 report the results in the two cases, and again we can observe that there are ambient combinations (position, receiver, time, etc.) that have completely different performance, sometimes favouring and sometimes hampering the localization.

Rx1, clean CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	100	100	100	100	95.7	95.7	98.6	100	100	90.0	**98.0**
Rx2, clean CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	100	98.6	100	98.6	97.1	25.7	67.1	100	100	100	**88.7**
Rx3, clean CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	100	100	100	100	100	100	100	98.6	100	100	**99.9**
Rx1, obfuscated CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	0.0	0.0	0.0	0.0	67.1	100	0.0	0.0	100	100	**36.7**
Rx2, obfuscated CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	0.0	0.0	100	11.4	18.6	100	97.1	100	28.6	100	**55.6**
Rx3, obfuscated CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	0.0	100	100	24.3	2.9	7.1	97.1	0.0	0.0	100	**43.1**

Table 1.2.13: Accuracy of the localization models learned by the three Rx for P1–10 on *clean* and *obfuscated* CSI with *robot* target; training and testing samples are selected from the same experiment.

Rx1, clean CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	67.1	100	5.7	0.0	0.0	1.4	67.1	0.0	1.4	25.7	**26.9**
Rx2, clean CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	0.0	75.7	92.9	97.1	37.1	38.6	1.4	0.0	0.0	1.4	**34.4**
Rx3, clean CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	0.0	0.0	95.7	0.0	0.0	90.0	0.0	7.1	0.0	0.0	**29.3**
Rx1, obfuscated CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	2.9	0.0	0.0	44.3	0.0	21.4	0.0	35.7	18.6	0.0	**12.3**
Rx2, obfuscated CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	100	0.0	0.0	0.0	2.9	0.0	0.0	0.0	0.0	4.3	**10.7**
Rx3, obfuscated CSI											
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	**Overall**
Acc. [%]	0.0	0.0	1.4	2.9	22.9	0.0	1.4	21.4	31.4	0.0	**8.1**

Table 1.2.14: Accuracy of the localization models learned by the three Rx for P1–10 on *clean* and *obfuscated* CSI with *robot* target, training and testing samples are selected from two different experiments.

Figure 1.2.22 reports the CSI collected at the three receivers for the one single position of the victim robot, once more it is clear that the receiver position dominates the CSI pattern, and this is a fundamental piece of information, that we collected thanks to **w-iLab.2** testbed, for the development of a theory of obfuscation and theoretically sound methodologies for obfuscation implementation.

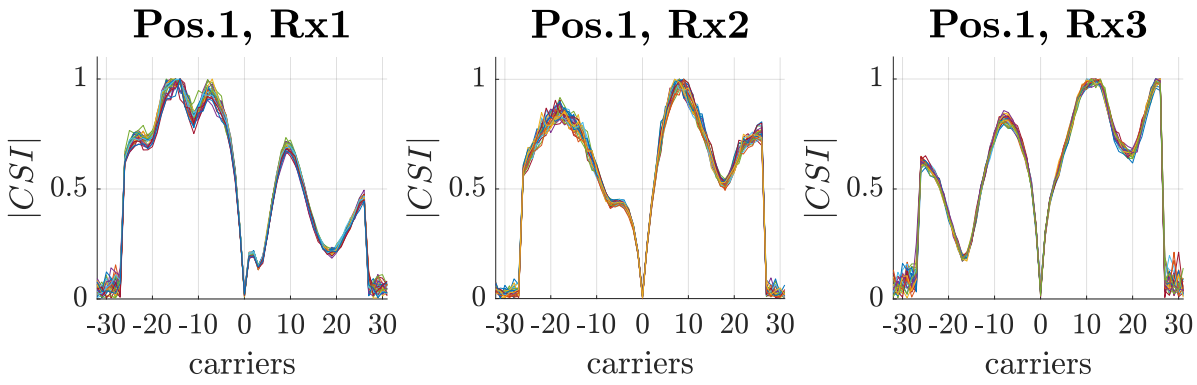


Figure 1.2.22: Comparison among CSI received by the three Rx for the same position of the robot "victim," showing the importance of the receiver position in determining the CSI characteristics, a very important piece of information for future works.

1.2.3.7 W3: Active Attack with robots: (w-iLab.2)

In scenario B4 we presented the preliminary results against active attacks obtained for a human victim in our local setup. Here we generalize those results by applying the same countermeasure in a different setting, namely the **w-iLab.2** testbed in Ghent with target robots as localization victims.

Table 1.2.15 reports the results in the Orca testbed showing that localization accuracy drops dramatically when the obfuscation mechanism is active both when localization is done on the same run of data and when it is done on a different run. We are using two different chains of a USRP B210 to emulate a system that reactively obfuscates user's location by injecting spurious information in the CSI transmitted by another node. Once more, we remark that the complex environment of **w-iLab.2**—rich of reflections and NLOS paths—characterizes the channel information in very complex ways, so that the effect of the obfuscation system is diminished with respect to scenario B4.

We already know that the reduced volume of a robot is not sufficient to produce large variations in the CSI collected for different positions of the victim. For this reason, we expect that localization performance in this case are worse with respect to the human localization case proposed in scenario B4. This is indeed verified by the results shown in Table 1.2.16, in which we once again verify that the proposed obfuscation mechanism is working effectively, but it also suffers from the extremely complex electromagnetic environment.

As already observed several times, the localization precision and the obfuscation efficiency are heavily dependent on the overall combination of transmitter, position, receiver and other parameters hard to control, so that results are not uniform and position independent, but highly skewed as a function of the specific position and receiver considered (the transmitter is always the same).

Clean CSI									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	100	100	100	100	100	98.6	100	100	**99.8**
Obfuscated CSI									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	78.6	88.6	98.6	61.4	2.9	10.0	82.9	100	**65.4**

Table 1.2.15: Accuracy of the localization models learned on *clean* and *obfuscated* CSI with *robot* target; training and testing samples are selected from the same experiment.

Clean CSI									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	0.0	1.4	64.3	0.0	0.0	95.7	100	100	**45.2**
Obfuscated CSI									
Pos.	P1	P2	P3	P4	P5	P6	P7	P8	**Overall**
Acc. [%]	20.0	37.1	4.3	24.3	34.3	2.9	68.6	0.0	**23.9**

Table 1.2.16: Accuracy of the localization models learned on *clean* and *obfuscated* CSI with *robot* target; training and testing samples are selected from two different experiment.

1.3 Impact

The Experiment proposed by CSI-MURDER is very specific and peculiar as it deals with the possibility of localizing people and objects using standard Wi-Fi devices. Differently from other work, where the target of the localization technique is a Wi-Fi transmitter, the CSI-MURDER experiments have been targeting the passive device-free localization of a person through the opportunistic analysis of the Wi-Fi frames preamble and the information it carries: this is normally used by receiving devices to derive the CSI (Channel State Information) and tune the equalization blocks that they use for optimizing the reception. Indeed, the Experiment goal is to demonstrate that it is possible to prevent such usage of Wi-Fi frames, and forbid malicious eavesdroppers from localising unaware people like neighbours or detecting their activities at their places.

Please describe how the Experiment / Extension contributes to the research or serves your business goals (in case of innovation Experiment by Industry).

The success of the CSI-MURDER experiment confirms that it is possible to re-enable the privacy of people by properly modifying in a non-destructive fashion the transmitted Wi-Fi frames: indeed frames can still be decoded as they would be in a communication system, but the information that they provide about the environment appears to an eavesdropper as noise. We believe that this has a great impact as it paves the way to new lines of research.

Describe in detail how this Experiment may impact your business and product development (in case of innovation Experiment by SME), or your scientific roadmap (in case of scientific excellence Experiment).

Before CSI-MURDER, besides creating tools for extracting CSI, we were using CSI to improve localization of transmitters [Ricciato2018] or for creating covert channels [Schulz2018]. After CSI-MURDER we are going to study the problem of CSI randomization with much higher emphasis as it will probably become one of our principal research themes. According to the output of the CSI-MURDER experiment, we will iteratively improve the capabilities of the localization system and that of the obfuscation procedure, until when the signal received by a Wi-Fi device will really appear as physically propagating within a random environment. This is of course a long-term goal, which will probably require a better understanding of the problem and access to even more complex experimentation tools, probably yet to be designed.

