Ultra-Wideband Channel State Information and Localization for Physical Layer Security

Paul Walther
TU Dresden
`paul.walther@tu-dresden.de`

Robert Knauer
TU Dresden
`robert.knauer@tu-dresden.de`

Thorsten Strufe
Karlsruhe Institute of Technology
`strufe@kit.deu`

February 11, 2021

Contents

1 Introduction

As an alternative to classical cryptography, Physical Layer Security (PhySec) provides primitives to achieve fundamental security goals like confidentiality, authentication or key derivation. Through its origins in the field of information theory, these primitives are rigorously analysed and their information theoretic security is proven. Nevertheless, the practical realizations of the different approaches do take certain assumptions about the physical world as granted.

Many PhySec rely on the knowledge of Channel State Information (CSI) or of properties derived from CSI. Examples are wiretap codes and their respective design [3], key derivation based on reciprocal channel properties [13, 12, 15, 4], and physical layer authentication [10, 11]. Despite the fact that many PhySec primitives rely on CSI knowledge for their functionality and security proofs, no definitive dataset is openly available that is suitable for verification and evaluation in practical scenarios. Different dataset are described for specific works, e.g. [15, 2, 5, 7, 9], but none are released for public access.

Hence, we aim to supply a dataset providing real world measurement of channel characteristics in relevant use case scenarios. With this dataset published here, we want to enable the research community to examine the underlying assumptions of PhySec in more detail, as well as provide a way to verify and support theoretical models and approaches with concrete data.

This dataset consists of Ultra-Wideband (UWB) Channel State Information accompanied with the respective location information of the participating terminals. Since we additionally want to enable analysis with regard to security, two eavesdroppers are integrated in each measurement setup, whose observation is also recorded and provided. All measurements are conducted with consumer grade hardware to demonstrate the relevance for actual practical use case.

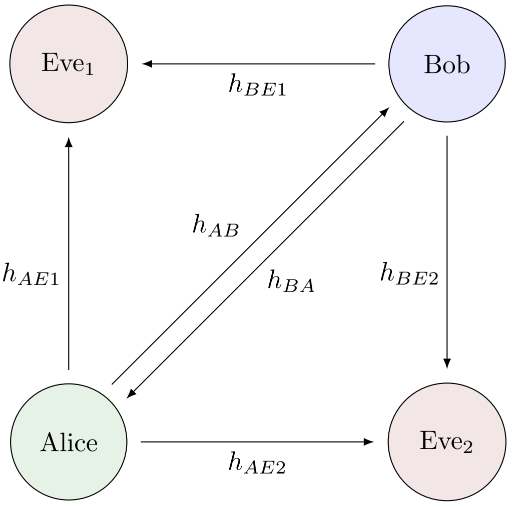


Figure 1: General system setup for all measurements.

To facilitate the unsupervised recording of measurements and thereby allow for long running measurements, we build an robot base moving autonomously within the measurement environment. Thereby, we were able to record measurements for about 112 hours, acquiring ≈ 1.7 million samples, making this dataset also suitable for machine learning methods.