What is the value you have perceived from this Experiment/Extension? E.g. gained knowledge; acquired new competences; practical implementation solutions; new ideas for experiments/products; etc.

Before CSI-MURDER we developed a CSI extraction tool within the Nexmon project that enables CSI data extraction from very common devices, i.e., those embedding a Broadcom wireless chipset [Gringoli2019]. Prior to the CSI-MURDER project, however, we were missing a specific use case validating the CSI data produced by the tool. Within CSI-MURDER we demonstrated for the first time (by implementing a passive localization system) that the CSI data produced by this tool is meaningful. More specifically, according to the positive results obtained, we proved that the data is as significant as that produced by competing tools, i.e., the Linux 802.11n CSI Tool for Intel card, and the Atheros CSI Tool for QCA chipset. If we consider that the chipset we target with our tool is embedded inside the majority of Android based mobile phones, it becomes clear that thanks to CSI-MURDER the tool can become one of the most adopted by the research community.

During the project we also got familiar with the OpenWiFi tool which makes the implementation of the randomization procedure straightforward. Without OpenWiFi, we should use complex SDR tools and create each frame on a host with Matlab, then apply the randomisation “at hand” before sending samples to the SDR radio. This approach had many drawbacks: first, the transmitter is not a real Wi-Fi

node running a full Wi-Fi stack; second, we cannot evaluate the randomisation performance within the dynamics of a real Wi-Fi network. With OpenWiFi, instead, we had the possibility to run localization experiments extracting the CSI from real *iperf* data sessions. The successful demonstration of the technique running inside OpenWiFi clearly demonstrates that the randomisation technique is lightweight enough to be implemented on the real ASIC that powers Wi-Fi chipset.

Finally, the possibility to drive robots through repeatable trajectories was a very interesting feature that we never used before and that we will probably deploy in our own lab, since we realized during the CSI-MURDER experiment that it provides great help and ease in the collection of results. For the sake of clarity, the robot available in the ORCA facility did not show properties similar to the human body and we would need for the future different robots: nevertheless, working with such robots was fundamental for understanding the difference with respect to object with human-like properties.

What was the direct or indirect value for your company/institution? What is the time frame this value could be incorporated within your current product(s) range or technical solution? Could you apply your results also to other scenarios, products, and industries?

The tools, both hardware (i.e., robots) and software (i.e., OpenWiFi), that the ORCA project made available for the CSI-MURDER experiment turned out to be key for running the research associated with the experiment. As we will keep working on this field and considering that we are just at the beginning of an almost unexplored topic, we believe that the facilities provided by the ORCA testbed had a great value for the future of our institution. This is especially true if we consider the possibility to publish the collected results on future venues like conferences or journal, which is one of the main activities targeted by our institution.

In addition, the knowledge and the skills acquired during the experiment will have a deep impact on our future projects, also those not directly connected with the CSI-MURDER experiment, as we get used to very simple tools that we now consider fundamental for our future activities. We plan, for instance, to bring tools like OpenWiFi in-house: the tool will be valuable also for other people working in Wi-Fi research.

Are there any follow-up activities planned by your company/institution? New projects or funding thanks to this Experiment/Extension? Do you intend to use the ORCA facility, and SW components again in the future?

All the people involved in the CSI-MURDER experiment plan to keep working on the same research line: we intend to push this project both from the theoretical and practical point of view. First, we need to gain a better understanding of the physical principles behind propagation - concepts that we usually use as black-box in our research. We now need to study the black-box and create tools that allow to modify the black-box internals, or better, how they appear to a receiver, for improving the randomisation technique. We also plan to extend the features of the OpenWiFi project and to incorporate a VHT-PHY compliant transceiver inside the FPGA based Zynq board.

It would be extremely interesting to keep using the facilities from the ORCA testbed as many of the hardware components are extremely expensive, not only in terms of cost but also maintenance.

How did ORCA influence your ability to conduct such an Experiment? Which of the enablers and functionalities provided by ORCA seemed the most important one's in the realization of your Experiment.

Prior to the CSI-MURDER project we were mostly limited i) on the transmitter side, and ii) on the repeatability of the experiment. First, the availability of OpenWiFi was key enabler for demonstrating that randomization techniques can be deployed on real ASIC-based nodes like all the commercial Wi-

Fi chipsets. Second, the main limit of our previous approach was in the way we were running experiments: replacing a moving human with a robot will be key for our future activities, even though we need to use special robots carrying volumes with properties similar to human tissues.

If the ORCA facility wasn't available, would it be possible to conduct your Experiment with pre-existing tools, and what would be the overhead in this case compared to your current status?

Without OpenWiFi, demonstrating that randomisation works in full stack Wi-Fi nodes would have not been possible as we were using SDR tools available in our lab like the Ettus N300 in a "raw" fashion. We would have needed to develop the randomisation technique inside our WARP V3 nodes, but given we are "users" of that platform and not developer, this would have required months for analysing where to add the randomisation procedure in the 802.11 stack developed by the Rice University.

Will you keep using ORCA facility, platforms and/or SW components for your future Experiments?

This is very likely. As mentioned before, some of the tools that we have only started using with this Experiment proved fundamental for our research and it would be great to continue using such components. Our plan is to keep working on this research topic in the future and we will surely consider the option of keep using the ORCA facility and platforms. For instance we would really like to test the active attack once implemented inside the OpenWiFi code: once ready we might collect useful data in the ORCA testbed and try submitting another original publication.

2 FEEDBACK TO ORCA

2.1 Testbeds/hardware/software resources used

Please indicate what is used in your experiment or extension in the table below. Please describe in additional paragraph in case the used facility/resource is not listed in the tables.

TESTBEDS	Required (Yes/No)
w.iLab.t (Heterogeneous wireless testbed @ imec, Ghent, Belgium)	yes
IRIS (Software Defined Radio testbed @ TCD, Dublin, Ireland)	
ORBIT (20 x 20 radio grid testbed @ Rutgers University, New Jersey, US)	
IMEC portable testbed	
TUD macro scale testbed (Macro scale testbed @ TUD, Dresden , Germany	
KU Leuven testbed (KU Leuven @Leuven, Belgium)	

SDR HARDWARE PLATFORMS	Number of nodes required
Nutaq ZeptoSDR	
Nutaq picoSDR	
PicoZed Xilinx Zynq®-7000 SoC	
USRP B200-mini	
USRP B210	1
USRP E310	
USRP N210	
USRP X310	
USRP 2920	
USRP 2921	
USRP RIO 2942R	
USRP RIO 2943R	
USRP RIO 2952R (+ GPS)	
USRP RIO 2953R (+ GPS)	
USRP RIO 2953R (+ EBD)	
WARPv2	
Xilinx ZC706 Evaluation Kit - Zynq® 7000 SoC + AD FMCOMM radio frontend	1 with OpenWiFi
ZedBoard Xilinx Zynq®-7000 SoC	
ZedBoard Xilinx Zynq®-7000 SoC + AD FMCOMM radio frontend	

BB – NI PXI 7975 Module	
BB – NI PXI 7965 Module	
FE – NI PXI 5644	
FE – NI PXI 7976R	

ORCA functionalities (categories)	**Used (Yes/No)**
mmWave	
Massive MIMO	
Full Duplex	
PHY& MAC	**yes**
Sensing	
Slicing	
RAT Interworking	
SDR management	
Full stack SDR solutions	**yes**

2.2 Feedback on the usage

2.2.1 Feedback on the testbed and experimentation tools

Please share your experience regarding the testbed usage, such as whether it is easy to get acquainted with the testbed:

- *How did you experience the learning curve regarding using the ORCA facility?*
  - In the end we have been using mobile phones Nexus 6 connected to mobile robots through PC Engines computers, fixed PC Engines computers equipped with QCA Wi-Fi cards, OpenWiFi nodes running on Zynq boards, and a SDR node connected to an Ettus B210.
  - Learning how to use the Zynq boards was somehow time-consuming, but in the end easy thanks to the instructions provided. Learning how to start the proper FPGA image with the randomization mechanism took a little bit. Learning how to use the randomization primitives was straightforward.
  - Learning how to use mobile robot nodes was more difficult: in particular how to keep active the Wi-Fi connection established on the carried PC Engines computers prior to undocking.
  - Learning the procedure for accessing and rooting the mobile phones was long but easy in the end.
  - Rate 4.
- *How do you rate the documentation provided for the testbeds supported in ORCA?*
  - We found the documentation excellent. Sometimes it does not cover some issues, but they turned out to be almost unexpected.