2 Measurement System

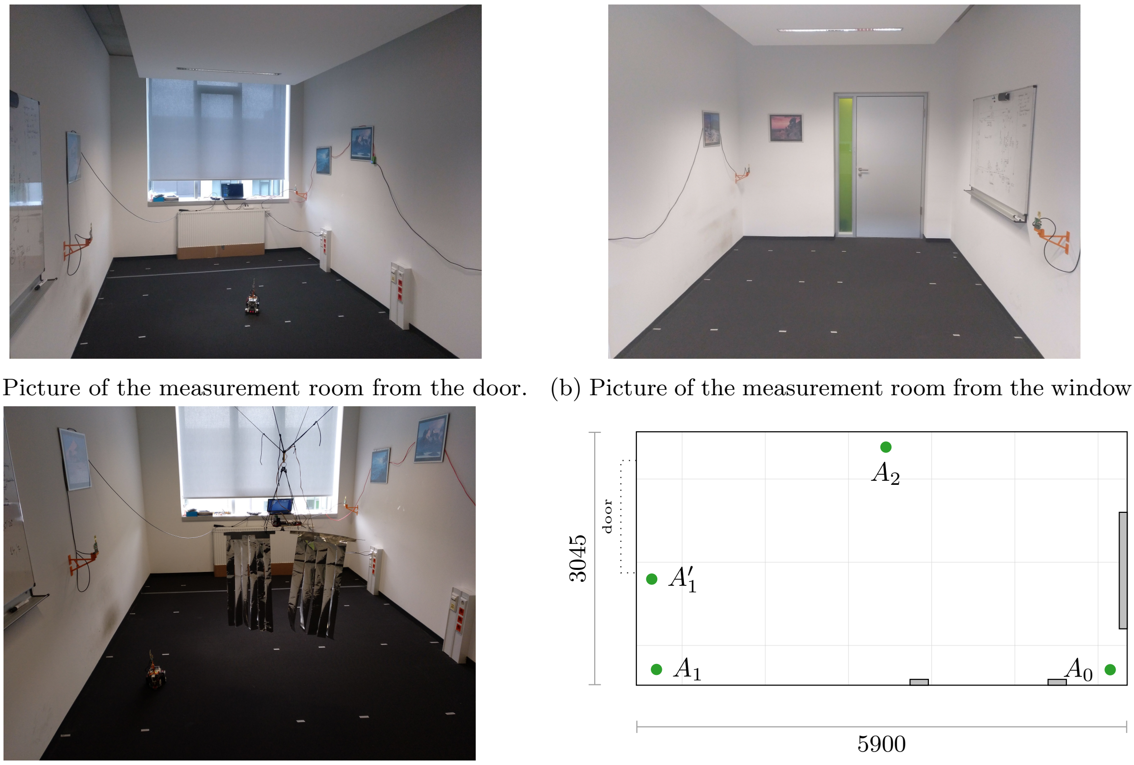
The general **system setup** is oriented at physical layer key derivations schemes, e.g. [4, 15, 6, 14, 13]. This means, we are assuming two legitimate partner, Alice and Bob, which exchange messages in a ping-pong fashion to estimate their the characteristics of their shared, reciprocal channel. Additionally, there is a passive adversary Eve overhearing the communication between Alice and Bob. In our concrete setup, there are two different eavesdropper listening to the communication, which facilitates the analysis of cooperating adversaries. This general system setup is depicted in Fig. 1.

The **measurement system** is designed around the DecaWave EVB1000 evaluation boards, each one equipped with an UWB transceiver, and the Lego Mindstorms Education EV3 kit. The DW1000 is "a fully integrated low power, single chip CMOS radio transceiver IC compliant with the IEEE 802.15.4-2011 ultra-wideband (UWB) standard"[1]. From the factory, the transceiver is calibrated to the integrated channel 2, which operates on a center frequency of 3.9936 GHz and with a bandwidth of 499.2 MHz. Following the IEEE standard, it uses preamble code 4 and a pulse repetition frequency of 16 MHz.

We employ 4 of the EVB1000 boards, which form a typical localization setup: 3 boards are mounted in fixed location acting as *anchors* and 1 board moves within the measurement environment as *tag*. These 4 boards used the *TREK1000* for localization as described in [8]. Additionally, we expanded the respective firmware to acquire the respective channel state estimates, which are taken during every exchange.

The EV3 kit is used to build a self-driving robot platform, which is employed to autonomously move the *tag* in the room. The main components of this platform are: two motors driving the accelerator wheels, a gyroscope an ultra-sonic range sensors, and two touch sensors. The motors are used for the basic movements of the platform. They are individually addressable allowing for rotations. To correctly determine the respective rotation angles, the gyroscope is facilitated. The ultra-sonic range sensor is used for obstacle detection and avoidance. Finally, the two touch sensor provide a backup solution to the obstacle avoidance by sensing possible collisions.

We attached the *tag* board onto the robot using a custom 3D-printed mount. By replacing the standard battery with a high capacity one, the robot could move autonomously for about 12 h.



(a) Picture of the measurement room from the door.

(b) Picture of the measurement room from the window.

(c) Picture of the room with the reflector set up.

(d) Schematic floor plan of the measurement room.

Figure 2: Measurement environment for the data set.