- Rate 5.

- *How did you experience the experimentation tools, such as jFed, or software-defined radio related toolkits.*
  - jFed worked almost flawlessly. We experienced sometimes its inability to mark a node as dead. We do not know if this is pertinent but the interface between jFed and the mobile robots fails sometimes: we underline, maybe the problem is due to some mechanical difficulties during docking that seems to be error-prone.
  - the SDR toolkit that we used was OpenWiFi. It works pretty well except for the Viterbi decoder that times-out after a few hours and require to restart the Zynq-based host.
  - Rate 4
- *Did you make use of all requested testbed infrastructure and hardware resources, as specified in your Open Call proposal? If not, please explain.*
  - Yes.
- *Did you have enough time to conduct your Experiment/Extension in the testbed(s) offered by ORCA?*
  - Regarding this point, there were unexpected delays due to the covid-19 lockdown that almost waisted three months between end of February and end of May when we had again the ability to meet in our lab. The lockdown prevented the possibility to work as we initially planned. The provided extension however almost solved this problem.
  - In any case, it seems that 6 months was really tight, in particular because of the time for understanding how to use the Zynq nodes and the mobile robots, plus the fact that there were some other users accessing the same nodes with medium-long reservations.
  - Rate 4.
- *Were the results of your Experiment/Extension below / in line with / exceeding your initial goals and expectations?*
  - Yes, in the end we were able to demonstrate the feasibility of the proposed experiments.
- *What were the hurdles / bottlenecks? What could not be executed? Was this due to technical limitations? In case you experienced technical limitations, please specify them.*
  - Apart from the issues reported above, that have been always sorted out, we found two minor technical difficulties: i) accessing the remote cameras was not immediately feasible from our lab because we do not have ipv6. We really suggest to switch to ipv4 to avoid any troubles; ii) scp files from local computers to those in the testbed is cumbersome, even if doable by hacking a little bit the ssh command that jFed opens in the local shell. Maybe this command can be modified to always open side-tunnels for easily copying files to/from the remote nodes.
  - Rate 5.
- *How was your experience with particular experimentation tools, such as jFed or remote access of other software toolkits?*
  - Satisfactory.
  - Rate 5.

2.2.2 Feedback on the interactions and communications

Please share your experience with respect to the administrative process, patron communication, and support received from the ORCA consortium:

- *How do you rate the level of work for administration / feedback / writing documents / attending conference calls or meetings compared to the timeframe of the Experiment/Extension?*
    - The submission process was straightforward as well as the procedure for signing the agreement. Same for sending the invoice. All administrative procedures so far have been lightweight and fast.
    - Rate 5.
- *How was your experience concerning the communication and support of your Patron? Is there any other kind of support that you would expect from the patron, which is not available today?*
    - Our patron was great! People in the team helped a lot providing immediate reaction times and effective solutions!
    - Rate 5.

2.2.3 Main added value and what is missing?

Describe why ORCA was useful for your conducting your Experiment/Extension? Which components were perceived as most valuable for you?

Please share your opinion on what you wanted to have, what should be changed or was missing.

The main feature that made ORCA key for the success of the experiment are the mobile robots with the Nexus 6P phones for capturing CSI over repeatable trajectories, and the OpenWiFi modified by the patron that made randomization technique possible.

It would be interesting to repeat the experiments once new PHY are available, like the 11ax that is already in the roadmap of the project.

3 EXPLANATION OF COSTS

The funding from ORCA was completely allocated to cover personnel costs as shown in the table below. The original budget included travel costs (8000 €) to participate in ORCA meetings and possibly to visit the laboratories in Ghent, however due to the Covid-19 pandemics any travel was obviously cancelled, and we dedicated more resources to run experiments. The subdivision of the work lead Francesco Gringoli to allocate about 3.5 months, Renato Lo Cigno 1.5, and Marco Cominelli about 5 months.

	Total PM	**Cost (€)**
(1) Direct personnel costs	8	38000
(2) Other direct costs, of which:		
Travel		
Equipment		
Other goods and services		
(3) Indirect costs		9500
(4) Total costs (Sum of 1, 2 and 3)		47500

The effort was put in learning the management tools of ORCA, designing the obfuscation methodology, implementing the software and the Matlab tools for testing, and obviously to run the experiments interpret the results and prepare this report. It must be highlighted that this report includes results from many experiments run in Brescia too, and not only in w.iLab.2. Most of the time devoted to these experiments has not been covered by ORCA and is not claimed in the table above. Thus, in some sense, CSI-MURDER has been co-financed with internal resources, though without a formal procedure.

Regarding the specific questions posed in the report template we can answer as follows.

- *Was the allocated budget for conducting the Experiment/Extension satisfactory? Was it sufficient in order to allow the successful completion of the Experiment/Extension, or have additional resources been used from your side?*

Yes the budget was sufficient, in particular it was enough to run all the experiments in the ORCA testbeds and to cover the specific efforts needed to port our design and software within the ORCA framework.

- *Did you receive other funding for executing this Experiment/Extension besides the ORCA Open Call (e.g. internal, national, other EC projects, etc.)?*

No, we have not received and additional funding for the execution of this Experiment, though we do put some internal effort (i.e., person months not covered by ORCA funding) for running experiments in Brescia that were necessary to tune the solutions before running them in the ORCA testbeds.

- *Would you also execute this Experiment/Extension without receiving any external funding?*

Yes. We see the advantage of using ORCA testbeds, and in general we are willing to use them without specific funding even if they sometimes introduce an overhead in the workflow. In particular, we are eager to implement the active attacks presented in this report in the OpenWiFi

stack and to experiment with it, this has not been possible during the CSI-MURDER timeframe also due to Covid-19 that complicated work and schedules, but if the OpenWiFi development team is willing to cooperate in the implementation this is an activity that can be pursued.

However, regarding CSI-MURDER, we have also to highlight that the facilities available within the ORCA consortium are not fully up-to-date (regarding the communication hardware in particular, but also the robots available), so that experiments involving localization and localization obfuscation end up in being particularly complex and time consuming.

In conclusion, we appreciate the possibility of using ORCA facilities in general, but, as detailed in Part 2 (Feedback) we think that sometimes the benefits are not a-la-par with the effort, so that its use without specific, dedicated resources (not necessarily coming from the ORCA consortium) may probably not give enough return on the investment needed.

- *Would you even consider paying for conducting such an Experiment/Extension? If so, what do you see as most valuable component(s) to pay for (SDR platforms, SW components, etc.)?*

Even if we perceive the high value of these platforms, as a public University in Italy it is very difficult to find internal resources to invest in external laboratories. Furthermore, we do not normally have an allowance that let us spend for research, rather we are bound to find resources to fund our own PhDs and PostDocs, thus, albeit we see the value of the ORCA infrastructure it would be very difficult to find institutional resources to pay for its use. An exception would obviously be the case when experiments are run in ORCA as part of an industrial or similar project that actually pays for them. In this case the ORCA testbed can become an enabler to execute industry-driven research and as such very valuable.

4 PROMOTIONAL MATERIALS

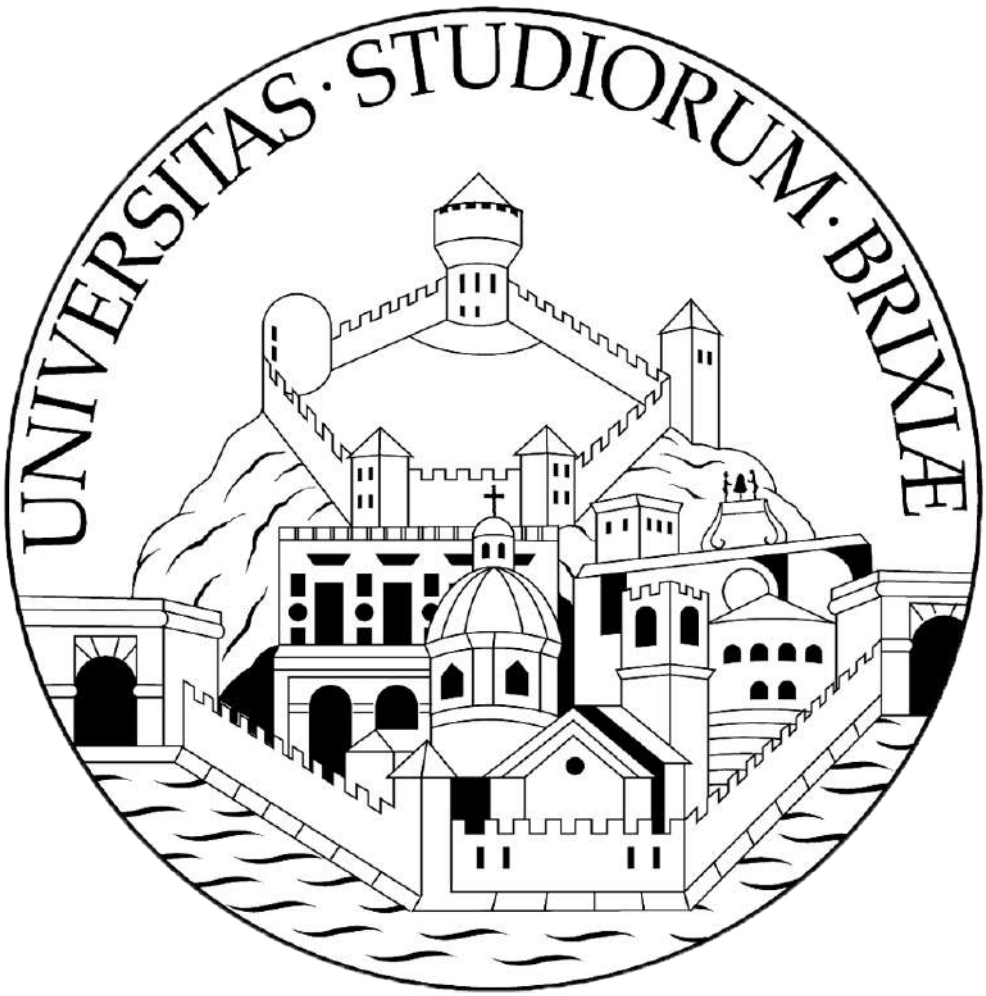
1. *Title of the Experiment*

   Experimental analysis of CSI based anti-sensing techniques (CSI-MURDER)

2. *Name of organisation and logo*

   University of Brescia

   Università degli Studi di Brescia (legal name for use in official documents)



3. *Goal(s) of Experiment/Extension (about 50 words, but definitely not more than 400 characters including spaces)*

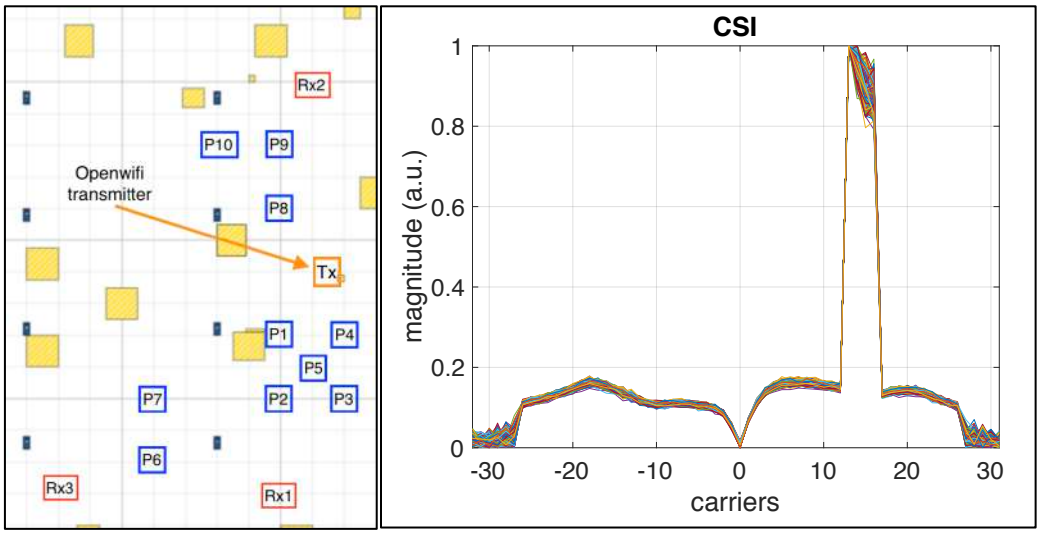
   The goal of the Experiment is to study and propose an anti-sensing technique against novel device-free CSI-based localization frameworks. In particular, we intend to safeguard users' privacy by preventing both passive and active environment sensing attacks without affecting too much the ongoing Wi-Fi communications.

4. *Main challenge(s) of Experiment/Extension (about 50 words, but definitely not more than 400 characters including spaces)*

   Two main challenges:

- Choose from the Wi-Fi sensing literature a passive localization technique and deploy it using lab facilities;
- Find and implement a randomization mechanism at the Wi-Fi physical layer that makes the localization technique above useless without compromising the communication capabilities of the randomized devices, should they be actively adopting randomization or passively being randomized from an external device.

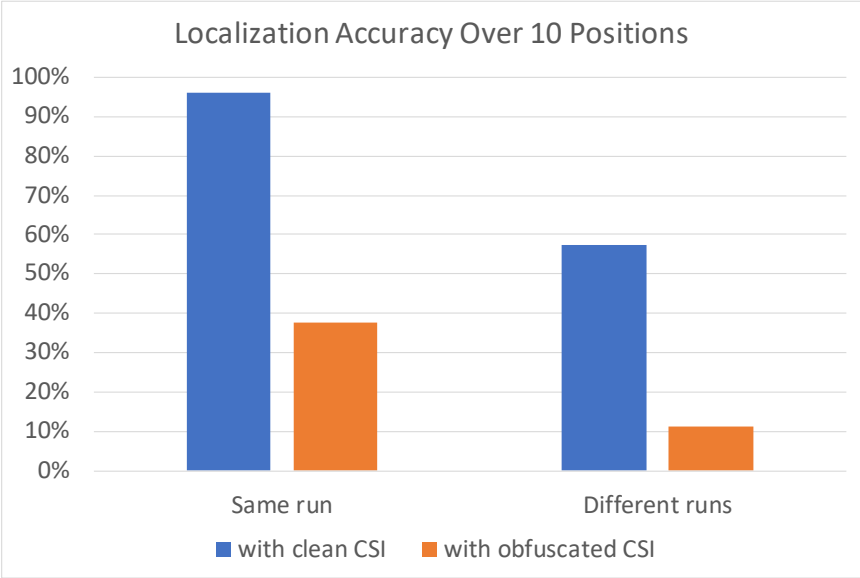
5. *Description of setup of Experiment or concept of Extension, including 1 or maximum 2 figures*



a) b)

The experiment demonstrated that it is possible i) to use the facilities in w.iLab.2 to discover the location of a victim moving in the lab (e.g. among 10 target positions as shown in Figure a) by analyzing the CSI received at a given node; and ii) to adopt a proper countermeasure at the transmitter to make the deployed localization technique useless. In fact, the countermeasure is able to modify the CSI almost arbitrarily. We show the effect of amplifying 4 adjacent subcarriers in Figure b, but in general we can generate random patterns that do not depend on the actual channel condition, so that CSI cannot be used anymore for localization purposes.

6. *Main results, illustrated by 1 or maximum 2 figures with clear, but concise figure captions.*



The Figure shows the average classification accuracy of a person over 10 possible target positions in the w.iLab.2 testbed. The label *same run* refers to the fact that training and testing samples are drawn (without reinsertion) from the same CSI collection experiment, while for label *different runs* we collected training and testing CSI samples at two different times. It is interesting to notice that the localization system still works fairly well in the second case, but more importantly we show that the proposed anti-sensing techniques disrupts localization accuracy in both cases.

7. *Conclusions (about 50 words, but definitely not more than 400 characters including spaces)*

   This is the first study to characterize the possibility of obfuscating Wi-Fi frames to prevent environment sensing. Our experiments in the w.iLab.2 testbed confirm that an eavesdropper is not able to infer the location of a victim in a room, while Wi-Fi communications are preserved. The outcome of this experiment can be used for designing future privacy-aware chipsets.