3 Data Acquisition

3.1 Environment

The measurement environment is an typical indoor office room, 5.9 m long and 3.05 m wide as shown in Fig. 2d. Within this measurement room the following obstacles are present:

- the doorway on the right side of the schema is an inset into wall 1350 mm wide and 186 mm deep
- two cable channels at the bottom, each 220 mm wide, 70 mm deep and 600 mm high
- a radiator on the left, which is 1400 mm wide, 95 mm deep and 600 mm high, while sitting sitting 155 mm above the ground

The three *anchors* $A1, A2, A3$ are attached to the walls at 1150 mm height. To realize the recommend distance to the wall of at least 150 mm, we attached them using 3D-printed mounts (visible in Fig. 2a). The coordinates of the anchor position are given in Table 1.

Table 1: Anchor positions in mm measured from lower left corner of the room.

Anchor	x	y
A0	5696	186
A1	242	189
A1'	185	1275
A2	3000	2861

To break the symmetry between $A1$ and $A0$, a subset of the measurement were taken with $A1$ in position $A1'$.

3.2 Procedure

The measurement procedure itself is based on the *TREK 1000* software, provided by Decwave [8]. At its core it is two way ranging procedure, which realizes a *Time-of-Flight* (ToF) based localization. Following this procedure, the respective distances between the *tag* and each *anchor* is acquired. Based on this single distance and the known anchor positions, the tag can perform a multilateration to calculate its own position.

To estimate the respective ToF, the *tag* T broadcasts a start signal. Upon reception of this signal, each *anchor* $A0-2$ immediately replies to the tag. After the final anchor message arrives at the *tag*, it sends a final message to conclude the ranging [8]. The general procedure is depicted in Fig. 3.

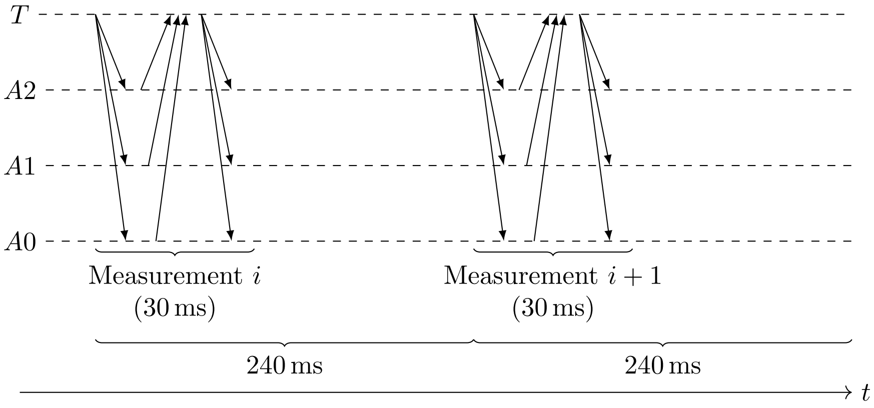


Figure 3: The general measurement procedure for data set.

For our intended use case, we adapted the DW1000 firmware realizing this regular localization procedure in the following ways: Upon reception of any message, all DW1000 chips record the content of the internal registers describing CSI and RSSI of this transmission. The DW1000 calculates RSSI values from the *RX Level* and the *First Path Power Level.* To increase the processing speed, we do not record to full 1016 possible sample of the CSI register. Instead we facilitate the internal first path detection of the DW1000 and record only 50 samples before and 200 sample after the first path. With this adaption, we could reduce the time required for the actual ranging to 30 ms. Additionally, we reduces to time slots for participating anchors from 10 to 3, which allows for faster consecutive measurements every 240 ms.

3.3 Scenarios

Within this room we realized different measurement scenarios. For all scenarios, the robot moves randomly within the room at $0.15\,\mathrm{m\,s^{-1}}$. The pseudo-random movement is realized by stopping and rotating over certain angle at random points in time. For this, each second the robot decides whether to rotate or not with probability $\mathcal{B}(1/60)$, i.e. approximately once in a minute. The rotation angle to rotate is sample from $\mathcal{U}(10°, 90°)$. If the robots detects an obstacle in its way, e.g. a wall, it backs off 10 cm, rotates randomly for $\mathcal{U}(100°, 180°)$ and then proceeds its movement. The following different scenarios were realized:

symmetric This is the basic setting. The robot operates as descibed above. The anchors are positioned at locations $A0$, $A1$ and $A2$. Due to the symmetry of $A0$ and $A1$ this scenario is dubbed "symmetric".