8. *Feedback (about 50 words, but definitely not more than 400 characters including spaces).*

   Our experience has been positive. Many results in this Experiment could not have been achieved without the tools available in the testbed and the constant support of our patron, which promptly solved a few issues that we encountered while using the facility.

9. *Quote(s): please provide at least one quote we could use for further dissemination activities. By completing the following sentence: "Thanks to the ORCA facility that we were able to... [Other phrasings are also fine.]*

   Thanks to the ORCA facility, we have obtained the necessary resources and support to conduct the first systematic and experimental study of an obfuscation technique to prevent unauthorized use of CSI information to breech people privacy. Such results, beyond opening an entire new field of research, are also fundamental to guarantee the future socio-economic sustainability of Wi-Fi technology.

This information is obviously public.

High-quality logo and figures with sufficiently high resolution should also be provided separately in one of the following formats: jpg, png or pdf. Please make sure that text on figures is still readable when printed on leaflet with A4 format (try to avoid too small fonts).

All the material above can be used to produce leaflets, posters, and any other dissemination and promotional material the ORCA consortium deem fit, also in conjunction with material provided by other experiments. The authors are available to cooperate in the preparation or review of such material to guarantee its adherence to the results obtained and maximize the impact of the ORCA project.

APPENDIX A: CONFUSION MATRICES OF SOME EXPERIMENTS

We report in this appendix some of the confusion matrices relative to results in Section 1.2.3.5 and Section 1.2.3.6 (setup W1 and W2). In each experiment we consider 70 test samples per class. From these data it is easy to determine accuracy and recall for each experiment, although we do not compute them explicitly.

Setup W1: human target, same run

Rx1, clean CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	70	0	0	0	0	0	0	0	0	0	100
P2	0	70	0	0	0	0	0	0	0	0	100
P3	0	0	70	0	0	0	0	0	0	0	100
P4	0	0	0	65	0	0	0	0	0	0	100
P5	0	0	0	0	70	0	0	0	0	0	100
P6	0	0	0	0	0	70	0	1	0	0	98,6
P7	0	0	0	5	0	0	70	0	0	0	93,3
P8	0	0	0	0	0	0	0	69	0	0	100
P9	0	0	0	0	0	0	0	0	70	0	100
P10	0	0	0	0	0	0	0	0	0	70	100
%	100	100	100	92,9	100	100	100	98,6	100	100	**99,1**

Rx1, obfuscated CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	55	0	0	0	0	0	0	0	0	0	100
P2	0	0	0	0	0	29	0	0	10	0	0,0
P3	0	0	70	0	0	0	0	0	0	0	100
P4	0	0	0	0	0	0	0	0	17	16	0,0
P5	13	70	0	0	0	2	0	0	0	0	0,0
P6	0	0	0	0	70	0	0	0	1	0	0,0
P7	2	0	0	70	0	0	70	0	7	0	47,0
P8	0	0	0	0	0	0	0	70	34	0	67,3
P9	0	0	0	0	0	0	0	0	0	0	-
P10	0	0	0	0	0	39	0	0	0	54	58,1
%	78,6	0,0	100	0,0	0,0	0,0	100	100	0,0	77,1	**45,6**

Rx2, clean CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	70	0	0	0	0	0	0	0	0	0	100
P2	0	70	0	0	0	0	0	0	0	0	100
P3	0	0	70	0	0	10	0	0	0	0	87,5
P4	0	0	0	70	1	0	0	0	0	0	98,6
P5	0	0	0	0	69	0	0	0	0	64	51,9
P6	0	0	0	0	0	59	1	0	0	0	98,3
P7	0	0	0	0	0	1	69	0	0	0	98,6
P8	0	0	0	0	0	0	0	70	0	0	100
P9	0	0	0	0	0	0	0	0	70	0	100
P10	0	0	0	0	0	0	0	0	0	6	100
%	100	100	100	100	98,6	84,3	98,6	100	100	8,6	**89,0**

Rx2, obfuscated CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	70	0	0	0	0	9	0	0	0	0	88,6
P2	0	34	0	0	0	28	0	2	0	0	53,1
P3	0	0	0	70	0	1	0	0	0	62	0,0
P4	0	0	0	0	0	0	0	1	0	0	0,0
P5	0	36	0	0	0	1	0	50	0	8	0,0
P6	0	0	70	0	22	0	2	0	70	0	0,0
P7	0	0	0	0	0	0	68	12	0	0	85,0
P8	0	0	0	0	48	31	0	3	0	0	3,7
P9	0	0	0	0	0	0	0	2	0	0	0,0
P10	0	0	0	0	0	0	0	0	0	0	-
%	100	48,6	0,0	0,0	0,0	0,0	97,1	4,3	0,0	0,0	**25,0**

Rx3, clean CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	70	0	0	0	0	0	0	0	0	0	100
P2	0	70	0	0	0	0	0	0	0	0	100
P3	0	0	70	0	0	0	0	0	0	0	100
P4	0	0	0	70	0	0	0	0	0	0	100
P5	0	0	0	0	70	0	0	1	0	0	98,6
P6	0	0	0	0	0	70	0	0	0	0	100
P7	0	0	0	0	0	0	70	0	0	0	100
P8	0	0	0	0	0	0	0	68	0	0	100
P9	0	0	0	0	0	0	0	0	70	0	100
P10	0	0	0	0	0	0	0	1	0	70	98,6
%	100	100	100	100	100	100	100	97,1	100	100	**99,7**

Rx3, obfuscated CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	11	0	0	0	0	0	0	0	0	0	100
P2	0	10	0	0	0	0	0	1	0	0	90,9
P3	0	0	0	0	11	0	0	0	0	0	0,0
P4	0	51	0	70	24	0	0	21	0	0	42,2
P5	0	1	25	0	32	0	0	0	0	0	55,2
P6	34	2	6	0	0	70	0	0	57	0	41,4
P7	0	6	0	0	3	0	70	1	0	59	50,4
P8	25	0	0	0	0	0	0	21	13	0	35,6
P9	0	0	28	0	0	0	0	26	0	0	0,0
P10	0	0	11	0	0	0	0	0	0	11	50,0
%	15,7	14,3	0,0	100	45,7	100	100	30,0	0,0	15,7	**42,1**

Setup W1: human target, different runs

Rx1, clean CSI

Target Class

Predicted Class	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	68,0
P3	20	0	70	0	0	0	9	0	0	0	70,7
P4	3	0	0	70	0	0	17	0	0	0	77,8
P5	0	0	0	0	58	0	0	0	0	0	100
P6	5	0	0	0	12	45	3	0	0	11	59,2
P7	1	0	0	0	0	0	10	0	0	0	90,9
P8	5	0	0	0	0	0	0	54	1	0	90,0
P9	25	0	0	0	0	0	16	16	48	0	45,7
P10	0	0	0	0	0	25	15	0	0	58	59,2
%	0,0	100	100	100	82,9	64,3	14,3	77,1	68,6	82,9	**69,0**

Rx1, obfuscated CSI

Target Class

Predicted Class	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	40,0
P3	0	0	0	0	0	0	0	0	27	0	0,0
P4	0	0	0	68	0	0	0	0	0	0	100
P5	20	0	17	0	0	0	2	0	0	0	0,0
P6	0	18	1	0	0	3	0	13	0	0	8,6
P7	0	0	0	0	0	0	10	57	0	0	14,9
P8	0	37	0	0	0	21	3	0	0	22	0,0
P9	0	2	52	0	70	45	34	0	1	0	0,5
P10	50	11	0	0	0	0	21	0	42	48	27,9
%	0	2,9	0	97,1	0	4,3	14,3	0	1,4	68,6	**18,9**

Rx2, clean CSI

Target Class

Predicted Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	35	0	0	0	0	0	0	6	0	0	85,4
P2	0	70	0	0	0	0	0	0	0	1	98,6
P3	0	0	37	0	0	0	0	0	0	0	100
P4	2	0	0	0	2	0	0	0	0	11	0,0
P5	4	0	2	22	56	0	8	0	1	9	54,9
P6	7	0	8	0	0	37	1	0	0	0	69,8
P7	22	0	23	48	2	33	61	64	8	2	23,2
P8	0	0	0	0	0	0	0	0	0	0	-
P9	0	0	0	0	10	0	0	0	61	0	85,9
P10	0	0	0	0	0	0	0	0	0	47	100
%	50,0	100	52,9	0,0	80,0	52,9	87,1	0,0	87,1	67,1	**57,7**

Rx2, obfuscated CSI

Target Class

Predicted Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	0	0	0	0	12	0	0	0	0	0	0,0
P2	0	6	0	0	0	70	0	0	0	70	4,1
P3	0	63	0	0	58	0	0	0	0	0	0,0
P4	0	0	0	0	0	0	0	0	0	0	-
P5	2	0	70	67	0	0	0	36	35	0	0,0
P6	0	0	0	0	0	0	70	34	0	0	0,0
P7	68	0	0	0	0	0	0	0	0	0	0,0
P8	0	1	0	0	0	0	0	0	0	0	0,0
P9	0	0	0	3	0	0	0	0	35	0	92,1
P10	0	0	0	0	0	0	0	0	0	0	-
%	0,0	8,6	0,0	0,0	0,0	0,0	0,0	0,0	50,0	0,0	**5,9**

Rx3, clean CSI

Target Class

Predicted Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	0	0	0	0	4	0	0	0	0	0	0
P2	54	33	39	0	0	3	16	69	0	0	15,4
P3	1	0	20	0	12	0	0	0	4	0	54,1
P4	0	0	0	70	2	59	6	0	1	0	50,7
P5	6	21	0	0	9	0	0	1	0	0	24,3
P6	1	11	0	0	0	5	0	0	0	0	29,4
P7	0	0	5	0	0	3	48	0	0	0	85,7
P8	0	5	0	0	3	0	0	0	0	0	0
P9	8	0	0	0	36	0	0	0	64	0	59,3
P10	0	0	6	0	4	0	0	0	1	70	86,4
%	0,0	47,1	28,6	100	12,9	7,1	68,6	0,0	91,4	100	**45,6**

Rx3, obfuscated CSI

Target Class

Predicted Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	16	0	0	0	0	1	1	0	0	0	88,9
P2	0	0	0	0	0	0	0	0	0	70	0,0
P3	1	0	0	0	1	0	0	0	0	0	0,0
P4	0	0	0	0	0	0	0	0	0	0	-
P5	0	0	70	70	0	0	0	1	0	0	0,0
P6	47	0	0	0	0	0	0	0	20	0	0,0
P7	3	0	0	0	0	0	1	63	0	0	1,5
P8	1	66	0	0	0	69	53	0	0	0	0,0
P9	2	4	0	0	69	0	15	6	50	0	34,2
P10	0	0	0	0	0	0	0	0	0	0	-
%	22,9	0,0	0,0	0,0	0,0	0,0	1,4	0,0	71,4	0,0	**9,6**

Setup W2: robot target, same run

Rx1, clean CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	70	0	0	0	0	0	0	0	0	0	100
P2	0	70	0	0	0	0	0	0	0	0	100
P3	0	0	70	0	0	0	0	0	0	0	100
P4	0	0	0	70	0	0	0	0	0	0	100
P5	0	0	0	0	67	0	0	0	0	0	100
P6	0	0	0	0	0	67	1	0	0	6	90,5
P7	0	0	0	0	0	3	69	0	0	1	94,5
P8	0	0	0	0	3	0	0	70	0	0	95,9
P9	0	0	0	0	0	0	0	0	70	0	100
P10	0	0	0	0	0	0	0	0	0	63	100
%	100	100	100	100	95,7	95,7	98,6	100	100	90,0	**98,0**

Rx1, obfuscated CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	0	0	0	0	0	0	0	0	0	0	-
P2	0	0	0	0	0	0	0	0	0	0	-
P3	0	0	0	70	0	0	0	0	0	0	0,0
P4	0	55	0	0	0	0	0	0	0	0	0
P5	0	0	0	0	47	0	0	0	0	0	100
P6	0	0	0	0	0	70	0	0	0	0	100
P7	0	0	0	0	1	0	0	0	0	0	0,0
P8	70	0	70	0	0	0	18	0	0	0	0,0
P9	0	0	0	0	22	0	52	70	70	0	32,7
P10	0	15	0	0	0	0	0	0	0	70	82,4
%	0	0,0	0	0	67,1	100	0	0	100	100	**36,7**

Rx2, clean CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	70	0	0	0	0	0	0	0	0	0	100
P2	0	69	0	0	1	0	0	0	0	0	98,6
P3	0	0	70	1	0	41	17	0	0	0	54,3
P4	0	0	0	69	0	0	2	0	0	0	97,2
P5	0	1	0	0	68	0	0	0	0	0	98,6
P6	0	0	0	0	0	18	3	0	0	0	85,7
P7	0	0	0	0	0	9	47	0	0	0	83,9
P8	0	0	0	0	0	0	0	70	0	0	100
P9	0	0	0	0	1	2	1	0	70	0	94,6
P10	0	0	0	0	0	0	0	0	0	70	100
%	100	98,6	100	98,6	97,1	25,7	67,1	100	100	100	**88,7**

Rx2, obfuscated CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	0	0	0	0	0	0	0	0	1	0	0,0
P2	0	0	0	0	0	0	2	0	0	0	0,0
P3	0	0	70	33	0	0	0	0	0	0	68,0
P4	0	0	0	8	0	0	0	0	0	0	100
P5	0	70	0	0	13	0	0	0	0	0	15,7
P6	0	0	0	0	10	70	0	0	1	0	86,4
P7	0	0	0	0	0	0	68	0	44	0	60,7
P8	70	0	0	0	1	0	0	70	2	0	49,0
P9	0	0	0	29	38	0	0	0	20	0	23,0
P10	0	0	0	0	8	0	0	0	2	70	87,5
%	0,0	0,0	100	11,4	18,6	100	97,1	100	28,6	100	**55,6**

Rx3, clean CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	70	0	0	0	0	0	0	0	0	0	100
P2	0	70	0	0	0	0	0	1	0	0	98,6
P3	0	0	70	0	0	0	0	0	0	0	100
P4	0	0	0	70	0	0	0	0	0	0	100
P5	0	0	0	0	70	0	0	0	0	0	100
P6	0	0	0	0	0	70	0	0	0	0	100
P7	0	0	0	0	0	0	70	0	0	0	100
P8	0	0	0	0	0	0	0	69	0	0	100
P9	0	0	0	0	0	0	0	0	70	0	100
P10	0	0	0	0	0	0	0	0	0	70	100
%	100	100	100	100	100	100	100	98,6	100	100	**99,9**

Rx3, obfuscated CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	0	0	0	0	0	0	0	0	0	0	-
P2	0	70	0	0	0	64	0	0	0	0	52,2
P3	0	0	70	0	0	0	0	70	10	0	46,7
P4	0	0	0	17	5	0	0	0	0	0	77,3
P5	0	0	0	0	2	0	0	0	1	0	66,7
P6	13	0	0	0	1	5	0	0	0	0	26,3
P7	13	0	0	0	62	0	68	0	0	0	47,6
P8	0	0	0	53	0	0	0	0	59	0	0,0
P9	0	0	0	0	0	0	0	0	0	0	-
P10	44	0	0	0	0	1	2	0	0	70	59,8
%	0,0	100	100	24,3	2,9	7,1	97,1	0,0	0,0	100	**43,1**

Setup W2: robot target, different runs

Rx1, clean CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	47	0	66	0	0	0	0	0	0	0	41,6
P2	0	70	0	0	0	0	0	23	0	0	75,3
P3	0	0	4	0	69	0	0	0	0	0	5,5
P4	0	0	0	0	0	0	0	0	0	0	-
P5	0	0	0	0	0	0	0	0	0	0	-
P6	0	0	0	0	0	1	19	47	69	51	0,5
P7	22	0	0	0	0	62	47	0	0	1	35,6
P8	0	0	0	0	0	0	0	0	0	0	-
P9	0	0	0	70	1	0	0	0	1	0	1,4
P10	1	0	0	0	0	7	4	0	0	18	60,0
%	67,1	100	5,7	0,0	0,0	1,4	67,1	0,0	1,4	25,7	**26,9**