asymmetric The same as above, but $A1$ resides at position $A1'$.

varying speed The overall setting is as in the first one. Instead of fixing the speed at $0.15\,\mathrm{m\,s^{-1}}$, we started at 50% of the robots possible speed, i.e. $0.25\,\mathrm{m\,s^{-1}}$. The speed was then reduced by $0.05\,\mathrm{m\,s^{-1}}$ until it reaches the final value of $0.05\,\mathrm{m\,s^{-1}}$.

reflector The robot moves in the same way as in the asymmetric setting. Additionally, a moving reflector is mounted in the room to induce further interferences (see Fig. 2c). The reflector consists

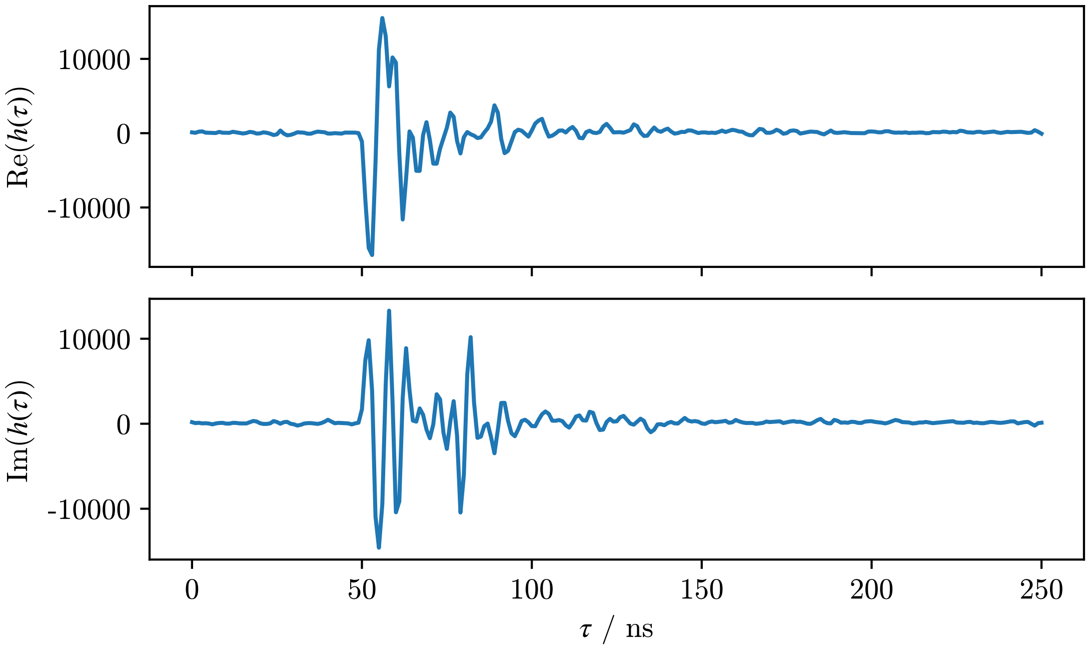


Figure 4: Example of a complex CSI measured using the *EVB1000*.

of 750 mm long aluminium foil strips attached to a 1000 mm long wire. A stepper motor in the middle of the wire rotates the reflector with a constant speed of 2 rotations per minute. The construction is suspended from the ceiling with simple string, so that the wire is at 1200 mm height.

no movement We predefined spots in the room which form a 1 m × 1 m grid. The grid is visualized in Fig. 2d — the grid intersections are the predefined spots. The robot is manually put into the single spots and does not move.

4 Resulting Data

The overall recorded rangings for each scenario is described in Tab. 2.

Table 2: Data recorded in the data set. Every realization consists of 12 impulse responses (3 for each of the 4 terminals) with 251 samples each.