Rx1, obfuscated CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	2	30	0	0	0	0	0	0	0	12	4,55
P2	0	0	0	0	0	0	0	0	0	6	0,0
P3	14	20	0	0	2	22	0	0	0	0	0,0
P4	2	0	0	31	44	14	0	0	0	0	34,1
P5	0	0	70	0	0	0	0	2	5	52	0,0
P6	23	0	0	0	0	15	0	43	51	0	11,4
P7	0	15	0	0	22	19	0	0	0	0	0,0
P8	0	0	0	0	0	0	0	25	1	0	96,2
P9	1	2	0	9	2	0	0	0	13	0	48,1
P10	28	3	0	30	0	0	70	0	0	0	0,0
%	2,86	0,0	0,0	44,3	0,0	21,4	0,0	35,7	18,6	0,0	**12,3**

Rx2, clean CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	0	0	0	0	1	0	0	0	0	0	0,0
P2	48	53	0	0	0	0	0	0	0	34	39,3
P3	22	3	65	0	0	39	68	0	70	14	23,1
P4	0	0	4	68	0	0	0	0	0	0	94,4
P5	0	12	0	0	26	0	0	0	0	21	44,1
P6	0	2	1	1	17	27	0	0	0	0	56,3
P7	0	0	0	1	13	4	1	0	0	0	5,3
P8	0	0	0	0	0	0	1	0	0	0	0,0
P9	0	0	0	0	13	0	0	56	0	0	0,0
P10	0	0	0	0	0	0	0	14	0	1	6,7
%	0,0	75,7	92,9	97,1	37,1	38,6	1,4	0,0	0,0	1,4	**34,4**

Rx2, obfuscated CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	70	0	0	20	3	0	12	0	4	2	63,1
P2	0	0	0	24	0	0	0	0	0	33	0,0
P3	0	0	0	18	7	70	0	32	0	0	0,0
P4	0	0	0	0	41	0	0	0	0	0	0,0
P5	0	0	70	0	2	0	37	0	1	31	1,4
P6	0	70	0	0	0	0	0	38	0	0	0,0
P7	0	0	0	0	0	0	0	0	0	0	-
P8	0	0	0	0	0	0	1	0	37	0	0,0
P9	0	0	0	1	1	0	0	0	0	1	0,0
P10	0	0	0	7	16	0	20	0	28	3	4,1
%	100	0,0	0,0	0,0	2,9	0,0	0,0	0,0	0,0	4,3	**10,7**

Rx3, clean CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	0	0	3	1	0	0	0	0	22	0	0,0
P2	16	0	0	0	0	0	22	0	41	0	0,0
P3	0	0	67	65	0	0	0	0	0	0	50,8
P4	0	0	0	0	0	0	0	0	0	0	-
P5	0	0	0	0	0	0	0	0	0	0	-
P6	0	0	0	0	70	63	0	0	0	0	47,4
P7	2	59	0	0	0	0	0	0	7	0	0,0
P8	5	0	0	1	0	0	48	5	0	0	8,47
P9	0	0	0	0	0	0	0	0	0	0	-
P10	47	11	0	1	0	7	0	65	0	70	34,8
%	0,0	0,0	95,7	0	0,0	90,0	0,0	7,1	0,0	100	**29,3**

Rx3, obfuscated CSI

Predicted Class \ Target Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	%
P1	0	0	0	0	0	0	0	0	29	0	0,0
P2	0	0	0	0	0	0	0	0	0	0	-
P3	40	20	1	2	0	70	0	0	0	70	0,5
P4	0	0	29	2	0	0	49	51	0	0	1,5
P5	1	0	13	55	16	0	18	0	0	0	15,5
P6	0	0	1	1	0	0	0	0	0	0	0,0
P7	19	50	21	8	0	0	1	4	19	0	0,8
P8	8	0	4	2	53	0	2	15	0	0	17,9
P9	1	0	1	0	1	0	0	0	22	0	88,0
P10	1	0	0	0	0	0	0	0	0	0	0,0
%	0,0	0,0	1,4	2,9	22,9	0,0	1,4	21,4	31,4	0,0	**8,1**

APPENDIX B: MATLAB SIMULATION CODE

We report in this appendix the Matlab code that we wrote for testing the feasibility of the active attack. The code that follows generate a Wi-Fi frame using the Matlab WLAN toolbox, add the artificial "CSI-randomizer" sequence, which is composed by a number of sinusoidal tones, propagate the signal through a random channel, and try to decode it, again using the Matlab WLAN toolbox. It then plots the CSI of each correctly received frame.

Main script code, uses function in script "decodeframe.m" below.

```
BW = 20;                                          % do analysis for 20MHz frames

% create Wi-Fi signal
vhtCfg = wlanVHTConfig;                           % Create packet configuration
vhtCfg.ChannelBandwidth = sprintf('CBW%d', BW);   % 20 MHz channel bandwidth
vhtCfg.NumTransmitAntennas = 1;                   % 1 transmit antenna
vhtCfg.NumSpaceTimeStreams = 1;                   % 1 space-time stream
vhtCfg.GuardInterval = 'LONG';
vhtCfg.MCS = 0;                                   % Modulation: QPSK Rate: 1/2
scramblerInitialization = randi([1 127], 1, 1);   % Initialize scrambler
macCfg = wlanMACFrameConfig('FrameType', 'QoS Data');
macCfg.FrameFormat = 'VHT';                       % Frame format
macCfg.MSDUAggregation = false;                   % do not use AMSDU
macCfg.MPDUAggregation = false;                   % do not use AMPDU
lenByte = 50;                                     % frame length (payload)
symbols = ['0':'9' 'a':'f'];
payload = symbols(randi(numel(symbols), [1 lenByte * 2]));
[macFrame, frameLength] = wlanMACFrame(payload, macCfg, vhtCfg);
vhtCfg.APEPLength = frameLength;
decimalBytes = hex2dec(macFrame);
bitsPerByte = 8;
frameBits = reshape(de2bi(decimalBytes, bitsPerByte)', [], 1);
txWaveform = wlanWaveformGenerator(frameBits, vhtCfg, ...
    'NumPackets', 1,'IdleTime', 0, ...
    'ScramblerInitialization', scramblerInitialization);
txWaveform = transpose(txWaveform);

% Generate CSI modifier signal
noise = 0 * txWaveform;
Tc = 1 / (BW * 1e6);                              % time step
t = (0:length(txWaveform) - 1) * Tc;              % time domain
DELTAF = 312500;                                  % carrier spacing
carriers = [12 13 14 15 16 17 18 ];               % carriers to modify
for carrier = carriers,
  noise = noise + exp(j * 2 * pi * carrier * DELTAF * t);
end;
noise = noise / length(carriers);

% signal to propagate
txsignal = [txWaveform + noise zeros([1 1000])];

% iterate over different random channels and plot received CSI
figure(1); hold on;
cbw = vhtCfg.ChannelBandwidth;
for kk = 1:100,
  tgacChan = wlanTGacChannel('SampleRate',BW*1e6,'ChannelBandwidth',cbw, ...
                             'LargeScaleFadingEffect','Pathloss and shadowing', ...
                             'DelayProfile','Model-D');
  preChSigPwr_dB = 10*log10(mean(abs(txsignal)));
  sigPwr = 10^((preChSigPwr_dB-tgacChan.info.Pathloss)/1);
  chNoise = comm.AWGNChannel( 'NoiseMethod','Signal to noise ratio (SNR)',...
                             'SNR',10,'SignalPower', sigPwr);
  txreceive = transpose(chNoise(tgacChan(transpose(txsignal))));
  try
```

```
    [bytesRecoveredString, eq] = decodeframe(txreceive, sprintf('CBW%d', BW));
    NN = length(eq);
    myfreqs = -NN/2:NN/2 - 1;
    plot(myfreqs, abs(eq));
    drawnow;
  catch
    disp 'cannot decode';
  end;
end;

xl = xlabel('Carrier #'); set(xl, 'FontSize', 18);
yl = ylabel('Magnitude (a.u.)'); set(yl, 'FontSize', 18);
grid on;
```