Scenario	Duration	Realizations	Dataset files
symmetric	33 h	467 879	*meas_symm_{1-5}.npz*
asymmetric	19 h	281 595	*meas_asymm_{1,2}.npz*
reflector	20 h	299 730	*meas_asymm_reflector_{1,2}.npz*
var. speed	16 h	237 424	*meas_symm_varspeed_{1,2}.npz*
no movement	2 h	10 836	*meas_{a,}symm_1.npz*
total	90 h	1 297 464	

Each ranging consists of 12 CSI (6 possible channels, each measured reciprocally). Each single CSI consists of 251 samples, taken every 1 ns Fig. 4 shows an example CSI measured using the *EVB1000*.

The data is provided as zipped *NumPy* arrays with custom headers. To load an file the *NumPy* package is required. The respective *loadz* primitive allows for a straight forward loading of the datasets. To load a file “file.npz” the following code is sufficient:

Listing 1: Python code to load a provided dataset.

```
import numpy as np

measurement = np.load('file.npz', allow_pickle=False)
header, data = measurement['header'], measurement['data']
```

The “data” object created by this code contains all measurements stored in the respective file. Every dataset row is compromised of the columns described in the following Tab. 3 (also accessible via the header dictionary).

Table 3: Columns of the single datasets.

Name	Content
timestamp	Timestamp calculated from running index and sampling interval 240 ms. Single float64 value
tag_position	Position of the tag/robot as recorded by the TREK1000 ranging. Tuple of 3 float64
speeds	The 3 configured speeds of the robot, i.e. *forward*, *backward* and *rotation*, each reported as $\mathrm{m\,s^{-1}}$. Dictionary with direction as key and values as float64
cirs	For all possible channels (see header), 251 samples of the CSI, each with real and imaginary part. numpy.array of shape (len(channels),251,2)
rssi	RSSI values as calculated by the EVB1000 boards. Signle int16 value

The respective “header” object contains a description of the respective data rows.

The dataset comes with a supplementary script *example.py* illustrating the basic usage of the dataset.

References

[1] DW1000 Datasheet, 2017.

[2] S. Ben Hamida, J.-B. Pierrot, B. Denis, C. Castelluccia, and B. Uguen. On the Security of UWB Secret Key Generation Methods against Deterministic Channel Prediction Attacks. Sept. 2012.

[3] K.-L. Besser, P.-H. Lin, C. R. Janda, and E. A. Jorswieck. Wiretap code design by neural network autoencoders. *IEEE Transactions on Information Forensics and Security*, 15:3374–3386, 2019.

[4] M. Bloch and J. Barros. *Physical-Layer Security: From Information Theory to Security Engineering*. Cambridge University Press, 2011.

[5] M. Edman, A. Kiayias, and B. Yener. On passive inference attacks against physical-layer key extraction? In *Proceedings of the Fourth European Workshop on System Security*, EUROSEC '11, pages 1–6, Salzburg, Austria, Apr. 2011. Association for Computing Machinery.

[6] C. S. F. Huth. *Physical-Layer Security Architectures for the Internet of Things*. PhD thesis, Ruhr Universität Bonn, 2018.

[7] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy. On the effectiveness of secret key extraction from wireless signal strength in real environments. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, pages 321–332. ACM, 2009.

[8] D. Ltd. DecaRangeRTLS ARM Source Code Guide, 2015.

[9] M. G. Madiseh. Wireless secret key generation versus capable adversaries. 2011.

[10] G. J. Simmons. Authentication theory/coding theory. In *Workshop on the Theory and Application of Cryptographic Techniques*, pages 411–431. Springer, 1984.

[11] G. J. Simmons. A survey of information authentication. *Proceedings of the IEEE*, 76(5):603–620, 1988.

[12] P. Walther, E. Franz, and T. Strufe. Blind Synchronization of Channel Impulse Responses for Channel Reciprocity-based Key Generation. In *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, pages 76–83. IEEE, 2019.

[13] P. Walther, C. Janda, E. Franz, M. Pelka, H. Hellbrück, T. Strufe, and E. Jorswieck. Improving quantization for channel reciprocity based key generation. In *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, pages 545–552. IEEE, 2018.

[14] P. Walther and T. Strufe. Blind Twins: Siamese Networks for Non-Interactive Information Reconciliation. In *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–7. IEEE, 2020.

[15] C. Zenger. *Physical-Layer Security for the Internet of Things.* PhD thesis, Ruhr Universität Bonn, 2017.