Function “decodeframe.m” used by the code above.

```
function [bytesRecoveredString, chEstVHTLLTF] = decodeframe(txWaveform, chanBW);

cfgVHTRx = wlanVHTConfig ('ChannelBandwidth', chanBW);
idxLSTF = wlanFieldIndices (cfgVHTRx, 'L-STF');
idxLLTF = wlanFieldIndices (cfgVHTRx, 'L-LTF');
idxLSIG = wlanFieldIndices (cfgVHTRx, 'L-SIG');
idxVHTSIGA = wlanFieldIndices (cfgVHTRx, 'VHT-SIG-A');
idxVHTSTF = wlanFieldIndices (cfgVHTRx, 'VHT-STF');
idxVHTLTF = wlanFieldIndices (cfgVHTRx, 'VHT-LTF');
idxVHTSIGB = wlanFieldIndices (cfgVHTRx, 'VHT-SIG-B');
idxVHTData = wlanFieldIndices (cfgVHTRx, 'VHT-Data');

rx = transpose(txWaveform);
pktOffset = wlanPacketDetect (rx, chanBW, 0);
LSTF = rx (pktOffset + (idxLSTF (1):idxLSTF (2)), :);
coarseFreqOffset = wlanCoarseCFOEstimate (LSTF, chanBW);

% Symbol timing synchronization
LLTFSearchBuffer = rx (pktOffset + (idxLSTF (1):idxLSIG (2)),:);
pktOffset = pktOffset + wlanSymbolTimingEstimate (LLTFSearchBuffer, chanBW);

% Timing synchronization complete: packet detected
fprintf ('Packet detected at index %d\n\n', pktOffset + 1);

% Fine frequency offset estimation using L-LTF
LLTF = rx (pktOffset + (idxLLTF (1):idxLLTF (2)), :);
fineFreqOffset = wlanFineCFOEstimate (LLTF, chanBW);
demodLLTF = wlanLLTFDemodulate (LLTF, chanBW);
chanEstLLTF = wlanLLTFChannelEstimate (demodLLTF, chanBW);

noiseVar = helperNoiseEstimate (demodLLTF);

fmt = wlanFormatDetect (rx (pktOffset + idxLSIG (1):end), chanEstLLTF, noiseVar,
chanBW);
disp([fmt ' format detected']);

[rxLSIGBits, failCheck, eqLSIGSym] = wlanLSIGRecover (rx (pktOffset + (idxLSIG
(1):idxLSIG (2)), :), chanEstLLTF, noiseVar, chanBW);

if failCheck
  disp '** L-SIG check fail **'; return;
end;

[recBits,failCRC,eqSym] = wlanVHTSIGARecover (rx (pktOffset + (idxVHTSIGA (1):
idxVHTSIGA(2)), :), chanEstLLTF, noiseVar, chanBW);
BW = bi2de (double (recBits(1:2).'));

if BW == 3,
  disp 'Supports only 20/40/80MHz';
```

```
  return;
end;
GID = bi2de (double (recBits(5:10).'));
if GID ~= 0 & GID ~= 63,
  disp 'Not SU frame';
  return;
end;
NSTS = bi2de (double (recBits(11:13).'));
if NSTS ~= 0,
  disp 'Only 1 SS supported';
  return;
end;
NSTS = 1;
GUARDINT = bi2de (double (recBits(25).'));
MCS = bi2de (double (recBits(29:32).'));
BEAMFORMED = bi2de (double (recBits(33).'));
if BEAMFORMED,
  disp 'Not compatible with beamforming'; return;
end;
VHTLLTF = rx (pktOffset + (idxVHTLTF (1): idxVHTLTF (2)), :);
demodVHTLLTF = wlanVHTLTFDemodulate (VHTLLTF, chanBW, NSTS);
chEstVHTLLTF = wlanVHTLTFChannelEstimate (demodVHTLLTF, chanBW, NSTS);

[recBits,eqSym]   =   wlanVHTSIGBRecover   (rx   (pktOffset   +   (idxVHTSIGB   (1):
idxVHTSIGB(2)), :), chEstVHTLLTF, noiseVar, chanBW);

if BW == 0, % 20MHz
  LEN = bi2de (double (recBits(1:17).'));
elseif BW == 1, % 40MHz
  LEN = bi2de (double (recBits(1:19).'));
elseif BW == 2, % 80MHz
  LEN = bi2de (double (recBits(1:21).'));
end;

LENBYTE = LEN * 4;
vhtCfgRX = wlanVHTConfig;
vhtCfgRX.ChannelBandwidth = chanBW; % sprintf('CBW%d', chan80';
vhtCfgRX.NumTransmitAntennas = 1;
vhtCfgRX.NumSpaceTimeStreams = 1;
vhtCfgRX.GuardInterval = 'LONG';
if GUARDINT ~= 0,
  vhtCfgRX.GuardInterval = 'SHORT';
  disp 'Using short guard interval';
else
  disp 'Using long guard internval';
end;
vhtCfgRX.MCS = MCS;
vhtCfgRX.APEPLength = LENBYTE;

bitsRecovered = wlanVHTDataRecover(rx (pktOffset + idxVHTData(1):end), chEstVHTLLTF,
noiseVar, vhtCfgRX);
bytesRecovered = bi2de (reshape (bitsRecovered, 8, length(bitsRecovered) / 8)');
bytesRecoveredString = sprintf('%02x', bytesRecovered);
```

REFERENCES

[Abbas2019] Moustafa Abbas, Moustafa Elhamshary, Hamada Rizk, Marwan Torki, and Moustafa Youssef. 2019. WiDeep: WiFi-based Accurate and Robust Indoor Local- ization System using Deep Learning. In *Int. Conf. on Pervasive Computing and Communications (PerCom)*. IEEE, Kyoto, Japan, Mar. 2019, 10.

[Adib2013] Fadel Adib and Dina Katabi. 2013. See through walls with WiFi!. In *Conf. of the Special Interest Group on Data Communication (SIGCOMM)*. ACM, Hong Kong, Aug. 2013, pp. 75–86.

[Cai2018] Chenwei Cai, Li Deng, Mingyang Zheng, and Shufang Li. 2018. PILC: Passive Indoor Localization Based on Convolutional Neural Networks. In *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*. IEEE, Wuhan, China, 6 pages.

[Cominelli2020] Marco Cominelli, Felix Kosterhon, Francesco Gringoli, Renato Lo Cigno, and Arash Asadi. 2020. An Experimental Study of CSI Management to Pre- serve Location Privacy. In *14th International Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization (WiNTECH'20)*, September 21, 2020, London, UK. ACM, 8 pages. https://doi.org/10.1145/3411276.3412187

[Gezici2016] S. Gezici, M. R. Gholami, S. Bayram and M. Jansson. 2016. Jamming of Wireless Localization Systems. In *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2660-2676, June 2016, doi: 10.1109/TCOMM.2016.2558560.

[Gringoli2019] Francesco Gringoli, Matthias Schulz, Jakob Link and Matthias Hollick. 2019. Free Your CSI: A Channel State Information Extraction Platform For Modern Wi-Fi Chipsets. In *13th International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH'19)*, October 25, 2019, Los Cabos, Mexico. ACM, 8 pages.

[Kosterhon2020] Felix Kosterhon. April 2020. Device-Free Indoor Localization: A User-Privacy Perspective. Master's thesis. Technische Universität Darmstadt, Secure Mobile Networking Lab, Department of Computer Science.

[Ma2019] Yongsen Ma, Gang Zhou, and Shuangquan Wang. 2019. WiFi Sensing with Channel State Information: A Survey. *ACM Comput. Surv.* 52, 3, Article 46 (July 2019), 36 pages.

[Ricciato2018] Fabio Ricciato, Savio Sciancalepore, Francesco Gringoli, Nicolò Facchi, and Gennaro Boggia. 2018. Position and Velocity Estimation of a Non-cooperative Source From Asynchronous Packet Arrival Time Measurements. *IEEE Transactions on Mobile Computing*, 17(9), 2018, 2166-2179.

[Sanam2018] Tahsina F. Sanam and Hana Godrich. 2018. An Improved CSI Based Device Free Indoor Localization Using Machine Learning Based Classification Approach. In *26th Eur. Signal Proc. Conf. (EUSIPCO)*. IEEE, Rome, Italy, Sept. 2018, 2390–2394.

[Schulz2017] Matthias Schulz, Daniel Wegemer, and Matthias Hollick. 2017. Nexmon: The C-based Firmware Patching Framework. https://nexmon.org

[Schulz2018] Matthias Schulz, Francesco Gringoli, Jakob Link, and Matthias Hollick. 2018. 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, 256–268.

[Wu2018] Guan-Sian Wu and Po-HsuanTseng. 2018. A Deep Neural Network-Based Indoor Positioning Method using Channel State Information. In *Int. Conf. on Computing, Networking and Comm. (ICNC)*. IEEE, Maui, HI, USA, Mar. 2018, 290–294.